Taming Delegations in Anonymous Signatures: k-Times Anonymity for Proxy and Sanitizable Signature

Xavier Bultel, Charles Olivier-Anclin

Taming Delegations in Anonymous Signatures: *k*-Times Anonymity for Proxy and Sanitizable Signature

Xavier Bultel[1][0000−0002−8309−8984] and Charles Olivier-Anclin[1,2,3][0000−0002−9365−3259]

[1] LIFO, Université d'Orléans, INSA Centre Val de Loire, Inria, Bourges, France
[2] Université Clermont Auvergne, LIMOS, CNRS, Clermont-Ferrand, France
[3] be ys Pay

Abstract. Fully traceable k-times anonymity is a security property concerning anonymous signatures: if a user produces more than k anonymous signatures, its identity is disclosed and all its previous signatures can be identified. In this paper, we show how this property can be achieved for delegation-supported signature schemes, especially proxy signatures, where the signer allows a delegate to sign on its behalf, and sanitizable signatures, where a signer allows a delegate to modify certain parts of the signed messages. In both cases, we formalize the primitive, give a suitable security model, provide a scheme and then prove its security under the DDH assumption. The size of the keys/signatures is logarithmic in k in our two schemes, making them suitable for practical applications, even for large k.

1 Introduction

Proxy signature [25], which enables the signer to delegate the ability to sign messages on its behalf to a delegate, is a standard cryptographic primitive that has attracted a great deal of interest in recent decades. In some contexts, it is preferable to hide the delegate's identity from the signature verifier. Such a signature is called an *anonymous proxy signature* [18]. A trivial way of achieving this property is to give the delegate the signing key directly, however, this technique allows the delegate to impersonate the signer without any constraint, which is clearly not desirable. The signer therefore needs a way of tracing its delegates if one of them abuses their power. This leads to two inherent issues: the signer must be active to manage the trace, and must have access to the signatures.

The concept of traceable k-times anonymity offers an alternative way to delegate tracing. Signature schemes following this paradigm allow users to create k signatures anonymously. If they exceed this limit, a verifier can then publicly link two signatures and trace the identity of the signer. This property has been defined for *ring signatures* [19], *group signatures* [2] and *anonymous authentication* [26]. Moreover, a k-times signature is said to be *fully traceable* when the verifier can retrieve all the signatures generated by the signer which has exceeded the k limit *a posteriori*. To the best of our knowledge, this more powerful property has only been defined for ring signatures [7].

However, k-times anonymity has never been applied directly to proxy signatures, even though they seem naturally suited to this property. This would enable a verifier, which has access to all signatures, to publicly trace dishonest proxies on its own, while preserving the anonymity of honest proxies, without the intervention of the signer. In this paper, we close this gap by modeling and instantiating the first fully traceable k-times anonymous proxy signature.

On the other hand, *sanitizable signatures* [1] are conceptually close to proxy signatures: in this primitive, the delegate (called the sanitizer) can no longer produce signatures by itself, but can modify certain parts of a signed message. When considering a setting where the sanitizer must remain anonymous, the same problems arise as with proxy signatures. Applying a similar approach, we propose the first fully traceable k-times anonymous sanitizable signatures.

Contributions and Technical Overview. We give a formal definition, a security model, and an efficient scheme (in term of size of the keys/signatures) for fully traceable k-times anonymous proxy signatures and fully traceable k-times anonymous sanitizable signatures. We give security proofs for these schemes. From a technical point of view, we rely on the method proposed in [7]: the delegate has k different public/secret

keys; if it reuses the same key twice, then it is possible to link the two signatures to the user and extract an element that links all its other signatures. However, this method requires a number of keys linear in k; our main technical contribution is a method for generating k distinct and mutually unlinkable keys from $2\log_2(k)$ keys only. The idea is to compose, at the i^{th} signature, the keys corresponding to the bits of i to obtain a new public/secret key pair. These keys must be certified by the delegator, but must be unlinkable. To achieve these two properties simultaneously, we use a signature on equivalence class [17], which allows the delegate to randomize the $2\log_2(k)$ keys while maintaining the validity of their certificate. This method requires the creation of an ad-hoc zero-knowledge proof ensuring the verifier that the delegate has correctly generated its key. For the special case where k is not a power of 2, we build another ad-hoc zero-knowledge proof to ensure that i is indeed less than k. Both of these proofs have logarithmic complexity in size, enabling us to obtain logarithmic complexity in size for both our keys and our signatures. This method is fairly generic, so we think it could be of independent interest in other primitives requiring the generation of several certified keys. Our sanitizable signature scheme uses the same technique as the one proposed in [3] combined with the method described above to make it k-times anonymous. The main technical challenge here is to adapt the signature to enable the signer to simulate the use of the $2\log_2(k)$ keys in the original signature, so that it is not possible to determine whether it has been sanitized or not.

For each signature primitive, we define the following properties in addition to unforgeability:

Anonymity: signatures are anonymous as long as the delegate does not exceed k signatures. In particular, they cannot be linked to each other.

Traceability: if the delegate exceeds the k signatures limit, it cannot prevent anyone from linking all its signatures and recovering its identity.

Non-framability: a delegate cannot produce a signature that can be traced back to someone else.

We also adapt the security properties of sanitizable signatures:

Immutability: it is not possible to modify parts of messages that are not intended to be modified.

Transparency: it is not possible to guess whether a signature has been sanitized or not. This property implies privacy: it is not possible to determine any information about the original message.

Unlinkability: it is not possible to link a sanitized signature to the original signature, or to link sanitized signatures from the same original signature. A few schemes such as [16] achieve this property. Note that unlinkability differs from anonymity, which ensures that it is not possible to link signatures from the same user. We provide more details about this on Section 6.

Invisibilty: it is not possible to identify which part of the message is modifiable. Note that designing schemes that are both unlinkable and invisible is challenging, and there are only two schemes in the literature that combine these properties [8, 3].

Motivations. Anonymous proxy signatures are used wherever an entity wishes to delegate the ability to sign on its behalf to others, without making the delegation policy transparent to the recipient of the messages. Anonymity can also protect proxies when their identity must remain secret, for example in legal proceedings where retaliation is possible. Conversely, anonymity provides a high level of protection for proxies who might be tempted to abuse their power. The fully traceable k-times anonymity property can significantly limit this, even in the absence of the delegate. Sanitizable signatures extend proxy signatures by adding a degree of control over the messages sent by delegates. For example, they can be used to force the use of message templates.

For instance, consider a manager who delegates the ability to sign and send emails on their behalf from their email address to multiple entities. These could be employees or servers that automatically send emails that contain, depending on their role, specific messages, appointments, contracts, payments, invoices, reminders, summonses or other legal or commercial documents. If too many emails are being sent from the same entity on behalf of the manager, the company's mail server can use the k-times mechanism to locate the offending entity, block the emails it is sending, list all its signatures and alert the manager and anyone else who has received emails from this entity in the past. Note that in our case the server is honest but curious: we trust it to check signatures and detect anomalies (it cannot be fully corrupted by an active attacker),

but the information it processes does not allow it to learn anything about the identity of honest proxies or the delegation policy (a passive attacker can observe everything that passes through the server without compromising anonymity).

To control the content of messages, it is helpful to use sanitizable signatures that force delegates to use templates. For example, by setting the metadata it is possible to allow emails to be sent only to certain people, on certain dates, with certain subjects, or by forcing the addition of copy users who can check the content of the email. In the case of automatic emails, such as invoices, it is possible to impose a very precise template where only the customer's name, date and amount can be changed. Note that thanks to the security properites of our sanitizable signatures (transparency, anonymity, unlinkability, and invisibility), the company's delegation policy remains entirely private from the point of view of the verifiers and the mail server as long as the k limit is not exceeded,

Related works. Anonymous proxy signatures have been introduced by Fuchsbauer and Pointcheval [18]. Since then, several other anonymous proxy signature schemes have been proposed [27, 28]. However, as mentioned above, they all consider active traceability management by the original signer or a dedicated semi-trusted proxy. Note that unlike our scheme, Fuchsbauer and Pointcheval's scheme allows hierarchical management of proxies (a delegate can allow a sub-delegate to sign in its place, etc.). This feature could naturally be achieved by extending our scheme, despite a linear growth in the number of delegations. This function is left outside of the scope of this work.

k-times anonymity has been introduced for authentication, group signature (where the group is managed by an authority that generates keys), and ring signature (where the group is chosen ad-hoc at the time of signing) in [26], [2], and [19] respectively. In some schemes, the identity of the signer leaks if it produces more than k signatures. Fully traceable k-times anonymity [7] extends this concept by making it possible to trace all signatures produced a priori by the user that exceed the k signatures (and not just a pair of signatures). To the best of our knowledge, the only scheme that matches this property is the ring signature described in [7], and this at the cost of a signature size in $O(nk)$ where n is the number of users, and a secret key size in $O(k)$.

In [18], Fuchsbauer and Pointcheval mention that anonymous proxy signatures can be seen as group signatures: the delegator becomes the group manager and each delegate (*i.e.,* each group member) can sign anonymously on behalf of the manager (*i.e.,* within the group). We can therefore see our fully traceable k-times proxy signature as the first fully traceable k-times group signature. Since our aim is also to design sanitizable signatures, which have some similarities with proxy signatures (in both cases a delegator gives a delegate the power to create new signatures on his behalf), we have chosen to present our scheme as an anonymous proxy signature rather than as a group signature. In comparison with the only k-times group signature scheme [2] in the literature, our scheme achieves full traceability, in return the key/signature size is in $O(\log(k))$ whereas [2] claims a constant key/signature size (note, however, that in this scheme, the delegator must produce and share a public key of size linear in k, moreover if the limit k is different for each delegate, then this key must be kept secret by each delegate, which significantly restrains its practicability for large k).

Sanitizable signatures were introduced by Ateniese *et al.* in [1], who identified several security properties (unforgeability, immutability, privacy, transparency, and accountability) later formally defined in [4]. They show that privacy (the original message does not leak from the sanitized signature) is implied by transparency. Invisibility was also introduced in [1] but received formal treatment much later in [9]. Last but not least, unlinkability has been introduced and formalized in [5] and studied in [6, 16]. Only two schemes guarantee all these properties at once [8, 3]. In this paper, we adapt and prove all these properties on our scheme, with the exception of accountability, which consists in allowing the signer to reveal the author (*i.e.,* the original signer or the sanitizer) of a problematic signature, since this information leaks spontaneously if the sanitizer exceeds the limit of k sanitizations.

Traceable k-times anonymous proxy signatures should not be confused with the k-times (not anonymous) proxy signatures introduced by Liu *et al.* in [24], where if the (non-anonymous) proxy exceeds a limit of k signatures, then its secret key leaks. This primitive is close to ours, but differs in two crucial points: (i) the proxy is not anonymous, so there is no need to trace it or link signatures, thus full traceability makes no sense

in [24], and (ii) unlike [24] we do not want to leak the proxy's secret key for security reasons. Indeed, if a verifier recovers a proxy secret key, it can sign messages as a proxy without the original signer having chosen to give it this power. As a result, users which have not had access to the $k+1$ proxy signatures (including the original signer) are unaware that this verifier can impersonate the signer, which causes serious security problems in most applications. Similarly, one line of work, started in [22], aims to limit the sanitizer's power in various ways in sanitizable signatures [11]. In particular, in [22, 11] the authors propose a scheme where if the (non-anonymous) sanitizer exceeds a limit of k signatures, then its secret key leaks, as in k-times (not anonymous) proxy signatures [24]. The differences between this primitive and ours are the same as those between k-times proxy signatures [24] and our k-times anonymous proxy signatures.

Finally, k-times anonymous sanitizable signatures should not be confused with γ-times sanitizable signatures [3], where γ bounds the number of blocks that can be modified instead of the number of times the signature can be sanitized. In the primitive introduced in [3], the sanitizer is not anonymous, and cannot (in the computational sense) sanitize a signature by modifying more than γ blocks. The mechanism is therefore very different, as there is no intention of triggering some secret information leak when the limit is exceeded.

2 Preliminaries

Notations. $r \leftarrow\!\!\$\, S$ means that r is chosen uniformly at random over the set S and $|S|$ is the cardinal of S. The operator $\xrightarrow{p}$ denotes the parsing of a tuple or a set of elements. We denote by $y \leftarrow \mathcal{A}(x)$ the execution of an algorithm $\mathcal{A}$ outputting y on input x. When $\mathcal{A}$ is probabilistic, $[\mathcal{A}]$ denotes the set of all its possible outputs. Considering a second algorithm $\mathcal{O}$, $\mathcal{A}^{\mathcal{O}}$ means that the algorithm $\mathcal{A}$ has access to $\mathcal{O}$ as a black-box oracle. PPT means *Probabilistic Polynomial Time*. $[\![n]\!]$ denotes the set $[\![n[\![$. For a vector $m = (m_1, \cdots, m_n)$ and an integer μ, m^μ denotes the vector $(m_1^\mu, \cdots, m_n^\mu)$. $\sqcup$ operates as the union $\cup$ while preserving the repetition of elements, hence producing a multi-set. Finally, $\eta[i]$ refers to the i^{th} bit of some integer η.

Mathematical background. Throughout this paper, we consider a bilinear group setting $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e)$ where $\mathbb{G}_1$ $\mathbb{G}_2$, and $\mathbb{G}_t$ are multiplicative groups of prime order p, $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, and e is a type-3 bilinear pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$. We assume the *Decision Diffie-Hellman* (DDH) assumptions over these three groups: given $(g, g^a, g^b, g^z) \in \mathbb{G}^4$, there exists no PPT algorithm in $|p|$ able to decide whether $z = a \cdot b$ or not with non negligible probability[4]. This assumption implies hardness of the *Discrete Logarithm* (DL) problem: given $(g, g^x) \in \mathbb{G}^2$, there exists no PPT algorithm in $|p|$ able to return x with non-negligible probability. We also consider the relation $\mathcal{R}$ over $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e)$ defined by $\mathcal{R} = \{(m, m') \in \mathbb{G}^l \times \mathbb{G}^l \mid \exists \mu \in \mathbb{Z}_p, m' = m^\mu\}$ defining equivalence classes $[M]_{\mathcal{R}} \subset \mathbb{G}$ for an element $M \in \mathbb{G}^l$.

In what follows, we recall the definitions of structure-preserving signatures for equivalent class, zero-knowledge proofs, and encryption scheme.

Definition 1 (Class-hiding). *Let $l > 1$ be an integer, and $\mathbb{G}$ be group. $(\mathbb{G}^*)^l$ is class-hiding if for all PPT adversaries $\mathcal{A}$, the following probability is negligible:*

$$\Pr\left[\begin{matrix} b \leftarrow\!\!\$\, \{0,1\},\ M \leftarrow\!\!\$\, (\mathbb{G}^*)^l,\ M^{(0)} \leftarrow\!\!\$\, (\mathbb{G}^*)^l, \\ M^{(1)} \leftarrow\!\!\$\, [M]_{\mathcal{R}},\ b^* \leftarrow \mathcal{A}(M, M^{(b)}) \end{matrix} : b = b^* \right].$$

Lemma 1 (Fuchsbauer *et al.* [17]). *Let $l > 1$ be an integer, and $\mathbb{G}$ be a group of prime order p. Then $(\mathbb{G}^*)^l$ is a class hiding message space if and only if the DDH assumption holds in $\mathbb{G}$.*

Definition 2 (SPS-EQ [17]). *A* Structure-Preserving Signatures for Equivalence Classes $\mathcal{R}$ SPS-EQ *over a group $\mathbb{G}$ is a tuple of algorithms:*

$\mathsf{KeyGen}_{\mathsf{SPS\text{-}EQ}}(1^\lambda, l; \mathcal{R})$: *given an integer $l > 1$, return a key pair* $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}(\mathsf{sk}, m; \mathcal{R})$: *given a secret* sk *and a message m, return a signature σ.*

$\mathsf{ChgRep}_{\mathsf{SPS\text{-}EQ}}(m, \sigma, \mu, \mathsf{pk}; \mathcal{R})$: *given a representative m of an equivalent class, a signature σ, a scalar μ, and a public key* pk*, return an updated signature σ' for the message m^μ.*

[4] A function $\epsilon\colon \mathbb{N} \to \mathbb{R}^+$ is called *negligible*, if $\forall c > 0, \exists k_0 \in \mathbb{N}, \forall k > k_0, |\epsilon(k)| < \frac{1}{|k^c|}$.

$\mathsf{Verif}_{\mathsf{SPS\text{-}EQ}}(m, \sigma, \mathsf{pk}; \mathcal{R})$: *given a public key* pk*, a message* m*, a signature* σ*, return* 0 *or* 1 *(meaning* reject *or* accept*).*

We require that SPS-EQ *meets* Correctness*,* *EUF-CMA*, *and* Signature Adaptation *(stating that* $\mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}$ *and* $\mathsf{ChgRep}_{\mathsf{SPS\text{-}EQ}}$ *outputs are identically distributed). We further describe these properties in Appendix A. Since we always use the relation* $\mathcal{R}$ *defined above, we will no longer specify it in the input to the* SPS-EQ *algorithms.*

Definition 3 (NIZK [12] and SoK [21]). *A* Non-Interactive Zero-Knowledge proof (NIZK) *for a relation* $\mathcal{R}$ *is a pair of* PPT *algorithms:*
$\mathsf{ZK}\{w : (w, \phi) \in \mathcal{R}\}$: *given a witness* w *and a statement* ϕ*, return a proof* π*,*
$\mathsf{ZK.Verif}(\phi, \pi)$: *given a statement and a proof, return a bit 0 or 1.*
A NIZK *requires* Completeness, Simulation-Extractability *and* Zero-Knowledge *properties to be secure. We recall the definition of these properties in Appendix A.*

A Signature of Knowledge (SoK) *is similar to a* NIZK *except that the proof algorithm* $\mathsf{SoK}_m\{w : (w, \phi) \in \mathcal{R}\}$ *takes* m *as an additional parameter. As a consequence, the* Simulation-Extractability *of a* SoK*, similar to soudness of* NIZK *proofs, implies that it can be used as an* *EUF-CMA* *signature scheme, where* ϕ *is the public key,* w *is the secret key,* m *is the signed message, and* π *is the signature.* SoK *also achieve* Perfect Simulability *which is defined similarly to zero-knowledge for* NIZK *proofs. A signature of knowledge can be constructed from a non-interactive zero knowledge proof based on the Fiat–Shamir heuristic [10].*

Definition 4 (Asymmetric Encryption [20]). *An* asymmetric encryption scheme $\mathcal{E}$ *is a triple of* PPT *algorithms:*
$\mathsf{KeyGen}(1^\lambda)$: *return a key pair* $(\mathsf{pk}, \mathsf{sk})$.
$\mathsf{Enc}(\mathsf{pk}, p)$: *given a public key* pk *and a message* p*, return a ciphertext* c.
$\mathsf{Dec}(\mathsf{sk}, c)$: *given a secret key* sk *and a ciphertext* c*, return a message* p.
An encryption scheme $\mathcal{E}$ *has to achieve* Correctness *and* Indistinguishability under Chosen Ciphertext Attack *(IND-CCA). We recall this property in Appendix A.*

3 k-Times Anonymous Proxy Signature

In this section we give a formal definition and security model for *(fully traceable)* k*-times anonymous proxy signature*. In this primitive, a signer can delegate to a proxy the authority to anonymously produce at most k signatures. To do this, the signer generates a delegation certificate (denoted del) via the algorithm Delegate, using the proxy's public key and the limit k as input. To produce a proxy signature, the proxy uses this delegation with an integer $\eta \in \{0, \cdots, k-1\}$ that must be different for each k signature. Note that η must not appear in the signature to preserve anonymity (we will describe the corresponding security model in more detail later in this section), so it is not given as input to the verification algorithm. If the proxy decides to produce more than k signatures, it will be forced to use the same index η twice, triggering a mechanism that allows any user to link these two signatures using an algorithm Link, and to extract the identity of the proxy. The algorithm Link also returns a token w which, when used with a signature as input to the Trace algorithm, indicates whether or not the signature was generated by the same proxy, making it possible to find all signatures generated by the proxy in the past. Note that the signer can extend the limit by generating new delegations for the same proxy.

Definition 5 (k-APS). *A* k-times Anonymous Proxy Signature scheme *(*k-APS*) is a tuple of algorithms:*
$\mathsf{Setup}(1^\lambda)$: *given a security parameter, return a public parameter* params*. Note that* params *is considered as implicit input of all the following algorithms.*
$\mathsf{KeyGen}(1^\lambda, k)$: *given a limit* $k \in \mathbb{N}$*, return the signer secret/public keys* $(\mathsf{sk}, \mathsf{pk})$.
$\mathsf{PKeyGen}(1^\lambda)$: *return the proxy secret/public keys* $(\mathsf{psk}, \mathsf{ppk})$.
$\mathsf{Delegate}(\mathsf{sk}, \mathsf{ppk}, l)$: *given the keys* sk*,* ppk *and* $l \leq k$*, return a delegation certificate* del.
$\mathsf{Sign}(\mathsf{pk}, \mathsf{psk}, m, \mathsf{del}, \eta)$: *given the keys* pk*,* psk*, a message* m*, a certificate* del*, and an index* η*, return a signature* σ.

$\mathsf{Verify}(\mathsf{pk}, m, \sigma)$: *given the key* pk*, a message* m*, and a signature* σ*, return 0 or 1 (for* reject *or* accept*).*
$\mathsf{Link}(\mathsf{pk}, m, \sigma, m', \sigma')$: *given a public key* pk *and two message-signature pairs* (m, σ)*,* (m', σ')*, return identity denoted by the public key* ppk *of ther signer and a witness* w *or* $\perp$ *in case of failure.*
$\mathsf{Trace}(w, \sigma)$: *given a witness* w *and a signature* σ*, return 0 or 1.*

A k-APS is said to be *correct* if, using keys/certificateshonestly generated by the algorithms KeyGen, PKeyGen, and Delegate, (i) any signature produced by the algorithm Sign is verified by the algorithm Verify using the signer public key, (ii) 2 signatures are linked by the algorithm Link which outputs the corresponding public key if and only if they were produced with the same delegation certificate and the same η, and (iii) the algorithm Trace returns 1 on the token outputted by Link and any of the signatures produced from this delegation certificate.

Our security model is inspired both by that of anonymous proxy signatures [18] and that of k-times full traceability [7]. The security experiments and associated oracles are given in Figure 1, with Figure 2 providing a subroutine for the experiment associated to the traceability. For each oracle, the underlined inputs correspond to those chosen by the opponent. Experiments use multisets (sets that may contain the same element multiple times) that are considered to be global variables (and can therefore be accessed and modified in oracles): $\mathcal{U}$ stores the registered users, $\mathcal{D}$ stores the delegations, $\mathcal{S}$ stores the produced signatures, and $\mathcal{H}$ stores the signature indexes. The security properties required for k-APS are defined as follows:

Unforgeability. This property ensures that an adversary playing the role of proxies will not be able to produce a signature unless they have received a delegation certificate. For this property to hold, a PPT adversary $\mathcal{A}$ must forge a valid fresh message/signature pair (m^*, σ^*) for a message that has never been queried to the signature oracle. The adversary can request delegation certificates generated for non-corrupted proxies whose secret keys it does not know. A k-APS is *Unforgeable* if for any probabilistic polynomial time algorithm $\mathcal{A}$, the probability $\mathsf{Adv}^{\mathsf{unf}}_{\text{k-APS},\mathcal{A}}(1^\lambda) = \Pr[\mathsf{Exp}^{\mathsf{unf}}_{\text{k-APS},\mathcal{A}}(1^\lambda) = 1]$ is negligible.

Anonymity. The anonymity ensures that the signatures do not disclose the identity of the proxy signer (given by its public key) and that signatures generated by the same proxy signer remain unlinkable. Note that in our model, anonymity only concerns the signature verifiers, and not the delegator; indeed, in our application, there is no reason why the delegator should not know the identity of the proxy signing on its behalf, and it is even rather preferable that it should for accountability reasons. In the corresponding experiment, the adversary chooses a limit $t \leq k$, then it tries to distinguish the origin of a challenge signature produced by one of two honest proxies. The adversary can request to the oracles a maximum of $t-1$ signatures for each of the proxies, and a single signature for one of the two proxies (the one chosen by the challenger), which guarantees that the adversary cannot obtain more than t signatures for one of the two proxies (it would trivially link these signatures to the challenge, which is an inherent property of our primitive). For each of the two proxies, the signature oracle increments the index η at each signature, which ensures that a η is not used twice for the same proxy. Our model therefore considers adversaries trying to link two signatures from two different proxies with the same η, which would allow the adversary to infer that two signatures are not from the same proxy, and thus generally ensures that η is not leaked from the signature.

A k-APS is *anonymous* if for any PPT $\mathcal{A}$, the probability $\mathsf{Adv}^{\mathsf{Ano}}_{\text{k-APS},\mathcal{A}}(1^\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{Ano}}_{\text{k-APS},\mathcal{A}}(1^\lambda) = 1] - 1/2|$ is negligible.

Traceability. This property guarantees that the Trace algorithm leaks the identity of any adversary overpassing the delegation limit. In the corresponding experiment, the adversary's target is to produce more signatures than allowed by the delegator, without the Link and Trace algorithms being able to correctly link or trace the signatures. For that the adversary can obtain multiple delegation certificates for different limits and different public keys ppk. Since each delegation certificate allows it to produce k_i signatures, it is required to produce strictly more than $\sum_{i=1}^{n} k_i$ valid signatures. The adversary wins the experiment if the number of traced signatures is less than the number of signatures that would have been traced if they had been generated honestly. This test, which is described in Figure 2, has to take account of all the delegations that have been exceeded in any execution scenario. First, note that if the limit of a delegation certificate for a public key is exceeded, then it must be possible to trace all signatures generated by the owner of that public key, even if they were generated using a different delegation certificate. Thus, the number of

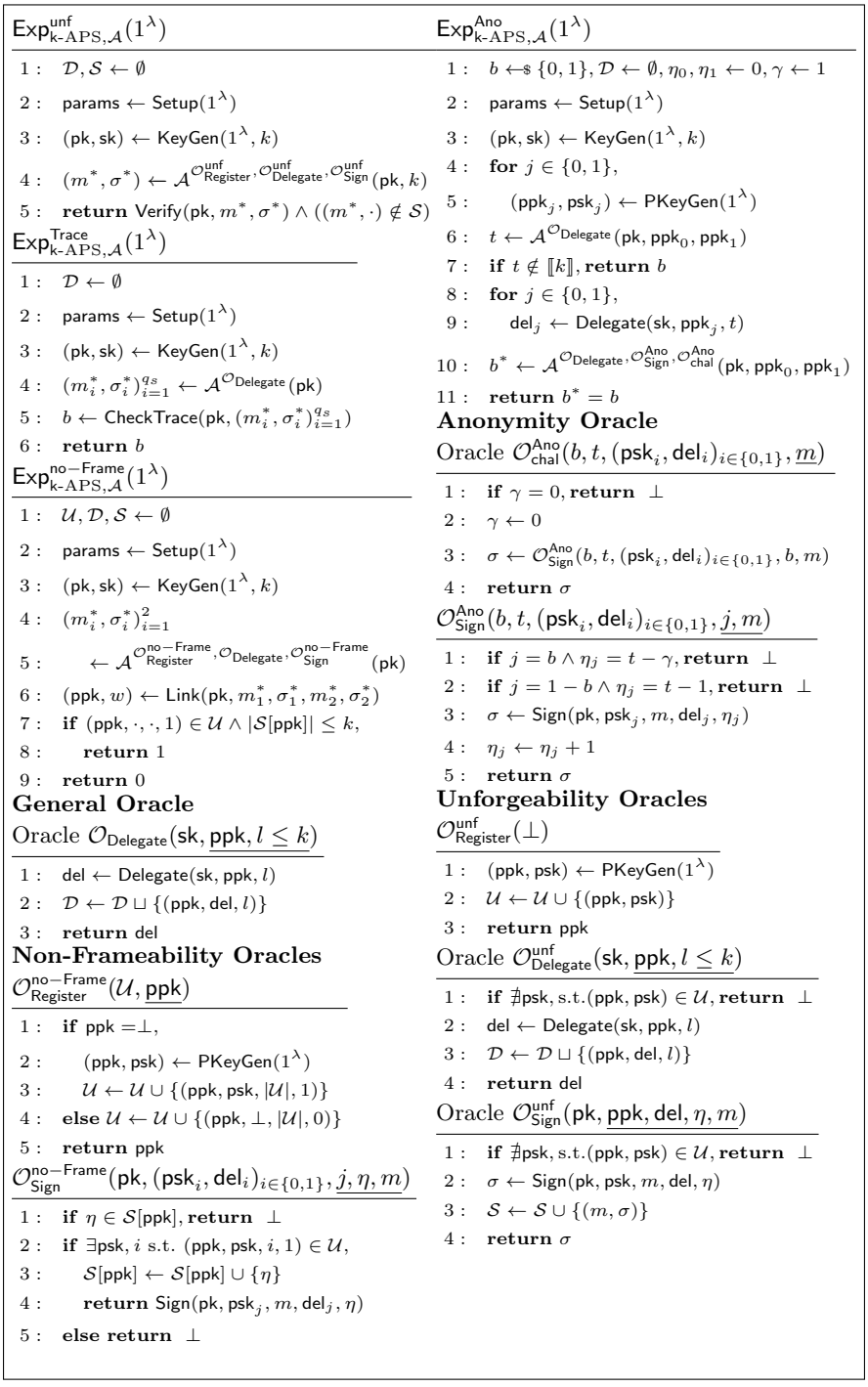

Figure 1: Experiments and Oracles modeling the Security of k-Times Anonymous Proxy Signatures. (Oracles inputs provided by the adversary are underlined, the other are provided by the challenger. The sets $\mathcal{U}, \mathcal{D}, \mathcal{S}, \mathcal{H}$ are global parameters. Subroutine $\mathsf{CheckTrace}$ is defined in Figure 2.)

$\mathsf{CheckTrace}(\mathsf{pk}, (m_i^*, \sigma_i^*)_{i=1}^{q_s})$

1 : **if** $\mathcal{D}$ is defined, $\mathcal{D} \xrightarrow{p} (\mathsf{pk}_i, \mathsf{del}_i, k_i)_{i=1}^{n}$ // For proxy signatures only.
2 : **if** $\mathcal{S}$ is defined $\wedge\, \exists i \in [\![q_s]\!], (m_i^*, \sigma_i^*, *, *) \in \mathcal{S}$, **return** 0 // For sanitizable signatures only.
3 : $T \leftarrow 0, \mathsf{W}, \mathsf{ID} \leftarrow \emptyset, \mathsf{diff} \leftarrow q_s - \sum_{1 \le i \le n} k_i$ // diff is required to be strictly positive.
4 : **if** $(\exists i \in [\![q_s]\!], \mathsf{Verify}(\mathsf{pk}, m_i^*, \sigma_i^*) = 0) \vee (\exists i, j \in [\![q_s]\!], j \neq i, (m_i^*, \sigma_i^*) = (m_j^*, \sigma_j^*)) \vee (\mathsf{diff} \le 0)$,
5 : **return** 0
6 : **for** $1 \le i < j \le q_s$,
7 : $(\mathsf{ppk}_{i,j}, w_{i,j}) \leftarrow \mathsf{Link}(\mathsf{pk}, m_i^*, \sigma_i^*, m_j^*, \sigma_j^*)$ // Try linking any two signatures.
8 : **if** $(\mathsf{ppk}_{i,j}, w_{i,j}) \neq \perp \wedge \mathsf{ppk}_{i,j} \notin \mathsf{ID}$, // Identities for which signatures have been linked.
9 : $\mathsf{ID} \leftarrow \mathsf{ID} \cup \{\mathsf{ppk}_{i,j}\}, W[\mathsf{ppk}_{i,j}] \leftarrow W[\mathsf{ppk}_{i,j}] \cup \{w_{i,j}\}$
10 : $T \leftarrow \sum_{\mathsf{ppk} \in \mathsf{ID}} \left(\sum_{w \in \mathsf{W}[\mathsf{ppk}]} \left(\sum_{i=1}^{q_s} \mathsf{Trace}(w, \sigma_i) \right) \right)$ // Sum up traced signatures and compare it to the number of allowed signatures for these entities.
11 : **if** $T < \sum_{\mathsf{pk}_i \in \mathsf{ID}} k_i + \mathsf{diff}$, **return** 1, **else** , **return** 0

Figure 2: $\mathsf{CheckTrace}$ Subroutine for the Traceability Experiment.

signatures traced T should be at least the sum of the limits k_i of each delegation produced for each public key traced (expressed as $\sum_{\mathsf{pk}_i \in \mathsf{ID}} k_i$ in Figure 2), to which we add the number of signatures that exceed the global sum of the limits for all delegation certificate used by the adversary (expressed as $\mathsf{diff} = q_s - \sum_{i=1}^{n} k_i$ in Figure 2, where q_s is the number of signatures outputted by the adversary). A k-APS is *traceable* if for any PPT algorithm $\mathcal{A}$, the probability $\mathsf{Adv}^{\mathsf{Trace}}_{kAPS,\mathcal{A}}(1^\lambda) = \Pr[\mathsf{Exp}^{\mathsf{Trace}}_{\text{k-APS},\mathcal{A}}(1^\lambda) = 1]$ is negligible.

Non-Frameability. This property prevents a PPT adversary from framing someone else by generating malformed yet valid signatures. More precisely, the goal of the adversary is to output two signatures traceable to a registered proxy who remains honest. To help it, the adversary has access to oracles that can be used to register users, obtain delegation certificates, and obtain signatures for honest users. The adversary must of course not have abused the signature oracle by producing more than k signatures for the proxy it wishes to trace. Note that in our model we implicitly assume that the linking of two signatures and the tracing are performed by the same user (we do not consider the case where an adversary only generates a tracing token w that traces an honest user without the linked signatures). In practice, this means that to delegate tracing, the delegate must be provided with the two linked signatures so that it can link them and produce its own token w. A k-APS is *Non-Frameable* if for any PPT algorithm $\mathcal{A}$, the probability $\mathsf{Adv}^{\mathsf{no-Frame}}_{\text{k-APS},\mathcal{A}}(1^\lambda) = \Pr[\mathsf{Exp}^{\mathsf{no-Frame}}_{\text{k-APS},\mathcal{A}}(1^\lambda) = 1]$ is negligible.

4 Our k-Times Anonymous Proxy Signature Scheme

In this section, we present our k-times anonymous proxy signature ($\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PKeyGen}, \mathsf{Delegate}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Link}, \mathsf{Trace}$), which uses a bilinear group setting $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e)$ and a SPS-EQ scheme.

Construction Intuition. The setup (algorithm Setup) of our construction returns several group elements and the description of a hash function. In particular, the group element g_1 will be used as a basis for the proxy public key $\mathsf{ppk} = g_1^{\mathsf{psk}}$ (where psk is the proxy secret key). The signer key pair (generated from based on KeyGen) is a SPS-EQ key pair supporting vectors of $4l + 1$ group elements in $\mathbb{G}_1$, where $l = \lceil \log_2(k) \rceil$.

To delegate (algorithm $\mathsf{Delegate}$) the power to create k anonymous signatures, the signer will create two sets of l public/secret keys $(y_{i,0}, x_{i,0})_{i \in [\![l]\!]}$ and $(y_{i,1}, x_{i,1})_{i \in [\![l]\!]}$. The idea behind this technique is to be able to create k public/secret keys by composing the previous $2\log_2(k)$ keys: given an integer $\eta < k$, the key corresponding to η will be composed of the keys corresponding to each of the bits in η. For each i, the signer also produce a Diffie-Hellman key $\mathsf{ppk}_{i,j} = g_1^{\mathsf{psk} \cdot x_{i,j}}$ between $y_{i,j} = g_1^{x_{i,j}}$ and $\mathsf{ppk} = g_1^{\mathsf{psk}}$. This will enable

us to link the keys produced by these elements to the ppk owner later on. Finally, all these public keys are signed with an SPS-EQ, acting as a certificate, so that they can be randomized. All these elements are stored in the delegation del. Thanks to del, we have already shown that the proxy can produce k distinct pairs of certified ElGamal public/secret keys. The idea of our signature algorithm (Sign) is to use one of its keys for each signature. If the same key is used several times, however, it will be possible to find its owner thanks to the mechanism introduced in [7] (this point will be discussed further in this section). However, to preserve anonymity, these keys must not be linkable, so they must be randomized (note that the SPS-EQ properties preserve their certification by the signer).

Assume that the delegate is using the algorithm for the η^{th} time. First, the delegate randomizes g_1 and all the elements $y_{i,j}$ and $\mathsf{ppk}_{i,j}$ using the same random r as an exponent, and adapts the SPS-EQ accordingly. The randomized version of g_1 is denoted $\widehat{g}_1$, and the keys are respectively denoted $\widehat{y}_{i,j}$ and $\widehat{\mathsf{ppk}}_{i,j}$. This first step randomizes all the elements in the delegation $y_{i,j}$ and $\mathsf{ppk}_{i,j}$, so that it is not possible to make the link between the randomized delegation $\widehat{y}_{i,j}$ and $\widehat{\mathsf{ppk}}_{i,j}$ and the original one. Then, the delegate chooses a new random s, randomize the basis $\widehat{g}_1$ in $\widetilde{g}_1$, and randomizes only the $\widehat{y}_{i,\eta[i]}$ and $\widehat{\mathsf{ppk}}_{i,\eta[i]}$ corresponding to the bits of η to obtain the keys $\widetilde{y}_i$ and $\widetilde{\mathsf{ppk}}_i$. The delegate uses a zero-knowledge proof to ensure that the randomization has been done correctly and with an integer η actually lower than k (the instantiation of this proof is rather technical and described in more details in the next section). In this way, it is possible to multiply the public keys $\widetilde{y} = \prod_{i=1}^{l} \widetilde{y}_i$ and add the corresponding secret keys $x = \sum_{i=1}^{l} x_{i,\eta[i]}$ to obtain a new public/secret key pair $(\widetilde{y}, x)$ that verifies $\widetilde{y} = \widetilde{g}_1^x$. This second step allows the proxy to hide its chosen η in the elements $\widetilde{y}_i$ and $\widetilde{\mathsf{ppk}}_i$ (by randomizing the $\widehat{y}_{i,\eta[i]}$ and $\widehat{\mathsf{ppk}}_{i,\eta[i]}$ again) while preserving the link between the randomized delegation $\widehat{y}_{i,j}$ and $\widehat{\mathsf{ppk}}_{i,j}$ and the generated key pair $(\widetilde{y}, x)$. Note that $\widetilde{\mathsf{ppk}} = \prod_{i=1}^{l} \widetilde{\mathsf{ppk}}_i$ is the Diffie-Hellman of $\widetilde{y}$ and ppk, which links $\widetilde{y}$ to the owner of ppk in a hidden way. It allows the delegate to prove in zero-knowledge that the identity revealed by the mechanism of [7] is indeed the identity of the delegate. This proof, denoted π_σ, also proves that the mechanism of [7] triggers correctly if the delegate signs more than k messages. The technical description of this proof is given in Appendix B.

The signature verification (algorithm Verify) consists of re-computing $\widetilde{y}$ and $\widetilde{\mathsf{ppk}}$ and checking that the proof π_σ is valid. Finally, the Link and Trace algorithms work in the same way as in [7]: each signature contains $\alpha_1 = h_1^x$, $\alpha_2 = g_2^t$, $\alpha_3 = h_2^x \cdot g_1^{u \cdot \mathsf{psk}}$, and $\alpha_4 = h_3^x \cdot h_4^{v \cdot \mathsf{psk}}$. Thus, if the same key x is used twice in signatures, they can be linked since they share the same $\alpha_1 = h_1^x$. Let's note $\alpha_3' = h_2^x \cdot g_1^{u' \cdot \mathsf{psk}}$ the element α_3 of the second signature. It is possible to find the identity of the delegate by computing $(\alpha_3/\alpha_3')^{1/(u-u')} = \mathsf{ppk}$. By a similar way, the token $\omega = h_4^{\mathsf{psk}}$ leaks from α_4 when two signatures are linked. Each signature has also the elements $\tau = e(\omega, \alpha_2)$. Without knowledge of ω, τ is indistinguishable from a random element under the DDH assumption, but a user which knows the delegate's token ω can retrieve its signatures by recomputing τ, thus achieving full traceability.

Formal Description. Below is the formal description of our scheme.

$\underline{\mathsf{Setup}(1^\lambda):}$ sample $g_1, h_1, h_2, h_3, h_4 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$, and return them as the common parameters.

$\underline{\mathsf{KeyGen}(1^\lambda, k):}$ set $l = \lceil \log_2(k) \rceil$, and return $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}_{\mathsf{SPS\text{-}EQ}}(1^\lambda, 4l+1)$.

$\underline{\mathsf{PKeyGen}(1^\lambda):}$ sample $\mathsf{psk} \leftarrow\!\!\$\ \mathbb{Z}_q$ and set $\mathsf{ppk} = g_1^{\mathsf{psk}}$. Return the pair $(\mathsf{psk}, \mathsf{ppk})$.

$\underline{\mathsf{Delegate}(\mathsf{sk}, \mathsf{ppk}, k):}$ set $l = \lceil \log_2(k) \rceil$, abort if the SPS-EQ key sk does not support message of $4l+1$ elements. For all $(i,j) \in [\![l]\!] \times \{0,1\}$, sample $x_{i,j} \leftarrow\!\!\$\ \mathbb{Z}_p^*$, set $y_{i,j} = g_1^{x_{i,j}}$, $\mathsf{ppk}_{i,j} = \mathsf{ppk}^{x_{i,j}}$ and produce $\widehat{\sigma} \leftarrow \mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}(\mathsf{sk}, g_1, y_{1,0}, \cdots, y_{l,1}, \mathsf{ppk}_{1,0}, \cdots, \mathsf{ppk}_{l,1})$. Return $\mathsf{del} = ((x_{i,j}, y_{i,j}, \mathsf{ppk}_{i,j})_{i \in [\![l]\!]; j \in \{0,1\}}, \widehat{\sigma})$.

$\underline{\mathsf{Sign}(\mathsf{pk}, \mathsf{psk}, m, \mathsf{del}, \eta):}$ set $l = \lceil \log_2(k) \rceil$. Sample $r, s \leftarrow\!\!\$\ \mathbb{Z}_p^*$, set $\widehat{g}_1 = g_1^r$ and $\widetilde{g}_1 = \widehat{g}_1^{\,s}$. For all $i \in [\![l]\!]$ and $j \in \{0,1\}$, compute $\widehat{y}_{i,j} = y_{i,j}^r$ and $\widehat{\mathsf{ppk}}_{i,j} = \mathsf{ppk}_{i,j}^r$, adapt the SPS-EQ signature $\widehat{\sigma} \leftarrow \mathsf{ChgRep}_{\mathsf{SPS\text{-}EQ}}((g_1, y_{1,0}, \cdots, y_{l,1}, \mathsf{ppk}_{1,0}, \cdots, \mathsf{ppk}_{l,1}), \widehat{\sigma}, r, \mathsf{pk})$, compute $\widetilde{y}_i = \widehat{y}_{i,\eta[i]}^s$, and $\widetilde{\mathsf{ppk}}_i = \widehat{\mathsf{ppk}}_{i,\eta[i]}^{\,s}$. Generate a zero-knowledge proof $\mathit{\Pi}_{<k}$ of knowledge of s and η which proves that (i) $\widetilde{y}_i$ and $\widetilde{\mathsf{ppk}}_i$ are well formed according to s and some integer η of l bits and (ii) $\eta < k$. We defer the formalisation of this zero-knowledge proof to

Section 5. Set $x = \sum_{i=1}^{l} x_{i,\eta[i]}$, $\widetilde{y} = \prod_{i=1}^{l} \widetilde{y}_i$, $\widetilde{\mathsf{ppk}} = \prod_{i=1}^{l} \widetilde{\mathsf{ppk}}_i$, sample $t \leftarrow\$ \mathbb{Z}_p^*$ and compute $\alpha_1 = h_1^x$, $\alpha_2 = g_2^t$, $u = H(m, 0, \alpha_2)$ and $v = H(m, 1, \alpha_2)$. Generate the matching elements $\alpha_3 = h_2^x \cdot g_1^{u \cdot \mathsf{psk}}$ and $\alpha_4 = h_3^x \cdot h_4^{v \cdot \mathsf{psk}}$, and the tracing element $\tau = e(h_4, \alpha_2)^{\mathsf{psk}}$. Also generate:

$$\pi_\sigma \leftarrow \mathsf{ZK}\left\{ \mathsf{psk}, x, t \colon \begin{array}{l} \widetilde{y} = \widetilde{g_1}^{\,x} \wedge \widetilde{\mathsf{ppk}} = \widetilde{y}^{\mathsf{psk}} \wedge \alpha_1 = h_1^x \wedge \alpha_2 = g_2^t \\ \wedge \alpha_3 = h_2^x \cdot g_1^{u \cdot \mathsf{psk}} \wedge \alpha_4 = h_3^x \cdot h_4^{v \cdot \mathsf{psk}} \wedge \tau = e(h_4, \alpha_2)^{\mathsf{psk}} \end{array} \right\}.$$

Set $\sigma_{\mathsf{del}} = (\widehat{g_1}, ((\widehat{y}_{i,b}, \widehat{\mathsf{ppk}}_{i,b})_{b\in\{0,1\}}, \widetilde{y}_i, \widetilde{\mathsf{ppk}}_i)_{i\in[\![l]\!]}, \widehat{\sigma}, \Pi_{<k})$ and return the signature $\sigma = (\widetilde{g_1}, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \tau, \pi_\sigma, \sigma_{\mathsf{del}})$.

$\underline{\mathsf{Verify}(m, \sigma, \mathsf{pk})\colon}$ parse $\sigma \xrightarrow{p} (\widetilde{g_1}, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \tau, \pi_\sigma, \sigma_{\mathsf{del}})$ and $\sigma_{\mathsf{del}} \xrightarrow{p} (\widehat{g_1}, ((\widehat{y}_{i,b}, \widehat{\mathsf{ppk}}_{i,b})_{b\in\{0,1\}}, \widetilde{y}_i, \widetilde{\mathsf{ppk}}_i)_{i\in[\![l]\!]}, \widehat{\sigma}, \Pi_{<k})$. Compute $u = H(m, 0, \alpha_2)$, $v = H(m, 1, \alpha_2)$, $\widetilde{y}_n = \prod_{i=1}^{l} \widetilde{y}_i$, and $\widetilde{\mathsf{ppk}} = \prod_{i=1}^{l} \widetilde{\mathsf{ppk}}_i$. Verify the signature $\widehat{\sigma}$ and the proofs $\Pi_{<k}$ and π_σ. If all checks are correct, returns 1, otherwise 0

$\underline{\mathsf{Link}(\mathsf{pk}, m, \sigma, m', \sigma')\colon}$ verify that $\mathsf{Verify}(\mathsf{pk}, m, \sigma) = \mathsf{Verify}(\mathsf{pk}, m', \sigma') = 1$ and return 0 if $\alpha_1 \neq \alpha_1'$ or if one of the verification failed. Compute $u = H(m, 0, \alpha_2)$, $v = H(m, 1, \alpha_2)$, $u' = H(m', 0, \alpha_2')$, $v' = H(m', 1, \alpha_2')$, $\mathsf{ppk} = (\alpha_3/\alpha_3')^{1/(u-u')}$ and $w = (\alpha_4/\alpha_4')^{1/(v-v')}$. Return (ppk, w).

$\underline{\mathsf{Trace}(w, \sigma)\colon}$ return 1 if and only if $\tau = e(w, \alpha_2)$.

In the following, we informally explain why each of the security properties presented in Section 3 holds in our scheme.

Unforgeability. Zero-knowledge proofs produced during the signing process ensure that the delegate has used its certificate correctly. This means that a user who has not been delegated cannot produce a valid fresh signature under the assumption that the proof is sound.

Anonymity. Since the elements of the certificate are randomized for each new signature, and since the delegate is able to create public keys $\widetilde{y}$ for k different secret keys x, it is not possible to link two signatures from the elements $\widehat{y}_{i,j}$ and $\widetilde{y}_i$ under the DDH assumption. Recall also that the $\widehat{\mathsf{ppk}}_{i,j}$ and $\widetilde{\mathsf{ppk}}_i$ do not allow the signature to be linked to ppk under the DDH assumption either. On the other hand, the element h_2^x (resp. h_3^x) is the Diffie-Hellman of h_2 (resp. h_3) and $\widetilde{y}$ (where $\widetilde{y}$ varies with each of the k signatures), and therefore hides the elements linked to the identity of the delegate composing α_3 (resp. α_4). Finally, τ hides the value of ppk under the DDH assumtion on the elements h_4 and ppk. Assuming that the proofs are indeed zero-knowledge, it is not possible to link two signatures from the same honest user.

Traceability and Non-Frameability. Zero-knowledge proofs ensure that the signature is correct and that the delegate knows the secret key corresponding to the public key used in the certificate. Thus, the delegate cannot bypass the mechanism for linking/tracing its signatures if it exceeds the limit, which ensures traceability, and the delegate can only use elements of its own delegation, which ensures non-frameability.

We therefore have the following theorem, the proofs are available in Appendix C.

Theorem 1. *Instanciated by a signature on equivalent classes that is unforgeable, class-hiding, and signature adaptatable, by* NIZK *proofs which are zero-knowledge and sound, and by a collision-resistante hash function, our* k-APS *scheme is unforgeable, anonymous, traceable and non-frameable under the* DDH *assumption in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$.

5 Zero-knowledge Proof Instantiation

The proof $\Pi_{<k}$ requires several building blocks. In [13], Chaum and Pedersen introduce a zero-konwledge proof of knowledge for discrete logarithm equality $\mathsf{ZK}\{x : y_1 = g_1^x \wedge y_2 = g_2^x\}$ in a group of prime order p. This proof is a sigma protocol: the prover sends a commitment, the verifier sends a challenge (chosen in $\mathbb{Z}_p^*$), and the prover sends a response. This proof can be extended to prove the equality of more than two discret logarithms equalities, in this case the size of the resulted transcript is linear in the number of statements. In general, if two sigma protocols for two instances ϕ_1 and ϕ_2 and two relations $\mathcal{R}_1$ and $\mathcal{R}_2$ use the same

challenge space, it is possible to merge the proofs by using the same challenge in order to obtain an `and`-proof $\mathsf{ZK}\{w_1, w_2 : (w_1, \phi_1) \in \mathcal{R}_1 \wedge (w_2, \phi_2) \in \mathcal{R}_2\}$. This method can also be extended to any number of instances. In [14], Cramer *et al.* propose a zero-knowledge proof to prove the knowledge of the witness corresponding to one of two instances $\mathsf{ZK}\{w : (w, \phi_1) \in \mathcal{R}_1 \vee (w, \phi_2) \in \mathcal{R}_2\}$, under the hypothesis that $\mathsf{ZK}\{w : (w, \phi_1) \in \mathcal{R}_1\}$ and $\mathsf{ZK}\{w : (w, \phi_2) \in \mathcal{R}_2\}$ are sigma protocols that use the same challenge space. The challenge space of the resulting proof remains the same as that of the two combined proofs. The method can be extended to prove the knowledge of a witness in relation to one instance among n, in which case the transcript size is equal to the sum of the transcript sizes of the proofs of each instance.

The proof $\Pi_{<k}$ ensures that the prover knows s and η such that (i) $\widetilde{y}_i$ and $\widetilde{\mathsf{ppk}}_i$ are well formed according to s and some integer η of l bits, and (ii) $\eta < k$. Proving (i) is equivalent to prove $(\widetilde{g}_1 = \widehat{g}_1^s$ and $\widetilde{y}_i = \widehat{y}_{i,0}^s$ and $\widetilde{\mathsf{ppk}}_i = \widehat{\mathsf{ppk}}_{i,0}^s)$ or $(\widetilde{g}_1 = \widehat{g}_1^s$ and $\widetilde{y}_i = \widehat{y}_{i,1}^s$ and $\widetilde{\mathsf{ppk}}_i = \widehat{\mathsf{ppk}}_{i,1}^s)$ for all $i \in [\![0, l]\!]$. The tools introduced in the previous paragraph allow us to construct the following proof for such a statement: $\mathsf{ZK}\left\{s : \bigwedge_{i=0}^{l} \bigvee_{j=0}^{1} \left(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_i = \widehat{y}_{i,j}^s \wedge \widetilde{\mathsf{ppk}}_i = \widehat{\mathsf{ppk}}_{i,j}^s\right)\right\}$. The transcript of this proof is linear in l. On the other hand, proving (ii) consists in proving $\eta < k$, where each bit $\eta[i]$ of η is committed in $\widetilde{y}_i = \widehat{y}_{i,\eta[i]}^s$. So to prove that η is smaller than k, we need to compare k and η as binary words across commitments $\widetilde{y}_i$, by going through the bits from most to least significant. For instance, using $k = 1001101$, proving that $\eta < k$ consists in proving that $\eta[0] = 0$ or $(\eta[1] = 0$ and $\eta[2] = 0$ and $(\eta[3] = 0$ or $(\eta[4] = 0$ or $(\eta[5] = 0$ and $\eta[6] = 0)))$. In this case, the required proof is:

$$\mathsf{ZK}\left\{ s : \begin{array}{l} (\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_0 = \widehat{y}_{0,0}^s) \vee ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_1 = \widehat{y}_{1,0}^s) \wedge (\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_2 = \widehat{y}_{2,0}^s) \\ \wedge((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_3 = \widehat{y}_{3,0}^s) \vee ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_4 = \widehat{y}_{4,0}^s) \\ \vee((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_5 = \widehat{y}_{5,0}^s) \wedge (\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_6 = \widehat{y}_{6,0}^s))))) \end{array} \right\}.$$

This technique can be generalized as follows. Let $(i_j)_{0 \leq j \leq n}$ be the indices of the 1's in the binary word k. Note that i_0 is always 1. Proving that $\eta < k$ consists in the following proof:

$$\mathsf{ZK}\left\{ s : \begin{array}{l} (\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_{i_0} = \widehat{y}_{i_0,0}^s) \vee (\bigwedge_{i=i_0+1}^{i_1-1}(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_i = \widehat{y}_{i,0}^s) \wedge \\ ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_{i_1} = \widehat{y}_{i_1,0}^s) \vee (\bigwedge_{i=i_1+1}^{i_2-1}(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_i = \widehat{y}_{i,0}^s) \wedge (\cdots \\ (\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_{i_n} = \widehat{y}_{i_n,0}^s) \vee (\bigwedge_{i=i_n+1}^{l}(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_i = \widehat{y}_{i,0}^s)) \cdots)))) \end{array} \right\}$$

The relation of this proof is a boolean combination of l proofs of equality of discrete logarithms. Using the techniques presented above, we thus obtain a proof whose transcript size is l times the transcript size of a proof of equality of discrete logarithms. This may seem surprising, since the development of the boolean formula gives on the order of l^2 terms, however the generic transformations we use for the `and` and `or` proofs depend on how the formula is expressed, and the size of the proofs is linear in the number of termes in the formula. In Appendix D, we detail an example to illustrate this point. Finally, we use the Fiat-Shamir [15] transform to change these proofs into non-interactive ones. The proof $\Pi_{<k}$ is the composition of the two proofs presented above.

6 Extension to Sanitizable Signatures

Sanitizable signature schemes [1] enable a delegate called the *sanitizer* to modify specific sections of a signed message $m = m_1 \| \cdots \| m_n$ and update the signature consistently with these modifications. They can also be seen as a more restrictive variant of proxy signatures, in which the sanitizer receives delegations prescribing portions of the messages it can sign: the (sanitizable) signature algorithm produces both a signature and data enabling the delegate to produce new signatures using the sanitization algorithm (corresponding to the delegation certificate in proxy signature) as long as the restrictions chosen by the signer are respected.

In this section we formalise the notion of a *(fully traceable) k-times anonymous sanitizable signature* (k-SAN). Due to space limitations, we only briefly describe the formal definition and the security model for k-SAN (the full definitions are given in the Appendix E). We then extend our proxy signature scheme to the case of sanitizable signatures. Our formal definition combines the features of the k-times anonymous

proxy signatures defined in Section 3 with the standard features of sanitizable signatures [4, 11, 23, 3]. In particular, each sanitization requires the use of a delegation that can only be used k times, even if it is used for different signatures.

A k-SAN is composed by the algorithms Setup, KeyGen, SaKeyGen, Delegate, Sign, Sanitize, Verify, Link and Trace. With the exception of Sign and Sanitize, all these algorithms are defined in a similar way to k-APS (SaKeyGen correspond to PKeyGen and generate the sanitizer key pair $(\mathsf{ssk}, \mathsf{spk})$). The algorithms Sign and Sanitize are defined as follows:

$\mathsf{Sign}(m, \mathsf{ADM}, \mathsf{sk}, \mathsf{spk})$: given a signer secret key sk, a sanitizer public key spk, a message m, a admissible set ADM (which describes the set of all modifications MOD that can be applied to the message), return a signature σ.

$\mathsf{Sanitize}(m, \sigma, \mathsf{MOD}, \mathsf{ssk}, \mathsf{pk}, \mathsf{del}, \eta)$: given the signer public key pk, the sanitizer secret key ssk, a message-signature pair (m, σ), a modification MOD, a delegation del, a signature index η, return a signature σ' (for the modified message $\mathsf{MOD}(m)$).

A k-times Anonymous Sanitizable Signature scheme is required to achieve *Unforgeability*, *Immutability*, *Transparency*, *Invisibility*, *Unlinkability*, *Anonymity*, *Traceability* and *Non-Frameability*. After describring our scheme, we give some intuition on these requirement and set out how they are achieved.

Note that sanitizable signatures usually have two additional algorithms, Prove and Judge, which allow the delegating signer to reveal a posteriori that a given signature was produced by the sanitizer. In this case, an additional security property, accountability, is required to ensure that the signer cannot blame the sanitizer for a signature it did not produce, and that the sanitizer will not be able to produce a signature that cannot be traced by the signer. Since in this article we are considering a scenario where the tracing of dishonest users is not done by the signer, but by the verifier using the mechanism triggered when the sanitizer produces too many signatures, we have not provided our construction with these algorithms and have not adapted the accountability model.

Our k-times anonymous sanitizable signature combines the design of the sanitizable signatures in [8, 3] with the mecanism we introduced in our k-times anonymous proxy signature. The signature contains commitements that allows the sanitizer to show that only admissible blocks are modified. More precisely, the sanitizer gives a proof that for every block within the altered message, the commitment corresponds to the hash of the index or the hash of the index combined with its content. If any unauthorized block is altered, then the sanitizer is unable to generate the proof. In addition, the sanitizer produces elements that enable our tracing mechanism to work if it exceeds its sanitization limit. In order to achieve transparency, we show how the signer can simulate these elements in the original signature. This results in two computationally identically distributed signatures outputed by Sign and Sanitize. In what follows, we describe our k-SAN scheme. The Setup algorithm is the same as in Section 4.

$\underline{\mathsf{KeyGen}(1^\lambda, k, n)}$: if $n > 1$, set $l = \lceil \log_2(k) \rceil$, and generate two SPS-EQ keys pairs $(\mathsf{pk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}, \mathsf{sk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}) \leftarrow \mathsf{KeyGen}_{\mathsf{SPS\text{-}EQ}}(1^\lambda, 4l+1)$ and $(\mathsf{pk}^{\mathsf{MOD}}_{\mathsf{S}}, \mathsf{sk}^{\mathsf{MOD}}_{\mathsf{S}}) \leftarrow \mathsf{KeyGen}_{\mathsf{SPS\text{-}EQ}}(1^\lambda, 2n)$. Sample $\mathsf{sk}_{\log} \leftarrow\!\$\ \mathbb{Z}_p^*$ and set $\mathsf{pk}_{\log} = g_1^{\mathsf{sk}_{\log}}$. Return $\mathsf{pk} = (\mathsf{pk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}, \mathsf{pk}^{\mathsf{MOD}}_{\mathsf{SPS\text{-}EQ}}, \mathsf{pk}_{\log})$ and $\mathsf{sk} = (\mathsf{sk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}, \mathsf{sk}^{\mathsf{MOD}}_{\mathsf{SPS\text{-}EQ}}, \mathsf{sk}_{\log})$.

$\underline{\mathsf{SaKeyGen}(1^\lambda)}$: sample $\mathsf{ssk}_{\log} \leftarrow\!\$\ \mathbb{Z}_q$, set $\mathsf{spk}_{\log} = g_1^{\mathsf{ssk}_{\log}}$, run $(\mathsf{ssk}_e, \mathsf{spk}_e) \leftarrow \mathsf{KeyGen}_{\mathcal{E}}(1^\lambda)$ and return $\mathsf{ssk} = (\mathsf{ssk}_{\log}, \mathsf{ssk}_e)$ as the secret key and $\mathsf{spk} = (\mathsf{spk}_{\log}, \mathsf{spk}_e)$ as the public key.

$\underline{\mathsf{Delegate}(\mathsf{sk}, \mathsf{spk}, k)}$: set $l = \lceil \log_2(k) \rceil$, abort if the SPS-EQ key $\mathsf{sk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}$ does not support messages of $4l+1$ group elements. For all $(i,j) \in [\![l]\!] \times \{0,1\}$, sample $x_{i,j} \leftarrow\!\$\ \mathbb{Z}_p^*$, set $y_{i,j} = g_1^{x_{i,j}}$, $\mathsf{spk}_{i,j} = \mathsf{spk}_{\log}{}^{x_{i,j}}$ and produce the SPS signature $\widehat{\sigma} \leftarrow \mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}(\mathsf{sk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}, g_1, y_{1,0}, \cdots, \mathsf{spk}_{l,1})$. Return $\mathsf{del} = ((x_{i,j}, y_{i,j}, \mathsf{spk}_{i,j})_{i \in [\![l]\!]; j \in \{0,1\}}, \widehat{\sigma})$.

Below, we describe the Sign and Sanitize algorithms, drawing parallels between their similarities and specifying their respective executions when they differ.

Both $\mathsf{Sign}(m, \mathsf{ADM}, \mathsf{sk}, \mathsf{spk})$ and $\mathsf{Sanitize}(m, \sigma, \mathsf{MOD}, \mathsf{ssk}, \mathsf{pk}, \mathsf{del}, \eta)$ set $l = \lceil \log_2(k) \rceil$. Then:

$\underline{\mathsf{Sign}}$**:** Parse $m \xrightarrow{p} m_1 \| \cdots \| m_n$, sample $\eta \leftarrow\!\$\ [\![0, k-1]\!]$, $s \leftarrow\!\$\ \mathbb{Z}_p^*$, and $\widehat{g}_1, \widehat{y}_{i,j}, \widehat{\mathsf{spk}}_{i,j} \leftarrow\!\$\ \mathbb{G}_1$ for all $(i,j) \in [\![l]\!] \times \{0,1\}$. Simulate a delegation by singing $\widehat{\sigma} \leftarrow \mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}(\mathsf{sk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}, \widehat{g}_1, \widehat{y}_{1,0}, \cdots, \widehat{\mathsf{spk}}_{l,1})$. For all $i \in [\![l]\!]$ and $j \in \{0,1\}$, set $\widetilde{y}_i = \widehat{y}^s_{i,\eta[i]}$, and $\widetilde{\mathsf{spk}}_i = \widehat{\mathsf{spk}}^{\,s}_{i,\eta[i]}$.

Sanitize: Parse $\mathsf{MOD}(m) \xrightarrow{p} m_1 \| \cdots \| m_n, \sigma \xrightarrow{p} (\mathsf{del}_\sigma, \mathsf{tra}, \pi_\sigma), \mathsf{del}_\sigma \xrightarrow{p} (\widehat{g_1}, \widetilde{g_1}, ((\widehat{y}_{i,b}, \widehat{\mathsf{spk}}_{i,b})_{b\in\{0,1\}}, \widetilde{y}_i, \widetilde{\mathsf{spk}}_i)_{i\in[\![l]\!]}, \widehat{\sigma}, \Pi_{<k})$ and $\mathsf{tra} \xrightarrow{p} ((u_i, v_i)_{i=1}^n, \widetilde{y}, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \tau, \pi_{\mathsf{MOD}}, \sigma_{\mathsf{MOD}}, e)$ (Note that the values of most of these variables will be updated by reallocation during the algorithm). Then proceeds similarly to the initial steps of the Sign algorithm of the k-APS signature scheme, halting before the execution of the proof $\Pi_{<k}$.

Both algorithms generate the proof $\Pi_{<k}$ of knowledge of s and η which proves that (i) $\widetilde{y}_i$ and $\widetilde{\mathsf{spk}}_i$ are well formed according to s and some integer η of l bits and (ii) $\eta < k$. This proof follows the same instantiation as before. To conclude this first part, set $\mathsf{del}_\sigma = (\widehat{g_1}, \widetilde{g_1}, ((\widehat{y}_{i,b}, \widehat{\mathsf{spk}}_{i,b})_{b\in\{0,1\}}, \widetilde{y}_i, \widetilde{\mathsf{spk}}_i)_{i\in[\![l]\!]}, \widehat{\sigma}, \Pi_{<k})$.

Both algorithms start the second phase by setting the message blocks:

Sign: To mandate the sanitizer for a set of modifiable blocks: sample $a \leftarrow\!\$\ \mathbb{Z}_p^*$. For all $i \in \mathsf{ADM}$ let $u_i = H(m_i, i, 0)^a$ and $v_i = H(m_i, i, 1)^a$, otherwise let $u_i = H(i, 0)^a$ and $v_i = H(i, 1)^a$. Encrypt $e \leftarrow \mathsf{Enc}(\mathsf{spk}_e, a)$

Sanitize: Sample $b \leftarrow\!\$\ \mathbb{Z}_p^*$, decrypt $a \leftarrow \mathsf{Dec}(\mathsf{ssk}_e, e)$ and update $e \leftarrow \mathsf{Enc}(\mathsf{spk}_e, a \cdot b)$. Set $\mathsf{ADM} = \emptyset$ and $\forall i \in [\![n]\!]$, let $u_i = H(m_i, i, 0)^{a\cdot b}$ and $v_i = H(m_i, i, 1)^{a\cdot b}$ when the signature contains $H(m_i, i, 0)^a$ and $H(m_i, i, 1)^a$, otherwise let $u_i = H(i, 0)^{a\cdot b}$ and $v_i = H(i, 1)^{a\cdot b}$. Check $\mathsf{MOD} \subset \mathsf{ADM}$ and set $a = a \cdot b$.

Both algorithms prove:

$$\pi_{\mathsf{MOD}} \leftarrow \mathsf{SoK}_e \left\{ a\colon \bigwedge_{1\le i\le n} \begin{array}{l} (u_i = H(m_i, i, 0)^a \wedge v_i = H(m_i, i, 1)^a) \\ \vee\, (u_i = H(i, 0)^a \wedge v_i = H(i, 1)^a) \end{array} \right\}.$$

Sign: execute $\sigma_{\mathsf{MOD}} \leftarrow \mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}\left(\mathsf{sk}^{\mathsf{MOD}}_{\mathsf{SPS\text{-}EQ}}, u_1, v_1, \cdots, u_n, v_n\right)$. Set $\widetilde{y} = \prod_{i=1}^l \widetilde{y}_i$, $\widetilde{\mathsf{spk}} = \prod_{i=1}^l \widetilde{\mathsf{spk}}_i$, $u = \sum_{i=1}^n u_i$, $v = \sum_{i=1}^n v_i$ and sample the elements $\alpha_1, \alpha_3, \alpha_4 \leftarrow\!\$\ \mathbb{G}_1$, $\alpha_2 \leftarrow\!\$\ \mathbb{G}_2$ and a tracing element $\tau \leftarrow\!\$\ \mathbb{G}_t$.

Sanitize: Adapt σ_{MOD} according to the randomness b: $\sigma_{\mathsf{MOD}} \leftarrow \mathsf{ChgRep}_{\mathsf{SPS\text{-}EQ}}((u_1, v_1, \cdots, u_n, v_n), \sigma_{\mathsf{MOD}}, b, \mathsf{pk}^{\mathsf{MOD}}_{\mathsf{SPS\text{-}EQ}})$. Set $x = \sum_{i=1}^l x_{i,\eta[i]}$, $\widetilde{y} = \prod_{i=1}^l \widetilde{y}_i$, $\widetilde{\mathsf{spk}} = \widetilde{y}^{\mathsf{ssk}_{\log}}$, and compute $\alpha_1 = h_1^x$. Let $u = \sum_{i=1}^n u_i$, $v = \sum_{i=1}^n v_i$ and sample $t \leftarrow\!\$\ \mathbb{Z}_p^*$. Compute $\alpha_2 = g_2^t$, the matching elements $\alpha_3 = h_2^x \cdot g_1^{u\cdot\mathsf{ssk}_{\log}}$, $\alpha_4 = h_3^x \cdot h_4^{v\cdot\mathsf{ssk}_{\log}}$ and a tracing element $\tau = e(h_4, \alpha_2)^{\mathsf{ssk}_{\log}}$.

The vector of elements $\mathsf{tra} = ((u_i, v_i)_{i=1}^n, \widetilde{y}, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \tau, \pi_{\mathsf{MOD}}, \sigma_{\mathsf{MOD}}, e)$ is set by both entities and embed in a signature of knowledge where the sanitizer proves the first part of the `or` statement and the signer the second part:

$$\pi_\sigma \leftarrow \mathsf{SoK}_{(\mathsf{del},\mathsf{tra})}\{\mathsf{sk}_{\log}\colon (\widetilde{y} = \widehat{g_1}^{x\cdot s} \wedge \widetilde{\mathsf{spk}} = \widetilde{y}^{\mathsf{ssk}_{\log}} \wedge \alpha_1 = h_1^x \wedge \alpha_2 = g_2^t \wedge$$
$$\alpha_3 = h_2^x \cdot g_1^{u\cdot\mathsf{ssk}_{\log}} \wedge \alpha_4 = h_3^x \cdot h_4^{v\cdot\mathsf{ssk}_{\log}} \wedge \tau = e(h_4, \alpha_2)^{\mathsf{ssk}_{\log}}) \vee (\mathsf{pk}_{\log} = g_1^{\mathsf{sk}_{\log}})\}.$$

Finally Sign and Sanitize return the signature $\sigma = (\mathsf{del}_\sigma, \mathsf{tra}, \pi_\sigma)$.

The *signature verification* consists of re-computing the elements that are necessary for the verification of every SPS-EQ and signature of knowledge.

We will now informally describe the security properties of k-SAN (we recall that the formal definition can be found in Appendix E) and explain why they hold on our scheme, except for anonymity, traceability and non-frameability that are reached in a similar way to our k-APS (Section 4).

Unforgeability. The users cannot generate a valid signature without knowing a secret key which has obtained a delegation. This property relies on the hardness of recovering the secret key of the signer or one of the sanitizers, which is ensured by the DDH assumption. Once this have been ruled out, we can reduce the ability of an adversary to forge a signature to its ability to forge SPS signatures.

Immutability. A sanitizable signature is *immutable* when no adversary is able to sanitize with unauthorized modification. This property relies on the collision resistance of the hash function, as well as the soundness and zero-knowledge properties of the signature of knowledge π_{MOD} (as they link the message to the signature). Moreover, the EUF-CMA security of the SPS-EQ and the DDH assumption prevent impersonation of the signer.

Transparency. The verifier cannot decide whether a given signature has been sanitized or not, which means that the outputs of Sign and Sanitize should be computationally indistinguishable. The randomised delegation encompassed in the signature is identically distributed to a newly produced one. All SoK can be produced by both the signer and the sanitizer, while the other elements are shown to computationally indistinguishable based on the DDH problem.

Invisibility. The invisibility property prevents an adversary which is not the signer nor the sanitizer of a signature from determining any information on the modifiable blocks. The difference between a modifiable block and a non-modifiable block is the input of the hash function serving as a commitment. The obtained hash is then elevated to a secret random power. Therefore, invisibility mainly relies on the class hiding property (Definition 1).

Unlinkability. Considering a fixed sanitizer assigned with two signatures, the verifier cannot link a sanitized signature with its original version. In the proposed signature scheme, all elements undergo randomization during sanitization or are entirely new, which ensures this property.

Anonymity versus unlinkability. We highlight the fact that, although conceptually close, the properties of unlinkability and anonymity capture independent attack scenarios. In unlinkability, the adversary tries to link signatures modified for a single known sanitizer, while in anonymity, the adversary has to guess the identity of an unknown sanitizer for a given message and can control the modifications this sanitizer makes to these signatures. Since in anonymity the adversary chooses for itself how and by whom signatures are modified via oracles, knowing how to link a sanitized signatures to its original gives it no advantage. Note that for signatures sanitized by the unknown sanitizer that the adversary has to determine, the sanitization oracle will always use the key of the unknown sanitizer, thus avoiding trivial attacks where the adversary tests whether the sanitization of its signature by a chosen sanitizer fails or not.

On the other hand, in unlinkability, the adversary receives a signature sanitized by a given user, and must determine the original signature used. As the original signature can only be sanitized by one sanitizer chosen *a priori* by the signer, guessing the identity of this sanitizer by attacking anonymity gives the adversary no advantage. So there is no implication between unlinkability and anonymity.

Note that when the k limit is exceeded, it is the identity of the sanitizer and the link between their signatures that are leaked, but it is still not possible to link the sanitized signatures to the original signatures; we link the signatures of a sanitizer, but the unlinkability property still holds for these signatures.

We therefore have the following theorem, the proofs are available in Appendix F.

Theorem 2. *Instanciated by a signature on equivalent classes that is unforgeable, class-hiding, and signature adaptatable, by* NIZK *proofs which are zero-knowledge and sound, by a collision-resistante hash function, by a* SoK *that is perfectly-simulability and simulation-extractability, and by an* IND-CCA *public key encryption, our* k-SAN *is unforgeable, immutable, transparent, unlinkable, anonymity, invisible, k-traceable and non-frameable under the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$ in the random oracle model.*

7 Conclusion

In this paper, we mitigate for practical purposes the delegations carried out through some signatures with anonymity. To this end, we define full traceable k-times anonymity for proxy signatures and sanitizable signatures. In both cases, we define a security model, give an efficient scheme (in the sense that the size of keys and signatures is logarithmic in k), and prove its security. In the future, we would like to address two problems that we leave open: the construction of k-times proxy/sanitizable signature schemes that produce signatures of constant size, and the construction of schemes that do not require the generic group model (required for equivalence class signatures).

Acknowledgments. This article is an extended version of a paper published at the CANS 24 conference. The authors would like to thank Jan Bobolz and the anonymous reviewers of CANS 2024 conference for their valuable

and insightful comments. Xavier Bultel's research was supported by the ANR project PRIV-SIQ (ANR-23-CE39-0008). Charles Olivier-Anclin's research was partially supported by the ANR projet MobiS5 (ARN-18-CE39-0019).

References

1. Ateniese, G., Chou, D.H., De Medeiros, B., Tsudik, G.: Sanitizable signatures. In: Computer Security–ESORICS 2005: 10th European Symposium on Research in Computer Security (2005)
2. Au, M.H., Susilo, W., Yiu, S.M.: Event-oriented k-times revocable-iff-linked group signatures. In: ACISP 2006 (2006)
3. Bossuat, A., Bultel, X.: Unlinkable and invisible γ-sanitizable signatures. In: International Conference on Applied Cryptography and Network Security (2021)
4. Brzuska, C., Fischlin, M., Freudenreich, T., Lehmann, A., Page, M., Schelbert, J., Schröder, D., Volk, F.: Security of sanitizable signatures revisited. In: Public Key Cryptography–PKC 2009: 12th International Conference on Practice and Theory in Public Key Cryptography (2009)
5. Brzuska, C., Fischlin, M., Lehmann, A., Schröder, D.: Unlinkability of sanitizable signatures. In: PKC 2010 (2010)
6. Brzuska, C., Pöhls, H.C., Samelin, K.: Efficient and perfectly unlinkable sanitizable signatures without group signatures. In: Public Key Infrastructures, Services and Applications: 10th European Workshop (2014)
7. Bultel, X., Lafourcade, P.: k-times full traceable ring signature. In: 2016 11th International Conference on Availability, Reliability and Security (2016)
8. Bultel, X., Lafourcade, P., Lai, R.W., Malavolta, G., Schröder, D., Thyagarajan, S.A.K.: Efficient invisible and unlinkable sanitizable signatures. In: 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography (2019)
9. Camenisch, J., Derler, D., Krenn, S., Pöhls, H.C., Samelin, K., Slamanig, D.: Chameleon-hashes with ephemeral trapdoors. In: PKC 2017 (2017)
10. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups. In: Advances in Cryptology — CRYPTO '97 (1997)
11. Canard, S., Jambert, A.: On extended sanitizable signature schemes. In: Cryptographers' Track at the RSA Conference (2010)
12. Chase, M., Lysyanskaya, A.: On signatures of knowledge. In: Advances in Cryptology-CRYPTO: 26th Annual International Cryptology Conference (2006)
13. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Advances in Cryptology — CRYPTO' 92 (1993)
14. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: CRYPTO '94 (1994)
15. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: CRYPTO' 86 (1987)
16. Fleischhacker, N., Krupp, J., Malavolta, G., Schneider, J., Schröder, D., Simkin, M.: Efficient unlinkable sanitizable signatures from signatures with re-randomizable keys. In: 19th IACR International Conference on Practice and Theory in Public-Key Cryptography (2016)
17. Fuchsbauer, G., Hanser, C., Slamanig, D.: Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. Journal of Cryptology (2019)
18. Fuchsbauer, G., Pointcheval, D.: Anonymous proxy signatures. In: Security and Cryptography for Networks: 6th International Conference (2008)
19. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Public Key Cryptography – PKC 2007 (2007)
20. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences (1984)
21. Groth, J., Maller, M.: Snarky signatures: Minimal signatures of knowledge from simulation-extractable snarks. In: Annual International Cryptology Conference (2017)
22. Klonowski, M., Lauks, A.: Extended sanitizable signatures. In: Proceedings of the 9th International Conference on Information Security and Cryptology. ICISC'06 (2006)
23. Krenn, S., Samelin, K., Sommer, D.: Stronger security for sanitizable signatures. In: International Workshop on Data Privacy Management (2015)
24. Liu, W., Yang, G., Mu, Y., Wei, J.: k-time proxy signature: Formal definition and efficient construction. In: Provable Security: 7th International Conference, ProvSec 2013, Melaka, Malaysia, October 23-25, 2013. Proceedings 7 (2013)
25. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures for delegating signing operation. In: Proceedings of the 3rd ACM Conference on Computer and Communications Security. CCS '96 (1996). https://doi.org/10.1145/238168.238185

26. Teranishi, I., Furukawa, J., Sako, K.: k-times anonymous authentication (extended abstract). In: ASIACRYPT 2004 (2004)
27. Wei, J., Yang, G., Mu, Y.: Anonymous proxy signature with restricted traceability. In: 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (2014)
28. Wei, J., Yang, G., Mu, Y., Liang, K.: Anonymous Proxy Signature with Hierarchical Traceability. The Computer Journal (2015)

Auxiliary Supporting Material

A Security Properties of our Building Blocks

Security of SPS-EQ *Schemes* We require that SPS-EQ meets the following requirements:

Correctness: for all $l \in \mathbb{N}$, $(\mathsf{pk}, \mathsf{sk})$, $m \in \mathbb{G}^l$, and $\mu \in \mathbb{Z}_p^*$, the following equations should be true : $\mathsf{Verif}_{\mathsf{SPS\text{-}EQ}}(m, \mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}(\mathsf{sk}, m; \mathcal{R}), \mathsf{pk}; \mathcal{R}) = 1$ and $\mathsf{Verif}_{\mathsf{SPS\text{-}EQ}}(\mu m, \mathsf{ChgRep}_{\mathsf{SPS\text{-}EQ}}(m, \mathsf{Sign}(\mathsf{sk}, m; \mathcal{R}), \mu, \mathsf{pk}; \mathcal{R}), \mathsf{pk}; \mathcal{R}) = 1$.

EUF-CMA (existantial unforgeability under adaptative chosen-message attacks)**:** let $l > 1$ and 1^λ a given security parameter;

$$\Pr\begin{bmatrix}(\mathsf{pk}, \mathsf{sk}) \in [\mathsf{KeyGen}_{\mathsf{SPS\text{-}EQ}}(1^\lambda, l; \mathcal{R})], & \\ (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}(\cdot, \mathsf{sk}; \mathcal{R})}(\mathsf{pk}, l) & : \begin{matrix}\forall m \in \mathcal{S}, m^* \notin [m]_{\mathcal{R}} \wedge \\ \mathsf{Verif}_{\mathsf{SPS\text{-}EQ}}(m, \sigma, \mathsf{pk}; \mathcal{R})\end{matrix}\end{bmatrix},$$

is negligible for every PPT adversary $\mathcal{A}$ where $\mathcal{S}$ is the set of queries that $\mathcal{A}$ has issued to the signing oracle.

Signature Adaptation: let $l > 1$ and 1^λ a given security parameter, $(\mathsf{pk}, \mathsf{sk}) \in [\mathsf{KeyGen}(1^\lambda, l; \mathcal{R})]$, $\mu \in \mathbb{Z}_p^*$ and $m \in \mathbb{G}^l$. For all tuples $(\mathsf{sk}, \mathsf{pk}, m, \sigma, \mu)$ the distributions of $\mathsf{Sign}(\mathsf{sk}, m; \mathcal{R})$ and $\mathsf{ChgRep}(m, \sigma, \mu, \mathsf{pk}; \mathcal{R})$ are identical.

Security of NIZK *proofs* A NIZK requires the following properties:

Completeness: For any $(w, \phi) \in \mathcal{R}$, $\mathsf{ZK.Verif}(\phi, \mathsf{ZK}\{w : (w, \phi) \in \mathcal{R}\}) = 1$.

Simulation-Extractability: For all PPT adversary $\mathcal{A}$ returning a valid proof on the statement ϕ^* with non-negligible probability, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ returning w^* such that $(w^*, \phi^*) \in \mathcal{R}$ with overwhelming probability. This implies that no PPT algorithm can output a valid proof on a false statement with non-negligible probability.

Zero-Knowledge: For any $(w, \phi) \in \mathcal{R}$, there exists a PPT algorithm $\mathsf{Sim}(\phi)$ that follows the same probability distribution as $\mathsf{ZK}\{w : (w, \phi) \in \mathcal{R}\}$.

Security for Asymmetric Encryption Schemes An encryption scheme $\mathcal{E}$ has to achieve *Correctness* and *Indistinguishability under Chosen Ciphertext Attack.* (IND-CCA) Formally, for all PPT adversary $\mathcal{A}$, given a decryption oracle $\mathsf{Dec}(\mathsf{sk}, \cdot)$ (which rejects the challenge c), has at most a negligible probability

$$\left|\Pr\left[(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ \mathsf{KeyGen}(1^\lambda), (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{Dec}(\mathsf{sk}, \cdot)}(\mathsf{pk}) \atop b \leftarrow\!\!\$\ \{0,1\}, c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b), b^* \leftarrow \mathcal{A}^{\mathsf{Dec}(\mathsf{sk}, \cdot)}(c)} : b = b^*\right] - \frac{1}{2}\right|.$$

B Instantiation of the Proof Π_σ

In this section, we show how to instantiate the proof π_σ used in Section 4. For the sake of clarity, we rewrite this proof with more generic notations (where for any integer i, each g_i, γ_i, and h_i is an element of a group $\mathbb{G}_i$ of the same prime order p):

$$\pi_\sigma \leftarrow \mathsf{ZK}\left\{x, y, z : \begin{matrix}h_1 = {g_1}^x \wedge h_2 = {g_2}^y \wedge h_3 = {g_3}^x \wedge h_4 = {g_4}^z \\ h_5 = {g_5}^x \cdot \gamma_5^y \wedge h_6 = {g_6}^x \cdot \gamma_6^y \wedge h_7 = {g_7}^y\end{matrix}\right\}.$$

Our construction follows the Schnorr protocol structure:

– The prover picks $(r, s, t) \xleftarrow{\$} \mathbb{Z}_p^3$ and sets $R_1 = g_1^r$; $S_2 = g_2^s$; $R_3 = g_3^r$; $T_4 = g_4^t$; $R_5 = g_5^r$; $S_5 = g_5^s$; $R_6 = g_6^s$; $S_5 = g_6^s$; and $S_7 = g_7^s$. The prover sends $(R_1, S_2, R_3, T_4, R_5, S_5, R_6, S_5, S_7)$ to the verifier.
– The verifier picks a challenge $c \xleftarrow{\$} \mathbb{Z}_p$ and sends it to the prover.
– The prover computes $\alpha = r + x \cdot c$, $\beta = s + y \cdot c$, and $\delta = t + z \cdot c$, then sends (α, β, γ) to the verifier.
– If the following equations holds, then the verifier accepts the proof, else the verifier rejects: $g_1^\alpha = R_1 \cdot h_1^c$; $g_2^\beta = S_2 \cdot h_2^c$; $g_3^\alpha = R_3 \cdot h_3^c$; $g_4^\delta = T_4 \cdot h_4^c$; $g_5^\alpha \cdot \gamma_5^\beta = R_5 \cdot S_5 \cdot h_5^c$; $g_6^\alpha \cdot \gamma_6^\beta = R_6 \cdot S_6 \cdot h_6^c$; and $g_7^\beta = S_7 \cdot h_7^c$.

This proof is **complete** by construction.

To show that this proof is **sound**, we show that knowing two valid transcripts $\tau_0 = ((R_1, S_2, R_3, T_4, R_5, S_5, R_6, S_5, S_7), c_0, (\alpha_0, \beta_0, \gamma_0))$ and $\tau_1 = ((R_1, S_2, R_3, T_4, R_5, S_5, R_6, S_5, S_7), c_1, (\alpha_1, \beta_1, \gamma_1))$ using both the same commitment $(R_1, S_2, R_3, T_4, R_5, S_5, R_6, S_5, S_7)$ but different challenges c_0 and c_1, it is possible to deduce (x, y, z) in polynomial time (special soundness). Since the two transcripts are valid, we have: $g_1^{\alpha_0} = R_1 \cdot h_1^{c_0}$; $g_2^{\beta_0} = S_2 \cdot h_2^{c_0}$; $g_3^{\alpha_0} = R_3 \cdot h_3^{c_0}$; $g_4^{\delta_0} = T_4 \cdot h_4^{c_0}$; $g_5^{\alpha_0} \cdot \gamma_5^{\beta_0} = R_5 \cdot S_5 \cdot h_5^{c_0}$; $g_6^{\alpha_0} \cdot \gamma_6^{\beta_0} = R_6 \cdot S_6 \cdot h_6^{c_0}$; $g_7^{\beta_0} = S_7 \cdot h_7^{c_0}$; $g_1^{\alpha_1} = R_1 \cdot h_1^{c_1}$; $g_2^{\beta_1} = S_2 \cdot h_2^{c_1}$; $g_3^{\alpha_1} = R_3 \cdot h_3^{c_1}$; $g_4^{\delta_1} = T_4 \cdot h_4^{c_1}$; $g_5^{\alpha_1} \cdot \gamma_5^{\beta_1} = R_5 \cdot S_5 \cdot h_5^{c_1}$; $g_6^{\alpha_1} \cdot \gamma_6^{\beta_1} = R_6 \cdot S_6 \cdot h_6^{c_1}$; and $g_7^{\beta_1} = S_7 \cdot h_7^{c_1}$. Setting $x = (\alpha_1 - \alpha_0)/(c_1 - c_0)$; $y = (\beta_1 - \beta_0)/(c_1 - c_0)$; and $z = (\delta_1 - \delta_0)/(c_1 - c_0)$ and using the equations above, we find that: $h_1 = {g_1}^x$; $h_2 = {g_2}^y$; $h_3 = {g_3}^x$: $h_4 = {g_4}^z$; $h_5 = {g_5}^x \cdot \gamma_5^y$; $h_6 = {g_6}^x \cdot \gamma_6^y$; and $h_7 = {g_7}^y$, which concludes the proof of soundness.

We show that this proof is **zero-knowledge** by giving a polynomial-time simulator that outputs transcrpits indistinguishable from the transcripts of the real protocol without using the secret value (x, y, z). The simulator picks $(c, \alpha, \beta, \delta) \xleftarrow{\$} \mathbb{Z}_p^4$; $R_5 \xleftarrow{\$} \mathbb{G}_5$; and $R_6 \xleftarrow{\$} \mathbb{G}_6$. Then the simulator computes: $R_1 = g_1^\alpha / h_1^c$; $S_2 = g_2^\beta / h_2^c$; $R_3 = g_3^\alpha / h_3^c$; $T_4 = g_4^\delta / h_4^c$; $S_5 = (g_5^\alpha \cdot \gamma_5^\beta)/(R_5 \cdot h_5^c)$; $S_6 = (g_6^\alpha \cdot \gamma_6^\beta)/(R_6$
$cdot h_6^c)$; and $S_7 = g_7^\beta / h_7^c$. The simulator returns $((R_1, S_2, R_3, T_4, R_5, S_5, R_6, S_5, S_7), c, (\alpha, \beta, \gamma))$.

Finally, as this proof is a sigma protocol, it can be made **non-interactive** using the Fiat-Shamir transformation [15].

C Proof of Theorem 1

Proof. Let $\mathcal{A}$ be a PPT adversary against each of the experiments. $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_i, \mathsf{G}_{i+1}}(\mathcal{A})$ denotes the probability $|\Pr[\mathsf{G}_i(\mathcal{A}) = 1] - Pr[\mathsf{G}_{i+1}(\mathcal{A}) = 1]|$. We investigate each of the properties independently.

Correctness. It is verified by investigation.

Unforgeability. Let $\mathsf{Game}_0^{\mathsf{unf}}$ denote the experiment $\mathsf{Exp}^{\mathsf{unf}}_{\mathsf{k\text{-}APS}, \mathcal{A}}(1^\lambda)$ instantiated by the Log Size k-APS.

$\underline{\mathsf{Game}_1^{\mathsf{unf}}}$: is similar to $\mathsf{Game}_0^{\mathsf{unf}}$ but we abort if there is a collision for the responses of the hash function in the elements that the challenger sees.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_0^{\mathsf{unf}}$ and $\mathsf{Game}_1^{\mathsf{unf}}$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_0, \mathsf{G}_1}(\mathcal{A}) \leq \mathsf{Adv}_H^{\mathsf{col-resist}}$.
The reduction is straightforwardly achieved based on a record of the hashs. The reduction returns the collisions it sees.

$\underline{\mathsf{Game}_2^{\mathsf{unf}}}$: is similar to $\mathsf{Game}_1^{\mathsf{unf}}$ but the witness $(\mathsf{psk}^*, \mathsf{psk}_i{}^*, r^*)$ is extracted from the proof NIZK proof π_σ^* and matched with the signature's elements.
Claim. We claim that the adversary's advantage in hybrids $\mathsf{Game}_1^{\mathsf{unf}}$ and $\mathsf{Game}_2^{\mathsf{unf}}$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_1, \mathsf{G}_2}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{sound}}_{\mathsf{NIZK}}$.
The reduction is direct to the soundness of the NIZK proof.

$\underline{\mathsf{Game}_3^{\mathsf{unf}}}$ (enabling step): is similar to $\mathsf{Game}_2^{\mathsf{unf}}$ but the proof π_σ is simulated on calls to the signing oracle. The reduction directly follows from the zero-knowledge property of the NIZK proof. Hence, the adversary's advantage only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_2, \mathsf{G}_3}(\mathcal{A}) \leq q_{\mathsf{Sign}} \cdot \mathsf{Adv}^{\mathsf{ZK}}_{\mathsf{NIZK}}$, where q_{Sign} represent the number of calls to the signing oracle.

$\underline{\mathsf{Game}_4^{\mathsf{unf}}}$ (enabling step): is similar to $\mathsf{Game}_3^{\mathsf{unf}}$ but we define $h_4 = g_1^{r_4}$ based on a random value $r_4 \leftarrow\!\$\ \mathbb{Z}_p^*$. This elements keeps the same distribution, thus, the adversary has indistinguishable viewing of these two experiments.

$\underline{\mathsf{Game}_5^{\mathsf{unf}}}$: is similar to $\mathsf{Game}_4^{\mathsf{unf}}$ but we abort if the signature σ^* has been produced for a registered user. This is check by verifying if $(g_1^{\mathsf{psk}^*}, \mathsf{psk}^*) \in \mathcal{U}$.

Claim. We claim that the adversary's advantage in hybrids $\mathsf{Game}_4^{\mathsf{unf}}$ and $\mathsf{Game}_5^{\mathsf{unf}}$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_4,\mathsf{G}_5}(\mathcal{A}) \leq q_{\mathcal{U}} \cdot \mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}$.

Reduction. Consider a sequence of hybrids $H_1, \cdots, H_{q_{\mathcal{U}}}$, where we expect the difference in between two consecutive experiments to be at most $\mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}$. Define H_i as $\mathsf{Game}_4^{\mathsf{unf}}$ where the game is aborted if $(g_1^{\mathsf{psk}^*}, \mathsf{psk}^*) \in \mathcal{U}$ was produced during the i first calls to $\mathcal{O}^{\mathsf{unf}}_{\mathsf{Register}}$. It follows that $H_0 = \mathsf{Game}_4^{\mathsf{unf}}$ and $H_{q_{\mathcal{U}}} = \mathsf{Game}_5^{\mathsf{unf}}$. Consider $\mathcal{R}_i$ the reduction simulating H_i and additionally receiving a challenge $X = g_1^x \in \mathbb{G}_1$ to the DL problem. On the i^{th} call to $\mathcal{O}^{\mathsf{unf}}_{\mathsf{Register}}$, $\mathcal{R}_i$ sets the key of the user generated on as $\mathsf{ppk} = X$. From the previous bridging, we can compute the elements involving the key psk: $\alpha_3 = h_2^x \cdot \mathsf{ppk}^u; \alpha_4 = h_3^x \cdot \mathsf{ppk}^{v \cdot r_4}; \tau = e(\mathsf{ppk}^{r_4}, \alpha_2)$ an output π_σ based on the simulator, which allows us to output valid signatures. Receiving a response from $\mathcal{A}$, we can transfer the extracted value psk^* as an answer to the challenger of the DL problem. An adversary winning against H_i and not against H_{i+1} would have output a valid answer, as this is only possible with negligible probability $\mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}$, we have proven our claim.

Analysis. Forgery of the NIZK proof π_σ or adversarial produced NIZK proofs for registered users leads to an abort in $\mathsf{Game}_5^{\mathsf{unf}}$. Hence, the adversary $\mathcal{A}$ can only output signatures for unregistered users.

Claim. The adversary's $\mathcal{A}$ advantage in hybrid $\mathsf{Game}_5^{\mathsf{unf}}$ is negligible, given that the SPS-EQ scheme is existentially unforgeable under adaptive chosen-message attacks, *i.e.,* $\mathsf{Adv}^{\mathsf{unf}}_{\mathsf{G}_5}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SPS\text{-}EQ}}$.

Reduction. Consider an adversary $\mathcal{A}$ winning against $\mathsf{Game}_5^{\mathsf{unf}}$. Let $\mathcal{R}$ be a reduction emulating between the answers of $\mathcal{A}$ and $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SPS\text{-}EQ}}$. We implement $\mathcal{R}$ straightforwardly setting pk as the public key received from the challenger against $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SPS\text{-}EQ}}$. To issue $\widehat{\sigma}$ on a call to $\mathcal{O}^{\mathsf{unf}}_{\mathsf{Delegate}}$, $\mathcal{R}$ calls the signing oracle for the message $(g_1, y_{1,0}, \cdots, \mathsf{ppk}_{l,1})$. For a winning adversary outputting a pair (m^*, σ^*), it holds that $\mathsf{Verif}_{\mathsf{SPS\text{-}EQ}}(\mathsf{pk}, (\widehat{g_1}^*, \widehat{y}^*_{1,0}, \cdots, \widehat{\mathsf{ppk}^*_{l,1}}), \widehat{\sigma}^*) = 1$. The message-signature pair is transfer to the challenger of the EUF-CMA experiment and a bit b is returned. As previously established, a successful adversary $\mathcal{A}$ must generate a new delegation to win in $\mathsf{Game}_5^{\mathsf{unf}}$. This implies that $\mathcal{R}$ would produce a valid forgery for the SPS-EQ signature, which contradicts the EUF-CMA property of the SPS-EQ signature.

Anonymity. Let $\mathsf{Game}_0^{\mathsf{Ano}}$ denote the experiment $\mathsf{Exp}^{\mathsf{Ano}}_{\mathsf{k\text{-}APS},\mathcal{A}}(1^\lambda)$ instantiated by our k-APS. What we then call *challenge* is the output of the oracle $\mathcal{O}^{\mathsf{Ano}}_{\mathsf{chal}}$ which is generated when the adversary calls this oracle. We modify the *challenge* sent to the adversary to decorrelate it from the identity of the proxy signer who issued it. In this game hope, we modify the *challenge* to decoralt it from the identity of the proxy signer making it.

$\underline{\mathsf{Game}_1^{\mathsf{Ano}}}$: is similar to $\mathsf{Game}_0^{\mathsf{Ano}}$ but generates new SPS-EQ signatures $\widehat{\sigma}$ for the randomised messages base on the secret sk instead of randomising the certificate.

Claim. We claim that adversary's advantage is left unchanged under this modification, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_0,\mathsf{G}_1}(\mathcal{A}) = 0$.

Reduction. Randomised signatures and newly generated ones follow identical distributions, hence $\mathcal{A}$ has indistinguishable viewing of these two experiments.

$\underline{\mathsf{Game}_2^{\mathsf{Ano}}}$: is similar to $\mathsf{Game}_1^{\mathsf{Ano}}$ but the NIZK proofs $\Pi_{<k}$ and π_σ in the challenge σ are simulated. As argued previously, this results in a negligible difference in the adversary's advantage, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_1,\mathsf{G}_2}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\mathsf{ZK}}_{\mathsf{NIZK}}$.

$\underline{\mathsf{Game}_3^{\mathsf{Ano}}}$ (enabling step): is similar to $\mathsf{Game}_2^{\mathsf{Ano}}$ but sets $h_1 = g_1^{r_1}$, $h_2 = g_1^{r_2}$, $h_3 = g_1^{r_3}$ and $h_4 = g_1^{r_4}$ from sampled values $r_1, r_2, r_3, r_4 \leftarrow\!\$\ \mathbb{Z}_p^*$. The distribution of these elements remains unchanged, hence $\mathcal{A}$ has indistinguishable viewing of these experiments, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_2,\mathsf{G}_3}(\mathcal{A}) = 0$.

$\underline{\mathsf{Game}_4^{\mathsf{Ano}}}$: is similar to $\mathsf{Game}_3^{\mathsf{Ano}}$ but for all $k \in \{0,1\}$, $i \in [\![l]\!]$, $j \in \{0,1\}$ elements $\mathsf{ppk}_{i,j}^k$, are sampled at random within $\mathbb{G}_1$.
Claim. We claim that the adversary's advantage in hybrids $\mathsf{Game}_3^{\mathsf{Ano}}$ and $\mathsf{Game}_4^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_3,\mathsf{G}_4}(\mathcal{A}) \leq 4l \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}$.

Reduction. Consider a sequence of hybrids $H_0, \cdots H_{2l}$, such that for all $i \in [\![l]\!]$, $j \in \{0,1\}$ in H_{2i+j}, the first $2i+j$ elements of the vector $(\mathsf{ppk}_{1,0}, \cdots, \mathsf{ppk}_{l,1})$ are sampled randomly at uniform, the remaining ones are generated similarly to $\mathsf{Game}_3^{\mathsf{Ano}}$. We see that $H_0 = \mathsf{Game}_3^{\mathsf{Ano}}$ and $H_{2l} = \mathsf{Game}_4^{\mathsf{Ano}}$. Now consider the reduction $\mathcal{R}_{2i+j}$ in between H_{2i+j} and H_{2i+j+1} receiving a DDH challenge (X, Y, Z). The reduction sets $\mathsf{ppk} = X$, and $y_{2i,1} = Y$, if $j = 0$ or $y_{2(i+1),0} = Y$, when $j = 1$. Based on the simulator of the proof π_σ is simulated, and the following equalities: $\alpha_3 = \widetilde{y}^{r_2} \cdot \mathsf{ppk}^u$; $\alpha_4 = \widetilde{y}^{r_3} \cdot \mathsf{ppk}^{v \cdot r_4}$; $\tau = e(\mathsf{ppk}^{r_4}, \alpha_2)$, and $\alpha_1 = Y^{\sum_{k=1;k\neq i}^{l} x_{k,\eta[k]}}$, then, $\mathcal{R}$ can perfectly simulate the remaining of the actions prescribed by the experiments. Given a distinguisher between hybrids H_{2i+j} and H_{2i+j+1} emulated by the reduction $\mathcal{R}_{2i+j}$, the latter returns the obtained decisions bit b to the challenger of the DDH problem. $\mathcal{R}_{2i+j}$ succeeds to the DDH problem with the same probability that the distinguisher has to differentiate in between the two hybrids. The same reduction is now applied for the elements generated to proxy of index 1, which leads to the proof of the claim.

$\underline{\mathsf{Game}_5^{\mathsf{Ano}}}$: is similar to $\mathsf{Game}_4^{\mathsf{Ano}}$ but elements $(\widehat{g_1}, \widehat{y}_{1,0}, \cdots, \widehat{\mathsf{ppk}}_{l,1})$ used to produce σ are sample uniformly at random in $\mathbb{G}_1^{4l+1}$.
Claim. We claim that the adversary's advantage in hybrids $\mathsf{Game}_4^{\mathsf{Ano}}$ and $\mathsf{Game}_5^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_4,\mathsf{G}_5}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{class-hid}}_{\mathbb{G}_1}$.

Reduction. Let $\mathcal{R}$ be a reduction based on a challenge from the class-hiding experiment in $\mathbb{G}_1$. The reduction $\mathcal{R}$ receives two elements $M, M' \in \mathbb{G}_1^{4l+1}$. During the setup it defines $g_1 \leftarrow M_1$ (the first element of vector M), then executing $\mathsf{Delegate}(\mathsf{sk}, \mathsf{ppk}^b, t)$, it signs M as $\widehat{\sigma}$. During the execution of $\mathsf{Sign}(\mathsf{pk}, \mathsf{psk}^b, m^*, \mathsf{del}_b, i^*)$, it inputs M' into the SPS-EQ signature scheme, thus obtaining $\widehat{\sigma} \leftarrow \mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}(\mathsf{sk}, M')$ embedded in the signature σ with M'. The rest of the experiment is executed normally. Based on the challenge M', we either emulate $\mathsf{Game}_4^{\mathsf{Ano}}$ when M' has been picked in the equivalent class of M, or $\mathsf{Game}_5^{\mathsf{Ano}}$ when M' has been picked at random. As a result, a distinguisher between $\mathsf{Game}_4^{\mathsf{Ano}}$ and $\mathsf{Game}_5^{\mathsf{Ano}}$, has a negligible probability of success.

$\underline{\mathsf{Game}_6^{\mathsf{Ano}}}$: is similar to $\mathsf{Game}_5^{\mathsf{Ano}}$ but α_3 is sampled uniformly at random in $\mathbb{G}_1$, when producing the signature σ.
Claim. We claim that the adversary's advantage in hybrids $\mathsf{Game}_5^{\mathsf{Ano}}$ and $\mathsf{Game}_6^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_5,\mathsf{G}_6}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}$.

Reduction. Based on a DDH challenge $(X = g_1^x, Y = g_1^y, Z)$, a reduction $\mathcal{R}$ set $h_2 = X$ during the setup, $\widetilde{y} = Y$ to produce the signature σ and generates the $\widetilde{y}_i$ in order to preserve $\widetilde{y} = \prod_{i=1}^{l} \widetilde{y}_i$. Among $(\widetilde{y}_i)_{i \in [\![l]\!]}$, $l-1$ elements are generated normally and the remaining element is set to $\widetilde{y}_l = \widetilde{y} \cdot \left(\prod_{i=1}^{l-1} \widetilde{y}_i\right)^{-1}$, thus ensuring the same distribution as before. Then set $\alpha_3 = Z \cdot g_1^{u \cdot \mathsf{psk}}$. Knowing the discrete logarithm of h_3, $\mathcal{R}$ we can compute $\alpha_4 = Y^{r_3} \cdot h_4^{v \cdot \mathsf{psk}}$ and it simulates the NIZK proofs $\Pi_{<k}$ and π_σ as prescribed by the experiment.When $Z = g_1^{xy}$ we have perfectly simulated $\mathsf{Game}_5^{\mathsf{Ano}}$ and when $Z \leftarrow\!\$\ \mathbb{G}_1$, we have perfectly simulated $\mathsf{Game}_6^{\mathsf{Ano}}$. Hence, distinguishing between these two experiments implies distinguishing between the two event of the DDH problem.

$\underline{\mathsf{Game}_7^{\mathsf{Ano}}}$: is similar to $\mathsf{Game}_6^{\mathsf{Ano}}$ but we sample α_4 it at random $\alpha_4 \leftarrow\!\$\ \mathbb{G}_1$ when producing the signature σ.
Claim. We claim that the adversary's advantage in hybrids $\mathsf{Game}_6^{\mathsf{Ano}}$ and $\mathsf{Game}_7^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_6,\mathsf{G}_7}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}$.

Reduction. The reduction is analogous to the previous one: consider a reduction $\mathcal{R}$ basing itself on a DDH challenge $(X = g_1^x, Y = g_1^y, Z)$ and emulating either $\mathsf{Game}_6^{\mathsf{Ano}}$ or $\mathsf{Game}_7^{\mathsf{Ano}}$ based on the value of Z. The reduction sets $h_3 = X$ during the setup, it sets $\widetilde{y} = Y$ and the generates elements $\widetilde{y_i}$ for $i \in [\![l]\!]$, preserving $\widetilde{y} = \prod_{i=1}^{l} \widetilde{y}_i$. Then it sets $\alpha_4 = Z \cdot h_4^{v \cdot \mathsf{psk}}$ based on the challenge and compute $\alpha_1 = Y^{\sum_{k=1;k\neq i}^{l} x_{k,\eta[k]}}$, and $\alpha_3 = \widetilde{y}^{r_2} \cdot g_1^{\mathsf{psk} \cdot u}$. The rest of the experiment is executed as it should have been in both game. Distinguishing between these two experiments implies distinguishing between the two event of the DDH problem as these two games only differs by a DDH challenge.

$\underline{\mathsf{Game}_8^{\mathsf{Ano}}}$: is similar to $\mathsf{Game}_7^{\mathsf{Ano}}$ but we sample an element $Z \leftarrow\!\!\$\ \mathbb{G}_1$ at the beginning of the game and let $\tau = e(Z, \alpha_2)$ to produce a signature with psk_0.
Claim. We claim that the adversary's advantage in hybrids $\mathsf{Game}_7^{\mathsf{Ano}}$ and $\mathsf{Game}_8^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_7,\mathsf{G}_8}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}$.

Reduction. Consider a reduction $\mathcal{R}$ taking as input a DDH challenge $(X, Y, Z) \in \mathbb{G}_1^3$. $\mathcal{R}$ defines $h_4 = X$ during the setup and $\mathsf{ppk}_0 = Y$ during the key generation of the proxy associated to index 0. The element τ is meant to be computed as $\tau = e(h_4, \alpha_2)^{\mathsf{psk}_b} = e(h_4^{\mathsf{psk}_b}, \alpha_2)$. Based on the DDH challenger we set $\tau = e(Z, \alpha_2)$. The remaining computation are conducted as follow: $\alpha_3 = h_2^x \cdot Y^u$ $\alpha_4 = h_3^x \cdot Z^v$ and algorithms Setup, KeyGen; $\mathsf{PKeyGen}$, $\mathsf{Delegate}$ and Sign for psk_1 remain unchanged. It is important to ensure that the threshold of k signature is not overpasses as no tracing could be possible for the produced signature as it is done in $\mathcal{O}^{\mathsf{Ano}}_{\mathsf{Sign}}$. The bit returned by the adversary $\mathcal{A}$ is then transferred as the decision against the DDH challenge. When $Z = g^{xy}$ we perfectly emulate $\mathsf{Game}_7^{\mathsf{Ano}}$ otherwise $\mathsf{Game}_8^{\mathsf{Ano}}$, our claim follows.

$\underline{\mathsf{Game}_9^{\mathsf{Ano}}}$: is similar to $\mathsf{Game}_8^{\mathsf{Ano}}$ but we sample a new Z for each signature request to the signer of index 0 and set $\tau = e(Z, \alpha_2)$.
Claim. We claim that the adversary's advantage in hybrids $\mathsf{Game}_8^{\mathsf{Ano}}$ and $\mathsf{Game}_9^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_8,\mathsf{G}_9}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{class-hid}}_{\mathbb{G}_2}$.

Reduction. This reduction rely on the fact that $\tau = e(Z, \alpha_2) = e(g_1, \alpha_2^{\log_{g_1}(Z)})$. Consider a reduction $\mathcal{R}$ based on a challenge from the class hiding experiment in $\mathbb{G}_2$ receiving two elements $M, M' \in \mathbb{G}_2^{q_S}$. We refer to α_2 (*resp.* τ) on the i^{th} call from $\mathcal{A}$ to the $\mathcal{O}^{\mathsf{Ano}}_{\mathsf{Sign}}$ as $\alpha_{2,i}$ (*resp.* τ_i). Let $\alpha_{2,i} = M_i$ and $\tau_i = e(g_1, M'_i)$ for all $i \in [\![q_S]\!]$. Based on the challenge, we have either $\tau_i = e(g_1, M_i^r)$, for all i and an integer r fixed for all sanitization of index 0, or either $\tau_i = e(g_1, M'_i)$, for a new random element M'_i changed for each sanitization of index 0. As a result, we emulate perfectly one or other of the games. We can forward the adversary's response to the challenger of the class hiding experiment and we win against this game with equal probability. This prove the claim.

$\underline{\mathsf{Game}_{10}^{\mathsf{Ano}}}$: is similar to $\mathsf{Game}_9^{\mathsf{Ano}}$ but we apply the last two modifications to the actions of the proxy using the key psk_1. This leads the same modifications of the adversary's advantage.
In Experiment $\mathsf{Game}_{10}^{\mathsf{Ano}}$, the elements provided to $\mathcal{A}$ are entirely independent of the value b. Any guessing strategy employed by $\mathcal{A}$ would result in a zero advantage because the distribution of the outputs produced by the adversary $\mathcal{A}$ is unrelated to the uniformly distributed value $b \leftarrow\!\!\$\ \{0, 1\}$. This concludes our proof for this property.

Traceability. Let $\mathsf{Game}_0^{\mathsf{Trace}}$ be the experiment $\mathsf{Exp}^{\mathsf{Trace}}_{\mathsf{k\text{-}APS},\mathcal{A}}(1^\lambda)$ instantiate with the k-APS signature of Section 4.

$\underline{\mathsf{Game}_1^{\mathsf{Trace}}}$: is similar to $\mathsf{Game}_0^{\mathsf{Trace}}$ but we abort if there is a collision for the responses of the hash function. As argued in $\mathsf{Game}_1^{\mathsf{unf}}$ the adversary's $\mathcal{A}$ advantage differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_0,\mathsf{G}_1}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{col-resist}}_{H}$.

$\underline{\mathsf{Game}_2^{\mathsf{Trace}}}$: is similar to $\mathsf{Game}_1^{\mathsf{Trace}}$ but for all $\mathcal{A}$'s responses, the NIZK proofs $\pi^*_{\sigma,j}$ and $\Pi^j_{<k}$ are extracted. The experiment is aborted if any of the extractions fails or if a valid proof for an invalid statement has

been produced. The reduction to the soundness of the proofs is direct through an hybrid sequence. We can conclude that: $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_1,\mathsf{G}_2}(\mathcal{A}) \leq 2q_s \cdot \mathsf{Adv}^{\mathsf{sound}}_{\mathsf{NIZK}}$.

$\underline{\mathsf{Game}^{\mathsf{Trace}}_3}$: is similar to $\mathsf{Game}^{\mathsf{Trace}}_2$ but the indices η_j extracted from the proofs $\Pi^j_{<k}$, for $j \in [\![q_S]\!]$ are used to verify if the adversary overpasses the signature limitations. If there exist η such that the public key ppk did not receive at least η delegation or if there exist a second occurrence of η for the same public key, we abort the experiment. The adversary's $\mathcal{A}$ advantage is left unmodified under this change: $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_2,\mathsf{G}_3}(\mathcal{A}) = 0$.

Analysis. It is now ensured that $\mathcal{A}$ has produced valid proofs of knowledge, in particularly from all the π^*_σ, the elements $\alpha_3 = h_2^x \cdot g_1^{u \cdot \mathsf{psk}}$, $\alpha_4 = h_3^x \cdot h_4^{v \cdot \mathsf{psk}}$ $\tau = e(h_4, \alpha_2)^{\mathsf{psk}}$ are well formed and correspond to certified keys. Based on the correctness, we can always recover $\mathsf{ppk} = (\alpha_3/\alpha'_3)^{1/(u-u')}$ and $w = (\alpha_4/\alpha'_4)^{1/(v-v')}$ when the number of signature has overpass the number obtained in the delegations for one user. Hence, it cannot win this game unless it forges a SPS-EQ signature for the original signer's key.
Claim. The adversary's $\mathcal{A}$ advantage in hybrid $\mathsf{Game}^{\mathsf{Trace}}_3$ is negligible, given the SPS-EQ scheme is existentially unforgeable under adaptive chosen-message attacks, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_2,\mathsf{G}_3}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SPS\text{-}EQ}}$.

Reduction. Similar to the conclusion of the proof of unforgeability.

Non-Frameability. Let $\mathsf{Game}^{\mathsf{no-Frame}}_0$ be the experiment of Non-Frameability instantiated with our k-APS scheme of Section 4.

$\underline{\mathsf{Game}^{\mathsf{no-Frame}}_1}$: is similar to $\mathsf{Game}^{\mathsf{no-Frame}}_0$ but we abort if there is a collision for the responses of the hash function in the elements that the challenger sees. Once more the adversary's $\mathcal{A}$ advantage differs by $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_0,\mathsf{G}_1}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{col-resist}}_{H}$.

Analysis. This prevent from an adversary outputting $u^1 = H(m^1, 0, \alpha^1_2) = H(m^2, 0, \alpha^2_2) = u_2$ such that ppk would be set to 0 in the Trace algorithm.

$\underline{\mathsf{Game}^{\mathsf{no-Frame}}_2}$: is similar to $\mathsf{Game}^{\mathsf{no-Frame}}_1$ but we abort the experiment if two proxy public keys produced by the challenger are the same.
Claim. We claim that the adversary's advantage in hybrids $\mathsf{Game}^{\mathsf{no-Frame}}_1$ and $\mathsf{Game}^{\mathsf{no-Frame}}_2$ only differs by a negligible factor, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_1,\mathsf{G}_2}(\mathcal{A}) \leq |\mathcal{U}|/|G_1|$.
Secret keys psk are sampled uniformly within the group $\mathbb{Z}^*_p$, which has the same order as $\mathbb{G}_1$. The probability to draw to equal keys based on $|\mathcal{U}|$ independent and identically distributed draw is $|\mathcal{U}|/|G_1|$.

$\underline{\mathsf{Game}^{\mathsf{no-Frame}}_3}$: is similar to $\mathsf{Game}^{\mathsf{no-Frame}}_2$ but the NIZK proofs π_σ in the signatures returned by $\mathcal{A}$ are extracted. The soundness of the proof is verified and the experiment is aborted if valid proof for invalid statements are provided. As argued before we obtain the following difference in the advantages: $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_2,\mathsf{G}_3}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\mathsf{sound}}_{\mathsf{NIZK}}$.

Analysis. Soundness of the NIZK proof ensures that $\mathcal{A}$ holds the witness $(\mathsf{psk}^i, x^i, r^i)$ associated to π^i_σ for $i \in \{1, 2\}$ and that (α^1_3, α^1_4) and (α^2_3, α^2_4) are correctly computed. If $\mathcal{A}$ wins, then we want to show that we have extracted the discrete logarithm of the public keys of one of the users.

$\underline{\mathsf{Game}^{\mathsf{no-Frame}}_4}$ (enabling step): is similar to $\mathsf{Game}^{\mathsf{no-Frame}}_3$ but the challenger defines $h_4 = g_1^{r_4}$ for $r_4 \leftarrow\!\!\$\ \mathbb{Z}^*_p$. The adversary has indistinguishable viewing of these experiments as the distribution of h_4 is left unchanged.

$\underline{\mathsf{Game}^{\mathsf{no-Frame}}_5}$ (enabling step): is similar to $\mathsf{Game}^{\mathsf{no-Frame}}_4$ but the NIZK proofs π_σ are simulated. This leads to a negligible difference of the adversary's advantage, *i.e.,* $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_4,\mathsf{G}_5}(\mathcal{A}) \leq q_{\mathsf{Sign}} \cdot \mathsf{Adv}^{\mathsf{ZK}}_{\mathsf{NIZK}}$.
Claim. The adversary's $\mathcal{A}$ advantage in hybrid $\mathsf{Game}^{\mathsf{no-Frame}}_5$ is negligible, given that the discrete logarithm problem is hard, *i.e.,* $\mathsf{Adv}^{\mathsf{no-Frame}}_{\mathsf{G}_5}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}$.

Reduction. Consider a reduction $\mathcal{R}$ emulating $\mathsf{Game}^{\mathsf{no-Frame}}_5$ based on a challenge X for the DL problem. For a registration request from $\mathcal{A}$, it sets set $\mathsf{ppk} = X^{s_i}$ for $s_i \leftarrow\!\!\$\ \mathbb{Z}_p$. On signing requests, the elements involving the public key ppk are computed as follows: $\alpha_3 = h_2^x \cdot \mathsf{ppk}^u$, $\alpha_4 = h_3^x \cdot \mathsf{ppk}^{r_4 \cdot v}$ and $\tau = e(\mathsf{ppk}^{r_4}, \alpha_2)$. All

these elements follows the same distribution as before, hence, $\mathsf{Game}_5^{\mathsf{no-Frame}}$ is perfectly simulated. On $\mathcal{A}$'s success, sk^1 and sk^2 are extracted consistently from both proofs. The value $\mathsf{sk} = \mathsf{sk}^1 \cdot s_i^{-1}$ for the according index i is returned to $\mathcal{R}$'s challenger. The reduction $\mathcal{R}$ has the same probability as $\mathcal{A}$ to win against the DL problem. □

D An Example for the Proof $\Pi_{<k}$

In this section, we give more details about the structure of the second part of the proof $\Pi_{<k}$, then we show an example that illustrates how the proof works and why it is linear in l. We first recall some facts about sigma protocols and `or`-proofs. A Sigma protocol is made up of three interractions, enabling the exchange of a commitment R, a challenge c, and a response z. Usually, the simulator of such a protocol for some discrete logarithm relation in a group of prime order p randomly picks a challenge $c \in \mathbb{Z}_p^*$ and a response $z \in \mathbb{Z}_p^*$, then computes the comitment R from (c, z) to complete the simulated transcript (R, c, z).

The Cramer *et al.* `or`-proof transformation [14] transfoms n sigma protocols sharing the same challenge space for the respectives statements/relations $(\phi_i)_{i\in[\![n]\!]}$ and $(\mathcal{R}_i)_{i\in[\![n]\!]}$ denoted $\mathsf{ZK}\{w : (w, \phi_i) \in \mathcal{R}_i\}$ into a `or`-proof sigma protocol $\mathsf{ZK}\left\{w : \bigvee_{i\in[\![n]\!]}(w, \phi_i) \in \mathcal{R}_i\right\}$. This transformation works as follows: assume that the prover knows the witness w_j for the statement/relation ϕ_j and $\mathcal{R}_j$. It first produces the commitment R_j for ϕ_j as in the proof $\mathsf{ZK}\{w : (w, \phi_j) \in \mathcal{R}_j\}$, then simulates the transcripts (R_i, c_i, z_i) for the other statements ϕ_i where $i \neq j$. It sends the commitments $(R_i)_{i\in[\![n]\!]}$ to the verifier and receives the challenge c. The prover then computes $c_j = c \oplus \bigoplus_{i\in[\![n]\!];i\neq j} c_i$, computes the response z_j from w_j, R_j and c_j as in $\mathsf{ZK}\{w : (w, \phi_j) \in \mathcal{R}_j\}$, and returns $(c_i, z_i)_{i\in[\![n]\!]}$ to the verifier. The verifier checks that $c = \bigoplus_{i\in[\![n]\!]} c_i$, and checks that each transcript (R_i, c_i, z_i) is valid according to ϕ_i and $\mathcal{R}_i$ for $i \in [\![n]\!]$.

On the other hand, the `and`-proof transformation we use to build the zero-knowledge proofs

$$\mathsf{ZK}\left\{(w_i)_{i\in[\![n]\!]} : \bigwedge_{i\in[\![n]\!]}(w, \phi_i) \in \mathcal{R}_i\right\}$$

consists in running the proofs $\mathsf{ZK}\{w_i : (w_i, \phi_i) \in \mathcal{R}_i\}$ in parallel by using a unique challenge c: the prouver sends the commitments $(R_i)_{i\in[\![n]\!]}$, receives a challenge c, and outputs the responses $(z_i)_{i\in[\![n]\!]}$ such that each (R_i, c, z_i) is a valid transcript for the statement/relation $(\phi_i, \mathcal{R}_i)$.

In what follows, we will show how the second part of the proof $\Pi_{<k}$ works for the example $k = 1001101$ given in Section 4:

$$\mathsf{ZK}\left\{s : \begin{array}{l}(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_0 = \widehat{y}_{0,0}^s) \vee ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_1 = \widehat{y}_{1,0}^s)\\ \wedge(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_2 = \widehat{y}_{2,0}^s) \wedge ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_3 = \widehat{y}_{3,0}^s)\\ \vee((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_4 = \widehat{y}_{4,0}^s) \vee ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_5 = \widehat{y}_{5,0}^s)\\ \wedge(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_6 = \widehat{y}_{6,0}^s)))))\end{array}\right\}.$$

Throughout this section:

- $\mathcal{R}$ denotes the relation:

$$\begin{array}{l}(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_0 = \widehat{y}_{0,0}^s) \vee ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_1 = \widehat{y}_{1,0}^s) \wedge (\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_2 = \widehat{y}_{2,0}^s)\\ \wedge((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_3 = \widehat{y}_{3,0}^s) \vee ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_4 = \widehat{y}_{4,0}^s) \vee ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_5 = \widehat{y}_{5,0}^s)\\ \wedge(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_6 = \widehat{y}_{6,0}^s))))).\end{array}$$

- $\mathcal{R}_0$ denotes the relation $(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_0 = \widehat{y}_{0,0}^s)$.
- $\mathcal{R}_{1,2-}$ denotes the relation:

$$\begin{array}{l}(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_1 = \widehat{y}_{1,0}^s) \wedge (\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_2 = \widehat{y}_{2,0}^s) \wedge ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_3 = \widehat{y}_{3,0}^s)\\ \vee((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_4 = \widehat{y}_{4,0}^s) \vee ((\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_5 = \widehat{y}_{5,0}^s) \wedge (\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_6 = \widehat{y}_{6,0}^s)))).\end{array}$$

- $\mathcal{R}_3$ denotes the relation $(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_3 = \widehat{y}_{3,0}^s)$.

– $\mathcal{R}_4$ denotes the relation $(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_4 = \widehat{y}_{4,0}^s)$.
– $\mathcal{R}_{5,6}$ denotes the relation $(\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_5 = \widehat{y}_{5,0}^s) \wedge (\widetilde{g}_1 = \widehat{g}_1^s \wedge \widetilde{y}_6 = \widehat{y}_{6,0}^s)$.

Moreover, (R_x, c_x, z_x) will denote the transcript for the relation $\mathcal{R}_x$ in the proof. According to the boolean structure of the relation $\mathcal{R}$, the challenges c chosen by the verifier and the challenges $c_0, c_{1,2-}, c_3, c_4$ and $c_{5,6}$ sent by the verifier must to verify the following equations:

$$c = c_0 \oplus c_{1,2-} \ ;$$
$$c_{1,2-} = c_3 \oplus c_4 \oplus c_{5,6} \ .$$

If the prover is honest (*i.e.,* $\mathcal{R}$ holds), then we have the following cases:

Case $\eta = 1001100$**:** the relations $\mathcal{R}_{1,2-}$ and $\mathcal{R}_{5,6}$ hold, but the relations $\mathcal{R}_0$, $\mathcal{R}_3$ and $\mathcal{R}_4$ are not verified. The prover chooses (c_0, z_0), (c_3, z_3) and (c_4, z_4), then simulates the transcripts for these relations. It then receives c from the verifier, which fixes the values of $c_{1,2-}$ and $c_{5,6}$:

$$c_{1,2-} = c_0 \oplus c \ ;$$
$$c_{5,6} = c_3 \oplus c_4 \oplus c_{1,2-} \ .$$

Since $\mathcal{R}_{1,2-}$ and $\mathcal{R}_{5,6}$ hold, the prover is able to compute the responses $z_{1,2-}$ and $z_{5,6}$ from $c_{1,2-}$ and $c_{5,6}$.

Case $\eta = 10010\mathtt{xx}$ **(where each** `x` **can be replaced by any bit):** the relations $\mathcal{R}_{1,2-}$ and $\mathcal{R}_4$ hold, but the relations $\mathcal{R}_0$, $\mathcal{R}_3$ and $\mathcal{R}_{5,6}$ may be not verified. The prover chooses (c_0, z_0), (c_3, z_3) and $(c_{5,6}, z_{5,6})$, then simulates the transcripts for these relations. It then receives c from the verifier, which fixes the values of $c_{1,2-}$ and c_4:

$$c_{1,2-} = c_0 \oplus c \ ;$$
$$c_4 = c_3 \oplus c_{5,6} \oplus c_{1,2-} \ .$$

Since $\mathcal{R}_{1,2-}$ and $\mathcal{R}_4$ hold, the prover is able to compute the responses $z_{1,2-}$ and z_4 from $c_{1,2-}$ and c_4.

Case $\eta = 1000\mathtt{xxx}$ **(where each** `x` **can be replaced by any bit):** the relations $\mathcal{R}_{1,2-}$ and $\mathcal{R}_3$ hold, but the relations $\mathcal{R}_0$, $\mathcal{R}_4$ and $\mathcal{R}_{5,6}$ may be not verified. The prover chooses (c_0, z_0), (c_4, z_4) and $(c_{5,6}, z_{5,6})$, then simulates the transcripts for these relations. It then receives c from the verifier, which fixes the values of $c_{1,2-}$ and c_3:

$$c_{1,2-} = c_0 \oplus c \ ;$$
$$c_3 = c_4 \oplus c_{5,6} \oplus c_{1,2-} \ .$$

Since $\mathcal{R}_{1,2-}$ and $\mathcal{R}_3$ hold, the prover is able to compute the responses $z_{1,2-}$ and z_3 from $c_{1,2-}$ and c_3.

Case $\eta = 0\mathtt{xxxxxx}$ **(where each** `x` **can be replaced by any bit):** the relation $\mathcal{R}_0$ holds, but the relations $\mathcal{R}_{1,2-}$, $\mathcal{R}_3$, $\mathcal{R}_4$ and $\mathcal{R}_{5,6}$ may be not verified. The prover chooses $z_{1,2-}$, (c_3, z_3), (c_4, z_4) and $(c_{5,6}, z_{5,6})$, which fixes the value $c_{1,2-}$:

$$c_{1,2-} = c_3 \oplus c_4 \oplus c_{5,6} \ .$$

The prover then simulates the transcripts for these relations and receives c from the verifier, which fixes the values of c_0:

$$c = c_0 \oplus c_{1,2-} \ .$$

Since $\mathcal{R}_0$ holds, the prover is able to compute the responses z_0 from c_0.

On the other hand, if the prover is dishonest (*i.e.,* $\mathcal{R}$ does not hold), then we have the following cases:

Case $\eta = 1001101$**:** the relations $\mathcal{R}_0$, $\mathcal{R}_3$, $\mathcal{R}_4$, and $\mathcal{R}_{5,6}$ are not verified. The prover chooses (c_0, z_0), (c_3, z_3), (c_4, z_4), and $(c_{5,6}, z_{5,6})$ then simulates the transcripts for these relations. It then receives c from the verifier, which fixes the value of $c_{1,2-}$ in order that the equation $c_{1,2-} = c_0 \oplus c$ holds. However, since $c_{1,2-}$, c_3, c_4, and $c_{5,6}$ are fixed, the probability that the equation $c_{1,2-} = c_3 \oplus c_4 \oplus c_{5,6}$ holds is $1/p$ (each challenge is chosen in $\mathbb{Z}_p^*$), which is negligible.

Case $\eta = 100111\mathtt{x}$ (where x can be replaced by any bit): this case is similar to the previous one.
Case $\eta = 101\mathtt{xxxx}$ (where each x can be replaced by any bit): the relations $\mathcal{R}_0$ and $\mathcal{R}_{1,2-}$ are not verified. The prover chooses (c_0, z_0) and $(c_{1,2-}, z_{1,2-})$ then simulates the transcripts for these relations. It then receives c from the verifier, however the probability that the equation $c_{1,2-} = c_0 \oplus c$ holds is $1/p$, which is negligible.
Case $\eta = 11\mathtt{xxxxx}$ (where each x can be replaced by any bit): this case is similar to the previous one.

This example covers all cases in the structure of the binary word k, and can easily be generalized. Note that the size of the transcript of this proof is linear in l. As the prover/verifier needs to check the equations on the challenges that follow a tree structure, the time complexity is quadratic in l, however, we note that the number of exponentiations remains linear in l, making this proof efficient.

E Security Model for k-Times Anonymous Sanitizable Signatures

In this section we propose a fully detail model for k-times Anonymous Sanitizable Signature schemes.

Definition 6 (k-SAN). *A k-times Anonymous Sanitizable Signature scheme (k-SAN) is a tuple of polynomial time algorithms:*

$\mathsf{Setup}(1^\lambda)$: *given a security parameter, returns public parameters* params.
$\mathsf{KeyGen}(1^\lambda, k, n)$: *given a security parameter and two integers k and n, return a pair of key* $(\mathsf{sk}, \mathsf{pk})$.
$\mathsf{SaKeyGen}(1^\lambda)$: *given the public parameters, a security parameter, return a pair of key* $(\mathsf{ssk}, \mathsf{spk})$.
$\mathsf{Delegate}(\mathsf{sk}, \mathsf{spk}, k)$: *given the keys* sk *and* spk *and an integer k, return a delegation* del.
$\mathsf{Sign}(m, \mathsf{ADM}, \mathsf{sk}, \mathsf{spk})$: *given the keys* sk, spk, *a message m and a admissible set* $\mathsf{ADM} \subset [\![n]\!]$, *return a signature σ.*
$\mathsf{Sanitize}(m, \sigma, \mathsf{MOD}, \mathsf{ssk}, \mathsf{pk}, \mathsf{del}, \eta)$: *given the keys* pk, ssk, *a message-signature pair (m, σ), a modification* MOD, *a delegation* del *and a signature index η, return a signature σ'.*
$\mathsf{Verify}(\mathsf{pk}, m, \sigma)$: *given a key* pk, *a message m and a signature σ, returns 0 or 1.*
$\mathsf{Link}(\mathsf{pk}, m, \sigma, m', \sigma')$: *given a key* pk, *two message-signature pair m, σ and m', σ', return an identity* ppk *and a witness w or $\perp$.*
$\mathsf{Trace}(w, \sigma)$: *given a witness w and a signature σ, return 0 or 1.*

The security properties of sanitizable signatures have already been investigated in numerous previous works [1, 4, 11, 3]. We have adapted the existing security properties to the newly introduced model. We also add the properties related to the k-times mechanism: *anonymity*, *traceability* and *non-frameability*. These properties stayes consistant with what has been defined for proxy signatures (Section 3) as both notions share conceptual similarities but diverge in practical usages. Security experiments are provided in Figure 3, with associated oracles detailed in Figure 4 and 5.

Unforgeability. The users cannot generate a valid signature without knowing a secret key which has obtained a delegation. A k-times anonymous sanitizable signature is *unforgeable* when for any PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the SUF experiment is negligible for every $n \in \mathbb{N}$.

Immutability. A sanitizable signature is *immutable* when no adversary is able to sanitize with unauthorized modification. A k-times anonymous sanitizable signature is *immutable* when for any PPT time adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the Immut experiment is negligible for every $n \in \mathbb{N}$. The adversary has access to a delegation and a signature oracle.

Transparency. The verifier cannot decide whether a given signature has been sanitized or not. A k-times anonymous sanitizable signature is *transparent* when for any PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the $\{\mathcal{O}^{\mathsf{tran}}_{\mathsf{Sa/Si}}, \mathcal{O}_{\mathsf{del}}, \mathcal{O}_{\mathsf{Sign}}, \mathcal{O}^{\mathsf{tran}}_{\mathsf{San}}\}$-Sanitize experiment is negligible for every $n \in \mathbb{N}$.

Invisibility. The invisibility property prevents an adversary which is not the signer nor the sanitizer of a signature from determining any information on the modifiable blocks.A k-times anonymous sanitizable signature is *invisible* when for any PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the $\{\mathcal{O}^{\mathsf{Invis}}_{\mathsf{LRADM}}, \mathcal{O}_{\mathsf{del}}, \mathcal{O}_{\mathsf{Sign}}, \mathcal{O}^{\mathsf{Invis}}_{\mathsf{San}}\}$-Sanitize experiment is negligible for every $n \in \mathbb{N}$.

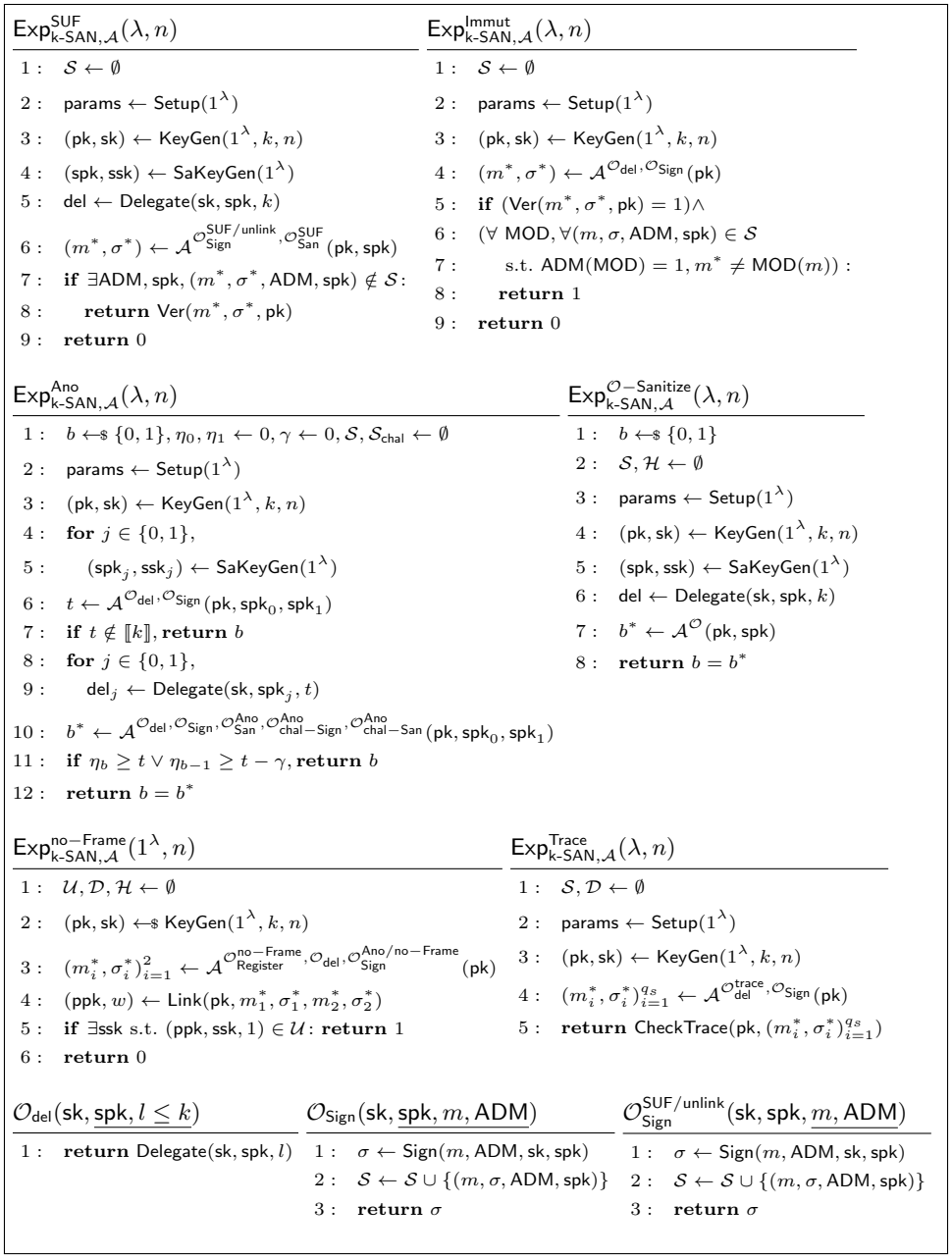

Figure 3: Experiments and Oracles for k-Times Anonymous Sanitizable Signatures. (Additional Oracles are Provided in Figure 4 and 5. Oracles inputs provided by the adversary are underlined, the other are provided by the challenger. Sets $\mathcal{U},\mathcal{D},\mathcal{S},\mathcal{H}$ are global parameters.)

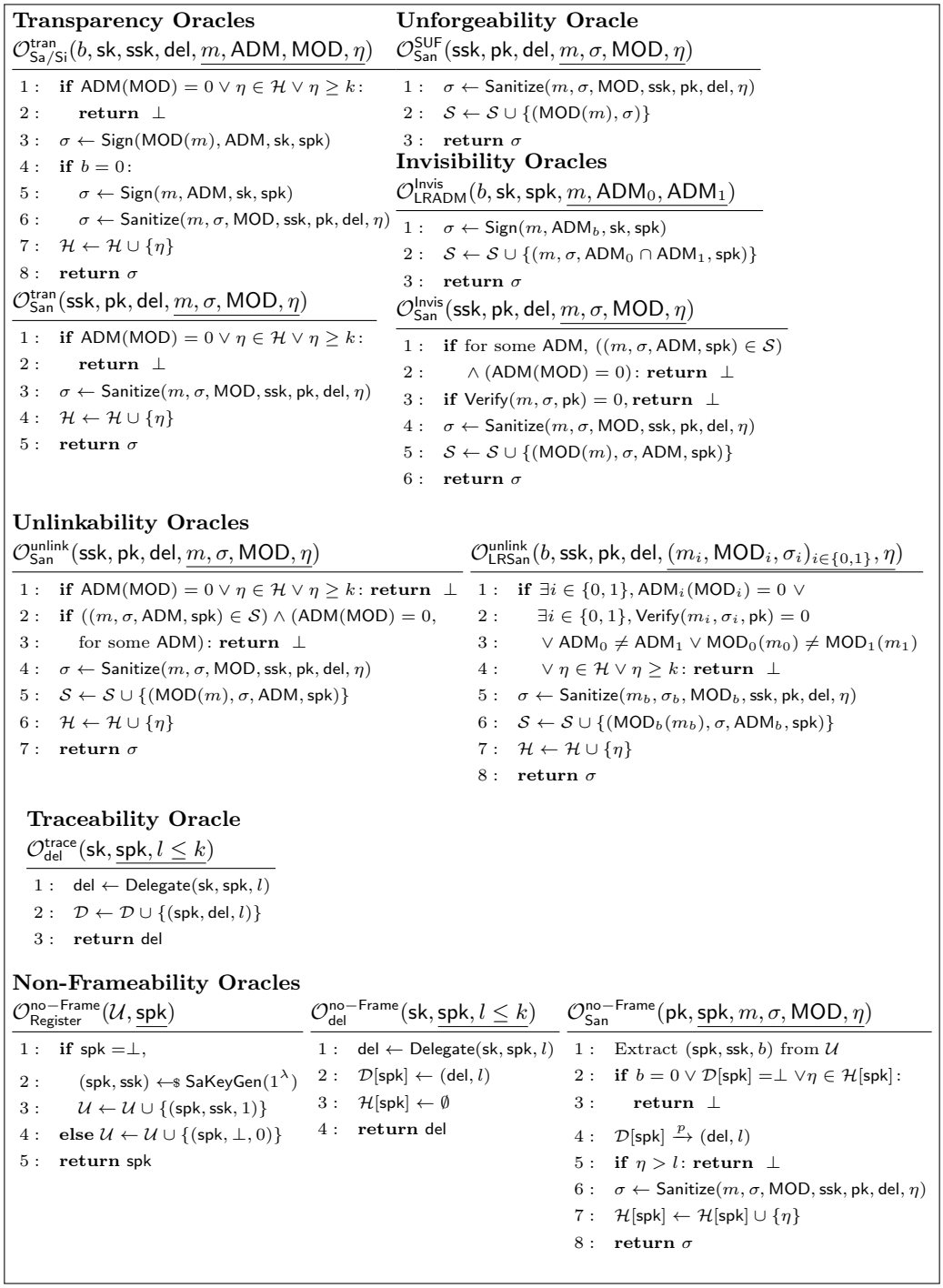

Figure 4: Oracles for k-Times Anonymous Sanitizable Signatures. (Oracles inputs provided by the adversary are underlined, the other are provided by the challenger. Sets $\mathcal{U}, \mathcal{D}, \mathcal{S}, \mathcal{H}$ are global parameters.)

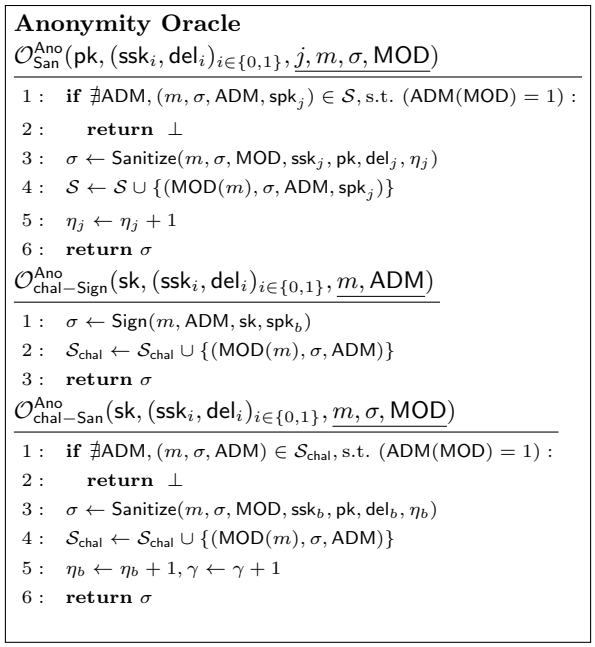

Figure 5: Oracles for k-Times Anonymous Sanitizable Signatures. (Oracles inputs provided by the adversary are underlined, the other are provided by the challenger. Sets $\mathcal{S}$ are global parameters.)

Unlinkability. Considering a fixed sanitizer assigned with two signature, the verifier cannot link a sanitized signature with its original version. A k-times anonymous sanitizable signature is *unlinkable* when for any PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the $\{\mathcal{O}^{\mathsf{unlink}}_{\mathsf{LRSan}}, \mathcal{O}_{\mathsf{del}}, \mathcal{O}^{\mathsf{SUF/unlink}}_{\mathsf{Sign}}, \mathcal{O}^{\mathsf{unlink}}_{\mathsf{San}}\}$-Sanitize experiment is negligible for every $n \in \mathbb{N}$.

F Proof of Theorem 2

Proof. Each of the 8 properties are proven individually to establish the security of the k-SAN. It's important to note that properties of *unforgeability*, *immutability*, *k-traceability*, and *non-frameability* only necessitate collision-resistance in the hash function, whereas *invisibility* requires programming the random oracle in the proof. *Transparency*, *unlinkability*, and *anonymity* proofs, on the other hand, are not dependent on the specific hash function employed. These reductions are applicable for any value of $n \in \mathbb{N}$. Denote by $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_i,\mathsf{G}_{i+1}}(\mathcal{A})$, the probability $|\Pr[\mathsf{G}_i(A) = 1] - Pr[\mathsf{G}_{i+1}(A) = 1]|$.

Correctness. It is verified by investigation.

Unforgeability. Let $\mathsf{Game}^{\mathsf{SUF}}_0$ denote the experiment $\mathsf{Exp}^{\mathsf{SUF}}_{\mathsf{k\text{-}SAN},\mathcal{A}}(1^\lambda)$ instantiated by the k-Times Anonymous Sanitizable Signature of Section E.

Based on the condition $(m^*, \sigma^*, \cdot, \cdot) \notin \mathcal{S}$, we are ensured that $\mathcal{A}$ outputted a signature that was not outputted by the challenger. The SoK proof π_σ signs all elements in the signature, hence, any modification of the signature implies modifying π_σ. We elaborate our reduction based on this fact.

$\mathsf{Game}^{\mathsf{SUF}}_1$: we abort if there is a collision for the responses of the hash function. As argued in the proof of Theorem 1, the adversary's $\mathcal{A}$ advantage differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_0,\mathsf{G}_1}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{col-resist}}_{H}.$$

$\mathsf{Game}_2^{\mathsf{SUF}}$: the SoK π_{MOD} and π_σ, and the NIZK proof $\Pi_{<k}$ are simulated based on their respective simulator for each request to the oracles $\mathcal{O}_{\mathsf{Sign}}$ and $\mathcal{O}_{\mathsf{San}}$. This reduction is achieved straightforwardly for each of the proofs independently leading to

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_1,\mathsf{G}_2}(\mathcal{A}) \leq (q_{\mathsf{Sign}} + q_{\mathsf{San}}) \cdot (2 \cdot \mathsf{Adv}^{\mathsf{Sim}}_{\mathsf{SoK}} + \mathsf{Adv}^{\mathsf{ZK}}_{\mathsf{NIZK}}).$$

$\mathsf{Game}_3^{\mathsf{SUF}}$ (enabling step): instead of sampling $h_4 \leftarrow\!\!\$\ \mathbb{G}_1$, we sample a random value $r_4 \leftarrow\!\!\$\ \mathbb{Z}_p^*$ and define $h_4 = g_1^{r_4}$. This elements keeps the same distribution, no modification of the advantages is needed.

$\mathsf{Game}_4^{\mathsf{SUF}}$: based on the extraction of $\mathsf{ssk}_{\log}{}^*$, we abort if the adversary $\mathcal{A}$ has produced a signature passing all other conditions and such that $\mathsf{spk} = g_1^{\mathsf{ssk}_{\log}{}^*}$. A similar reduction for multiples users has been given in the experiment $\mathsf{Game}_5^{\mathsf{unf}}$ of the proof of Theorem 1. This leads

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_3,\mathsf{G}_4}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}.$$

$\mathsf{Game}_5^{\mathsf{SUF}}$: based on the extraction of $\mathsf{sk}^*_{\log}$, we abort if the adversary $\mathcal{A}$ has produced a signature passing all other conditions such that $\mathsf{pk} = g_1^{\mathsf{sk}^*_{log}}$. Based on similar arguments this leads to:

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_4,\mathsf{G}_5}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}.$$

Analysis. The challenger reject any forged NIZK or SoK without knowledge of the witnesses. Moreover any valid NIZK or SoK based on the secret keys of the signer or a sanitizer make the challenger returns failure. It is still required that $\mathcal{A}$ has outputted a valid SoK proof π_σ otherwise failing the verification. Hence, it must be for a different key, thus, the adversary has obtained a valid delegation for another public sanitization key. *Claim.* The adversary's $\mathcal{A}$ advantage in hybrid $\mathsf{Game}_5^{\mathsf{SUF}}$ is negligible, given the $\mathsf{SPS\text{-}EQ}$ scheme is existentially unforgeable under adaptive chosen-message attacks, *i.e.,*

$$\mathsf{Adv}^{\mathsf{SUF}}_{\mathsf{G}_5}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SPS\text{-}EQ}}.$$

Reduction. This reduction is similar to the one provided to conclude to unforgeability of the k-APS signature in Theorem 1.

Immutability. Let game $\mathsf{Game}_0^{\mathsf{Immut}}$ represent experiment $\mathsf{Exp}^{\mathsf{Immut}}_{\mathsf{k\text{-}SAN},\mathcal{A}}(\lambda)$ instantiated with our k-Times Anonymous Sanitizable Signature.

$\mathsf{Game}_1^{\mathsf{Immut}}$: we abort if there is a collision for the responses of the hash function in the elements that the challenger sees during the experiment. As argued before, the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_0^{\mathsf{Immut}}$ and $\mathsf{Game}_1^{\mathsf{Immut}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_0,\mathsf{G}_1}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{col-resist}}_{H}.$$

$\mathsf{Game}_2^{\mathsf{Immut}}$: we extract the SoK π_{MOD} recovering a witness c such that, for all $i \in [\![n]\!]$, any of the these two statement: $u_i = H(m_i, i, 0)^c \wedge v_i = H(m_i, i, 1)^c$ or $u_i = H(i, 0)^c \wedge v_i = H(i, 1)^c$, should hold true. If the above does not hold the challenger aborts the experiment. The reduction is similar to the previous ones involving the simulation-extractability of a SoK. We conclude that the adversary's $\mathcal{A}$ advantage differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_1,\mathsf{G}_2}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{Sim}}_{\mathsf{SoK}}.$$

Claim. The adversary's $\mathcal{A}$ advantage in hybrid $\mathsf{Game}_2^{\mathsf{Immut}}$ is negligible, given the $\mathsf{SPS\text{-}EQ}$ scheme is existentially unforgeable under adaptive chosen-message attacks, *i.e.,*

$$\mathsf{Adv}^{\mathsf{Immut}}_{\mathsf{G}_2}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SPS\text{-}EQ}}.$$

Reduction. Once again, this reduction is similar to the one provided to conclude to unforgeability of our k-APS signature in Theorem 1.

Transparency. Let $\mathsf{Game}_0^{\mathsf{tran}}$ represent $\mathsf{Exp}_{\mathsf{k\text{-}SAN},\mathcal{A}}^{\{\mathcal{O}_{\mathsf{Sa/Si}}^{\mathsf{tran}},\mathcal{O}_{\mathsf{del}},\mathcal{O}_{\mathsf{Sign}},\mathcal{O}_{\mathsf{San}}^{\mathsf{tran}}\}-\mathsf{Sanitize}}(\lambda)$ instantiated with our k-SAN signature.

$\mathsf{Game}_1^{\mathsf{tran}}$: on a call to $\mathcal{O}_{\mathsf{San}}$, we output new SPS-EQ signatures $\widehat{\sigma}$ instead of randomised ones.
Claim. We claim that hybrids $\mathsf{Game}_0^{\mathsf{tran}}$ and $\mathsf{Game}_1^{\mathsf{tran}}$ are identically distributed, *i.e.,*

$$\mathsf{Adv}_{\mathsf{G}_0,\mathsf{G}_1}^{\mathsf{diff}}(\mathcal{A}) = 0.$$

Reduction. The reduction $\mathcal{R}$ is straightforward and has been provided in a similar context for $\mathsf{Game}_1^{\mathsf{Ano}}$ in the proof of Theorem 1.

$\mathsf{Game}_2^{\mathsf{tran}}$: the same change as in hybrid $\mathsf{Game}_1^{\mathsf{tran}}$ is applied for the signature σ_{MOD}. The reduction is analogous, hence,

$$\mathsf{Adv}_{\mathsf{G}_1,\mathsf{G}_2}^{\mathsf{diff}}(\mathcal{A}) = 0.$$

$\mathsf{Game}_3^{\mathsf{tran}}$: the signature of knowledge π_σ produced during signature or sanitization on calls from $\mathcal{A}$ to oracles $\mathcal{O}_{\mathsf{Sign}}$, $\mathcal{O}_{\mathsf{San}}$ or $\mathcal{O}_{\mathsf{Sa/Si}}$ are now simulated. This reduction is achieved straightforwardly for each of the proofs independently. The adversary's $\mathcal{A}$ advantage differs by,

$$\mathsf{Adv}_{\mathsf{G}_2,\mathsf{G}_3}^{\mathsf{diff}}(\mathcal{A}) \leq (q_{\mathsf{Sign}} + q_{\mathsf{San}} + q_{\mathsf{Sa/Si}}) \cdot \mathsf{Adv}_{\mathsf{SoK}}^{\mathsf{Sim}}.$$

$\mathsf{Game}_4^{\mathsf{tran}}$: instead of computing $\alpha_3 = h_2^x \cdot g_1^{u \cdot \mathsf{ssk}_{\log}}$, we sample it at random $\alpha_3 \leftarrow\!\!\$\ \mathbb{G}_1$, when producing the signature σ.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_3^{\mathsf{tran}}$ and $\mathsf{Game}_4^{\mathsf{tran}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}_{\mathsf{G}_3,\mathsf{G}_4}^{\mathsf{diff}}(\mathcal{A}) \leq q_{\mathsf{Sa/Si}} \cdot \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{DDH}}.$$

Reduction. Based on a DDH challenge $(X = g_1^x, Y = g_1^y, Z)$, we highlight the reduction to prove our claim. Let $h_2 = X$ during the setup, $\widetilde{y} = Y$, while producing the signature σ and generating all $\widetilde{y}_i$ in order to preserve $\widetilde{y} = \prod_{i=1}^{l} \widetilde{y}_i$. Among the l elements $\widetilde{y}_i$, $l-1$ are generated and the remaining element is define as $\widetilde{y}_l = \widetilde{y} \cdot \left(\prod_{i=1}^{l-1} \widetilde{y}_i\right)^{-1}$. Here all elements follow an uniform distribution. . Note that we are simulating the NIZK proof $\Pi_{<k}$ and the SoK π_σ, hence allowing us to be unaware of the witness. Then set $\alpha_3 = Z \cdot g_1^{u \cdot \mathsf{ssk}_{\log}}$. Knowing the discrete logarithm of h_3, we can compute $\alpha_4 = Y^{r_3} \cdot h_4^{v \cdot \mathsf{ssk}_{\log}}$. In order to simulate the signing oracle, we execute it as usual. The values defined in the setup are used, *i.e.,* $h_2 = X$. Note that the game is aborted if the adversary tries to query a signature for the same index a second time, hence no other signature for index i^* could be computed nor leak information to the adversary. When $Z = g_1^{xy}$ we have perfectly simulated $\mathsf{Game}_3^{\mathsf{tran}}$ and when $Z \leftarrow\!\!\$\ \mathbb{G}_1$, we have perfectly simulated $\mathsf{Game}_4^{\mathsf{tran}}$. Hence distinguishing between these two experiments implies distinguishing between the two event of the DDH problem.

$\mathsf{Game}_5^{\mathsf{tran}}$: instead of computing $\alpha_4 = h_3^x \cdot h_4^{v \cdot \mathsf{ssk}_{\log}}$, we sample it at random $\alpha_4 \leftarrow\!\!\$\ \mathbb{G}_1$ when producing the signature σ. The reduction to show indistinguishablility of these two problem is analogous to the previous one, we directly conclude that

$$\mathsf{Adv}_{\mathsf{G}_4,\mathsf{G}_5}^{\mathsf{diff}}(\mathcal{A}) \leq q_{\mathsf{Sa/Si}} \cdot \mathsf{Adv}_{\mathbb{G}_1}^{\mathsf{DDH}}.$$

$\mathsf{Game}_6^{\mathsf{tran}}$: instead of computing $\tau = e(h_4, \alpha_2)^{\mathsf{ssk}_{\log}}$, we sample $Z \leftarrow\!\!\$\ \mathbb{G}_1$ and define $\tau = e(Z, \alpha_2)$ for any sanitized signature.

Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_5^{\mathsf{tran}}$ and $\mathsf{Game}_6^{\mathsf{tran}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_5,\mathsf{G}_6}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}.$$

Reduction. We consider a reduction $\mathcal{R}$ emulating experiments $\mathsf{Game}_5^{\mathsf{tran}}$ and $\mathsf{Game}_6^{\mathsf{tran}}$ against a distinguisher $\mathcal{A}$. This reduction takes as input a DDH challenge $(X, Y, Z) \in \mathbb{G}_1^3$. It defines $h_4 = X$ during the setup and $\mathsf{spk}_{\log} = Y$ during the key generation of the sanitizer. The remaining actions of algorithms Setup and SaKeyGen stay unchanged. The algorithms KeyGen and Delegate are not affected by this change and can still be execute as they should be in $\mathsf{Game}_5^{\mathsf{tran}}$ and $\mathsf{Game}_6^{\mathsf{tran}}$. In contrary, sanitizations require slight changes at the end of the algorithms. First, relying on the previously produced signature σ and the delegation del, we can execute sanitization up to the signature σ_{MOD} and produce x, $\widetilde{y}$, $\widetilde{\mathsf{spk}}$, α_1, u, v and α_2 just like they used to be in description of the scheme in Section 6. Since the previous games, α_3 and α_4 are sampled at uniform in $\mathbb{G}_1$. Our concern is now on τ supposed to be computed as $\tau = e(h_4, \alpha_2)^{\mathsf{ssk}_{\log}} = e(h_4^{\mathsf{ssk}_{\log}}, \alpha_2)$. Based on the DDH challenger define $\tau = e(Z, \alpha_2)$. When $Z = g^{xy}$ we perfectly emulate $\mathsf{Game}_5^{\mathsf{tran}}$, otherwise $\mathsf{Game}_6^{\mathsf{tran}}$. At last the signature of knowledge is emulated which produce valid signatures. It is important to ensure that the threshold of k signature is not overpasses as no tracing could be possible for the produced signature. A condition checks the limit of k signatures in both $\mathcal{O}_{\mathsf{Sa/Si}}$ and $\mathcal{O}^{\mathsf{tran}}_{\mathsf{San}}$. The bit returned by the adversary $\mathcal{A}$ is then transferred as the decision against the DDH challenge. $\mathcal{R}$ has a probability of success similar to the success of the distinguisher $\mathcal{A}$. This conclude to our claim.

$\mathsf{Game}_7^{\mathsf{tran}}$: instead of sampling $Z \leftarrow\!\!\$\ \mathbb{G}_1$ and computing $\tau = e(Z, \alpha_2)$ for all signatures produced by the sanitizer, we sample a new $Z \leftarrow\!\!\$\ \mathbb{G}_2$ for each of the signatures and define $\tau = e(g_1, Z)$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_6^{\mathsf{tran}}$ and $\mathsf{Game}_7^{\mathsf{tran}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_6,\mathsf{G}_7}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{class-hid}}_{\mathbb{G}_2}.$$

Reduction. Consider a reduction $\mathcal{R}$ based on a challenge from the class hiding experiment in $\mathbb{G}_2$ playing against a distinguisher $\mathcal{A}$. The reduction $\mathcal{R}$ receives two elements M and $M^{(b)}$ both in $\mathbb{G}_2^{(q_{\mathsf{San}}+q_{\mathsf{Sa/Si}})}$. We only modify how α_2 and τ are computed, the rest remains similar to both $\mathsf{Game}_6^{\mathsf{tran}}$ and $\mathsf{Game}_7^{\mathsf{tran}}$. We refer to α_2 (*resp.* τ) on the i^{th} call from $\mathcal{A}$ to the oracle $\mathcal{O}_{\mathsf{Sa/Si}}$ or $\mathcal{O}^{\mathsf{tran}}_{\mathsf{San}}$ as $\alpha_{2,i}$ (*resp.* τ_i). Let $\alpha_{2,i} = M_i$ and $\tau_i = e(g_1, M_i^{(b)})$ for all $i \in [\![q_{\mathsf{San}} + q_{\mathsf{Sa/Si}}]\!]$. Based on the value of b, we have $\tau_i = e(g_1, M_i^r)$, for all i and a integer r fixed for all sanitizations, otherwise, $\tau_i = e(g_1, M_i')$, for a random element M_i'. As a result, we emulate perfectly one or other of the games. We can forward the adversary $\mathcal{A}$'s response to the challenger of the class hiding experiment and we win against this game with equal probability. This prove the claim.

$\mathsf{Game}_8^{\mathsf{tran}}$: on execution of the sanitization algorithm on calls to $\mathcal{O}_{\mathsf{Sa/Si}}$, we sample $\widehat{y}_{i,j} \leftarrow\!\!\$\ \mathbb{G}_1$ instead of defining them as a power of the $\widehat{y}_{i,j} = y_{i,j}^r$. We have replaced the randomisation of these elements by newly produced, one which are then signed since $\mathsf{Game}_1^{\mathsf{tran}}$. Hence the signature $\widehat{\sigma}$ remains valid under this modification.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_7^{\mathsf{tran}}$ and $\mathsf{Game}_8^{\mathsf{tran}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_7,\mathsf{G}_8}(\mathcal{A}) \leq 2 \cdot q_{\mathsf{Sa/Si}} \cdot l \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}.$$

Reduction. We consider a sequence of hybrids H_{2i+j} with associated reduction $\mathcal{R}_{i,j}$ for $i \in [\![q_{\mathsf{Sa/Si}} \cdot l]\!]$ and $j \in \{0, 1\}$. All reductions take as input the decision Diffie-Hellman instance (X, Y, X) in $\mathbb{G}_1$ and instead of defining $\widehat{g}_1 = g_1^r$ and $\widehat{y}_{i,j} = y_{i,j}^r$ after having defined $y_{i,j} = g_1^{x_{i,j}}$ in the delegation process, it sets $\widehat{g_1} = X$, $y_{i,j} = Y$ and $\widehat{y}_{i,j} = Z$. To compute $\widehat{y}_{i',j'}$, for $i' > i$,and $j' > j$, its set $\widehat{y}_{i',j'} = X^{x_{i',j'}}$. In order to compute $\mathsf{spk}_{i,j} = {\mathsf{spk}_{\log}}^{x_{i,j}} = y_{i,j}^{\mathsf{ssk}_{\log}}$, we take advantage of the knowledge of $\mathsf{ssk}_{\log}$ and proceed as before for the other parts of the algorithms. $\mathcal{R}_{i,j}$ simulate the rest of the experiment straightforwardly and on obtaining the answer of a distinguisher between experiments $\mathsf{Game}_7^{\mathsf{tran}}$ and $\mathsf{Game}_8^{\mathsf{tran}}$, forward it to the DDH challenger. Clearly, if the tuple $(X = g_1^x, Y = g_1^y, X = g_1^{xy})$ is a decisional Diffie-Hellman tuple, then $\mathcal{R}_{i,j}$ perfectly

simulates H_{2i+j}. On the other hand, if the tuple is not a strong decisional Diffie-Hellman tuple, then $\mathcal{R}_{i,j}$ simulates H_{2i+j+1} perfectly. It is easy to see that H_1 correspond to $\mathsf{Game}_7^{\mathsf{tran}}$ and that $H_{2\cdot q_{\mathsf{Sa/Si}}\cdot l+2}$ correspond to $\mathsf{Game}_8^{\mathsf{tran}}$. The claim follow since we proceeded to a sequence of $2\cdot q_{\mathsf{Sa/Si}}\cdot l$ reductions to the DDH problem.

$\mathsf{Game}_9^{\mathsf{tran}}$: on calls to $\mathcal{O}_{\mathsf{Sa/Si}}$, while executing the sanitization algorithm, we sample all elements $\widehat{\mathsf{spk}}_{k,l} \leftarrow\!\!\$\ \mathbb{G}_1$ instead of defining them as a power of the $\mathsf{spk}_{k,l}$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_8^{\mathsf{tran}}$ and $\mathsf{Game}_9^{\mathsf{tran}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_8,\mathsf{G}_9}(\mathcal{A}) \leq q_{\mathsf{Sa/Si}} \cdot \mathsf{Adv}^{\mathsf{class-hid}}_{\mathbb{G}_1}.$$

Reduction. We consider a sequence of hybrids H_i with associated reduction $\mathcal{R}_i$ for $i \in [\![q_{\mathsf{Sa/Si}}]\!]$. Where in H_i, the elements $\widehat{\mathsf{spk}}_{k,l}$ are sampled as $\widehat{\mathsf{spk}}_{k,l} \leftarrow\!\!\$\ \mathbb{G}_1$ during the first i calls to sanitization of the $q_{\mathsf{Sa/Si}}$ oracle instead of being computed as they used to be.

Consider a reduction $\mathcal{R}_i$ in between each of the hybrids H_i and H_{i+1}. The reduction $\mathcal{R}_i$ simulating either of H_i or H_{i+1} with similar probability for an adversary $\mathcal{A}$ trying to distinguish in between those two hybrids. The reduction $\mathcal{R}_i$ takes as input a challenge $(m, m') \in (\mathbb{G}_1^{2l})^2$ from the challenger of the class-hiding experiment in $\mathbb{G}_1$. The vector m' is either a randomisation of m or a completely new message sampled uniformly at random. $\mathcal{R}_i$ simulate the identical parts of the experiments H_i or H_{i+1} except that it sets $(\mathsf{spk}_{1,0}, \cdots, \mathsf{spk}_{l,1}) = m$ and based on $\mathsf{ssk}_{\log}$ sets $y_{k,l} = \mathsf{spk}_{k,l}^{-\mathsf{ssk}_{\log}}$, for all $k \in [\![l]\!]$ and $l \in \{0,1\}$. As the elements of m are randomly sampled at uniform, the distribution of the $y_{k,l}$ is unchanged. Based on the received message $m' = (m'_1, \cdots, m'_{2l})$, it defines $(\widehat{\mathsf{spk}}_{1,0}, \cdots, \widehat{\mathsf{spk}}_{l,1}) = m'$. Note that as we are only generating new SPS-EQ signatures since the change introduced in experiment $\mathsf{Game}_1^{\mathsf{tran}}$, hence the signature $\widehat{\sigma}$ remains valid under the proposed change. Moreover the elements α_3, α_4 and τ are sampled at random during sanitization in $\mathcal{O}_{\mathsf{Sa/Si}}$, hence they are independent of the values of the $\widehat{\mathsf{spk}}_j$ and we can simulate either H_i or H_{i+1} based on the received challenge m'. Trying to distinguish between both experiments, $\mathcal{A}$ returns a bit b, the latter is forwarded to the challenger of the class-hiding experiment. $\mathcal{R}$ wins with the same probability that $\mathcal{A}$ has to win against this experiment.
Claim. An adversary $\mathcal{A}$ against $\mathsf{Game}_9^{\mathsf{tran}}$ has no advantage, *i.e.,* $\mathsf{Adv}^{\mathsf{tran}}_{\mathsf{G}_9}(\mathcal{A}) = 0$.

In $\mathsf{Game}_9^{\mathsf{tran}}$ we reached the point where algorithmsx $\mathsf{Sign}(m, \mathsf{ADM}, \mathsf{sk}, \mathsf{spk})$ and $\mathsf{Sanitize}(m, \sigma, \mathsf{MOD}, \mathsf{ssk}, \mathsf{pk}, \mathsf{del}, \eta)$ leads to executing the same algorithm. Indeed, in the Sanitize algorithms all elements included in the signature follow the same distribution as in Sign. Hence, an adversary is unable to distinguish a signature $\sigma \leftarrow \mathsf{Sign}(m, \mathsf{ADM}, \mathsf{sk}, \mathsf{spk})$ outputted by the $\mathcal{O}_{\mathsf{Sa/Si}}$ oracle from a signature produced by $\sigma \leftarrow \mathsf{Sanitize}(m, \sigma, \mathsf{MOD}, \mathsf{ssk}, \mathsf{pk}, \mathsf{del}, \eta)$ as the latest correspond to a second execution of $\sigma \leftarrow \mathsf{Sign}(m, \mathsf{ADM}, \mathsf{sk}, \mathsf{spk})$.

Invisibility. Let $\mathsf{Game}_0^{\mathsf{Invis}}$ represent $\mathsf{Exp}^{\{\mathcal{O}^{\mathsf{Invis}}_{\mathsf{LRADM}}, \mathcal{O}_{\mathsf{del}}, \mathcal{O}_{\mathsf{Sign}}, \mathcal{O}^{\mathsf{Invis}}_{\mathsf{San}}\}-\mathsf{Sanitize}}_{\mathsf{k\text{-}SAN},\mathcal{A}}(\lambda)$ instantiated with our k-Times Anonymous Sanitizable Signature in the ROM.

$\mathsf{Game}_1^{\mathsf{Invis}}$: we abort the experiment if a signature given to one of the oracle by $\mathcal{A}$ is valid and has not been produced by one of the oracles.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_0^{\mathsf{Invis}}$ and $\mathsf{Game}_1^{\mathsf{Invis}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_0,\mathsf{G}_1}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{SUF}}_{\mathsf{k\text{-}SAN}}.$$

Reduction. The reduction is straightforward based on a record of the produced signatures kept by the challenger.

$\mathsf{Game}_2^{\mathsf{Invis}}$: we abort the game if there is a collision in the responses of the random oracle.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_1^{\mathsf{Invis}}$ and $\mathsf{Game}_2^{\mathsf{Invis}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_1,\mathsf{G}_2}(\mathcal{A}) \leq \frac{q_H}{2^\lambda}.$$

We can apply a union bound over all q_h queries to the random oracle, and the claim follows.

$\mathsf{Game}_3^{\mathsf{Invis}}$: we rely on the perfect simulatability of the signature of knowledge π_{MOD} to make them independent of the witness a used to compute values u_i and v_i for $i \in [\![n]\!]$ while signing or sanitizing a signature on calls to the oracles $\mathcal{O}_{\mathsf{LRADM}}$ and $\mathcal{O}_{\mathsf{San}}^{\mathsf{Invis}}$. The same reduction has already been provided, hence we directly conclude that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_2^{\mathsf{Invis}}$ and $\mathsf{Game}_3^{\mathsf{Invis}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_2,\mathsf{G}_3}(\mathcal{A}) \leq (q_{\mathsf{LRADM}} + q_{\mathsf{San}}) \cdot \mathsf{Adv}^{\mathsf{Sim}}_{\mathsf{SoK}}.$$

Analysis. The SoK being simulated, π_{MOD} does not allow to recover the witnesses. Thus, the value a could only leak through the ciphertext e.

$\mathsf{Game}_4^{\mathsf{Invis}}$: on calls to $\mathcal{O}_{\mathsf{LRADM}}$, the ciphertext e is sampled at random during the signature. A record A of the random value e, a is kept. On calls to $\mathcal{O}_{\mathsf{San}}^{\mathsf{Invis}}$ for messages signed within $\mathcal{O}_{\mathsf{LRADM}}$, instead of decrypting e the value a is recovered from the record.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_3^{\mathsf{Invis}}$ and $\mathsf{Game}_4^{\mathsf{Invis}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_3,\mathsf{G}_4}(\mathcal{A}) \leq q_{\mathsf{LRADM}} \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{E}}.$$

Reduction. The reduction is direct based on the IND-CCA property of the encryption scheme $\mathcal{E}$. It is achieved through a sequence of hybrids experiment $H_0, \cdots, H_{q_{\mathsf{LRADM}}}$. Experiment H_0 is defined as $\mathsf{Game}_3^{\mathsf{Invis}}$. For all $i \in [\![q_{\mathsf{LRADM}}]\!]$, H_i execute the same action as H_{i-1} except that on the i^{th} call to $\mathcal{O}_{\mathsf{LRADM}}$, it generate the value e by sampling it at random in the encryption space.

To obtain a reduction $\mathcal{R}_i$ in between H_{i-1} and H_i, for all $i \in [\![q_{\mathsf{LRADM}}]\!]$. The reduction $\mathcal{R}_i$ simulate either H_{i-1} or H_i to an adversary $\mathcal{A}$ trying to distinguish between the two games. $\mathcal{R}_i$ obtains the sanitizer public encryption key pk_e from the challenger of the IND-CCA encryption scheme. On the $i-1$ first calls (except for $i = 1$ as there exist no call 0) to $\mathcal{O}_{\mathsf{LRADM}}$, samples the values e at random during signature. On call i to $\mathcal{O}_{\mathsf{LRADM}}$, sends a random value a_0 and a to the IND-CCA challenger and obtain a response c, set $e = c$ and include it in the signature. On obtaining the decision bit from $\mathcal{A}$, $\mathcal{R}_i$ sends the same answer to IND-CCA challenger. As this encrypted value is the only difference between all the H_{i-1} and H_i, it is as hard to distinguish both hybrids as to win against the IND-CCA challenge.

$\mathsf{Game}_5^{\mathsf{Invis}}$: on a call to $\mathcal{O}_{\mathsf{San}}$, we output new SPS-EQ signatures $\widehat{\sigma}$ instead of randomised ones.
Claim. We claim that hybrids $\mathsf{Game}_4^{\mathsf{Invis}}$ and $\mathsf{Game}_5^{\mathsf{Invis}}$ are identically distributed, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_4,\mathsf{G}_5}(\mathcal{A}) = 0.$$

Reduction. The reduction $\mathcal{R}$ is straightforward. Consider "maliciously" generated keys, outputted by $\mathcal{R}$ executing $(\mathsf{pk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}, \mathsf{sk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}) \leftarrow \mathsf{KeyGen}_{\mathsf{SPS\text{-}EQ}}(1^\lambda, l)$. Hence, passing any potential key verification as honestly generated. Answer every signature and sanitization request as usual. All request to $\mathcal{O}_{\mathsf{Sa/Si}}$ are answered by producing a new signature $\widehat{\sigma}$ of the randomised u_i, v_i when sanitization is required instead of randomizing the existing one. From the Signature Adaptation property, ChgRep and Sign outputs identically distributed signatures when executed based on the same key and messages randomised by the same random value.

$\mathsf{Game}_6^{\mathsf{Invis}}$: this game is the same as the previous one, except that the values u_i and v_i computed during one of the signature of the $\mathcal{O}_{\mathsf{LRADM}}$ oracle are generated randomly. The signature of knowledge π_{MOD} is already simulated, hence can be computed even for these random elements.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_5^{\mathsf{Invis}}$ and $\mathsf{Game}_6^{\mathsf{Invis}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_5,\mathsf{G}_6}(\mathcal{A}) \leq q_{\mathsf{LRADM}} \cdot \mathsf{Adv}^{\mathsf{class-hid}}_{\mathsf{SPS\text{-}EQ}}.$$

Reduction. We use a sequence of hybrids to demonstrate this claim. For all $i \in [\![0, q_{\mathsf{LRADM}}]\!]$, consider the sequence of hybrids H_i where for all $j \leq i$, the values u_j and v_j are chosen at random while executing $\mathcal{O}_{\mathsf{LRADM}}$ and the remaining ones are defined as originally prescribed in experiment $\mathsf{Game}_5^{\mathsf{Invis}}$. Now we show

through a reduction that the difference between two consecutive hybrids are negligible: the reduction obtains the setup for the $\mathsf{SPS\text{-}EQ}$ signature from the challenger of the class hiding experiment, based on these setup it generate its own keys. The remaining of the setup and the set definitions are executed as usual. The classe-hiding experiment is executed for messages in $\mathbb{G}_1^{2n}$. We obtain two messages m and m_b. The element m is used to program the random oracle: for $i \in \mathsf{ADM}$ $H(m_i, i, 0) \leftarrow m_i$, $H(m_i, i, 1) \leftarrow m_{2i}$ and for $i \notin \mathsf{ADM}$, $H(i, 0) \leftarrow m_i$, $H(m_i, i, 1) \leftarrow m_{2i}$. The second message m_b is used as $(u_1, \cdots, u_n, v_1, \cdots, v_n)$. The remaining of the algorithms are executed as it is done in both hybrids H_i and H_{i+1}. Once an answer $b \in \{0, 1\}$ is received from $\mathcal{A}$, it is forwarded to the DDH challenger as the reduction's response.
Claim. An adversary $\mathcal{A}$ against $\mathsf{Game}_6^{\mathsf{Invis}}$ has no advantage, *i.e.,* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Invis},\mathsf{G}_6} = 0$.

The adversary's $\mathcal{A}$ response is a decisional bit b that should reflect if the signature outputted by $\mathcal{O}_{\mathsf{LRADM}}$ can be modified based on ADM_0 or ADM_1. Considering that all signatures outputted by $\mathcal{O}_{\mathsf{LRADM}}$ are now independent of ADM_0 and ADM_1. We conclude that the adversary's bit distribution is independent of the uniform distribution of b. As a direct consequence the distribution of $b = b^*$ is uniform within $\{0, 1\}$ and then $\mathsf{Adv}_{\mathsf{G}_6}^{\mathsf{Invis}}(\mathcal{A}) = 0$.

Unlinkability. Let $\mathsf{Game}_0^{\mathsf{unlink}}$ represent $\mathsf{Exp}_{\mathsf{k\text{-}SAN},\mathcal{A}}^{\{\mathcal{O}_{\mathsf{LRSan}}^{\mathsf{unlink}}, \mathcal{O}_{\mathsf{del}}, \mathcal{O}_{\mathsf{Sign}}^{\mathsf{SUF/unlink}}, \mathcal{O}_{\mathsf{San}}^{\mathsf{unlink}}\}-\mathsf{Sanitize}}(\lambda)$ instantiated with our k-Times Anonymous Sanitizable Signature.

Starting from $\mathsf{Game}_0^{\mathsf{unlink}}$ and executing $\mathsf{Sanitize}(m_b, \sigma_b, \mathsf{MOD}_b, \mathsf{ssk}, \mathsf{pk}, \mathsf{del}, \eta)$ on calls to $\mathcal{O}_{\mathsf{LRSan}}$, we introduces independent changes leading to an execution which is independent of the bit b. As these steps only implies negligible changes to the adversary's advantage this is sufficient to show unlinkability. We highlight that the delegation del, the sanitizer's key and the signature index η remains unchanged through this process. As the tuple del returned in the signature only depends on these value, both values will lead to identically distributed elements.

$\mathsf{Game}_1^{\mathsf{unlink}}$: on a call to $\mathcal{O}_{\mathsf{LRSan}}$, we output new $\mathsf{SPS\text{-}EQ}$ signatures σ_{MOD} instead of randomised ones.
Claim. We claim that hybrids $\mathsf{Game}_0^{\mathsf{unlink}}$ and $\mathsf{Game}_1^{\mathsf{Invis}}$ are identically distributed, *i.e.,*

$$\mathsf{Adv}_{\mathsf{G}_0,\mathsf{G}_1}^{\mathsf{diff}}(\mathcal{A}) = 0.$$

Reduction. The reduction $\mathcal{R}$ is straightforward. Consider "maliciously" generated keys, outputted by $\mathcal{R}$ executing $(\mathsf{pk}_{\mathsf{SPS\text{-}EQ}}^{\mathsf{del}}, \mathsf{sk}_{\mathsf{SPS\text{-}EQ}}^{\mathsf{del}}) \leftarrow \mathsf{KeyGen}_{\mathsf{SPS\text{-}EQ}}(1^\lambda, n)$, hence, passing any potential key verification as honestly generated. Answer every signature and sanitization request as usual. All request to $\mathcal{O}_{\mathsf{LRSan}}$ are answered by producing a new signature σ_{MOD} of the randomised u_i, v_i when sanitization is required instead of randomizing the existing one. From the Signature Adaptation property, $\mathsf{ChgRep}_{\mathsf{SPS\text{-}EQ}}$ and $\mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}$ outputs identically distributed signatures when executed based on the same key and messages randomised by the same random value.

Under this change, the signature has been totally randomised during the sanitization, all element being newly produced at sanitization. The NIZK proof $\Pi_{<k}$ is new and only depends on witness η and del. While the mandatory equality $\mathsf{ADM}_0 = \mathsf{ADM}_1$, implies that the immutable base of both message is the same. Hence the $u_{i,j}$ and the $v_{i,j}$ are the same for all $i \in [\![l]\!]$ and $j \in \{0, 1\}$ and still randomised by a value b, stored in a new ciphertext. Finally the last part of the signature is only dependent of the randomised values and does not leak any information on the previously used message. Under these considerations, we conclude to our proof.

Anonymity. Let game $\mathsf{Game}_0^{\mathsf{Ano}}$ represent experiment $\mathsf{Exp}_{\pi,\mathcal{A}}^{\mathsf{Ano}}(\lambda)$ instantiated with our k-Times Anonymous Sanitizable Signature.

In contrary to the unlinkability where the message and the modifications under sanitization change for a fixed sanitizer, in the anonymity experiment the delegation del_b and the sanitizer's identity *i.e.,* spk_b are changing based on the challenge bit b while the message and the modification is fixed. If we shown that there are only negligible steps between an execution of $\mathsf{Sign}(m, \mathsf{ADM}, \mathsf{sk}, \mathsf{spk}_b)$ and $\mathsf{Sanitize}(m, \sigma, \mathsf{MOD}, \mathsf{ssk}_b,$

$\mathsf{pk}, \mathsf{del}_b, \eta)$ and their respective execution that does not rely on b, then, we can conclude that the identity of the sanitizer remains hidden.

$\mathsf{Game}_1^{\mathsf{Ano}}$: during the sanitization executed on call to oracle $\mathcal{O}_{\mathsf{chal-San}}^{\mathsf{Ano}}$ of the experiment $\mathsf{Exp}_{\pi,\mathcal{A}}^{\mathsf{Ano}}(\lambda)$, we output new SPS-EQ signatures $\widehat{\sigma}$ instead of randomising it from $\widehat{\sigma}$ of del_b.
Claim. We claim that hybrids $\mathsf{Game}_0^{\mathsf{Ano}}$ and $\mathsf{Game}_1^{\mathsf{Ano}}$ are identically distributed, *i.e.,*

$$\mathsf{Adv}_{\mathsf{G}_0,\mathsf{G}_1}^{\mathsf{diff}}(\mathcal{A}) = 0.$$

Reduction. The reduction $\mathcal{R}$ is straightforward by an hybrid over the $\eta \leq t \leq k$ signatures randomised by the challange sanitization orcale. Consider the keys outputted by $\mathcal{R}$ executing $(\mathsf{pk}_{\mathsf{SPS-EQ}}^{\mathsf{del}}, \mathsf{sk}_{\mathsf{SPS-EQ}}^{\mathsf{del}}) \leftarrow \mathsf{KeyGen}_{\mathsf{SPS-EQ}}(1^\lambda, 4l+1)$. Answer every signature and sanitization request as usual. The sanitization request $\sigma \leftarrow \mathsf{Sanitize}(m, \sigma, \mathsf{MOD}, \mathsf{ssk}_b, \mathsf{pk}, \mathsf{del}_b, \eta)$ and answeres it by producing a new signature $\widehat{\sigma}$ of $(\widehat{g}_1, \widehat{y}_{1,0}, \cdots, \widehat{\mathsf{spk}}_{l,1})$ instead on the randomised vector $\widehat{\sigma}$ of del_b. From the signature adaptation property, $\mathsf{ChgRep}_{\mathsf{SPS-EQ}}$ and $\mathsf{Sign}_{\mathsf{SPS-EQ}}$ outputs identically distributed signatures when executed based on the same key and messages randomised by the same random value.

$\mathsf{Game}_2^{\mathsf{Ano}}$: instead of producing the proof π_σ on calls to the sanitization oracle, the proof is simulated based on its simulator.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_1^{\mathsf{Ano}}$ and $\mathsf{Game}_2^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}_{\mathsf{G}_1,\mathsf{G}_2}^{\mathsf{diff}}(\mathcal{A}) \leq k \cdot \mathsf{Adv}_{\mathsf{SoK}}^{\mathsf{Sim}}.$$

The reduction is direct based on the zero-knowledge property of the SoK.

$\mathsf{Game}_3^{\mathsf{Ano}}$ (enabling step): instead of directly sampling $h_1, h_2, h_3, h_4 \leftarrow\!\!\$\ \mathbb{G}_1$, we sample a random value $r_1, r_2, r_3, r_4 \leftarrow\!\!\$\ \mathbb{Z}_p^*$ and define $h_i = g_1^{r_i}$, for $i \in [\![4]\!]$.
Claim. We claim that it is a bridging step, meaning that the adversary's $\mathcal{A}$ advantage is not modified under this change:

$$\mathsf{Adv}_{\mathsf{G}_2,\mathsf{G}_3}^{\mathsf{diff}}(\mathcal{A}) = 0.$$

As this elements keeps the same distribution in the group, the adversary has indistinguishable viewing of these experiments.

$\mathsf{Game}_4^{\mathsf{Ano}}$: instead of encrypting $a \cdot b$ into e during the sanitization of σ, the sanitizer sampled the ciphertext e at random and $a \cdot b$ is kept in a record to be used in further sanitization.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_3^{\mathsf{Ano}}$ and $\mathsf{Game}_4^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}_{\mathsf{G}_3,\mathsf{G}_4}^{\mathsf{diff}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathcal{E}}^{\text{IND-CCA}}.$$

This is a direct reduction to the IND-CCA experiment.

$\mathsf{Game}_5^{\mathsf{Ano}}$: instead of encrypting a into e during the signature and the sanitizations of σ, the challenger sample the ciphertext e at random and a is kept in a record to be used in further sanitization. Let q_s^{chal} be the number of calls to the oracle $\mathcal{O}_{\mathsf{chal-San}}^{\mathsf{Ano}}$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_4^{\mathsf{Ano}}$ and $\mathsf{Game}_5^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}_{\mathsf{G}_4,\mathsf{G}_5}^{\mathsf{diff}}(\mathcal{A}) \leq (q_s^{\mathsf{chal}} + k) \cdot \mathsf{Adv}_{\mathcal{E}}^{\text{IND-CCA}}.$$

This is a direct reduction to the IND-CCA experiment.

Step 5 directly leads a signature totaly decorelated of the sanitizer's identity. All elements are either sampled at random or independent of the sanitizer's identity by construction.

$\mathsf{Game}_6^{\mathsf{Ano}}$: instead of computing $\alpha_3 = h_2^x \cdot g_1^{u \cdot \mathsf{ssk}_{\log_b}}$, we sample it at random $\alpha_3 \leftarrow\!\!\$\ \mathbb{G}_1$, when producing the a sanitization σ when requested to the oracle $\mathcal{O}_{\mathsf{chal-San}}^{\mathsf{Ano}}$.

Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_5^{\mathsf{Ano}}$ and $\mathsf{Game}_6^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_5,\mathsf{G}_6}(\mathcal{A}) \leq k \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}.$$

Reduction. The same reduction is provided in proof of Theorem 1. We apply an hybrid argument over it for a constant number of executions.

$\mathsf{Game}_7^{\mathsf{Ano}}$: instead of computing $\alpha_4 = h_3^x \cdot h_4^{v \cdot \mathsf{ssk}_{\log_b}}$, we sample it at random $\alpha_4 \leftarrow\!\!\$\ \mathbb{G}_1$ when producing a signature σ requested to the oracle $\mathcal{O}^{\mathsf{Ano}}_{\mathsf{chal-San}}$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_6^{\mathsf{Ano}}$ and $\mathsf{Game}_7^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_6,\mathsf{G}_7}(\mathcal{A}) \leq k \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}.$$

Reduction. The same reduction has been provided in the proof of Theorem 1. We apply an hybrid argument over it for a constant number of executions.

$\mathsf{Game}_8^{\mathsf{Ano}}$: instead of computing $\tau = e(h_4, \alpha_2)^{\mathsf{ssk}_{\log_0}}$ on sanitization of a signature, the challenger samples a fixed $Z_0 \leftarrow\!\!\$\ \mathbb{G}_1$ at the beginning of the experiment and define $\tau = e(Z_0, \alpha_2)$ for any signature sanitized with the oracle $\mathcal{O}^{\mathsf{Ano}}_{\mathsf{chal-San}}$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_7^{\mathsf{Ano}}$ and $\mathsf{Game}_8^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_7,\mathsf{G}_8}(\mathcal{A}) \leq k \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}.$$

Reduction. We consider a reduction $\mathcal{R}$ emulating experiments $\mathsf{Game}_7^{\mathsf{Ano}}$ and the same experiment with the first sanitization by $\mathcal{O}^{\mathsf{Ano}}_{\mathsf{chal-San}}$ encompassing this modification. $\mathcal{R}$ is against a distinguisher $\mathcal{A}$. This reduction takes as input a DDH challenge $(X, Y, Z) \in \mathbb{G}_1^3$. It defines $h_4 = X$ during the setup and $\mathsf{spk}_{\log_0} = Y$ during the key generation of the sanitizer associated to index 0. The remaining actions of algorithms Setup and $\mathsf{SaKeyGen}$ stay unchanged. The algorithms KeyGen and $\mathsf{Delegate}$ are not affected by this change and can still be execute as they should be in $\mathsf{Game}_7^{\mathsf{Ano}}$ and its modification. In contrary, sanitizations require slight changes when executed with the unknown key $\mathsf{ssk}_{\log_0}$. First, relying on the previously produced signature σ and the delegation del, we can execute the algorithm normally up to the signature σ_{MOD} and produce x, $\widetilde{y}$, $\widetilde{\mathsf{spk}}$, α_1, u, v and α_2. Since the previous games, α_3 and α_4 are sampled at uniform in $\mathbb{G}_1$. The element τ is supposed to be computed as $\tau = e(h_4, \alpha_2)^{\mathsf{ssk}_{\log_0}} = e(h_4^{\mathsf{ssk}_{\log_0}}, \alpha_2)$. Based on the DDH challenger define $\tau = e(Z, \alpha_2)$. When $Z = g^{xy}$ we perfectly emulate $\mathsf{Game}_7^{\mathsf{Ano}}$ otherwise the modified game where the first sanitization is made based of $Z = g^z$. At last the signature of knowledge is emulated which produce valid signatures. It is important to ensure that the threshold of k signature is not overpasses as no tracing could be possible for the produced signature as it is done in $\mathcal{O}^{\mathsf{Ano}}_{\mathsf{San}}$. Indeed, the outputted τ is random and does not allow tracing. The bit returned by the adversary $\mathcal{A}$ is then transferred as the decision against the DDH challenge. $\mathcal{R}$ has a probability of success similar to the success of the distinguisher $\mathcal{A}$. This conclude to our claim. Now based on an hybrid argument we can conclude the reduction between experiments $\mathsf{Game}_7^{\mathsf{Ano}}$ and $\mathsf{Game}_8^{\mathsf{Ano}}$.

$\mathsf{Game}_9^{\mathsf{Ano}}$: instead of computing $\tau = e(h_4, \alpha_2)^{\mathsf{ssk}_{\log_1}}$ on sanitization of a signature, the challenger samples a fixed $Z_1 \leftarrow\!\!\$\ \mathbb{G}_1$ at the beginning of the experiment and define $\tau = e(Z_1, \alpha_2)$ for any signature sanitized with the oracle $\mathcal{O}^{\mathsf{Ano}}_{\mathsf{chal-San}}$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_8^{\mathsf{Ano}}$ and $\mathsf{Game}_9^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_8,\mathsf{G}_9}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}.$$

Reduction. This reduction is the identical to the previous reduction with the bit 0 flipped to 1.

$\mathsf{Game}_{10}^{\mathsf{Ano}}$: instead of sampling $Z_0 \leftarrow\!\!\$\ \mathbb{G}_1$ and computing $\tau = e(Z_0, \alpha_2)$ for all signatures produced by the sanitizer associated to index 0, for each new sanitization a new $Z_0 \leftarrow\!\!\$\ \mathbb{G}_2$ is sampled to define $\tau = e(g_1, Z_0)$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_9^{\mathsf{Ano}}$ and $\mathsf{Game}_{10}^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_9,\mathsf{G}_{10}}(\mathcal{A}) \leq k \cdot \mathsf{Adv}^{\mathsf{class-hid}}_{\mathbb{G}_2}.$$

Reduction. Consider a reduction $\mathcal{R}$ based on a challenge from the class hiding experiment in $\mathbb{G}_2$ playing against a distinguisher $\mathcal{A}$. The reduction $\mathcal{R}$ receives two elements $M, M' \in \mathbb{G}_2^{q_{\mathsf{San}}}$. The only modification needed is on how α_2 and τ are computed, the rest remains similar to both $\mathsf{Game}_9^{\mathsf{Ano}}$ and $\mathsf{Game}_{10}^{\mathsf{Ano}}$. We refer to α_2 (*resp.* τ) on the i^{th} call from $\mathcal{A}$ to the oracle $\mathcal{O}_{\mathsf{San}}^{\mathsf{Ano}}$ as $\alpha_{2,i}$ (*resp.* τ_i). Let $\alpha_{2,i} = M_i$ and $\tau_i = e(g_1, M_i')$ for all $i \in [\![q_{\mathsf{San}}]\!]$. Based on the challenge, we have either $\tau_i = e(g_1, M_i^r)$, for all i and an integer r fixed for all sanitization of index 0, or either $\tau_i = e(g_1, M_i')$, for a new random element M_i' changed for each sanitization of index 0. As a result, we emulate perfectly one or other of the games. We can forward the adversary $\mathcal{A}$'s response to the challenger of the class hiding experiment and we win against this game with equal probability. This prove the claim.

$\mathsf{Game}_{11}^{\mathsf{Ano}}$: instead of sampling $Z_1 \leftarrow\!\!\$\ \mathbb{G}_1$ and computing $\tau = e(Z_1, \alpha_2)$ for all signatures produced by the sanitizer associated to index 1, for each new sanitization a new $Z_1 \leftarrow\!\!\$\ \mathbb{G}_2$ is sampled to define $\tau = e(g_1, Z_1)$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_{10}^{\mathsf{Ano}}$ and $\mathsf{Game}_{11}^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_{10},\mathsf{G}_{11}}(\mathcal{A}) \leq k \cdot \mathsf{Adv}^{\mathsf{class-hid}}_{\mathbb{G}_2}.$$

Reduction. This reduction is the identical to the previous reduction with the bit 0 flipped to 1.

$\mathsf{Game}_{12}^{\mathsf{Ano}}$: while the challenger produces the delegation for the proxy signer, instead of generating $\mathsf{spk}_{i,j} = \mathsf{spk}^{x_{i,j}}$ for all $i \in [\![l]\!]$, $j \in \{0,1\}$, they are sampled uniformly at random within $\mathbb{G}_1$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_{11}^{\mathsf{Ano}}$ and $\mathsf{Game}_{12}^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_{11},\mathsf{G}_{12}}(\mathcal{A}) \leq 2l \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1}.$$

Reduction. The same reduction is provided in proof of Theorem 1.

$\mathsf{Game}_{12}^{\mathsf{Ano}}$: since $\mathsf{Game}_{11}^{\mathsf{Ano}}$ the vector $(g_1, y_{1,0}, \cdots, \mathsf{spk}_{l,1})$ is sampled at random during the delegations of both sanitizers. Instead of randomising one of them to obtain $(\widehat{g_1}, \widehat{y}_{1,0}, \cdots, \widehat{\mathsf{spk}_{l,1}})$ embedded in the sanitized signature σ returned to the adversary, we sample these elements randomly for all the signatures sanitized with the oracle $\mathcal{O}_{\mathsf{chal-San}}^{\mathsf{Ano}}$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_{11}^{\mathsf{Ano}}$ and $\mathsf{Game}_{12}^{\mathsf{Ano}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_{11},\mathsf{G}_{12}}(\mathcal{A}) \leq k \cdot \mathsf{Adv}^{\mathsf{class-hid}}_{\mathbb{G}_1}.$$

Reduction. Let $\mathcal{R}$ be a reduction based on a challenge from the class hiding experiment in $\mathbb{G}_1$ playing against a distinguisher $\mathcal{A}$ against one modification in one of the answer of the oracle. The reduction $\mathcal{R}$ receives two elements $M, M' \in \mathbb{G}_1^{4l+1}$. During the setup it defines $g_1 \leftarrow M_1$ (the first element of vector M), then while executing $\mathsf{Delegate}(\mathsf{sk}, \mathsf{spk}^b, k)$, it signs M in $\widehat{\sigma}$. Based on the challenge M', during the execution of $\mathsf{Sanitize}(m, \sigma, \mathsf{MOD}, \mathsf{ssk}_b, \mathsf{pk}, \mathsf{del}_b, \eta)$, it inputs M' into the $\mathsf{SPS\text{-}EQ}$ signature, thus obtaining $\widehat{\sigma} \leftarrow \mathsf{Sign}_{\mathsf{SPS\text{-}EQ}}(\mathsf{sk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}, M')$ embedded in the signature with M'. The rest of the experiment is executed normally. Based on the value of M', we either emulate $\mathsf{Game}_{11}^{\mathsf{Ano}}$ when M' has been picked in the equivalent class of M, or its modified version, when M' has been picked at random. As a result it is hard to distinguish between both experiments. Based on an hybrid argument we conclude this reduction between $\mathsf{Game}_{11}^{\mathsf{Ano}}$ and $\mathsf{Game}_{12}^{\mathsf{Ano}}$.

In experiment $\mathsf{Game}_{12}^{\mathsf{Ano}}$, the elements that $\mathcal{A}$ sees are completely independent of the value b which is supposed to be guessed by $\mathcal{A}$. Any strategy of guess would then inevitably lead to a null advantage as the

distribution of the adversary's $\mathcal{A}$ outputs are independent of the uniformly distributed value $b \leftarrow\$ \{0,1\}$. This conclude to our proof for this property.

Traceability. Let game $\mathsf{Game}_0^{\mathsf{Trace}}$ represent experiment $\mathsf{Exp}^{\mathsf{Trace}}_{\mathsf{k\text{-}SAN},\mathcal{A}}(\lambda)$ instantiated with our k-Times Anonymous Sanitizable Signature.

$\mathsf{Game}_1^{\mathsf{Trace}}$: we abort if there is a collision for the responses of the hash function in the elements that the challenger sees during the experiment. As argued before, the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_0^{\mathsf{Trace}}$ and $\mathsf{Game}_1^{\mathsf{Trace}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_0,\mathsf{G}_1}(\mathcal{A}) \leq \mathsf{Adv}_H^{\mathsf{col-resist}}.$$

$\mathsf{Game}_2^{\mathsf{Trace}}$: each of the SoK $\pi^*_{\sigma,i}$ contained in the signatures $(\sigma_i^*)_{i=1}^{q_s}$ outputted by the adversary are extracted. The witnesses $\mathsf{ssk}_{\log_i}{}^*, x_i^*, s_i^*, t_i^*, \mathsf{sk}^*_{\log,i}$ are recovered and based on the publicly known elements and the ones inside the signatures, we can check soundness of the proofs. On failure of the extraction or proof of invalid statements, the experiment is aborted. This leads to the following difference based on a straightforward sequence of reductions:

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_1,\mathsf{G}_2}(\mathcal{A}) \leq q_S \cdot \mathsf{Adv}^{\mathsf{ZK}}_{\mathsf{SoK}}.$$

Analysis. Under simulation-extractability of the SoK π_σ, it is ensured that $\mathcal{A}$ has sanitized the signature produced by the signer if $\mathsf{CheckTrace}$ returned 1. Unless it knows the DL of $\mathsf{pk}_{\log}$, $\mathcal{A}$ has correctly computed the elements $\widetilde{y}$, $\widetilde{\mathsf{spk}}$, α_1, α_2, α_3, α_4 and τ. Based on the correctness of the sanitizable signature we are ensured that no delegation where forged, we can always recover $\mathsf{ppk} = (\alpha_3/\alpha_3')^{1/(u-u')}$ and $w = (\alpha_4/\alpha_4')^{1/(v-v')}$ when the same combination of keys spk_i was used twice. This does not implies that $\mathcal{A}$ has not overpasses the limitation $k \leq 2^l$. We now ensure this through another reduction.

$\mathsf{Game}_3^{\mathsf{Trace}}$: witnesses s are extracted from $\Pi_{<k}$. Based on the extracted witness we verify the soundness of the proof. The experiment is aborted and returns 0 if the extracted values s are not consistent with the elements in the signatures. The probability that an adversary has outputted a valid proof for an invalid statement is negligible:

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_2,\mathsf{G}_3}(\mathcal{A}) \leq q_S \cdot \mathsf{Adv}^{\mathsf{sound}}_{\mathsf{NIZK}}.$$

Analysis. The previous reduction guarantees that no key index greater than k can be used to sanitize a signature. Hence, on receiving a delegation del for k, $\mathcal{A}$ has not been able to use more than the k first combination of keys. The last line of attacks that remains is to produced a valid delegation that remains unknown to the signer *i.e.,* forging a signature for its key or based on a forged delegation. We proceeds in three steps, one enabeling step and two steps to conclude:

$\mathsf{Game}_4^{\mathsf{Trace}}$ (enabling step): instead of producing the SoK π_σ on calls to the signing oracle, the signature of knowledge is simulating with its simulator.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_3^{\mathsf{Trace}}$ and $\mathsf{Game}_4^{\mathsf{Trace}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_3,\mathsf{G}_4}(\mathcal{A}) \leq q_S \cdot \mathsf{Adv}^{\mathsf{ZK}}_{\mathsf{SoK}}.$$

The reduction is direct based on the perfect simulatability of the SoK.

$\mathsf{Game}_5^{\mathsf{Trace}}$: we abort if one of the recovered element $\mathsf{sk}^*_{\log}$ extracted from the proofs π_σ verify $\mathsf{sk}^*_{\log} = \mathsf{sk}_{\log}$.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_5^{\mathsf{Trace}}$ is negligible, *i.e.,*

$$\mathsf{Adv}^{\mathsf{Trace}}_{\mathsf{G}_5}(\mathcal{A}) \leq q_s \cdot \mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}.$$

Reduction. The reduction $\mathcal{R}$ is straight forward. $\mathcal{R}$ receive a challenge (g_1, X) for the DL problem, uses g_1 as the base element and set $\mathsf{pk} = X$. It simulates the $\mathcal{O}_{\mathsf{del}}$ as usually after having generate the necessary keys

for the SPS-EQ signatures. On calls from $\mathcal{A}$ to the $\mathcal{O}_{\mathsf{Sign}}$, $\mathcal{R}$ execute it normally and simulate the proof π_σ as prescribed by the latest experiment. On receiving an answer $(m_i^*, \sigma_i^*)_{i=1}^{q_s}$ from $\mathcal{A}$, if the experiment succeeds for the given values, we return a random $\mathsf{sk}_{\log}^*$ to the challenger of the DL problem.

$\mathsf{Game}_6^{\mathsf{Trace}}$: unforgeability of SPS-EQ signature $\widehat{\sigma}$ implies that it is not possible to produce dishonest delegation. We claim that:

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_5,\mathsf{G}_6}(\mathcal{A}) \leq q_S \cdot \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SPS\text{-}EQ}}.$$

Reduction. Consider an adversary $\mathcal{A}$ winning against $\mathsf{Game}_6^{\mathsf{Trace}}$. Let $\mathcal{R}$ be a reduction emulating between the answers of $\mathcal{A}$ and $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SPS\text{-}EQ}}$. We implement the reduction $\mathcal{R}$ straightforwardly. Instead of using $\mathsf{KeyGen}_{\mathsf{SPS\text{-}EQ}}(1^\lambda, 4l+1)$ to generate the keys $(\mathsf{pk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}, \mathsf{sk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}})$, set $\mathsf{pk}^{\mathsf{del}}_{\mathsf{SPS\text{-}EQ}}$ as the public key received from the challenger against $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SPS\text{-}EQ}}$. Moreover, to issue elements $\widehat{\sigma}$ on a call from $\mathcal{A}$ to $\mathcal{O}_{\mathsf{Sign}}$, $\mathcal{R}$ uses the provided signing oracle obtaining $\widehat{\sigma}$. Finally, for a winning adversary outputting a triple $(\mathsf{spk}^*, m^*, \sigma^*)$ for which we have $\mathsf{Ver}(m^*, \sigma^*, \mathsf{pk}) = 1$, it holds that $\mathsf{Verif}_{\mathsf{SPS\text{-}EQ}}(\mathsf{pk}^{\mathsf{MOD}}_{\mathsf{SPS\text{-}EQ}}, (\widehat{g}_1, \widehat{y}_{1,0}, \cdots, \widehat{\mathsf{spk}}_{l,1}), \sigma_{\mathsf{MOD}}) = 1$ from the passing verification. For a winning adversary we can then, transfer one of the message-signature pair $(u_1, v_1, \cdots, u_n, v_n), \sigma_{\mathsf{MOD}}$ to the challenger of the EUF-CMA experiment of the SPS-EQ signature. The response given by the challenger of the EUF-CMA experiment, is outputted by the challenger simulating $\mathsf{Game}_6^{\mathsf{Trace}}$ instead of a winning bit. The claim follows as for any adversary there are only negligible chances to forge a SPS-EQ signature and there must be at least one tuple that the returned message-signature pair is a forge.

With this reduction we prevent from an adversary forging a new delegation. This allows us to conclude the proof.

Non-Frameability. The proof of Non-Frameability is the same as for our k-APS signature. Let $\mathsf{Game}_0^{\mathsf{no-Frame}}$ be the original experiment of Non-Frameability instantiated with our k-SAN scheme of Section 6.

$\mathsf{Game}_1^{\mathsf{no-Frame}}$: we abort if there is a collision in the responses of the random oracle $\mathcal{O}_H$. This prevent from an adversary outputting two values $u^1 = H(m^1, 0, \alpha_2^1) = H(m^2, 0, \alpha_2^2) = u_2$ such that ppk would be set to 0 during the computation in the Trace algorithm.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_0^{\mathsf{no-Frame}}$ and $\mathsf{Game}_1^{\mathsf{no-Frame}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_0,\mathsf{G}_1}(\mathcal{A}) \leq \frac{q_H}{2^\lambda}.$$

We can apply a union bound over all q_h queries to the random oracle, and the claim follows.

$\mathsf{Game}_2^{\mathsf{no-Frame}}$: we abort the experiment if two public keys $\mathsf{spk}_{\log}$ produced by the challenger for the proxies are the same.
Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_1^{\mathsf{no-Frame}}$ and $\mathsf{Game}_2^{\mathsf{no-Frame}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_1,\mathsf{G}_2}(\mathcal{A}) \leq |\mathcal{U}|/|G_1|.$$

Secret keys $\mathsf{ssk}_{\log}$ are sampled uniformly within the group $\mathbb{Z}_p^*$, which is of the order of the group $\mathbb{G}_1$. Each $\mathsf{ssk}_{\log}$ leads to a unique public key $\mathsf{spk}_{\log}$. Hence, the probability to draw to equal keys based on $|\mathcal{U}|$ independent and identically distributed draw is $|\mathcal{U}|/|\mathbb{G}_1|$.

$\mathsf{Game}_3^{\mathsf{no-Frame}}$: the SoK proofs π_σ in the signature returned by $\mathcal{A}$ are extracted. Based on the extracted values $(\mathsf{ssk}_{\log}{}^{\mathsf{Ext},i}, x^{\mathsf{Ext},i}, s^{\mathsf{Ext},i}, t^{\mathsf{Ext},i}, \mathsf{sk}_{\log}^{\mathsf{Ext},i})_{i=1,2}$, we verify the soundness of the proofs by checking if it belong to the language. As argued before we obtain the following difference in the advantages:

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_2,\mathsf{G}_3}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\mathsf{sound}}_{\mathsf{SoK}}.$$

Analysis. From this point, it is ensured that $\mathcal{A}$ holds a witness for the proofs π_σ^1 and π_σ^2 and has computed (α_3^1, α_4^1) and (α_3^2, α_4^2) based on these values or knows the discrete logarithm of the signer's key $\mathsf{sk}_{\log}$.

$\mathsf{Game}_4^{\mathsf{no-Frame}}$: we abort if one of the recovered element $\mathsf{sk}_{\log}^{\mathsf{Ext},i}$ extracted from the proofs π_σ verify $\mathsf{sk}_{\log}^{\mathsf{Ext},i} = \mathsf{sk}_{\log}$ for any of $i \in \{1,2\}$.

Claim. We claim that the adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_3^{\mathsf{no-Frame}}$ and $\mathsf{Game}_4^{\mathsf{no-Frame}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_3,\mathsf{G}_4}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}.$$

Reduction. Consider a reduction $\mathcal{R}$ emulating $\mathsf{Game}_5^{\mathsf{no-Frame}}$ based on a challenge X for the discrete logarithm problem. For each registration request from $\mathcal{A}$, it sets set $\mathsf{spk}_{\log} = X^{s_i}$ for a random $s_i \leftarrow\!\!\$\, \mathbb{Z}_p$. As $\mathcal{A}$ is not provided with a sanitization oracle, their is no need to simulate any action for the sanitizers. On $\mathcal{A}$'s success, ${\mathsf{ssk}_{\log}}^{\mathsf{Ext},i}$ and ${\mathsf{ssk}_{\log}}^{\mathsf{Ext},i}$ where extracted consistently from both proofs. The value $\mathsf{ssk}_{\log} = {\mathsf{ssk}_{\log}}^{1} \cdot s_i^{-1}$ for the correct i is returned as the answer to the DL problem. The witness has the same probability as $\mathcal{A}$ to be right. Hence, $\mathsf{Adv}^{\mathsf{no-Frame},\mathsf{G}_3-\mathsf{G}_4}_{\mathcal{A}} \leq \mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}$.

$\mathsf{Game}_5^{\mathsf{no-Frame}}$: the signature of knowledge π_σ is simulated based on its simulator for each request to the oracles $\mathcal{O}_{\mathsf{Sign}}^{\mathsf{no-Frame}}$. The adversary's $\mathcal{A}$ advantage in hybrids $\mathsf{Game}_4^{\mathsf{no-Frame}}$ and $\mathsf{Game}_5^{\mathsf{no-Frame}}$ only differs by a negligible factor, *i.e.,*

$$\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{G}_4,\mathsf{G}_5}(\mathcal{A}) \leq q_{\mathsf{Sign}} \cdot \mathsf{Adv}^{\mathsf{Sim}}_{\mathsf{SoK}}.$$

Claim. The adversary's $\mathcal{A}$ advantage in hybrid $\mathsf{Game}_5^{\mathsf{no-Frame}}$ is negligible, given that the discrete logarithm problem is hard, *i.e.,*

$$\mathsf{Adv}^{\mathsf{no-Frame}}_{\mathsf{G}_5}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{DL}}_{\mathbb{G}_1}.$$

Reduction. Consider a challenge X for the discrete logarithm problem. During the signer's key generation KeyGen, we define $\mathsf{pk}_{\log} = X$. The others keys for the $\mathsf{SPS\text{-}EQ}$ signature are produced normally, hence delegation request $\mathcal{O}_{\mathsf{del}}$ can executed as usual as they do not depend on the key $\mathsf{pk}_{\log}$ or the associated secret key. The same holds for request to $\mathcal{O}_{\mathsf{Register}}^{\mathsf{no-Frame}}$. It remains to produced coherent answer for the signature request. As the proof π_σ is simulated, this is straightforwardly achieved by the challenger. Once $\mathcal{A}$ returns $(m_i^*, \sigma_i^*)_{i=1,2}$ both $\mathsf{sk}_{\log}^{\mathsf{Ext},1}$ and $\mathsf{sk}_{\log}^{\mathsf{Ext},2}$ are extracted. At the end of both hybrids, if the proof does not holds for a valid statement or holds under the witnesses associated to one of the registered sanitizer, the experiment is aborted. Thus, the proof must holds true for a

The value $\mathsf{sk} = \mathsf{sk}^1 \cdot s_i^{-1}$ for the correct i is returned as the answer to the DL problem. The witness has the same probability as $\mathcal{A}$ to be right.

Based on the two previous reductions, $\mathcal{A}$ has not produced a proof π_σ for any of the keys produced by the challenger. Hence, if the signatures σ_1^* and σ_2^* verifies, the proof π_σ holds true for some keys generated by the adversary. Moreover the elements α_3, α_4 and τ are well formed and tracing an adversary's registered user. The condition $(\mathsf{ppk}, \cdot, \cdot, 1) \in \mathcal{U}$ implies that $\mathcal{A}$ has probability 0 to win this experiment. □