Anonymous Whistleblowing over Authenticated Channels

Thomas Agrikola, Geoffroy Couteau, Sven Maier

Anonymous Whistleblowing over Authenticated Channels

Thomas Agrikola[2,⋆], Geoffroy Couteau[1,⋆⋆], and Sven Maier[2,⋆]

[1] CNRS, IRIF, Université de Paris, France geoffroy.couteau@irif.fr
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany {thomas.agrikola,sven.maier}@kit.edu

Abstract. The goal of *anonymous whistleblowing* is to publicly disclose a message while at the same time hiding the identity of the sender in a way that even if suspected of being the sender, this cannot be proven. While many solutions to this problem have been proposed over the years, they all require some form of interaction with trusted or non-colluding parties. In this work, we ask whether this is fundamentally inherent. We put forth the notion of *anonymous transfer* as a primitive allowing to solve this problem *without* relying on any participating trusted parties.

We initiate the theoretical study of this question, and derive negative and positive results on the existence of such a protocol. We refute the feasibility of *asymptotically* secure anonymous transfer, where the message will be received with overwhelming probability while at the same time the identity of the sender remains hidden with overwhelming probability. On the other hand, resorting to *fine-grained* cryptography, we provide a heuristic instantiation (assuming ideal obfuscation) which guarantees that the message will be correctly received with overwhelming probability and the identity of the sender leaks with vanishing probability. Our results provide strong foundations for the study of the possibility of anonymous communications through authenticated channels, an intriguing goal which we believe to be of fundamental interest.

1 Introduction

The term whistleblowing denotes "the disclosure by a person, usually an employee in a government agency or private enterprise, to the public or to those in authority, of mismanagement, corruption, illegality, or some other wrongdoing" [Whi]. Consider the following scenario. You are happily employed by some government agency. However, one day, you learn that your employer violates human rights. You strongly disagree with this breach of trust and law but you are bound by law to keep internal information secret. Consequently, you are faced with a dilemma: either you ignore the human rights violation, or you face dishonorable discharge or even jail. In fact, whistleblowers often take an immense personal risk, and face sentences ranging from exile [BEA14] to incarceration [Phi18] or worse. Whistleblowing is crucial for democracy to educate the public of misdeeds and to call those in power to account. Therefore, it is desirable to cryptographically protect the identity of the whistleblower to allow a low-risk disclosure of wrongdoing.

The importance of this question is well recognized in cryptography and security. It has been the subject of several influential works (e.g. DC-nets [Cha88], Riposte [CBM15] or Blinder [APY20]). Concrete solutions include the use of secure messaging apps [CGCD+20; Ber16], mix-nets [Cha03], onion routing systems such as the Tor network [DMS04], or solutions built on top of DC-nets and secure computation techniques [CBM15; APY20] (see also [ECZ+21; NSSD21]).

Yet, all current approaches to anonymous whistleblowing rely on trusted parties (or non-colluding partially trusted servers), which either receive privately the communication, or implement a distributed protocol to emulate an anonymous network. Therefore, however ingenious and scalable some of these solutions are, whistleblowers must ultimately trust that they will interact with parties or servers which will (at least for some of them) remain honest and refuse to collude throughout the transmission.

In this work, we ask whether this is fundamentally inherent, or whether anonymous whistleblowing is possible in theory without having to privately communicate with trusted parties. In its most basic form, the question we ask is the following:

⋆ Supported by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.
⋆⋆ Supported by ANR SCENE.

Is it possible for a whistleblower (who is communicating solely through authenticated point-to-point or broadcast channels) to publicly reveal some message m while remaining anonymous without assuming trusted participating parties?

We do allow a Common Reference String (CRS) for technical reasons, and stress that while it is technically also a trust assumption, it is much weaker; instead of trusting a set of parties *every time* to follow the exact protocol and to not cheat in any way, we *only* require a CRS to be set up *once*: A CRS that was successfully sampled just once can be used for all future interactions.

The above is, of course, trivially impossible if the whistleblower is the only communicating party. However, it becomes meaningful in a multiparty setting, where a number of parties (unaware of the intent of the whistleblower) exchange innocent-looking messages (think of a group of people having a conversation, or using some public messaging service like Twitter or Facebook to broadcast information). In this context, the question translates as follows: could the whistleblower somehow disguise its communication as an innocent-looking conversation with the other parties, such that the message m can be publicly extracted (by anyone) from the *entire conversation*, yet the identity of which party was indeed the whistleblower remains hidden? To our knowledge, this intriguing question has never been studied in the past. Our main contributions are threefold:

1. **A definitional framework.** We put forth a formal definition for a cryptographic primitive that realizes the above goal, which we call an *Anonymous Transfer*. We study the relation between variants of the notion.
2. **Impossibility results.** We prove a strong impossibility result: we show that Anonymous Transfer with overwhelming correctness and anonymity cannot be realized in any polynomial number of rounds, by exhibiting a general attack against any such protocol. This non-trivial result demonstrates that anonymously communicating over authenticated channels is impossible with standard cryptographic security levels, even assuming strong cryptographic primitives such as ideal obfuscation.
3. **Feasibility result.** We complement our impossibility result by an intriguing *feasibility* result: we show that *fine-grained* Anonymous Transfer is possible assuming ideal obfuscation. The term fine-grained refers to cryptographic constructions which are only guaranteed secure against adversaries whose computational power is a fixed polynomial in the computing power of the honest parties (in our case, the gap is quadratic). Our instantiation is a plausible heuristic candidate when instantiating the ideal obfuscation by candidate indistinguishability obfuscation schemes.

Both our negative and positive results are highly non-trivial and require a very careful analysis. We view our work as addressing a fundamental question regarding the a priori possibility of secure whistleblowing without interacting with trusted parties, through the lens of anonymous communications over authenticated channels. Nevertheless, our study is of a purely theoretical nature, and does not have immediate practical relevance. In particular, we do not compare our results to the practical real-world methods which whistleblowers can employ.

Anonymous Transfer and plausible deniability. The fundamental goal of an Anonymous Transfer protocol is to achieve plausible deniability: the whistleblower should be able to hide its identity among a group of parties, such that even if it is strongly *suspected* that he is the whistleblower, this cannot be *proven* – any party could equally be the whistleblower. Importantly, the involved parties are never required to be aware that a message is being transmitted: their consent or collaboration is not needed for the Anonymous Transfer to take place, and they themselves have no advantage in finding out who the whistleblower was.

1.1 Undetectable Secure Computation

Secure Multiparty Computation (MPC) allows a set of parties to jointly evaluate a function on their inputs without revealing these inputs. In certain scenarios, however, the standard guarantees of MPC become insufficient: the mere fact that a party is participating to a certain protocol already reveals information about that party. Consider for example the following scenario: your company

was hacked, but you do not have enough forensic data to trace the attackers. If several companies fell victim to the same hacker, a joint effort may yield enough information to successfully trace the hacker. However, the very fact that you are *initiating* such a protocol reveals that your company has been hacked.

The notion of Covert Multiparty Computation (CMPC) [vHL05; CGO+07] was introduced to cope with situations in which even revealing one's participation to the protocol is undesirable. CMPC allows a set of parties to securely compute a protocol among n parties with the following two guarantees: (1) If all parties are actually willing to participate in the protocol (and are not simply having innocent conversations), and if the output of the protocol was *acceptable* (which is specified by some function g of the joint input), then everyone learns the result of the protocol. (2) Otherwise (if at least *one* party was not participating, or the output was not acceptable), no one learns anything about who were the participating parties (or even whether there was any).

CMPC is a powerful strengthening of secure computation. However, it still has two important downsides: a single non-participating party is sufficient to make the entire protocol fail (no one gets any output), and when all parties participate, they all learn that they participated (hence, no one can deny anymore having participated in the protocol). One of the primary motivations behind the study of Anonymous Transfer, which we put forth in this work, is to open the avenue to the study of a significantly more powerful form of secure computation that provides the strongest deniability guarantees one can hope for: a secure computation protocol where, even after the successful protocol execution, no one learns *who the participants were*. Specifically, we consider the following setting: N individuals are interacting. Among them, k players are willing to jointly compute a public function f on their private inputs $(x_1, \ldots, x_k)$, while the remaining $(N - k)$ are not interested in taking part to the protocol (nor are even aware of the fact that a secure computation might be taking place). At the end of the protocol, the k participants should all receive the output, but no party should be able to find out which of the parties were actually participating. We call this strengthening of secure computation *undetectable secure computation*.

Since undetectable secure computation is stronger than Anonymous Transfer (which it implies), our impossibility results for Anonymous Transfer also translate to impossibility results for undetectable secure computation[3]. Furthermore, building on our positive result, we show how to construct *anonymous oblivious transfer* (in the fine-grained security setting), a core building block for constructing undetectable secure computation for more general functionalities.

1.2 Defining Anonymous Transfer

An Anonymous Transfer (AT) protocol describes the interaction between a sender, a receiver and a non-participant. We assume all parties to interact in the synchronous model over a public broadcast channel, i.e., in each round each participant broadcasts a message which only depends on messages from previous rounds. The non-participant is not aware that a protocol takes place, and is only having an innocent conversation (we call them the "dummy player", or the "dummy friend"). We follow [vHL05; CGO+07] and model non-participating parties as parties that only broadcast uniform randomness in each round, since any ordinary communication pattern can be viewed as an embedding of the uniform distribution due to standard techniques [vHL05; HLv02; vH04]. The sender aims to transmit a message to the receiver in a way that does not leak its identity (the notion easily generalizes to more non-participating parties). We say that an AT protocol is ε-correct if the probability that the receiver successfully receives the message is at least ε. Further, we say that an AT protocol is δ-anonymous if no adversary is able to determine the identity of the sender (given the transcript and the receiver's random tape) with advantage more than $(1-\delta)/2$ over guessing. These are the core properties which shape an AT protocol. If the protocol allows the receiver to remain silent throughout the protocol execution, sending a message corresponds to publicly revealing the message (*i.e.*, whistleblowing). Eventually, we call *fine-grained* AT an Anonymous Transfer, where anonymity is only required to hold against adversaries from a restricted complexity class (typically, adversaries whose runtime is bounded by a fixed polynomial in the runtime of the honest parties).

[3] This follows directly from the fact that given undetectable secure computation for *any* function f, we can directly construct AT by computing a function that lets two potential senders insert either a bitstring for transfer or $\perp$ and outputs one of them (*i.e.* the one input that is not $\perp$) to the receiver.

1.3 Impossibility Result

Our first main result shows that AT is impossible in a strong sense.

Theorem 1 (Impossibility of AT, informal). *There is no Anonymous Transfer protocol with overwhelming correctness and anonymity, with any polynomial number of rounds and any number $n \geq 1$ of non-participating parties, even for transmitting a single bit message.*

Our proof proceeds in several steps. First, we show that any Anonymous Transfer for transmitting a single bit with n non-participants, with overwhelming correctness and anonymity implies (in a black-box way) a *silent-receiver* Anonymous Transfer (where the receiver never speaks) for transmitting κ *bits* (where κ is some security parameter) with a single non-participating party. This reduction uses a relatively standard indistinguishability-based hybrid argument.

Then, the core of the proof rules out the existence of κ-bit silent-receiver 1-non-participant Anonymous Transfer with overwhelming correctness and anonymity. The key intuition is the following: let $\mathsf{P}_0, \mathsf{P}_1$ be the two parties interacting with the receiver, where P_b is the sender, and P_{1-b} is the non-participant. Let Π^{κ}_{AT} be the protocol which these two parties execute, and assume that it satisfies ε-correctness and δ-anonymity. Suppose that during their interaction, the parties produce a transcript π. We consider an adversary $\mathcal{A}$ which replaces the last message of P_0 by a random value, before running the receiver algorithm to reconstruct the transmitted message. Then if $b = 1$, the adversary just replaced the last (random) message of the non-participating party by another random message, and the transcript is still a perfectly valid transcript for Π^{κ}_{AT}, hence the reconstruction algorithm must still output the right string Σ with overall probability ε. On the other hand, if $b = 0$, then the transcript is a valid transcript for a "round-reduced" version of Π^{κ}_{AT}, where the last round is replaced by two random messages. By the δ-anonymity, $\mathcal{A}$ should not distinguish between the two situations with advantage better than $(1 - \delta)/2$. This implies that the correctness of the round-reduced protocol cannot be much lower than ε, hence that we constructed a δ-anonymous $(c - 1)$-round protocol with non-trivial correctness guarantees. Then, $\mathcal{A}$ keeps repeating this procedure until we reach a 0-round protocol, which cannot possibly have any non-trivial correctness guarantee.

While the above provides an intuition of the approach, the real strategy is much more involved. In particular, using $\mathcal{A}$ to distinguish between a random transcript of Π^{κ}_{AT} and a random round-reduced transcript does not suffice to rule out arbitrary polynomial-round protocols (more precisely, it would only rule out logarithmic-round protocols, since the correctness guarantees would decrease roughly by a factor two at each step of round reduction). Instead, $\mathcal{A}$ will replace independently the last message of each party by a random value, getting two distinct transcripts (π_0, π_1). Then, $\mathcal{A}$ attempts to distinguish whether π_0 is a transcript of Π^{κ}_{AT} and π_1 is a round-reduced transcript, or the other way around. While this is the proper way to attack the protocol, the analysis is more involved, since now π_0, π_1 are not independent random variables anymore, as they share a common prefix (the transcript of the first $c - 1$ rounds). Nevertheless, a more careful analysis shows that this dependency cannot significantly lower the distinguishing probability of $\mathcal{A}$.

In Section 4.3 we further prove that no AT protocol for $N > 3$ parties with overwhelming correctness and anonymity can exist unless a $N = 3$-party protocol exists with overwhelming correctness and anonymity—which cannot exist. It suffices to prove that any N-party Silent Receiver AT for $N > 3$ implies a (N)-party "normal" (*i.e.* with an actively participating receiver) AT, without losing the overwhelming correctness and anonymity in the process.

Intuitively, the receiver does not broadcast any messages in the N-party protocol; all communication comes from the $(N - 1)$ potential senders. We construct an $(N - 1)$-party protocol by letting the receiver play one non-participant, with the one difference being that this party is known not to be the sender (since it is the receiver); the sender can only be one of the $(N - 2)$ other parties. While the correctness remains unaffected, the anonymity decreases due to the fact that *guessing* with one party less yields better results; yet we show that the anonymity still remains overwhelming in the security parameter. We then transform any N-party AT to an N-party SR-AT as described above and that to a $(N - 1)$-party AT, until we have a 3-party AT that, assuming that the N-party AT has overwhelming anonymity and correctness, maintains these properties.

On a high level, this process lets the actual participants *simulate* non-participants behavior in their head; one-by-one their random tape is moved to the CRS until only three parties are left: a sender, a receiver, and a non-participant.

Our negative result applies to a weak model. In particular, non-participants are modeled to be semi-honest. Hence, our negative result does not leave much room for positive results.

1.4 A Candidate Fine-Grained Anonymous Transfer

To circumvent the above impossibility result, we need to give up asymptotic security and resort to the fine-grained setting: We only require anonymity against adversaries which require polynomially—quadratically, in our case—more resources than an honest protocol execution.

That is, our second main result shows that (perhaps surprisingly) non-trivial AT is indeed possible in a weaker setting:

Theorem 2 (Feasibility of AT, informal). *Let $N = 3$ be the number of individuals. Assuming ideal obfuscation, for any anonymity δ, there is a c-round Anonymous Transfer protocol Π^1_{AT} (for ℓ bit messages) that has overwhelming correctness, where anonymity δ holds against any adversary $\mathcal{A}$ with runtime $\ll c^2$.*

That is, for our second main contribution we propose a protocol which—assuming ideal obfuscation—allows to reduce the problem of de-anonymizing the sender to a distribution testing problem. More precisely, we show that determining the real sender in a c-round protocol given only a transcript of the AT protocol is as hard as differentiating between two *Bernoulli* oracles, where one returns 1 with probability p and the other returns 1 with $p + 1/(2c)$. For this distribution testing problem, strong lower bounds on the number of required samples and thus the adversarial runtime are known.

The protocol proceeds in rounds, where each honest message from the sender gradually increases the probability that the transmitted bit is correctly received. The sender first encrypts a verification key that is to-be-used be the obfuscated circuit, and in each successive round the sender encrypts the bit and a signature on both messages from the previous round to limit the ability of the adversary to manipulate the transcript when attacking anonymity. The non-participant only broadcasts random bits in each round. The Common Reference String contains an obfuscated program with hard-coded keys for the pseudorandom encryption scheme. The circuit checks the validity of the signatures of each round. Each consecutive valid round increases the confidence in the transmitted bit. Finally, the circuit outputs random bit according to the confidence gained. If all rounds are valid, the correct bit will be output with probability 1, if no round is valid, the correct bit will be output with probability 0.5.

While the high level intuition of the protocol is relatively clear, its exact instantiation is particularly delicate – any small variant in the design seems to open the avenue to devastating attacks. Furthermore, its analysis relies on long and complex hybrid arguments that progressively reduce the advantage of the adversary to contradictions with respect to known distribution testing bounds with a limited number of samples. In order not to disturb the general reading flow, most of the proof is deferred to the Appendix.

Our proof can be split in two parts. The first part exploits properties of the encryption schemes, the signature scheme, and ideal obfuscation to prove indistinguishability (against even PPT adversaries) between the actual protocol and a hybrid, where all reported messages are truly random and independent from the sender and the transferred bit, and the obfuscated circuit only *counts* how many input messages are identical to those from the challenge transcript.

This game still contains information on the sending party as it treats those messages differently. To remove this dependency, we resort to *distribution testing* and view the obfuscated circuit as a *Bernoulli oracle* which follows one of two (known) distributions, and where the goal is to determine which one.

1.5 Discussions and Implications

In this section, we further discuss some implications and relations of our results to the literature.

'Philosophical implications:' between obfustopia and impossibilitopia. There is a small remaining gap between our negative and positive results: the possibility of building anonymous

transfer secure against arbitrary polytime adversaries, but with non-negligible (e.g. inverse polynomial) anonymity error remains open. Closing this gap would have an intriguing philosophical consequence: stretching the terminology of Impagliazzo on the "worlds" of cryptography, it would establish the existence of a cryptographic primitive that plausibly exists in obfustopia (the world where indistinguishability obfuscation is possible) in the fine-grained setting, yet does not exist ("reside in impossibilitopia") with standard hardness gaps. Interestingly, there are several known examples where fine-grained constructions of a "higher world" primitive reside in a lower world; for example, (exponentially secure) one-way functions (a Minicrypt assumption) imply fine-grained public-key encryption (a Cryptomania assumption). Our work seems to provide a new example of this behavior, at the highest possible level of the hierarchy, showing that impossible primitives might end up existing if we weaken their security to the fine-grained setting.

Relation to the anonymous whistleblowing literature. We clarify how our (positive and negative) results relate to the literature on anonymous broadcast and secure whistleblowing. In general, a whistleblower willing to reveal something anonymously has two alternative choices: (1) the whistleblower has access to an anonymous communication channel, for example by putting their message (say, encrypted with the receiver public key) on some public website that somehow cannot be traced to them. However, access to an anonymous channel is typically a 'physical' assumption, and one which is *very* hard to guarantee. This issue is developed in great detail in the literature: see for example the discussion in Spectrum [NSSD21] about how metadata have been used by federal judges to trace and prosecute people who leaked data through secure messaging apps, or the discussion in Riposte [CBM15] and Express [ECZ+21] on how traffic analysis can be used to trace whistleblowers on the Tor network or the SecureDrop service. Hence, most of the literature focuses on scenario (2): the individuals interact over a communication network, and we do not assume that this network guarantees anonymity in itself. In this case, what we want is to *emulate* this anonymity, by developing a strategy to help the whistleblower transmit a message anonymously to the receiver.

The literature on this subject is incredibly vast, but this emulated anonymity is *always* achieved using the same template in all solutions we are aware of (including Spectrum, Blinder, Riposte, Express, Talek, P3, Pung, Riffle, Atom, XRD, Vuvuzela, Alpenhorn, Stadium (or any other Mixnet-based solution), Karaoke, Dissent, Verdict, and many more): when the whistleblower wants to anonymously transmit a message, either to everyone (anonymous broadcast) or to a target receiver, other users generate 'honest' traffic in which communications can be hidden. To do so, the users interact with a set of *non-colluding* servers (sometimes two servers, sometimes more, some with honest majority, some without). This is never even discussed or remarked: it is taken as an obvious fact that this is *the* structure of an anonymous broadcast (or messaging) protocol. And indeed, the need to generate honest traffic feels clear – if the whistleblower is the sole sender, observing traffic directly leaks their identity. That the use of non-colluding servers was never challenged or even discussed probably means that it also *feels* clear – but this assumption is precisely what we challenge in our work: we do assume that some users generate honest traffic, but we ask whether the assumption of non-colluding participating servers is avoidable. Of course, any scientific treatment of a broad question ('are non-colluding helpful participants required for anonymous broadcast?') is bound to move from the broad question to a formal model, in which (feasibility or impossibility) results can be achieved. Nevertheless, we believe that our impossibility result demonstrates that the use of non-colluding servers in all previous works was indeed unavoidable, at least insofar as their aim was to achieve anonymity against arbitrary polynomial-time adversaries.

Non-participating parties versus malicious parties. Our choice of formalism, with the notion of anonymous transfer, allows to study whether the assumption of honest, non-colluding, participating servers can be replaced by a considerably weaker trust assumption: that of non-participating parties, not trying to take part to the protocol in any way (and not even required to be *aware* of the execution of the protocol) beyond generating traffic. As we show, this weaker assumption does not suffice against arbitrary polynomial-time adversaries, but possibly suffices against bounded polynomial-time adversaries (where the bound is sub-quadratic). As a natural next step, one could push the question even further and ask: what if some of the non-participating parties

were in fact planted by a malicious adversary, and now play *maliciously* during the protocol? It seems plausible, that our general strategy can be extended to deal with malicious non-participants. However, we expect the analysis to require different techniques than the ones we used. We leave a formal proof of this to future work.

1.6 Further Results and Open Questions

In Appendix F, we extend our fine-grained AT such that it transfers ℓ-bit messages directly. To achieve the same level of security as the single-bit AT, this only stretches the number of required rounds by a factor of 2. In Appendix E, we provide an asymptotically secure AT instantiation in the designated-sender setting which achieves non-trivial but not useful parameters for ε and δ. In Section 6, we define an extension of AT called *Strong AT* which we require for Undetectable Computation. We define our novel primitive Undetectable Oblivious Transfer in Section 6 which allows two parties to hide their OT execution in a group of N individuals and instantiate it using strong AT in Appendix G. Finally, in Section 7 we provide definitions for Undetectable Multiparty Computation, which allows a number of k parties to perform an MPC protocol while hiding their identities among N individuals and instantiate UMPC using Undetectable Oblivious Transfer for $k = 3$.

Our work leaves open two exciting questions:

(1) *Can our impossibility result for asymptotically secure AT with overwhelming correctness and anonymity be extended to rule out asymptotically secure AT with anonymity* $1 - 1/\mathsf{poly}(\kappa)$*?*
(2) *Is it possible to instantiate AT in the fine-grained setting from "Obfustopia" standard assumptions achieving similar parameter as our instantiation?*

Given that both our open questions can be answered affirmatively, this would separate the realm of asymptotic security from the realm of fine-grained security.

1.7 Acknowledgements

We thank Rafael Pass for insightful comments and contributions to early stages of this work.

2 Preliminaries

2.1 Notations

For any party P we denote by T_{P} the random tape of P.

For events (A, B), $\bar{A}$ denotes the complementary even of A, $\Pr[A \mid B]$ denotes the probability of A happening conditioned on B happening. For values (a, b), the notation $[\![a = b]\!]$ denote the bit value of the corresponding predicate. We let κ be a security parameter; we write $\mathsf{negl}(\kappa)$ to denote any function negligible in κ and $\mathsf{owhl}(\kappa)$ to denote a function overwhelming in κ (that is, $1 - \mathsf{owhl}(\kappa) = \mathsf{negl}(\kappa)$). For any probability distribution D, we denote by $\mathsf{Supp}(\mathsf{D})$ the support of D, and by $x \xleftarrow{\$} \mathsf{D}$ we denote that x is uniformly sampled from D.

For probability distributions p and q we write $p^{\otimes t}$ as the distribution arising from taking t sample from p, and $p \circ q$ as the distribution obtained by sampling one time from p and one time from q. We write $\|p\|_1$ to denote the L_1 norm of p.

For two bitstrings $A, B \in \{0,1\}^m$, $A \oplus B$ denotes the bitwise XOR of A and B. We write by $[n]$ for $n \in \mathbb{N}$ the set of numbers $\{1, \ldots, n\}$.

A c-round protocol between k parties is defined as the evaluation of the parties' next message algorithms on input of all previously exchanged messages. The output of the protocol is the emerging transcript.

Definition 1 (c-round k-party protocol). *Let $\mathsf{P}_1, \ldots, \mathsf{P}_k$ be PPT algorithms. A* c-round k-party interaction *between $\mathsf{P}_1, \ldots, \mathsf{P}_k$ on input $x \in \{0,1\}^*$ is the sequence of the strings $a_1, a_2, \ldots, a_{c \cdot k} \in$*

$\{0,1\}^m$, *where in each round, parties broadcast:*

$$\begin{aligned}
a_1 &\coloneqq \mathsf{P}_1(x; r_{1,1}) \\
a_2 &\coloneqq \mathsf{P}_2(x; r_{2,1}) \\
&\vdots \\
a_k &\coloneqq \mathsf{P}_k(x; r_{k,1}) \\
a_{k+1} &\coloneqq \mathsf{P}_1(x, a_1, \ldots, a_k; r_{1,2}) \\
&\vdots \\
a_{2k} &\coloneqq \mathsf{P}_k(x, a_1, \ldots, a_k; r_{k,2}) \\
&\vdots \\
a_{c \cdot k} &\coloneqq \mathsf{P}_k(x, a_1, \ldots, a_{c \cdot k-1}; r_{k,c})
\end{aligned}$$

and $r_{i,j}$ *is the random tape used by* P_i *in round* j*. We write*

$$\langle \mathsf{P}_1, \ldots, \mathsf{P}_k \rangle (x; r_{1,1} \| \ldots \| r_{1,c}, \ldots, r_{k,1} \| \ldots \| r_{k,c})$$

to make the used random tape of the parties explicit.

At the end of this document we provide a comprehensive list of symbols and acronyms alongside a detailed glossary.

2.2 Steganography

Informally, steganography describes the art of hiding information in such a way, that no outsider is able to even notice that any information is hidden at all. The concept has been around for a long time, yet it was first brought up as a field of scientific research by Simmons [Sim83] and put into a more complexity-theoretic and cryptographic context by Hopper, Langford, and von Ahn [HLv02]. Simmons [Sim83] described the problem analogously to prison communication: Two prisoners can talk to each other and want to come up with an escape plan, but a warden hears everything they say. The problem of steganography can roughly be described by the following question: How can the two prisoners discuss their escape plan without the warden noticing that the two prisoners are up to something?

While it is true that in most scenarios, the two prisoners could simply encrypt their messages and discuss their escape plan through a secure channel, this method is insufficient for the scenario where a warden is present: If the warden notices that the two prisoners are up to something, they end up in solitary confinement. Thus the problem is much harder than the one considered in *classical* encryption, as the *actual communication itself* needs to be undetectable for the warden. In contrast, it suffices for classical encryption schemes if any other person can *detect* messages as long as they are unable to *decrypt* them. We consider the differentiation to be similar to the one between Multiparty Computation and Covert Multiparty Computation, where the former only hides *information* whereas the latter hides the *entire execution*.

A steganographic channel is modeled as indistinguishable from the *uniform* channel over m bits [HLv02; vH04; vHL05], where each party broadcasts *uniformly random* bitstrings of length m. This is neither a restriction nor an additional trust assumption: steganographic methods use actual sources of randomness like the least significant bit in timestamps or the least significant bits of images (where minor differences can occur due to the noise induced by the camera) that follow two basic rules: (1) when naturally occurring, they are uniformly random, and (2) it is efficiently possible by a sender to change this source such that it has a target value, without acting in a noticeably different way. A random source is suitable for steganography if it fulfills both these requirements: Unaware parties *unknowingly* broadcast random bits automatically, while sending parties can efficiently embed any message.

As a practical example that not only motivates how steganography can be used in practice, but also illustrates the fact that the natural messages are indistinguishable from a sampling of the

uniform channel, think of a public forum where all registered parties can post images of cats. Each post has its timestamp attached (in UNIX-time, *i.e.*, seconds since the first of January in 1970), which is set by the sender during the upload and not by the server. As such, this can be exploited as a steganographic channel: in this (arguably inefficient) steganographic channel the timestamps of each post encodes a single bit, which can be extracted directly by taking the least significant bit of the timestamp.

It is easy to see that under normal circumstances, people who are unaware of the fact that the steganographic channel exists and who only upload images whenever they feel like it can be modeled as parties that only broadcast random bits. Furthermore, the delay of up to one second ends up unnoticed, so it is possible to covertly embed a message by manipulating the least significant bit of the timestamp.

Even though in such forums it might not be the case that all parties send the *same* number of messages, which might be required if we execute a protocol over several rounds by embedding (random looking) messages into the steganographic channel as was done by von Ahn, Hopper, and Langford [vHL05]), this does not pose a restriction. If party P_i has finished a given round by publishing m bits, while P_j still only published $m' < m$ bits, P_i continues as normal and publishes images at arbitrary timestamps until all other parties also broadcast m bits. The protocol then only considers the first m messages for each party in each round and ignores all other messages of the resp. parties until the next round starts.

2.3 Distribution Testing

In this section, we introduce preliminaries for probability testing. We start by describing the *Total Variational Distance* between two distributions.

Definition 2 (Total Variational Distance). *Let p and q be two probability distributions over the countable set of possible outcomes Ω. The* total variational distance *between p and q is defined as:*

$$\mathsf{d}_{\mathsf{TV}}(p,q) \coloneqq \frac{1}{2}\sum_{\omega\in\Omega}|p(\omega)-q(\omega)| = \frac{1}{2}\|p-q\|_1 \tag{1}$$

An important property of the total variational distance is that it acts *sublinear* when taking many samples. When taking t samples from a Bernoulli distribution the corresponding distribution can be described by taking a single sample from a t-bit *Binomial* distribution. The sub-additivity then bounds the total variational distance of the corresponding binomial distribution:

Lemma 1 (Total variational distance of a t-fold probability distribution, folklore). *Let p and q be two Bernoulli distributions with total variational distance $\mathsf{d}_{\mathsf{TV}}(p,q)$. Then it holds for the binomial distributions $p^{\otimes t}$ and $q^{\otimes t}$ that result from sampling t times from the respective distributions:*

$$\mathsf{d}_{\mathsf{TV}}(p^{\otimes t}, q^{\otimes t}) \leq t \cdot \mathsf{d}_{\mathsf{TV}}(p,q) \tag{2}$$

Thus we can bound the distinguishing advantage of *any* distinguisher who has taken t samples form the same oracle using the total variational distance of the respective distributions directly.

A similar rule also holds for two *different* distributions, where the distinguisher has to distinguish whether two samples originate from $p \otimes r$ or from $q \otimes s$ for known values of p, q, r and s. In this case the rule states that:

Lemma 2 (Sub-Additivity of the Total Variational Distance for Product Distributions, folklore). *Let p and q be a probability distribution over $\{0,1\}^m$ with total variational distance $\mathsf{d}_{\mathsf{TV}}(p,q)$. Let r and s be two Bernoulli distributions with total variational distance $\mathsf{d}_{\mathsf{TV}}(r,s)$. Then it holds for the distribution derived from sampling from each distribution once and concatenating the outputs (which yields a sample from $\{0,1\}^{m+1}$ originating either from $p \circ r$ or $q \circ s$) that*

$$\mathsf{d}_{\mathsf{TV}}(p \circ r, q \circ s) \leq \mathsf{d}_{\mathsf{TV}}(p,q) + \mathsf{d}_{\mathsf{TV}}(r,s)$$

The following lemma limits the distinguishing advantage of any distinguisher that tries to distinguish two distributions p and q based on a single sample.

Lemma 3 (Distinguishing distributions based on the Total Variational Distance). *Let p and q be two distributions with total variational distance $\mathsf{d_{TV}}(p,q)$. If $\mathsf{d_{TV}}(p,q) < \frac{1}{3}$, then no algorithm can exist that distinguishes p and q with probability $\geq \frac{2}{3}$ based on a single sample.*

A proof of this lemma is given in Appendix B.

Using Lemmas 1 and 3 we can provide lower bounds on the sampling complexity of distinguishing two distributions p and q with advantage $\alpha/2$.

Corollary 1 (Distinguishing two Bernoulli-Distributions with t samples). *Any distinguisher $\mathcal{D}$ that distinguishes between p and q with probability $\geq \frac{1}{2} + \frac{\alpha}{2}$ requires $t \in \Omega\left(\frac{\alpha}{\mathsf{d_{TV}}(p,q)}\right)$ samples.*

A proof is given in Appendix C.

3 Anonymous Transfer

We consider the following situation: some secret agent P_b is willing to transfer a message Σ to a receiver R, while hiding his identity b among two individuals. We call Anonymous Transfer (AT) an interactive protocol that achieves this goal.

3.1 Network Model and Non-Participating Parties

The goal of an anonymous transfer protocol is to hide the transferred message among innocent conversations by individuals, which are not taking part in the protocol. By a well-established folklore result in steganography (cf. Section 2.2), this task can be reduced to the simpler task of hiding the transferred message among *uniformly random beacons*, broadcast by the other individuals: the uniform channel, where all protocol messages look uniformly random, can be compiled into any other ordinary communication pattern [vHL05; HLv02; vH04]. Therefore, as in previous works (see von Ahn, Hopper, and Langford [vHL05] and Chandran, Goyal, Ostrovsky, and Sahai [CGO+07]), we consider a set of k parties who interact with each other via broadcast channels and focus, without loss of generality, on protocols for the uniform channel. Consequently, we will model the non-participating parties as "dummy parties" that only broadcast uniformly random messages of a fixed length at each round.

3.2 The Model

Let $b \in \{1, \cdots, N-1\}$ denote the index of the sender and let $\Sigma \in \{0,1\}^\ell$ be the message that P_b wants to transfer to the receiver. We consider an interactive protocol in the Common Reference String (CRS) model between N players $(\mathsf{P}_1, \cdots, \mathsf{P}_{N-1}, \mathsf{R})$, where R and P_b participate in the protocol, and P_i for $i \neq b$ are non-participating but present players that only broadcast random strings. The receiver R gets the CRS as input and the sender P_b gets the CRS and the message Σ as input. For any player P, let T_P denote the random tape from which P draws his random coins. The players interact through authenticated broadcast channels in the synchronous model: the protocol proceeds in rounds, and each player broadcasts a message at each round. We denote by $\langle \mathsf{R}, \mathsf{P}_1, \cdots, \mathsf{P}_{N-1} \rangle (crs, b, \Sigma)$ the distribution of the possible transcripts of the protocol in this setting (*i.e.*, the sequence of all messages broadcasted by the players during an execution of the protocol), where the probabilities are taken over the random coins T_P of the players $\mathsf{P} \in \{\mathsf{R}, \mathsf{P}_1, \cdots, \mathsf{P}_{N-1}\}$ and the random choice of the CRS crs.

Definition 3 ($(\varepsilon, \delta, c, \ell)$-Anonymous Transfer). *An N-party $(\varepsilon, \delta, c, \ell)$-Anonymous Transfer (AT) for $\varepsilon, \delta \in \mathbb{R}_{[0,1]}$ and $N, c, \ell \in \mathbb{N}$ (all possibly functions in κ) is a tuple containing three PPT algorithms $(\mathsf{Setup}, \mathsf{Transfer}, \mathsf{Reconstruct})$. The number of rounds in the $\mathsf{Transfer}$ protocol is given as c and the bitlength ℓ defines the length of the transferred message Σ. The algorithms are defined as follows:*

$\mathsf{Setup}(1^\kappa)$ *takes as input the security parameter 1^κ in unary encoding and outputs a Common Reference String crs.*

$\mathsf{Transfer}(crs, b, \Sigma)$ *defines a* c*-round protocol*[4] *that takes as input the Common Reference String* crs*, an index* $b \leq N-1$ *specifying the sender, and the message* $\Sigma \in \{0,1\}^\ell$ *from the sender and outputs a transcript* π*. The non-sender sends independent uniformly distributed noise in each round. All protocol messages sent by the receiver, the sender and the non-participating parties at each round are bitstrings of length* $m = m(\kappa)$*, where* m *is implicitly specified by the* $\mathsf{Transfer}$ *protocol.*

$\mathsf{Reconstruct}(crs, \pi, T_\mathsf{R})$ *is a local algorithm executed by the receiver that takes as input the CRS* crs*, the protocol transcript* π *and the receiver's random tape* T_R *and outputs a message* Σ'*.*

The algorithms additionally satisfy the ε*-correctness and the* δ*-anonymity properties defined in Definitions 4 and 5.*

Definition 4 (ε-Correctness). *For any sufficiently large security parameter* κ*, for any number of individuals* $N \in \mathsf{poly}(\kappa)$*, for any participant* $b \in [N-1]$*, for any message length* $\ell \in \mathsf{poly}(\kappa)$*, for any message* $\Sigma \in \{0,1\}^\ell$*, and for any CRS* $crs \leftarrow \mathsf{Setup}(1^\kappa)$*, an Anonymous Transfer protocol* Π^ℓ_{AT} *between players* $(\mathsf{P}_1, \ldots, \mathsf{P}_{N-1}, \mathsf{R})$ *is* ε-correct *if the following holds:*

$$\Pr\left[\begin{array}{l}\pi \xleftarrow{\$} \mathsf{Transfer}_{\langle \mathsf{R}, \mathsf{P}_1, \ldots, \mathsf{P}_{N-1}\rangle}(crs, b, \Sigma) \\ \Sigma' \leftarrow \mathsf{Reconstruct}(crs, \pi, T_\mathsf{R})\end{array} : \Sigma = \Sigma'\right] \geq \varepsilon \tag{3}$$

Note that ε *can take on any value between* 0 *and* 1*. The naive algorithm that lets the receiver sample a uniformly random* ℓ*-bit string has* $\varepsilon = 1/2^\ell$*.*

Definition 5 (δ-Anonymity). *For any PPT algorithm* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$*, for all sufficiently large security parameters* κ*, for any number of individuals* $N \in \mathsf{poly}(\kappa)$*, and for any message length* $\ell \in \mathsf{poly}(\kappa)$*, an Anonymous Transfer protocol* Π^ℓ_{AT} *between players* $(\mathsf{P}_1, \ldots, \mathsf{P}_{N-1}, \mathsf{R})$ *is* δ-anonymous *if it holds that*

$$\left| \Pr_{b \xleftarrow{\$} [N-1]}\left[\mathrm{Exp}^{anon}_{\Pi^\ell_{AT}, \mathsf{A}, b}(\kappa) = b\right] - \frac{1}{N-1}\right| \leq (1-\delta) \cdot \frac{N-2}{N-1} \tag{4}$$

where $\mathrm{Exp}^{anon}_{\Pi^\ell_{AT}, \mathsf{A}, b}(\kappa)$ *is defined in Fig. 1.*

The value δ can take any value between 0 and 1. The higher δ the stronger the provided anonymity guarantees. If a protocol is $\delta = 1$-anonymous, the advantage over guessing at random equals 0, and if a protocol is $\delta = 0$-anonymous, the advantage over guessing at random equals 1. The right-hand-side of Definition 5 contains a scaling factor of $(N-1)/(N-2)$. This is due to the fact that even under perfect anonymity ($\delta = 1$), the receiver can still *guess* the sender. Knowing that one of the N parties—namely itself—is *not* the sender, there are $(N-1)$ potential senders, of which $(N-2)$ are just dummy friends. Thus, the probability of *guessing wrong* is given by the aforementioned factor.

Note that we require anonymity to hold, in particular, against the receiver. Therefore, the adversary in the anonymity game may know the receiver's random tape T_R from the beginning.

The guessing algorithm is split between A_0 who is given the CRS and the random tape T_R the receiver is going to use during the protocol, and outputs the target message Σ that should be transferred and a state st. In the second phase, the algorithm A_1 which is given the inputs π and the state.

Unless stated otherwise, we consider the case $N = 3$, i.e., one non-participant.

3.3 Fine-grained Anonymous Transfer

Fine-grained cryptographic primitives are only secure against adversaries with an a-priori bounded runtime which is greater than the runtime of the honest algorithms, [Mer78; DVV16]. We use the notion of [DVV16]. In the following, $\mathfrak{C}_1$ and $\mathfrak{C}_2$ are function classes.

[4] A c-round protocol corresponds to a synchronous model, where each message is broadcasted and the messages in each round only depend on messages from previous rounds, see Definition 1 for a formal definition.

$\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},N,b}(\kappa)$

$crs \xleftarrow{\$} \mathsf{Setup}(1^{\kappa})$

$T_{\mathsf{R}} \xleftarrow{\$} \{0,1\}^{\mathsf{poly}(\kappa)}$

$(\Sigma, st) \leftarrow \mathsf{A}_0(crs, T_{\mathsf{R}})$

$\pi \xleftarrow{\$} \mathsf{Transfer}_{\langle \mathsf{R},\mathsf{P}_1,\ldots,\mathsf{P}_{N-1}\rangle}(crs, b, \Sigma; T_{\mathsf{R}}, \cdot, \cdot)$

return $\mathsf{A}_1(\pi, T_{\mathsf{R}}, st)$

Fig. 1: Definition of the game $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},b}(\kappa)$.

Definition 6 ($\mathfrak{C}_1$-fine-grained $(\varepsilon,\delta,c,\ell)$-Anonymous Transfer against $\mathfrak{C}_2$). *The tuple* (Setup, Transfer, Reconstruct) *(as defined in Definition 3) is a* $\mathfrak{C}_1$-fine-grained $(\varepsilon,\delta,c,\ell)$-Anonymous Transfer *for* $\varepsilon,\delta \in \mathbb{R}_{[0,1]}$ *and* $c,\ell \in \mathbb{N}$ *against* $\mathfrak{C}_2$ *if the following two conditions hold:*

Efficiency. *The algorithms* (Setup, Transfer, Reconstruct) *are in* $\mathfrak{C}_1$.
Security. *Anonymity (Definition 5) is only required to hold against adversaries in* $\mathfrak{C}_2$.

The definition of correctness remains as in Definition 4.

Example 1 (Merkle-Puzzles). Merkle-Puzzles [Mer78] are a fine-grained protocol to exchange a shared key from symmetric encryptions where successful encryptions can be efficiently distinguished from false ones. The sender S creates n_{mer} many ciphertexts, each under a different (relatively short) key, containing a unique identifier and a symmetric key. The receiver R then randomly picks one of the ciphertexts and runs a brute-force attack (which we assume to cost m_{mer} many steps) to recover the key and to send the identifier back to the sender.

Here $\mathfrak{C}_1 := \mathcal{O}(n_{\mathrm{mer}} + m_{\mathrm{mer}})$ as the sender has to create n_{mer} puzzles and the receiver must use m_{mer} steps to break one of them, and $\mathfrak{C}_2 := \mathcal{O}(n_{\mathrm{mer}} \cdot m_{\mathrm{mer}})$ as an adversary has to break at worst all the n_{mer} ciphertexts to recover the key.

3.4 Trivial Anonymous Transfers

For simplicity, we focus on 3-party anonymous transfer in the following discussions, with two players $\mathsf{P}_0, \mathsf{P}_1$ and a receiver R.

Remark 1 (Perfect correctness.). A perfectly correct (*i.e.* $\varepsilon = 1$) protocol is impossible. Given a player P_b with input Σ, there is always a probability that the non-participating player P_{1-b} behaves exactly as a participating player with input $\Sigma' \neq \Sigma$, in which case R cannot obtain the correct output for sure.

Therefore, the best one can hope for is a correctness statistically close to 1. In the following, we demonstrate ATs with trivial parameters.

Example 2 (Trivial single-bit AT). Consider the following trivial single-round AT to transfer a single bit σ: P_b broadcasts his input σ (and P_{1-b} broadcasts a random bit). Upon receiving (σ_0, σ_1) from P_0 and P_1, if $\sigma_0 = \sigma_1$, R outputs σ_0; otherwise, R outputs a uniformly random bit. As P_{1-b} broadcasts a random bit, it holds that $\sigma_0 = \sigma_1$ with probability $1/2$, in which case R obtains the correct output $\sigma = \sigma_b$; else, R obtains the correct output with probability $1/2$. Overall, R obtains the correct output with probability $3/4$. The protocol is $1/2$-anonymous since the adversary knows the message to be transmitted and can hence determine the sender whenever the transmitted bits are distinct and guess with probability $1/2$ otherwise. Hence, the above protocol is a $(3/4, 1/2, 1, 1)$-AT.

Example 3 (Trivial ℓ-bit AT). One can also construct a trivial ℓ-bit AT. To transmit a message $\Sigma \in \{0,1\}^{\ell}$: P_b simply sends Σ repeated κ times. Clearly, (not only) R finds out both Σ and b with overwhelming probability. Hence, the above protocol is a $(1 - \mathsf{negl}(\kappa), \mathsf{negl}(\kappa), \kappa \cdot \ell, \ell)$-AT.

In this work, we study whether ATs with non-trivial parameters can exist.

3.5 Reductions Among AT Protocols

In this section, we show that several simplified variants of anonymous transfer are equivalent to the original definition.

AT implies silent-receiver AT. We say that an anonymous transfer has *silent receiver* if the receiver never sends messages during the Transfer protocol, and Reconstruct is a deterministic function of the CRS and the transcript π. Any AT directly implies a silent-receiver AT with the same parameters for correctness and anonymity, but at the cost of secrecy: Any (non-)participant is able to reconstruct the message given only the transcript of broadcasted messages, not just the receiving party of the protocol, which might be undesirable for practical applications. Let Π^{ℓ}_{AT} be a $(\varepsilon, \delta, c, \ell)$-Anonymous Transfer. Define the silent-receiver AT Π^{ℓ}_{SR} as follows:

$\Pi^{\ell}_{SR}.\mathsf{Setup}(1^{\kappa})$ runs $crs \leftarrow \Pi^{\ell}_{AT}.\mathsf{Setup}(1^{\kappa})$ and samples a uniform random tape T_{R} for R. It outputs (crs, T_{R}).

$\Pi^{\ell}_{SR}.\mathsf{Transfer}(crs, b, \Sigma)$ proceeds exactly as $\Pi^{\ell}_{AT}.\mathsf{Transfer}(crs, b, \Sigma)$, except that the receiver R does not broadcast any message. At each round $\chi = 1$ to $\chi = c$, the sender P_b locally appends the χ-th receiver message x_χ in $\Pi^{\ell}_{AT}.\mathsf{Transfer}(crs, b, \Sigma; T_{\mathsf{R}}, \cdot, \cdot)$ to the current transcript $\pi[\chi]$ (note that x_χ can be computed deterministically from $\pi[\chi]$ and T_{R}), and compute its next message as in $\Pi^{\ell}_{AT}.\mathsf{Transfer}$ using the transcript $\pi[\chi] \| x_\chi$.

$\Pi^{\ell}_{SR}.\mathsf{Reconstruct}(crs, \pi, T_{\mathsf{R}})$ is defined exactly as $\Pi^{\ell}_{AT}.\mathsf{Reconstruct}(crs, \pi, T_{\mathsf{R}})$, except that it first expands the transcript π by recomputing (deterministically) the messages of R in $\Pi^{\ell}_{AT}.\mathsf{Transfer}(crs, b, \Sigma; T_{\mathsf{R}}, \cdot, \cdot)$ and appending them to π at each round.

The notion of silent receiver AT captures the notion of an anonymous transfer whose aim is to *publicly reveal* a message (*i.e.*, whistleblowing) rather than sending it to a single receiver. An other way to look at it is to consider that the silent receiver transformation can be seen as *passive to active security transformation* for the receiver: If there is a secure AT protocol against a *passive* receiver, then there is a secure silent receiver AT against an *active* receiver, simply because the receiver has no option to cheat as no messages are sent.

Lemma 4. *Π^{ℓ}_{SR} is an $(\varepsilon, \delta, c, \ell)$-Anonymous Transfer.*

Proof (sketch). Correctness and number of rounds follow directly from the description of Π^{ℓ}_{SR}, which simply mimics Π^{ℓ}_{AT}, except that the random tape of the receiver is made public, and its messages are computed on the fly locally by the sender and during the reconstruction. Anonymity follows also immediately by observing that T_{R} is given to the adversary in the anonymity game, hence making it public cannot harm anonymity. □

Since the converse direction is straightforward, AT and silent receiver AT are therefore equivalent.

Single-bit AT implies many-bit AT. In this section, we analyze how a single-bit AT can be generically transformed into an AT which allows to transmit bitstrings. We construct an ℓ-bit AT by executing the single-bit AT ℓ times (sequentially) to transmit the message bit-by-bit. Let Π^{1}_{AT} be a $\mathfrak{C}_1$-fine-grained-$(\varepsilon, \delta, c, 1)$-Anonymous Transfer against $\mathfrak{C}_2$. Further, let Π^{ℓ}_{AT} be the protocol which uses ℓ instances of Π^{1}_{AT} to transmit ℓ-bit messages bit-by-bit.

We analyze Π^{ℓ}_{AT} using the fine-grained definition. The results directly apply using asymptotic security.

Lemma 5. *Let Π^{1}_{AT} be a $\mathfrak{C}_1$-fine-grained $(\varepsilon, \delta, c, 1)$-Anonymous Transfer against $\mathfrak{C}_2$. Then, the protocol Π^{ℓ}_{AT} is a $\mathfrak{C}'_1 := \mathfrak{C}_1 \cdot \ell$-fine-grained $(\varepsilon', \delta', c \cdot \ell, \ell)$-AT against $\mathfrak{C}'_2 := \mathfrak{C}_2 - \ell \cdot \mathfrak{C}_1$, where $\varepsilon' = \varepsilon^{\ell}$ and $\delta' = (\delta\ell - \ell - \delta + 2)$.*[5]

Proof. For $\Sigma \in \{0,1\}^{\ell}$, we have $\varepsilon' = \Pr_{crs, \pi, \Sigma'}[\Sigma = \Sigma'] = \varepsilon^{\ell}$.

For the purpose of avoiding notational overhead, we prove anonymity for $N = 3$ parties, i.e., for one non-participant. The general case follows by generalizing notation. Let A be an adversary against

[5] We slightly abuse notation but we believe the meaning to be clear.

the anonymity of Π^{ℓ}_{AT}. We define a sequence of hybrid games $\mathsf{H}_1, \ldots, \mathsf{H}_\ell$ between $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},0}(\kappa)$ and $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},1}(\kappa)$ in Fig. 2. H_1 is identical to $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},1}(\kappa)$ and H_ℓ is identical to $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},0}(\kappa)$.

We construct an adversary B against the anonymity of Π^{1}_{AT} in Fig. 2. If B plays $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{1}_{AT},\mathsf{B},0}(\kappa)$, then B simulates H_{i+1} for A. Otherwise, if B plays $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{1}_{AT},\mathsf{B},1}(\kappa)$, then B simulates H_i for A.

H_i

for $j \in [\ell]$ **do**
  $crs_j \leftarrow \mathsf{Setup}(1^\kappa)$
$crs' := (crs_1, \ldots, crs_\ell)$
$T'_{\mathsf{R}} := (T_{\mathsf{R},1}, \ldots, T_{\mathsf{R},\ell}) \leftarrow (\{0,1\}^{\mathsf{poly}(\kappa)})^\ell$
$(\Sigma, st_{\mathsf{A}}) \leftarrow \mathsf{A}_0(crs', T'_{\mathsf{R}})$
for $j \in \{1, \ldots, i-1\}$ **do**
  $\pi_j \leftarrow \mathsf{Transfer}(crs, 0, \Sigma[j]; T_{\mathsf{R},i}, \cdot, \cdot)$
for $j \in \{i, \ldots, \ell\}$ **do**
  $\pi_j \leftarrow \mathsf{Transfer}(crs, 1, \Sigma[j]; T_{\mathsf{R},i}, \cdot, \cdot)$
return $\mathsf{A}_1((\pi_1, \ldots, \pi_\ell), st_{\mathsf{A}})$

$\mathsf{B}_0(crs, T_{\mathsf{R}})$

$i \leftarrow \{1, \ldots, \ell-1\}$
for $j \in [\ell] \setminus \{i\}$ **do**
  $crs_j \leftarrow \mathsf{Setup}(1^\kappa)$
  $T_{\mathsf{R},j} \leftarrow \{0,1\}^{\mathsf{poly}(\kappa)}$
$crs_i := crs$, $T_{\mathsf{R},i} := T_{\mathsf{R}}$
$crs' := (crs_1, \ldots, crs_\ell)$
$T'_{\mathsf{R}} := (T_{\mathsf{R},1}, \ldots, T_{\mathsf{R},\ell})$
$(\Sigma, st_{\mathsf{A}}) \leftarrow \mathsf{A}_0(crs', T'_{\mathsf{R}})$
$st := (\Sigma, i, st_{\mathsf{A}})$
return $(\Sigma[i], st)$

$\mathsf{B}_1(\pi, st)$

parse $st =: (\Sigma, i, st_{\mathsf{A}})$
for $j \in \{1, \ldots, i-1\}$ **do**
  $\pi_j \leftarrow \mathsf{Transfer}(crs, 0, \Sigma[j]; T_{\mathsf{R},j}, \cdot, \cdot)$
for $j \in \{i+1, \ldots, \ell\}$ **do**
  $\pi_j \leftarrow \mathsf{Transfer}(crs, 1, \Sigma[j]; T_{\mathsf{R},j}, \cdot, \cdot)$
$\pi_i := \pi$
return $\mathsf{A}_1((\pi_1, \ldots, \pi_\ell), st_{\mathsf{A}})$

Fig. 2: Hybrid games for the expansion of single-bit AT to multi-bit AT (left) and the adversary (middle and right).

Provided that B is in $\mathfrak{C}_2$, we have

$$
\begin{aligned}
1-\delta &\geq \left|\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{B},0}(\kappa)] - \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{B},1}(\kappa)]\right| \\
&= \sum_{j=1}^{\ell-1}\Big(\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{B},0}(\kappa) \wedge i = j] - \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{B},1}(\kappa) \wedge i = j]\Big) \\
&= \frac{1}{\ell-1}\sum_{j=1}^{\ell-1}\left(\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{B},0}(\kappa) | i = j] - \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{B},1}(\kappa) | i = j]\right) \\
&= \frac{1}{\ell-1}\sum_{j=1}^{\ell-1}\left(\Pr[\mathsf{H}_{j+1}] - \Pr[\mathsf{H}_j]\right) \\
&= \frac{1}{\ell-1}\left(\Pr[\mathsf{H}_\ell] - \Pr[\mathsf{H}_1]\right) = \frac{1}{\ell-1}\Big(\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},0}(\kappa)] - \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},1}(\kappa)]\Big)
\end{aligned}
$$

We have that $\mathfrak{T}(\mathsf{B}) = \mathfrak{T}(\mathsf{A}) + (\ell-1)\cdot\mathfrak{C}_1 = \mathfrak{T}(\mathsf{A}) + \ell\cdot\mathfrak{C}_1$. Hence, given that $\mathfrak{T}(\mathsf{A}) = \mathfrak{T}(\mathsf{B}) - \mathfrak{C}_1 \in \mathfrak{C}_2 - \ell\cdot\mathfrak{C}_1$, the anonymity advantage of A is $(1-\delta)(\ell-1)/2$, yielding anonymity of $\delta' = \delta\ell - \ell - \delta + 2$. □

4 Impossibility of Anonymous Transfer

In this section, we prove that no anonymous transfer protocol, with an arbitrary polynomial number of rounds, can simultaneously enjoy overwhelming correctness ($\varepsilon = 1 - \mathsf{negl}(\kappa)$) and overwhelming anonymity ($\delta = 1 - \mathsf{negl}(\kappa)$), even for transmitting single bit messages.

Theorem 3 (Impossibility of AT). *Let $\mu : \mathbb{N} \mapsto \mathbb{R}$ be any negligible function and p be any polynomial. There is no $(1-\mu(\kappa), 1-\mu(\kappa), p(\kappa), 1)$-Anonymous Transfer, for any number of parties.*

Theorem 3 will follow as a corollary from a more general result bounding the relation between ε and δ in any c-round protocol. Throughout this section we will focus on $N = 3$, that is, the case with one dummy player. This is without loss of generality as we will show in Section 4.3 that any N-party anonymous transfer with $N > 3$ implies in particular a 3-party anonymous transfer, for which we will show here that it can not exist.

Attacker $\mathcal{A}^t = (\mathsf{A}_0^t, \mathsf{A}_1^t)$

Algorithm A_0^t

- On input crs, sample $(\Sigma_1, \cdots, \Sigma_t) \xleftarrow{\$} \mathcal{DR}$, and set Σ^t to be an arbitrary element of $\{0,1\}^\kappa \setminus \{\Sigma_1, \cdots, \Sigma_t\}$ (which exists since $t \ll 2^\kappa$).
- Output $(\Sigma^t, st = (crs, \Sigma^t))$.

Algorithm A_1^t

- On input Σ, st, parse st as (crs, Σ^t) and π as a triple $(\pi[c-1], x_0, x_1)$, where $\pi[c-1]$ is a transcript for the first $c-1$ rounds (if $c = 1$, it is the empty string), and $(x_0, x_1) \in \{0,1\}^m \times \{0,1\}^m$ are the last-round messages from P_0 and P_1 respectively.
- Pick $(x_0', x_1') \xleftarrow{\$} \{0,1\}^m \times \{0,1\}^m$, set $\pi_0 \leftarrow (\pi[c-1], x_0, x_1')$, $\pi_1 \leftarrow (\pi[c-1], x_0', x_1)$, and compute $\Sigma_{b^*}' \leftarrow \mathsf{Reconstruct}(crs, \pi_{b^*})$ for $b^* = 0, 1$.
- Return the following:
  - if $\Sigma_0' = \Sigma^t$, output 0;
  - else, if $\Sigma_1' = \Sigma^t$, output 1;
  - else, return a uniformly random bit $b' \xleftarrow{\$} \{0,1\}$.

Fig. 3: Attacker $\mathcal{A}^t$ against the δ-anonymity of the silent-receiver κ-bit AT protocol Π_{AT}^κ, parameterized by a polynomial $t = t(\kappa)$.

4.1 The Attacker

From now on, we focus on building a generic attack against 3-party *silent-receiver* anonymous transfer for κ-bit messages. The theorem will follow from the reductions from 1-bit anonymous transfer to multibit silent-receiver anonymous transfer described in Section 3.5.

Let Π_{AT}^κ be a silent-receiver $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer. Let $m = m(\kappa)$ be the bitlength of the message from the non-participating party. Let Rand denote the following procedure: on input a transcript π of Π_{AT}^κ, $\mathsf{Rand}(\pi)$ truncates π to $c-1$ rounds of the AT protocol, and replaces the messages of the last round by two uniformly random length-m bitstrings[6]. It outputs the new rerandomized transcript π'. For every $\Sigma \in \{0,1\}^\kappa$ and $b \in \{0,1\}$, we let $\mathcal{D}_{b,\Sigma}, \mathcal{D}'_{b,\Sigma}, \mathcal{DR}$ denote the following distribution:

$$\mathcal{D}_{b,\Sigma} = \{\Sigma' \ : \ crs \leftarrow \mathsf{Setup}(1^\kappa), \pi \leftarrow \mathsf{Transfer}(b, \Sigma), \Sigma' \leftarrow \mathsf{Reconstruct}(crs, \pi)\}$$
$$\mathcal{D}'_{b,\Sigma} = \{\Sigma' \ : \ crs \leftarrow \mathsf{Setup}(1^\kappa), \pi' \leftarrow \mathsf{Rand}(\mathsf{Transfer}(b, \Sigma)), \Sigma' \leftarrow \mathsf{Reconstruct}(crs, \Sigma')\}$$
$$\mathcal{DR} = \{\Sigma' \ : \ crs \leftarrow \mathsf{Setup}(1^\kappa), \pi \xleftarrow{\$} (\{0,1\}^m \times \{0,1\}^m)^c, \Sigma' \leftarrow \mathsf{Reconstruct}(crs, \pi)\}$$

Fix an arbitrary polynomial t. We define an attacker $\mathcal{A}^t = (\mathsf{A}_0^t, \mathsf{A}_1^t)$ against the anonymity of Π_{AT}^κ, parameterized by the polynomial t, on Figure 3. In the following, we will not use $\mathcal{A}^t$ directly to attack the full c-round protocol: rather, we will use $\mathcal{A}^t$ as a distinguisher between the c-round protocol Π_{AT}^κ, and the $(c-1)$-round protocol obtained by running Π_{AT}^κ for $(c-1)$ rounds, and replacing the messages of the last round by uniformly random m-bit strings. From there, the proof of impossibility will proceed by induction; we refer the reader to the introduction for a high-level intuition of our proof.

Base case: advantage of $\mathcal{A}^t$ when $c = 1$. We start the induction by bounding the advantage of $\mathcal{A}^t$ in the anonymity game when Π_{AT}^κ is non-interactive (i.e., $\mathsf{Transfer}$ consists of a single message from each of $\mathsf{P}_0, \mathsf{P}_1$ to the receiver). Before proceeding, we make two key observations:

(1) When $c = 1$, $\mathcal{D}'_{b,\Sigma} = \mathcal{DR}$ for any (b, Σ). In particular, this means that $\mathcal{D}'_{b,\Sigma}$ is independent of (b, Σ).

[6] Since the protocol is silent-receiver, there is no message from the receiver; furthermore, assuming that the sender message is m-bit is without loss of generality, since otherwise the protocol is trivially not anonymous.

(2) When $c = 1$ and $b = 0$, the distribution of the values (Σ'_0, Σ'_1) constructed by A_1^t given as input a random transcript $\pi \leftarrow \mathsf{Transfer}(0, \Sigma^t)$ is exactly the distribution $\mathcal{D}_{0,\Sigma^t} \times \mathcal{DR}$. This is because x_0 is a random message from the sender with input $b = 0$ and value Σ^t, and (x_1, x'_0, x'_1) are three uniformly random elements of $\{0,1\}^m$, hence (x_0, x'_1) is exactly a random transcript of Π^κ_{AT} with (b, Σ^t), while (x'_0, x_1) is just a pair of random messages. Similarly, if $b = 1$, the distribution of the values (Σ'_0, Σ'_1) constructed by A_1^t given as input a random transcript $\pi \leftarrow \mathsf{Transfer}(1, \Sigma^t)$ is exactly the distribution $\mathcal{DR} \times \mathcal{D}_{1,\Sigma^t}$.

Both observations follow directly from the definitions of $\mathcal{D}_{b,\Sigma}, \mathcal{D}'_{b,\Sigma}, \mathcal{DR}$ and of A_1^t. Building on the above observations, we show that for an appropriate choice of t, the advantage of $\mathcal{A}^t$ in the anonymity game can be made arbitrarily close to $(\varepsilon - 1)/2$:

Claim. For any polynomial n, there is a polynomial t such that

$$\left| \Pr_{b \xleftarrow{\$} \{0,1\}} \left[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT}, \mathcal{A}^t, b}(\kappa) = b \right] - 1/2 \right| \geq \frac{\varepsilon}{2} - \frac{1}{n}, \tag{5}$$

which implies that any silent-receiver $(\varepsilon, \delta, 1, \kappa)$-Anonymous Transfer must satisfy $\delta \leq 1 - \varepsilon + 2/n$ for any polynomial n; equivalently, $\delta \leq 1 - \varepsilon + \mathsf{negl}(\kappa)$. In particular, this means that if the AT has overwhelming correctness ($\varepsilon = 1 - \mathsf{negl}(\kappa)$), then δ must be negligible.

Proof. Let $t \coloneqq \kappa \cdot n$, and let $\mathsf{Bad} \subset \{0,1\}^\kappa$ denote the set

$$\mathsf{Bad} = \left\{ \Sigma \in \{0,1\}^\kappa \ : \ \Pr_{\Sigma' \xleftarrow{\$} \mathcal{D}'} [\Sigma' = \Sigma] > \frac{2}{n} \right\}.$$

We first show that

$$\Pr_{\Sigma^t \xleftarrow{\$} \mathsf{A}_0^t(crs)} \left[\Sigma^t \in \mathsf{Bad} \right] \leq e^{-\kappa}.$$

Fix any $\Sigma \in \mathsf{Bad}$. Then

$$\begin{aligned} \Pr_{\Sigma^t \xleftarrow{\$} \mathsf{A}_0^t(crs)} [\Sigma = \Sigma^t] &\leq \Pr_{\Sigma_1, \cdots, \Sigma_t \xleftarrow{\$} \mathcal{D}'} [\Sigma \notin \{\Sigma_1, \cdots, \Sigma_t\}] \\ &= \Pr_{\Sigma_1, \cdots, \Sigma_t \xleftarrow{\$} \mathcal{D}'} \left[\bigwedge_{i=1}^{t} (\Sigma_i \neq \Sigma) \right] \\ &< \left(1 - \frac{2}{n}\right)^t \leq e^{-2t/n} = e^{-2\kappa}. \end{aligned}$$

Therefore, by a union bound over all $\Sigma \in \mathsf{Bad}$, $\Pr_{\Sigma^t \xleftarrow{\$} \mathsf{A}_0^t(crs)}[\Sigma^t \in \mathsf{Bad}] \leq |\mathsf{Bad}| \cdot e^{-2\kappa} \leq 2^\kappa \cdot e^{-2\kappa} \leq e^{-\kappa}$. Now, denoting b the identity of the sender, we bound the success probability of $\mathcal{A}^t = (\mathsf{A}_0^t, \mathsf{A}_1^t)$ in guessing b.

Consider first the case $b = 0$. Then by observation (2), the values (Σ'_0, Σ'_1) form a random sample from $\mathcal{D}_{0,\Sigma^t} \times \mathcal{D}'$ (in particular, they are independent samples from these two distributions). Now, by definition of A_1^t, we have

$$\begin{aligned} \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT}, \mathcal{A}^t, b}(\kappa) = 0 \mid b = 0] &= \Pr[(\Sigma'_0 = \Sigma^t) \vee ((\Sigma'_0 \neq \Sigma^t) \wedge b' = 0)] \\ &= \Pr[\Sigma'_0 = \Sigma^t] + \frac{1}{2} \cdot \Pr[\Sigma'_0 \neq \Sigma^t] \\ &= \frac{1}{2} \cdot \left(1 + \Pr[\Sigma'_0 = \Sigma^t]\right) \geq \frac{1+\varepsilon}{2}, \end{aligned}$$

where the second equality is because the events are disjoint, and the last inequality is by the ε-correctness of the AT. Now, consider the case $b = 1$; in this case, the values (Σ'_0, Σ'_1) computed

by $\mathsf{A}_1^t(\pi, st)$ are distributed exactly as a random sample from $\mathcal{DR} \times \mathcal{D}_{1,\Sigma^t}$. Then, if we condition on Σ^t being outside of the set Bad, we have

$$\begin{aligned}
&\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 1 \mid b = 0, \Sigma^t \notin \mathsf{Bad}] \\
&= \Pr[((\Sigma'_0 \neq \Sigma^t) \wedge (\Sigma'_1 = \Sigma^t)) \vee ((\Sigma'_0 \neq \Sigma^t) \wedge (\Sigma'_1 \neq \Sigma^t) \wedge b' = 1) \mid \Sigma^t \notin \mathsf{Bad}] \\
&= \Pr[(\Sigma'_0 \neq \Sigma^t) \wedge (\Sigma'_1 = \Sigma^t) \mid \Sigma^t \notin \mathsf{Bad}] + \frac{1}{2} \cdot \Pr[(\Sigma'_0 \neq \Sigma^t) \wedge (\Sigma'_1 \neq \Sigma^t) \mid \Sigma^t \notin \mathsf{Bad}] \\
&= \Pr[\Sigma'_0 \neq \Sigma^t \mid \Sigma^t \notin \mathsf{Bad}] \cdot \Pr[\Sigma'_1 = \Sigma^t \mid \Sigma^t \notin \mathsf{Bad}] \\
&\qquad + \frac{1}{2} \cdot \Pr[\Sigma'_0 \neq \Sigma^t \mid \Sigma^t \notin \mathsf{Bad}] \cdot \Pr[\Sigma'_1 \neq \Sigma^t \mid \Sigma^t \notin \mathsf{Bad}] \\
&= \frac{1}{2} \cdot \Pr[\Sigma'_0 \neq \Sigma^t \mid \Sigma^t \notin \mathsf{Bad}] \cdot (1 + \Pr[\Sigma'_1 = \Sigma^t \mid \Sigma^t \notin \mathsf{Bad}]) \\
&\geq \left(1 - \frac{2}{n}\right) \cdot \frac{1+\varepsilon}{2},
\end{aligned}$$

where the third equality follows from the fact that Σ'_0 and Σ'_1 are independent samples, and the last inequality follows from ε-correctness and the definition of the set Bad. Now, since

$$\begin{aligned}
&2 \cdot \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = b] \\
&= \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 0 \mid b = 0] + \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 1 \mid b = 1] \\
&\geq \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 0 \mid b = 0] + \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 1 \mid b = 1, \Sigma^t \notin \mathsf{Bad}] \cdot \Pr[\Sigma^t \notin \mathsf{Bad}] \\
&\geq \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 0 \mid b = 0] + \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 1 \mid b = 1, \Sigma^t \notin \mathsf{Bad}] \cdot (1 - e^{-\kappa}),
\end{aligned}$$

we get

$$\begin{aligned}
\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = b] &\geq \frac{1}{2} \cdot \left(\frac{1+\varepsilon}{2} + \left(1 - \frac{2}{n}\right) \cdot \frac{1+\varepsilon}{2} \cdot (1 - e^{-\kappa})\right) \\
&\geq \frac{1+\varepsilon}{2} \cdot \left(1 - \frac{1}{n}\right) \cdot (1 - e^{-\kappa}) \\
&\geq \frac{1+\varepsilon}{2} - \frac{1}{n},
\end{aligned}$$

where the last inequality uses the fact that $(1+\varepsilon) \cdot (1 - e^{-\kappa})/2n \leq 1/n$. This concludes the proof. □

Induction case. Fix a polynomial number of rounds $c(\kappa)$ and assume that any silent-receiver $(\varepsilon, \delta, c-1, \kappa)$-Anonymous Transfer must satisfy

$$\frac{1-\delta}{2} \geq \frac{1}{c-1} \cdot \left(\frac{\varepsilon}{2} - \frac{1}{n}\right) \text{ for any polynomial } n.$$

equivalently, this means that $(1-\delta)/2 \geq \varepsilon/2(c-1) - \mathsf{negl}(\kappa)$. Let Π^κ_{AT} be any silent-receiver $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer.

Claim. For any polynomial n,

$$\frac{1-\delta}{2} \geq \frac{1}{c} \cdot \left(\frac{\varepsilon}{2} - \frac{1}{n}\right). \tag{6}$$

Proof. We have

$$\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = b] = \frac{1}{2} \cdot \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 0 \mid b = 0] + \frac{1}{2} \cdot \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 1 \mid b = 1].$$

We bound separately the two probabilities on the right hand side:

$$\begin{aligned}\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\kappa}_{AT},\mathcal{A}^t,b}(\kappa)=0 \mid b=0] &= \Pr[(\Sigma'_0=\Sigma^t)\vee((\Sigma'_0\neq\Sigma^t)\wedge b'=0)]\\ &= \Pr[\Sigma'_0=\Sigma^t]+\frac{1}{2}\cdot\Pr[\Sigma'_0\neq\Sigma^t]\\ &= \frac{1}{2}\cdot\left(1+\Pr[\Sigma'_0=\Sigma^t]\right)\geq\frac{1+\varepsilon}{2},\end{aligned}$$

where the second equality is because the events are disjoint, and the last inequality is by the ε-correctness of the AT. Note that nothing really changes when $b=0$ compared to the base case, because the independence of Σ'_0 and Σ'_1 needs only to be invoked in the case $b=1$ (this is because the attacker "favors Σ'_0" in case of tie). Now, consider the case $b=1$. In this case, the values (Σ'_0,Σ'_1) computed by $\mathsf{A}^t_1(\pi,st)$ are distributed as *correlated samples* from $\mathcal{D}'_{1,\Sigma^t}$ and $\mathcal{D}_{1,\Sigma^t}$. We have

$$\begin{aligned}&\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\kappa}_{AT},\mathcal{A}^t,b}(\kappa)=1 \mid b=1]\\ &\quad=\Pr[((\Sigma'_0\neq\Sigma^t)\wedge(\Sigma'_1=\Sigma^t))\vee((\Sigma'_0\neq\Sigma^t)\wedge(\Sigma'_1\neq\Sigma^t)\wedge b'=1)]\\ &\quad=\Pr[(\Sigma'_0\neq\Sigma^t)\wedge(\Sigma'_1=\Sigma^t)]+\frac{1}{2}\cdot\Pr[(\Sigma'_0\neq\Sigma^t)\wedge(\Sigma'_1\neq\Sigma^t)]\\ &\quad=\frac{1}{2}\cdot\left(\Pr[(\Sigma'_0\neq\Sigma^t)\wedge(\Sigma'_1=\Sigma^t)]+\Pr[\Sigma'_0\neq\Sigma^t]\right),\end{aligned}$$

where the second equality follows from the fact that the events $\Sigma'_1=\Sigma^t$ and $\Sigma'_1\neq\Sigma^t$ are disjoint. Now, observe that

$$\begin{aligned}\Pr[\Sigma'_1=\Sigma^t] &= \Pr[\Sigma'_1=\Sigma^t \mid \Sigma'_0=\Sigma^t]\cdot\Pr[\Sigma'_0=\Sigma^t]+\Pr[\Sigma'_1=\Sigma^t\wedge\Sigma'_0\neq\Sigma^t]\\ &\leq \Pr[\Sigma'_0=\Sigma^t]+\Pr[\Sigma'_1=\Sigma^t\wedge\Sigma'_0\neq\Sigma^t].\end{aligned}$$

By construction, $\Pr[\Sigma'_1=\Sigma^t]$ is exactly the probability that $\mathsf{Reconstruct}$ outputs Σ^t given a random transcript for $b=1$ and transmitted value Σ^t. By ε-correctness of the protocol, we therefore have $\Pr[\Sigma'_1=\Sigma^t]\geq\varepsilon$. Plugging this into the previous inequality, we get

$$\Pr[\Sigma'_1=\Sigma^t\wedge\Sigma'_0\neq\Sigma^t]\geq\varepsilon-\Pr[\Sigma'_0=\Sigma^t],$$

which implies that

$$\begin{aligned}\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\kappa}_{AT},\mathcal{A}^t,b}(\kappa)=1 \mid b=1] &\geq \frac{1}{2}\cdot\left(\varepsilon-\Pr[\Sigma'_0=\Sigma^t]+\Pr[\Sigma'_0\neq\Sigma^t]\right)\\ &= \frac{\varepsilon+1}{2}-\Pr[\Sigma'_0=\Sigma^t].\end{aligned}$$

It remains to bound $\Pr[\Sigma'_0=\Sigma^t]$. To do so, we define a *round-reduced* anonymous transfer protocol $(\mathsf{Setup}',\mathsf{Transfer}',\mathsf{Reconstruct}')$, built on top of Π^{κ}_{AT}:

$\mathsf{Setup}'=\mathsf{Setup}$

$\mathsf{Transfer}'(crs,b,\Sigma)$: proceeds as $\mathsf{Transfer}$, except that the parties interrupt the protocol $\mathsf{Transfer}(crs,b,\Sigma)$ after $c-1$ rounds, obtaining a transcript π. In the $(c-1)$'s round, P_0 and P_1 both additionally broadcast $2m$ random bits, denoted (r_0,r_1). Define the final transcript of $\mathsf{Transfer}'$ to be (π,r_0,r_1).

$\mathsf{Reconstruct}'(crs,(\pi,r_0,r_1))$: on input a transcript (π,r_0,r_1), set $r:=r_0\oplus r_1$ and parse r as a concatenation of two m-bit messages (x_0,x_1). Let π' be a c-round transcript where the transcript of the first $c-1$ rounds is π, and the last round transcript is (x_0,x_1). Output $\mathsf{Reconstruct}(crs,\pi')$.

A transcript for the round-reduced protocol is exactly a transcript for Π^{κ}_{AT} where the last two messages have been replaced by uniform random strings[7]. Now, observe that when $b=1$, replacing

[7] We define these strings to be the XOR of two strings sent in the previous rounds by P_0 and P_1 to guarantee that $\mathsf{Reconstruct}$ remains deterministic; this is just a syntactic manipulation which is not crucial, but simplifies the rest of the exposition.

only the message of b_1 by random in the last round still leads to the exact same distribution over transcript. In other terms, $\Pr[\Sigma'_0 = \Sigma^t]$ is exactly the probability that $\mathsf{Reconstruct}'$ outputs Σ^t given a transcript for the round-reduced protocol. Of course, the round-reduced protocol is δ-anonymous[8], and has $c-1$ rounds. Hence, by the induction hypothesis, we can bound the correctness of the round-reduced protocol:

$$\begin{aligned} \frac{1-\delta}{2} &\geq \frac{1}{c-1} \cdot \left(\frac{\Pr[\Sigma'_0 = \Sigma^t]}{2} - \frac{1}{n} \right) \\ \implies \Pr[\Sigma'_0 = \Sigma^t] &\leq (c-1) \cdot (1-\delta) + 2/n. \end{aligned}$$

Therefore, we get

$$\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = 1 \mid b = 1] \geq \frac{\varepsilon+1}{2} - (c-1) \cdot (1-\delta) - \frac{2}{n}.$$

Putting everything together, we obtain

$$\begin{aligned} \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = b] &\geq \frac{1}{2} \cdot \left(\frac{\varepsilon+1}{2} + \frac{\varepsilon+1}{2} - (c-1) \cdot (1-\delta) - \frac{2}{n} \right) \\ &= \frac{\varepsilon+1}{2} - \frac{(c-1) \cdot (1-\delta)}{2} - \frac{1}{n}, \end{aligned}$$

hence

$$\left| \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = b] - \frac{1}{2} \right| \geq \frac{\varepsilon}{2} - \frac{(c-1) \cdot (1-\delta)}{2} - \frac{1}{n}.$$

Now, by definition of δ-anonymity, we also have

$$\begin{aligned} \frac{1-\delta}{2} &\geq \left| \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = b] - \frac{1}{2} \right| \geq \frac{\varepsilon}{2} - \frac{(c-1) \cdot (1-\delta)}{2} - \frac{1}{n} \\ \implies c \cdot \frac{1-\delta}{2} &\geq \frac{\varepsilon}{2} - \frac{1}{n}, \end{aligned}$$

which concludes the proof of the claim. □

4.2 Putting the Pieces Together

With the above analysis, we showed that for any silent-receiver $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer, it must necessarily hold that $(1-\delta)/2 \geq \varepsilon/2c - \mathsf{negl}(\kappa)$. Since any $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer implies a silent-receiver $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer (with the exact same parameters, see Section 3.5), we obtain:

Corollary 2. *Any $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer must satisfy*

$$\frac{1-\delta}{2} \geq \frac{\varepsilon}{2c} - \mathsf{negl}(\kappa).$$

In particular, this implies that there exists no κ-bit AT with overwhelming correctness and anonymity, for any polynomial number of rounds.

Furthermore, as shown in Section 3.5, any *single-bit* c-round AT with correctness $\varepsilon = 1 - \mathsf{negl}(\kappa)$ and anonymity $\delta = 1 - \mathsf{negl}(\kappa)$ implies a κ-bit AT with correctness $\varepsilon' = \varepsilon^\kappa = (1 - \mathsf{negl}(\kappa))^\kappa = 1 - \mathsf{negl}(\kappa)$, and anonymity $\delta' = (\delta - 1) \cdot \kappa - \delta + 2 = 1 - \mathsf{negl}(\kappa)$. Combining this reduction with Corollary 2 concludes the proof of Theorem 3.

[8] Π^κ_{AT} is δ-anonymous; truncating the transcript can trivially not decrease anonymity.

A c-round Anonymous Transfer Protocol $\Pi^1_{AT}{}'$ for $(N-1)$ Parties.

On input (b, σ), P_b follows the role of P_b on input (b, σ) according to Π^1_{AT}.
On input $\perp$, $\mathsf{P}_{b'}$ for $b' \in [N-2] \setminus b$ follows the role of $\mathsf{P}_{b'}$ on input $\perp$ according to Π^1_{AT}.
On input $\perp$, R follows the role of P_{N-1} on input $\perp$ according to Π^1_{AT}.
For each round χ **from** 1 **to** c :
P_b computes the next message according to P_b in Π^1_{AT} with previous messages $\left(\left(x_1^{(1)}, \ldots, x_{N-2}^{(1)}, x_{\mathsf{R}}^{(1)}\right), \ldots, \left(x_1^{(\chi-1)}, \ldots, x_{N-2}^{(\chi-1)}, x_{\mathsf{R}}^{(\chi-1)}\right)\right)$.
$\mathsf{P}_{b'}$ for $b' \in [N-2] \setminus b$ broadcasts $x_{b'}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$.
R broadcasts $x_{\mathsf{R}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$.
R outputs $\Pi^1_{AT}.\mathsf{Reconstruct}\left(\left(x_1^{(1)}, \ldots, x_{N-2}^{(1)}, x_{\mathsf{R}}^{(1)}\right), \ldots, \left(x_1^{(c)}, \ldots, x_{N-2}^{(c)}, x_{\mathsf{R}}^{(c)}\right)\right)$.

Fig. 4: The protocol $\Pi^1_{AT}{}'$ that constructs $(N-1)$-party AT from a given N-party SR-AT for N parties.

4.3 Impossibility of Anonymous Transfer for $N > 3$

The impossibility proof showed thus far only considered the setting for $N = 3$. In this section we will expand the impossibility proof to an arbitrary $N > 3$. For that purpose, we prove the following claim:

Claim. Let Π^1_{AT} be an (ε, δ) Silent Receiver Anonymous Transfer protocol for N parties. Then, protocol $\Pi^1_{AT}{}'$ defined in Fig. 4 that makes black-box access to Π^1_{AT} is an (ε, δ') AT protocol for $(N-1)$ parties, where $\delta' > 1 - (1-\delta) \cdot \frac{N-2}{N-3}$.

Combining this with the insight from Lemma 4 that (ε, δ)-AT imply (ε, δ)-SR-AT, this implies that an (ε, δ)-AT for N parties implies an (ε, δ')-SR-AT for $(N-1)$ parties.

We then proceed to show that if δ is overwhelming, then δ' is overwhelming. The correctness ε does not change by our transformation. Hence, we can iteratively apply this step for *any* number of parties N. Therefore, if there is an N-party $(\mathsf{owhl}(\kappa), \mathsf{owhl}(\kappa))$-AT, then there is also a three-party $(\mathsf{owhl}(\kappa), \mathsf{owhl}(\kappa))$-AT. Since the latter is impossible due to Theorem 3, an N-party AT with overwhelming correctness *and* anonymity cannot exist for any polynomial number of parties N.

To prove our claim we construct an $(N-1)$-party AT $\Pi^1_{AT}{}'$ as shown in Fig. 4. The protocol is based on an (ε, δ)-SR-AT Π^1_{AT}. All parties in $\Pi^1_{AT}{}'$ follow their respective roles in Π^1_{AT}, which means that the sender is constructing *special* messages as advised by Π^1_{AT} whereas the dummy friends only broadcast random bits. To compensate for the missing party (the N-th party that does not exist in the $N-1$ party protocol) the messages of the N-th party are sent by the receiver, who takes on the role of one dummy friend. This is possible because the N-party protocol is a *Silent Receiver* AT where the receiver sends no messages, whereas in the $(N-1)$-party protocol the receiver is allowed to report messages in each round. Then parties continue with the *extended* transcript and proceed to the next round.

Security Analysis of $\Pi^1_{AT}{}'$

Correctness. Correctness of $\Pi^1_{AT}{}'$ follows directly from correctness of Π^1_{AT}.

Lemma 6 (Correctness). *Let Π^1_{AT} be an N-party (ε, δ)-SR-AT. Then $\Pi^1_{AT}{}'$ has correctness ε.*

Proof. This trivially follows from the fact that the extended transcript is distributed identically to an execution of Π^1_{AT}: The messages of the first $(N-1)$ parties are distributed correctly by design, as they follow the original protocol, and the messages of the N-th party are also distributed the same, but accounted to a different party. Thus, the reconstruction algorithm is queried with the same inputs. □

Anonymity. Anonymity requires some work, since one party (the non-silent receiver) in $\Pi^1_{AT}{}'$ is guaranteed to not be the sender.

Lemma 7 (Anonymity). *Let Π^1_{AT} be an N-party (ε, δ)-SR-AT. Then $\Pi^1_{AT}{}'$ has anonymity $\delta' = 1 - (1-\delta) \cdot \frac{N-2}{N-3}$.*

Proof. We want to find a bound δ' for the δ'-anonymity of $\Pi^1_{AT}{}'$, that is, a value δ' for which the following holds for any PPT algorithm A':[9]

$$\Pr[\mathsf{A}' \text{ successful}] \leq \frac{1}{(N-2)} + (1-\delta') \cdot \frac{N-3}{N-2} \tag{7}$$

Now let A' be an adversary on the anonymity of the $(N-1)$-party protocol $\Pi^1_{AT}{}'$. We will now demonstrate how A' can be used by a guessing algorithm A to break the anonymity of Π^1_{AT}:

Upon receiving the challenge transcript $\pi_{\mathcal{C}}$ for the N-party AT, A picks a uniformly random party from the group of potential senders (*i.e.*, among everyone except the receiver) and *erases* that party from the transcript. A reports this party's messages as coming from the receiver in $\Pi^1_{AT}{}'$. Observe that the resulting transcript is perfectly distributed as an honest transcript of $\Pi^1_{AT}{}'$ (given that a non-participant was erased). Now, A feeds this simulated transcript to A', and outputs whatever A' outputs.

If A selects a non-participant—which is the case with probability $(N-2)/(N-1)$ as the party is uniformly sampled among all parties except for the receiver and there is only one actual sender—then the modified transcript is a valid transcript of $\Pi^1_{AT}{}'$ and the probability that A' returns the correct sender is returned is given by the success probability $\Pr[\mathsf{A}' \text{ successful}]$ of A'. Otherwise, if A selected the actual sender, then all possible return values for A' are dummy friends and the probability that A is correct equals 0.

Hence, the overall probability that A is successful is given by the following term:

$$\begin{aligned} \Pr[\mathsf{A} \text{ successful}] &= \frac{N-2}{N-1} \cdot \Pr[\mathsf{A}' \text{ successful}] \\ \iff \Pr[\mathsf{A}' \text{ successful}] &= \frac{N-1}{N-2} \cdot \Pr[\mathsf{A} \text{ successful}] \end{aligned} \tag{8}$$

Due to the δ-anonymity of Π^1_{AT} it holds that

$$\Pr[\mathsf{A} \text{ successful}] \leq \frac{1}{N-1} + (1-\delta) \cdot \frac{N-2}{N-1} \tag{9}$$

By merging Eqs. (8) and (9) we get that

$$\begin{aligned} \Pr[\mathsf{A}' \text{ successful}] &= \frac{N-1}{N-2} \cdot \Pr[\mathsf{A} \text{ successful}] \\ &\leq \frac{N-1}{N-2}\left(\frac{1}{N-1} + (1-\delta) \cdot \frac{N-2}{N-1}\right) \\ \implies \Pr[\mathsf{A}' \text{ successful}] &\leq \frac{1}{N-2} + (1-\delta) \end{aligned} \tag{10}$$

However, to apply δ'-anonymity we need Eq. (10) to be in the form from Eq. (7), which means that we have to set:

$$\Pr[\mathsf{A}' \text{ successful}] \leq \frac{1}{N-2} + (1-\delta) \stackrel{!}{=} \frac{1}{N-2} + (1-\delta')\frac{N-3}{N-2} \tag{11}$$

Solving Eq. (11) for δ' yields the following result:

$$\delta' = 1 - (1-\delta) \cdot \frac{N-2}{N-3} \tag{12}$$

[9] We ignore the absolute value here as by definition, A' is better than guessing and hence $\Pr[\mathsf{A} \text{ successful}] - 1/(N-2) > 0$.

For this value of δ', the guessing algorithm A' on ${\Pi^1_{AT}}'$ can not be used to violate the δ-anonymity of Π^1_{AT}.

In contrast, if there is some guessing algorithm A' which violates the δ'-anonymity for the above value of δ', then it holds that:

$$\Pr[\mathsf{A}' \text{ successful}] > \frac{1}{N-2} + (1-\delta') \cdot \frac{N-3}{N-2} \tag{13}$$

Then the probability that A can successfully determine the sender in the N-party AT is given as follows:

$$\begin{aligned} \Pr[\mathsf{A} \text{ successful}] &= \frac{N-2}{N-1} \cdot \Pr[\mathsf{A}' \text{ successful}] \\ &> \frac{N-2}{N-1} \cdot \left(\frac{1}{N-2} + (1-\delta') \cdot \frac{N-3}{N-2} \right) \\ \Pr[\mathsf{A} \text{ successful}] &> \frac{1}{N-1} + (1-\delta') \cdot \frac{N-3}{N-1} \end{aligned} \tag{14}$$

By inserting the computed value of δ' from Eq. (12) into Eq. (14) and get:

$$\Pr[\mathsf{A} \text{ successful}] > \frac{1}{N-1} + \left(1 - \left(1 - (1-\delta) \cdot \frac{N-2}{N-3}\right)\right) \cdot \frac{N-3}{N-1} \tag{15}$$

So, in total A would be successful with probability

$$\begin{aligned} &\Pr[\mathsf{A} \text{ successful}] > \frac{1}{N-1} + \left(1 - 1 + (1-\delta) \cdot \frac{N-2}{N-3}\right) \cdot \frac{N-3}{N-1} \\ \implies &\Pr[\mathsf{A} \text{ successful}] > \frac{1}{N-1} + (1-\delta) \cdot \frac{N-2}{N-1} \end{aligned} \tag{16}$$

which would contradict the assumed δ-anonymity of Π^1_{AT}. □

From Lemmas 6 and 7 our claim follows:

Corollary 3. *If Π^1_{AT} is an (ε, δ) Silent Receiver Anonymous Transfer protocol for N parties, then there exists a protocol ${\Pi^1_{AT}}'$ with black-box access to Π^1_{AT} that is an (ε, δ') AT protocol for $(N-1)$ parties, where $\delta' > 1 - (1-\delta) \cdot \frac{N-2}{N-3}$.*

This insight yields the following corollary:

Corollary 4. *Let Π^1_{AT} be an (ε, δ) Anonymous Transfer protocol for N parties where $\varepsilon, \delta \in \mathsf{owhl}(\kappa)$. Then there is a protocol three-party (ε, δ')-AT protocol ${\Pi^1_{AT}}'$ where $\varepsilon, \delta \in \mathsf{owhl}(\kappa)$.*

Proof. Let the number N of parties be given. From Π^1_{AT} we can construct a Silent Receiver AT with the same values for ε and δ, so the constructed Silent Receiver AT still has overwhelming correctness and anonymity.

By then applying Corollary 3 it follows that there is a protocol ${\Pi^1_{AT}}'$ for $(N-1)$ parties which still has the same (hence overwhelming) correctness and anonymity $\delta' > 1 - (1-\delta) \cdot \frac{N-2}{N-3}$.

As we assume that $\delta \in \mathsf{owhl}(\kappa)$ it holds that $(1-\delta) \in \mathsf{negl}(\kappa)$, and since $\frac{N-2}{N-3} \in \mathsf{poly}(\kappa)$ it holds that $(1-\delta) \cdot \frac{N-2}{N-3} \in \mathsf{negl}(\kappa)$.

So it total it holds that $\delta' > 1 - \mathsf{negl}(\kappa) \in \mathsf{owhl}(\kappa)$.

By then alternating between an application of Lemma 4 to construct a Silent Receiver AT from the normal AT and Corollary 3 to reduce the number of parties by one we can proceed until $N = 3$, where still it holds that both correctness and anonymity are overwhelming. □

This insight, however, contradicts Theorem 3, which states that no three-party AT with overwhelming correctness and anonymity can exist, which leads to our final corollary:

Corollary 5. *Let κ be the security parameter. Let $N \in \mathsf{poly}(\kappa)$ be the number of individuals present in an execution of Anonymous Transfer. Then for any N-party AT protocol Π^1_{AT} it holds that Π^1_{AT} is not a $(\mathsf{owhl}(\kappa), \mathsf{owhl}(\kappa))$-AT.*

Proof. This proof is by contradiction. Assuming that Π^1_{AT} is an $(\mathsf{owhl}(\kappa), \mathsf{owhl}(\kappa))$-AT means that due to Corollary 4, there is an $(\mathsf{owhl}(\kappa), \mathsf{owhl}(\kappa))$-AT for three parties. However, such a protocol cannot exist due to Theorem 3. □

4.4 Extensions and Limitations

The adversary in our impossibility result makes a black-box use of an arbitrary 3-party silent receiver multibit anonymous transfer; the reduction to N-party single-bit anonymous transfer is black-box as well. In particular, this means that our impossibility result relativizes: it remains true relative to any oracle, where access to the oracle is granted to all participants and all algorithms (including the adversary).

In the next section, we will provide a heuristic construction of *fine-grained* anonymous transfer. The aim of this construction is to complement our impossibility result, and to draw an interesting and surprising picture: anonymous transfer appears to be impossible to realize with the standard superpolynomial cryptographic hardness gaps, but becomes feasible if one settles for a small polynomial hardness gap. Our fine-grained construction is described and formally proven secure using an ideal obfuscation scheme; instantiating the scheme with candidate indistinguishability obfuscation schemes gives a plausible heuristic construction (the same way that instantiating the random oracle model with standard hash functions gives plausible heuristic constructions of various cryptographic primitives, when the construction is not pathological). Because our impossibility result relativizes, in contrast, standard anonymous transfer remains provably impossible relative to an ideal obfuscation oracle (while fine-grained anonymous transfer, as we will see, provably exist relative to such an oracle).

Impossibility of fine-grained multibit AT with overwhelming correctness and anonymity. In the multibit setting, where the sender wants to transmit $\omega(\log \kappa)$ bits to the receiver, our result further demonstrates that there exists no *fine-grained* anonymous transfer with overwhelming correctness and anonymity $1 - \mathsf{negl}(\kappa)$, even with an *arbitrary small polynomial gap* between the runtime of the honest parties and that of the adversary. Indeed, let $r = \mathcal{O}(c \cdot m)$ be a lower bound on the runtime of the honest parties (r is the total number of bits sent by the sender, hence it is a clear lower bound on its running time), and consider an adversary $\mathcal{A}^t$ with $t = \kappa \cdot c^g$, where $g > 0$ is an arbitrarily small constant. Then by construction, the runtime of $\mathcal{A}^t$ is $\mathcal{O}(\kappa \cdot r \cdot c^g) \leq \mathcal{O}(\kappa \cdot r^{1+g})$ (as it is dominated by the cost of sampling t random transcripts for A_0^t). Then this adversary satisfies

$$\frac{1-\delta}{2} \geq \left| \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\kappa_{AT},\mathcal{A}^t,b}(\kappa) = b] - \frac{1}{2} \right| \geq \frac{1}{c} \cdot \left(\frac{\varepsilon}{2} - \frac{1}{c^g} \right), \tag{17}$$

which implies that δ and ε cannot be simultaneously equal to $1 - \mathsf{negl}(\kappa)$ (since $1/(2c) - 1/c^{1+g}$ cannot be a negligible function for any polynomial c and any constant $g > 0$).

Limitations of the impossibility result. Even putting aside the heuristic security guarantee of our fine-grained construction (or its security in an idealized model), a gap remains between our impossibility result and our construction: our impossibility result does not rule out the possibility of having, say, a $(1 - \mathsf{negl}(\kappa), 1 - 1/c, c, \kappa)$-Anonymous Transfer – that is, an anonymous transfer with overwhelming correctness, and vanishing anonymity error $1/c$ in c rounds, with standard (superpolynomial) security. In contrast, our heuristic construction only achieves overwhelming correctness and anonymity arbitrarily close to $1/c$ against *fine-grained* adversaries. It is an interesting open question to close this gap. We conjecture that the true answer is negative:

Conjecture 1. There exists no $(1 - \mathsf{negl}(\kappa), 1 - 1/c, c, \kappa)$-Anonymous Transfer.

What follows assumes that the reader is familiar with standard philosophical considerations on the worlds of Impagliazzo. Proving the above conjecture would have a very interesting (theoretical) consequence: it would demonstrate the existence of a natural cryptographic primitive that plausibly exists within the realm of fine-grained cryptography, yet is impossible with standard hardness gap. It is known that fine-grained constructions sometimes allow building "high-end" cryptographic primitives in "low-end" cryptographic realms. For example, Merkle puzzles, which can be instantiated under exponentially strong one-way functions [BGI08], provide a fine-grained key exchange; borrowing Impagliazzo's terminology [Imp95], this places "fine-grained Cryptomania" inside (a strong form of) Minicrypt. Proving the conjecture would induce a comparable result, but at the highest level of the hierarchy: it would, in a sense, place fine-grained Impossibilitopia (a world of cryptographic primitives so powerful that they simply cannot exist) inside Obfustopia.

Fine-grained Protocol Π^1_{AT}.

Upon activation, R draws $OTP \xleftarrow{\$} \{0,1\}$ and computes $(\mathsf{k_R}, \mathsf{vk_R}) \leftarrow \text{SIG.KeyGen}(1^\kappa)$. Then R sets $x_{\mathsf{R}}^{(0)} \leftarrow \text{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk_R}))$ and broadcasts $x_{\mathsf{R}}^{(0)}$.

On input (b, σ), P_b computes a signature key pair $(\mathsf{vk}_b, \mathsf{k}_b) \leftarrow \text{SIG.KeyGen}(1^\kappa)$ and a symmetric key $\mathsf{sk}_b \leftarrow \text{SKE.KeyGen}(1^\kappa)$.

Then, P_b computes a signature $\mu \leftarrow \text{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$ and broadcasts $x_b^{(0)} \leftarrow \text{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \text{SKE.Enc}(\mathsf{sk}_b, (\sigma, \mu))$.

Upon activation , P_{1-b} sets uniformly random $x_{1-b}^{(0)}$.

For each round χ **from** 1 **to** c :

P_b computes $\mu \leftarrow \text{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$ and sets $x_b^{(\chi)} \leftarrow \text{SKE.Enc}(\mathsf{sk}_b, (\sigma, \mu))$.

P_{1-b}**:** Broadcast $x_{1-b}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$.

R**:** computes $\mu \leftarrow \text{SIG.Sig}(\mathsf{k_R}, (x_{\mathsf{R}}^{(0)}, (x_0^{(0)}, x_1^{(0)}), \ldots, (x_0^{(c)}, x_1^{(c)})))$, compute $\sigma' := P_{AT}(x_{\mathsf{R}}^{(0)}, (x_0^{(0)}, x_1^{(0)}), \ldots, (x_0^{(c)}, x_1^{(c)}), \mu)$ and output $OTP \oplus \sigma'$.

Fig. 5: The protocol Π^1_{AT} for fine-grained Anonymous Transfer. The circuit P_{AT} is defined in Fig. 6.

$P_{AT}[\mathsf{pk}_P, c]\left(x_{\mathsf{R}}^{(0)}, \left(x_0^{(0)}, x_1^{(0)}\right), \left(x_0^{(1)}, x_1^{(1)}\right) \ldots, \left(x_0^{(c)}, x_1^{(c)}\right)\right)$

$(OTP, \mathsf{vk_R}) := \text{PKE.Dec}^*(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$,

$(\mathsf{sk}_0, \mathsf{vk}_0) := \text{PKE.Dec}^*(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$, $\quad (\sigma_0, \mu_0) := \text{SKE.Dec}^*(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$,

$(\mathsf{sk}_1, \mathsf{vk}_1) := \text{PKE.Dec}^*(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$, $\quad (\sigma_1, \mu_1) := \text{SKE.Dec}^*(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$,

if $\neg\text{SIG.Vfy}(\mathsf{vk_R}, (x_{\mathsf{R}}^{(0)}, (x_0^{(0)}, x_1^{(0)}), \ldots, (x_0^{(c)}, x_1^{(c)})))$ **then** :

  return $\text{CointossS}^{(\pi)}_{(0.5)}(0,1)$

$\chi_0 := [\![\text{SIG.Vfy}(\mu_0, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$, $\quad \chi_1 := [\![\text{SIG.Vfy}(\mu_1, \mathsf{vk}_1, x_{\mathsf{R}}^{(1)})]\!] \cdot (c+1)$,

foreach $\chi \in \{1, \ldots, c\}$ **do**:

  foreach $b \in \{b' | b' \in \{0,1\}, \chi_b = (c+1)\}$ **do**: // Take on the role of each potential sender.

    $X_b := \text{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$, $\quad \sigma'_b := X_b[0]$, $\quad \mu_b := X_b[1 : |X_b|]$

    if $\neg\text{SIG.Vfy}(\mu_b, \mathsf{vk}_b, \pi[\chi - 1]) \vee \sigma_b \neq \sigma'_b$ **then** :

      $\chi_b := \chi$ // Remember first bad round.

$b' := \mathbf{argmax}_b(\chi_b)$

return $OTP \oplus \text{CointossS}^{(\pi)}_{(1/2\cdot(1+\chi_{b'}/c))}(\sigma_{b'}, (1-\sigma_{b'}))$

Fig. 6: Obfuscated program P_{AT} for the fine-grained setting with c rounds.

5 Fine-Grained AT from Ideal Obfuscation

In this section, we focus on realizing Anonymous Transfer with fine-grained security according to Definition 6. More precisely, we construct a c-round protocol which achieves anonymity δ, where the honest parties have runtime in $\mathfrak{C}_1 := \mathcal{O}(c)$ against adversaries in $\mathfrak{C}_2 := \mathrm{o}(c^2(1-\delta))$, where $c = c(\kappa)$ is a polynomial in κ. For the sake of simplicity we introduce the protocol with $N = 3$, implying a single dummy friend. However, expanding this protocol to an arbitrary $N \in \mathbb{N}$ is straightforward as the behavior of all dummy friends is the same by definition and instead of two messages, each round now contains $N - 1$ messages.

We exploit the limited runtime of the adversary and provide a protocol in Fig. 5 with c rounds. In each new round (or with each valid sender message) the probability that the correct bit is eventually returned increases, *i.e.*, each valid round increases the receiver's confidence in the message. Each round lets the sender compute a signature μ using a sEUF-CMA secure signature scheme[10] for the transcript of the previous round. The transferred bit σ and the signature μ are then sent. The verification key for the signature scheme is transmitted by the sender in the first round. In order to

[10] See Appendix A.3 for a definition of sEUF-CMA.

make the sent messages look random the message is not sent directly. Instead, the sender encrypts the message using an IND$-CCA secure encryption scheme[11], [Rog04]. Since not every length m bitstring is a valid ciphertext, we use a special function Dec^* instead of the normal function Dec, which is defined as follows: If Dec on input ct returns $\perp$ then Dec^* returns $\mathsf{F}(ct)$, otherwise Dec^* returns $\mathsf{Dec}(ct)$. Hence, every possible input allows an interpretation as a cleartext. We use those for both the asymmetric and symmetric schemes.

In order to make the output unusable for any other party, the receiver draws a One-Time-Pad as first message which eventually masks the final output, and a verification key of a signature scheme. The latter is used to ensure that the receiver *approves* with the transcript; after the two potential senders provided all messages, the receiver *signs* the entire transcript and only if this signature verifies the entire previous transcript, the circuit continues. The first message of the receiver is broadcast, while the signature is only used locally.

The receiver obtains its output by computing the signature as described above and feeding the final transcript alongside the signature into an obfuscated circuit which is supplied in a common reference string. The circuit is obfuscated using ideal obfuscation[12]. It hides a PRF key and a secret decryption key sk_P for the IND$-CCA secure PKE. The corresponding encryption key pk_P is also part of the CRS and, hence, known to all parties. This encryption scheme is used by the sender and the receiver to hide their respective *first* message. This uniquely determines the symmetric key used to decrypt the remaining messages of each potential sender. The message also contains a verification key used to sign the previous messages in future rounds, the bit that the sender wants to transfer, and the initial signature on the receivers message. The remaining rounds of the sender are encrypted using a *symmetric* scheme, namely the IND$-CCA secure SKE scheme, using the key transferred to the circuit in the first round.

The circuit is shown in Fig. 6. It starts by extracting the verification keys and symmetric encryption keys (one per potential sender) alongside the bits that the respective party wants to transfer and the initial signatures on the first receiver messages from the respective initial messages of both parties, and the receivers OTP and verification key from the receiver message. Then the circuit starts by verifying the signature of the receiver on the entire transcript, and if that does not match, returns a uniformly random bit[13]. Otherwise, if the receiver's signature is valid, the circuit searches for the first *faulty* round of each potential sender. That is, the first round of each potential sender where the signature on the previous round fails to verify or where the encoded bit differs from the bit extracted from the initial message. The party who sent the most consecutive valid rounds is selected as the sending party. The circuit outputs the bit transmitted by that party with probability depending on the ratio between valid sender messages and the total number of rounds, which ranges between 1/2 (*i.e.*, a uniformly random bit) if no round was valid for any party and 1 (*i.e.*, deterministically returning the correct bit) if all rounds were correct. However, as stated before, the circuit does not output that bit directly, but instead *masks* it using the OTP extracted from the receiver's first message. This ensures secrecy, as other parties only get a masked output which information-theoretically hides the actual bit.

5.1 Security Analysis

Theorem 4 (Correctness). *If the protocol from Fig. 5 is instantiated with an Ideally Obfuscated version of the circuit from Fig. 6 the protocol is ε-correct with $\varepsilon = (1 - \mathsf{negl}(\kappa))$.*

At the end of an honest protocol execution, the maximum round in which a valid signature has been provided equals the number of rounds c. With overwhelming probability, the sending parties' input is the only one that contains c many valid rounds. Hence, the correctly masked bit is returned. Since the mask is input by the receiver and later applied to the output, the receiver obtains the correctly masked bit. We refer the reader to Appendix D.1 for a formal proof.

[11] See Appendix A.2 for a definition of IND$-CCA.

[12] See Appendix A.4 for a definition of ideal obfuscation.

[13] This is denoted in the figure by the $\mathsf{CointossS}^{(\pi)}_{(p)}(\sigma, \overline{\sigma})$ function, which returns σ, *i.e.* the first argument, with probability p, and $\overline{\sigma}$, *i.e.* the second argument, with the complementary probability $(1 - p)$, where the randomness for p is extracted from the argument provided by π.

Oracle O_i^β

if $\beta = 0$ **then**

$p_i \coloneqq \dfrac{i + c - 1}{2c}$

else

$p_i \coloneqq \dfrac{i + c}{2c}$

return $\mathsf{Ber}(p_i)$

$\mathcal{C}(c)$

$\beta \xleftarrow{\$} \{0, 1\}$

return $\mathcal{A}^{\mathsf{O}_1^\beta, \ldots, \mathsf{O}_c^\beta}(1^\kappa)$

$\mathcal{A}(1^\kappa)$

for $j = 1 \ldots t$ **do**

$i_j \leftarrow \mathit{Computations}$

$x_j \xleftarrow{\$} \mathsf{O}_{i_j}$

$\beta' \leftarrow \mathit{Computations}((i_j, x_j)_{j=1}^t)$

return β'

Fig. 7: Game to distinguish whether Bernoulli oracles follow a given distribution p or $q = p - 1/2c$.

Theorem 5 (Anonymity). *Let* PKE *be an IND$-CCA secure asymmetric encryption scheme, let* SKE *be a tightly secure multi-challenge IND$-CCA secure symmetric encryption scheme, let* SIG *be an sEUF-CMA secure signature scheme, let* $\mathbb{O}$ *be an ideal obfuscator, let* F *be a secure PRF, and let* κ *be the security parameter. Then the* c*-round protocol* Π^1_{AT} *for* $N = 3$ *satisfies* δ*-anonymity for all adversaries in* $\mathfrak{C}_2 \coloneqq \mathrm{o}(c^2(1 - \delta))$.

Proof (sketch). An outline of the entire proof is given in Appendix D.2. On a high level, the proof is structured into two parts. In the first part, we successively modify the anonymity game $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^1_{AT},\mathcal{A},b}(\kappa)$ and the obfuscated circuit oracle P_{AT} to remove as much computationally hidden information about b as possible. More precisely, we exploit the non-malleability of PKE and sEUF-CMA security of Sig to unnoticeably alter the oracle to determine the number of valid rounds by counting how many rounds of the input transcript are identical to the challenge transcript provided by $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^1_{AT},\mathcal{A},b}(\kappa)$. The first round which is not entirely identical to the challenge transcript (i.e. either the sender message or the non-sender message differ) increases the valid rounds count only if the input sender message is identical to the challenge sender message or if the input sender message decrypts to the same content as the challenge sender message. The following round will be counted as invalid since the signature verification will fail. After this step, the decryption keys of SKE and PKE are not necessary for chosen-ciphertext simulation anymore. Then, we first replace the sender messages which are encrypted using SKE and then the first round sender message which is encrypted using PKE with uniform randomness exploiting the IND$-CCA security of both encryption schemes.

The only information about the bit b that is left in the present game is due to the oracle which counts valid sender messages by comparing the input sender message with the challenge sender message. Clearly, the final modification of the game must be the removal of this dependency on b. However, this removal will noticeably alter the output distribution of the oracle. Hence, an adversary with arbitrary polynomial runtime will be able to distinguish this hop with constant probability [CDV+14]. However, if we can limit the runtime of the adversary to be sub-quadratic in the runtime of the honest protocol execution, we are able to apply results from distribution testing to achieve a good bound for this distinguishing advantage. We will elaborate on this final game hop in more detail below and will refer to the second last game as $\mathrm{Game}^\sigma_7(\kappa)$ and to the last game (i.e. the game, where no information about b remains) as $\mathrm{Game}^\sigma_8(\kappa)$. For detailed game descriptions and the remaining game hops, we refer the reader to Appendices D.2 and D.3, respectively.

For the sake of reducing complexity of the problem of proving indistinguishability between $\mathrm{Game}^\sigma_7(\kappa)$ and $\mathrm{Game}^\sigma_8(\kappa)$ we describe an intermediate game in Fig. 7 that is provably as hard to solve as distinguishing the two games.

The key idea is the following: The challenger $\mathcal{C}$ creates c oracles where the probability to return 1 is equally distributed between $1/2$ and 1 in c steps.

On $\beta = 0$ the oracles are distributed equally between $[1/2, 1)$. On $\beta = 1$ the oracles are distributed equally between $(1/2, 1]$. That is, on $\beta = 0$ the oracle χ returns 1 with probability $(c + \chi - 1)/(2c)$ and on $\beta = 1$ it returns 1 with probability $(c + \chi)/(2c)$.

We now stress that this game is *as hard* as the problem of distinguishing the two games from $\mathrm{Game}^\sigma_7(\kappa)$ and $\mathrm{Game}^\sigma_8(\kappa)$:

Lemma 8. *Let* $\mathcal{D}$ *be a distinguisher distinguishing* $\mathrm{Game}^\sigma_7(\kappa)$ *and* $\mathrm{Game}^\sigma_8(\kappa)$ *with advantage* α *over guessing. Let* t *be the number of queries that* $\mathcal{D}$ *sends to the obfuscated circuit. There is a reduction adversary* $\mathcal{A}$ *that uses* $\mathcal{D}$ *which has advantage* α *over guessing in winning Fig. 7.*

Proof. Creating the Transcript. Upon activation the adversary samples a bit $\sigma \in \{0,1\}$ that is to be transferred in the transcript and a bit $b \in \{0,1\}$ that defines the sending party. Using c as the number of rounds and m as the length of each message $\mathcal{A}$ creates the transcript by reporting two uniformly sampled messages of length m for each round inside the loop, and for the preparational round two messages of length $2m$ each for the dummy friend and the sender and one message of length m for the receivers messages. The resulting transcript $\pi_{\mathcal{C}}$ is reported to the distinguisher $\mathcal{D}$.

Simulating the Circuit. When receiving a transcript π, $\mathcal{A}$ checks if $\pi[0] \neq \pi_{\mathcal{C}}[0]$. If this is the case $\mathcal{A}$ executes the program honestly as defined in $\mathrm{Game}_7^{\sigma}(\kappa)$ and returns the output bit. Note that the normal execution of a *different* transcript is entirely independent of the secret β chosen by $\mathcal{C}$ and effectively of the whole change induced in the transition from $\mathrm{Game}_7^{\sigma}(\kappa)$ to $\mathrm{Game}_8^{\sigma}(\kappa)$. Hence $\mathcal{A}$ has all the information to simulate *new* transcripts.

Otherwise, if $\pi[0] = \pi_{\mathcal{C}}[0]$, $\mathcal{A}$ embeds the challenge oracle by finding $\chi^* = \max_{\chi} \pi[0 \dots \chi] = \pi_{\mathcal{C}}[0 \dots \chi]$ by comparing the input to the previously reported challenge transcript. Then $\mathcal{A}$ checks the messages $\pi[\chi^* + 1]$. Let ${x^{\mathcal{C}}}_0$ and ${x^{\mathcal{C}}}_1$ be the message from the challenge transcript at round $\chi^* + 1$ of parties P_0 and P_1, respectively, and let x_0 and x_1 be the respective messages from the transcript that was input by $\mathcal{D}$.

$\mathcal{A}$ only considers the message reported for P_b; if $x_b = {x^{\mathcal{C}}}_b$, that is, if at round $\chi^* + 1$, the input transcript contains the same message of the *sending* party, $\mathcal{A}$ sends χ^* to the oracle O_{χ^*} provided by $\mathcal{C}$ and obtains a binary output σ^*. Since the output of $\mathcal{C}$ is going towards 1, whereas the bit $\sigma_{\mathcal{C}}$ was uniformly distributed, $\mathcal{A}$ has to *adjust* the output if $\sigma_{\mathcal{C}} = 0$, such that 0 is output with higher probability. Hence, $\mathcal{A}$ sends $\sigma^* \oplus \bar{\sigma_{\mathcal{C}}}$ back as output to $\mathcal{D}$; this essentially flips the output bit iff $\sigma_{\mathcal{C}} = 0$, in which case the output of the oracle lies between 0 and 1/2.

Otherwise, if $x_b \neq {x^{\mathcal{C}}}_b$, $\mathcal{A}$ locally samples a Bernoulli-distributed bit according to p_{χ^*} and sends the result back to $\mathcal{D}$.

Translating the Result. After (at most) t queries the distinguisher terminates and sends its guess. If $\mathcal{D}$ guesses it was playing $\mathrm{Game}_8^{\sigma}(\kappa)$ then $\mathcal{A}$ reports $\beta' = 0$ to $\mathcal{C}$, thus guessing that the χ-th oracle returns 1 with probability $(\chi + c - 1)/2c$. If $\mathcal{D}$ guesses it was playing $\mathrm{Game}_7^{\sigma}(\kappa)$ then $\mathcal{A}$ sends the output $\beta' = 1$ to $\mathcal{C}$, indicating that the probabilities were given as $(\chi + c)/2$.

For a tight reduction it remains to show that the view of $\mathcal{D}$ is statistically close to that from $\mathrm{Game}_7^{\sigma}(\kappa)$ and $\mathrm{Game}_8^{\sigma}(\kappa)$, depending on the choice of $\mathcal{C}$:

First, note that the way that $\mathcal{C}$ handles the queries in Fig. 7 mimics exactly the situation from the game change: either a query for χ returns the 1 with probability $\frac{\chi+c}{2c}$ or with probability $\frac{\chi-1+c}{2c}$. The former implies that the sending parties message is still considered as in $\mathrm{Game}_7^{\sigma}(\kappa)$, the latter representing $\mathrm{Game}_8^{\sigma}(\kappa)$ where the oracle ignores a valid message in round $\chi + 1$ and only considers the round χ^* until *both* messages match the challenge transcript.

The challenge oracle is only queried when the *right* sending message for the following round is used. This is also the only instance where $\mathrm{Game}_7^{\sigma}(\kappa)$ and $\mathrm{Game}_8^{\sigma}(\kappa)$ differ from each other.

Finally, note that at most t sample were sent to the challenge oracle. Thus, the view is simulated perfectly and the view of $\mathcal{D}$ corresponds to $\mathrm{Game}_8^{\sigma}(\kappa)$ iff the additional factor of $-1/2c$ is ignored and to $\mathrm{Game}_7^{\sigma}(\kappa)$ otherwise.

So if $\mathcal{D}$ has non-negligible advantage α over guessing then $\mathcal{A}$ inherits this advantage for winning the game from Fig. 7, albeit if $\mathcal{D}$ sends t queries to $\mathcal{A}$, less than t queries are forwarded to $\mathcal{C}$. □

Proving indistinguishability has thus been reduced to showing that no fine-grained adversary can win the game from Fig. 7 with non-negligible advantage. The interface of an adversary in this game is given as a set of $2c$ *oracles*. Each oracle follows a *Bernoulli* distribution that returns the correct bit $\sigma_{\mathcal{C}}$ with probability p. For each round $\chi < c$ any distinguisher $\mathcal{D}$ is given access to two oracles. Each oracle can be queried by copying the first χ messages of *both* parties, but then using (exactly) one new message for round $(\chi + 1)$—which replaces either the sending parties message or that of the dummy friend. Any upper bound on winning the game from Fig. 7 translates to the underlying problem of distinguishing the final two games.

Analyzing the game from Fig. 7 comes down to probability theory. Recall from Corollary 1 that in order to distinguish two Bernoulli distributions p and q with advantage $\alpha/2$ we require $\Omega(\alpha/\mathsf{d}_{\mathsf{TV}}(p,q))$ many samples. Applying this corollary to Fig. 7 implies that we have c instances where the χ-th instance is to distinguish $p = \frac{\chi+c}{2c}$ from $q = \frac{\chi+c-1}{2c}$. This implies the following

L_1-norm between p and q in round χ:

$$\begin{aligned}\mathsf{d}_{\mathsf{TV}}(p,q) &= \frac{1}{2}(|\Pr[p=1]-\Pr[q=1]|+|\Pr[p=0]-\Pr[q=0]|)\\ &= \frac{1}{2}\left(\left|\frac{c+\chi}{2c}-\frac{c+\chi-1}{2c}\right|+\left|\frac{c-\chi}{2c}-\frac{c-\chi+1}{2c}\right|\right)=\frac{1}{2}\left(\frac{1}{2c}+\frac{1}{2c}\right)=\frac{1}{2c}\end{aligned} \tag{18}$$

Note here that the total variational distance in round χ is *independent* from the round χ and the same for all c oracles. Combining this information with Lemma 2 means that *any* distribution p and q resulting from sampling t times from *arbitrary* oracles results in a total variational distance $\leq t\frac{1}{2c}$. [14]

We now merge this insight with the result of Eq. (18) and the bound of Corollary 1. This leads a lower bound of:

$$t \in \Omega\left(\frac{\alpha}{\mathsf{d}_{\mathsf{TV}}(p,q)}\right)=\Omega(\alpha c) \tag{19}$$

We thus have:

Corollary 6. *Let $\mathcal{D}$ be a distinguisher in Fig. 7 that uses t samples and has runtime in $\mathfrak{C}_2 := o(c^2/\alpha)$. Let the cost of acquiring a single sample be $\mathcal{O}(c)$. Then the distinguisher $\mathcal{D}$ is correct with probability at most $1/2+\alpha/2$.*

Proof. The bound from Eq. (19) covers any adversary trying to win Fig. 7 regardless of how the t samples are distributed between the c oracles. This follows from the subadditional property of the total variational distance shown in Lemma 2 and the computation in Eq. (18) showing that the total variational distance is the same between all oracles; thus the bound from Lemma 1 still is valid and the total variational distance between any pair of t-fold distributions is at most $t \cdot \frac{1}{2c}$.

Thus Lemma 3 maintains its validity. Hence the lower bound of Eq. (19) matches our setting. The bound is linear in c with the linear cost of querying a single sample (as the adversary has to evaluate the entire circuit for each sample, which requires $\mathcal{O}(c)$ runtime) this limits the distinguisher in such a way that only strictly less samples can be drawn than required according to Eq. (19). □

Putting everything together, we have that for all PPT distinguishers D, $|\Pr[\mathsf{out}_{0,\mathsf{D}}=1]-\Pr[\mathsf{out}_{8,\mathsf{D}}=1]|$ is negligible in κ. In particular, $|\Pr[\mathsf{out}_{0,\mathsf{D}}=1]-\Pr[\mathsf{out}_{8,\mathsf{D}}=1]|$ is negligible for distinguishers D in $\mathfrak{C}_2$. Additionally, the employed reductions are in $\mathfrak{C}_1=\mathcal{O}(c)$. Furthermore, for all adversaries $\mathcal{A}$, $|\Pr[\mathsf{out}_{8,\mathcal{A}}=1|b=0]-\Pr[\mathsf{out}_{8,\mathcal{A}}=1|b=1]|\leq\alpha$, where the runtime of the game also is in $\mathfrak{C}_1$. Hence, we may conclude that for all adversaries $\mathcal{A}$ in $\mathfrak{C}_2$, $|\Pr_{b\xleftarrow{\$}\{0,1\}}[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^1_{AT},\mathcal{A},b}(\kappa)=b]-1/2|\leq\alpha/2$. □

On the Need for Stronger Obfuscation. Due to [CLT+15], indistinguishability obfuscation (or more precisely, its probabilistic variant) can only guarantee indistinguishability if the distance between the output distributions of two circuits is statistically close to zero. This is not the case in our final game hop. Therefore, we crucially require a stronger form of obfuscation such as virtual black-box obfuscation or ideal obfuscation. Due to [JLL+22], employing ideal obfuscation yields a heuristic candidate proven secure in an idealized model. Hence, our result constitutes a first step towards instantiating anonymous transfer.

Stronger Anonymity Notions. Our positive result demonstrates that despite our strong negative result, *some* non-trivial anonymity is achievable. Note, however, that our positive result is still weak in many regards. Strengthening the achieved notion to, for instance, achieve anonymity against malicious non-participants, seems highly non-trivial. In particular, malicious non-participants may easily nullify any correctness guarantee by behaving exactly like a sender. Straightforward attempts to address this problem, e.g. letting the obfuscated circuit output all messages with equal confidence, open the gates for new attacks. For instance, in the above setup, replacing the last message of half of all possible senders causes the circuit to output either both the sender message and the injected message or only the injected message, depending on whether the real sender is part of the parties whose messages are replaced. This strategy allows to de-anonymize the sender in runtime $\mathcal{O}(c \log c)$.

[14] This is in contrast to the Hellinger-distance H which yields tighter bounds but where the amount of information from a single query really depends on the oracle O_χ which is queried. This makes it harder to provide meaningful bounds for adversaries querying different oracles with their t samples.

5.2 Final Result

Let $c = c(\kappa)$ be a polynomial in κ. Let $\mathfrak{C}_1 := \mathcal{O}(c)$ and let $\mathfrak{C}_2 := \mathrm{o}(c^2(1-\delta))$ for some $\delta \in \mathbb{R}_{[0,1]}$. Putting Theorems 4, 5 and 6 together, we have:

Corollary 7. *The protocol Π^1_{AT} is a strong $\mathfrak{C}_1$-fine-grained $(1-\mathsf{negl}(\kappa), \delta, 1-\mathsf{negl}(\kappa), c, 1)$-AT against $\mathfrak{C}_2$.*

Applying Lemma 5 to transform our single-bit AT into an ℓ-bit AT yields:

Corollary 8. *The protocol Π^ℓ_{AT} is a strong $\mathfrak{C}'_1$-fine-grained $(1-\mathsf{negl}(\kappa), (\delta\ell - \ell - \delta + 2), 1-\mathsf{negl}(\kappa), c\cdot\ell, \ell)$-AT against $\mathfrak{C}'_2$, where $\mathfrak{C}'_1 = \ell \cdot \mathfrak{C}_1$ and $\mathfrak{C}'_2 = \mathfrak{C}_2 - \ell \cdot \mathfrak{C}_1$.*

Using $\delta = 1 - \frac{1}{\sqrt{c}}$ and $c = \Omega(\ell^2)$ for the single-bit AT Π^1_{AT} we get that $\delta' := 1 - \frac{\ell-1}{c}$ and $\mathfrak{C}'_1 = \mathcal{O}(\ell \cdot c)$ and $\mathfrak{C}'_2 = \mathrm{o}(c^2(1-\delta) - \ell \cdot c) = \mathrm{o}(c^2(1-\delta)) = \mathrm{o}(c^{1.5})$.

A non-black-box change to the protocol Π^1_{AT} from Figs. 5 and 6 leads to better overall parameters. We introduce the necessary changes to the protocol alongside a security analysis in Appendix F.

Corollary 9. *The protocol $\Pi^{\ell\,\prime}_{AT}$ defined in Appendix F is a strong $\mathfrak{C}_1$-fine-grained $(1-\mathsf{negl}(\kappa), \delta, 1-\mathsf{negl}(\kappa))$-AT against $\mathfrak{C}_2$, where $\mathfrak{C}_1 := \mathcal{O}(c)$ and $\mathfrak{C}_2 := \mathrm{o}(c^2(1-\delta))$.*

6 Undetectable Oblivious Transfer

For a candidate application of Anonymous Transfer we propose Undetectable Oblivious Transfer (UOT): An N party protocol with $k = 2$ participants, one sender and one receiver, performing classical Oblivious Transfer without (1) the sender knowing which of the $N-1$ other individuals is the receiver, (2) the receiver knowing which of the $N-1$ other individuals is the sender, and (3) the dummy friends knowing that a computation is in progress at all.

To that end we require an additional property for Anonymous Transfer protocols which we define in Section 6.1 which hides the output of an AT from the dummy friends.

We then define in Section 6.2 the security properties of an Undetectable Oblivious Transfer scheme. We additionally provide a candidate instantiation for $N = 3$ that can be canonically extended to any number $N > 3$ in Appendix G.

6.1 Strong Anonymous Transfer

For several applications we require the additional property that the real message can only be extracted by the real receiver. This gives rise to a stronger definition of AT:

Definition 7 (Strong $(\varepsilon, \delta, \varsigma, c, \ell)$-Anonymous Transfer). *A strong $(\varepsilon, \delta, \varsigma, c, \ell)$-Anonymous Transfer is defined analogously to a $(\varepsilon, \delta, c, \ell)$-Anonymous Transfer from Definition 3 but additionally satisfies ς-secrecy from Definition 8.*

The additional property we require formally states that without the receivers random tape, the message cannot be successfully reconstructed from the transcript. We focus on the 3-party case, since case captures the primitive we will need to construct undetectable oblivious transfer; the definition can however easily be generalized to any number of parties.

Definition 8 (ς-Secrecy). *For any sufficiently large security parameter κ, any message length $\ell \in \mathsf{poly}(\kappa)$, any message $\Sigma \in \{0,1\}^\ell$, any number of individuals $N \in \mathsf{poly}(\kappa)$, any PPT reconstruction algorithm A, and any CRS $crs \leftarrow \mathsf{Setup}(1^\kappa)$, an Anonymous Transfer protocol Π^ℓ_{AT} between players $(\mathsf{P}_0, \mathsf{P}_1, \mathsf{R})$ is ς-secret if it holds that:*

$$\left| \Pr\left[\begin{matrix} \pi \leftarrow \mathsf{Transfer}_{\langle \mathsf{R}, \mathsf{P}_0, \mathsf{P}_1 \rangle}(crs, b, \Sigma) \\ \Sigma' \leftarrow \mathsf{A}(crs, \pi) \end{matrix} : \Sigma = \Sigma' \right] - \frac{1}{2^\ell} \right| \leq (1-\varsigma) \cdot \frac{2^\ell - 1}{2^\ell} \tag{20}$$

The secrecy is a value between 0 and 1, where $\varsigma = 0$ implies that A can reconstruct Σ with absolute certainty and $\varsigma = 1$ means that Σ can at best be guessed. Note that secrecy requires T_R to remain secret. Hence, the transformation to silent receiver AT from Section 3.5 does not preserve secrecy.

$\underline{\mathrm{Exp}^{\text{anon-ot}}_{\Pi_{UOT},\mathcal{A},\varpi}(\kappa)}$

$crs \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$

$(\Sigma_0, \Sigma_1, \mathsf{P}, st) \leftarrow \mathsf{A}_0(crs)$

$\pi \xleftarrow{\$} \mathsf{OT}_{\varpi(\mathsf{P}_1,\ldots,\mathsf{P}_N)}(crs, b, \Sigma)$

return $\mathsf{A}_1(\pi, T_\mathsf{P}, st)$

Fig. 8: Definition of the game $\mathrm{Exp}^{\text{anon-ot}}_{\Pi_{UOT},\mathcal{A},\varpi}(\kappa)$.

6.2 Definitions for Undetectable Oblivious Transfer

Towards our goal of undetectable multiparty computation we consider a notion of undetectable oblivious transfer. In Undetectable Oblivious Transfer we have a set of N parties. Two of which want to run an Oblivious Transfer protocol without the participants learning with whom they executed the OT, and without the $N-2$ other parties realizing that the protocol is executed.

Definition 9 ($(\varepsilon, \delta, c, \ell)$-Undetectable Oblivious Transfer). *An $(\varepsilon, \delta, c, \ell)$-Undetectable Oblivious Transfer for $\varepsilon, \delta \in \mathbb{R}_{[0,1]}$ and $\ell, c \in \mathbb{N}$ is a tuple containing three PPT algorithms ($\mathsf{Setup}, \mathsf{OT}$, $\mathsf{Reconstruct}$). The number of rounds in the OT protocol is given as c and the bitlength ℓ defines the length of the transferred message Σ_σ.*

The algorithms are defined as follows:

$\mathsf{Setup}(1^\kappa)$ *takes as input the security parameter 1^κ in unary encoding and outputs a Common Reference String crs.*

$\mathsf{OT}(crs, \varpi, \Sigma_0, \Sigma_1, \sigma)$ *is a c-round protocol that takes as input the Common Reference String crs, a permutation $\varpi \in \mathbb{S}_N$ to determine which of $(\mathsf{P}_1, \ldots, \mathsf{P}_N)$ is the sender, which is the receiver and which are the dummy friends, two messages Σ_0 and $\Sigma_1 \in \{0,1\}^\ell$ from the sender and one bit $\sigma \in \{0,1\}$ from the receiver and outputs a transcript π. The non-sender send independent uniformly distributed noise in each round.*

$\mathsf{Reconstruct}(crs, \pi, T_\mathsf{R})$ *is a local algorithm executed by the receiver that takes as input the CRS crs, protocol transcript π and the receiver's random tape T_R and outputs a message Σ_σ.*

The algorithms additionally satisfy the ε-correctness, the δ-anonymity and the privacy from Definitions 10 to 12.

Definition 10 (ε-Correctness). *An Undetectable Oblivious Transfer protocol between players $(\mathsf{P}_1, \ldots, \mathsf{P}_N)$ is ε-Correct if for any $\varpi \in \mathbb{S}_N$, any $(\Sigma_0, \Sigma_1) \in \{0,1\}^{2\ell}$, any $\sigma \in \{0,1\}$ and any $crs \leftarrow \mathsf{Setup}(1^\kappa)$,*

$$\Pr\left[\begin{array}{c} \pi \xleftarrow{\$} \mathsf{OT}_{\langle\varpi(\mathsf{P}_1,\ldots,\mathsf{P}_N)\rangle}(crs, (\Sigma_0, \Sigma_1), \sigma) \\ \Sigma'_\sigma \leftarrow \mathsf{Reconstruct}(crs, \pi, T_\mathsf{R}) \end{array} : \Sigma_\sigma = \Sigma'_\sigma\right] \geq \varepsilon \tag{21}$$

This is quite similar to the definition of correctness for AT from Definition 4.

Definition 11 (δ-Anonymity). *An Undetectable Oblivious Transfer protocol between players $(\mathsf{P}_1, \ldots, \mathsf{P}_N)$ is δ-Anonymous if for any PPT guessing algorithm $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$, it holds that*

$$\left| \Pr_{\varpi \xleftarrow{\$} \mathbb{S}_N}\left[\mathrm{Exp}^{\text{anon-ot}}_{\Pi_{UOT},\mathsf{A},\varpi}(\kappa) = \varpi\right] - 1/(N-1) \right| \leq (1-\delta) \cdot \frac{(N-2)}{(N-1)} \tag{22}$$

$\mathbb{S}_N$ is the symmetric group over N elements alongside all possible bijections into themselves and ϖ is a random permutation drawn from that set. This reflects the intuition that a *participant* is trying to recover the permutation *after* finishing the execution. The guessing algorithm A_0 selects which party to *corrupt* (in a semi-honest setting, hence post-execution) and which messages should be transferred. A_1 gets the *random tape* of the corrupted party and the *state* st of the adversary that selected which party to corrupt. The definition implies uncertainty of the sender regarding which of the two parties was acting as receiver and of the receiver regarding which of the two parties was the sender. Additionally we require that the dummy friend can only guess as well.

Definition 12 (Privacy). *An Undetectable Oblivious Transfer protocol between players* $(\mathsf{P}_1, \ldots, \mathsf{P}_N)$ *is private for the sender and the receiver if for any guessing algorithm* A*, any* $\varpi \in \mathbb{S}_N$*, any* $(\Sigma_0, \Sigma_1) \in \{0,1\}^{2\ell}$*, any* $\sigma \in \{0,1\}$ *and any* $crs \leftarrow \mathsf{Setup}(1^\kappa)$*, the following two conditions hold against every PPT guessing algorithm* A*:*

$$\Pr\left[\begin{matrix}\pi \leftarrow \mathsf{OT}_{\varpi(\mathsf{P}_1,\ldots,\mathsf{P}_N)}(crs, (\Sigma_0, \Sigma_1), \sigma) \\ \Sigma'_{\overline{\sigma}} \leftarrow \mathsf{A}(\pi, T_\mathsf{R})\end{matrix} : \Sigma_{\overline{\sigma}} = \Sigma'_{\overline{\sigma}}\right] \in \mathsf{negl}(\kappa) \tag{23}$$

$$\left|\Pr\left[\begin{matrix}\pi \leftarrow \mathsf{OT}_{\varpi(\mathsf{P}_1,\ldots,\mathsf{P}_N)}(crs, (\Sigma_0, \Sigma_1), \sigma) \\ \sigma' \leftarrow \mathsf{A}(\pi, T_\mathsf{S})\end{matrix} : \sigma = \sigma'\right] - 1/2\right| \in \mathsf{negl}(\kappa) \tag{24}$$

For the sender this means that even with the random tape T_R of the receiver it is not efficiently possible for the guessing algorithm A to predict the message $\Sigma_{\overline{\sigma}}$ that was *not* transferred. For the receiver this implies security of the choice despite having the random tape T_S of the sender.

7 Towards Undetectable Multiparty Computation

In this section, we make a first step to define Undetectable Multiparty Computation by providing an informal definition of the security requirement in Section 7.1 and a candidate instantiation for Undetectable Two-Party Computation (U2PC) in Section 7.2.

7.1 Defining Undetectable Multiparty Computation

In this section, we informally define the security requirements of Undetectable Multiparty Computation as the increment to the respective definition of Undetectable Oblivious Transfer from Section 6.2.

Again, we have N individuals, but now k of them are trying to compute a function f on k secret inputs.

ε-Correctness: This definition states that all players obtain the correct result of f evaluated on the k inputs.

δ-Anonymity: This definition states that any of the N players, each of the $\binom{N-1}{k-1}$ subsets of remaining individuals is equally likely to have participated in the execution.

Privacy: This definition enforces that each participants individual input remains private; that is, after the computation no party learns more on the other participants' input than what can be inferred from the result of the computation.

7.2 Towards constructing Undetectable Two-Party Computation from Undetectable Oblivious Transfer

We provide a candidate construction for the simpler case of $k = 2$ which we call Undetectable Two-Party Computation. That is, two parties out of N try to compute a bivariate function f on their respective secret inputs. To that end we use a covert two-party computation protocol like that from von Ahn, Hopper, and Langford [vHL05] that uses COT for communication and replace all invocations of COT between the sender and the receiver by invocations of Undetectable Oblivious Transfer between all participants. That way, the message is transferred from the sender to the receiver without leaking information regarding the identity of the respective parties.

Assuming a $(\varepsilon, \delta, c, \ell)$-Undetectable Oblivious Transfer with $\varepsilon \in \mathsf{owhl}(\kappa)$, the protocol Π_{UMPC} for computing a bivariate function f using Undetectable Two-Party Computation we obtain by replacing all COTs by UOTs in [vHL05, Protocol 3] is a $(\varepsilon, \delta, c \cdot |f|)$-Undetectable Two-Party Computation.

The protocol by von Ahn, Hopper, and Langford [vHL05] essentially uses garbled circuits [Yao86] and distributes the keys using COT instead of using classical OT. The protocol we propose replaces the COTs with UOTs where from the N parties only two are actually participating, one as sender and one as receiver.

Assuming that each UOT is correct with overwhelming probability the correctness of the entire scheme then follows from the correctness of garbled circuits.

Anonymity remains the same as in order to determine any of the participating parties, the only way is to de-anonymize any of the UOTs, the success probability of which is bounded by δ.

Privacy is inherited by the Garbled Circuits; since the Undetectable Oblivious Transfers are private as well our changes induce no additional leakage.

Finally, note that for a garbled circuit we require $|f|$ many OTs and each requires c rounds, resulting in the new number of rounds.

References

[APY20] I. Abraham, B. Pinkas, and A. Yanai. Blinder - scalable, robust anonymous committed broadcast. In J. Ligatti, X. Ou, J. Katz, and G. Vigna, editors, *ACM CCS 2020*, pages 1233–1252. ACM Press, November 2020.

[BEA14] B. Burrough, E Ellison, and S. Andrews. The snowden saga: a shadowland of secrets and light. *Vanity Fair*, 23, 2014.

[Ber16] C. Berret. Guide to securedrop, 2016.

[BGI08] E. Biham, Y. J. Goren, and Y. Ishai. Basing weak public-key cryptography on strong one-way functions. In R. Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 55–72. Springer, Heidelberg, March 2008.

[CBM15] H. Corrigan-Gibbs, D. Boneh, and D. Mazières. Riposte: an anonymous messaging system handling millions of users. In *2015 IEEE Symposium on Security and Privacy*, pages 321–338. IEEE Computer Society Press, May 2015.

[CDV+14] S. Chan, I. Diakonikolas, P. Valiant, and G. Valiant. Optimal algorithms for testing closeness of discrete distributions. In *25th SODA*, pages 1193–1203, 2014.

[CGCD+20] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila. A formal security analysis of the signal messaging protocol. *Journal of Cryptology*, 33(4):1914–1983, 2020.

[CGO+07] N. Chandran, V. Goyal, R. Ostrovsky, and A. Sahai. Covert multi-party computation. In *48th FOCS*, pages 238–248. IEEE Computer Society Press, October 2007.

[Cha03] D. Chaum. Untraceable electronic mail, return addresses and digital pseudonyms. In D. Gritzalis, editor, *Secure Electronic Voting*. Volume 7, Advances in Information Security, pages 211–219. Springer, 2003.

[Cha88] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, January 1988.

[CLT+15] R. Canetti, H. Lin, S. Tessaro, and V. Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 468–497. Springer, Heidelberg, March 2015.

[DMS04] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: the second-generation onion router. In M. Blaze, editor, *USENIX Security 2004*, pages 303–320. USENIX Association, August 2004.

[DVV16] A. Degwekar, V. Vaikuntanathan, and P. N. Vasudevan. Fine-grained cryptography. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 533–562. Springer, Heidelberg, August 2016.

[ECZ+21] S. Eskandarian, H. Corrigan-Gibbs, M. Zaharia, and D. Boneh. Express: lowering the cost of metadata-hiding communication with cryptographic privacy. In M. Bailey and R. Greenstadt, editors, *USENIX Security 2021*, pages 1775–1792. USENIX Association, August 2021.

[HLv02] N. J. Hopper, J. Langford, and L. von Ahn. Provably secure steganography. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 77–92. Springer, Heidelberg, August 2002.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147. IEEE, 1995.

[JLL+22] A. Jain, H. Lin, J. Luo, and D. Wichs. The pseudorandom oracle model and ideal obfuscation. Cryptology ePrint Archive, Report 2022/1204, 2022. https://eprint.iacr.org/2022/1204.

[Mer78] R. C. Merkle. Secure communications over insecure channels. *Commun. ACM*, 21(4):294–299, 1978.

[NSSD21] Z. Newman, S. Servan-Schreiber, and S. Devadas. Spectrum: high-bandwidth anonymous broadcast with malicious security. Cryptology ePrint Archive, Report 2021/325, 2021. https://eprint.iacr.org/2021/325.

[Phi18] D. Philipps. Reality winner, former nsa translator, gets more than 5 years in leak of russian hacking report. *New York Times*, 23, 2018.

[Rog04] P. Rogaway. Nonce-based symmetric encryption. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 348–359. Springer, Heidelberg, February 2004.

[Sim83] G. J. Simmons. The prisoners' problem and the subliminal channel. In D. Chaum, editor, *CRYPTO'83*, pages 51–67. Plenum Press, New York, USA, 1983.

[vH04] L. von Ahn and N. J. Hopper. Public-key steganography. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 323–341. Springer, Heidelberg, May 2004.

[vHL05] L. von Ahn, N. J. Hopper, and J. Langford. Covert two-party computation. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 513–522. ACM Press, May 2005.

[Whi] Whistleblowing. https://legal-dictionary.thefreedictionary.com/Whistleblowing. Accessed: 2021-09-29 from West's Encyclopedia of American Law, edition 2. (2008).

[Yao86] A. C.-C. Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.

Appendix

Table of Contents

1 Introduction . . . 1
- 1.1 Undetectable Secure Computation . . . 2
- 1.2 Defining Anonymous Transfer . . . 3
- 1.3 Impossibility Result . . . 4
- 1.4 A Candidate Fine-Grained Anonymous Transfer . . . 5
- 1.5 Discussions and Implications . . . 5
- 1.6 Further Results and Open Questions . . . 7
- 1.7 Acknowledgements . . . 7

2 Preliminaries . . . 7
- 2.1 Notations . . . 7
- 2.2 Steganography . . . 8
- 2.3 Distribution Testing . . . 9

3 Anonymous Transfer . . . 10
- 3.1 Network Model and Non-Participating Parties . . . 10
- 3.2 The Model . . . 10
- 3.3 Fine-grained Anonymous Transfer . . . 11
- 3.4 Trivial Anonymous Transfers . . . 12
- 3.5 Reductions Among AT Protocols . . . 13

4 Impossibility of Anonymous Transfer . . . 14
- 4.1 The Attacker . . . 15
- 4.2 Putting the Pieces Together . . . 19
- 4.3 Impossibility of Anonymous Transfer for $N > 3$ . . . 20
- 4.4 Extensions and Limitations . . . 23

5 Fine-Grained AT from Ideal Obfuscation . . . 24
- 5.1 Security Analysis . . . 25
- 5.2 Final Result . . . 29

6 Undetectable Oblivious Transfer . . . 29
- 6.1 Strong Anonymous Transfer . . . 29
- 6.2 Definitions for Undetectable Oblivious Transfer . . . 30

7 Towards Undetectable Multiparty Computation . . . 31
- 7.1 Defining Undetectable Multiparty Computation . . . 31
- 7.2 Towards constructing Undetectable Two-Party Computation from Undetectable Oblivious Transfer . . . 31

A Preliminaries . . . 36
- A.1 Covert Oblivious Transfer . . . 36
- A.2 Indistinguishability from Random Bits under Chosen Ciphertext Attacks . . . 36
- A.3 Strong Existential Unforgeability under Chosen Message Attacks . . . 37
- A.4 Ideal Obfuscation . . . 37

B Proof of Lemma 3 . . . 37

C Proof of Corollary 1 . . . 38

D Security proof for the Fine-Grained Protocol . . . 38
- D.1 Full Proof of Correctness . . . 38
- D.2 Graphical Depiction of the Anonymity Proof . . . 39
- D.3 Full Proof of Anonymity . . . 47
- D.4 Proof of Lemma 9 . . . 48
- D.5 Proof of Lemma 10 . . . 48
- D.6 Proof of Lemma 11 . . . 55
- D.7 Proof of Lemma 12 . . . 55
- D.8 Proof of Lemma 13 . . . 57

D.9 Proof of Lemma 14 57
D.10 Graphical Depiction of the Secrecy Proof Outline 58
D.11 Full Proof of Secrecy 63
E Asymptotically Secure AT in the Designated Sender Model 65
E.1 A two-round protocol 65
E.2 A c-rounds protocol. 98
E.3 On the security 103
F A fine-grained Anonymous Transfer for ℓ-bit messages 103
G Undetectable Oblivious Transfer from Anonymous Transfer and two-round Covert Oblivious Transfer 123
G.1 Correctness 124
G.2 Privacy 125
G.3 Anonymity 126

A Preliminaries

In this section we introduce some additional preliminaries we consider necessary for understanding the remainder of the appendix.

A.1 Covert Oblivious Transfer

Covert Oblivious Transfer is an extension of classical Oblivious Transfer first defined and used by von Ahn, Hopper, and Langford [vHL05]. It extends classical OT by two important properties that ensure the computation cannot be distinguished from an innocent-looking conversation: (1)the sender cannot distinguish between the case that the receiver is following the COT protocol or if the receiver is sending uniformly random messages, and (2)if the transferred messages are uniformly random, the receiver cannot distinguish between the case that the sender is following the COT protocol and the case where the sender is sending uniformly random messages.

The latter has to hold even after the protocol execution is finished, as the authors use Covert Oblivious Transfer during the execution of their Covert Multiparty Computation protocol and actual participation should not be revealed before the computation finishes.

Definition 13 (Covert Oblivious Transfer, [vHL05, Appendix A]). *A 2-party Covert Oblivious Transfer (COT) between a sender* S *and a receiver* R *is an Oblivious Transfer but additionally fulfills the following two requirements:*

(1) S *cannot distinguish between the case that* R *is following the COT protocol and the case that* R *is drawing uniformly random messages from* $\mathcal{U}_m$.
(2) If $\Sigma_0, \Sigma_1 \xleftarrow{\$} \mathcal{U}_m$, R *cannot distinguish between the case that* S *is following the COT protocol and the case that* S *is drawing uniformly random messages from* $\mathcal{U}_m$.

A.2 Indistinguishability from Random Bits under Chosen Ciphertext Attacks

For our instantiations it is crucial that protocol messages can be transferred covertly, that is, without bystanders noticing. To that end we require an encryption that yields ciphertexts which look like uniformly random strings. A formal property ensuring this was defined by Rogaway [Rog04] as Indistinguishability from Random Bits.

While the paper also defines IND$-CPA, we focus on chosen *ciphertext* attacks only and provide definitions for symmetric and asymmetric schemes. For symmetric encryption schemes the definition looks as follows:

Definition 14 (IND$-CCA for Symmetric Encryption). *A Symmetric Encryption Scheme* $\mathit{SKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *fulfills Indistinguishability from Random Bits under Chosen Ciphertext Attacks (IND$-CCA) if the following condition holds for all PPT valid adversaries* $\mathcal{A}$ *that do not decrypt a challenge:*

$$\begin{aligned} &|\Pr[\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\kappa) \colon \mathcal{A}^{\mathsf{Enc}(\mathsf{sk},\cdot),\mathsf{Dec}(\mathsf{sk},\cdot)} = 1] \\ &-\Pr[\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\kappa) \colon \mathcal{A}^{\mathsf{F}(\cdot),\mathsf{Dec}(\mathsf{sk},\cdot)} = 1]| \in \mathsf{negl}(\kappa) \end{aligned} \tag{25}$$

F is a random function. It is hence the adversaries task to distinguish whether it is interacting with a random oracle or with an actual encryption oracle.

The definition for asymmetric schemes is equivalent except that the adversary is additionally given a public key. Yet the challenge still comes from either an encryption oracle or a random oracle.

Definition 15 (IND$-CCA for Asymmetric Encryption). *An Asymmetric Encryption Scheme* $\mathit{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *fulfills Indistinguishability from Random Bits under Chosen Ciphertext Attacks (IND$-CCA) if the following condition holds for all PPT valid adversaries* $\mathcal{A}$ *that do not decrypt a challenge:*

$$\begin{aligned} &|\Pr[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa) \colon \mathcal{A}^{\mathsf{Enc}(\mathsf{sk},\cdot),\mathsf{Dec}(\mathsf{sk},\cdot)}(\mathsf{pk}) = 1] \\ &-\Pr[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa) \colon \mathcal{A}^{\mathsf{F}(\cdot),\mathsf{Dec}(\mathsf{sk},\cdot)}(\mathsf{pk}) = 1]| \in \mathsf{negl}(\kappa) \end{aligned} \tag{26}$$

A.3 Strong Existential Unforgeability under Chosen Message Attacks

For our construction we rely on a signature scheme. Their security is defined as follows [Gol04]:

Definition 16 (Strong Existential Unforgeability under Chosen Message Attacks for Signatures). *A signature scheme* $\mathit{SIG} = (\mathsf{KeyGen}, \mathsf{Sig}, \mathsf{Vfy})$ *fulfills Strong Existential Unforgeability under Chosen Message Attacks (sEUF-CMA) if for every PPT-adversary* $\mathcal{A}$ *with access to a signing oracle* $\mathsf{Sig}(\mathsf{k}, \cdot)$ *it holds that:*

$$\Pr\left[\begin{matrix}(\mathsf{vk}, \mathsf{k}) \leftarrow \mathit{SIG}.\mathsf{KeyGen}(1^\kappa) \\ (\mu^*, X^*) \leftarrow \mathcal{A}^{\mathsf{Sig}(\mathsf{k},\cdot)}(\mathsf{vk}, 1^\kappa)\end{matrix} : \mathit{SIG}.\mathsf{Vfy}(\mathsf{vk}, \mu^*, X^*) = 1\right] \in \mathsf{negl}(\kappa) \tag{27}$$

and for all queries X *to* $\mathsf{Sig}(\mathsf{k}, \cdot)$ *which were returned by a signature* μ *it holds that* $X \neq X^*$ *and* $\mu \neq \mu^*$.

A.4 Ideal Obfuscation

We use an abstract concept of idealized obfuscation which replaces a given program P directly with an oracle that evaluates the program on the given input. This notion has recently been rigorously studied by Jain, Lin, Luo, and Wichs [JLL+22], who treat it as a theoretical, idealized concept (comparable to the Random Oracle Model), rather than a security property (like Virtual Black-Box Obfuscation). Due to Jain, Lin, Luo, and Wichs [JLL+22], a proof under ideal obfuscation can be seen as heuristic evidence that the protocol is secure. In the following we adapt the definitions provided by Jain, Lin, Luo, and Wichs [JLL+22, Section 5].

Definition 17 (Ideal Obfuscation). *An ideal obfuscator describes the existence of two oracles* $(\mathsf{Obfuscate}, \mathsf{Evaluate})$*, where*

$\mathsf{Obfuscate}$ *takes as input a program* P *and randomly samples a handle* $h \in \{0,1\}^\kappa$*, saves* (P, h) *in a list and returns* h*;*

$\mathsf{Evaluate}$ *takes as input a handle* h *and an input* x *and searches for* h *in the list. If* h *is in the list, evaluate the corresponding program* P *on the given input* x *and return its output, otherwise output* $\perp$*.*

Ideal obfuscation is called secure *if for all efficient adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *there are efficient simulators* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ *such that the following advantage is negligible:*

$$\Pr\left[\begin{matrix}(P, st) \leftarrow \mathcal{A}_1^{O_{BF}}(1^\kappa) \\ P^* \leftarrow \mathsf{Obfuscate}(1^\kappa, P)\end{matrix} : \mathcal{A}_2^{O_{BF}}(P^*, st) = 1\right] - \Pr\left[\begin{matrix}(P, st) \leftarrow \mathcal{A}_1^{\mathcal{S}_1}(1^\kappa) \\ P^* \leftarrow \mathcal{S}_2^P(1^\kappa, 1^{|P|}, P)\end{matrix} : \mathcal{A}_2^{\mathcal{S}_3^P}(P^*, st) = 1\right]$$

B Proof of Lemma 3

As this lemma is crucial for proving security of our fine-grained construction of Anonymous Transfer in Section 5 we provide a formal proof here for the following lemma:

Lemma 3 (Distinguishing distributions based on the Total Variational Distance). *Let* p *and* q *be two distributions with total variational distance* $\mathsf{d_{TV}}(p, q)$*. If* $\mathsf{d_{TV}}(p, q) < \frac{1}{3}$*, then no algorithm can exist that distinguishes* p *and* q *with probability* $\geq \frac{2}{3}$ *based on a single sample.*

We stress here that this is not our proof but instead adapted from http://cs.brown.edu/courses/csci1951-w/lec/lec%2011%20notes.pdf. Since we did not find a citeable source our adapted proof follows, which basically follows the source:

Proof. Let $\mathcal{D}$ be the distinguisher between p and q. Due to symmetry it holds for the correctness that the optimal algorithm has:

$$\begin{aligned}\Pr[\mathsf{out}_\mathcal{D} = p|p] - \Pr[\mathsf{out}_\mathcal{D} = p|q] &= \Pr[\mathsf{out}_\mathcal{D} = p|q] - \Pr[\mathsf{out}_\mathcal{D} = q|q] \\ &= \mathsf{d_{TV}}(p, q)\end{aligned} \tag{28}$$

With a single sample the adversary can only output a single bit indicating whether it thinks the sample is from p or from q. This can not be done with better than $\mathsf{d_{TV}}(p,q)$. To get an intuition as to why this is the case, the total variational distance can also be written as:

$$\mathsf{d_{TV}}(p,q) \coloneqq \sup_{X \subseteq [n]} p(X) - q(X) \tag{29}$$

that is, it corresponds to the largest difference in the probability of occurrence of some output. Thus, no distinguisher can—with a single sample—guess correctly with larger advantage than $\mathsf{d_{TV}}(p,q)$. Guessing is always an option, which would result in the distinguisher being correct with probability $1/2$. Thus, there can be no distinguisher which fulfills *both* these inequalities:

$$\begin{aligned} \Pr[\mathsf{out}_{\mathcal{D}} = p | p] &> \frac{1}{2} + \frac{1}{2} \cdot \mathsf{d_{TV}}(p,q) \\ \Pr[\mathsf{out}_{\mathcal{D}} = q | q] &> \frac{1}{2} + \frac{1}{2} \cdot \mathsf{d_{TV}}(p,q) \end{aligned} \tag{30}$$

as this would imply an overall correctness larger than $\mathsf{d_{TV}}(p,q)$.

By using the requirement that $\mathsf{d_{TV}}(p,q) < 1/3$ in Eq. (30) we get the bound of $2/3$ on the correctness of $\mathcal{D}$. □

C Proof of Corollary 1

In this section we provide a full proof of the following corollary:

Corollary 1 (Distinguishing two Bernoulli-Distributions with t samples). *Any distinguisher $\mathcal{D}$ that distinguishes between p and q with probability $\geq \frac{1}{2} + \frac{\alpha}{2}$ requires $t \in \Omega\left(\frac{\alpha}{\mathsf{d_{TV}}(p,q)}\right)$ samples.*

Proof. The total variational distance limits the advantage as $\alpha = \mathsf{d_{TV}}(p^{\otimes t}, q^{\otimes t})$.

We know from the subadditivity of $\mathsf{d_{TV}}$ that

$$\mathsf{d_{TV}}(p^{\otimes t}, q^{\otimes t}) \leq t \cdot \mathsf{d_{TV}}(p,q) \tag{31}$$

and as such,

$$\Pr[\mathcal{D} \text{ correct}] \leq 1/2 + 1/2 \cdot t \cdot \mathsf{d_{TV}}(p,q) \tag{32}$$

We have that $t \in \Omega\left(\frac{\alpha}{\mathsf{d_{TV}}(p,q)}\right) \implies t \geq \frac{\alpha \cdot C}{\mathsf{d_{TV}}(p,q)}$. We claim that this even holds for $C = \frac{1}{4}$. This implies for Lemma 1 that

$$\begin{aligned} \mathsf{d_{TV}}(p^{\otimes t}, q^{\otimes t}) &\leq \frac{C \cdot \alpha}{\mathsf{d_{TV}}(p,q)} \mathsf{d_{TV}}(p,q) \\ &= \alpha C = \frac{\alpha}{4} \end{aligned} \tag{33}$$

Thus, the total variation distance is less than $\alpha/4$ for $\alpha \leq 1$, which in term is less than $1/3$.

Remember that $\mathsf{d_{TV}}(p^{\otimes t}, q^{\otimes t})$ corresponds to distinguishing the t-fold (binomial) distribution with a single sample. Thus Lemma 3 states that with total variation distance less than $1/3$ no distinguisher $\mathcal{D}$ can successfully distinguish with probability $\geq 2/3$. □

D Security proof for the Fine-Grained Protocol

D.1 Full Proof of Correctness

In this section we provide the full proof of correctness for the fine-grained instantiation of Anonymous Transfer. In particular, we provide a proof for the following lemma:

Theorem 4 (Correctness). *If the protocol from Fig. 5 is instantiated with an Ideally Obfuscated version of the circuit from Fig. 6 the protocol is ε-correct with $\varepsilon = (1 - \mathsf{negl}(\kappa))$.*

Proof. In a correct execution, there is one sender P_b who encodes the bit σ_b in the first round. Every round that follows has to have a valid signature μ on both messages of the previous round. With honest people doing that, the only way that correctness is not given is if the circuit confuses P_{1-b} to be P_b. This only happens if the inputs sent by P_{1-b} are interpreted as valid inputs, causing the **argmax** to return the bit of P_{1-b}.

Since we assume the sending party to act honestly, the value of χ_b is c which is as high as this variable can go. Thus, if the **argmax** ever returns $(1-b)$, then P_{1-b} has sent for c rounds a valid encryption of $\sigma_{1-b} \coloneqq (1-\sigma_b)$ (if we assume a non-correct output, the bit has to differ, naturally) and a valid signature on $\pi[\chi - 1]$ in each of the c rounds.

By requirement, PKE has sparse ciphertexts. With P_{1-b} sending uniformly random ciphertexts, the probability that this is a valid ciphertext is negligible in κ.

Assuming that P_{1-b} had a valid ciphertext by chance, the probability that it encodes the same bit that was decrypted in the first round is given by $1/2$; yet the probability that the signature verifies the previous round is negligible.

Now we have χ rounds. In each round we have a probability of $\mathsf{negl}(\kappa) \cdot 1/2 \cdot \mathsf{negl}(\kappa)$ that the message is valid. So for P_{1-b} to accidentally send valid messages of $1-\sigma_b$, the probability is given by $(\mathsf{negl}(\kappa) \cdot 1/2 \cdot \mathsf{negl}(\kappa))^c$. Additionally, in round 0 there has to be a valid encryption of $1-\sigma_b$, which also happens with negligible probability as once more we have the sparseness of PKE. And in the negligible case that the message counts as cipher, there is only a probability of $1/2$ that the message encodes the correct bit.

Thus, in an honest protocol execution, the output of the circuit is $\sigma'_b = OTP \oplus \sigma_b$ with overwhelming probability. Thus by outputting $OTP \oplus \sigma'_b = OTP \oplus OTP \oplus \sigma_b = \sigma_b$ the receiver will output the correct bit with overwhelming probability.

Thus according to the definition of ε-correctness from Eq. (3) we get $\varepsilon \in (1 - \mathsf{negl}(\kappa))$. This concludes our proof. □

D.2 Graphical Depiction of the Anonymity Proof

$\mathrm{Game}_1(\kappa)$

1 : –
2 : –
3 : –
4 : –
5 : –
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : $(OTP, \mathsf{vk}_\mathsf{R}) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_\mathsf{R}^{(0)})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_\mathsf{R}, \pi)$ **then**
20 : **return** $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
21 : $\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1,\dots,c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\sigma'_b \coloneqq X_b[0]$
27 : $\mu_b^\chi \coloneqq X_b[1\colon |X_b|]$
28 : **if** $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $(\mathsf{k}_\mathsf{R}, \mathsf{vk}_\mathsf{R}) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
4 : $x_\mathsf{R}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_\mathsf{R}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
7 : $(\mathsf{sk}_b) \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^\kappa)$
8 : $(\mu_b^{(0)}) \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x_\mathsf{R}^{(0)})$
9 : $x_b^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1,\dots,c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : –
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19 : R:
20 : $\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_\mathsf{R}, \pi)$
21 : **return** $\mathcal{S}_\mathcal{A}^{P'_{AT}}(\pi, T_\mathsf{R})$

$\mathrm{Game}_2(\kappa)$

1 : **if** $\pi = \pi_\mathcal{C}$ **then**
2 : **return** $OTP_\mathcal{C} \oplus \sigma_\mathcal{C}$
3 : –
4 : –
5 : –
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : $(OTP, \mathsf{vk}_\mathsf{R}) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_\mathsf{R}^{(0)})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_\mathsf{R}, \pi)$ **then**
20 : **return** $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
21 : $\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1,\dots,c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\sigma'_b \coloneqq X_b[0]$
27 : $\mu_b^\chi \coloneqq X_b[1\colon |X_b|]$
28 : **if** $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $(\mathsf{k}_\mathsf{R}, \mathsf{vk}_\mathsf{R}) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
4 : $x_\mathsf{R}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_\mathsf{R}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
7 : $(\mathsf{sk}_b) \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^\kappa)$
8 : $(\mu_b^{(0)}) \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x_\mathsf{R}^{(0)})$
9 : $x_b^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1,\dots,c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : –
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19 : R:
20 : $\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_\mathsf{R}, \pi)$
21 : **return** $\mathcal{S}_\mathcal{A}^{P'_{AT}}(\pi, T_\mathsf{R})$

$\mathrm{Game}_2(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
–
–
–
–
–
–
–
–
–
–
–
$(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
    if $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
P_{b}:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$(\mathsf{sk}_b) \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^\kappa)$
$(\mu_b^{(0)}) \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  $\mu^{(\chi)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, \pi[0]) = \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
–
–
–
–
–
–
–
–
–
–
–
$(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
    if $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
P_{b}:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$(\mathsf{sk}_b) \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^\kappa)$
$(\mu_b^{(0)}) \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  $\mu^{(\chi)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$

1 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
2 : **return** $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
3 : **elseif** $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
4 : $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
5 : **abort**
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq \mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
20 : **return** $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
21 : $\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\sigma'_b \coloneqq X_b[0]$
27 : $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
28 : **if** $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
4 : $x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
7 : $(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^\kappa)$
8 : $(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
9 : $x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : –
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19 : R:
20 : $\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
21 : **return** $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

1 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
2 : **return** $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
3 : **elseif** $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
4 : $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
5 : **abort**
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : **if** $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
14 : $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
20 : **return** $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
21 : $\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\sigma'_b \coloneqq X_b[0]$
27 : $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
28 : **if** $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
4 : $x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
7 : $(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^\kappa)$
8 : $(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
9 : $x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : –
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19 : R:
20 : $\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
21 : **return** $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

1 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
2 : **return** $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
3 : **elseif** $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
4 : $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
5 : **abort**
6 : $-$
7 : $-$
8 : $-$
9 : $-$
10 : $-$
11 : $-$
12 : $-$
13 : **if** $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
14 : $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
20 : **return** $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
21 : $\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\sigma'_b \coloneqq X_b[0]$
27 : $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
28 : **if** $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
4 : $x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
7 : $(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^\kappa)$
8 : $(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
9 : $x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : $-$
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19 : R:
20 : $\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
21 : **return** $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_5(\kappa)$

1 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
2 : **return** $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
3 : **elseif** $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
4 : $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
5 : **abort**
6 : **elseif** $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
7 : $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
8 : **if** $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
9 : $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ **then**
10 : $\chi^* \coloneqq \chi^* + 1$
11 : $p \coloneqq 1/2 + \chi^*/2c$
12 : **return** $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
13 : **if** $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
14 : $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
20 : **return** $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
21 : $\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\sigma'_b \coloneqq X_b[0]$
27 : $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
28 : **if** $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
4 : $x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
7 : $(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^\kappa)$
8 : $(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
9 : $x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : $-$
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19 : R:
20 : $\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
21 : **return** $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_5(\kappa)$

1: **if** $\pi = \pi_{\mathcal{C}}$ **then**
2: **return** $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
3: **elseif** $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
4: $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
5: **abort**
6: **elseif** $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
7: $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
8: **if** $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
9: $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ **then**
10: $\chi^* \coloneqq \chi^* + 1$
11: $p \coloneqq 1/2 + \chi^*/2c$
12: **return** $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}_{(p)}^{(\pi)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
13: **if** $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
14: $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
15: $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16: $(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17: $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18: $(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19: **if** $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
20: **return** $\mathsf{CointossS}_{(1/2)}^{(\pi)}(0,1)$
21: $\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
22: $\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
23: **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24: **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25: $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26: $\sigma'_b \coloneqq X_b[0]$
27: $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
28: **if** $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
29: $\chi_b \coloneqq \chi$
30: $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31: **return** $OTP \oplus \mathsf{CointossS}_{(1/2+\chi_b/2c)}^{(\pi)}(\sigma^{(b')}, 1-\sigma^{(b')})$
32:

1: R:
2: $OTP \xleftarrow{\$} \{0,1\}$
3: $(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
4: $x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
5: $\mathsf{P_b}$:
6: $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
7: $(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^\kappa)$
8: $(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
9: $x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
10: $\mathsf{P}_{\overline{\mathsf{b}}}$:
11: $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12: **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13: $\mathsf{P_b}$:
14: $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15: $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
16: $\mathsf{P}_{\overline{\mathsf{b}}}$:
17: $-$
18: $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19: R:
20: $\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
21: **return** $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_6(\kappa)$

1: **if** $\pi = \pi_{\mathcal{C}}$ **then**
2: **return** $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
3: **elseif** $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
4: $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
5: **abort**
6: **elseif** $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
7: $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
8: **if** $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
9: $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ **then**
10: $\chi^* \coloneqq \chi^* + 1$
11: $p \coloneqq 1/2 + \chi^*/2c$
12: **return** $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}_{(p)}^{(\pi)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
13: **if** $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
14: $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
15: $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16: $(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17: $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18: $(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19: **if** $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
20: **return** $\mathsf{CointossS}_{(1/2)}^{(\pi)}(0,1)$
21: $\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
22: $\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
23: **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24: **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25: $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26: $\sigma'_b \coloneqq X_b[0]$
27: $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
28: **if** $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
29: $\chi_b \coloneqq \chi$
30: $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31: **return** $OTP \oplus \mathsf{CointossS}_{(1/2+\chi_b/2c)}^{(\pi)}(\sigma^{(b')}, 1-\sigma^{(b')})$
32:

1: R:
2: $OTP \xleftarrow{\$} \{0,1\}$
3: $(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
4: $x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
5: $\mathsf{P_b}$:
6: $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
7: $(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^\kappa)$
8: $(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
9: $x_b^{(0)} \xleftarrow{\$} \{0,1\}^m \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
10: $\mathsf{P}_{\overline{\mathsf{b}}}$:
11: $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12: **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13: $\mathsf{P_b}$:
14: $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15: $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
16: $\mathsf{P}_{\overline{\mathsf{b}}}$:
17: $-$
18: $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19: R:
20: $\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
21: **return** $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_6(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ then
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ then
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ then
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ then
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ then
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ then
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ do
  foreach $b \in \{0,1\} | \chi_b > c$ do
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
    if $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ then
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^\kappa)$
$(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^m \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ do
  P_{b}:
  $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_7(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ then
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ then
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ then
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ then
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ then
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ then
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ do
  foreach $b \in \{0,1\} | \chi_b > c$ do
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
    if $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ then
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ do
  P_{b}:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_7(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| \mu_{\mathcal{C}}{}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
    if $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
–
–
–
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  –
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_8(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  –
  –
  –
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{\chi} \coloneqq X_b[1\colon |X_b|]$
    if $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
–
–
–
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  –
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

D.3 Full Proof of Anonymity

Theorem 5 (Anonymity). *Let PKE be an IND$-CCA secure asymmetric encryption scheme, let SKE be a tightly secure multi-challenge IND$-CCA secure symmetric encryption scheme, let SIG be an sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, let F be a secure PRF, and let κ be the security parameter. Then the c-round protocol Π^1_{AT} for $N = 3$ satisfies δ-anonymity for all adversaries in $\mathfrak{C}_2 := o(c^2(1-\delta))$.*

We proceed over a series of games.

$\mathrm{Game}^{\sigma}_1(\kappa)$ This is the original game (after replacing the obfuscated circuit with oracle access to the circuit, the PRF with an actual random oracle and the adversary with the simulator), where the sending party P_b is chosen uniformly at random.

$\mathrm{Game}^{\sigma}_2(\kappa)$ This game follows $\mathrm{Game}^{\sigma}_1(\kappa)$, but during the simulation of the oracle P_{AT} from Fig. 5 the simulation enforces correctness of the challenge transcript $\pi_{\mathcal{C}}$: if the input transcript π matches the challenge transcript $\pi_{\mathcal{C}}$, it returns $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$. Due to Theorem 4, no PPT adversary can distinguish between $\mathrm{Game}^{\sigma}_1(\kappa)$ and $\mathrm{Game}^{\sigma}_2(\kappa)$.

$\mathrm{Game}^{\sigma}_3(\kappa)$ This game follows $\mathrm{Game}^{\sigma}_2(\kappa)$ but during simulation of the circuit the adversary aborts if any of the first-round messages from P_0 or P_1 differ from the messages reported in the challenge transcript *and* the decryptions still yield the same verification key or symmetric key. This game hop is justified by the non-malleability of PKE. See Appendix D.5 for the full proof.

$\mathrm{Game}^{\sigma}_4(\kappa)$ This game follows $\mathrm{Game}^{\sigma}_3(\kappa)$ but simulates the circuit slightly different: If the first receiver message of the input transcript π is the same as that of the challenge transcript $\pi_{\mathcal{C}}$, instead of decrypting it the circuit directly sets $OTP = OTP_{\mathcal{C}}$ and $\mathsf{vk}_{\mathsf{R}} = \mathsf{vk}_{\mathcal{C}\mathsf{R}}$ as the values used in the creation of the challenge transcript. This game hop is justified by the perfect correctness of PKE, as we show in Appendix D.6.

$\mathrm{Game}^{\sigma}_5(\kappa)$ This game follows $\mathrm{Game}^{\sigma}_4(\kappa)$ but simulates the oracle differently if the first-round messages of both parties match the first-round messages in the challenge transcript $\pi_{\mathcal{C}}$. In this case, the program compares the input transcript π with the challenge transcript $\pi_{\mathcal{C}}$ until it finds the first round χ^* in which the input differs from the challenge transcript. It then checks round $\chi^* + 1$, and if it contains the same message from the sending party, it adds one to χ.
Finally, the circuit flips a biased coin, which returns the correct bit $\sigma_{\mathcal{C}}$ with probability $p := 1/2 + \chi^*/2c$ and the complementary bit $(1 - \sigma_{\mathcal{C}})$ otherwise.
This game hop is indistinguishable due to the sEUF-CMA security of Sig. The only way to make the output distribution of the circuit oracle differ in both games is by supplying a valid signature forgery. See Appendix D.7 for the full proof.

$\mathrm{Game}^{\sigma}_6(\kappa)$ This game is the same as $\mathrm{Game}^{\sigma}_5(\kappa)$, but in creating the challenge transcript $\pi_{\mathcal{C}}$, this game only reports randomness for the first-round message x^0_b that specifies the symmetric key sk_b and the verification key vk_b to be used for the remaining communication with the circuit alongside the receiver message $x^{(0)}_{\mathsf{R}}$ that specifies the One-Time-Pad and the verification key. Note that all keys are still created as they are needed for the remaining rounds.
This game hop is justified by the IND$-CCA security of PKE. See Appendix D.8 for the full proof.

$\mathrm{Game}^{\sigma}_7(\kappa)$ This game is the same as $\mathrm{Game}^{\sigma}_6(\kappa)$ but in creating the challenge transcript $\pi_{\mathcal{C}}$, this game also reports randomness instead of transcripts for all messages x^{χ}_b for $\chi \in [c]$ that shift the bit towards $\sigma_{\mathcal{C}}$. That means that instead of using the IND$-CPA secure symmetric scheme SKE with the symmetric key sk_b the challenge transcript now only contains randomly sampled messages.
We also do not let the adversary create the keys for SIG and SKE as they are no longer needed for creating the transcript.
This game hop is justified by the IND$-CCA security of SKE. See Appendix D.9 for the full proof.

$\mathrm{Game}^{\sigma}_8(\kappa)$ This game follows $\mathrm{Game}^{\sigma}_7(\kappa)$, but instead of choosing a random sender at the beginning of the game and considering this message to be the right one, the oracle ignores the additional check and only looks for the first round where *both* messages are identical to the challenge.

Note that $\mathrm{Game}^{\sigma}_8(\kappa)$ is entirely independent of the real sender, hence given a challenge transcript, it is trivially impossible to obtain a non-negligible advantage to determine the sending party.

D.4 Proof of Lemma 9

Lemma 9 (Indistinguishability of $\mathrm{Game}_1^\sigma(\kappa)$ and $\mathrm{Game}_2^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme, let SKE be a tightly secure multi-challenge IND$-CCA secure symmetric encryption scheme, let SIG be an sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, and let F be a secure PRF. Then, for all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_1^\sigma(\kappa)$ and $\mathrm{Game}_2^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_1^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_2^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Indistinguishability automatically follows from the fact that both simulated circuits behave identical with overwhelming probability due to Theorem 4. □

D.5 Proof of Lemma 10

Lemma 10 (Indistinguishability of $\mathrm{Game}_2^\sigma(\kappa)$ and $\mathrm{Game}_3^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme, let SKE be a tightly secure multi-challenge IND$-CCA secure symmetric encryption scheme, let SIG be an sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, and let F be a secure PRF. Then, for all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_2^\sigma(\kappa)$ and $\mathrm{Game}_3^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_2^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_3^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. In order to successfully cause a situation where $\mathrm{Game}_2^\sigma(\kappa)$ would yield output whereas $\mathrm{Game}_3^\sigma(\kappa)$ aborts a distinguisher $\mathcal{D}$ would have to essentially *rerandomize* the first ciphertext.

To that end we use several hybrids to capture all the cases in which rerandomization might occur. We start with the case where the same verification key vk is encoded and then handle the case that the ciphertext has the same symmetric key sk.

H_0

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$

H_1

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{b_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort

H_2

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{b_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}}) \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{\bar{b}_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{\bar{b}_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort

H_3

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{b_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{\bar{b}_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{\bar{b}_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} = \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{\bar{b}_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{\bar{b}_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort

H_4

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{b_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{\bar{b}_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{\bar{b}_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} = \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{\bar{b}_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{\bar{b}_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{b_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) \neq \perp$ **then**
  abort

H_5

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{b_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{\bar{b}_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{\bar{b}_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} = \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{\bar{b}_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{\bar{b}_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{b_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) \neq \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{\mathcal{C}}_{\bar{b}_{\mathcal{C}}}) \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{\bar{b}_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{\bar{b}_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b}_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{\bar{b}_{\mathcal{C}}}) \neq \perp$ **then**
  abort

H_6

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{b_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\cdot, \mathsf{vk}_{\bar{b_{\mathcal{C}}}}) \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}\,(0)}_{\bar{b_{\mathcal{C}}}} \neq \bot \wedge$
  $x^{(0)}_{\bar{b_{\mathcal{C}}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b_{\mathcal{C}}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\cdot, \mathsf{vk}_{\bar{b_{\mathcal{C}}}}) \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}\,(0)}_{\bar{b_{\mathcal{C}}}} = \bot \wedge$
  $x^{(0)}_{\bar{b_{\mathcal{C}}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b_{\mathcal{C}}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\cdot, \mathsf{vk}_{b_{\mathcal{C}}}) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) \neq \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\cdot, \mathsf{vk}_{\bar{b_{\mathcal{C}}}}) \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{\mathcal{C}}_{\bar{b_{\mathcal{C}}}}) \neq \bot \wedge$
  $x^{(0)}_{\bar{b_{\mathcal{C}}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b_{\mathcal{C}}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{\bar{b_{\mathcal{C}}}}) \neq \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\cdot, \mathsf{vk}_{\bar{b_{\mathcal{C}}}}) \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}\,(0)}_{\bar{b_{\mathcal{C}}}} = \bot \wedge$
  $x^{(0)}_{\bar{b_{\mathcal{C}}}} \neq x^{\mathcal{C}\,(0)}_{\bar{b_{\mathcal{C}}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) \neq \bot$ **then**
  abort

H_7

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\exists_{b \in \{0,1\}} x^{\mathcal{C}\,(0)}_{b} \neq x^{(0)}_{b} \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x^{(0)}_{b}) = (\cdot, \mathsf{vk}_b)$ **then**
  abort

H_8

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\exists_{b \in \{0,1\}} x^{\mathcal{C}\,(0)}_{b} \neq x^{(0)}_{b} \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x^{(0)}_{b}) = (\cdot, \mathsf{vk}_b)$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\mathsf{sk}_{b_{\mathcal{C}}}, \cdot) \wedge$
  $x^{(0)}_{b_{\mathcal{C}}} \neq x^{\mathcal{C}\,(0)}_{b_{\mathcal{C}}} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{(0)}_{b_{\mathcal{C}}}) = \bot$ **then**
  abort

H_9

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\exists_{b\in\{0,1\}} x^{\mathcal{C}}{}_b^{(0)} \neq x_b^{(0)} \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_b^{(0)}) = (\cdot, \mathsf{vk}_b)$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\mathsf{sk}_{b_{\mathcal{C}}}, \cdot) \wedge$
  $x_{b_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{b_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\mathsf{sk}_{\bar{b_{\mathcal{C}}}}, \cdot) \wedge$
  $x_{\bar{b_{\mathcal{C}}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}^{(0)}) = \perp$ **then**
  abort

H_{10}

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\exists_{b\in\{0,1\}} x^{\mathcal{C}}{}_b^{(0)} \neq x_b^{(0)} \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_b^{(0)}) = (\cdot, \mathsf{vk}_b)$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\mathsf{sk}_{b_{\mathcal{C}}}, \cdot) \wedge$
  $x_{b_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{b_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\mathsf{sk}_{\bar{b_{\mathcal{C}}}}, \cdot) \wedge$
  $x_{\bar{b_{\mathcal{C}}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}^{(0)}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} = \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\mathsf{sk}_{\bar{b_{\mathcal{C}}}}, \cdot) \wedge$
  $x_{\bar{b_{\mathcal{C}}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) = \perp$ **then**
  abort

H_{11}

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\exists_{b\in\{0,1\}} x^{\mathcal{C}}{}_b^{(0)} \neq x_b^{(0)} \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_b^{(0)}) = (\cdot, \mathsf{vk}_b)$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\mathsf{sk}_{b_{\mathcal{C}}}, \cdot) \wedge$
  $x_{b_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{b_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\mathsf{sk}_{\bar{b_{\mathcal{C}}}}, \cdot) \wedge$
  $x_{\bar{b_{\mathcal{C}}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}^{(0)}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} = \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\mathsf{sk}_{\bar{b_{\mathcal{C}}}}, \cdot) \wedge$
  $x_{\bar{b_{\mathcal{C}}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\mathsf{sk}_{b_{\mathcal{C}}}, \cdot) \wedge$
  $x_{b_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{b_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) \neq \perp$ **then**
  abort

H_{12}

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\exists_{b\in\{0,1\}} x^{\mathcal{C}}{}_b^{(0)} \neq x_b^{(0)} \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_b^{(0)}) = (\cdot, \mathsf{vk}_b)$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\mathsf{sk}_{b_{\mathcal{C}}}, \cdot) \wedge$
  $x_{b_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{b_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\mathsf{sk}_{\bar{b_{\mathcal{C}}}}, \cdot) \wedge$
  $x_{\bar{b_{\mathcal{C}}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}^{(0)}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} = \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\mathsf{sk}_{\bar{b_{\mathcal{C}}}}, \cdot) \wedge$
  $x_{\bar{b_{\mathcal{C}}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) = \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\mathsf{sk}_{b_{\mathcal{C}}}, \cdot) \wedge$
  $x_{b_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{b_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) \neq \perp$ **then**
  abort
elseif $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}) \neq \perp \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}) = (\mathsf{sk}_{\bar{b_{\mathcal{C}}}}, \cdot) \wedge$
  $x_{\bar{b_{\mathcal{C}}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b_{\mathcal{C}}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\bar{b_{\mathcal{C}}}}^{(0)}) \neq \perp$ **then**
  abort

H_{13}

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\exists_{b\in\{0,1\}} x^{\mathcal{C}}{}_b^{(0)} \neq x_b^{(0)} \wedge$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_b^{(0)}) = (\cdot, \mathsf{vk}_b)$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\mathsf{sk}_{b_{\mathcal{C}}}, \cdot) \wedge$
  $x_{b_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{b_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) = \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\mathsf{sk}_{\bar{b}_{\mathcal{C}}}, \cdot) \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b}_{\mathcal{C}}}^{(0)} \neq \bot \wedge$
  $x_{\bar{b}_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b}_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}^{(0)}) = \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\mathsf{sk}_{\bar{b}_{\mathcal{C}}}, \cdot) \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b}_{\mathcal{C}}}^{(0)} = \bot \wedge$
  $x_{\bar{b}_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b}_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}^{(0)}) = \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{b_{\mathcal{C}}}) = (\mathsf{sk}_{b_{\mathcal{C}}}, \cdot) \wedge$
  $x_{b_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{b_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{b_{\mathcal{C}}}^{(0)}) \neq \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\mathsf{sk}_{\bar{b}_{\mathcal{C}}}, \cdot) \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x^{\mathcal{C}}{}_{\bar{b}_{\mathcal{C}}}) \neq \bot \wedge$
  $x_{\bar{b}_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b}_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}^{(0)}) \neq \bot$ **then**
  abort
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}) = (\mathsf{sk}_{\bar{b}_{\mathcal{C}}}, \cdot) \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P), x^{\mathcal{C}}{}_{\bar{b}_{\mathcal{C}}}^{(0)} = \bot \wedge$
  $x_{\bar{b}_{\mathcal{C}}}^{(0)} \neq x^{\mathcal{C}}{}_{\bar{b}_{\mathcal{C}}}^{(0)} \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\bar{b}_{\mathcal{C}}}^{(0)}) \neq \bot$ **then**
  abort

H_{14}

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$
elseif $\exists_{b\in\{0,1\}} x^{\mathcal{C}}{}_b^{(0)} \neq x_b^{(0)} \wedge$
  $\big(\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_b^{(0)}) = (\mathsf{sk}_b, \cdot) \vee$
  $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_b^{(0)}) = (\cdot, \mathsf{vk}_b)\big)$ **then**
  abort

H_0 is the game from $\mathrm{Game}_2^{\sigma}(\kappa)$.

H_1 is as H_0 but the circuit aborts only if the first-round message of the sending party $\mathsf{P}_{b_{\mathcal{C}}}$ in the input transcript encrypts the same verification key as that from the challenge transcript and the decryption of the new first-round message is from the random oracle—that is, the decryption of that message using the first-round message of the input transcript using the actual decryption algorithm yields the error symbol $\bot$.
So the goal of the adversary is to distinguish, which is only possible if a different pre-image for the random oracle is found which evaluates to something that shares the same verification key as the random oracle when evaluated with the first sending parties ciphertext. However, as we already use a genuine random oracle in this game, this is information-theoretically impossible and hence the adversary cannot distinguish.
Thus it holds that

$$Adv_{\mathcal{D}} = |\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_1} = 1] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_0=1}]| \in \mathsf{negl}(\kappa) \tag{34}$$

Now let E be the event that $\mathcal{D}$ can find a collision for a fixed value. We can incorporate this event into Eq. (34) as follows:

$$\begin{aligned} Adv_{\mathcal{D}} =& |\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_1} = 1] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_0} = 1]| \\ =& |\Pr[E] \cdot (\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_1} = 1|E] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_0} = 1|E]) \\ &+ (1 - \Pr[E]) \cdot (\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_1} = 1|\neg E] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_0} = 1|\neg E])| \end{aligned} \tag{35}$$

It trivially holds that $\neg E \implies \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_1} = 1] = \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_0}]$ as there the two games act exactly the same. Hence it holds for the second line of Eq. (35) that $(1 - \Pr[E]) \cdot (\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_1} = 1|\neg E] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_0} = 1|\neg E]) = 0$. So it holds that:

$$Adv_{\mathcal{D}} = |\Pr[E] \cdot (\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_1} = 1|E] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_0} = 1|E])| \tag{36}$$

Which holds information-theoretically due to the properties of the random oracle. Thus it holds that $\Pr[E] \in \mathsf{negl}(\kappa)$ and hence $Adv_{\mathcal{D}}$ is negligible.

This concludes our proof.

H_2 is as H_1 only that the circuit additionally aborts if (1) the first message of the dummy friend in the *challenge transcript* is a **valid** ciphertext, and (2) the first message of the dummy friend in the *input transcript* is an **invalid** ciphertext, and (3) the first message of the dummy friend in the *challenge transcript* differs from the first message of the dummy friend in the *input transcript*, and (4) the Dec^* algorithm outputs the same verification key for the challenge- and input-transcript.

With the message in the challenge transcript being a valid ciphertext we can reduce a distinguisher $\mathcal{D}$ between H_2 and H_1 to an adversary breaking the pre-image resistance of the random oracle. Let E be the event that $\mathcal{D}$ can find a different pre-image for which the second half, which is interpreted as the verification key, is the same as for the challenge message. In this case this pre-image has to be a value that is mapped by the random oracle to the value that was reported by the adversary as first dummy-friend message.

First note that the distinguisher can only ever win this game if the challenge-transcript contains an invalid ciphertext for the dummy friend in the first round. Denote by C the event that while sampling a random value for the first message of the dummy friend the message is not a valid ciphertext. We stress that $\neg C$ implies that the two games can not be distinguished as the change induced by the gamehop will never be noted; simply because the additional check will never succeed.

We can thus conclude for now that with probability $(1 - \Pr[C])$ the gamehop is perfectly indistinguishable and focus on the case where the adversary reported a dummy-friend message that is not a valid cipher. Note that in this case we can use the same argument as in the last game hop, as any distinguisher would fail if $\neg E$ where E is the event that $\mathcal{D}$ breaks the collision resistance, and $\Pr[E]$ is negligible due to the collision resistance of the random oracle.

H_3 is as H_2 only that the circuit additionally aborts if (1) the first message of the dummy friend in the *challenge transcript* is an **invalid** ciphertext, and (2) the first message of the dummy friend in the *input transcript* is an **invalid** ciphertext, (3) the first message of the dummy friend in the *challenge transcript* differs from the first message of the dummy friend in the *input transcript*, and (4) the Dec^* algorithm outputs the same verification key for the challenge- and input-transcript of the dummy friend. This then means that both decryptions are internally replaced by the PRF. Thus for distinguishing (which only works by causing an abort in H_3 where H_2 would have continued with the execution as the remaining part of the two games do not differ) the distinguisher would have to find a different message—which is a valid pre-image for the random oracle—that points to an output that shares the same second half as the value reported in the challenge transcript, which violates the collision resistance of the random oracle. Hence the two games can not be distinguished.

H_4 is as H_3 but the circuit aborts only if the first-round message of the sending party P_{bc} in the input transcript encrypts the same verification key as that from the challenge transcript and the decryption of the new first-round message yields a valid message—that is, the message is a valid cipher and the random oracle is not used for this message.

We want to show that for all PPT distinguishes $\mathcal{D}$ the following statement is true:

$$|\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_4} = 1] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_3} = 1]| \in \mathsf{negl}(\kappa)$$

We first define an intermediate event $E_{\mathcal{D}}$ as the event that $\mathcal{D}$ can *rerandomize* parts of a ciphertext. This leads to the following observation:

$$\begin{aligned} Adv_{\mathcal{D}} :=& |\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_4} = 1] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_3}] = 1| \\ =& E \wedge |\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_4} = 1] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_3}] = 1| \vee \\ & \neg E \wedge |\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_4} = 1] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_3} = 1]| \\ =& E \wedge |\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_4} = 1] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_3} = 1]| + \\ & \neg E \wedge |\Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_4} = 1] - \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_3} = 1]| \end{aligned} \tag{37}$$

Now note that $\neg E$ implies an equivalent behavior in both games, hence:

$$\neg E \wedge \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_4} = 1] = \neg E \wedge \Pr[\mathsf{out}_{\mathcal{D},\mathrm{H}_3} = 1]$$

Note next that E is negligible due to the IND-CCA-property of the encryption system PKE; if $E \notin \mathsf{negl}$ the following attack on the IND-CCA security would be possible: (1) The adversary creates two random messages $x_0 = (\cdot, \mathsf{vk}_0)$ and $x_1 = (\cdot, \mathsf{vk}_1)$, where the first half is not important but the second half differs, and sends them as challenges. (2) The challenger returns a ciphertext y. (3) The adversary rerandomizes y to y' such that the second half remains the same and sends y' to the decryption oracle. (4) If the second half of the result is vk_0 then $\mathcal{A}$ returns $\beta = 0$ and if it is vk_1 it returns $\beta = 1$. Otherwise it returns a uniformly random bit. If E can occur with non-negligible probability then $\mathcal{A}$ would violate the IND-CCA security (which is implied by the IND$-CCA security of PKE), thus we stress that $\Pr[E]$ is at most negligible.
Our claim follows.

H_5 is as H_4 only that the circuit additionally aborts if (1) the first message of the dummy friend in the *challenge transcript* is a **valid** ciphertext, and (2) the first message of the dummy friend in the *input transcript* is a **valid** ciphertext, (3) the first message of the dummy friend in the *challenge transcript* differs from the first message of the dummy friend in the *input transcript*, and (4) the Dec^* algorithm outputs the same verification key for the challenge- and input-transcript.
In this case we would have a distinguisher $\mathcal{D}$ that can effectively *partially re-randomize* transcripts. This distinguisher could be used in order to break the IND-CCA property of PKE as was the case in H_4.

H_6 is as H_5 only that the circuit additionally aborts if (1) the first message of the dummy friend in the *challenge transcript* is an **invalid** ciphertext, and (2) the first message of the dummy friend in the *input transcript* is a **valid** ciphertext, and (3) the first message of the dummy friend in the *challenge transcript* differs from the first message of the dummy friend in the *input transcript*, and (4) the Dec^* algorithm outputs the same verification key for the challenge- and input-transcript.
Essentially, to distinguish the distinguisher $\mathcal{D}$ has to create a ciphertext that encodes the same verification key that is in the image domain of the random oracle. Since this is truly random *and* the distinguisher can not query the oracle it follows that this is possible with negligible probability only which proves our claim.

H_7 This game-hop is purely cosmetic and hence does not change the distribution.

H_8 is as H_7 but the circuit aborts only if the first-round message of the sending party P_{bc} in the input transcript encrypts the same symmetric key as that from the challenge transcript and the decryption of the new first-round message is from the random oracle—that is, the decryption of that message using the first-round message of the input transcript using the actual decryption algorithm yields the error symbol $\perp$.
This is similar to the game hop from H_1 and hence indistinguishable.

H_9 is as H_8 only that the circuit additionally aborts if (1) the first message of the dummy friend in the *challenge transcript* is a **valid** ciphertext, and (2) the first message of the dummy friend in the *input transcript* is an **invalid** ciphertext, and (3) the first message of the dummy friend in the *challenge transcript* differs from the first message of the dummy friend in the *input transcript*, and (4) the Dec^* algorithm outputs the same symmetric key for the challenge- and input-transcript.
This is the same situation as the game hop between H_2 and H_1 and thus indistinguishable.

H_{10} is as H_9 only that the circuit additionally aborts if (1) the first message of the dummy friend in the *challenge transcript* is an **invalid** ciphertext, and (2) the first message of the dummy friend in the *input transcript* is an **invalid** ciphertext, (3) the first message of the dummy friend in the *challenge transcript* differs from the first message of the dummy friend in the *input transcript*, and (4) the Dec^* algorithm outputs the same symmetric key for the challenge- and input-transcript of the dummy friend. This then means that both decryptions are internally replaced by the PRF. Showing indistinguishability here is similar to that of H_3 and H_2.

H_{11} is as H_{10} but the circuit aborts only if the first-round message of the sending party P_{b_C} in the input transcript encrypts the same verification key as that from the challenge transcript and the decryption of the new first-round message yields a valid message—that is, the message is a valid cipher and the random oracle is not used for this message.
Indistinguishability follows from the indistinguishability between H_3 and H_4 as the proof is almost identical.

H_{12} is as H_{11} only that the circuit additionally aborts if (1) the first message of the dummy friend in the *challenge transcript* is a **valid** ciphertext, and (2) the first message of the dummy friend in the *input transcript* is a **valid** ciphertext, (3) the first message of the dummy friend in the *challenge transcript* differs from the first message of the dummy friend in the *input transcript*, and (4) the Dec^* algorithm outputs the same symmetric key for the challenge- and input-transcript.
Again the proof is analogous to the indistinguishability of H_{11} and H_{12} and we do not write it up specifically.

H_{13} is as H_{12} only that the circuit additionally aborts if (1) the first message of the dummy friend in the *challenge transcript* is an **invalid** ciphertext, and (2) the first message of the dummy friend in the *input transcript* is a **valid** ciphertext, and (3) the first message of the dummy friend in the *challenge transcript* differs from the first message of the dummy friend in the *input transcript*, and (4) the Dec^* algorithm outputs the same symmetric key for the challenge- and input-transcript.
The non-existence of a successful PPT distinguisher $\mathcal{D}$ which is better than the naive distinguisher that randomly guesses a bit follows from the non-existence of a distinguisher for the hybrid game hop from H_5 to H_6.

H_{14} This game-hop is purely cosmetic and hence does not change the distribution. We only include it for consistency with $\mathrm{Game}_3^\sigma(\kappa)$.

□

D.6 Proof of Lemma 11

Lemma 11 (Indistinguishability of $\mathrm{Game}_3^\sigma(\kappa)$ and $\mathrm{Game}_4^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme, let SKE be a tightly secure multi-challenge IND$-CCA secure symmetric encryption scheme, let SIG be an sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, and let F be a secure PRF. Then, for all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_3^\sigma(\kappa)$ and $\mathrm{Game}_4^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_3^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_4^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Indistinguishability easily follows from the perfect correctness of the encryption scheme PKE. If there was any message $x_{\mathsf{R}}^{(0)}$ created by the adversary as $\mathrm{PKE}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$ which does *not* decrypt with sk_P to $(OTP, \mathsf{vk}_{\mathsf{R}})$ this would mean that $\mathrm{PKE}.\mathsf{Dec}(\mathsf{sk}_P, \mathrm{PKE}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))) \neq (OTP, \mathsf{vk}_{\mathsf{R}})$. As this can not happen by requirement indistinguishability follows. □

D.7 Proof of Lemma 12

Lemma 12 (Indistinguishability of $\mathrm{Game}_4^\sigma(\kappa)$ and $\mathrm{Game}_5^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme, let SKE be a tightly secure multi-challenge IND$-CCA secure symmetric encryption scheme, let SIG be an sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, and let F be a secure PRF. Then, for all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_4^\sigma(\kappa)$ and $\mathrm{Game}_5^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_4^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_5^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. We reduce a distinguisher $\mathcal{D}$ who differentiates between $\text{Game}_4^{\sigma}(\kappa)$ and $\text{Game}_5^{\sigma}(\kappa)$ to an adversary $\mathcal{A}$ breaking the sEUF-CMA security of the used signature scheme Sig.

Creating the Transcript. The creation of the transcript works similar in both games, hence the transcript can be created by letting the adversary play all three parties roles according to $\text{Game}_2^{\sigma}(\kappa)$ for a randomly chosen sender $\mathsf{P}_{b_\mathcal{C}}$ and a randomly transmitted bit $\sigma_\mathcal{C}$. The only change is with respect to signatures. While the verification key from the challengers signature oracle is known to the adversary and hence can be embedded into the senders first message all signatures of the form $\mu \leftarrow \text{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$ created in line 11 of the protocol are replaced by queries $(x_0^{(\chi-1)}, x_1^{(\chi-1)})$ to the signing oracle.

The complete transcript $\pi_\mathcal{C}$ is then sent to the distinguisher $\mathcal{D}$.

Simulating the Oracle. When $\mathcal{D}$ sends some transcript π for evaluation to the oracle the adversary $\mathcal{A}$ behaves according to Item $\text{Game}_5^{\sigma}(\kappa)$: If the challenge transcript corresponds to the input transcript then it returns the correct bit, and if any of the first-round messages differ it simulates the circuit entirely:

If both parties input different messages than in the original transcript then we know due to the check in line 3 of the circuit code that both encrypt a different message and hence a use a new verification key (and also a symmetric key) for the remaining rounds. By knowing the secret key sk_P the adversary can thus extract both and simulate accordingly.

The same reasoning is true if only the sending parties message is replaced as then still both parties messages are independent from the challenge.

Special care thus only has to be taken into the simulation for transcripts where the first sending party message matches.

If the dummy friends message differs then the circuit also executes the same circuit, which the adversary can simulate as it has all the required information; the only thing that depends on the challenge oracle are the signatures and being in possession of the verification key the simulator can efficiently verify a given signature.

If the dummy friends message also matches then the circuit acts by counting rounds until the messages differ for the first time. Let $\mathcal{D}$ be a distinguisher that uses this behavior to distinguish. In $\text{Game}_5^{\sigma}(\kappa)$ the round χ^* is fixed due to lines $7-11$ as the first round where any of the messages differs from the challenge transcript, where potentially 1 is added in case the sending parties message for the next round is different. Thus for a given transcript π we can fix p for $\text{Game}_5^{\sigma}(\kappa)$; in order to distinguish this value p' must differ in $\text{Game}_3^{\sigma}(\kappa)$. (i) $p' < p$, meaning that the circuit outputs the correct bit with a lower probability when the message is handled by the original code of the circuit. This, however, would contradict the correctness of the scheme. (ii) $p' > p$, meaning that the probability of output $\sigma_\mathcal{C}$ is *larger* when the transcript is handled by the actual code than when it is handled by our modification. By requirement we know that up until round χ^* all messages are takes directly from the challenge transcript, and that $\chi^* < c$ (as otherwise the case would have been handled by the code in line 1). We can now differentiate between the two possible replacements n round (χ^*+1): (i.1) If the *sending* parties message has been replaced in this round then χ^* stays the same and $p \coloneqq 1/2 + \chi^*/2c$. In order to get a larger p' the adversary would have to create a *replacement* message which encrypts the same bit and a different signature on the same round-(χ^*-1)-messages. Knowing the secret key used for the transcript the reduction adversary can extract this signature and send it as valid signature on $(x_0^{(\chi^*-1)}, x_1^{(\chi^*-1)})$ to the challenger. Since the content of the message differs but the bit has to be the same (as otherwise the message would be rejected in $\text{Game}_3^{\sigma}(\kappa)$) it follows that the distinguisher created a new signature that was never returned from the challenge-oracle. (ii.2) If the *dummy friend* parties message has been replaced in this round then χ^* is increased by 1 and in order to get $p' > p$ the distinguisher has to create a message for round (χ^*+2) that contains a valid signature for this new message. Since by requirement the dummy friends message has been replaced in round (χ^*+1) this too corresponds to a new message and hence is a valid forgery for the challenger.

As $\mathcal{A}$ can only have a negligible chance to create a forgery due to the sEUF-CMA security of the signature scheme Sig it follows that $\alpha \in \mathsf{negl}(\kappa)$. □

D.8 Proof of Lemma 13

Lemma 13 (Indistinguishability of $\text{Game}_5^\sigma(\kappa)$ and $\text{Game}_6^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme, let SKE be a tightly secure multi-challenge IND$-CCA secure symmetric encryption scheme, let SIG be an sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, and let F be a secure PRF. Then, for all PPT guessing algorithms A, the distinguishing advantage for $\text{Game}_5^\sigma(\kappa)$ and $\text{Game}_6^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_5^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_6^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Let $\mathcal{D}$ be an efficient distinguisher that can predict if it is in $\text{Game}_5^\sigma(\kappa)$ or in $\text{Game}_6^\sigma(\kappa)$ with probability $1/2 + \alpha$. Out of $\mathcal{D}$ we can construct an adversary $\mathcal{A}$ on the IND$-CCA property of PKE.

The reduction algorithm $\mathcal{A}$ starts by creating the key-pair for the signature scheme and the key for the symmetric scheme honestly and follows $\text{Game}_5^\sigma(\kappa)$ the creation of the transcript, only that instead of encrypting the first message using the public-key scheme PKE directly, $\mathcal{A}$ forwards the messages $(OTP, \mathsf{vk}_\mathsf{R})$ and $(\mathsf{sk}_b, \mathsf{vk}_b)$ as challenge to the challenger $\mathcal{C}$ of the IND$-CCA game.

For further evaluation of the oracle the reduction adversary uses the *decryption* oracle from the challenger $\mathcal{C}$ for any first-round message that is not from the reported transcript π and continues simulation using the extracted secret key of that respective party; the same first-round message is covered automatically since $\text{Game}_5^\sigma(\kappa)$ and the same receiver message in $\text{Game}_4^\sigma(\kappa)$.

The reported transcript contains exactly the same views that are required by the two games: if the output is a proper encryption then the view is equivalent to $\text{Game}_5^\sigma(\kappa)$, and if the output of the oracle is uncorrelated randomness, the view corresponds to $\text{Game}_6^\sigma(\kappa)$.

Thus, if $\mathcal{D}$ can differentiate the two games with probability $1/2 + \alpha$, then $\mathcal{A}$ can differentiate the oracles with the same probability. The IND$-CCA requirement for PKE thus implies $\alpha \in \mathsf{negl}(\kappa)$. □

D.9 Proof of Lemma 14

Lemma 14 (Indistinguishability of $\text{Game}_6^\sigma(\kappa)$ and $\text{Game}_7^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme, let SKE be a tightly secure multi-challenge IND$-CCA secure symmetric encryption scheme, let SIG be an sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, and let F be a secure PRF. Then, for all PPT guessing algorithms A, the distinguishing advantage for $\text{Game}_6^\sigma(\kappa)$ and $\text{Game}_7^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_6^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_7^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. We reduce a distinguisher $\mathcal{D}$ that distinguishes $\text{Game}_6^\sigma(\kappa)$ from $\text{Game}_7^\sigma(\kappa)$ to an adversary $\mathcal{A}$ on the IND$-CPA property of the symmetric encryption scheme SKE. Again, we adapt the LR-view of Rogaway [Rog04] to account for the multiple challenges required, in that the game is played via oracle access to an oracle which either outputs valid encryptions of the input, or which outputs uncorrelated randomness.

We now describe how the adversary $\mathcal{A}$ behaves in order to embed the challenge of the IND$-CCA challenger $\mathcal{C}$ into a challenge for the distinguisher $\mathcal{D}$.

The adversary starts by following the protocol according to $\text{Game}_6^\sigma(\kappa)$ and creating the messages, with the one exception that any call to $\text{SKE}.\mathsf{Enc}(\mathsf{sk}_b, \sigma\|\mu)$ is replaced by an *oracle* call to the challenge oracle with input $(\sigma\|\mu)$. This causes the transcript to either only have truly random messages (in which case the view corresponds to $\text{Game}_7^\sigma(\kappa)$) or actual encryptions under the challengers secret key sk (which results in a valid transcript for $\text{Game}_6^\sigma(\kappa)$).

Thus the adversary inherits the advantage α as long as $\mathcal{A}$ can simulate the decryption oracle accordingly. Fortunately this is the case here. Further, as we assume a lazy evaluation of the or[15] the decryption oracle is only called on the first round message which differs from the challenge transcript and, hence, the decryption oracle is never called on the challenge ciphertext. The leaked public key pk_P is controlled by the reduction adversary and hence any new transcript that contains

[15] This means that first the simulator checks for equal messages and if that is already true, the second condition is ignored, and hence the sending parties message is only forwarded to the decryption oracle if it differs from the challenge.

a different first-round message than the challenge transcript can be simulated by first decrypting the secret key (which is different from the one used by the challenger with overwhelming probability) and then decrypting each round individually. □

D.10 Graphical Depiction of the Secrecy Proof Outline

$\mathrm{Game}_1(\kappa)$

$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$(OTP, \mathsf{vk}_\mathsf{R}) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_\mathsf{R}^{(0)})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_\mathsf{R}, \pi)$ **then**
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{(\chi)} \coloneqq X_b[1\colon |X_b|]$
    if $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_\mathsf{R}, \mathsf{vk}_\mathsf{R}) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$x_\mathsf{R}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_\mathsf{R}))$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$(\mathsf{sk}_b) \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^\kappa)$
$(\mu_b^{(0)}) \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x_\mathsf{R}^{(0)})$
$x_b^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (0, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $\mu^{(\chi)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, 0, \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_\mathsf{R}, \pi)$
return $\mathcal{S}_\mathcal{A}^{P'^{AT}}(\pi, T_\mathsf{R})$

$\mathrm{Game}_2(\kappa)$

if $\pi = \pi_\mathcal{C}$ **then**
  return $OTP_\mathcal{C} \oplus 0$
elseif $\textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, \pi[0]) = \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, \pi_\mathcal{C}[0]) \wedge$
    $\pi[0] \neq \pi_\mathcal{C}[0]$ **then**
  abort
elseif $\pi[0] = \pi_\mathcal{C}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_\mathcal{C}[\chi'])$
  if $\pi[x_{b_\mathcal{C}}^{c^*+1}] = \pi_\mathcal{C}[x_{b_\mathcal{C}}^{\chi^*+1}] \vee$
    $\textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_{b_\mathcal{C}}, x_{b_\mathcal{C}}^{(\chi^*+1)}) = (\sigma_\mathcal{C} \| \mu_\mathcal{C}{}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_\mathcal{C} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(0,1)$
if $\pi[x_\mathsf{R}^{(0)}] = \pi_\mathcal{C}[x_\mathsf{R}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_\mathsf{R}) \coloneqq (OTP_\mathcal{C}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_\mathsf{R}, \pi)$ **then**
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{(\chi)} \coloneqq X_b[1\colon |X_b|]$
    if $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_\mathsf{R}, \mathsf{vk}_\mathsf{R}) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$x_\mathsf{R}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_b}$:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_\mathsf{R}, \pi)$
return $\mathcal{S}_\mathcal{A}^{P'^{AT}}(\pi, T_\mathsf{R})$

$\mathrm{Game}_2(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus 0$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{c^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(0,1)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{(\chi)} \coloneqq X_b[1\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \sigma_b \neq \sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1 - \sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
–
–
–
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  –
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'^{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus 0$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{c^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(0,1)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}] \wedge \pi \neq \pi_{\mathcal{C}}$ **then**
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{(\chi)} \coloneqq X_b[1\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \sigma_b \neq \sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1 - \sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
–
–
–
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  –
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'^{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ then
  return $OTP_{\mathcal{C}} \oplus 0$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ then
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ then
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{c^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ then
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(0,1)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}] \wedge \pi \neq \pi_{\mathcal{C}}$ then
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ then
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ do
  foreach $b \in \{0,1\} | \chi_b > c$ do
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{(\chi)} \coloneqq X_b[1\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \sigma_b \neq \sigma'_b$ then
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1 - \sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P}_{\mathbf{b}}$:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathbf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ do
  $\mathsf{P}_{\mathbf{b}}$:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathbf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'^{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ then
  return $OTP_{\mathcal{C}} \oplus 1$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ then
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ then
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{c^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ then
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(1,0)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}] \wedge \pi \neq \pi_{\mathcal{C}}$ then
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ then
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ do
  foreach $b \in \{0,1\} | \chi_b > c$ do
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{(\chi)} \coloneqq X_b[1\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \sigma_b \neq \sigma'_b$ then
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1 - \sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P}_{\mathbf{b}}$:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathbf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ do
  $\mathsf{P}_{\mathbf{b}}$:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathbf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'^{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ then
  return $OTP_{\mathcal{C}} \oplus 1$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ then
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ then
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) = (\sigma_{\mathcal{C}} \| \mu_{\mathcal{C}}{}^{(\chi^*+1)})$ then
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(1,0)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}] \wedge \pi \neq \pi_{\mathcal{C}}$ then
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ then
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ do
  foreach $b \in \{0,1\} | \chi_b > c$ do
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{(\chi)} \coloneqq X_b[1\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ then
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ do
  P_{b}:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'^{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_5(\kappa)$

$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ then
  return $\mathsf{CointossS}^{(\pi)}_{(1/2)}(0,1)$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ do
  foreach $b \in \{0,1\} | \chi_b > c$ do
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\sigma'_b \coloneqq X_b[0]$
    $\mu_b^{(\chi)} \coloneqq X_b[1\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \sigma_b \neq \sigma'_b$ then
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2+\chi_b/2c)}(\sigma^{(b')}, 1-\sigma^{(b')})$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
P_{b}:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^\kappa)$
$(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (1, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ do
  P_{b}:
  $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, (1, \mu^{(\chi)}))$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'^{AT}}(\pi, T_{\mathsf{R}})$

D.11 Full Proof of Secrecy

Here we provide a full proof of the following theorem:

Theorem 6 (Secrecy). *Let Π_{PKE} be an IND$-CCA secure asymmetric encryption scheme, let Π_{SKE} be an IND$-CCA secure symmetric encryption scheme, let Π_{SIG} be a sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, and let F be a secure PRF. Then, Π^1_{AT} satisfies ς-secrecy with $\varsigma \in \mathsf{owhl}(\kappa)$.*

Essentially the proof shows that no PPT distinguisher that gets the transcript but not the receivers transcript can exist that can distinguish between the case where we always transfer a 0 and where we always transfer the 1 better than by guessing. This follows from the fact that the One-Time-Pad chosen uniformly at random by the receiver masks the output bit. As this masking bit is never revealed and only sent in an encrypted form with the key of the circuit our claim follows.

$\mathrm{Game}^{\sigma}_1(\kappa)$ This is the original game (after replacing the obfuscated circuit with oracle access to the circuit, the PRF with an actual random oracle and the adversary with the simulator), where the sending party P_b is chosen uniformly at random but the sending party always sends $\sigma = 0$.

$\mathrm{Game}^{\sigma}_2(\kappa)$ This game follows $\mathrm{Game}^{\sigma}_1(\kappa)$, but with the following changes:

- During the simulation of the oracle P_{AT} from Fig. 5 the simulation enforces correctness of the challenge transcript $\pi_{\mathcal{C}}$: if the input transcript π matches the challenge transcript $\pi_{\mathcal{C}}$, it returns $OTP_{\mathcal{C}} \oplus \sigma_{\mathcal{C}}$.
- During simulation of the circuit the adversary aborts if any of the first-round messages from P_0 or P_1 differ from the messages reported in the challenge transcript *and* the decryptions still match.
- During simulation of the circuit, if the first receiver message of the input transcript π is the same as that of the challenge transcript $\pi_{\mathcal{C}}$, instead of decrypting it the circuit directly sets $OTP = OTP_{\mathcal{C}}$ and $\mathsf{vk}_{\mathsf{R}} = \mathsf{vk}_{\mathcal{C}\mathsf{R}}$ as the values used in the creation of the challenge transcript.
- During simulation of the oracle, if the first-round messages of both parties match the first-round messages in the challenge transcript $\pi_{\mathcal{C}}$. In this case, the program compares the input transcript π with the challenge transcript $\pi_{\mathcal{C}}$ until it finds the first round χ^* in which the input differs from the challenge transcript. It then checks round $\chi^* + 1$, and if it contains the same message from the sending party, it adds one to χ.
  Finally, the circuit flips a biased coin, which returns the correct bit $\sigma_{\mathcal{C}}$ with probability $p := 1/2 + \chi^*/2c$ and the complementary bit $(1 - \sigma_{\mathcal{C}})$ otherwise.

Lemma 15. *Let Π_{PKE} be an IND$-CCA secure public-key encryption scheme. Let Π_{SKE} be an IND$-CCA secure secret-key encryption scheme. Let Π_{SIG} be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}^{\sigma}_1(\kappa)$ and $\mathrm{Game}^{\sigma}_2(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}^{\sigma}_1(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}^{\sigma}_2(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Follows from Lemmas 9 to 14. □

$\mathrm{Game}^{\sigma}_3(\kappa)$ This game follows $\mathrm{Game}^{\sigma}_2(\kappa)$ but the oracle is simulated slightly different: If the first receiver message is the same as the one reported in the challenge transcript and the input transcript is *not* the challenge transcript, then the circuit reports a uniformly random bit.
Note that this does not work in the anonymity proof as there we assume that the adversary is given access to the receivers random tape, and hence can create their own new signature on the modified transcript.

Lemma 16. *Let Π_{PKE} be an IND$-CCA secure public-key encryption scheme. Let Π_{SKE} be an IND$-CCA secure secret-key encryption scheme. Let Π_{SIG} be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}^{\sigma}_3(\kappa)$ and $\mathrm{Game}^{\sigma}_4(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}^{\sigma}_3(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}^{\sigma}_4(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. We reduce a distinguisher $\mathcal{D}$ between these two games to an adversary $\mathcal{A}$ on the EUF-CMA security of the signature scheme Π_{SIG}.

Creating the Transcript. The creation of the transcript is straightforward and works by sampling random messages for each party.

Simulating the Oracle. The used verification key vk_R is set to be the verification key of the challenger. Thus the final signature μ on the entire transcript, which is required for the circuit to not *abort* (by outputting a random bit), needs to be forged. Hence for each input transcript the adversary first checks if the receiver message is equivalent to that from the challenge transcript. If it isn't then the transcript can not be used for distinguishing anyways. If it is, the adversary checks if the remaining transcript is the same as well. If it is, the transcript can not be used for distinguishing and simulation just continues as the path taken is equivalent in both games. If it isn't then the adversary checks if the signature μ verifies under the used key. If it doesn't then both games act exactly the same and output a random bit. Hence to distinguish the signature has to verify. Then, however, the signature is a valid forgery.

Translating the Result. If no valid signature was queried then—as mentioned above—the distinguishing advantage must be negligible. For a non-negligible advantage the distinguisher has to query at least one signature. This can be used as forgery to break the EUF-CMA security of SIG.
As stated above, the advantage of the distinguisher is directly related to the probability of successfully distinguishing. Hence $\mathcal{A}$ will have a valid forgery with non-negligible advantage.
This would contradict the EUF-CMA security of the signature scheme and thus completes our proof. □

$\mathrm{Game}_4^\sigma(\kappa)$ This game is as $\mathrm{Game}_3^\sigma(\kappa)$ but instead of fixing $\sigma := 0$, we now act as if the sending party sends $\sigma := 1$.

Lemma 17. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SKE be an IND$-CCA secure secret-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_3^\sigma(\kappa)$ *and* $\mathrm{Game}_4^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_3^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_4^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Security automatically follows from the statistical security of the One-Time-Pad encryption: if there was a distinguisher $\mathcal{D}$ with non-negligible advantage α in distinguishing these two games then there would be an adversary $\mathcal{A}$ which can decrypt One-Time-Pad encrypted bits with non-negligible advantage.

Simulating the Transcript. As the transcript is the same in both games, we construct the transcript by uniformly sampling each message.

Simulating the Oracle. If $\mathcal{A}$ is given a ciphertext ct (which by definition is defined as $OTP \oplus \sigma$ for some random OTP and σ) we simulate the oracle as described in both games (since they are equivalent) and whenever the code says **return** $\mathsf{CointossS}_{(p)}^{(\pi)}(OTP_\mathcal{C} \oplus \sigma, OTP_\mathcal{C} \oplus \overline{\sigma})$, we replace $OTP_\mathcal{C} \oplus \sigma$ with x and $OTP_\mathcal{C} \oplus \overline{\sigma}$ with $\overline{x}$.

Translating the Result. If the distinguisher assumes $\mathrm{Game}_3^\sigma(\kappa)$ then $\mathcal{A}$ assumes $\sigma = 0$ (in which case the transferred bit is $\sigma_\mathcal{C} := x$), and if the distinguisher assumes $\mathrm{Game}_4^\sigma(\kappa)$ then $\mathcal{A}$ guesses $\sigma = 1$ (and the transferred bit is $\sigma_\mathcal{C} := \overline{x}$).
Note that if $\mathcal{D}$ is correct with advantage α over guessing, hence the guess of $\mathcal{A}$ is correct with advantage α as well. □

$\mathrm{Game}_5^\sigma(\kappa)$ This game follows $\mathrm{Game}_4^\sigma(\kappa)$ but undoes all the changes from the first-to-second gamehop and that from $\mathrm{Game}_3^\sigma(\kappa)$.

Lemma 18. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SKE be an IND$-CCA secure secret-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_4^\sigma(\kappa)$ *and* $\mathrm{Game}_5^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_4^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_5^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Two-round protocol Π^1_{AT} in the designated sender setting.

We assume three parties are present in this protocol, two of which are participating. Let P_0 and P_1 be the potential sender, one of which is not participating, and let R denote the receiver.
Let $\textsc{Pke} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a *public-key* IND$-CCA-secure encryption scheme with Pseudorandom Ciphertexts.
Let $\textsc{Sig}$ be an sEUF-CMA-secure signature scheme.
Let P_{AT} be an obfuscated program from Fig. 10, which hides a verification key vk and a secret key sk_P, the corresponding public key pk_P is leaked.
The protocol Π^1_{AT} works as follows:

On input $(\bot)$, R samples a uniformly random bit $OTP \xleftarrow{\$} \{0,1\}$ and broadcasts $x^{(0)}_{\mathsf{R}} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$.
On input (b, σ, k), P_b computes $\mu^{(0)}_b \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x^{(0)}_{\mathsf{R}}))$ and broadcasts $x^{(0)}_b \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$.
On input $(\bot)$, P_{1-b} sets uniformly random $x^{(0)}_{1-b}$.
P_b computes $\mu^{(1)}_b \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$ and broadcasts $x^{(1)}_b \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu^{(1)}_b)$.
P_{1-b} samples uniformly random $x^{(1)}_{1-b}$.
R outputs $OTP \oplus P_{AT}\left(x^{(0)}_0, x^{(0)}_1, x^{(0)}_{\mathsf{R}}, x^{(1)}_0, x^{(1)}_1\right)$.

Fig. 9: The designated sender protocol for Anonymous Transfer. We generally assume that once all messages of one round have been determined, parties publish them.

Proof. Follows from Lemmas 9 to 14 and 16. □

Thus our theorem follows:

Theorem 6 (Secrecy). *Let $\textsc{Pke}$ be an IND$-CCA secure asymmetric encryption scheme, let $\textsc{Ske}$ be an IND$-CCA secure symmetric encryption scheme, let $\textsc{Sig}$ be a sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, and let F be a secure PRF. Then, Π^1_{AT} satisfies ς-secrecy with $\varsigma \in \mathsf{owhl}(\kappa)$.*

E Asymptotically Secure AT in the Designated Sender Model

E.1 A two-round protocol

In this section, we introduce a candidate protocol for Anonymous Transfer in the *designated sender* setting. The sender is given secret information which is in some known relation with the Common Reference String. The protocol itself is displayed in Fig. 9. It requires a Common Reference String, which contains an ideally obfuscated circuit which behaves as described in Fig. 10.

The protocol consists only of two rounds. The ideally obfuscated circuit knows a verification key vk, to which the sending party is given the signing key k. For both messages, the sending party adds a signature *created with* k. In the first round, the sender only enters the bit σ that should be transferred and attaches a signature μ on σ with k. In the second round, the sender again enters the bit σ, but attaches a signature on the *first round*.

The circuit then checks for both second-round messages if either of them contains an encryption of a signature on the first round; if this is the case for at least one second-round message[16], it flips two biased coins, which return *tails* with probability $2d$. If the first coin lands *tails*, the program returns the bit encoded in the second message. Otherwise, the second coin is considered. If this lands on *tails*, then the bit encoded in the first round is returned. Otherwise, the program returns a random bit.

[16] In case both second-round messages contain a valid signature, always use the message of P_0.

$P_{AT}[\mathsf{pk}_P]\left(\left(x_{\mathsf{R}}^{(0)}, x_0^{(0)}, x_1^{(0)}, x_0^{(1)}, x_1^{(1)}\right)\right)$

$OTP \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}})$

$\left(\sigma_0^{(0)}, \mu_0^{(0)}\right) \coloneqq \textsc{Pke}.\mathsf{Dec}\left(\mathsf{sk}_P, x_0^{(0)}\right), \quad \left(\sigma_1^{(0)}, \mu_1^{(0)}\right) \coloneqq \textsc{Pke}.\mathsf{Dec}\left(\mathsf{sk}_P, x_1^{(0)}\right)$

$\left(\sigma_0^{(1)}, \mu_0^{(1)}\right) \coloneqq \textsc{Pke}.\mathsf{Dec}\left(\mathsf{sk}_P, x_0^{(1)}\right), \quad \left(\sigma_1^{(1)}, \mu_1^{(1)}\right) \coloneqq \textsc{Pke}.\mathsf{Dec}\left(\mathsf{sk}_P, x_1^{(1)}\right)$

if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then** :

  $b \coloneqq \mathbf{argmin}_{b'}(\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_{b'}^{(1)}, \pi[0]))$

  $coin \leftarrow \mathsf{Cointoss}_{(2d)}(x_b^{(1)})$

  if $coin = tails$ **then** :

    return $OTP \oplus \sigma_b^{(1)}$

  elseif $coin = heads$ **then** :

    $coin \leftarrow \mathsf{Cointoss}_{(2d)}(x_b^{(0)})$

    if $coin = tails$ **then** :

      return $OTP \oplus \sigma_b^{(0)}$

    elseif $coin = heads$ **then** :

      return $\mathsf{CointossS}_{(1/2)}^{(x_0^{(1)} \| x_1^{(1)})}(0,1)$

elseif $\exists b$: $\mathsf{Sig.Vfy}(\mathsf{vk}, \mu_b^{(0)}, \sigma_b^{(1)})$ **then** :

  // No valid round-2 message whatsoever, but round 1 is okay

  $b \coloneqq \mathbf{argmin}_{b'}(\mathsf{Sig.Vfy}(\mathsf{vk}, \mu_{b'}^{(0)}, \sigma_{b'}^{(1)}))$

  $coin \leftarrow \mathsf{Cointoss}_{(2d)}(x_b^{(0)})$

  if $coin = tails$ **then** :

    return $OTP \oplus \sigma_b^{(0)}$

  elseif $coin = heads$ **then** :

    return $\mathsf{CointossS}_{(1/2)}^{(x_0^{(1)} \| x_1^{(1)})}(0,1)$

else :

  return $\mathsf{CointossS}_{(1/2)}^{(x_0^{(1)} \| x_1^{(1)})}(0,1)$

Fig. 10: Obfuscated program P_{AT} for the designated sender setting.

If there is no second-round message with a valid encryption of the first round, the program continues by considering the first round messages. If there is a message that decrypts to a bit σ and a valid signature μ on σ with respect to vk[17], then the program flips a single coin which lands on *tails* with probability $2d$. In this case, the bit is returned. Otherwise, the circuit again returns a uniformly random bit.

It is crucial that the randomness inside the ideally obfuscated circuit is derived from designated parts of the circuit input only, the source of which is explicitly given inside the figures. We note that this is in contrast to the established way of obfuscating probabilistic circuits due to [CLT+15]. Recall that in [CLT+15], the obfuscated circuit always derives its randomness from the *entire input*. In our case, the randomness of, e.g., the first coin-toss is taken from the second sender message *only*, whereas the randomness of the second coin-toss is taken from the first sender message. This ensures a *one-shot* behavior, in that attacks which involve changing only a single message result in a circuit which behaves *deterministic*, regardless of the new value of the changed message. Thereby, this prevents attacks using repeated sampling to estimate output probabilities.

[17] Again, in case both first-round messages contain a valid signature, always use the message of P_0.

Security analysis.

Correctness. We first investigate the correctness of our protocol. That is, we prove the following lemma:

Lemma 19 (Correctness). *Let Π^1_{AT} be instantiated as described in Fig. 9. Then Π^1_{AT} is correct with $\varepsilon = 1 - \frac{(1-2d)^2}{2}$ according to Eq. (3).*

Hence, an honest protocol run guarantees that the bit σ will be received successfully with probability at least $2d(1-d) + 1/2 = 1 - \frac{(1-2d)^2}{2}$.

Proof. We assume as input an honest transcript. Without loss of generality we ignore the One-Time-Pad encoded by the receiver in this analysis by assuming that $OTP = 0$; the case for $OTP = 1$ is symmetrical only that the inverse bit is output by the circuit which is then flipped again by the receiver. An honest transcript is handled by P_{AT} as follows:

- The signature of the first round verifies on the sending parties input. With probability $(1 - \mathsf{negl}(\kappa))$, this is the only signature that verifies, meaning that with probability $(1 - \mathsf{negl}(\kappa)) + \frac{\mathsf{negl}(\kappa)}{2}$, this branch is taken. Now a biased coin is flipped:
  - The first coin yield *tails* with probability $2d$, in which case the correct output $\sigma =: \sigma_b^{(1)}$ encoded in the second-round message by P_b is returned.
  - With probability $(1 - 2d)$, the coin yields *heads*, in which case a second coin is flipped:
    - With probability $2d$, the coin yields *tails*, in which case the correct output $\sigma =: \sigma_b^{(0)}$ encoded in the first-round message by P_b is returned.
    - With probability $(1 - 2d)$, the coin lands on the *heads*-side. In this case, a new *unbiased* coin is flipped:
      - With probability $1/2$, this new coin returns σ.
      - With probability $1/2$, this new coin returns $(1 - \sigma)$.
- With negligible probability $\mathsf{negl}(\kappa)$, the signature of the first round verifies on the dummy friends input. In case of two signatures verifying, we deterministically choose the message from party P_0, which is correct with probability $1/2$; hence we enter this branch with probability $\frac{(\mathsf{negl}(\kappa))}{2}$. This too causes a coin flip.
  - The first coin yield *tails* with probability $2d$, in which case the encoded bit $\sigma^{(1)}_{1-b}$ encoded in the second-round message of the dummy friend P_{1-b} is returned. Since the message was random, this bit is uniformly distributed.
    - $\sigma_{1-b} = \sigma$ with probability $1/2$, meaning a correct output.
    - $\sigma_{1-b} \neq \sigma$ with probability $1/2$, causing a faulty output.
  - With probability $(1 - 2d)$, the coin yields *heads*, in which case a second coin is flipped:
    - With probability $2d$, the coin yields *tails*, in which case the first-round message of P_{1-b} is considered.
      - With probability $(1 - \mathsf{negl}(\kappa))$, this has the wrong form, that is, the attached message does not verify the encoded bit. In this case, the output bit is chosen at random:
        - With probability $1/2$, the output is σ.
        - With probability $1/2$, the output is $(1 - \sigma)$.
      - With probability $(\mathsf{negl}(\kappa))$, the first-round message contains a valid signature. This causes a new coin flip:
        - With probability $2d$, this coin lands on *tails*, causing the output to be the bit encoded in the first round. With this bit being uniformly distributed, we obtain:
        $\implies$ σ with probability $1/2$.
        $\implies$ $(1 - \sigma)$ with probability $1/2$.

        - With probability $(1 - 2d)$, the coin lands on *heads*. In this case, the output is taken at random, which implies we get
        $\implies$ σ with probability $1/2$.
        $\implies$ $(1 - \sigma)$ with probability $1/2$.

* With probability $(1-2d)$, the coin lands on the *heads*-side. In this case, a new *unbiased* coin is flipped:
  - With probability $1/2$, this new coin returns σ.
  - With probability $1/2$, this new coin returns $(1-\sigma)$.

So a wrong output is obtained with probability

$$\begin{aligned}&\left((1-\mathsf{negl}(\kappa))+\frac{\mathsf{negl}(\kappa)}{2}\right)\cdot\left((1-2d)\cdot\left((1-2d)\cdot\left(\frac{1}{2}\right)\right)\right)+\left(\frac{\mathsf{negl}(\kappa)}{2}\right)\cdot\frac{1}{2}\\ \approx&\frac{(1-2d)^2}{2}\end{aligned} \tag{38}$$

The approximation comes from setting $\mathsf{negl}(\kappa)=0$.

Thus, the correct output is returned with the complementary probability, which according to Eq. (3) yields:

$$\varepsilon = 1-\frac{(1-2d)^2}{2} \tag{39}$$

This concludes our proof. □

Anonymity. We now focus on the anonymity of Π^1_{AT} by proving the following lemma:

Lemma 20 (Anonymity (informal)). *Let Π^1_{AT} be instantiated as described in Fig. 9. Then Π^1_{AT} is anonymous with $\delta = 1-2d$ according to Eq. (4).*

Our proof looks as follows:

$\mathrm{Game}_1(\kappa)$	$\mathrm{Game}_2(\kappa)$
$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$	$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
–	**if** $\pi = \pi_{\mathcal{C}}$ **then**
–	**return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
–	–
–	–
–	–
–	–
–	–
–	–
–	–
–	–
–	–
–	–
–	–
–	–
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$	$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$	$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$	$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$	$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**	**if** $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$	**return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$	**return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**	**elseif** $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$	**return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi	**fi**
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$	**return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
R:	R:
$OTP \xleftarrow{\$} \{0,1\}$	$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$	$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:	$\mathsf{P_0}$:
$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$	$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$	$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:	$\mathsf{P_1}$:
–	–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$	$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:	$\mathsf{P_0}$:
$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$	$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$	$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:	$\mathsf{P_1}$:
–	–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$	$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$	**return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_2(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
–
–
–
–
–
–
–
–
–
–
–
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
–
–
–
–
–
–
–
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$

1 : $(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
3 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
4 : **elseif** $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
5 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : –
15 : –
16 : –
17 : –
18 : **fi**
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
4 : $\mathsf{P_0}$:
5 : $\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
6 : $x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
7 : $\mathsf{P_1}$:
8 : –
9 : $x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
10 : $\mathsf{P_0}$:
11 : $\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
12 : $x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
13 : $\mathsf{P_1}$:
14 : –
15 : $x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

1 : $(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
3 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
4 : **elseif** $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
5 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
6 : **elseif** $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
7 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : –
15 : –
16 : –
17 : –
18 : **fi**
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
4 : $\mathsf{P_0}$:
5 : $\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
6 : $x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
7 : $\mathsf{P_1}$:
8 : –
9 : $x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
10 : $\mathsf{P_0}$:
11 : $\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
12 : $x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
13 : $\mathsf{P_1}$:
14 : –
15 : $x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

1 : $(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
3 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
4 : **elseif** $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
5 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
6 : **elseif** $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
7 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : –
15 : –
16 : –
17 : –
18 : **fi**
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
4 : $\mathsf{P_0}$:
5 : $\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
6 : $x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
7 : $\mathsf{P_1}$:
8 : –
9 : $x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
10 : $\mathsf{P_0}$:
11 : $\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
12 : $x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
13 : $\mathsf{P_1}$:
14 : –
15 : $x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_5(\kappa)$

1 : $(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
3 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
4 : **elseif** $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
5 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
6 : **elseif** $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
7 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
8 : **elseif** $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
9 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
10 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
11 : –
12 : –
13 : –
14 : –
15 : –
16 : –
17 : –
18 : **fi**
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
4 : $\mathsf{P_0}$:
5 : $\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
6 : $x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
7 : $\mathsf{P_1}$:
8 : –
9 : $x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
10 : $\mathsf{P_0}$:
11 : $\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
12 : $x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
13 : $\mathsf{P_1}$:
14 : –
15 : $x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_5(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_6(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_6(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_7(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(1)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_1}$:
–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_7(\kappa)$

1 : $(OTP) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
3 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
4 : **elseif** $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
5 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
6 : **elseif** $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
7 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
8 : **elseif** $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
9 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
10 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
11 : **elseif** $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
12 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
13 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
14 : **elseif** $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
15 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(1)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
16 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
17 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
18 : **fi**
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, OTP)$
4 : $\mathsf{P_0}$:
5 : $\mu_0^{(0)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
6 : $x_0^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
7 : $\mathsf{P_1}$:
8 : $-$
9 : $x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
10 : $\mathsf{P_0}$:
11 : $\mu_0^{(1)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, \pi[0])$
12 : $x_0^{(1)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
13 : $\mathsf{P_1}$:
14 : $-$
15 : $x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_8(\kappa)$

1 : $(OTP) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
3 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
4 : **elseif** $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
5 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
6 : **elseif** $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
7 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
8 : **elseif** $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
9 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
10 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
11 : **elseif** $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
12 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
13 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
14 : **elseif** $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
15 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(1)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
16 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
17 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
18 : **fi**
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
4 : $\mathsf{P_0}$:
5 : $-$
6 : $x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
7 : $\mathsf{P_1}$:
8 : $-$
9 : $x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
10 : $\mathsf{P_0}$:
11 : $-$
12 : $x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
13 : $\mathsf{P_1}$:
14 : $-$
15 : $x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_8(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(1)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_9(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(1)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_9(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(1)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{10}(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(1)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_1^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_1^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{10}(\kappa)$

1 : $(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
3 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
4 : **elseif** $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
5 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
6 : **elseif** $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
7 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
8 : **elseif** $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
9 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
10 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
11 : **elseif** $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
12 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
13 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
14 : **elseif** $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
15 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(1)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
16 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(2d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
17 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
18 : **fi**
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
4 : $\mathsf{P_0}$:
5 : $-$
6 : $x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
7 : $\mathsf{P_1}$:
8 : $\mu_1^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
9 : $x_1^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
10 : $\mathsf{P_0}$:
11 : $-$
12 : $x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
13 : $\mathsf{P_1}$:
14 : $\mu_1^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
15 : $x_1^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{11}(\kappa)$

1 : $(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
3 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
4 : **elseif** $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
5 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
6 : **elseif** $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
7 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
8 : **elseif** $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
9 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
10 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
11 : **elseif** $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
12 : **return** $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
13 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
14 : –
15 : –
16 : –
17 : –
18 : **fi**
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
4 : $\mathsf{P_0}$:
5 : $-$
6 : $x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
7 : $\mathsf{P_1}$:
8 : $\mu_1^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
9 : $x_1^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
10 : $\mathsf{P_0}$:
11 : $-$
12 : $x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
13 : $\mathsf{P_1}$:
14 : $\mu_1^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
15 : $x_1^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{11}(\kappa)$

$(OTP) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(0)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_1^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(1)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, \pi[0])$
$x_1^{(1)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{12}(\kappa)$

$(OTP) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(0)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_1^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(1)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, \pi[0])$
$x_1^{(1)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{12}(\kappa)$

$(OTP) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_1^{(0)}])}_{(d)}(OTP \oplus \sigma_{\mathcal{C}}, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(0)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_1^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(1)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, \pi[0])$
$x_1^{(1)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{13}(\kappa)$

$(OTP) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(0)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_1^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(1)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, \pi[0])$
$x_1^{(1)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, \sigma \| \mu)$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{13}(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
–
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_1^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_1^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{14}(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
–
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_1^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_1^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{14}(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
–
–
–
–
–
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_1^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_1^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{15}(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
–
–
–
–
–
–
–
–
–
–
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
–
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
$x_1^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
$\mathsf{P_0}$:
–
$x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$\mu_1^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_1^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{15}(\kappa)$

1 : $(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
3 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(\sigma_{\mathcal{C}}, \overline{\sigma_{\mathcal{C}}})$
4 : –
5 : –
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : –
15 : –
16 : –
17 : –
18 : –
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :
1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
4 : $\mathsf{P_0}$:
5 : –
6 : $x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
7 : $\mathsf{P_1}$:
8 : $\mu_1^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
9 : $x_1^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
10 : $\mathsf{P_0}$:
11 : –
12 : $x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
13 : $\mathsf{P_1}$:
14 : $\mu_1^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
15 : $x_1^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_{16}(\kappa)$

1 : $(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
2 : –
3 : –
4 : –
5 : –
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : –
15 : –
16 : –
17 : –
18 : –
19 : $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
20 : $(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
21 : $(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
22 : $(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
23 : **if** $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
24 : **return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
25 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
26 : **elseif** $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
27 : **return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
28 : **fi**
29 : **return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
30 :
1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}$
3 : $x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
4 : $\mathsf{P_0}$:
5 : –
6 : $x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
7 : $\mathsf{P_1}$:
8 : $\mu_1^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (\sigma, x_{\mathsf{R}}^{(0)}))$
9 : $x_1^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
10 : $\mathsf{P_0}$:
11 : –
12 : $x_b^{(1)} \xleftarrow{\$} \{0,1\}^m$
13 : $\mathsf{P_1}$:
14 : $\mu_1^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
15 : $x_1^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$
16 : **return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

That is, we use the following games:

$\mathrm{Game}_1^\sigma(\kappa)$ This is the original game in which party P_0 is the sender, but where the obfuscated circuit has been replaced with oracle-access to the circuit and the PRF within the circuit is replaced by a random oracle.

$\mathrm{Game}_2^\sigma(\kappa)$ This game follows $\mathrm{Game}_1^\sigma(\kappa)$, but during the simulation of the oracle P_{AT} from Fig. 10, the simulation enforces correctness of the challenge transcript $\pi_{\mathcal{C}}$ by initially checking if the input matches the challenge transcript and in that case, returning the output of a biased coin that returns $OTP \oplus \sigma_{\mathcal{C}}$ with probability $(1-(1-2d)^2/2)$ and $OTP \oplus \overline{\sigma_{\mathcal{C}}}$ with the complementary probability $((1-2d)^2/2)$. The randomness is taken over the entire transcript.

Lemma 21 (Indistinguishability of $\mathrm{Game}_1^\sigma(\kappa)$ and $\mathrm{Game}_2^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_1^\sigma(\kappa)$ and $\mathrm{Game}_2^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_1^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_2^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 19. □

$\mathrm{Game}_3^\sigma(\kappa)$ This game follows $\mathrm{Game}_2^\sigma(\kappa)$, but during simulation of the oracle P_{AT} from Fig. 10, if the input has a different *first-round* message of the *sender* or of the *receiver*, the simulated program outputs a uniformly random bit. The randomness is taken over all three zero-round messages, that is, over $x_0^{(0)}$, $x_1^{(0)}$ and $x_{\mathsf{R}}^{(0)}$.

Lemma 22 (Indistinguishability of $\mathrm{Game}_2^\sigma(\kappa)$ and $\mathrm{Game}_3^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_2^\sigma(\kappa)$ and $\mathrm{Game}_3^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_2^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_3^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. In the first step, we simulate the oracle by first checking if the first sending parties message is a valid *rerandomization* of the message from the challenge transcript, *i.e.*, if the message differs but decrypts to the same value. We then additionally add a check whether the first sending parties message differs *and* has a different decryption.
We can then combine the two branches to obtain the behavior described in $\mathrm{Game}_3^\sigma(\kappa)$.

H_0	H_1
–	**if** $x_0^{(0)} = \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
–	$(\sigma_0^{(0)}, \mu_0^{(0)}) := \pi_{\mathcal{C}}[(\sigma_0^{(0)}, \mu_0^{(0)})]$
$p := -1$	$p := -1$
if $\pi = \pi_{\mathcal{C}}$ **then**	**if** $\pi = \pi_{\mathcal{C}}$ **then**
$p := 1-(1-2d)^2/2$	$p := 1-(1-2d)^2/2$
fi	**fi**

H_2

if $x_0^{(0)} = \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \pi_{\mathcal{C}}[(\sigma_0^{(0)}, \mu_0^{(0)})]$
$p \coloneqq -1$
if $\pi = \pi_{\mathcal{C}}$ **then**
  $p \coloneqq 1 - (1-2d)^2/2$
elseif $\big(\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}] \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}) = \pi_{\mathcal{C}}[(\sigma_0^{(0)}, \mu_0^{(0)})]\big)$ **then**
  $p \coloneqq 1/2$
–
–
fi

H_3

if $x_0^{(0)} = \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  $(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \pi_{\mathcal{C}}[(\sigma_0^{(0)}, \mu_0^{(0)})]$
$p \coloneqq -1$
if $\pi = \pi_{\mathcal{C}}$ **then**
  $p \coloneqq 1 - (1-2d)^2/2$
elseif $\big(\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}] \wedge$
  $\mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}) = \pi_{\mathcal{C}}[(\sigma_0^{(0)}, \mu_0^{(0)})]\big)$ **then**
  $p \coloneqq 1/2$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  $p \coloneqq 1/2$
fi

H_4

$p \coloneqq -1$
if $\pi = \pi_{\mathcal{C}}$ **then**
  $p \coloneqq 1 - (1-2d)^2/2$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  $p \coloneqq 1/2$
fi

H_0 This hybrid is identical to $\mathrm{Game}_2^{\sigma}(\kappa)$.

H_1 The oracle programs the content of the first sender message of the challenge transcript into the circuit. Due to perfect correctness of PKE, games H_0 and H_1 are distributed identically. This step prevents that decryption is called on the first sender message of the challenge transcript.

H_2 The oracle performs an initial check. If the first sending parties message is different from the message reported in $\pi_{\mathcal{C}}$, but still contains the same content, then the oracle returns a uniformly random bit.
This game hop is justified by the NM-CCA security of PKE, which PKE satisfies since it is IND$-CCA secure, [BDP+98]. Consider a PPT distinguisher $\mathcal{D}$ between H_1 and H_2. We construct an adversary $\mathcal{A}$ breaking the NM-CCA security of PKE as follows. $\mathcal{A}$ simulates H_2 for $\mathcal{D}$ by first uniformly sampling a bit $\sigma_0^{(0)} \xleftarrow{\$} \{0,1\}$ to-be-transferred, creating a signature key-pair $(\mathsf{k}, \mathsf{vk}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$ and computing the signature $\mu_0^{(0)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}, \sigma_0^{(0)})$. Next, $\mathcal{A}$ is sending the message $(\sigma_0^{(0)}, \mu_0^{(0)})$ to its NM-CCA challenger $\mathcal{C}$ who in return provides $\mathcal{A}$ with the challenge ciphertext ct^*. $\mathcal{A}$ simulates the oracle P_{AT} using the previously created verification key vk for signature checks and by forwarding all decryption queries to the decryption oracle provided by $\mathcal{C}$. Note that the decryption oracle is never queried for ct^* due to the changes made in H_1.
H_1 and H_2 behave fully identically except if $\mathcal{D}$ makes an oracle query that causes the simulated oracle to enter the newly introduced elseif-branch. If this event occurs, $\mathcal{A}$ returns $x_0^{(0)}$ and the relation "equality". Hence, $\mathcal{D}$'s advantage in distinguishing H_1 and H_2 is upper bounded by some negligible function.

H_3 In this step, we additionally consider first sender-party messages which decrypt to a *different* message.
Let $\mathcal{D}$ be a PPT distinguisher between H_2 and H_3. We build an adversary $\mathcal{A}$ breaking the sEUF-CMA security of SIG as follows. $\mathcal{A}$ simulates H_2 for $\mathcal{D}$ by choosing the encryption parameters via PKE.KeyGen and by using oracle-queries to obtain valid signatures of σ for the first round and of $\pi[1]$ for the second round.
In order to distinguish H_2 and H_3, a distinguisher $\mathcal{D}$ must cause the oracle to behave differently. That is, $\mathcal{D}$ needs to make the oracle in H_3 take the newly introduced elseif-branch such that the oracle in H_2 produces a different output distribution. Given the event that the oracle in H_3 enters the new elseif-branch and the oracle in H_2 does not behave identically, *i.e.* does not return a uniformly random bit. Then, either

$$\mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0]) \quad \text{or} \tag{40}$$

$$\mathrm{SIG.Vfy}(\mathsf{vk}, \mu_b^{(0)}, \sigma_b^{(0)}) \tag{41}$$

for $b = 0$ needs to be satisfied to ensure differing behavior. Let E_1 denote the event that Eq. (40) occurs and let E_2 denote the event that Eq. (41) occurs. Due to the changes introduced in H_2, the new elseif-branch can only be executed if $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **and**

$$(\sigma_0^{(0)}, \mu_0^{(0)}) := \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}) \neq \pi_{\mathcal{C}}[(\sigma_0^{(0)}, \mu_0^{(0)})].$$

Hence, if E_2 occurs, the sEUF-CMA adversary $\mathcal{A}$ outputs $(\sigma_0^{(0)}, \mu_0^{(0)})$ which is either a valid signature for message $1 - \sigma_0^{(0)}$ (which has never been queried via the Sig oracle[18]) or a fresh valid signature for $\sigma_0^{(0)}$. If E_1 occurs, $\mu_b^{(1)}$ is a valid signature for the first round transcript, *i.e.*, particularly for the message $[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$. In this case, the sEUF-CMA adversary $\mathcal{A}$ outputs $(\pi[0], \mu_b^{(1)})$. The message $\pi[0]$ contains $[x_0^{(0)}]$ and has hence never been sent to the Sig oracle. Hence, the advantage of $\mathcal{D}$ in distinguishing H_2 and H_3 is upper bounded by some negligible function.

H_4 The oracle in H_3 and H_4 behaves identically, hence H_3 and H_4 are identically distributed. Further, H_4 is identical to $\mathrm{Game}_3^{\sigma}(\kappa)$.

This concludes the proof that $|\Pr[\mathsf{out}_{\mathrm{Game}_3^{\sigma}(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_2^{\sigma}(\kappa)} = 1]|$ is negligible.

□

$\mathrm{Game}_4^{\sigma}(\kappa)$ This game is as $\mathrm{Game}_3^{\sigma}(\kappa)$, but during simulation of the oracle P_{AT} from Fig. 10, if the input shares the same first-round message of the sender but has a different first-round message of the receiver, the program outputs a uniformly random bit where the randomness is taken over all three first-round messages.

Lemma 23 (Indistinguishability of $\mathrm{Game}_3^{\sigma}(\kappa)$ and $\mathrm{Game}_4^{\sigma}(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_3^{\sigma}(\kappa)$ and $\mathrm{Game}_4^{\sigma}(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_3^{\sigma}(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_4^{\sigma}(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Let $\mathcal{D}$ be a distinguisher between $\mathrm{Game}_3^{\sigma}(\kappa)$ and $\mathrm{Game}_4^{\sigma}(\kappa)$ with non-negligible advantage α. From $\mathcal{D}$ we can construct an adversary $\mathcal{A}$ on the sEUF-CMA security of the used signature scheme SIG.

In order to differentiate $\mathcal{D}$ has to input a transcript π that in $\mathrm{Game}_4^{\sigma}(\kappa)$ returns a uniformly random bit, but that in $\mathrm{Game}_3^{\sigma}(\kappa)$ returns some bit σ with probability $1/2 + \alpha^*$ with non-negligible bias α^*. Yet the only difference in those two games is the case where the first sending parties message is the same as in the challenge transcript and the receiver message differs from the challenge transcript. Thus, in order to differentiate $\mathcal{D}$ has to insert a transcript π with the property that $\pi[x_0^{(0)}] = \pi_{\mathcal{C}}[x_0^{(0)}]$. Due to the overwhelming correctness of PKE this fixes both the bit $\sigma_{\mathcal{C}}$ and the signature μ. The signature, however, is on $(\sigma_{\mathcal{C}}, x_{\mathsf{R}}^{(0)})$. Thus, this *fixes* the first receiver message.

Assuming that $\mathcal{D}$ also uses the same second-round message for the sender, that is, if $\pi[x_0^{(1)}] = \pi_{\mathcal{C}}[x_0^{(1)}]$, then the check in line 23 fails as well as the one in line 26, causing the circuit to return a uniformly random bit. The only way to cause a differentiating behavior is thus to forge a second-round message of the sender as well. This would be a valid forgery of a signature on a message not queried before and thus would violate the sEUF-CMA security.

More formally, $\mathcal{A}$ works as follows:

Creation of the Transcript. The creation of the transcript is the same in both games. Thus the adversary samples a uniformly random bit $\sigma_{\mathcal{C}}$ and completely simulates the encryption scheme PKE. It replaces the signature creations of P_0 with queries to the signature oracle provided by the sEUF-CMA challenger. $\mathcal{A}$ reports the transcript $\pi_{\mathcal{C}}$ to the distinguisher $\mathcal{D}$.

[18] Note that the reduction only uses the Sig oracle to produce the challenge transcript and, hence, never queries the Sig oracle on $1 - \sigma_0^{(0)}$.

Simulation of the Oracle. To provide an output the adversary $\mathcal{A}$ uniformly randomly selects whether to return the bit according to the code from $\text{Game}_3^\sigma(\kappa)$ or $\text{Game}_4^\sigma(\kappa)$.
It then follows the respective code, which is trivially possible as $\mathcal{A}$ gets the verification key from the sEUF-CMA challenger. Additionally, on each input π, the adversary checks if the following conditions hold: (1)$\pi[x_0^{(0)}] = \pi_\mathcal{C}[x_0^{(0)}]$, (2)$\pi[x_\mathsf{R}^{(0)}] \neq \pi_\mathcal{C}[x_\mathsf{R}^{(0)}]$, (3)$\pi[x_0^{(1)}] \neq \pi_\mathcal{C}[x_0^{(1)}]$ (4) The signature μ encoded in $\pi[x_0^{(1)}]$ is a valid signature on $(x_\mathsf{R}^{(0)}, x_0^{(0)}, x_1^{(0)})$.
There are two possibilities. Either none of the queries made by $\mathcal{D}$ fulfills all four conditions, then—as mentioned before—the behavior of the circuit is exactly the same in both games and hence $\alpha = 0$. Or at least one of the queries made by $\mathcal{D}$ fulfills all four conditions—which by assuming $\alpha \notin \mathsf{negl}(\kappa)$ has to be the case. Then the signature μ stored in $x_0^{(1)}$ is a valid forgery to a message that was not previously queried by the adversary, hence μ can be used to break the sEUF-CMA security of the signature scheme Sig with non-negligible probability.
Since we assumed this not to be the case due to the sEUF-CMA security of Sig, it follows that $\mathcal{D}$ has at most a negligible advantage which proves our claim. □

$\text{Game}_5^\sigma(\kappa)$ This game follows $\text{Game}_4^\sigma(\kappa)$, but during simulation of the oracle P_{AT} from Fig. 10, if the input shares the same first-round message of the sender and receiver but has a different first-round message of the dummy-friend, the program outputs the correct bit $\sigma_\mathcal{C}$ with probability $1/2 + d$ and the wrong bit with the complementary probability.

Lemma 24 (Indistinguishability of $\text{Game}_4^\sigma(\kappa)$ and $\text{Game}_5^\sigma(\kappa)$). *Let Pke be an IND$-CCA secure public-key encryption scheme. Let Sig be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for $\text{Game}_4^\sigma(\kappa)$ and $\text{Game}_5^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_4^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_5^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Again, the protocol messages are equivalent, we only take a shortcut in the simulation of the oracle.
The case we're in implies that the first sending parties message to be equivalent to the challenge transcript, but the first dummy friend message is different. In this case, the game essentially claims that the only path that is taken is the one where the verification of the signature in the *second* sending parties message fails, and the *first* sending parties message is valid, thus returning the output with probability $1/2 + d$.
Again, a distinguisher $\mathcal{D}$ would have to cause some sort of different behavior; that is, creating some input transcript π that shares the first sending parties message with $\pi_\mathcal{C}$, has a different dummy friend first message than $\pi_\mathcal{C}$, and arbitrary second-round messages.
While the second dummy friend message is independent of the bit σ, as it was chosen randomly, the second sending parties message will be considered invalid by the circuit, since the signature is on a different dummy friend parties message.
As such, in order to obtain a different output probability, subject to the fact that the first sending parties message remains the same and the first dummy friends message is changed, the distinguisher $\mathcal{D}$ would either have to create a valid second-round message that contains a signature on the first round (which is equally hard for any party) or forge a valid first-round dummy-friend message that is considered a valid sending parties input.
However, assuming again Non-Malleability of Pke and sEUF-CMA of Sig, we can show that no adversary can cause an altering behavior in $\text{Game}_4^\sigma(\kappa)$, that is, providing an input with the same first sending-party message and a different dummy-friend message that causes a different output distribution than the one that is biased with $1/2 + d$ towards σ.

□

$\text{Game}_6^\sigma(\kappa)$ This game follows $\text{Game}_5^\sigma(\kappa)$, but during simulation of the oracle P_{AT} from Fig. 10, if the input shares the same first-round messages in the transcript, but a different second-round message than the sending party, the program outputs the correct bit $\sigma_\mathcal{C}$ with probability $1/2 + d$ and the wrong bit with the complementary probability.

Lemma 25 (Indistinguishability of $\text{Game}_5^\sigma(\kappa)$ and $\text{Game}_6^\sigma(\kappa)$). *Let Pke be an IND$-CCA secure public-key encryption scheme. Let Sig be an sEUF-CMA secure signature scheme. For*

all PPT guessing algorithms A*, the distinguishing advantage for* $\mathrm{Game}_5^\sigma(\kappa)$ *and* $\mathrm{Game}_6^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_5^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_6^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. This proof again just takes a shortcut in the simulation of the protocol. More precisely, we immediately capture the case where the second sending parties message is wrong.
Let π be a transcript that helps the distinguisher $\mathcal{D}$ in distinguishing these two games. We know from the two previous games that π differs from $\pi_\mathcal{C}$ in at least one second-round message, but the first-round messages are equivalent to the challenge transcript.
If the second dummy-friend message, $x_1^{(1)}$, differs, then the two games are actually equivalent, leaking absolutely no distinguishing information. Hence, in order to distinguish, $\mathcal{D}$ has to query the oracle with some input π where $\pi[x_0^{(1)}] \neq \pi_\mathcal{C}[x_0^{(1)}]$.
In the general program flow of $\mathrm{Game}_5^\sigma(\kappa)$, the probability is that one hard-coded in $\mathrm{Game}_6^\sigma(\kappa)$ unless either the first **if** in line 20 results in true, or the **elseif** in line 23 is false.
Line 23 is fixed, however, since the randomness in both games is taken from the same source, which we know to be equivalent to the challenge transcript.
Against line 20, we can create an efficient reduction algorithm breaking the sEUF-CMA-property of SIG, since this message has to contain a valid signature on the first round transcript.

□

$\mathrm{Game}_7^\sigma(\kappa)$ This game follows $\mathrm{Game}_6^\sigma(\kappa)$, but during simulation of the oracle P_{AT} from Fig. 10, if the input is equivalent to the challenge transcript *except for the second dummy-friend message*, output the correct bit with probability $1 - (1 - 2d)^2/2$.

Lemma 26 (Indistinguishability of $\mathrm{Game}_6^\sigma(\kappa)$ and $\mathrm{Game}_7^\sigma(\kappa)$). *Let* PKE *be an IND$-CCA secure public-key encryption scheme. Let* SIG *be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_6^\sigma(\kappa)$ *and* $\mathrm{Game}_7^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_6^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_7^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. In both games, the only distinguishing behavior can be induced by messages which differ from the challenge transcript $\pi_\mathcal{C}$ exactly in the final dummy-friend message. However, this message information-theoretically contains no information regarding the transferred bit. Replacing randomness with randomness changes nothing, so the only way a distinguisher can detect the change is by inserting a *new* message for the dummy friend, which contains a valid signature on the first round. Similar to $\mathrm{Game}_6^\sigma(\kappa)$, the games can be sufficiently well simulated to allow a reduction to the sEUF-CMA property of SIG.

□

$\mathrm{Game}_8^\sigma(\kappa)$ This game follows $\mathrm{Game}_7^\sigma(\kappa)$, but reports uniformly random messages for all parties in the challenge transcript.

Lemma 27 (Indistinguishability of $\mathrm{Game}_7^\sigma(\kappa)$ and $\mathrm{Game}_8^\sigma(\kappa)$). *Let* PKE *be an IND$-CCA secure public-key encryption scheme. Let* SIG *be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_7^\sigma(\kappa)$ *and* $\mathrm{Game}_8^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_7^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_8^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Let $\mathcal{D}$ be a distinguisher between $\mathrm{Game}_7^\sigma(\kappa)$ and $\mathrm{Game}_8^\sigma(\kappa)$ with success probability $1/2 + \alpha$. Out of $\mathcal{D}$, we construct an efficient adversary $\mathcal{A}$ on the IND$-CPA property of PKE:
Let $\mathcal{C}$ be the IND$-CPA challenger. We once more adapt the Left-Right (LR) view, where the challenger initially sets up an oracle and decides in the preparation phase, if on input x the oracle should always return $\mathrm{PKE.Enc}(\mathsf{sk}, x)$, or if it always returns a uniformly random string of the same size. The adversary $\mathcal{A}$ wins, iff it can guess which oracle was used.
After obtaining oracle access from $\mathcal{C}$, $\mathcal{A}$ samples a random OTP and sends OTP to the oracle. The result is used as receiver message. Now $\mathcal{A}$ creates a signature-pair $(\mathsf{k}, \mathsf{vk}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$ and draws a random bit $\sigma \xleftarrow{\$} \{0, 1\}$. Then, $\mathcal{A}$ computes a signature μ from σ using k and

sends $(\sigma\|\mu)$ to the oracle. The output is added to the challenge transcript $\pi_{\mathcal{C}}$ as $x_0^{(0)}$. The adversary then samples some uniformly random $x_1^{(0)}$ of the same size and adds it to the challenge transcript.
$\mathcal{A}$ then computes a signature μ on both those first-round messages and sends $x\|\mu$ to the oracle, the returning value is taken as $x_0^{(1)}$. Again, the dummy-friend message is chosen at random.
The adversary $\mathcal{A}$ then reports $\pi_{\mathcal{C}}$ to $\mathcal{D}$, who is allowed to make queries to the obfuscated circuit on its own. Note that simulation of the circuit is trivially possible: Due to our previous games, no input manages to pass line 15 in the depiction of the oracle, so all that $\mathcal{A}$ does is comparing similarity of inputs π with the challenge transcript $\pi_{\mathcal{C}}$ and acting accordingly.
Eventually, $\mathcal{D}$ outputs one of $\text{Game}_7^{\sigma}(\kappa)$ or $\text{Game}_8^{\sigma}(\kappa)$. On output $\text{Game}_7^{\sigma}(\kappa)$, $\mathcal{A}$ reports that a genuine encryption oracle has been used, on output $\text{Game}_8^{\sigma}(\kappa)$, $\mathcal{A}$ reports that the oracle returned randomness.
Those two cases are equivalent, as an encryption oracle yields exactly the distribution from $\text{Game}_7^{\sigma}(\kappa)$, meaning that the message was actually an encryption, and a random oracle causes all reported messages to be random; just as in $\text{Game}_8^{\sigma}(\kappa)$.
It thus follows that the success probability of $\mathcal{A}$ is $1/2+\alpha$ for $\alpha \in \mathsf{negl}(\kappa)$ from the IND$-CPA security of PKE, and since α from $\mathcal{A}$ is the same as the one from $\mathcal{D}$, this implies $\mathcal{D}$ cannot be better than $1/2+\alpha$. □

$\text{Game}_9^{\sigma}(\kappa)$ This game follows $\text{Game}_8^{\sigma}(\kappa)$, but changes the oracle simulation. Instead of considering P_0 to be the sending party of the challenge transcript, the oracle in $\text{Game}_9^{\sigma}(\kappa)$ acts as if P_1 is the sender and P_0 is the dummy friend. Note that this also implies a changed message to-be-considered for the resp. coin-tosses.

Lemma 28 (Indistinguishability of $\text{Game}_8^{\sigma}(\kappa)$ and $\text{Game}_9^{\sigma}(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. Let c be chosen such that $c \geq t$. For all PPT guessing algorithms A, the distinguishing advantage for $\text{Game}_8^{\sigma}(\kappa)$ and $\text{Game}_9^{\sigma}(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_8^{\sigma}(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_9^{\sigma}(\kappa)} = 1]| \leq d \cdot (3 - 2d)$$

Proof. Based on the randomness of the coin toss involved in creating the challenge messages, we can fix the world we are in based on the outcomes of the coin-tosses from F.
Fig. 11 shows the different worlds that exist based on the randomness involved in the creation of the challenge transcript. A node contains three parts: The top part shows the output of the correct challenge transcript and, if it exists, the corresponding world in the description that follows. The middle part shows the probability that we end up in this world. The bottom part shows the probability that an adversary can correctly guess the sender in this world; note that the adversary can always guess, so the optimum here is at 1/2.
As we can see in Fig. 11, the randomness extracted by the PRF brings us in one of several worlds:

$World_1$ fixes the randomness extracted from the two first-round messages to yield the correct bit, and contains a first sender message that causes the coin-toss to land on heads and hence to ignore the bit contained in this round. This is the optimal world, as any changes performed in the second round result in the circuit outputting the correct bit—there is no path that causes the circuit to output anything other than $\sigma_{\mathcal{C}}$. Similarly, since the original first message round is ignored due to the coin landing on heads, changing any first-round message replaces randomness with new randomness and hence only causes a new uniformly distributed bit.
Hence, in this world, an adversary can trivially only guess.
Finally, note that the adversary can uniquely identify this world: It is the only world in Fig. 11, where the correct bit $\sigma_{\mathcal{C}}$ is output, yet any new message of any party yields a new uniformly random bit.

$World_2$ contains a first-round message-pair where the corresponding coin-toss still yields the correct bit, but the first-round sender-message additionally fixes the bit to $\sigma_{\mathcal{C}}$. Unlike $World_1$, this allows for an efficient attack; the adversary can try out a polynomial number of random messages for $x_0^{(0)}$ and then a polynomial number of random messages for $x_1^{(0)}$ with the resp.

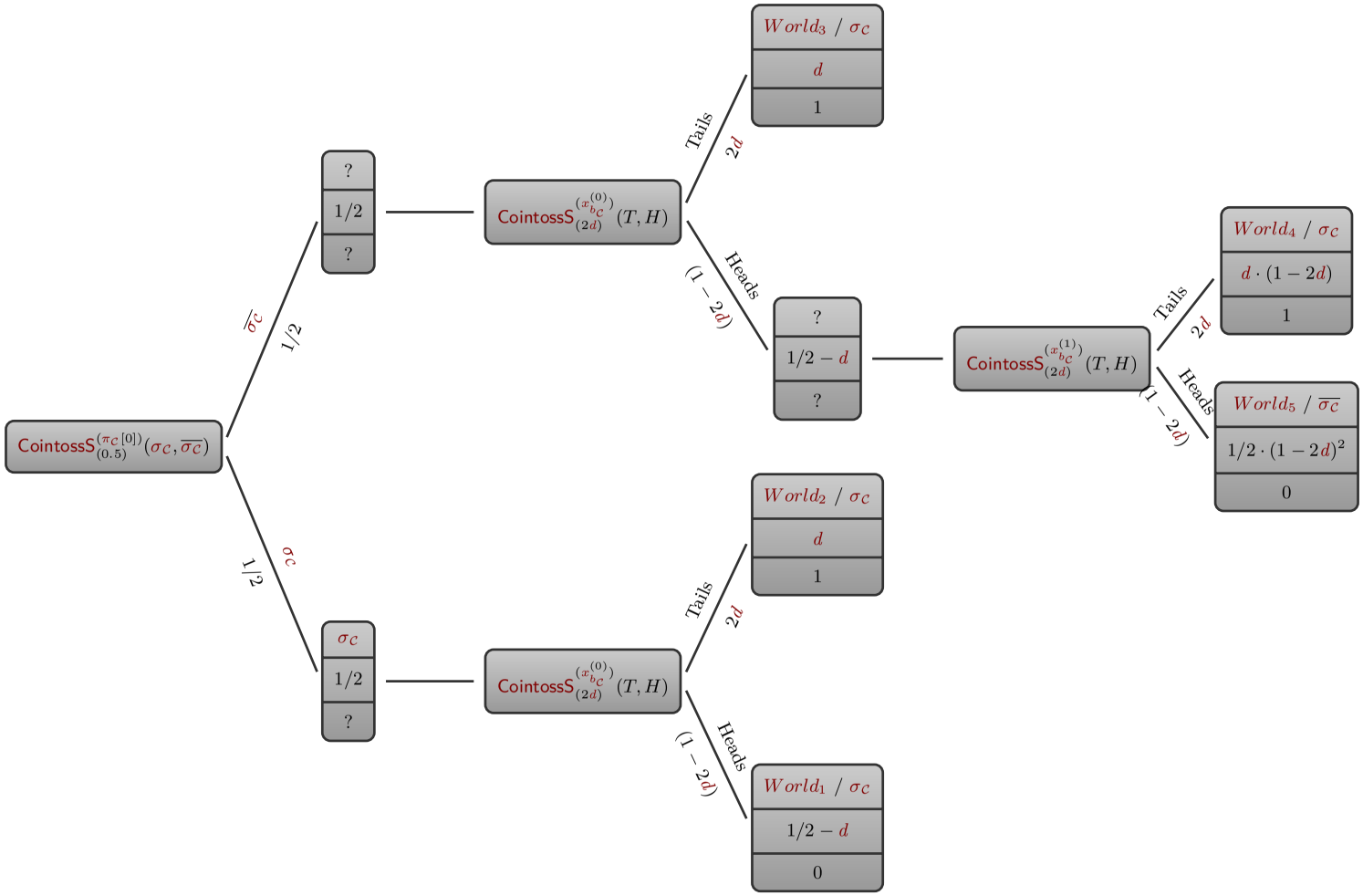


Fig. 11: Tree showcasing the different worlds, based on the randomness from a uniformly random challenge transcript.

other message taken from the challenge transcript. If the new message is from the sender, then the circuit always falls back to the uniform output, and the output is a uniform bit. If the new message is from the dummy friend, then the circuit extracts the correct bit from the sending parties first message and always outputs the correct bit. Hence, the adversary is able to clearly identify the sender.

$World_3$ has the first round messages such that the coin-toss yields a different bit $\overline{\sigma_{\mathcal{C}}}$. This line is more dangerous, as the path where all coin-tosses fail can now be efficiently distinguished from the one where any of the coin-tosses succeeds. Furthermore, in this world, the first sending parties coin lands on tails, thus fixing the bit immediately.
Note that in this world, the adversary can launch the same attack as in $World_2$; changing $x_b^{(0)}$ yields a new output with every new message.
We stress that this world is indistinguishable to $World_2$ for any (even statistical) adversary $\mathcal{A}$; the challenge transcript does not even consider the coin-toss based on both parties first-round messages, and any new first-round message of either the dummy friend or the sender yields new randomness to be used by F. But since they both share the same attack, output and probability, we stress that there is no advantage for any adversary in distinguishing $World_2$ and $World_1$. However, note that the adversary can trivially distinguish the cases of being in either $World_2$ or $World_1$ from being in any of the other worlds mentioned in Fig. 11.

$World_4$ has a first-round challenge transcript that causes the PRF to output the wrong bit, a first round sender-message that does not fix the output, but a second-round sender-message that does fix the bit.
In this world, the distinguishing attack is similar to the one for the first round, only that those messages are now taken from the challenge transcript, resulting in a one-shot behavior. With the entire challenge transcript outputting $\sigma_{\mathcal{C}}$, the circuit will also output $\sigma_{\mathcal{C}}$ for any other arbitrarily chosen dummy-friend message. But since the second sending parties message fixes the bit, changing that one causes all coin-tosses to fail and hence the circuit to output bit $\overline{\sigma_{\mathcal{C}}}$.
This world can again be uniquely recognized by the adversary, as it is the only world where changing any of the first-round message results in uniformly random output, but only a new second-round sending parties message flips the output bit.

$World_5$ again has the initial round challenge such that the PRF outputs the wrong bit $\overline{\sigma_{\mathcal{C}}}$, and the two sending parties messages such that none of the coin-tosses results in the circuit fixing the bit.
In this world, only randomness is involved, and the final output of even the challenge transcript is $\overline{\sigma_{\mathcal{C}}}$—the wrong bit.
This world is unique to the adversary, as it is the only one where the wrong bit is output. However, note that since no identifying information whatsoever is contained in the challenge transcript, an adversary trapped in this world can only guess who the sender is.

So we have seen that after any randomly sampled two-rounds challenge transcript $\pi_{\mathcal{C}}$, the adversary can be placed in one of five worlds. In Fig. 11, we additionally annotated the probability of ending up in this world. Since we have a binary outcome—the adversary can *either* uniquely determine the sender with 100% correctness, yielding an advantage of 1 in figuring out which game has been played, or do none better than guessing, which ends up with an advantage of 0. The advantage α of any distinguisher in correctly differentiating $\mathrm{Game}_8^{\sigma}(\kappa)$ from $\mathrm{Game}_9^{\sigma}(\kappa)$ comes down to the following formula:

$$\begin{aligned}\alpha \leq & \sum_{i=1}^{5} \Pr[World_i] \cdot \alpha_{World_i} \\ =& (1/2 - d + 1/2 \cdot (1-2d)^2) \cdot 0 + (d + d + d \cdot (1-2d)) \cdot 1 \\ =& d \cdot (3 - 2d)\end{aligned} \tag{42}$$

□

$\mathrm{Game}_{10}^{\sigma}(\kappa)$ This game follows $\mathrm{Game}_9^{\sigma}(\kappa)$, but reports messages for an honest sender in the name of P_1.

Lemma 29 (Indistinguishability of $\mathrm{Game}_9^\sigma(\kappa)$ and $\mathrm{Game}_{10}^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_9^\sigma(\kappa)$ and $\mathrm{Game}_{10}^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_9^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_{10}^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 27. □

$\mathrm{Game}_{11}^\sigma(\kappa)$ This game follows $\mathrm{Game}_{10}^\sigma(\kappa)$, but simulates the oracle in such a way, that the explicit check regarding equality of the second dummy friends message between the challenge transcript and the actual input is dropped.

Lemma 30 (Indistinguishability of $\mathrm{Game}_{10}^\sigma(\kappa)$ and $\mathrm{Game}_{11}^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_{10}^\sigma(\kappa)$ and $\mathrm{Game}_{11}^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_{10}^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_{11}^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 26. □

$\mathrm{Game}_{12}^\sigma(\kappa)$ This game follows $\mathrm{Game}_{11}^\sigma(\kappa)$, but simulates the oracle in such a way, that the explicit check regarding equality of the second senders message between the challenge transcript and the actual input is dropped.

Lemma 31 (Indistinguishability of $\mathrm{Game}_{11}^\sigma(\kappa)$ and $\mathrm{Game}_{12}^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_{11}^\sigma(\kappa)$ and $\mathrm{Game}_{12}^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_{11}^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_{12}^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 25 □

$\mathrm{Game}_{13}^\sigma(\kappa)$ This game follows $\mathrm{Game}_{12}^\sigma(\kappa)$, but simulates the oracle in such a way, that the explicit check regarding equality of the first dummy friends message between the challenge transcript and the actual input is dropped.

Lemma 32 (Indistinguishability of $\mathrm{Game}_{12}^\sigma(\kappa)$ and $\mathrm{Game}_{13}^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_{12}^\sigma(\kappa)$ and $\mathrm{Game}_{13}^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_{12}^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_{13}^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 24. □

$\mathrm{Game}_{14}^\sigma(\kappa)$ This game follows $\mathrm{Game}_{13}^\sigma(\kappa)$, but simulates the oracle in such a way, that the explicit check regarding equality of the receivers message between the challenge transcript and the actual input is dropped.

Lemma 33 (Indistinguishability of $\mathrm{Game}_{13}^\sigma(\kappa)$ and $\mathrm{Game}_{14}^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_{13}^\sigma(\kappa)$ and $\mathrm{Game}_{14}^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_{13}^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_{14}^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 23. □

$\mathrm{Game}_{15}^\sigma(\kappa)$ This game follows $\mathrm{Game}_{14}^\sigma(\kappa)$, but simulates the oracle in such a way, that the explicit check regarding equality of the first sending parties message between the challenge transcript and the actual input is dropped.

Lemma 34 (Indistinguishability of $\mathrm{Game}_{14}^{\sigma}(\kappa)$ and $\mathrm{Game}_{15}^{\sigma}(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_{14}^{\sigma}(\kappa)$ and $\mathrm{Game}_{15}^{\sigma}(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_{14}^{\sigma}(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_{15}^{\sigma}(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 22. □

$\mathrm{Game}_{16}^{\sigma}(\kappa)$ This game follows $\mathrm{Game}_{15}^{\sigma}(\kappa)$, but simulates the oracle in such a way, that the explicit check regarding equality of the challenge transcript and the actual input is dropped.

Lemma 35 (Indistinguishability of $\mathrm{Game}_{15}^{\sigma}(\kappa)$ and $\mathrm{Game}_{16}^{\sigma}(\kappa)$). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\mathrm{Game}_{15}^{\sigma}(\kappa)$ and $\mathrm{Game}_{16}^{\sigma}(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_{15}^{\sigma}(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_{16}^{\sigma}(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 21. □

We thus can go from a valid transcript for P_0 as sending party to a valid transcript for P_1 as sending party using only gamehops with negligible distinguishing advantage and one gamehop ($\mathrm{Game}_{9}^{\sigma}(\kappa)$) with advantage d. Thus, no algorithm for deanonymizing the sending party can have advantage significantly better than d.

Corollary 10 (Anonymity). *Let Π_{AT}^{1} be instantiated as described in Fig. 9. No guessing algorithm A can extract the identity b of the sending party P_b better than with probability $1/2 + d$.*

We can now investigate the actual privacy of our algorithm.

Lemma 36 (δ-Anonymity). *The protocol from Fig. 9 is δ-anonymous with $\delta = (1 - 2d)$.*

Proof. The formula for δ-anonymity is given in Eq. (4) as

$$\left| \Pr_{b \xleftarrow{\$} \{0,1\}} \left[\mathrm{Exp}^{anon}_{\Pi^{1}_{AT},\mathcal{A},b}(\kappa) = b \right] - 1/2 \right| \leq (1 - \delta)/2$$

We have seen in Corollary 10 that the probability of winning $\mathrm{Exp}_{\Pi^{1}_{AT},\mathcal{A},b}$ is bounded by $1/2 + d$, which yields

$$|1/2 + d - 1/2| = d \leq (1 - \delta)/2$$

This can be reformulated into $\delta \leq 1 - 2d$. Since δ is the maximum number that that fulfills this equation, we obtain our claim that $\delta = 1 - 2d$. □

Secrecy. Now we analyze the chances of an algorithm A being able to output the correct bit *without* knowing the receivers random tape. Those chances are low due to the One-Time-Pad—our proof basically proves that (1) without the One-Time-Pad the output can not be reconstructed better than by randomly guessing a bit, and (2) without the secret key sk_P hidden by the P_{AT} the One-Time-Pad can not be reconstructed. Thus secrecy follows from the statistical security of One-Time-Pads and the IND$-CCA-security of PKE.

We now show indistinguishable of the transferred bit to any guessing algorithm A that does not use the receiver's random tape. To that end, we have to show indistinguishability of the following two distributions against any adversary that does not know the receiver's random tape:

$$\begin{aligned} &\{\sigma := 0; \pi \xleftarrow{\$} \langle \mathsf{R}, \mathsf{P}_0, \mathsf{P}_1 \rangle(b, \sigma) \colon \pi\} \\ \approx\ &\{\sigma := 1; \pi \xleftarrow{\$} \langle \mathsf{R}, \mathsf{P}_0, \mathsf{P}_1 \rangle(b, \sigma) \colon \pi\}. \end{aligned} \tag{43}$$

We thus provide a proof reminiscent of that for anonymity but we deviate on different spots.

Essentially the game hops are the same until $\mathrm{Game}_{8}^{\sigma}(\kappa)$ which is distinguishable from $\mathrm{Game}_{1}^{\sigma}(\kappa)$ with only negligible advantage as shown in the anonymity proof. Our major difference is that we start with a sender who always transfers $\sigma = 0$, and in the situation resembling $\mathrm{Game}_{8}^{\sigma}(\kappa)$ we swap the message to be $\sigma = 1$. We claim that this can not be distinguished and hence show overwhelming secrecy $\varsigma \in (1 - \mathsf{negl}(\kappa))$.

$\mathrm{Game}_1(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}_{(2d)}^{(x_b^{(1)})}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}_{(2d)}^{(x_b^{(0)})}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}_{(2d)}^{(x_b^{(0)})}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}_{(1/2)}^{(\pi[0])}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:
$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (0, x_{\mathsf{R}}^{(0)}))$
$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, 0 \| \mu)$
$\mathsf{P_1}$:
$-$
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, 0 \| \mu)$
$\mathsf{P_1}$:
$-$
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_2(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}_{(1-(1-2d)^2/2)}^{(\pi_{\mathcal{C}})}(0,1)$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}_{(0.5)}^{(\pi[0])}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}_{(0.5)}^{(\pi[0])}(0,1)$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}_{(d)}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}(OTP \oplus 0, -)$
  return $\mathsf{CointossS}_{(0.5)}^{(\pi[0])}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}_{(d)}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}(OTP \oplus 0, -)$
  return $\mathsf{CointossS}_{(0.5)}^{(\pi[0])}(0,1)$
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
  return $\mathsf{CointossS}_{(2d)}^{(\pi_{\mathcal{C}}[x_0^{(1)}])}(OTP \oplus 0, -)$
  return $\mathsf{CointossS}_{(2d)}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}(OTP \oplus 0, -)$
  return $\mathsf{CointossS}_{(0.5)}^{(\pi[0])}(0,1)$
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}_{(2d)}^{(x_b^{(1)})}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}_{(2d)}^{(x_b^{(0)})}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}_{(2d)}^{(x_b^{(0)})}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}_{(1/2)}^{(\pi[0])}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$-$
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$-$
$-$
$x_0^{(1)} \leftarrow \{0,1\}^m$
$\mathsf{P_1}$:
$-$
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_2(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(0,1)$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus 0, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus 0, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(1)}])}_{(2d)}(OTP \oplus 0, -)$
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(2d)}(OTP \oplus 0, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$-$
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$-$
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$-$
$x_0^{(1)} \leftarrow \{0,1\}^m$
$\mathsf{P_1}$:
$-$
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$

$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(1,0)$
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus 1, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus 1, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(1)}])}_{(2d)}(OTP \oplus 1, -)$
  return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(2d)}(OTP \oplus 1, -)$
  return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$
fi
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
  return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b$: $\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
  return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$

R:
$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$-$
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_1}$:
$-$
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:
$-$
$x_0^{(1)} \leftarrow \{0,1\}^m$
$\mathsf{P_1}$:
$-$
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$	$\mathrm{Game}_4(\kappa)$
$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$	$(OTP) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
if $\pi = \pi_{\mathcal{C}}$ **then**	–
return $OTP \oplus \mathsf{CointossS}^{(\pi_{\mathcal{C}})}_{(1-(1-2d)^2/2)}(1,0)$	–
elseif $\pi[x_0^{(0)}] \neq \pi_{\mathcal{C}}[x_0^{(0)}]$ **then**	–
return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$	–
elseif $\pi[x_{\mathsf{R}}^{(0)}] \neq \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**	–
return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$	–
elseif $\pi[x_1^{(0)}] \neq \pi_{\mathcal{C}}[x_1^{(0)}]$ **then**	–
return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus 1, -)$	–
return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$	–
elseif $\pi[x_0^{(1)}] \neq \pi_{\mathcal{C}}[x_0^{(1)}]$ **then**	–
return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(d)}(OTP \oplus 1, -)$	–
return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$	–
elseif $\pi[x_1^{(1)}] \neq \pi_{\mathcal{C}}[x_1^{(1)}]$ **then**	–
return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(1)}])}_{(2d)}(OTP \oplus 1, -)$	–
return $\mathsf{CointossS}^{(\pi_{\mathcal{C}}[x_0^{(0)}])}_{(2d)}(OTP \oplus 1, -)$	–
return $\mathsf{CointossS}^{(\pi[0])}_{(0.5)}(0,1)$	–
fi	–
$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$	$(\sigma_0^{(0)}, \mu_0^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)})$
$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$	$(\sigma_1^{(0)}, \mu_1^{(0)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)})$
$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$	$(\sigma_0^{(1)}, \mu_0^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$	$(\sigma_1^{(1)}, \mu_1^{(1)}) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(1)})$
if $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**	**if** $\exists b \in \{0,1\}\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^{(1)}, \pi[0])$ **then**
return $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$	**return** $\mathsf{CointossS}^{(x_b^{(1)})}_{(2d)}(OTP \oplus \sigma_b^{(1)}, -)$
return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$	**return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
elseif $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**	**elseif** $\exists b\colon \textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}, \mu_b^0, (\sigma_b^{(0)}, x_{\mathsf{R}}^{(0)}))$ **then**
return $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$	**return** $\mathsf{CointossS}^{(x_b^{(0)})}_{(2d)}(OTP \oplus \sigma_b^{(0)}, -)$
fi	**fi**
return $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$	**return** $\mathsf{CointossS}^{(\pi[0])}_{(1/2)}(0,1)$
R:	R:
$OTP \xleftarrow{\$} \{0,1\}$	$OTP \xleftarrow{\$} \{0,1\}$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$	$x_{\mathsf{R}}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, OTP)$
$\mathsf{P_0}$:	$\mathsf{P_0}$:
–	$\mu_0^{(0)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, (1, x_{\mathsf{R}}^{(0)}))$
$x_0^{(0)} \xleftarrow{\$} \{0,1\}^m$	$x_0^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, 1 \| \mu)$
$\mathsf{P_1}$:	$\mathsf{P_1}$:
–	–
$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$	$x_1^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_0}$:	$\mathsf{P_0}$:
–	$\mu_0^{(1)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi[0])$
$x_0^{(1)} \leftarrow \{0,1\}^m$	$x_0^{(1)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, 1 \| \mu)$
$\mathsf{P_1}$:	$\mathsf{P_1}$:
–	–
$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$	$x_1^{(1)} \xleftarrow{\$} \{0,1\}^m$
return $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$	**return** $\mathcal{S}_{\mathcal{A}}^{P_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_1^\sigma(\kappa)$ This is the original game, in which party P_0 is the sender who transmits $\sigma = 0$.

$\mathrm{Game}_2^\sigma(\kappa)$ This game follows $\mathrm{Game}_1^\sigma(\kappa)$, but with the following changes:
During the simulation of the oracle P_{AT} from Fig. 10 the simulation enforces correctness of the challenge transcript $\pi_{\mathcal{C}}$ by initially checking if the input matches the challenge transcript and in that case, returning the output of a biased coin that returns $OTP \oplus \sigma_{\mathcal{C}}$ with probability $(1-(1-2d)^2/2)$ and $OTP \oplus \overline{\sigma_{\mathcal{C}}}$ with the complementary probability $((1-2d)^2/2)$. The randomness is taken over the entire transcript.
During simulation of the oracle P_{AT} from Fig. 10:

- if the input has a different *first-round* message of the *sender* or of the *receiver*, the simulated program outputs a uniformly random bit. The randomness is taken over all three zero-round messages, that is, over $x_0^{(0)}$, $x_1^{(0)}$ and $x_{\mathsf{R}}^{(0)}$.
- if the input shares the same first-round message of the sender but has a different first-round message of the receiver, the program outputs a uniformly random bit where the randomness is taken over all three first-round messages.
- if the input shares the same first-round message of the sender and receiver but has a different first-round message of the dummy-friend, the program outputs the correct bit $\sigma_{\mathcal{C}}$ with probability $1/2 + d$ and the wrong bit with the complementary probability.
- if the input shares the same first-round messages in the transcript, but a different second-round message than the sending party, the program outputs the correct bit $\sigma_{\mathcal{C}}$ with probability $1/2 + d$ and the wrong bit with the complementary probability.
- if the input is equivalent to the challenge transcript *except for the second dummy-friend message*, output the correct bit with probability $1-(1-2d)^2/2$.

Additionally, it reports uniformly random messages for the challenge transcript.

Lemma 37. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_1^\sigma(\kappa)$ *and* $\mathrm{Game}_2^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_1^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_2^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemmas 21 to 27. □

$\mathrm{Game}_3^\sigma(\kappa)$ This game follows $\mathrm{Game}_2^\sigma(\kappa)$, but changes the simulation of the circuit. Instead of using $\sigma_{\mathcal{C}} = 0$ for answering queries, this game uses $\sigma_{\mathcal{C}} := 1$.

Lemma 38. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_2^\sigma(\kappa)$ *and* $\mathrm{Game}_3^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_2^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_3^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Indistinguishability of these two games easily follows from the statistical security of One-Time-Pad encryption. Due to the messages being random the one-time-pad used to mask the bit is never shown directly to the distinguisher, and without the random tape it cannot be guessed. Hence in both games the distinguisher only ever gets a random bit. Indistinguishability follows. □

$\mathrm{Game}_4^\sigma(\kappa)$ This game is as $\mathrm{Game}_3^\sigma(\kappa)$ but undoes all the changes performed in $\mathrm{Game}_2^\sigma(\kappa)$.

Lemma 39. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_3^\sigma(\kappa)$ *and* $\mathrm{Game}_4^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_3^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_4^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemmas 21 to 27. □

Corollary 11 (Secrecy). *Let* Π_{AT}^1 *be instantiated as described in Fig. 9. No guessing algorithm* A *that works without knowledge of the receivers random tape* T_{R} *can extract the transferred bit using only the transcript and oracle-access to* P_{AT}*.*

Protocol $\Pi_{AT}^{1\,(c)}$ in the designated sender setting for $c > 2$ rounds.

We assume three parties are present in this protocol, two of which are participating. Let P_0 and P_1 be the potential sender, one of which is not participating, and let R denote the receiver.
Let $\textsc{Pke} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an *asymmetric* IND$-CCA-secure encryption scheme with Pseudo-random Ciphertexts.
Let $\textsc{Sig}$ be an sEUF-CMA-secure signature scheme.
Let P_{AT} be an obfuscated program from Fig. 10, which hides a verification key vk and a secret key sk_P, the corresponding public key pk_P is leaked.
The protocol $\Pi_{AT}^{1\,(c)}$ works as follows:

On input (b, σ, k), P_b enters (b, σ, k) into $\Pi_{AT}^{1\,(c-1)}$.
On input $(\bot)$, P_{1-b} enters $(\bot)$ into $\Pi_{AT}^{1\,(c-1)}$.
On input $(\bot)$, R does nothing.
On input $\pi^{(c-1)}$ from $\Pi_{AT}^{1\,(c-1)}$, P_b computes $\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}, \pi^{(c-1)})$ and sets $x_b^{(c)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, \sigma \| \mu)$.
P_{1-b} samples uniformly random $x_{1-b}^{(c)}$.
R outputs $OTP \oplus P_{AT}(x_{\mathsf{R}}^{(1)}, x_0^{(1)}, x_1^{(1)}, \ldots, x_0^{(c)}, x_1^{(c)})$.

Fig. 12: The designated sender protocol for Anonymous Transfer. We generally assume that once all messages of one round have been determined, parties publish them.

$P_{AT}[\mathsf{pk}_P]\left(\left(x_{\mathsf{R}}^{(1)}, x_0^{(1)}, x_1^{(1)}, \ldots, x_0^{(c)}, x_1^{(c)}\right)\right)$

$OTP \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_{\mathsf{R}}^{(1)})$,
$\left(\sigma_0^{(c)}, \mu_0\right) \coloneqq \textsc{Pke}.\mathsf{Dec}\left(\mathsf{sk}_P, x_0^{(c)}\right)$, $\left(\sigma_1^{(c)}, \mu_1\right) \coloneqq \textsc{Pke}.\mathsf{Dec}\left(\mathsf{sk}_P, x_1^{(c)}\right)$
if $\exists b \in \{0,1\}$: $\textsc{Sig}.\mathsf{Vfy}\left(\mathsf{vk}, \mu_b, \left(x_0^{(1)}, x_1^{(1)}, \ldots, x_0^{(c-1)}, x_1^{(c-1)}\right)\right)$ **then** :
  $coin \leftarrow \mathsf{Cointoss}_{(2d)}(x_b^{(c)})$
  if $coin = tails$ **then** :
    return $OTP \oplus \sigma_b^{(1)}$
return $P_{AT}\left(x_{\mathsf{R}}^{(1)}, x_0^{(1)}, x_1^{(1)}, \ldots, x_0^{(c-1)}, x_1^{(c-1)}\right)$

Fig. 13: Obfuscated program P_{AT} for the designated sender setting.

Corollary 12 (ς-Secrecy). *Π_{AT}^1 from Fig. 9 is ς-secret with $\varsigma \in (1 - \mathsf{negl}(\kappa))$.*

Proof. For $\ell = 1$ the formula from Eq. (20) is given as:

$$\left|\Pr\left[\begin{matrix}\pi \leftarrow \mathsf{Transfer}_{\langle \mathsf{R}, \mathsf{P}_0, \mathsf{P}_1\rangle}(crs, b, \sigma) \\ \sigma' \leftarrow \mathsf{A}(crs, \pi)\end{matrix}\right] - 1/2\right| \leq (1 - \varsigma)/2 \tag{44}$$

where the former probability was shown to be $\leq 1/2 + \mathsf{negl}(\kappa)$. Thus the equation becomes $\mathsf{negl}(\kappa) \leq (1 - \varsigma)/2 \implies \varsigma = 1 - \mathsf{negl}(\kappa)$ where we use that $2 \cdot \mathsf{negl}(\kappa) \in \mathsf{negl}(\kappa)$. □

Corollary 13 (Security). *Π_{AT}^1 is a $(4d(1-d), 1-2d, 1-\mathsf{negl}(\kappa))$-AT.*

E.2 A c-rounds protocol.

We now provide a recursive definition on how to extend the two-rounds protocol from Fig. 9 to a general c-rounds protocol.

The protocol is defined for any $c > 2$ and works as follows: In each round χ, both messages are decrypted. If either of them contains a valid signature on the transcript *up to round* $\chi - 1$, the program returns the bit encoded in the same message with probability $2d$. We require the signature to be on the *full* transcript, since otherwise, an adversary can learn if this round has been considered by the circuit for the transfer, by replacing both messages in round $\chi \in [1, c]$. The program recursively returns its own output on the same input *except for the last round*, if the coin lands on heads (*i.e.* with probability $(1 - 2d)$), or if the last-round messages do not contain a valid signature.

Security Analysis.

Correctness. We will now analyze the correctness of the new protocol:

Lemma 40 (Correctness). *Let Π^1_{AT} be given as described in Fig. 12. Let $c \in \mathbb{N} > 1$ be the number of rounds. Π^1_{AT} returns the correct bit with probability $1-(1-2d)^c/2$, and hence $\varepsilon = 1-(1-2d)^c/2$.*

Proof. We perform this proof by mathematical induction.

Base: $c = 2$. Here our claim states that σ is returned with probability $1 - \frac{(1-2d)^2}{2}$, which was shown in Lemma 19 (see Eq. (39)).

Inductive Step: If ${\Pi^1_{AT}}^{(c)}$ can transfer the bit σ with correctness $1-(1-2d)^c/2$, then ${\Pi^1_{AT}}^{(c+1)}$ can transfer the bit σ with correctness $1-(1-2d)^{c+1}/2$. On honest inputs, only the signature of party P_b verifies, the random input of P_{1-b} is a valid signature only with negligible probability, which we drop here. The protocol ${\Pi^1_{AT}}^{(c+1)}$ returns the bit with the following probability:

$$\begin{aligned}
&\Big(2d \cdot 1 + (1-2d) \cdot \Pr[{\Pi^1_{AT}}^{c} \text{ returns correct bit}]\Big) \\
=&\left(2d + (1-2d) \cdot \left(1 - \frac{(1-2d)^c}{2}\right)\right) \\
=&\left(2d + (1-2d) - \frac{(1-2d)\cdot(1-2d)^c}{2}\right) \\
=&\left(1 - \frac{(1-2d)^{c+1}}{2}\right)
\end{aligned} \tag{45}$$

Conclusion: The claim is true.

Plugging this into the Definition 4 for ε-correctness yields:

$$\varepsilon = 1 - (1-2d)^c/2 \tag{46}$$

□

Anonymity. We now also analyze our new protocol with respect to anonymity. But we first prove two lemmas we'll need later.

Lemma 41 (Changing the Senders Message). *Let ${\Pi^1_{AT}}^{(c)}$ be given as in Fig. 12. Let $c \in \mathbb{N} > 1$ be the number of rounds. Changing the sending parties message in round $\chi \in [1, c]$ yields the correct output with probability $1 - ((1-2d)^{\chi-1})/2$*

Proof. This proof follows from the correctness properties combined with the signatures on all previous messages: A changed sending parties message in round χ causes all signatures from rounds $\chi + 1$ to c to fail to verify, and in itself has a valid signature only with negligible probability; as such, the output is the same if two random messages are used for each future round.

Hence, our claim follows from the *correctness* property of Lemma 40, since this protocol is now equivalent to a valid $c - 1$-round protocol. □

Lemma 42 (Changing the Dummy Friends Message). *Let ${\Pi^1_{AT}}^{(c)}$ be given as in Fig. 12. Let $c \in \mathbb{N} > 1$ be the number of rounds. Changing the dummy friends message in round $\chi \in [1, c]$ yields the correct output with probability $1 - ((1-2d)^{\chi})/2$*

Proof. Likewise, this follows from the correctness property. A changed dummy friend message in round χ causes the sending parties signature to fail from round $\chi+1$ onward. Due to the recursion, all messages from rounds c to message $\chi+1$ are ignored, as they have an invalid signature. The output of the χ-message protocol is returned, which is correct with probability $1-((1-2d)^{\chi})/2$, according to Lemma 40. □

Lemmas 41 and 42 imply that most of the steps from Lemma 36 can be canonically extended from two to c rounds by introducing new *hybrid games*. The only game that requires further investigation is Lemma 28. In fact, the following diff is added to the games:

$\text{Game}_1^{\sigma}(\kappa)$ This game follows the protocol execution from Fig. 12 where the obfuscated circuit is replaced with oracle-access to the circuit and the PRF within the circuit is now a random oracle.

$\text{Game}_2^{\sigma}(\kappa)$ When the correct transcript is input, the oracle returns the correct bit with probability $(1-(1-2d)^c/2)$, simulating correctness according to Lemma 40. The randomness is taken over the whole transcript.

$\text{Game}_3^{\sigma}(\kappa)$ **to** $\text{Game}_7^{\sigma}(\kappa)$ Those games hard-code the one-shot behavior for different input messages. We replace these four game by a series of $2c$ games, which alternate between hard-coding behaviors for the sending party and for the dummy friend in an increasing order. More precisely, for each round $\chi \in [2c]$, there are 2 consecutive games $2\chi-1$ and 2χ, where the $(2\chi-1)$-th game adds a new check regarding the sending parties input in round χ, and returns the correct bit with probability $(1-\frac{(1-2d)^{\chi-1}}{2})$, and the (2χ)-th game adds a check regarding the dummy-friends input in round χ, where the correct bit is returned with probability $(1-\frac{(1-2d)^{\chi}}{2})$. Indistinguishability of the $(2\chi-1)$-th new game follows from Lemma 41, indistinguishability of the (2χ)-th new game is implied by Lemma 42.

$\text{Game}_8^{\sigma}(\kappa)$ This is adjusted canonically to all c rounds. By exploiting the LR-view used in the proof, no further adjustments are required.

The adjustment of the inverse games, $\text{Game}_{10}^{\sigma}(\kappa)$ to $\text{Game}_{16}^{\sigma}(\kappa)$, is analogous to the above. The one missing game is $\text{Game}_9^{\sigma}(\kappa)$, which requires a more rigorous analysis.

Lemma 43 (Adapting $\text{Game}_9^{\sigma}(\kappa)$ to c rounds). *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for a changed sender after adjusting $\text{Game}_8^{\sigma}(\kappa)$ to c rounds is bounded by:*

$$|\Pr[\mathsf{out}_{\mathsf{P}_0}=1]-\Pr[\mathsf{out}_{\mathsf{P}_1}=1]| \leq \frac{2d+1-(1-2d)^c}{2}$$

Proof. We use the same basic technique as in Lemma 28, in that we discretize the set of possible worlds based on the coin-toss outcomes of an honestly created, fixed transcript.

We claim that the view provided in Fig. 14 implies our claim. We once again use mathematical induction.

Base: $c=2$. Here our claim states that the change is detected with probability $\frac{2d+1-(1-2d)^2}{2} = d(3-2d)$, with the tree from Fig. 14 corresponding to Fig. 11. This was proven in Lemma 28.

Inductive Step: This step contains two parts. First, we show that, assuming that the view from Fig. 14 correctly represents the different worlds alongside their anonymity- and occurrence-guarantees, an additional round only adds one more relevant world to the one edge where the initial coin-toss yields $\bar{\sigma_C}$ and all other coin-tosses land on heads. We then show that, given the view from Fig. 14, the probability of successfully distinguishing the change induced by this game hop is $\frac{2d+1-(1-2d)^c}{2}$.

We start by the former, namely showing that our tree-expansion mechanism can be canonically expanded from c to $(c+1)$ rounds. For that, we refer to the following worlds:

$World_1$ is as $World_2$ in the two-round setting. Here, there already is an attack that de-anonymizes the sender, hence allows detection of the game hop. Hence a new round brings absolutely no advantage.

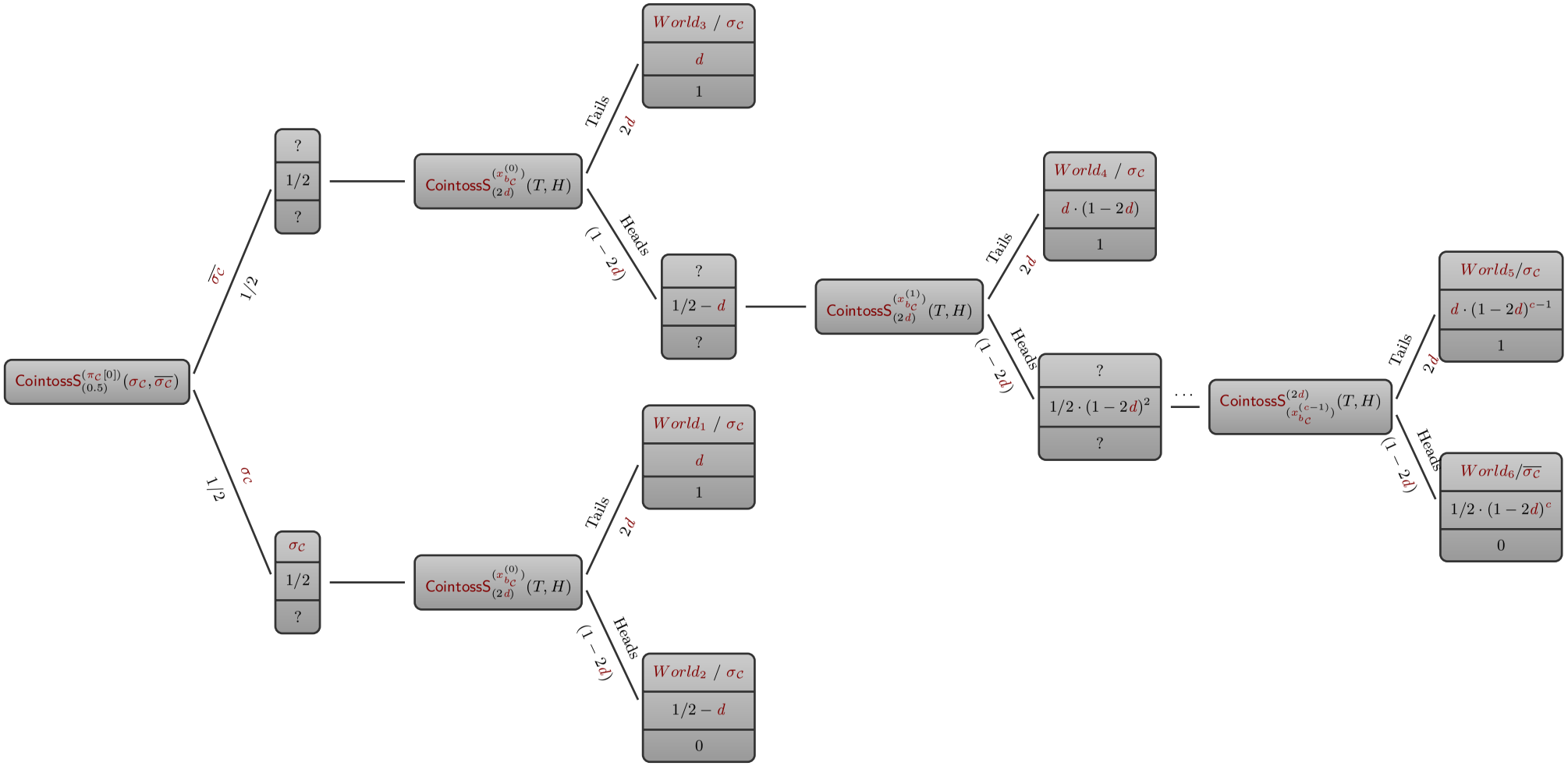


Fig. 14: Tree showcasing the different worlds, based on the randomness from a uniformly random challenge transcript.

$World_2$ is as $World_1$ in the two-round setting. We refer to this world as optimal in the following sense: Every new input of any first-round message yields a fresh random output, but every new input of any other message—regardless from which party—yields the correct bit $\sigma_{\mathcal{C}}$. Hence, here it automatically follows that no new round induces any changes, as the result does not change regardless of the output of the new coin-toss.

$World_3$ is as $World_3$ in the two-round setting. As we have already shown in Lemma 28 that there is an attack on the first round here, which works regardless of the number of rounds.

$World_4$ represents all worlds that have a sequence where the first coin-toss lands on $\bar{\sigma_{\mathcal{C}}}$, followed by $(\chi - 1)$ rounds of coin-tosses that land on heads and have a coin-toss in round χ that ends with tails. We stress that the same attack that was possible on round one of $World_4$ in Lemma 28 also works here, and that the respective probability of being in either of these worlds is given by $1/2 \cdot (1-2d)^{\chi-1} \cdot d$. Whatever follows is irrelevant, as the attack already works on round χ.

$World_5$ is the first of the two relevant new worlds that come up when we add a new round: We only are in this world, if the previous transcript has been such that all coin-tosses landed on heads, and the initial coin-toss yields the wrong bit $\bar{\sigma_{\mathcal{C}}}$. If the new round yields a coin-toss that lands on tail in the new round only, then this final round is susceptible to the same attack that was used in $World_4$ on trying new inputs for the final round. Any new sending parties message yields $\bar{\sigma_{\mathcal{C}}}$, but the same transcript with any new dummy-friend message causes the correct bit to be output, making it trivial to detect the change.

$World_6$ is the second and final of the two new relevant worlds. In this world, the game change is impossible to detect, as all coin-tosses in the challenge transcript land on heads, and all sending parties messages are ignored just like the dummy friends inputs; the output bit is determined by the coin-toss on both first-round messages, which in these worlds yield $\bar{\sigma_{\mathcal{C}}}$. While the output is wrong, we stress that the anonymity in here is perfectly, meaning that we get an advantage of 0.

Knowing that the world-view from Fig. 14 is consistent with the c rounds protocol, we can now put a bound on the success probability of the adversary. Intuitively, the adversary can perfectly detect the game hop, unless we are in $World_2$ or $World_6$, where detection is (even statistically) impossible. Hence, the probability of an adversary to successfully distinguish the two games is given by

$$\begin{aligned} & 1 - \Pr[World_2] - \Pr[World_6] \\ =& 1 - \left(\frac{1-2d}{2}\right) - \left(\frac{1}{2} \cdot (1-2d)^c\right) \\ =& \frac{2 - 1 + 2d - (1-2d)^c}{2} \\ =& \frac{1 + 2d - (1-2d)^c}{2} \end{aligned} \tag{47}$$

Conclusion: The claim is true for any c round protocol.

□

Thus, we have shown that all games for Lemma 36 can be canonically adjusted to $c > 2$ rounds, with the distinguishing advantage between the transcript where P_0 is the sending party and P_1 being the sending party being limited by $\frac{1+2d-(1-2d)^c}{2}$.

We now analyze what this means for our overall anonymity guarantee.

Lemma 44 (Anonymity of the c-round protocol). *The protocol from Fig. 12 is δ-anonymous with $\delta \leq 1 - |2d - (1-2d)^c|$.*

Proof. Recall from Eq. (4) that the anonymity δ is the maximum δ for which it holds that

$$\left| \Pr_{b \xleftarrow{\$} \{0,1\}} \left[\mathrm{Exp}_{\Pi^1_{AT}, \mathcal{A}, b}(\kappa) = b \right] - 1/2 \right| \leq (1-\delta)/2 \tag{48}$$

and that the former probability has been shown to be limited by $\frac{1+2d-(1-2d)^c}{2}$. Hence, we get that

$$\begin{aligned}\delta &= \max_{\delta'} \left| \frac{1+2d-(1-2d)^c}{2} - \frac{1}{2} \right| \leq \frac{1-\delta'}{2} \\ &= \max_{\delta'} \left| \frac{2d-(1-2d)^c}{2} \right| \leq \frac{1-\delta'}{2}\end{aligned} \tag{49}$$

This implies that $\delta \leq 1 - |2d - (1-2d)^c|$, which proves our claim. □

We now focus on the secrecy of the protocol. Note that the output is still encrypted by a OTP and hence the additional rounds do not change the fact that the actually transferred bit can not be extracted by any party other than the receiver.

We thus get

Lemma 45. *The protocol from Fig. 12 is ς-secret with $\varsigma \in (1 - \mathsf{negl}(\kappa))$.*

Thus, by combining Lemmas 40, 44 and 45, we obtain:

Corollary 14 (Security of the c-round protocol). *$\Pi_{AT}^{1\,(c)}$ from Fig. 12 is a $((1-(1-2d)^c), (1-|2d-(1-2d)^c|), (1-\mathsf{negl}(\kappa)))$-AT.*

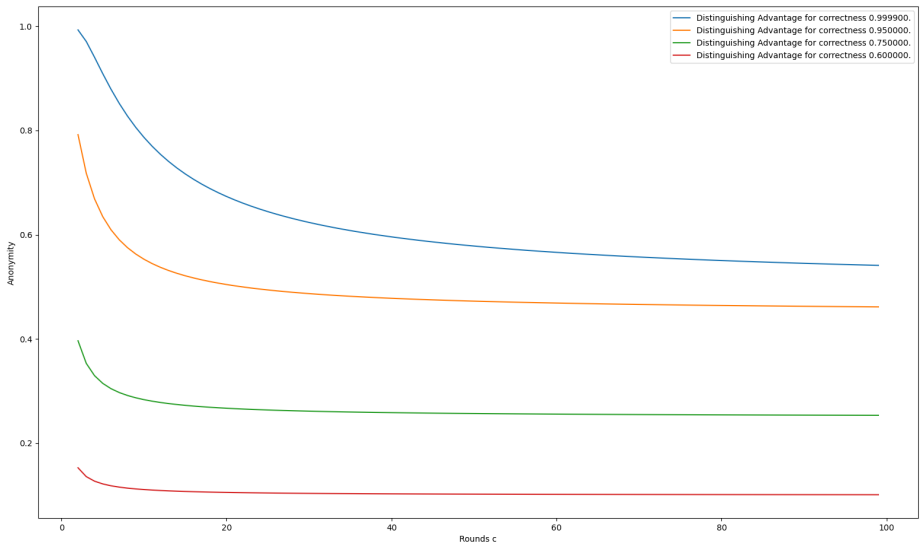
E.3 On the security

For a practical instantiation, we assume that the parameter d is chosen subject to a given correctness guarantee and then chosen according to the formula from Lemma 40. For example, for a 10-round AT which is correct with 90% probability, we have to solve $1-(1-2d)^{10}/2 = 0.9$, resulting in $d \approx 0.07433$. This would then yield a $(0.9, 0.94)$-AT, where the adversary is able to determine the sending party with a probability of 0.47433.

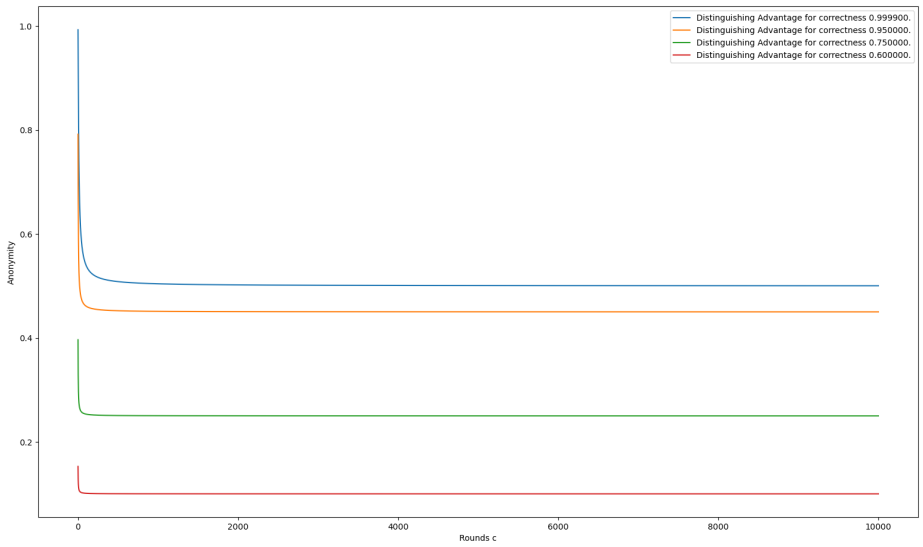
In Fig. 15, we show the trade-of between correctness and anonymity for a correctness of 0.9999, 0.9, 0.75 and 0.6, respectively.

F A fine-grained Anonymous Transfer for ℓ-bit messages

Instead of using the limited composability guarantees established in Lemma 5 to extend our fine-grained AT from transferring a single bit towards transforming an ℓ-bit string we perform some slight modifications of the protocol introduced in Section 5 to see how it can handle the transfer of ℓ-bit messages directly.



(a) The relation between correctness and anonymity for up to 100-round protocols.



(b) The relation between correctness and anonymity for up to 10.000-round protocols.

Fig. 15: Two images showing the anonymity obtained for a given correctness requirement. The x-axis showcases the number of rounds for the protocol, which for clarity we scaled to 100 and 10.000. The y-axis shows the anonymity guarantee, *i.e.* the maximum advantage of an adversary in finding the sending party.

Fine-grained Protocol Π^{ℓ}_{AT} for ℓ-bit messages $\Sigma \in \{0,1\}^{\ell}$.

We assume three parties are present in this protocol, two of which are participating. Let P_0 and P_1 be the potential sender, one of which is not participating, and let R denote the receiver.
Let $\textsc{Pke} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a IND\$-CCA-secure *public-key* encryption scheme.
Let $\textsc{Ske} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a *multi-challenge* IND\$-CPA-secure *symmetric* encryption scheme.
Let $\textsc{Sig}\colon \{0,1\}^* \times \{0,1\}^{\kappa} \mapsto \{0,1\}^{m}$ be a EUF-CMA-secure signature scheme.
Let P_{AT} be an obfuscated program from Fig. 17.
The protocol Π^{ℓ}_{AT} works as follows:

Upon activation , R draws a random $OTP \xleftarrow{\$} \{0,1\}^{\ell}$ and computes $(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^{\kappa})$. Then R sets $x^{(0)}_{\mathsf{R}} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$ and broadcasts $x^{(0)}_{\mathsf{R}}$.
On input (b, Σ), P_b computes a signature key pair $(\mathsf{vk}_b, \mathsf{k}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^{\kappa})$ and a symmetric key $\mathsf{sk}_b \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^{\kappa})$.
Then, P_b computes a signature $\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x^{(0)}_{\mathsf{R}})$ and broadcasts $x^{(0)}_b \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, \Sigma)$.
Upon activation , P_{1-b} sets uniformly random $x^{(0)}_{\overline{b}}$.
For each round χ **from** 1 **to** c :
P_b computes $\mu \leftarrow \textsc{Sig}.\mathsf{Sig}\left(\mathsf{k}_b, \left(x^{(\chi-1)}_0, x^{(\chi-1)}_1\right)\right)$ and sets $x^{(\chi)}_b \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (\Sigma, \mu))$.
P_{1-b}**:** Broadcast $x^{(\chi)}_{1-b} \xleftarrow{\$} \{0,1\}^{m}$.
R**:** computes $\mu \leftarrow \textsc{Sig}.\mathsf{Sig}\left(\mathsf{k}_{\mathsf{R}}, \left(x^{(0)}_{\mathsf{R}}, \left(x^{(0)}_0, x^{(0)}_1\right), \ldots, \left(x^{(c)}_0, x^{(c)}_1\right)\right)\right)$, compute $\Sigma' := P_{AT}\left(x^{(0)}_{\mathsf{R}}, \left(x^{(0)}_0, x^{(0)}_1\right), \ldots, \left(x^{(c)}_0, x^{(c)}_1\right), \mu\right)$ and output $OTP \oplus \Sigma'$.

Fig. 16: A protocol for fine-grained ℓ-bit Anonymous Transfer.

$P_{AT}[\mathsf{pk}_P, c]\left(x^{(0)}_{\mathsf{R}}, \left(x^{(0)}_0, x^{(0)}_1\right), \left(x^{(1)}_0, x^{(1)}_1\right) \ldots, \left(x^{(c)}_0, x^{(c)}_1\right)\right)$

$(OTP, \mathsf{vk}_{\mathsf{R}}) := \textsc{Pke}.\mathsf{Dec}^*\left(\mathsf{sk}_P, x^{(0)}_{\mathsf{R}}\right)$,

$(\mathsf{sk}_0, \mathsf{vk}_0) := \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x^{(0)}_0[1\colon m])$, $(\Sigma_0, \mu_0) := \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_0, x^{(0)}_0[m+1\colon 2m])$,

$(\mathsf{sk}_1, \mathsf{vk}_1) := \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x^{(0)}_1[1\colon m])$, $(\Sigma_1, \mu_1) := \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_1, x^{(0)}_1[m+1\colon 2m])$,

if $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_{\mathsf{R}}, (x^{(0)}_{\mathsf{R}}, (x^{(0)}_0, x^{(0)}_1), \ldots, (x^{(0)}_0, x^{(c)}_1)))$ **then** :
  return $\perp$

$\chi_0 := [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0, \mathsf{vk}_0, x^{(0)}_{\mathsf{R}})]\!] \cdot (c+1)$, $\chi_1 := [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1, \mathsf{vk}_1, x^{(1)}_{\mathsf{R}})]\!] \cdot (c+1)$,

foreach $\chi \in \{1, \ldots, c\}$ **do**:
  foreach $b \in \{b' | b' \in \{0,1\}, \chi_b = (c+1)\}$ **do**: // Take on the role of each potential sender.
    $X_b := \textsc{Ske}.\mathsf{Dec}^*\left(\mathsf{sk}_b, x^{(\chi)}_b\right)$, $\Sigma'_b := X_b[0\colon (m-1)]$, $\mu_b := X_b[m\colon |X_b|]$
    if $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b, \mathsf{vk}_b, \pi[\chi-1]) \vee \Sigma_b \neq \Sigma'_b$ **then** :
      $\chi_b := \chi$ // Remember first bad round.

$b' := \mathbf{argmax}_b(\chi_b)$

return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{((\chi_{b'}/c))}(\Sigma_{b'}, \perp)$

Fig. 17: Obfuscated program P_{AT} for the fine-grained setting. $c \in \mathsf{poly}(\kappa)$ denotes the number of rounds.

The protocol is given in Fig. 16, the corresponding obfuscated program in Fig. 17. The protocol in Fig. 16 is essentially the same as the one in Fig. 5 for single-bit AT, where the bit $\sigma \in \{0,1\}$ was replaced by a message $\Sigma \in \{0,1\}^\ell$ and the circuit is now the one from Fig. 17 instead of the one from Fig. 6. The same is true for the circuit Fig. 17, which is the one from Fig. 6 but with a message Σ instead of a bit σ but one additional thing had to be changed: Instead of returning the actual bit σ with probability proportional to the amount of correct rounds of the sender P_b and *the complementary bit* with the remaining probability the new circuit from Fig. 17 separates between returning either the message Σ_b or a dedicated *error state* $\perp$. The actual message Σ_b is implicitly defined by the respective first-round messages. Accordingly, the probability does not start at 50% between 0 and 1 but starts at returning $\perp$ with probability 100% and each new round moves this towards returning Σ_b with probability 100%.

We start by analyzing the *correctness* of the new protocol.

Lemma 46 (Correctness). *If the protocol from Fig. 16 is instantiated with an Ideally Obfuscated version of the circuit from Fig. 17 the protocol is ε-correct with $\varepsilon = (1 - \mathsf{negl}(\kappa))$.*

Proof. Again, in an honest execution the value of $\chi_b = c$ and hence the correct message is returned with probability 1, the argumentation here is similar to that of Theorem 4. We only have to assume that there is a dedicated failure-state $\perp$ differs from any message a party would ever send. □

Once again, the anonymity takes a lot more effort to analyze. However, the proof is still relatively close to that of single-bit AT leading to the anonymity in Corollary 7.

$\mathrm{Game}_1(\kappa)$

1 : –
2 : –
3 : –
4 : –
5 : –
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : $(OTP, \mathsf{vk}_\mathsf{R}) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_\mathsf{R}^{(0)})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\Sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\Sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_\mathsf{R}, \pi)$ **then**
20 : **return** $\bot$
21 : $\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\Sigma_b' \coloneqq X_b[0\colon \ell - 1]$
27 : $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
28 : **if** $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \Sigma_b \neq \Sigma_b'$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}^\ell$
3 : $(\mathsf{k}_\mathsf{R}, \mathsf{vk}_\mathsf{R}) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
4 : $x_\mathsf{R}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_\mathsf{R}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
7 : $(\mathsf{sk}_b) \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^\kappa)$
8 : $(\mu_b^{(0)}) \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x_\mathsf{R}^{(0)})$
9 : $x_b^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : –
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19 : R:
20 : $\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_\mathsf{R}, \pi)$
21 : **return** $\mathcal{S}_\mathcal{A}^{P'^{AT}}(\pi, T_\mathsf{R})$

$\mathrm{Game}_2(\kappa)$

1 : **if** $\pi = \pi_\mathcal{C}$ **then**
2 : **return** $OTP_\mathcal{C} \oplus \Sigma_\mathcal{C}$
3 : –
4 : –
5 : –
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : –
14 : $(OTP, \mathsf{vk}_\mathsf{R}) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_\mathsf{R}^{(0)})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\Sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\Sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_\mathsf{R}, \pi)$ **then**
20 : **return** $\bot$
21 : $\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\Sigma_b' \coloneqq X_b[0\colon \ell - 1]$
27 : $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
28 : **if** $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \Sigma_b \neq \Sigma_b'$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}^\ell$
3 : $(\mathsf{k}_\mathsf{R}, \mathsf{vk}_\mathsf{R}) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
4 : $x_\mathsf{R}^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_\mathsf{R}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
7 : $(\mathsf{sk}_b) \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^\kappa)$
8 : $(\mu_b^{(0)}) \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x_\mathsf{R}^{(0)})$
9 : $x_b^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : –
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
19 : R:
20 : $\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_\mathsf{R}, \pi)$
21 : **return** $\mathcal{S}_\mathcal{A}^{P'^{AT}}(\pi, T_\mathsf{R})$

$\mathrm{Game}_2(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
–
–
–
–
–
–
–
–
–
–
–
$(OTP, \mathsf{vk}_R) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_R^{(0)})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_R, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_R^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_R^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^\ell$
$(\mathsf{k}_R, \mathsf{vk}_R) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$x_R^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_R))$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$(\mathsf{sk}_b) \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^\kappa)$
$(\mu_b^{(0)}) \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x_R^{(0)})$
$x_b^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $\mu^{(\chi)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_R, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_R)$

$\mathrm{Game}_3(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, \pi[0]) = \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
–
–
–
–
–
–
–
–
–
–
–
$(OTP, \mathsf{vk}_R) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_R^{(0)})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \textsc{Ske}.\mathsf{Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_R, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_R^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_R^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^\ell$
$(\mathsf{k}_R, \mathsf{vk}_R) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$x_R^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_R))$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \textsc{Sig}.\mathsf{KeyGen}(1^\kappa)$
$(\mathsf{sk}_b) \leftarrow \textsc{Ske}.\mathsf{KeyGen}(1^\kappa)$
$(\mu_b^{(0)}) \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, x_R^{(0)})$
$x_b^{(0)} \leftarrow \textsc{Pke}.\mathsf{Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $\mu^{(\chi)} \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \textsc{Ske}.\mathsf{Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \textsc{Sig}.\mathsf{Sig}(\mathsf{k}_R, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_R)$

$\mathrm{Game}_3(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
  $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
–
–
–
–
–
–
–
–
$(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq \mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^{\kappa})$
$(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
  $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
–
–
–
–
–
–
–
–
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^{\kappa})$
$(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

1 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
2 : **return** $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
3 : **elseif** $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
4 : $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
5 : **abort**
6 : –
7 : –
8 : –
9 : –
10 : –
11 : –
12 : –
13 : **if** $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
14 : $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
20 : **return** $\bot$
21 : $\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\Sigma_b' \coloneqq X_b[0\colon \ell - 1]$
27 : $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
28 : **if** $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma_b'$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}^{\ell}$
3 : $(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
4 : $x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
7 : $(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^{\kappa})$
8 : $(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
9 : $x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : –
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^{m}$
19 : R:
20 : $\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
21 : **return** $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_5(\kappa)$

1 : **if** $\pi = \pi_{\mathcal{C}}$ **then**
2 : **return** $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
3 : **elseif** $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
4 : $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
5 : **abort**
6 : **elseif** $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
7 : $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
8 : **if** $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}}, \mu_{\mathcal{C}}{}^{(\chi^*+1)}) \vee$
9 : $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}]$ **then**
10 : $\chi^* \coloneqq \chi^* + 1$
11 : $p \coloneqq 1/2 + \chi^*/2c$
12 : **return** $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\Sigma_{\mathcal{C}}, \bot)$
13 : **if** $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
14 : $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
15 : $(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
16 : $(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
17 : $(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
18 : $(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
19 : **if** $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
20 : **return** $\bot$
21 : $\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
22 : $\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
23 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
24 : **foreach** $b \in \{0,1\} | \chi_b > c$ **do**
25 : $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
26 : $\Sigma_b' \coloneqq X_b[0\colon \ell - 1]$
27 : $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
28 : **if** $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma_b'$ **then**
29 : $\chi_b \coloneqq \chi$
30 : $b' \coloneqq \mathbf{argmax}_b(\chi_b)$
31 : **return** $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$
32 :

1 : R:
2 : $OTP \xleftarrow{\$} \{0,1\}^{\ell}$
3 : $(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
4 : $x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
5 : $\mathsf{P_b}$:
6 : $(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
7 : $(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^{\kappa})$
8 : $(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
9 : $x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
10 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
11 : $x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
12 : **foreach** $\chi \in \{1, \ldots, c\}$ **do**
13 : $\mathsf{P_b}$:
14 : $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
15 : $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
16 : $\mathsf{P}_{\overline{\mathsf{b}}}$:
17 : –
18 : $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^{m}$
19 : R:
20 : $\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
21 : **return** $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_5(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ then
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ then
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ then
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}}, {\mu_{\mathcal{C}}}^{(\chi^*+1)}) \vee$
    $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}]$ then
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}_{(p)}^{(\pi)}(\Sigma_{\mathcal{C}}, \perp)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ then
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ then
  return $\perp$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ do
  foreach $b \in \{0,1\} | \chi_b > c$ do
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \Sigma_b \neq \Sigma'_b$ then
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}_{(\chi_b/c)}^{(\pi)}(\Sigma^{(b')}, \perp)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^{\kappa})$
$(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ do
  $\mathsf{P_b}$:
  $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_6(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ then
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ then
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ then
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}}, {\mu_{\mathcal{C}}}^{(\chi^*+1)}) \vee$
    $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}]$ then
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}_{(p)}^{(\pi)}(\Sigma_{\mathcal{C}}, \perp)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ then
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ then
  return $\perp$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ do
  foreach $b \in \{0,1\} | \chi_b > c$ do
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \Sigma_b \neq \Sigma'_b$ then
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}_{(\chi_b/c)}^{(\pi)}(\Sigma^{(b')}, \perp)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^{\kappa})$
$(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^m \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ do
  $\mathsf{P_b}$:
  $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_6(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}}, {\mu_{\mathcal{C}}}^{(\chi^*+1)}) \vee$
    $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}]$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}_{(p)}^{(\pi)}(\Sigma_{\mathcal{C}}, \bot)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}_{(\chi_b/c)}^{(\pi)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^{\kappa})$
$(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^m \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $\mu^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \Sigma \| \mu^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_7(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}}, {\mu_{\mathcal{C}}}^{(\chi^*+1)}) \vee$
    $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}]$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}_{(p)}^{(\pi)}(\Sigma_{\mathcal{C}}, \bot)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}_{(\chi_b/c)}^{(\pi)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_b}$:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_7(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}}, {\mu_{\mathcal{C}}}^{(\chi^*+1)}) \vee$
    $\pi[x_{b_{\mathcal{C}}}^{\chi^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}]$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}_{(p)}^{(\pi)}(\Sigma_{\mathcal{C}}, \bot)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}_{(\chi_b/c)}^{(\pi)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
–
–
–
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  –
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_8(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  –
  –
  –
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}_{(p)}^{(\pi)}(\Sigma_{\mathcal{C}}, \bot)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma_0, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\Sigma_1, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}_{(\chi_b/c)}^{(\pi)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
–
–
–
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  –
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  –
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\text{Game}_1^\sigma(\kappa)$ This is the original game (after replacing the obfuscated circuit with oracle access to the circuit, the PRF with an actual random oracle and the adversary with the simulator), where the sending party P_b is chosen uniformly at random.

$\text{Game}_2^\sigma(\kappa)$ This game follows $\text{Game}_1^\sigma(\kappa)$, but during the simulation of the oracle P_{AT} from Fig. 16 the simulation enforces correctness of the challenge transcript $\pi_\mathcal{C}$: if the input transcript π matches the challenge transcript $\pi_\mathcal{C}$, it returns $\Sigma_\mathcal{C}$.

Lemma 47 (Indistinguishability of $\text{Game}_1^\sigma(\kappa)$ and $\text{Game}_2^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme. Let SKE be an IND$-CCA secure symmetric encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\text{Game}_1^\sigma(\kappa)$ and $\text{Game}_2^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_1^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_2^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 46. □

$\text{Game}_3^\sigma(\kappa)$ This game follows $\text{Game}_2^\sigma(\kappa)$ but during simulation of the circuit the adversary aborts if any of the first-round messages differ from the messages reported in the challenge transcript *and* the decryptions still match.

Lemma 48 (Indistinguishability of $\text{Game}_2^\sigma(\kappa)$ and $\text{Game}_3^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme. Let SKE be an IND$-CCA secure symmetric encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\text{Game}_2^\sigma(\kappa)$ and $\text{Game}_3^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_2^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_3^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. The claim is similar to that from Lemma 10 and thus follows from their proof. □

$\text{Game}_4^\sigma(\kappa)$ This game follows $\text{Game}_3^\sigma(\kappa)$ but simulates the circuit slightly different: If the first receiver message of the input transcript π is the same as that of the challenge transcript $\pi_\mathcal{C}$, instead of decrypting it the circuit directly sets $OTP = OTP_\mathcal{C}$ and $\mathsf{vk}_\mathsf{R} = \mathsf{vk}_{\mathcal{C}\mathsf{R}}$ as the values used in the creation of the challenge transcript.

Lemma 49 (Indistinguishability of $\text{Game}_3^\sigma(\kappa)$ and $\text{Game}_4^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme. Let SKE be an IND$-CCA secure symmetric encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\text{Game}_3^\sigma(\kappa)$ and $\text{Game}_4^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_3^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_4^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. This proof also follows from the correctness of the encryption scheme PKE just as Lemma 11. □

$\text{Game}_5^\sigma(\kappa)$ This game follows $\text{Game}_4^\sigma(\kappa)$ but simulates the oracle differently if the first-round messages of both parties match the first-round messages in the challenge transcript $\pi_\mathcal{C}$. In this case, the program compares the input transcript π with the challenge transcript $\pi_\mathcal{C}$ until it finds the first round χ^* in which the input differs from the challenge transcript. It then checks round $\chi^* + 1$, and if it contains the same message from the sending party, it adds one to χ.
Finally, the circuit flips a biased coin, which returns the correct message $\Sigma_\mathcal{C}$ with probability $p := 1/2 + \chi^*/2c$ and an error symbol $\perp$ otherwise.

Lemma 50 (Indistinguishability of $\text{Game}_4^\sigma(\kappa)$ and $\text{Game}_5^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme. Let SKE be an IND$-CCA secure symmetric encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\text{Game}_4^\sigma(\kappa)$ and $\text{Game}_5^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_4^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_5^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Note that the unforgeability of a valid round-zero message required for detecting the new branch is similar to Lemma 12. Thus we can focus entirely on the second claim, namely that the behavior inside the branch reflects that of the actual circuit.
In fact, we have to show that at line 31 in $\text{Game}_2^\sigma(\kappa)$ if the maximum round is χ_b then the input transcript is equal to the challenge transcript until *at least* round $(\chi_b - 1)$.
Let $\mathcal{D}$ be a PPT distinguisher who can create input π for which it holds that χ^* is such that any of the messages from round $(\chi^* - 2)$ differ from the challenge transcript. Then the only way that the verification in round $(\chi^* - 1)$ succeeds is if the sending parties message encodes a valid signature on the changed message from round $(\chi^* - 2)$, which would violate the EUF-CMA security of the used signature scheme SIG.
Thus the claim follows. □

$\text{Game}_6^\sigma(\kappa)$ This game is the same as $\text{Game}_5^\sigma(\kappa)$, but in creating the challenge transcript $\pi_\mathcal{C}$, this game only reports randomness for the first-round message $\widehat{x_b^0}$ that specifies the symmetric key sk_b to be used for the remaining communication with the circuit. Note that both the signature- and the symmetric-key are still created for the sending party P_b as they are needed for the remaining rounds.

Lemma 51 (Indistinguishability of $\text{Game}_5^\sigma(\kappa)$ and $\text{Game}_6^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme. Let SKE be an IND$-CCA secure symmetric encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\text{Game}_5^\sigma(\kappa)$ and $\text{Game}_6^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_5^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_6^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 13. □

$\text{Game}_7^\sigma(\kappa)$ This game is the same as $\text{Game}_6^\sigma(\kappa)$ but in creating the challenge transcript $\pi_\mathcal{C}$, this game also reports randomness instead of transcripts for all messages x_b^χ for $\chi \in [c]$ that shift the message towards $\Sigma_\mathcal{C}$. That means that instead of using the IND$-CPA secure symmetric scheme SKE with the symmetric key sk_b the challenge transcript now only contains randomly sampled messages.
We also do not let the adversary create the keys for SIG and SKE as they are no longer needed for creating the transcript.

Lemma 52 (Indistinguishability of $\text{Game}_6^\sigma(\kappa)$ and $\text{Game}_7^\sigma(\kappa)$). *Let PKE be an IND$-CCA secure asymmetric encryption scheme. Let SKE be an IND$-CCA secure symmetric encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms A, the distinguishing advantage for $\text{Game}_6^\sigma(\kappa)$ and $\text{Game}_7^\sigma(\kappa)$ is bounded by:*

$$|\Pr[\mathsf{out}_{\text{Game}_6^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_7^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. See Lemma 14. □

$\text{Game}_8^\sigma(\kappa)$ This game follows $\text{Game}_7^\sigma(\kappa)$, but instead of choosing a random sender at the beginning of the game and considering this message to be the right one, the oracle ignores the additional check and only looks for the first round where *both* messages are identical to the challenge.
Note that this game is entirely independent of the real sender, hence given a challenge transcript it is trivially impossible to obtain a non-negligible advantage to determine the sending party.

Claim (Indistinguishability of $\text{Game}_7^\sigma(\kappa)$ and $\text{Game}_8^\sigma(\kappa)$). Let $\mathcal{D}$ be a distinguisher with runtime $o(c^2/\alpha)$. Let the cost of acquiring a single sample be $\mathcal{O}(c)$. Then the distinguishing advantage is limited by

$$|\Pr[\mathsf{out}_{\text{Game}_7^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\text{Game}_8^\sigma(\kappa)} = 1]| \leq \alpha \tag{50}$$

The actual proof is similar to the single-bit case, only that we do not differentiate a *Bernoulli* oracle that returns a single bit but one that returns either a ℓ-bit message or an error state. However, we slightly adapt the game from Fig. 7 to our new scenario; basically instead of ranging from 0.5 to 1 the oracles now range from 0 to 1 in equally-sized steps:

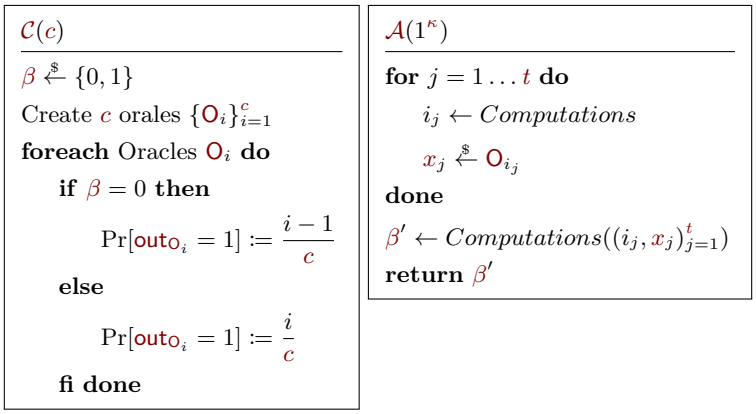


Fig. 18: Game to determine whether c Bernoulli-oracle have a common offset ξ in their probability distribution or if they follow the distribution they were given.

Lemma 53. *Let $\mathcal{D}$ be a distinguisher distinguishing $\mathrm{Game}_7^\sigma(\kappa)$ and $\mathrm{Game}_8^\sigma(\kappa)$ with advantage α over guessing. Let t be the number of queries that $\mathcal{D}$ sends to the obfuscated circuit. There is a reduction adversary $\mathcal{A}$ that uses $\mathcal{D}$ which has advantage α over guessing in winning Fig. 18.*

Proof. The overall idea of the proof is similar to that used in Fig. 7 but has some minor changes in order to incorporate the different range of the oracles and the fact that instead of 0 or 1, the oracle returns $\perp$ or Σ.

Creating the Transcript. The transcript is the same in both games and hence can be created canonically. Upon activation the adversary $\mathcal{A}$ samples the required information for the challenge transcript, that is, $\Sigma_{\mathcal{C}} \xleftarrow{\$} \{0,1\}^\ell$ and $b \xleftarrow{\$} \{0,1\}$, and creates and stores a transcript $\pi_{\mathcal{C}}$ by sampling uniformly random messages for both parties. This transcript is then reported to the distinguisher.

Simulating the Circuit. When $\mathcal{D}$ sends some input to the simulated circuit $\mathcal{A}$ has to return some message Σ according to the respective distributions. There are several possible inputs for the circuit but the below list covers all the possibilities:

(At least) one of the first-round messages is different. Then the adversary simulates the circuit by following the actual protocol. This case is equivalent for both games.

Same input until round χ, then different message for both parties. Then the adversary flips a biased coin that lands on heads with probability $p := \chi/c$ and on heads $\mathcal{A}$ returns Σ and otherwise $\mathcal{A}$ returns $\perp$.

Same input until round χ, then different message for P_b only. This case is equivalent to the one before and does not change in the game hop.

Same input until round χ, then different message for $\mathsf{P}_{\bar{b}}$ only. This is the interesting case as this class contains all the transcripts that are actually treated differently in the two games.
In this case the adversary queries the $\chi + 1$-th oracle that returns 1 with probability χ/c if $\beta = 0$ and with probability $(\chi + 1)/c$ if $\beta = 1$. When receiving output 0 the adversary returns $\perp$ as oracle-output. On output 1 the adversary returns the challenge message $\Sigma_{\mathcal{C}}$. It is easy to see that this reflects the two games; if $\beta = 0$ then Σ is returned with probability χ/c as it is in Item $\mathrm{Game}_7^\sigma(\kappa)$, and if $\beta = 1$ then Σ is returned with probability $(\chi + 1)/c$ as in Item $\mathrm{Game}_8^\sigma(\kappa)$.

Using the Response. After (at most) t queries the distinguisher terminates and sends its guess. If $\mathcal{D}$ guesses it was playing $\mathrm{Game}_8^\sigma(\kappa)$ then $\mathcal{A}$ reports $\beta' = 1$ to $\mathcal{C}$. And if $\mathcal{D}$ guesses it was playing $\mathrm{Game}_7^\sigma(\kappa)$ then $\mathcal{A}$ sends the output $\beta' = 0$ to $\mathcal{C}$.
As was described above the two distributions can be perfectly simulated, hence the probability that $\mathcal{D}$ guesses the correct game is the same that $\mathcal{A}$ has to correctly guess β. So if $\mathcal{D}$ has

non-negligible advantage α over guessing then $\mathcal{A}$ inherits this advantage for winning the game from Fig. 18, albeit if $\mathcal{D}$ sends t queries to $\mathcal{A}$, less than t queries are forwarded to $\mathcal{C}$. □

We now have a new game in Fig. 18 which we used for a more intuitive proof, however we still need to analyze how hard winning Fig. 18 really is. Fortunately, the game is already quite close to Fig. 7 only with a larger range. We thus borrow ideas from the proof of the single-bit case and adapt them to our new scenario.
Note that Lemma 3 can also be applied, and that Corollary 1 also holds for Fig. 18.
We can thus skip towards adjusting Corollary 6, for which we first need to embed Corollary 1 into our problem setting: We have c instances where the χ-th instance corresponds to distinguishing $p \coloneqq (\chi - 1)/c$ from $q \coloneqq (\chi)/c$. This implies the following L_1-norm between p and q in round χ:

$$\begin{aligned}
\mathsf{d}_{\mathsf{TV}}(p,q) &= \frac{\|p-q\|_1}{2} \\
&= \frac{1}{2}(|\Pr[p=1] - \Pr[q=1]| + |\Pr[p=0] - \Pr[q=0]|) \\
&= \frac{1}{2}\left(\left|\frac{\chi-1}{c} - \frac{\chi}{c}\right| + \left|\frac{\chi+1}{c} - \frac{\chi}{c}\right|\right) \\
&= \frac{1}{2}(\frac{2}{c}) \\
&= \frac{1}{c}
\end{aligned} \tag{51}$$

This implies again that the total variational distance in round χ is *independent* of the round χ and hence the same for all oracles. So when combining this observation with the subadditivity of the total variational distance (Lemma 2) we get that any distribution resulting from t samples have a total variational distance of at most $\frac{t}{c}$. Thus we get:

$$t \in \Omega(\alpha / \mathsf{d}_{\mathsf{TV}}(p,q)) = \Omega(\alpha \cdot c) \tag{52}$$

We thus end up with the following corollary.

Corollary 15. *Let $\mathcal{D}$ be a distinguisher playing the game from Fig. 18 using a fixed number t of samples and has runtime $o(c^2/\alpha)$. Let the cost to acquire a single sample be $\mathcal{O}(c)$. Then the distinguisher $\mathcal{D}$ is correct with probability at most $1/2 + \alpha/2$.*

Proof. See Corollary 6. Eq. (52) yields a lower bound for the number of samples required by $\mathcal{D}$ to distinguish with advantage α. With each sample having a cost of $\mathcal{O}(c)$ an advantage of α requires runtime in $\mathcal{O}(c^2 \cdot \alpha)$ to get the samples alone. With the runtime of $\mathcal{D}$ being bounded as $o(c^2/\alpha)$ it follows that $\mathcal{D}$ can not have constant advantage. □

Secrecy. We conclude this section with a secrecy analysis of the protocol. Fortunately, this proof is quite close to that from the single-bit protocol from Theorem 6. The main difference is that we have to change an entire message instead of only changing the single bit.

However, the crucial part is still the statistical security of the One-Time-Pad, which holds also in the multi-bit case. To prove our claim we thus show that the transfer of a uniformly random message cannot be distinguished efficiently from the transfer of the all-zero bitstring without the receiver's random tape.

In total we want to show that:

$$\begin{aligned}
&\{\pi \leftarrow \mathsf{Transfer}_{\langle \mathsf{R},\mathsf{P}_0,\mathsf{P}_1\rangle}(crs, b, \Sigma)\} \\
\approx\ &\{\pi \leftarrow \mathsf{Transfer}_{\langle \mathsf{R},\mathsf{P}_0,\mathsf{P}_1\rangle}(crs, b, \vec{0}^{\ell})\}
\end{aligned} \tag{53}$$

With that in mind we adjust the gamehops as follows:

$\mathrm{Game}_1(\kappa)$

$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$(OTP, \mathsf{vk}_\mathsf{R}) \coloneqq \mathrm{PKE.Dec}^*(\mathsf{sk}_P, x_\mathsf{R}^{(0)})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\vec{0}^\ell, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_\mathsf{R}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^\ell$
$(\mathsf{k}_\mathsf{R}, \mathsf{vk}_\mathsf{R}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_\mathsf{R}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_\mathsf{R}))$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^\kappa)$
$(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_\mathsf{R}^{(0)})$
$x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\Sigma, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $\mu_b^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, \Sigma \| \mu_b^{(\chi)})$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_\mathsf{R}, \pi)$
return $\mathcal{S}_\mathcal{A}^{P'^{AT}}(\pi, T_\mathsf{R})$

$\mathrm{Game}_2(\kappa)$

if $\pi = \pi_\mathcal{C}$ **then**
  return $OTP_\mathcal{C} \oplus \Sigma_\mathcal{C}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_\mathcal{C}[0]) \wedge$
    $\pi[0] \neq \pi_\mathcal{C}[0]$ **then**
  abort
elseif $\pi[0] = \pi_\mathcal{C}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_\mathcal{C}[\chi'])$
  if $\pi[x_{b_\mathcal{C}}^{c^*+1}] = \pi_\mathcal{C}[x_{b_\mathcal{C}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_\mathcal{C}}, x_{b_\mathcal{C}}^{(\chi^*+1)}) == (\sigma_\mathcal{C} \| {\mu_\mathcal{C}}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_\mathcal{C} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\Sigma_\mathcal{C}, \bot)$
if $\pi[x_\mathsf{R}^{(0)}] = \pi_\mathcal{C}[x_\mathsf{R}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_\mathsf{R}) \coloneqq (OTP_\mathcal{C}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\vec{0}^\ell, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg\mathrm{SIG.Vfy}(\mathsf{vk}_\mathsf{R}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_\mathsf{R}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg\mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi-1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^\ell$
$(\mathsf{k}_\mathsf{R}, \mathsf{vk}_\mathsf{R}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_\mathsf{R}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_b}$:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_\mathsf{R}, \pi)$
return $\mathcal{S}_\mathcal{A}^{P'^{AT}}(\pi, T_\mathsf{R})$

$\mathrm{Game}_2(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{c^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\Sigma_{\mathcal{C}}, \perp)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}]$ **then**
  $(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq (OTP_{\mathcal{C}}, \mathsf{vk}_{\mathcal{C}\mathsf{R}})$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\vec{0}^{\ell}, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\perp$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \perp)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{c^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\Sigma_{\mathcal{C}}, \perp)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}] \wedge \pi \neq \pi_{\mathcal{C}}$ **then**
  return $\perp$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\vec{0}^{\ell}, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\perp$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \perp)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
P_{b}:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  P_{b}:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_3(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \Sigma_{\mathcal{C}}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{c^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\Sigma_{\mathcal{C}}, \bot)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}] \wedge \pi \neq \pi_{\mathcal{C}}$ **then**
  return $\bot$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\vec{0}^\ell, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^\ell$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_b}$:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \vec{0}^\ell$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{c^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}} \| {\mu_{\mathcal{C}}}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}^{(\pi)}_{(p)}(\vec{0}^\ell, \bot)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}] \wedge \pi \neq \pi_{\mathcal{C}}$ **then**
  return $\bot$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\vec{0}^\ell, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(\chi_b/c)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^\ell$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^\kappa)$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_b}$:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_4(\kappa)$

if $\pi = \pi_{\mathcal{C}}$ **then**
  return $OTP_{\mathcal{C}} \oplus \vec{0}^{\ell}$
elseif $\mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi[0]) = \mathrm{PKE.Dec}^*(\mathsf{sk}_P, \pi_{\mathcal{C}}[0]) \wedge$
    $\pi[0] \neq \pi_{\mathcal{C}}[0]$ **then**
  abort
elseif $\pi[0] = \pi_{\mathcal{C}}[0]$ **then**
  $\chi^* \coloneqq \mathbf{argmin}_{\chi'}(\pi[\chi'] \neq \pi_{\mathcal{C}}[\chi'])$
  if $\pi[x_{b_{\mathcal{C}}}^{c^*+1}] = \pi_{\mathcal{C}}[x_{b_{\mathcal{C}}}^{\chi^*+1}] \vee$
    $\mathrm{SKE.Dec}^*(\mathsf{sk}_{b_{\mathcal{C}}}, x_{b_{\mathcal{C}}}^{(\chi^*+1)}) == (\sigma_{\mathcal{C}} \| \mu_{\mathcal{C}}{}^{(\chi^*+1)})$ **then**
    $\chi^* \coloneqq \chi^* + 1$
  $p \coloneqq 1/2 + \chi^*/2c$
  return $OTP_{\mathcal{C}} \oplus \mathsf{CointossS}_{(p)}^{(\pi)}(\vec{0}^{\ell}, \bot)$
if $\pi[x_{\mathsf{R}}^{(0)}] = \pi_{\mathcal{C}}[x_{\mathsf{R}}^{(0)}] \wedge \pi \neq \pi_{\mathcal{C}}$ **then**
  return $\bot$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\vec{0}^{\ell}, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}_{(\chi_b/c)}^{(\pi)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_b}$:
$-$
$-$
$-$
$x_b^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $-$
  $x_b^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_5(\kappa)$

$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
$-$
return $\bot$
$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$
$(\Sigma, \mu_0^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$
$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \mathrm{PKE.Dec}(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$
$(\vec{0}^{\ell}, \mu_1^{(0)}) \coloneqq \mathrm{SKE.Dec}(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$
if $\neg \mathrm{SIG.Vfy}(\mathsf{vk}_{\mathsf{R}}, \pi)$ **then**
  return $\bot$
$\chi_0 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_0^{(0)}, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
$\chi_1 \coloneqq [\![\mathrm{SIG.Vfy}(\mu_1^{(0)}, \mathsf{vk}_1, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  foreach $b \in \{0,1\} | \chi_b > c$ **do**
    $X_b \coloneqq \mathrm{SKE.Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$
    $\Sigma'_b \coloneqq X_b[0\colon \ell - 1]$
    $\mu_b^{(\chi)} \coloneqq X_b[\ell\colon |X_b|]$
    if $\neg \mathrm{SIG.Vfy}(\mu_b^{(\chi)}, \mathsf{vk}_b, \pi[\chi - 1]) \vee \Sigma_b \neq \Sigma'_b$ **then**
      $\chi_b \coloneqq \chi$
$b' \coloneqq \mathbf{argmax}_b(\chi_b)$
return $OTP \oplus \mathsf{CointossS}_{(\chi_b/c)}^{(\pi)}(\Sigma^{(b')}, \bot)$

R:
$OTP \xleftarrow{\$} \{0,1\}^{\ell}$
$(\mathsf{k}_{\mathsf{R}}, \mathsf{vk}_{\mathsf{R}}) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$x_{\mathsf{R}}^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (OTP, \mathsf{vk}_{\mathsf{R}}))$
$\mathsf{P_b}$:
$(\mathsf{k}_b, \mathsf{vk}_b) \leftarrow \mathrm{SIG.KeyGen}(1^{\kappa})$
$(\mathsf{sk}_b) \leftarrow \mathrm{SKE.KeyGen}(1^{\kappa})$
$(\mu_b^{(0)}) \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, x_{\mathsf{R}}^{(0)})$
$x_b^{(0)} \leftarrow \mathrm{PKE.Enc}(\mathsf{pk}_P, (\mathsf{sk}_b, \mathsf{vk}_b)) \| \mathrm{SKE.Enc}(\mathsf{sk}_b, (\vec{0}^{\ell}, \mu^{(0)}))$
$\mathsf{P}_{\overline{\mathsf{b}}}$:
$x_{\overline{b}}^{(0)} \xleftarrow{\$} \{0,1\}^{2m}$
foreach $\chi \in \{1, \ldots, c\}$ **do**
  $\mathsf{P_b}$:
  $\mu_b^{(\chi)} \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_b, (x_0^{(\chi-1)}, x_1^{(\chi-1)}))$
  $x_b^{(\chi)} \leftarrow \mathrm{SKE.Enc}(\mathsf{sk}_b, (\vec{0}^{\ell}, \mu_b^{(\chi)}))$
  $\mathsf{P}_{\overline{\mathsf{b}}}$:
  $-$
  $x_{\overline{b}}^{(\chi)} \xleftarrow{\$} \{0,1\}^m$
R:
$\mu \leftarrow \mathrm{SIG.Sig}(\mathsf{k}_{\mathsf{R}}, \pi)$
return $\mathcal{S}_{\mathcal{A}}^{P'_{AT}}(\pi, T_{\mathsf{R}})$

$\mathrm{Game}_1^\sigma(\kappa)$ This is the original game (after replacing the obfuscated circuit with oracle access to the circuit, the PRF with an actual random oracle and the adversary with the simulator), where the sending party P_b is chosen uniformly at random and the sender uses the input message Σ.

$\mathrm{Game}_2^\sigma(\kappa)$ This game follows $\mathrm{Game}_1^\sigma(\kappa)$, but with the following changes:

- During the simulation of the oracle P_{AT} from Fig. 16 the simulation enforces correctness of the challenge transcript $\pi_\mathcal{C}$: if the input transcript π matches the challenge transcript $\pi_\mathcal{C}$, it returns $OTP_\mathcal{C} \oplus \Sigma_\mathcal{C}$.
- During simulation of the circuit the adversary aborts if any of the first-round messages from P_0 or P_1 differ from the messages reported in the challenge transcript *and* the decryptions still match.
- During simulation of the circuit, if the first receiver message of the input transcript π is the same as that of the challenge transcript $\pi_\mathcal{C}$, instead of decrypting it the circuit directly sets $OTP = OTP_\mathcal{C}$ and $\mathsf{vk}_\mathsf{R} = \mathsf{vk}_{\mathcal{C}\mathsf{R}}$ as the values used in the creation of the challenge transcript.
- During simulation of the oracle, if the first-round messages of both parties match the first-round messages in the challenge transcript $\pi_\mathcal{C}$. In this case, the program compares the input transcript π with the challenge transcript $\pi_\mathcal{C}$ until it finds the first round χ^* in which the input differs from the challenge transcript. It then checks round $\chi^* + 1$, and if it contains the same message from the sending party, it adds one to χ.
  Finally, the circuit flips a biased coin, which returns the correct message $OTP_\mathcal{C} \oplus \Sigma_\mathcal{C}$ with probability $p \coloneqq \chi^*/c$ and an error message $\perp$ otherwise.

Lemma 54. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SKE be an IND$-CCA secure secret-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_1^\sigma(\kappa)$ *and* $\mathrm{Game}_2^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_1^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_2^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Follows from Lemmas 47 to 52. □

$\mathrm{Game}_3^\sigma(\kappa)$ This game follows $\mathrm{Game}_2^\sigma(\kappa)$ but the oracle is simulated slightly different: If the first receiver message is the same as the one reported in the challenge transcript and the input transcript is *not* the challenge transcript, then the circuit outputs an error symbol $\perp$.
Note that this does not work in the anonymity proof as there we assume that the adversary is given access to the receivers random tape, and hence can create their own new signature on the modified transcript.

Lemma 55. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SKE be an IND$-CCA secure secret-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_3^\sigma(\kappa)$ *and* $\mathrm{Game}_4^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_3^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_4^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. We reduce a distinguisher $\mathcal{D}$ between these two games to an adversary $\mathcal{A}$ on the EUF-CMA security of the signature scheme SIG.

Creating the ranscript. The creation of the transcript is straightforward and works by sampling random messages for each party.

Simulating the Oracle. The used verification key vk_R is set to be the verification key of the challenger. Thus the final signature μ on the entire transcript, which is required for the circuit to not *abort* (by outputting a random bit), needs to be forged. Hence for each input transcript the adversary first checks if the receiver message is equivalent to that from the challenge transcript. If it isn't then the transcript can not be used for distinguishing anyways. If it is, the adversary checks if the remaining transcript is the same as well. If it is, the transcript can not be used for distinguishing and simulation just continues as the path taken is equivalent in both games. If it isn't then the adversary checks if the signature μ verifies under the used key. If it doesn't then both games act exactly the same and output an error state $\perp$. Hence to distinguish the signature has to verify. Then, however, the signature is a valid forgery.

Translating the Result. If no valid signature was queried then—as mentioned above—the distinguishing advantage must be negligible. For a non-negligible advantage the distinguisher has to query at least one signature. This can be used as forgery to break the EUF-CMA security of SIG.
As stated above, the advantage of the distinguisher is directly related to the probability of successfully distinguishing. Hence $\mathcal{A}$ will have a valid forgery with non-negligible advantage. This would contradict the EUF-CMA security of the signature scheme and thus completes our proof. □

$\mathrm{Game}_4^\sigma(\kappa)$ This game is as $\mathrm{Game}_3^\sigma(\kappa)$ but instead of using a uniformly random Σ for the transferred message during simulation of the circuit we now fix $\Sigma = \vec{0}^\ell$ as the all-zero bitstring of appropriate size.

Lemma 56. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SKE be an IND$-CCA secure secret-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_3^\sigma(\kappa)$ *and* $\mathrm{Game}_4^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_3^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_4^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Again indistinguishability follows for any adversary who does not have the receivers random tape. The message is only ever used in its encrypted form via the uniformly random One-Time-Pad, and distinguishing two bitstrings after they have been masked is statistically impossible to do with non-negligible advantage. Thus indistinguishability follows. □

$\mathrm{Game}_5^\sigma(\kappa)$ This game follows $\mathrm{Game}_4^\sigma(\kappa)$ but undoes all the changes from the first-to-second gamehop and that from $\mathrm{Game}_3^\sigma(\kappa)$.

Lemma 57. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SKE be an IND$-CCA secure secret-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_4^\sigma(\kappa)$ *and* $\mathrm{Game}_5^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_4^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_5^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. Follows from Lemmas 47 to 52 and 55. □

Thus *OTP* is hidden to any guessing algorithm and so is Σ.

Corollary 16. *The Anonymous Transfer from Fig. 16 is* ς*-secret with* $\varsigma \in \mathsf{owhl}(\kappa)$.

In total, we thus have:

Corollary 9. *The protocol* $\Pi_{AT}^{\ell\,\prime}$ *defined in Appendix F is a strong* $\mathfrak{C}_1$*-fine-grained* $(1-\mathsf{negl}(\kappa), \delta, 1-\mathsf{negl}(\kappa))$*-AT against* $\mathfrak{C}_2$*, where* $\mathfrak{C}_1 := \mathcal{O}(c)$ *and* $\mathfrak{C}_2 := \mathrm{o}(c^2(1-\delta))$.

G Undetectable Oblivious Transfer from Anonymous Transfer and two-round Covert Oblivious Transfer

In this section we present an instantiation of Undetectable Oblivious Transfer based on any two-round Covert Oblivious Transfer (COT)[19] protocol such as the one from von Ahn, Hopper, and Langford [vHL05, Protocol 4]. The protocol is fairly canonical and basically just uses a different communication structure where messages are sent via Anonymous Transfer.

In Fig. 19 we present a generic transformation that takes any two-round Covert Oblivious Transfer protocol and a strong $(\varepsilon, \delta, \varsigma, c, \ell)$-Anonymous Transfer and constructs a (ε, δ)-Undetectable Oblivious Transfer. For the sake of simplicity we only introduce a generic three-party protocol (that has one dummy friend), but we stress that the same technique can easily be modified to incorporate more parties at the cost of using 2 more ATs with the respective dummy friend as the receiver.

[19] See Appendix A.1 for a definition of COT.

Protocol Π_{UOT} in the $\mathcal{F}_{AT}^{(2,3)}$-hybrid model.

On input (Σ_0, Σ_1), $\mathsf{P_S}$ stores both Σ_0 and Σ_1 and sets $x_0 := \bot$.
On input $\sigma \in \{0,1\}$, $\mathsf{P_R}$ chooses x_1 as the first message the receiver sends to the sender in Π_{COT} for bit σ.
On input $\bot$, $\mathsf{P_D}$ sets $x_\mathsf{D} := \bot$.
Each party P_i sends x_i to both instances of $\mathcal{F}_{AT}^{(2,3)}$ where party P_j for $j \neq i$ is the receiver.
On input x_0^* from $\mathcal{F}_{AT}^{(2,3)}$ where $\mathsf{P_S}$ is the receiver, $\mathsf{P_S}$ computes x_0 according to Π_{COT} on input (Σ_0, Σ_1) and first message x_0^*.
On input $\bot$ from $\mathcal{F}_{AT}^{(2,3)}$ where $\mathsf{P_R}$ is the receiver, $\mathsf{P_R}$ sets $x_1 := \bot$.
On input $\bot$ from $\mathcal{F}_{AT}^{(2,3)}$ where $\mathsf{P_D}$ is the receiver, $\mathsf{P_D}$ sets $x_2 := \bot$.
Each party P_i sends x_i to both instances of $\mathcal{F}_{AT}^{(2,3)}$ where party P_j for $j \neq i$ is the receiver.
On input $\bot$ from $\mathcal{F}_{AT}^{(2,3)}$ where $\mathsf{P_S}$ is the receiver, $\mathsf{P_S}$ outputs $\bot$.
On input x_0 from $\mathcal{F}_{AT}^{(2,3)}$ where $\mathsf{P_R}$ is the receiver, $\mathsf{P_R}$ reconstructs Σ_σ according to Π_{COT} on inputs (x_0, x_1, σ) and outputs Σ_σ.
On input $\bot$ from $\mathcal{F}_{AT}^{(2,3)}$ where $\mathsf{P_D}$ is the receiver, $\mathsf{P_D}$ outputs $\bot$.

Fig. 19: A protocol for Undetectable Oblivious Transfer.

G.1 Correctness

For analyzing the correctness we assume that the Covert Oblivious Transfer protocol reconstructs the bit correctly with overwhelming probability. Then it holds for the correctness of the Undetectable Oblivious Transfer:

Lemma 58 (Correctness of the Undetectable Oblivious Transfer protocol). *Let Π_{COT} be correct with overwhelming probability. Let Π_{AT}^{ℓ} be a strong $(\varepsilon, \delta, \varsigma, c, \ell)$-AT. Then the Undetectable Oblivious Transfer protocol from Fig. 19 is correct with $\varepsilon_{UOT} = \frac{\varepsilon^2+2\varepsilon-1}{2}$.*

Proof. We assume that when executed directly, Π_{COT} is correct assuming that all messages are present and wrong if a message was transferred incorrectly. Thus the probability that the message has been transferred correctly corresponds to the probability that *both* messages have been transferred correctly—the first-round message by the receiver to the sender and the second-round message by the sender to the receiver.

With the AT being ε-correct it follows from Eq. (3) that

$$\Pr\left[\sigma \xleftarrow{\$} \{0,1\}; \pi \xleftarrow{\$} \langle \mathsf{R}, \mathsf{P}_0, \mathsf{P}_1\rangle(b, \sigma); \sigma' \leftarrow f(\pi, T_\mathsf{R}) \colon \sigma = \sigma'\right] \geq (\varepsilon + 1)/2 \tag{54}$$

and the probability that *both* messages are transferred correctly squares that value. As such it holds that:

$$\begin{aligned}\left(\Pr\left[\sigma \xleftarrow{\$} \{0,1\}; \pi \xleftarrow{\$} \langle \mathsf{R}, \mathsf{P}_0, \mathsf{P}_1\rangle(b, \sigma); \sigma' \leftarrow f(\pi, T_\mathsf{R}) \colon \sigma = \sigma'\right]\right)^2 &\geq ((\varepsilon + 1)/2)^2 \\ &= \frac{1}{4}(\varepsilon^2 + 2\varepsilon + 1)\end{aligned} \tag{55}$$

So we have for the correctness of the OT:

$$\begin{aligned}&\Pr[\varpi \xleftarrow{\$} \mathbb{S}_3, (x_0, x_1) \xleftarrow{\$} \{0,1\}^2, \sigma \xleftarrow{\$} \{0,1\}, \\ &\quad x'_\sigma \leftarrow \mathrm{OT}(\varpi\{(x_0, x_1), \sigma, \bot\}) \colon x_\sigma = x'_\sigma] && \geq \frac{1}{4}(\varepsilon^2 + 2\varepsilon + 1) \\ &&& = \frac{\varepsilon^2}{4} + \frac{\varepsilon}{2} + \frac{1}{4} \\ &\Pr[\varpi \xleftarrow{\$} \mathbb{S}_3, (x_0, x_1) \xleftarrow{\$} \{0,1\}^2, \sigma \xleftarrow{\$} \{0,1\}, \\ &\quad x'_\sigma \leftarrow \mathrm{OT}(\varpi\{(x_0, x_1), \sigma, \bot\}) \colon x_\sigma = x'_\sigma] - \frac{1}{2} && \geq \frac{\varepsilon^2}{4} + \frac{\varepsilon}{2} - \frac{1}{4} \\ &&& = \frac{1}{2} \cdot \left(\frac{\varepsilon^2 + 2\varepsilon - 1}{2}\right)\end{aligned} \tag{56}$$

Thus we get that according to Eq. (21) we have $\varepsilon_{UOT} := \frac{\varepsilon^2+2\varepsilon-1}{2}$. □

G.2 Privacy

The analysis of (input-) privacy is pretty straightforward, simply because our transformation leaves no extra insecurities that would allow extraction of either the choice bit σ or the message $\Sigma_{\overline{\sigma}}$ *not* selected by the receiver. Yet in the following we provide a full proof of privacy.

Sender Privacy We first define the privacy notion of classical OT protocols in a game-based notion:

Definition 18 (Sender-Privacy in Oblivious Transfer). *Let OT be an oblivious transfer protocol. OT provides computational* sender-privacy *if for all PPT adversaries $\mathcal{A}$ it holds that:*

$$\begin{aligned}\Pr[(x_0,x_1)=(x_0^*,x_1^*)|(x_0,x_1) \xleftarrow{\$} \{0,1\}^{2\ell},(\sigma,st)\leftarrow\mathcal{A}_0(1^\kappa),\\ \pi\leftarrow \mathrm{OT}((x_0,x_1),\sigma),(x_0^*,x_1^*)\leftarrow\mathcal{A}_1(\pi,st,T_\mathsf{R})]\in\mathsf{negl}(\kappa)\end{aligned} \tag{57}$$

The definition enforces that the adversary cannot extract *both* bits input by the sender. While the bit x_σ is easy to extract given the transcript, the state of $\mathcal{A}_0$ and the random tape T_R, we enforce that the other bit $x_{\overline{\sigma}}$ can not be determined better than by guessing.

Definition 19 (Receiver-Privacy in Oblivious Transfer). *Let OT be an Oblivious Transfer protocol. OT provides computational* receiver-privacy *if for all PPT adversaries $\mathcal{A}$ it holds that:*

$$\begin{aligned}|\Pr[\sigma=\sigma^*|(x_0,x_1,st)\leftarrow\mathcal{A}_0(1^\kappa),\sigma\xleftarrow{\$}\{0,1\},\\ \pi\leftarrow\mathrm{OT}((x_0,x_1),\sigma),\sigma^*\leftarrow\mathcal{A}_1(\pi,st,T_\mathsf{S})]-1/2|\in\mathsf{negl}(\kappa)\end{aligned} \tag{58}$$

This definition is practically the same as that for sender-privacy in Definition 18 but lets the adversary play as the (semi-honest) *sender* who has to recover the choice bit by the *receiver*.

With that we can prove the following claim:

Lemma 59 (Privacy of the Undetectable Oblivious Transfer protocol). *Let Π_{COT} be an OT protocol with overwhelming sender-privacy. Then Π_{UOT} from Fig. 19 is private for both parties.*

Proof. **Sender-Privacy**

Any COT protocol automatically fulfills the privacy requirement of an ordinary OT from Definition 18, as such we reduce sender privacy of Π_{UOT} to the sender privacy of Π_{COT}. To that end, let A be a guessing algorithm that breaks the sender privacy of Π_{UOT} with non-negligible advantage α. From A we construct an adversary $\mathcal{A}$ who breaks the condition from Eq. (57) as follows:

$\mathcal{A}$ asks A_0 for receiver input σ and hands this to the challenger $\mathcal{C}$ of the COT sender privacy. From $\mathcal{C}$, $\mathcal{A}$ obtains a transcript π of sent messages. $\mathcal{A}$ simulates inserting each sender message of π into Π_{AT}^{ℓ} with receivers P_R and P_D and each receiver message into $\mathcal{F}_{AT}^{(2,3)}$ with receivers P_S and P_D.

This results in a transcript of Π_{UOT} which $\mathcal{A}$ hands to A. Eventually A returns a tuple (x_0^*,x_1^*) which $\mathcal{A}$ hands directly to the challenger $\mathcal{C}$.

Note that if A is correct, then so is $\mathcal{A}$, and if A is incorrect then the guess of $\mathcal{A}$ is also wrong, as the simulation did not change the sent messages.

Thus if A has non-negligible advantage then so has $\mathcal{A}$, which it does not have by requirement.

Receiver Privacy

We let A be a guessing algorithm attacking the receiver privacy of Π_{UOT} and construct an adversary $\mathcal{A}$ attacking the receiver privacy of Π_{COT} with the same success probability. The adversary asks A for the senders input $(x_0,x_1)\in\{0,1\}^2$ and forwards that to the challenger $\mathcal{C}$. The obtained transcript π is translated into a transcript for Π_{UOT} by simulating the AT with the inputs being the messages reported by the respective parties, where the input by the dummy friend is constantly $\perp$.

$\mathcal{A}$ then hands the transcript over to A who guesses a bit σ. The bit is then forwarded to $\mathcal{C}$.

Again, the transformation did not change any values, hence if A is correct with probability $1/2+\alpha$ then $\mathcal{A}$ is also correct with probability $1/2+\alpha$. The latter, however, is not possible by requirement from Π_{COT}, which concludes our proof. □

G.3 Anonymity

As the adversary is given the choice to corrupt any party we consider each potential party individually and show the two distributions that need to be indistinguishable.

P_0 sender	P_1 sender
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_S}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_S}$:
$(x_{COT^{\mathsf{R}}}, \bot)$	$(x_{COT^{\mathsf{R}}}, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:
$(\bot, \bot)$	$(x_{COT^{\mathsf{R}}}, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:
$(x_{COT^{\mathsf{R}}}, \bot)$	$(\bot, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_S}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_S}$:
$(\bot, \bot)$	$(\bot, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:
$(x_{COT^{\mathsf{S}}}, \bot)$	$(x_{COT^{\mathsf{S}}}, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:
$(x_{COT^{\mathsf{S}}}, \bot)$	$(x_{COT^{\mathsf{S}}}, \bot)$

Fig. 20: The two distributions we have to prove indistinguishable in case the adversary A_0 picks the sender.

Corrupted Sender The guessing algorithm has to distinguish the following two distributions:

$$\begin{aligned} &\{(\Sigma_0, \Sigma_1) \xleftarrow{\$} \{0,1\}^{2\ell}, \sigma \xleftarrow{\$} \{0,1\} \colon \pi \xleftarrow{\$} \langle \mathsf{P_R}, \mathsf{P}_0, \mathsf{P}_1 \rangle (\sigma, (\Sigma_0, \Sigma_1), \bot)\} \\ \approx &\{(\Sigma_0, \Sigma_1) \xleftarrow{\$} \{0,1\}^{2\ell}, \sigma \xleftarrow{\$} \{0,1\} \colon \pi \xleftarrow{\$} \langle \mathsf{P_R}, \mathsf{P}_0, \mathsf{P}_1 \rangle (\sigma, \bot, (\Sigma_0, \Sigma_1))\} \end{aligned} \tag{59}$$

In this section we prove anonymity against any adversary that corrupts the sender after the execution. This adversary then has to guess which of the remaining parties acted as a receiver and which was the dummy friend.

The two possible transcripts if he sender is fixed are depicted in Fig. 20.

$\mathrm{Game}_1(\kappa)$	$\mathrm{Game}_2(\kappa)$
$\mathsf{P_S}$:	$\mathsf{P_S}$:
$x_{\mathsf{S}}^{(0)} \xleftarrow{\$} \{0,1\}^m$	$x_{\mathsf{S}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_R}$:	$\mathsf{P_R}$:
$x_{\mathsf{R}}^{(0)} \leftarrow \Pi_{COT\mathsf{R}}(\sigma)$	$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_D}$:	$\mathsf{P_D}$:
$x_{\mathsf{D}}^{(0)} \xleftarrow{\$} \{0,1\}^m$	$x_{\mathsf{R}}^{(0)} \leftarrow \Pi_{COT\mathsf{R}}(\sigma)$
$\mathsf{P_i}$:	$\mathsf{P_i}$:
$\forall_{j\neq i}$: $x_i^{(0)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$	$\forall_{j\neq i}$: $x_i^{(0)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
$\mathsf{P_S}$:	$\mathsf{P_S}$:
$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{S})$	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{S})$
$x_{\mathsf{S}}^{(1)} := \Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$	$x_{\mathsf{S}}^{(1)} := \Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$
$\mathsf{P_R}$:	$\mathsf{P_R}$:
$x_{\mathsf{R}}^{(1)} \xleftarrow{\$} \{0,1\}^m$	$x_{\mathsf{R}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_D}$:	$\mathsf{P_D}$:
$x_{\mathsf{D}}^{(1)} \xleftarrow{\$} \{0,1\}^m$	$x_{\mathsf{D}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_i}$:	$\mathsf{P_i}$:
$\forall_{j\neq i}$: $x_i^{(1)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$	$\forall_{j\neq i}$: $x_i^{(1)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
$\mathsf{P_R}$:	$\mathsf{P_R}$:
$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$
return $\Pi_{COT}(x_{\mathsf{R}}^{(0)}, \Sigma)$	**return** $\Pi_{COT}(x_{\mathsf{R}}^{(0)}, \Sigma)$

The single game required for that—which is actually the same as in Fig. 20.
The formal description is as follows:

$\mathrm{Game}_1^{\sigma}(\kappa)$ is the original game where the sender is $\mathsf{P_S}$ and the receiver is $\mathsf{P_R}$.

$\mathrm{Game}_2^{\sigma}(\kappa)$ is as Item $\mathrm{Game}_1^{\sigma}(\kappa)$ but in providing the transcript the simulator follows Item $\mathrm{Game}_1^{\sigma}(\kappa)$ for the first round but inserts $x_{\mathsf{R}}^{(0)}$ in the name of the *dummy friend* and $x_{\mathsf{D}}^{(0)}$ in the name of the *receiver* into $\mathcal{F}_{AT}^{(2,3)}(\mathsf{S})$. Note that this corresponds to the final distribution from Fig. 20.

Lemma 60. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SKE be an IND$-CCA secure secret-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_1^{\sigma}(\kappa)$ *and* $\mathrm{Game}_2^{\sigma}(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_1^{\sigma}(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_2^{\sigma}(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. We reduce to the δ-anonymity of $\mathcal{F}_{AT}^{(2,3)}$. If there was some distinguisher $\mathcal{D}$ who can distinguish $\mathrm{Game}_1^{\sigma}(\kappa)$ from $\mathrm{Game}_2^{\sigma}(\kappa)$ with advantage $1/2 + \alpha$ then there is some adversary $\mathcal{A}$ (which implies a guessing algorithm A) who can determine the sender with advantage α. The limit δ then provides an upper bound on α.

Creating the Transcript. We denote by $\mathcal{C}$ the challenger for the anonymity-game of $\mathcal{F}_{AT}^{(2,3)}$. $\mathcal{A}$ determines the message to-be-transferred as $\Pi_{COT\mathsf{R}}(\sigma)$, that is, the first message sent by a receiver in Π_{COT} who wants to receive x_σ. From that $\mathcal{C}$ returns a transcript of the $\mathcal{F}_{AT}^{(2,3)}$ instance, which $\mathcal{A}$ inserts into the transcript provided to the distinguisher $\mathcal{D}$ instead of honestly simulating the first $\mathcal{F}_{AT}^{(2,3)}$ instance where the sender S of the AT is the receiver, the remainder is simulated as-is.

Translating the Result. Eventually $\mathcal{D}$ returns a guess which is either $\mathrm{Game}_1^{\sigma}(\kappa)$ or $\mathrm{Game}_2^{\sigma}(\kappa)$. In case $\mathcal{D}$ assumes to be in $\mathrm{Game}_1^{\sigma}(\kappa)$ we assume the party we refer to as $\mathsf{P_R}$ is the sender, and if $\mathcal{D}$ assumes to be in $\mathrm{Game}_2^{\sigma}(\kappa)$ we let $\mathcal{A}$ refer to $\mathsf{P_D}$ as the sender in the AT.
As can be seen the distributions match perfectly, as the only change is the sender and that one perfectly translates to our problem of de-anonymization.

Thus it follows that $\mathcal{D}$ is correct with probability

$$\begin{aligned} &\Pr[\mathcal{D} \text{ correct}] \leq 1/2 + \alpha \\ \implies &|\Pr[\mathcal{D} \text{ correct}] - 1/2| = |\alpha| = \alpha \leq (1-\delta)/2 \end{aligned} \tag{60}$$

and thus if the adversary corrupts the sender then the protocol provides δ-Anonymity where δ is inherited from $\mathcal{F}_{AT}^{(2,3)}$. □

P_0 sender	P_1 sender
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:
$(x_{COT^{\mathsf{R}}}, \perp)$	$(x_{COT^{\mathsf{R}}}, \perp)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:
$(x_{COT^{\mathsf{R}}}, \perp)$	$(x_{COT^{\mathsf{R}}}, \perp)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_R}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_R}$:
$(\perp, \perp)$	$(\perp, \perp)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:
$(\perp, \perp)$	$(x_{COT^{\mathsf{S}}}, \perp)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:
$(x_{COT^{\mathsf{S}}}, \perp)$	$(\perp, \perp)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_R}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_R}$:
$(x_{COT^{\mathsf{S}}}, \perp)$	$(x_{COT^{\mathsf{S}}}, \perp)$

Fig. 21: The two distributions we have to prove indistinguishable in case the adversary A_0 picks the receiver.

Corrupted Receiver In this section we analyze the anonymity of the remaining parties provided that the adversary corrupts the receiver of the message. In this case the party $\mathsf{P_R}$ is fixed and the adversary has to distinguish between the two cases depicted in Fig. 21: Either P_0 is the sender and P_1 is the dummy friend or vice versa. The guessing algorithm has to distinguish the following two distributions:

$$\begin{aligned} &\{(\Sigma_0, \Sigma_1) \xleftarrow{\$} \{0,1\}^{2\ell}, \sigma \xleftarrow{\$} \{0,1\} \colon \pi \xleftarrow{\$} \langle \mathsf{P_R}, \mathsf{P}_0, \mathsf{P}_1 \rangle (\sigma, (\Sigma_0, \Sigma_1), \perp)\} \\ \approx &\{(\Sigma_0, \Sigma_1) \xleftarrow{\$} \{0,1\}^{2\ell}, \sigma \xleftarrow{\$} \{0,1\} \colon \pi \xleftarrow{\$} \langle \mathsf{P_R}, \mathsf{P}_0, \mathsf{P}_1 \rangle (\sigma, \perp, (\Sigma_0, \Sigma_1))\} \end{aligned} \tag{61}$$

To prove the hardness of distinguishing these two cases we proceed as follows:

$\mathrm{Game}_1(\kappa)$	$\mathrm{Game}_2(\kappa)$
$\mathsf{P_S}$:	$\mathsf{P_S}$:
$x_{\mathsf{S}}^{(0)} \xleftarrow{\$} \{0,1\}^m$	$x_{\mathsf{S}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_R}$:	$\mathsf{P_R}$:
$x_{\mathsf{R}}^{(0)} \leftarrow \Pi_{COT\mathsf{R}}(\sigma)$	$x_{\mathsf{R}}^{(0)} \leftarrow \Pi_{COT\mathsf{R}}(\sigma)$
$\mathsf{P_D}$:	$\mathsf{P_D}$:
$x_{\mathsf{D}}^{(0)} \xleftarrow{\$} \{0,1\}^m$	$x_{\mathsf{D}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_i}$:	$\mathsf{P_i}$:
$\forall_{j\neq i}\colon x_i^{(0)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$	$\forall_{j\neq i}\colon x_i^{(0)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
$\mathsf{P_S}$:	$\mathsf{P_S}$:
$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{S})$	–
$x_{\mathsf{S}}^{(1)} := \Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$	$x_{\mathsf{S}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_R}$:	$\mathsf{P_R}$:
$x_{\mathsf{R}}^{(1)} \xleftarrow{\$} \{0,1\}^m$	$x_{\mathsf{R}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
$\mathsf{P_D}$:	$\mathsf{P_D}$:
–	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{S})$
$x_{\mathsf{D}}^{(1)} \xleftarrow{\$} \{0,1\}^m$	$x_{\mathsf{D}}^{(1)} := \Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$
$\mathsf{P_i}$:	$\mathsf{P_i}$:
$\forall_{j\neq i}\colon x_i^{(1)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$	$\forall_{j\neq i}\colon x_i^{(1)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
$\mathsf{P_R}$:	$\mathsf{P_R}$:
$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$
return $\Pi_{COT}(x_{\mathsf{R}}^{(0)}, \Sigma)$	**return** $\Pi_{COT}(x_{\mathsf{R}}^{(0)}, \Sigma)$

The formal description is as follows:

$\mathrm{Game}_1^{\sigma}(\kappa)$ is the original game where the sender is $\mathsf{P_S}$ and the receiver is $\mathsf{P_R}$.

$\mathrm{Game}_2^{\sigma}(\kappa)$ is as $\mathrm{Game}_1^{\sigma}(\kappa)$ but in providing the transcript the simulator follows Item $\mathrm{Game}_1^{\sigma}(\kappa)$ for the first round but inserts $x_{\mathsf{S}}^{(1)}$ in the name of the *dummy friend* and $x_{\mathsf{D}}^{(1)}$ in the name of the *sender* into $\mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$. Note that this corresponds to the final distribution from Fig. 21.

Lemma 61. *Let PKE be an IND$-CCA secure public-key encryption scheme. Let SKE be an IND$-CCA secure secret-key encryption scheme. Let SIG be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_1^{\sigma}(\kappa)$ *and* $\mathrm{Game}_2^{\sigma}(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_1^{\sigma}(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_2^{\sigma}(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. We reduce to the δ-anonymity of $\mathcal{F}_{AT}^{(2,3)}$. If there was some distinguisher $\mathcal{D}$ who can distinguish $\mathrm{Game}_1^{\sigma}(\kappa)$ from $\mathrm{Game}_2^{\sigma}(\kappa)$ with advantage $1/2 + \alpha$ then there is some adversary $\mathcal{A}$ (which implies a guessing algorithm A) who can determine the sender with advantage α. The limit δ then provides an upper bound on α.

Creating the Transcript. We denote by $\mathcal{C}$ the challenger for the anonymity-game of $\mathcal{F}_{AT}^{(2,3)}$. $\mathcal{A}$ determines the message to-be-transferred as $\Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$, that is, the response sent by a sender in Π_{COT} after having received the first message by the receiver. Note that Σ is the result of the first $\mathcal{F}_{AT}^{(2,3)}$ instance where R inserts the message, and we take the output of that which only corresponds to the desired message $\Pi_{COT\mathsf{R}}(\sigma)$ with probability $1/2 + \varepsilon/2$. However, we stress that this is not important to our proof, as the relevant part is only the insertion of that message as transferred message into the instance of $\mathcal{F}_{AT}^{(2,3)}$ where $\mathsf{P_R}$ is the receiver and which is played with the challenger $\mathcal{C}$.

$\mathcal{C}$ returns a transcript of the $\mathcal{F}_{AT}^{(2,3)}$ instance, which $\mathcal{A}$ inserts into the transcript provided to the distinguisher $\mathcal{D}$ instead of honestly simulating the second $\mathcal{F}_{AT}^{(2,3)}$ instance where the receiver R of the AT is also the receiver of the OT, the remainder is simulated as-is.

Translating the Result. Eventually $\mathcal{D}$ returns a guess which is either $\text{Game}_1^\sigma(\kappa)$ or $\text{Game}_2^\sigma(\kappa)$. In case $\mathcal{D}$ assumes to be in $\text{Game}_1^\sigma(\kappa)$ we assume the party we refer to as $\mathsf{P_S}$ is the sender, and if $\mathcal{D}$ assumes to be in $\text{Game}_2^\sigma(\kappa)$ we let $\mathcal{A}$ refer to $\mathsf{P_D}$ as the sender in the AT.

As can be seen the distributions match perfectly, as the only change is the sender and that one perfectly translates to our problem of de-anonymization.

Thus it follows that $\mathcal{D}$ is correct with probability

$$\begin{aligned} &\Pr[\mathcal{D} \text{ correct}] \leq 1/2 + \alpha \\ \implies &|\Pr[\mathcal{D} \text{ correct}] - 1/2| = |\alpha| = \alpha \leq (1-\delta)/2 \end{aligned} \tag{62}$$

and thus if the adversary corrupts the sender then the protocol provides δ-Anonymity where δ is inherited from $\mathcal{F}_{AT}^{(2,3)}$. □

P_0 sender	P_1 sender
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_D}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_D}$:
$(x_{COT^\mathsf{R}}, \bot)$	$(x_{COT^\mathsf{R}}, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:
$(x_{COT^\mathsf{R}}, \bot)$	$(\bot, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:
$(\bot, \bot)$	$(x_{COT^\mathsf{R}}, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_D}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_D}$:
$(x_{COT^\mathsf{S}}, \bot)$	$(x_{COT^\mathsf{S}}, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_0}$:
$(\bot, \bot)$	$(x_{COT^\mathsf{S}}, \bot)$
$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:	$\mathcal{F}_{AT}^{(2,3)}$ with receiver $\mathsf{P_1}$:
$(x_{COT^\mathsf{S}}, \bot)$	$(\bot, \bot)$

Fig. 22: The two distributions we have to prove indistinguishable in case the adversary A_0 picks the dummy friend.

Corrupted Dummy Friend If the adversary corrupts the dummy friend after the execution of the Undetectable Oblivious Transfer protocol then it is guaranteed that the two remaining parties are the sender and the receiver.

$$\begin{aligned} &\{(\Sigma_0, \Sigma_1) \xleftarrow{\$} \{0,1\}^{2\ell}, \sigma \xleftarrow{\$} \{0,1\} : \pi \xleftarrow{\$} \langle \mathsf{P}_0, \mathsf{P}_1, \mathsf{P_D} \rangle((\Sigma_0, \Sigma_1), \sigma, \bot)\} \\ \approx &\{(\Sigma_0, \Sigma_1) \xleftarrow{\$} \{0,1\}^{2\ell}, \sigma \xleftarrow{\$} \{0,1\} : \pi \xleftarrow{\$} \langle \mathsf{P}_0, \mathsf{P}_1, \mathsf{P_D} \rangle(\sigma, (\Sigma_0, \Sigma_1), \bot)\} \end{aligned} \tag{63}$$

However, distinguishing which is which is hard; to prove this claim we use the following games:

$\mathrm{Game}_1(\kappa)$	
1 :	$\mathsf{P_S}$:
2 :	$x_{\mathsf{S}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
3 :	$\mathsf{P_R}$:
4 :	$x_{\mathsf{R}}^{(0)} \leftarrow \Pi_{COT\mathsf{R}}(\sigma)$
5 :	$\mathsf{P_D}$:
6 :	$x_{\mathsf{D}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
7 :	$\mathsf{P_i}$:
8 :	$\forall_{j \neq i}$: $x_i^{(0)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
9 :	$\mathsf{P_S}$:
10 :	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{S})$
11 :	$x_{\mathsf{S}}^{(1)} := \Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$
12 :	$\mathsf{P_R}$:
13 :	–
14 :	$x_{\mathsf{R}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
15 :	$\mathsf{P_D}$:
16 :	$x_{\mathsf{D}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
17 :	$\mathsf{P_i}$:
18 :	$\forall_{j \neq i}$: $x_i^{(1)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
19 :	$\mathsf{P_R}$:
20 :	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$
21 :	**return** $\Pi_{COT}(x_{\mathsf{R}}^{(0)}, \Sigma)$

$\mathrm{Game}_2(\kappa)$	
1 :	$\mathsf{P_S}$:
2 :	$x_{\mathsf{S}}^{(0)} \leftarrow \Pi_{COT\mathsf{R}}(\sigma)$
3 :	$\mathsf{P_R}$:
4 :	$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
5 :	$\mathsf{P_D}$:
6 :	$x_{\mathsf{D}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
7 :	$\mathsf{P_i}$:
8 :	$\forall_{j \neq i}$: $x_i^{(0)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
9 :	$\mathsf{P_S}$:
10 :	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{S})$
11 :	$x_{\mathsf{S}}^{(1)} := \Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$
12 :	$\mathsf{P_R}$:
13 :	–
14 :	$x_{\mathsf{R}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
15 :	$\mathsf{P_D}$:
16 :	$x_{\mathsf{D}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
17 :	$\mathsf{P_i}$:
18 :	$\forall_{j \neq i}$: $x_i^{(1)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
19 :	$\mathsf{P_R}$:
20 :	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$
21 :	**return** $\Pi_{COT}(x_{\mathsf{R}}^{(0)}, \Sigma)$

$\mathrm{Game}_2(\kappa)$	
1 :	$\mathsf{P_S}$:
2 :	$x_{\mathsf{S}}^{(0)} \leftarrow \Pi_{COT\mathsf{R}}(\sigma)$
3 :	$\mathsf{P_R}$:
4 :	$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
5 :	$\mathsf{P_D}$:
6 :	$x_{\mathsf{D}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
7 :	$\mathsf{P_i}$:
8 :	$\forall_{j \neq i}$: $x_i^{(0)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
9 :	$\mathsf{P_S}$:
10 :	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{S})$
11 :	$x_{\mathsf{S}}^{(1)} := \Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$
12 :	$\mathsf{P_R}$:
13 :	–
14 :	$x_{\mathsf{R}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
15 :	$\mathsf{P_D}$:
16 :	$x_{\mathsf{D}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
17 :	$\mathsf{P_i}$:
18 :	$\forall_{j \neq i}$: $x_i^{(1)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
19 :	$\mathsf{P_R}$:
20 :	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$
21 :	**return** $\Pi_{COT}(x_{\mathsf{R}}^{(0)}, \Sigma)$

$\mathrm{Game}_3(\kappa)$	
1 :	$\mathsf{P_S}$:
2 :	$x_{\mathsf{S}}^{(0)} \leftarrow \Pi_{COT\mathsf{R}}(\sigma)$
3 :	$\mathsf{P_R}$:
4 :	$x_{\mathsf{R}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
5 :	$\mathsf{P_D}$:
6 :	$x_{\mathsf{D}}^{(0)} \xleftarrow{\$} \{0,1\}^m$
7 :	$\mathsf{P_i}$:
8 :	$\forall_{j \neq i}$: $x_i^{(0)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
9 :	$\mathsf{P_S}$:
10 :	–
11 :	$x_{\mathsf{S}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
12 :	$\mathsf{P_R}$:
13 :	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$
14 :	$x_{\mathsf{R}}^{(1)} := \Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$
15 :	$\mathsf{P_D}$:
16 :	$x_{\mathsf{D}}^{(1)} \xleftarrow{\$} \{0,1\}^m$
17 :	$\mathsf{P_i}$:
18 :	$\forall_{j \neq i}$: $x_i^{(1)} \rightarrow \mathcal{F}_{AT}^{(2,3)}(j)$
19 :	$\mathsf{P_R}$:
20 :	$\Sigma \leftarrow \mathcal{F}_{AT}^{(2,3)}(\mathsf{R})$
21 :	**return** $\Pi_{COT}(x_{\mathsf{R}}^{(0)}, \Sigma)$

$\mathrm{Game}_1^{\sigma}(\kappa)$ is the original game where the sender is $\mathsf{P_S}$ and the receiver is $\mathsf{P_R}$.
$\mathrm{Game}_2^{\sigma}(\kappa)$ is as $\mathrm{Game}_1^{\sigma}(\kappa)$ but in providing the transcript the simulator follows $\mathrm{Game}_1^{\sigma}(\kappa)$ for the first round but inserts $x_{\mathsf{R}}^{(0)}$ in the name of the *sender* and $x_{\mathsf{S}}^{(0)}$ in the name of the *receiver* into $\mathcal{F}_{AT}^{(2,3)}(\mathsf{D})$.

Lemma 62. *Let* PKE *be an IND$-CCA secure public-key encryption scheme. Let* SKE *be an IND$-CCA secure secret-key encryption scheme. Let* SIG *be an sEUF-CMA secure signature*

scheme. For all PPT guessing algorithms A*, the distinguishing advantage for* $\mathrm{Game}_1^\sigma(\kappa)$ *and* $\mathrm{Game}_2^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_1^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_2^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. We reduce to the δ-anonymity of $\mathcal{F}_{AT}^{(2,3)}$. If there was some distinguisher $\mathcal{D}$ who can distinguish $\mathrm{Game}_1^\sigma(\kappa)$ from $\mathrm{Game}_2^\sigma(\kappa)$ with advantage $1/2 + \alpha$ then there is some adversary $\mathcal{A}$ (which implies a guessing algorithm A) who can determine the sender with advantage α. The limit δ then provides an upper bound on α.

Creating the Transcript. We denote by $\mathcal{C}$ the challenger for the anonymity-game of $\mathcal{F}_{AT}^{(2,3)}$. $\mathcal{A}$ determines the message to-be-transferred as $\Pi_{COT\mathsf{R}}(\sigma)$, that is, the first message sent by a receiver in Π_{COT} who wants to receive x_σ. From that $\mathcal{C}$ returns a transcript of the $\mathcal{F}_{AT}^{(2,3)}$ instance, which $\mathcal{A}$ inserts into the transcript provided to the distinguisher $\mathcal{D}$ instead of honestly simulating the first $\mathcal{F}_{AT}^{(2,3)}$ instance where the sender S of the AT is the dummy friend, the remainder is simulated as-is.

Translating the Result. Eventually $\mathcal{D}$ returns a guess which is either $\mathrm{Game}_1^\sigma(\kappa)$ or $\mathrm{Game}_2^\sigma(\kappa)$. In case $\mathcal{D}$ assumes to be in $\mathrm{Game}_1^\sigma(\kappa)$ we assume the party we refer to as P_R is the sender, and if $\mathcal{D}$ assumes to be in $\mathrm{Game}_2^\sigma(\kappa)$ we let $\mathcal{A}$ refer to P_S as the sender in the AT.
As can be seen the distributions match perfectly, as the only change is the sender and that one perfectly translates to our problem of de-anonymization.
Thus it follows that $\mathcal{D}$ is correct with probability

$$\begin{aligned} &\Pr[\mathcal{D} \text{ correct}] \leq 1/2 + \alpha \\ \implies &|\Pr[\mathcal{D} \text{ correct}] - 1/2| = |\alpha| = \alpha \leq (1-\delta)/2 \end{aligned} \tag{64}$$

and thus if the adversary corrupts the sender then the protocol provides δ-Anonymity where δ is inherited from $\mathcal{F}_{AT}^{(2,3)}$. □

$\mathrm{Game}_3^\sigma(\kappa)$ is as $\mathrm{Game}_2^\sigma(\kappa)$ but in providing the transcript the simulator follows $\mathrm{Game}_2^\sigma(\kappa)$ but inserts $x_\mathsf{S}^{(1)}$ in the name of the *receiver* and $x_\mathsf{R}^{(1)}$ in the name of the *sender* into $\mathcal{F}_{AT}^{(2,3)}(\mathsf{D})$.

Lemma 63. *Let* Π_{KE} *be an IND$-CCA secure public-key encryption scheme. Let* Π_{SKE} *be an IND$-CCA secure secret-key encryption scheme. Let* Π_{SIG} *be an sEUF-CMA secure signature scheme. For all PPT guessing algorithms* A*, the distinguishing advantage for* $\mathrm{Game}_2^\sigma(\kappa)$ *and* $\mathrm{Game}_3^\sigma(\kappa)$ *is bounded by:*

$$|\Pr[\mathsf{out}_{\mathrm{Game}_2^\sigma(\kappa)} = 1] - \Pr[\mathsf{out}_{\mathrm{Game}_3^\sigma(\kappa)} = 1]| \in \mathsf{negl}(\kappa)$$

Proof. We reduce to the δ-anonymity of $\mathcal{F}_{AT}^{(2,3)}$. If there was some distinguisher $\mathcal{D}$ who can distinguish $\mathrm{Game}_2^\sigma(\kappa)$ from $\mathrm{Game}_3^\sigma(\kappa)$ with advantage $1/2 + \alpha$ then there is some adversary $\mathcal{A}$ (which implies a guessing algorithm A) who can determine the sender with advantage α. The limit δ then provides an upper bound on α.

Creating the Transcript. We denote by $\mathcal{C}$ the challenger for the anonymity-game of $\mathcal{F}_{AT}^{(2,3)}$. $\mathcal{A}$ determines the message to-be-transferred as $\Pi_{COT\mathsf{S}}(x_0, x_1, \Sigma)$, that is, the response sent by a sender in Π_{COT} after having received the first message by the receiver. Note that Σ is the result of the first $\mathcal{F}_{AT}^{(2,3)}$ instance where S inserts the message and sends it to R since $\mathrm{Game}_2^\sigma(\kappa)$ (where the roles for the first round has been changed), and we take the output of that which only corresponds to the desired message $\Pi_{COT\mathsf{R}}(\sigma)$ with probability $1/2 + \varepsilon/2$. However, we stress that this is not important to our proof, as the relevant part is only the insertion of that message as transferred message into the instance of $\mathcal{F}_{AT}^{(2,3)}$ where P_R is the receiver and which is played with the challenger $\mathcal{C}$.
$\mathcal{C}$ returns a transcript of the $\mathcal{F}_{AT}^{(2,3)}$ instance, which $\mathcal{A}$ inserts into the transcript provided to the distinguisher $\mathcal{D}$ instead of honestly simulating the second $\mathcal{F}_{AT}^{(2,3)}$ instance where the receiver R of the AT is also the receiver of the OT, the remainder is simulated as-is.

Translating the Result. Eventually $\mathcal{D}$ returns a guess which is either $\text{Game}_2^\sigma(\kappa)$ or $\text{Game}_3^\sigma(\kappa)$. In case $\mathcal{D}$ assumes to be in $\text{Game}_2^\sigma(\kappa)$ we assume the party we refer to as $\mathsf{P_S}$ is the sender, and if $\mathcal{D}$ assumes to be in $\text{Game}_3^\sigma(\kappa)$ we let $\mathcal{A}$ refer to $\mathsf{P_R}$ as the sender in the AT.
As can be seen the distributions match perfectly, as the only change is the sender and that one perfectly translates to our problem of de-anonymization.
Thus it follows that $\mathcal{D}$ is correct with probability

$$\begin{aligned} &\Pr[\mathcal{D} \text{ correct}] \leq 1/2 + \alpha \\ \implies &|\Pr[\mathcal{D} \text{ correct}] - 1/2| = |\alpha| = \alpha \leq (1-\delta)/2 \end{aligned} \tag{65}$$

and thus if the adversary corrupts the sender then the protocol provides δ-Anonymity where δ is inherited from $\mathcal{F}_{AT}^{(2,3)}$. □

Given that this is the least efficient strategy it thus follows that the anonymity is bounded by δ for any possible A_0 corrupting any party.

References

[BDP+98] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer, Heidelberg, August 1998.

[CLT+15] R. Canetti, H. Lin, S. Tessaro, and V. Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 468–497. Springer, Heidelberg, March 2015.

[Gol04] O. Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004.

[JLL+22] A. Jain, H. Lin, J. Luo, and D. Wichs. The pseudorandom oracle model and ideal obfuscation. Cryptology ePrint Archive, Report 2022/1204, 2022. https://eprint.iacr.org/2022/1204.

[Rog04] P. Rogaway. Nonce-based symmetric encryption. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 348–359. Springer, Heidelberg, February 2004.

[vHL05] L. von Ahn, N. J. Hopper, and J. Langford. Covert two-party computation. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 513–522. ACM Press, May 2005.

Acronyms

AT Anonymous Transfer
CMPC Covert Multiparty Computation
COT Covert Oblivious Transfer
CRS Common Reference String
EUF-CMA Existential Unforgery under Chosen Message Attacks
IND-CCA Indistinguishability under Chosen Ciphertext Attacks
IND$ Indistinguishability from Random Bits
IND$-CCA Indistinguishability from Random Bits under Chosen Ciphertext Attacks
IND$-CPA Indistinguishability from Random Bits under Chosen Plaintext Attacks
ITM Interactive Turing Machine
LR Left-Right
MPC Multiparty Computation
NM Non-Malleability
NM-CCA Non-Malleability under Chosen Ciphertext Attacks
OT Oblivious Transfer
OTP One-Time-Pad
PKE Public Key Encryption
PPT Probabilistic Polynomial-Time
PRF Pseudorandom Function
ROM Random Oracle Model
sEUF-CMA Strong Existential Unforgeability under Chosen Message Attacks
SKE Symmetric Key Encryption
SR Silent Receiver
U2PC Undetectable Two-Party Computation
UMPC Undetectable Multiparty Computation
UOT Undetectable Oblivious Transfer
VBB Virtual Black-Box Obfuscation

Symbols

$\mathcal{A}$: The adversary in a cryptographic framework.
A: An algorithm that *guesses* the identity of the participating player of any AT.
α: The advantage of a distinguisher over guessing.
B: An algorithm that *guesses* the identity of the participating player of any AT.
b: A value between 1 and N describing which player P_b participates in the protocol.
$b_{\mathcal{C}}$: The challenge player in a game-based proof.
Ber: A bernoulli-distribution which returns 1 with probability p.
β: The bit that is sampled by a challenger $\mathcal{C}$ in a security game which has to be guessed by the adversary.
C: The constant factor that binds a given function in Landau notation. That is, $f \in \mathcal{O}(g) \iff \lim_{x\to\infty}|f(x)|/|g(x)| \leq C$.
$\mathfrak{C}$: A function class, that is, the set of functions that belong in a certain complexity class.
$\mathcal{C}$: The challenger in a game-based proof.
ct: Cipher text, that is, an encryption of a message.
c: Number of rounds of an interactive protocol.
χ: The current round of an interactive protocol.
$coin$: The result of a coin toss.
$\mathsf{Cointoss}$: A function realizing a (biased) coin toss. We write $\mathsf{Cointoss}_{(p)}(r)$ to denote the coin toss that returns tails with probability p and tails with probability $(1-p)$, where the randomness is taken from a PRF on input r.
$\mathsf{CointossS}$: A function realizing a (biased) coin toss. We write $\mathsf{CointossS}^{(r)}_{(p)}(x,y)$ to denote the coin toss that returns x with probability p and y with probability $(1-p)$, where the randomness is taken from a PRF on input r.
crs: A common reference string.
D: The dummy friend who is not participating in any protocol but is still present for some reason.
$\mathcal{D}$: The distinguisher, *i.e.* a PPT ITM which distinguishes two distributions.
D: A distribution.
d: Tweaking parameter which acts as a compromise between anonymity and correctness; if $d = 0$, we have a perfectly correct protocol, but if $d = 1$, we have a perfectly anonymous protoocl.
Dec: A decryption protocol.
δ: The measurement of anonymity for an AT or UOT protocol.
$\mathsf{d_{TV}}$: The Total Variational Distance, defined over two probability distributions p and q as $\mathsf{d_{TV}}(p,q) \coloneqq \frac{1}{2}\sum_i |p_i - q_i|$.
Enc: An encryption protocol.
ε: The measurement of correctness for an AT or UOT protocol.
$\mathsf{Evaluate}$: The method of an obfuscator $\mathbb{O}$ to evaluate a given program given only the handle.
Exp: An experiment as part of a game.
F: A Pseudorandom Function.
$\mathcal{F}$: The functionality in a simulation-based framework.
f: The to-be-computed function for deniable secure computation.
$\mathcal{F}_{AT}^{(2,3)}$: Ideal functionality for realizing Anonymous Transfer with three parties, two of which are aware that the protocol is executed.
Game: The game in a game-hop based proof.
H: The Hellinger Distance, defined over two probability distributions p and q as $\mathsf{H}(p,q) \coloneqq \frac{1}{\sqrt{2}}\sqrt{\sum_i \left(\sqrt{p_i} - \sqrt{q_i}\right)^2}$.
H: A hybrid, that is, a minor change in a simulation which we claim to be not efficiently noticable.
h: The handle an obfuscator returns that enables evaluating the program.
$heads$: The heads-side of a coin toss.
k: The signing key of a signature scheme, can be used to sign a given message such that third parties can verify the authenticity.
k: The number of players, that is, participants.

κ: The security parameter.
KeyGen: A key generation protocol.
ℓ: The length of the message.
m: The length of the protocol messages.
μ: The signature of a given message.
$\mu_{\mathcal{C}}$: The signature used in the challenge transcript.
N: The number of individuals, that is, (non-)participants.
negl: The set of negligible functions with respect to a given argument.
$\mathbb{O}$: An ideal obfuscator that obfuscates a given program.
O: An oracle.
Obf: An ideal obfuscation scheme.
Obfuscate: The method of an obfuscator $\mathbb{O}$ to obfuscate a given program.
OT: Transfer algorithm for an UOT.
Ot: An Oblivious Transfer scheme.
OTP: A one-time pad.
$OTP_{\mathcal{C}}$: The used One-Time-Pad in the challenge transcript.
out: Output of a game.
owhl: The set of overwhelming functions with respect to a given argument.
P: An individual present during the protocol execution.
P: An obfuscated program that takes as input a transcript π outputs a message Σ for the receiver.
Π: A protocol.
ϖ: A random permutation on a given set.
π: Transcript of a protocol.
Π^1_{AT}: A protocol realizing single-bit Anonymous Transfer.
Π^{ℓ}_{AT}: A protocol realizing ℓ-bit Anonymous Transfer.
Π^{κ}_{AT}: A protocol realizing κ-bit Anonymous Transfer.
$\pi_{\mathcal{C}}$: The challenge transcript of a game-based proof.
Π_{COT}: A protocol for realizing Covert Oblivious Transfer.
Π^{ℓ}_{SR}: The Silent Receiver AT protocol, *i.e.* the AT protocol where the receiver sends no messages.
Π_{UMPC}: The Undetectable MPC protocol that allows k parties to hide a computation inside innocent-looking conversations of N individuals.
Π_{UOT}: A protocol for realizing Undetectable Oblivious Transfer.
pk: The public key of an encryption scheme, can be used to encrypt a message into a cipher text.
Pke: A public-key encryption scheme.
poly: An arbitrary polynomial.
p: Probability
R: The receiving party.
r: A random / undefined bitstring.
Rand: The rerandomization of the last round of an AT.
Reconstruct: Reconstruction algorithm for an AT.
$\mathbb{S}$: A symmetric group, that is, the group of elements alongside all bijections/permutations onto itself.
S: The sending party.
$\mathcal{S}$: The simulator in a cryptographic framework.
Setup: The setup algorithm that creates a Common Reference String.
Sig A signature scheme.
Sig: The function to compute the signature of a given message.
Σ: The to-be-transferred message of an AT or UOT.
σ: The to-be-transferred bit of an AT.
ς: The measurement of secrecy for an AT, which is high if the non-receiver is unable to reconstruct the transferred bit better than by guessing.
$\Sigma_{\mathcal{C}}$: The transferred message in the challenge transcript.
$\sigma_{\mathcal{C}}$: The transferred bit in the challenge transcript.
sk: The secret key of an encryption scheme, can be used to decrypt a cipher text.
Ske: A (symmetric) encryption scheme.
st: The (secret) state of a party.
Supp: The support of a distribution.

$\mathfrak{T}$: The runtime of a turing machine.
T: The (random) tape of a player.
t: The number of samples an adversary is given from a given distribution.
tails: The tails-side of a coin toss.
Transfer: The transfer protocol for an AT.
$\mathcal{U}$: The uniform distribution.
Vfy: The verification function of a given MAC or Signature scheme.
vk: The verification key of a signature scheme, suffices to check authenticity of a signature over a known message.
$\mathsf{vk}_{\mathcal{C}}$: The verification key used in the challenge transcript.
X: Parameterized by a value $b \in [N-1]$, this is the value obtained in the program P_{AT} after decrypting the message x_b with the supposed secret key sk_b.
x: Message sent during a protocol execution.
$x^{\mathcal{C}}$: A message reported in the challenge transcript.
ξ: The slack, that is, the difference between two variables x and y: if $x < y$, we can write $x + \xi = y$, allowing for an easier analysis.