# From Identification Using Rejection Sampling to Signatures via the Fiat-Shamir Transform: Application to the BLISS Signature

Pauline Bert, Adeline Roux-Langlois

# From Identification using Rejection Sampling to Signatures via the Fiat-Shamir Transform: Application to the BLISS Signature

Pauline Bert and Adeline Roux-Langlois

Univ Rennes, CNRS, IRISA
first.last@irisa.fr

**Abstract.** In this paper, we present a reduction from non-lossy/lossy identification scheme using rejection sampling to signature in the Random Oracle Model (ROM). The rejection sampling is used to ensure that the last step in the identification scheme does not leak information about the secret key of the scheme. This last step may fail, and to hide these failures to an adversary we use a Fiat-Shamir transform where we rerun the identification protocol until we get a valid output. We also apply our result for non-lossy identification scheme to the well-known BLISS signature [DDLL13] and compare with the original proof.
**Keywords.** Signature schemes, Identification schemes, Fiat-Shamir transform, Rejection Sampling, Lattices.

## 1 Introduction

The Fiat-Shamir transform [FS86] is a well-studied transform from an identification scheme to a digital signature. In the lattice literature, Fiat-Shamir signatures are probably the most efficent ones [Lyu12,GLP12,DDLL13], compared to hash-and-sign, or even standard model signatures. In this paper, we propose a reduction where almost every Fiat-Shamir transform on lattices can fit into, and we apply our reduction to the BLISS signature [DDLL13].

*From Identification to Signature.* An identification scheme $\mathcal{ID}$ is a three-move protocol Commitment-Challenge-Response. The prover, using its secret key, sends a commitment Cmt to the verifier. The verifier responds a random challenge Ch. The prover finally sends a response Rsp. The verifier, having access to the corresponding public key, accepts or not the complete transcript Cmt||Ch||Rsp. The Fiat-Shamir (FS) transform [FS86] is a way to construct a digital signature scheme in the Random Oracle Model (ROM) from an identification scheme. The signer runs the identification scheme by itself by choosing the challenge via a hash function $\text{Ch} \leftarrow H(\text{Cmt}, m)$. The signature of a message $m$ is $\sigma = (\text{Cmt}, \text{Rsp})$, and to verify such a signature, we recompute the challenge $\text{Ch} \leftarrow H(\text{Cmt}, m)$ and check whether Cmt||Ch||Rsp is a valid transcript or not.

The final publication is available at Springer via
https://doi.org/10.1007/978-3-319-97916-8_19

*Security.* A basic security for an identification scheme is the security against passive impersonation. The adversary, often called the impersonator, has access to the public key of the scheme and to a transcript generation oracle [AABN02]. This oracle, depending on the identification scheme $\mathcal{ID}$ and on a key pair $(pk, sk)$, outputs a random transcript of a honest execution. The goal of the adversary is to impersonate the prover. By interacting with an honest verifier, the adversary wants the verifier to accept at the end of the execution of the protocol. The signature scheme obtained by applying the FS transform is secure against chosen-message attack in the ROM if, and only if, the underlying identification scheme is secure against impersonation under passive attack (and non-trivial meaning that the challenge space is super-polynomial) [AABN02]. This transformation is not tight, it looses a factor at least $q_H$ (number of queries to the random oracle $H$) in the advantage of the impersonator compared to the advantage of the forger.

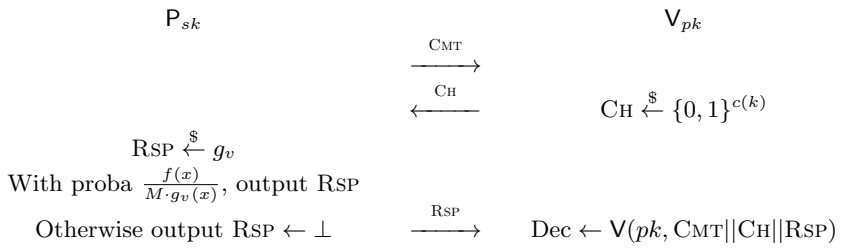
*Lossy/Non-Lossy Identification Scheme.* Motivating by the work of Katz and Wang [KW03], Abdalla et al. in [AFLT12] introduced the idea of lossy identification scheme and give a tight analogous of the reduction of [AABN02], starting from a lossy identification scheme. A lossy identification scheme comes with an additional lossy key generation algorithm which outputs a lossy public key, which is computationally indistinguishable from a honestly generated one. Such scheme has also a property of simulatability, meaning that we can construct a simulated transcript generation oracle, with no access to the secret key of the identification scheme but still outputs transcripts whose distribution is statistically close to those from the original transcript generation oracle. The security of a lossy identification scheme is a notion of impersonation with respect to lossy keys, where the adversary has access to a lossy public key of the scheme and to the simulated variant of the transcript generation oracle. In the lattice literature, we can find some lossy identification schemes: the lattice instantiation at the end of [AFLT12], the underlying identification scheme of NIST submissions TESLA, and Dilithium. In [ABB$^+$17], the authors showed that the TESLA signature is secure in the Quantum Random Oracle Model (QROM) and recently in [KLS17], the proof is generalized to Fiat-Shamir signature starting from lossy identification scheme with an application to Dilithium. There exist also non-lossy lattice-based identification schemes, for example the ones underlying the signatures [Lyu12,DDLL13]. To prove the security of such schemes, a solution is to use the Forking Lemma [PS00,BN06], resulting in a non-tight proof.

*Rejection Sampling.* The use of rejection sampling in lattice constructions is due to Lyubashevsky [Lyu08,Lyu12]. He first describes an abort technique, allowing the prover to abort the protocol instead of returning its response. The idea behind the abort technique is to shorten the response by allowing it to fall in a smaller space/interval. This will happen with small probability, and if it does the prover simply aborts the protocol. When we construct a signature via the Fiat-Shamir transform from such identification scheme, the aborts can be hidden by simply rerun the protocol. The abort/rejection sampling technique is also used to ensure that the response of the prover is independent from its secret

key. Rejection sampling is a method to sample from an arbitrary distribution $f$, given the access to a family of probability distributions $g_v$ indexed by some $v$. A sample $x \xleftarrow{\$} g$ drawn from $g$ is accepted with probability $\frac{f(x)}{M \cdot g_v(x)}$ where $M$ is a constant satisfying $M \cdot g_v(x) \geq f(x)$ for all $x$ drawn from $f$. This procedure succeeds with probability at least $\frac{1}{M}$.

*Identification Scheme using Rejection Sampling.* In an identification scheme using rejection sampling, the probability distribution $f$ corresponds to the target output distribution of the prover responses. Unlike the distribution $f$, the family of probability distribution $g_v$ will depend on the prover secret key; and will be indexed by a random value $v$, being a function of the prover secret key and a uniformly random challenge. For example in [Lyu12], the distribution of

Fig. 1: Identification Scheme using Rejection Sampling




the responses $f$ is a known discrete Gaussian distribution $D^m_\sigma$ of parameter $\sigma$. However, the distribution $g_{\mathbf{v}}$ is a shifted discrete Gaussian distribution $D^m_{\mathbf{v},\sigma}$ with the same parameter $\sigma$ but centered on a vector $\mathbf{v} = \mathbf{Sc}$ depending on the prover secret key $\mathbf{S}$ and on a particular uniformly random challenge $\mathbf{c}$.

**Our contribution.** In this paper, we give a definition of an identification scheme using rejection sampling starting from the definition of rejection sampling from [Lyu12]. This kind of identification scheme has two inherent and quite classical properties:

1. *Correctness Error:* The probability that a honestly generated transcript contains a non-valid response is negligible, here it corresponds to $\left(1 - \frac{1}{M}\right)$.
2. *Simulatability:* There exists a simulated transcript generation oracle, who does not have access to the secret key and is able to output transcripts statistically close to those from the original transcript generation oracle.

The security we consider for such identification scheme is the impersonation against passive attacks where the adversary has access to the real public key of the scheme and also to the simulated transcript oracle.

Our main result is a transformation from an identification scheme using rejection sampling to an existentially unforgeable signature in the ROM. This signature is obtained by applying the Fiat-Shamir transform on an identification scheme using rejection sampling. The only significant modification from existing Fiat-Shamir transform [AFLT12,KLS17] is that we repeat the execution of the identification protocol in the signing algorithm as long as the response of the prover is non-valid. We then discuss whether or not the identification scheme is also lossy. If the identification scheme is lossy, we get a tight proof as in [AFLT12] and if not, we get a non-tight proof loosing a factor of roughly $q_H$ as in [AABN02]. To link the advantage of the impersonator and the advantage against the underlying search problem, we use a propriety of soundness, i.e. if we have access to two valid transcripts on a same commitment, we can extract a solution of a instance of this search problem. Next, by using the Reset Lemma [BP02] we can link the advantage of the impersonator playing the impersonation experiment to the advantage of an adversary playing twice this experiment with different randomness and getting two valid transcripts on a same commitment.

We give an example of this by applying our main result to the well-known BLISS signature [DDLL13] with its underlying non-lossy identification scheme. We choose the BLISS signature because the construction follows exactly our Fiat-Shamir transform (i.e. rerun the identification scheme in the signing algorithm to get a valid response) and we remark that the BLISS paper does not take into account this feature in the proof.

**Overview of our main result.** The idea behind this proof is to use honest transcripts of the identification scheme to answer the signing queries of the forger. If we have a valid transcript $\textsc{Cmt}||\textsc{Ch}||\textsc{Rsp}$, we will set the random oracle $H(\textsc{Cmt}||m) \leftarrow \textsc{Ch}$ to ensure that $\sigma = (\textsc{Cmt}, \textsc{Rsp})$ is a valid signature for the message $m$. The first step of our proof is to limit the number of signing attempts to $l$, where we can take $l$ greater than $M$. Doing this modification implies that the forger might see invalid signatures, when after $l$ tries, the response of the prover is non-valid. This happens for each signing query, so the probability that a signature is non-valid is at most $q_S(1 - \frac{1}{M})^l$ where $\left(1 - \frac{1}{M}\right)$ is the probability that an honestly generated transcript contains a non-valid response.

On sign query $m$, we may overwrite the value $H(\textsc{Cmt}, m)$. Such collisions happen with probability at most $\frac{l(q_S+q_H+1)q_S}{2^\beta}$ where $\beta$ corresponds to the min-entropy of commitments.

Then we apply a series of small changes to get a signing algorithm that no longer needs the secret key $sk$. To do that the major change is to switch from the transcript generation algorithm to its simulated counterpart thanks to the use of rejection sampling. If the statistical distance between the distribution of the transcripts outputted by this two oracles is at most $\varepsilon_{rs}$, the advantage of the forger changes by at most $q_S\varepsilon_{rs}$ due to the rejection sampling technique.

*Non-Lossy Identification Scheme.* If the identification is non-lossy, we can link the advantage of the forger to the advantage of the impersonator. For this step,

we make a guess about which hash query will be used in the forgery. If our guess is correct, we are able to break the underlying impersonation problem, that's why we loose at least a factor $q_H$ in the advantage of the impersonator compared to the advantage of the forger, like [AABN02].

*Lossy Identification Scheme.* If the identification is lossy, we can add another step which involves switching the real public key of the scheme to a lossy one. The advantage of the adversary is modified by at most the advantage in distinguishing a real public key from a lossy one. The reduction is tight because the advantage of the forger is tightly related to the advantage of breaking the underlying search problem, for example the decision-LWE problem in [AFLT12]. Finally, the last step link the advantage of the forger to the advantage of the impersonator with respects to lossy keys as with non-lossy identification scheme.

## 2 Preliminaries

*Notation.* Let $\mathrm{A}(\cdot,\cdot,\cdots)$ be a randomized algorithm, then $x \leftarrow \mathrm{A}(a,b,\cdots;R)$ is the unique output on inputs $a,b,\cdots$ and coins $R$, while $x \xleftarrow{\$} \mathrm{A}(a,b,\cdots)$ means that we first pick a random $R \xleftarrow{\$} \mathrm{Coins}(k)$ and then assigned $x \leftarrow \mathrm{A}(a,b,\cdots;R)$.

### 2.1 Identification Scheme using Rejection Sampling

To hide the secret key of a prover in a identification scheme, Lyubashevsky [Lyu12] proposed to use a rejection technique. Informally, the prover generates a candidate for its response and rejects it with a certain probability to ensure that the distribution of the response is independent from the prover secret key.

**Lemma 1 (Rejection Sampling [Lyu12]).** *Let $V$ be an arbitrary set, $h : V \to \mathbb{R}$ and $f : \mathbb{Z}^m \to \mathbb{R}$ be probability distributions. If $g_v : \mathbb{Z}^m \to \mathbb{R}$ is a family of probability distributions indexed by all $v \in V$ with the property that there exists a constant $M \in \mathbb{R}$ such that $\forall v, Pr\left[M \cdot g_v(x) \geq f(x), x \xleftarrow{\$} f\right] \geq 1 - \varepsilon_{rs}$, then, the output distribution of*

$v \xleftarrow{\$} h$
$x \xleftarrow{\$} g_v$
***return*** $(x,v)$ *with probability* $\min\left(\frac{f(x)}{M \cdot g_v(x)}, 1\right)$

*is within statistical distance $\varepsilon_{rs}/M$ of the output distribution of*

$v \xleftarrow{\$} h$
$x \xleftarrow{\$} f$
***return*** $(x,v)$ *with probability* $1/M$

*Moreover, the probability $p_{out}$ that the first algorithm output something is bounded by $(1-\varepsilon_{rs})/M \leq p_{out} \leq 1/M$.*

*ID.* An identification scheme using rejection sampling (Fig. 1) is a classical one using rejection sampling to ensure that responses follow a probability distribution $f$, by first generating them following a family of probability $g_v$.

**Definition 1.** *An identification scheme using rejection sampling $\mathcal{ID}$ is defined by $\mathcal{ID} = (\mathsf{KeyGen}, \mathsf{P}, \mathsf{V}, c, g_v, f)$ where:*

- $\mathsf{KeyGen}(1^k)$ *is the key generation algorithm, taking the security parameter $k \in \mathbb{N}$ of the scheme and outputting a pair of keys $(pk, sk)$. The secret key $sk$ is given to the prover algorithm $\mathsf{P}$, and the public $pk$ is given to the verifier algorithm $\mathsf{V}$.*
- $\mathsf{P}$ *is the prover algorithm, which takes as input the secret key $sk$ and the current conversation transcript and outputs the next message to be sent to the verifier.*
- $\mathsf{V}$ *is a deterministic algorithm which takes as input the public key $pk$ and the complete transcript conversation* Cmt||Ch||Rsp *and outputs a boolean decision* Dec.
- $c(k)$ *is a function of the security parameter $k$, which corresponds to the length of the challenge.*
- $g_v$ *is a family of probability distributions indexed by $v$, a function of the secret key $sk$, a particular challenge* Ch *and in some case $v$ can also depend on a particular commitment* Cmt *or on the secret used to construct the commitment,*
- $f$ *is the output distribution of the prover responses such that there exists a constant $M \in \mathbb{R}$ verifying $\forall$ Ch $\in \{0,1\}^{c(k)}, \forall\, x, M \cdot g_v(x) \geq f(x)$.*

*Transcript generation oracle.* Like in [AABN02,AFLT12], we associate a transcript generation oracle $\mathrm{Tr}^{\mathcal{ID}}_{pk,sk,k}$ to an identification scheme $\mathcal{ID}$. The transcript generation oracle $\mathrm{Tr}^{\mathcal{ID}}_{pk,sk,k}$ returns a random transcript Cmt||Ch||Rsp of an honest execution of $\mathcal{ID}$ with key pair $(pk, sk)$ and security parameter $k$. In an identification scheme using rejection sampling, the prover may output a response Rsp $= \perp$, in this case the transcript generation oracle will output $\perp||\perp||\perp$.

$\mathrm{Tr}^{\mathcal{ID}}_{pk,sk,k}$ :

Cmt $\xleftarrow{\$} \mathsf{P}(sk)$,
Ch $\xleftarrow{\$} \{0,1\}^{c(k)}$, Rsp $\xleftarrow{\$} g_v$
**return** Cmt||Ch||Rsp with probability $\frac{f(x)}{M \cdot g_v(x)}$, otherwise $\perp||\perp||\perp$.

*Inherent properties.* Thanks to the rejection sampling (Lemma 1) we can simulate the transcript generation oracle $\mathrm{Tr}^{\mathcal{ID}}_{pk,sk,k}$ by an algorithm $\tilde{\mathrm{Tr}}^{\mathcal{ID}}_{pk,k}$ with no access to the secret key $sk$. It proceeds by first generating Cmt $\xleftarrow{\$} \mathsf{P}(sk)$ and Ch $\xleftarrow{\$} \{0,1\}^{c(k)}$ and outputting Cmt||Ch||Rsp with Rsp $\xleftarrow{\$} f$ with probability $\frac{1}{M}$, and otherwise $\perp||\perp||\perp$. This property is called Non-Abort Honest-Verifier Zero-Knowledge (naHVZK) in [KLS17].

**Definition 2 (naHVZK).** *$\mathcal{ID}$ is said to be $\varepsilon$-perfect naHVZK if there exists an algorithm $\tilde{\mathrm{Tr}}^{\mathcal{ID}}_{pk,k}$, given only the public key $pk$ and the security parameter $k$, outputs* Cmt||Ch||Rsp *such that the following conditions hold:*

1. *The distribution of* Cmt||Ch||Rsp $\xleftarrow{\$} \tilde{\mathrm{Tr}}^{\mathcal{ID}}_{pk,k}$ *has statistical distance at most $\varepsilon$ from* Cmt′||Ch′||Rsp′ $\xleftarrow{\$} \mathrm{Tr}^{\mathcal{ID}}_{pk,sk,k}$,
2. *The distribution of* Ch *from* Cmt||Ch||Rsp $\xleftarrow{\$} \tilde{\mathrm{Tr}}^{\mathcal{ID}}_{pk,k}$ *is uniform in the challenge set $\{0,1\}^{c(k)}$.*

Our identification scheme also satisfy the correctness property from [KLS17], with $\varepsilon_c = 1 - 1/M$.

**Definition 3 (Correctness Error).** *An identification scheme $\mathcal{ID}$ has correctness error $\varepsilon$ if for all $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$ the following holds:*

1. *All possible transcripts* Cmt||Ch||Rsp *satisfying* Rsp $\neq \perp$ *are valid,*
2. *The probability that a honestly generated transcript* Cmt||Ch||Rsp *contains* Rsp $= \perp$ *is bounded by $\varepsilon$.*

*Security.* The security of the identification scheme we consider here is a security against passive impersonation where the goal of the adversary is to impersonate the prover without the knowledge of the secret key $sk$. This impersonator is modeled as a probabilistic algorithm $\mathcal{I}$ which is given as input the public key $pk$ of the identification scheme and also has access to the simulation of the transcript oracle $\tilde{\mathrm{Tr}}^{\mathcal{ID}}_{pk,k}$ described above. After looking at these transcripts, the impersonator $\mathcal{I}$ interacts with an honest verifier in the three-move protocol and wants the verifier to accept at the end of this protocol.

$\mathbf{Exp}^{\text{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}}(k)$:

$(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$, $st \| \text{Cmt} \xleftarrow{\$} \mathcal{I}^{\tilde{\mathrm{Tr}}^{\mathcal{ID}}_{pk,k}}(pk)$
Ch $\xleftarrow{\$} \{0,1\}^{c(k)}$, Rsp $\xleftarrow{\$} \mathcal{I}(st, \text{Ch})$, Dec $\leftarrow \mathsf{V}(pk, \text{Cmt}||\text{Ch}||\text{Rsp})$
**return** Dec

The advantage of $\mathcal{I}$ playing the game above is

$$\mathbf{Adv}^{\text{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}}(k) = \Pr\left[\mathbf{Exp}^{\text{imp-pa-sim}}_{ID,\mathcal{I}}(k) = 1\right].$$

An $\mathcal{ID}$ is polynomially-secure against impersonation under passive attack if $\mathbf{Adv}^{\text{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}}(\cdot)$ is negligible for every poly$(k)$-time impersonator $\mathcal{I}$.

### 2.2 Lossy Identification Scheme using Rejection Sampling

A lossy identification scheme using rejection sampling is defined like a classical identification scheme plus an algorithm $\mathsf{LossyKeyGen}(1^k)$ which takes the security parameter $k \in \mathbb{N}$ and outputs a lossy public key $pk$. We will replace a truly

generated public key in $\mathbf{Exp}^{\text{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}}(k)$ by a lossy one in the impersonation experiment with respect to lossy keys $\mathbf{Exp}^{\text{los-imp-pa}}_{\mathcal{ID},\mathcal{I}}(k)$ and we have no need of a secret key in this case. A lossy identification scheme satisfies two properties, a simulatability property like the naHVZK defined above and the following one:

**Definition 4 (Indistinguishability of keys).** *Consider the two experiments* $\boldsymbol{Exp}^{ind\text{-}keys\text{-}real}_{\mathcal{ID},\mathcal{D}}(k)$ *and* $\boldsymbol{Exp}^{ind\text{-}keys\text{-}lossy}_{\mathcal{ID},\mathcal{D}}(k)$ *in which we respectively generate pk via* $\mathsf{KeyGen}(1^k)$ *and via* $\mathsf{LossyKeyGen}(1^k)$*, and provide it as input to the distinguishing algorithm* $\mathcal{D}$*. We say that* $\mathcal{D}$ *can* $(t,\varepsilon)$*-solve the key-indistinguishability problem if* $\mathcal{D}$ *runs in time* $t$ *and*

$$\left| Pr\left[\boldsymbol{Exp}^{ind\text{-}keys\text{-}real}_{\mathcal{ID},\mathcal{D}}(k)=1\right] - Pr\left[\boldsymbol{Exp}^{ind\text{-}keys\text{-}lossy}_{\mathcal{ID},\mathcal{D}}(k)\right]\right| \geq \varepsilon.$$

*We say that* $\mathcal{ID}$ *is* $(t,\varepsilon)$*-key-indistinguishable if no algorithm* $(t,\varepsilon)$*-solve the key-indistinguishability problem.*

*Min-Entropy of commitments.* Let $\mathcal{C}(sk) = \{\mathsf{P}(sk;R) : R \in \text{Coins}(k)\}$ be the set of commitments associated to $sk$, where $\text{Coins}(k)$ is a set of binary string depending on the security parameter $k$. The maximum probability that a commitment takes a particular value is:

$$\alpha(sk) = \max_{\text{CMT}\in\mathcal{C}(sk)} \left\{ \Pr\left[\mathsf{P}(sk;R) = \text{CMT} : R \xleftarrow{\$} \text{Coins}(k)\right]\right\}.$$

Then, the min-entropy function associated to $\mathcal{ID}$ is $\beta(sk) = \min_{sk}\left\{\log_2 \frac{1}{\alpha(sk)}\right\}$, where the minimum is taken over all the $(pk, sk)$ generated by $\mathsf{KeyGen}(1^k)$.

### 2.3 Reset Lemma

Here we recall the Reset Lemma from [BP02] which apply to identification scheme in the same way the Forking Lemma [PS00,BN06] applies to signature scheme.

**Lemma 2 (Reset Lemma [BP02]).** *Let* $\mathsf{P}$ *be a prover in a canonical identification scheme with verifier* $\mathsf{V}$ *and let* $q, v$ *be inputs for the prover and verifier respectively. Let* $\text{acc}(p,v)$ *be the probability that* $\mathsf{V}$ *accepts after its interaction with* $\mathsf{P}$*, i.e. the probability that the following experiment returns 1:*

$R \xleftarrow{\$} \text{Coins}(k)$, $st||\text{CMT} \leftarrow \mathsf{P}(p;R)$
$\text{CH} \xleftarrow{\$} \{0,1\}^{c(k)}$, $\text{RSP} \xleftarrow{\$} \mathsf{P}(st,\text{CH})$, $\text{Dec} \leftarrow \mathsf{V}(v,\text{CMT}||\text{CH}||\text{RSP})$
***return*** Dec

*Let* $\text{res}(q,v)$ *be the probability that the following reset experiment outputs 1:*

$R \xleftarrow{\$} \text{Coins}(k)$, $st||\text{CMT} \leftarrow \mathsf{P}(p;R)$
$\text{CH}_1 \xleftarrow{\$} \{0,1\}^{c(k)}$, $\text{RSP}_1 \xleftarrow{\$} \mathsf{P}(st,\text{CH}_1)$, $\text{Dec}_1 \leftarrow \mathsf{V}(v,\text{CMT}||\text{CH}_1||\text{RSP}_1)$
$\text{CH}_2 \xleftarrow{\$} \{0,1\}^{c(k)}$, $\text{RSP}_2 \xleftarrow{\$} \mathsf{P}(st,\text{CH}_2)$, $\text{Dec}_2 \leftarrow \mathsf{V}(v,\text{CMT}||\text{CH}_2||\text{RSP}_2)$
***return*** $\text{Dec}_1 \wedge \text{Dec}_2 \wedge \text{CH}_1 \neq \text{CH}_2$

*Then* $\text{acc}(q,v) \leq \frac{1}{2^{c(k)}} + \sqrt{\text{res}(q,v)}$.

### 2.4 Lattice Background

*Lattices.* An $m$-dimensional full-rank lattice $\Lambda$ is a discrete additive subgroup of $\mathbb{R}^m$. A lattice is the set of all integer combinations of some linearly independent basis vectors, $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_m\} \in \mathbb{R}^{m \times m}$, $\Lambda(\mathbf{B}) = \{\sum_{i=1}^{m} z_i \mathbf{b}_i : z_i \in \mathbb{Z}\}$.

*Gaussian distribution.* The continuous Gaussian distribution of center $\mathbf{c} \in \mathbb{R}^m$ and width parameter $\sigma$ is defined as $\rho^m_{\mathbf{c},\sigma}(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2})$. The discrete Gaussian distribution over the lattice $\mathbb{Z}^m$ is defined as $D^m_{\mathbf{c},\sigma} = \frac{\rho^m_{\mathbf{c},\sigma}(\mathbf{x})}{\rho^m_\sigma(\mathbb{Z}^m)}$ where $\rho^m_\sigma(\mathbb{Z}^m) = \sum_{\mathbf{x} \in \mathbb{Z}^m} \rho^m_\sigma(\mathbf{x})$.

**Lemma 3 ([Lyu12]).** *For any $\eta > 1$, $Pr_{\mathbf{z} \leftarrow D^m_\sigma}[\|\mathbf{z}\| > \eta\sigma\sqrt{m}] < \eta^m \exp^{\frac{m}{2}(1-\eta^2)}$.*

*SIS.* A classical hard problem in lattice based literature is the Short Integer Solution (SIS) problem, introduced by Ajtai [Ajt96] where he also gives a reduction from worst-case lattice problems to the average-case SIS problem.

**Definition 5 ($\mathbf{SIS}_{q,n,m,\beta}$).** *Given an uniformly random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{x} = 0 \mod q$ and $0 < \|\mathbf{x}\| \leq \beta$.*

## 3 Signature Scheme using Rejection Sampling

In this part, we will describe formally the Fiat-Shamir transform we use to construct a digital signature from our definition of an identification scheme using rejection sampling. Will we show that applying this Fiat-Shamir to such identification scheme gives us a secure digital signature in the ROM.

### 3.1 Fiat-Shamir Transform

**Definition 6.** *Let $\mathcal{ID} = (\mathsf{KeyGen}, \mathsf{P}, \mathsf{V}, c, g_v, f)$ be an identification scheme using rejection sampling, and $H : \{0,1\}^* \to \{0,1\}^{c(k)}$ be a hash function modeled as a random oracle, then we can construct a signature $\mathcal{DS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$. The signature has the same key generation algorithm as the identification scheme, and the output length of the hash function equals the challenge length. The signing and verifying algorithms are defined as follows:*

$\mathsf{Sign}(sk, m)$:
  ***while*** $\textsc{Rsp} = \perp$ ***do***
    $\textsc{Cmt} \leftarrow \mathsf{P}(sk)$
    $\textsc{Ch} \leftarrow H(\textsc{Cmt}, m)$
    $\textsc{Rsp} \xleftarrow{\$} g_v$
    ***return*** $\sigma = (\textsc{Cmt}, \textsc{Rsp})$ *with probability* $\frac{f(x)}{M \cdot g_v(x)}$, *otherwise*
    $\textsc{Rsp} \leftarrow \perp$

$\mathsf{Verify}(pk, m, \sigma)$:
  *parse* $\sigma$ *as* $(\textsc{Cmt}, \textsc{Rsp})$
  $\textsc{Ch} \leftarrow H(\textsc{Cmt}, m)$
  ***return*** $\mathsf{V}(pk, \textsc{Cmt}||\textsc{Ch}||\textsc{Rsp})$

### 3.2 General Result from Non-Lossy Identification Scheme

Our first and main result gives us a reduction from non-lossy identification scheme using rejection sampling to an existential unforgeable secure signature in the ROM by supposing that the underlying ID scheme satisfies the two properties naHVZK and correctness error defined in definitions 2 and 3.

**Theorem 1.** *Let $\mathcal{ID} = (\mathsf{KeyGen}, \mathsf{P}, \mathsf{V}, c, g_v, f)$ be an identification scheme using rejection sampling whose commitment space has min-entropy $\beta(k)$, let $H : \{0,1\}^* \to \{0,1\}^{c(k)}$ be a hash function modeled as a random oracle, and let $\mathcal{DS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be the associated signature as in Def. 6. If $\mathcal{ID}$ is $\varepsilon_{rs}$-perfect naHVZK, has correctness error $\varepsilon_c$ and is secure against impersonation under passive attacks then $\mathcal{DS}$ is existentially unforgeable secure against adaptive chosen-message attack in the random oracle model such that:*

$$\boldsymbol{Adv}^{\textit{uf-cma}}_{\mathcal{DS},\mathcal{F}}(k) \leq (q_H + 1)\boldsymbol{Adv}^{\textit{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}}(k) + q_S\varepsilon_{rs} + \frac{l(q_S + q_H + 1)q_S}{2^\beta} + q_S\varepsilon_c^l.$$

*Furthermore, if $\mathcal{I}$ runs in at most time $t'$, then $\mathcal{F}$ runs in times $t = t' - O(q_S t_{\text{Sign}})$, where $t_{\text{Sign}}$ designed the average time of the signing algorithm.*

*Proof.* This proof uses code-based game-playing à la [AABN02,AFLT12], by constructing a sequence of experiments $\mathbf{Exp}_0, \cdots, \mathbf{Exp}_7$ starting with the experiment catching the existentially unforgeability of the signature scheme. We defined $\delta_i$ as the event that experiment $\mathbf{Exp}_i$ returns 1, i.e. that the adversary $\mathcal{F}$ outputs a valid forgery. We will assume that before outputting a forgery $(m^*, \sigma^* = (\textsc{Cmt}^*, \textsc{Rsp}^*))$, $\mathcal{F}$ already queried the corresponding hash query on $\textsc{Cmt}^*, m^*$, which increase the number of hash query by one.

$\mathbf{Exp}_0$. In this first experiment, the challenger generates the pair of keys $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$, sets the hash counter $hc$ and the sign counter $sc$ to zero, and also initializes the set of queried messages $\mathcal{M}$ to empty in **Initialize** and returns the public key $pk$ to $\mathcal{F}$. On hash query $\textsc{Cmt}, m$, the challenger checks if $H(\textsc{Cmt}, m)$ has already been set. If $H(\textsc{Cmt}, m) = \perp$, the counter $hc$ is incremented by one, the challenger chooses a random challenge $\textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$ and sets $H(\textsc{Cmt}, m) \leftarrow \textsc{Ch}$. The challenger finally outputs $H(\textsc{Cmt}, m)$. On sign query $m$, the counter $sc$ is incremented by one, the queried message $m$ is added to the set $\mathcal{M}$ and the challenger computes the signature $\sigma = (\textsc{Cmt}, \textsc{Ch})$ as in the signing algorithm. During the signing phase, the challenger also checks if $H(\textsc{Cmt}, m) = \perp$, if so it performs the same steps as for an hash query on $\textsc{Cmt}, m$. Finally, when $\mathcal{F}$ outputs a forgery $(m^*, \sigma^*)$, the challenger runs $\text{Dec} \leftarrow \mathsf{V}(pk, \textsc{Cmt}^* || \textsc{Ch}^* || \textsc{Rsp}^*)$ and returns $\text{Dec} \wedge (m^* \notin \mathcal{M})$. By definition, $\Pr[\delta_0] = \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},\mathcal{F}}(k)$.

$\mathbf{Exp}_1$. Let bad be a boolean variable initialize to false. In $\mathbf{Exp}_1$, we limit the number of signing attempts to $l$, with $l \geq M$ in practice. We also set bad to true if after $l$ signing attempts, we do not output a valid signature. For each signature, the probability that $\textsc{Rsp} = \perp$ after at most $l$ attempts is equal to $\varepsilon_c^l$

Fig. 2: $\mathbf{Exp}_0$ and $\boxed{\mathbf{Exp}_1}$

**Initialize**:
1: $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$
2: $hc \leftarrow 0$, $sc \leftarrow 0$, $\mathcal{M} \leftarrow \{\,\}$
3: **return** $pk$

**On Hash-query** $\textsc{Cmt}, m$:
1: **if** $H(\textsc{Cmt}, m) = \perp$ **then**
2: $\quad hc \leftarrow hc + 1$
3: $\quad \textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$
4: $\quad H(\textsc{Cmt}, m) \leftarrow \textsc{Ch}$
5: **return** $H(\textsc{Cmt}, m)$

**Finalize**$(m^*, \sigma^*)$:
1: Parse $\sigma^*$ as $\textsc{Cmt}^*, \textsc{Rsp}^*$
2: $\textsc{Ch}^* \leftarrow H(\textsc{Cmt}^*, m^*)$
3: $\text{Dec} \leftarrow \mathsf{V}(pk, \textsc{Cmt}^* || \textsc{Ch}^* || \textsc{Rsp}^*)$
4: **return** $\text{Dec} \wedge (m^* \notin \mathcal{M})$

**On Sign-query** $m$:
1: $sc \leftarrow sc + 1$, $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
2: $\boxed{ctr \leftarrow 0}$
3: **while** $\textsc{Rsp} = \perp$ $\boxed{\text{and } ctr \leq l}$ **do**
4: $\quad \boxed{ctr \leftarrow ctr + 1}$
5: $\quad \textsc{Cmt} \leftarrow \mathsf{P}(sk)$
6: $\quad$ **if** $H(\textsc{Cmt}, m) = \perp$ **then**
7: $\quad\quad hc \leftarrow hc + 1$
8: $\quad\quad \textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$
9: $\quad\quad H(\textsc{Cmt}, m) \leftarrow \textsc{Ch}$
10: $\quad \textsc{Ch} \leftarrow H(\textsc{Cmt}, m)$
11: $\quad \textsc{Rsp} \xleftarrow{\$} g_v$
12: $\quad$ **return** $\sigma = (\textsc{Cmt}, \textsc{Rsp})$ with probability $\frac{f(x)}{M \cdot g_v(x)}$, otherwise $\textsc{Rsp} \leftarrow \perp$
13: **if** $\boxed{\textsc{Rsp} = \perp}$ **then**
14: $\quad \boxed{\text{bad} \leftarrow \text{true}}$
15: $\quad$ **return** $\boxed{\sigma = (\perp, \perp)}$

where we recall that $\varepsilon_c$ corresponds to the error correctness of the scheme. We get $|\Pr[\delta_1] - \Pr[\delta_0]| \leq \sum_{i=1}^{q_S} \Pr[\text{bad} = \text{true in the } i\text{-th sign query}] = q_S \varepsilon_c^l$ (Fig. 2).

$\mathbf{Exp}_2$. In this experiment, the challenger no longer sets bad to true due to a non-valid signature. This modification does not change the output of the experiment, we have $\Pr[\delta_2] = \Pr[\delta_1]$.

$\mathbf{Exp}_3$. In this experiment, on sign query $m$, the challenger sets bad to true if the value $H(\textsc{Cmt}, m)$ has already been defined. If bad is set, the challenger also chooses a new value $\textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$ and overwrites the old value $H(\textsc{Cmt}, m) \leftarrow \textsc{Ch}$ (Fig. 3). To compute the probability of bad sets to true, we assume that all the hash queries have been already ask at the beginning of the experiment. The probability that during the $i$-th signing query bad sets to true is $(l(i-1) + q_H + 1)/2^\beta$ where we recall that $l$ is the number of signing attempts we introduce in $\mathbf{Exp}_1$. So we get $|\Pr[\delta_3] - \Pr[\delta_2]| \leq \sum_{i=1}^{q_S} \Pr[\text{bad} = \text{true in the } i\text{-th sign query}] \leq \frac{l(q_S + q_H + 1)q_S}{2^\beta}$.

$\mathbf{Exp}_4$. In this experiment, the challenger no longer sets bad to true due to an overwriting of the value $H(\textsc{Cmt}, m)$. This modification does not change the output of the experiment, we have $\Pr[\delta_4] = \Pr[\delta_3]$.

$\mathbf{Exp}_5$. In the previous experiment, to answer signing query, the challenger generates a new uniform challenge $\textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$ like in the transcript generation

Fig. 3: $\mathbf{Exp}_2$ and $\boxed{\mathbf{Exp}_3}$

**Initialize**:
1: $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$
2: $hc \leftarrow 0$, $sc \leftarrow 0$, $\mathcal{M} \leftarrow \{\,\}$
3: **return** $pk$

**On Hash-query** $\textsc{Cmt}, m$:
1: **if** $H(\textsc{Cmt}, m) = \perp$ **then**
2: $\quad hc \leftarrow hc + 1$
3: $\quad \textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$
4: $\quad H(\textsc{Cmt}, m) \leftarrow \textsc{Ch}$
5: **return** $H(\textsc{Cmt}, m)$

**Finalize**$(m^*, \sigma^*)$:
1: Parse $\sigma^*$ as $\textsc{Cmt}^*, \textsc{Rsp}^*$
2: $\textsc{Ch}^* \leftarrow H(\textsc{Cmt}^*, m^*)$
3: $\text{Dec} \leftarrow \mathsf{V}(pk, \textsc{Cmt}^* || \textsc{Ch}^* || \textsc{Rsp}^*)$
4: **return** $\text{Dec} \wedge (m^* \notin \mathcal{M})$

**On Sign-query** $m$:
1: $sc \leftarrow sc + 1$, $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
2: $ctr \leftarrow 0$
3: **while** $\textsc{Rsp} = \perp$ and $ctr \leq l$ **do**
4: $\quad ctr \leftarrow ctr + 1$
5: $\quad \textsc{Cmt} \leftarrow \mathsf{P}(sk)$
6: $\quad$ **if** $H(\textsc{Cmt}, m) = \perp$ **then**
7: $\quad\quad hc \leftarrow hc + 1$
8: $\quad\quad \textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$
9: $\quad\quad H(\textsc{Cmt}, m) \leftarrow \textsc{Ch}$
10: $\quad$ **else**
11: $\quad\quad \boxed{\text{bad} \leftarrow \text{true}}$
12: $\quad\quad \boxed{\textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}}$
13: $\quad\quad \boxed{H(\textsc{Cmt}, m) \leftarrow \textsc{Ch}}$
14: $\quad \textsc{Ch} \leftarrow H(\textsc{Cmt}, m)$
15: $\quad \textsc{Rsp} \xleftarrow{\$} g_v$
16: $\quad$ **return** $\sigma = (\textsc{Cmt}, \textsc{Rsp})$ with probability $\frac{f(x)}{M \cdot g_v(x)}$, otherwise $\textsc{Rsp} \leftarrow \perp$

oracle $\text{Tr}^{\mathcal{ID}}_{pk,sk,k}$ (Fig. 4). So in this experiment, we change the simulation of the signing algorithm such that the value $\textsc{Cmt}||\textsc{Ch}||\textsc{Rsp}$ are generated thanks to $\text{Tr}^{\mathcal{ID}}_{pk,sk,k}$. This change does not affect the output of the game, $\Pr[\delta_5] = \Pr[\delta_4]$.

$\mathbf{Exp}_6$. We can go further, by generating all the transcripts needed to answer signing queries at the beginning of the experiment which does not change the output of the experiment, $\Pr[\delta_6] = \Pr[\delta_5]$.

$\mathbf{Exp}_7$. In the last experiment, we replace the transcript generation oracle $\text{Tr}^{\mathcal{ID}}_{pk,sk,k}$ by its simulated counterpart $\tilde{\text{Tr}}^{\mathcal{ID}}_{pk,k}$. Since the statistical distance between the distribution outputted by $\text{Tr}^{\mathcal{ID}}_{pk,sk,k}$ and by $\tilde{\text{Tr}}^{\mathcal{ID}}_{pk,k}$ is at most $\varepsilon_{rs}$, we have $|\Pr[\delta_7] - \Pr[\delta_6]| \leq q_S \varepsilon_{rs}$ (Fig. 5).

The last step of the proof is to show that we can use the forger from $\mathbf{Exp}_7$ to construct an impersonator $\mathcal{I}$ playing the experiment $\mathbf{Exp}^{\text{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}}(k)$. The impersonator receives a public key $pk$ from an honest verifier $\mathsf{V}$, chooses an index $fp$ uniformly at random from $[1, q_H + 1]$ and sends $pk$ to $\mathcal{F}$. It also generates the transcripts $(\textsc{Cmt}_i, \textsc{Ch}_i, \textsc{Rsp}_i)$ for $i = 1, \cdots, q_S$ thanks to $\tilde{\text{Tr}}^{\mathcal{ID}}_{pk,k}$. On the $j$-th hash query $(\textsc{Cmt}_j, m_j)$ of $\mathcal{F}$, it first checks if $j \neq fp$. If so, $\mathcal{I}$ works like in $\mathbf{Exp}_7$ and if not, $\mathcal{I}$ returns $\textsc{Cmt}_{fp}$ as the first interaction with the verifier $\mathsf{V}$. In that case, the verifier outputs a challenge $\textsc{Ch}^*$, the impersonator sets $H(\textsc{Cmt}_{fp}, m_{fp}) \leftarrow \textsc{Ch}^*$ and returns this value to $\mathcal{F}$. On the $i$-th signing query, $\mathcal{I}$ returns $\sigma = (\textsc{Cmt}_i, \textsc{Rsp}_i)$ as in $\mathbf{Exp}_7$. Eventually, the forger $\mathcal{F}$ outputs a forgery $(m^*, \sigma^* = (\textsc{Cmt}^*, \textsc{Rsp}^*))$

Fig. 4: $\mathbf{Exp}_5$

**Initialize**:
1: $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$
2: $hc \leftarrow 0$, $sc \leftarrow 0$, $\mathcal{M} \leftarrow \{\,\}$
3: **return** $pk$

**On Sign-query** $m$:
1: $sc \leftarrow sc + 1$, $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
2: $ctr \leftarrow 0$
3: **while** $\textsc{Rsp} = \perp$ and $ctr \leq l$ **do**
4: $\quad ctr \leftarrow ctr + 1$
5: $\quad hc \leftarrow hc + 1$
6: $\quad \textsc{Cmt}||\textsc{Ch}||\textsc{Rsp} \xleftarrow{\$} \mathrm{Tr}^{\mathcal{ID}}_{pk,sk,k}$
7: $\quad$ **return** $\sigma = (\textsc{Cmt}, \textsc{Rsp})$ if $\textsc{Rsp} \neq \perp$

**On Hash-query** $\textsc{Cmt}, m$:
1: **if** $H(\textsc{Cmt}, m) = \perp$ **then**
2: $\quad hc \leftarrow hc + 1$
3: $\quad \textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$
4: $\quad H(\textsc{Cmt}, m) \leftarrow \textsc{Ch}$
5: **return** $H(\textsc{Cmt}, m)$

**Finalize**$(m^*, \sigma^*)$:
1: Parse $\sigma^*$ as $\textsc{Cmt}^*, \textsc{Rsp}^*$
2: $\textsc{Ch}^* \leftarrow H(\textsc{Cmt}^*, m^*)$
3: $\mathrm{Dec} \leftarrow \mathsf{V}(pk, \textsc{Cmt}^*||\textsc{Ch}^*||\textsc{Rsp}^*)$
4: **return** $\mathrm{Dec} \wedge (m^* \notin \mathcal{M})$

Fig. 5: $\mathbf{Exp}_6$ and $\boxed{\mathbf{Exp}_7}$

**Initialize**:
1: $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$
2: $hc \leftarrow 0$, $sc \leftarrow 0$, $\mathcal{M} \leftarrow \{\,\}$
3: **for** $i = 1, \cdots, q_S$ **do**
4: $\quad ctr \leftarrow 0$
5: $\quad$ **while** $\textsc{Rsp}_i = \perp$ and $ctr \leq l$ **do**
6: $\quad\quad (\textsc{Cmt}_i, \textsc{Ch}_i, \textsc{Rsp}_i) \xleftarrow{\$} \begin{cases} \mathrm{Tr}^{\mathcal{ID}}_{pk,sk,k} \\ \boxed{\tilde{\mathrm{Tr}}^{\mathcal{ID}}_{pk,k}} \end{cases}$
7: **return** $pk$

**On Sign-query** $m$:
1: $sc \leftarrow sc + 1$, $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
2: **return** $\sigma = (\textsc{Cmt}_{sc}, \textsc{Rsp}_{sc})$

**On Hash-query** $\textsc{Cmt}, m$:
1: **if** $H(\textsc{Cmt}, m) = \perp$ **then**
2: $\quad hc \leftarrow hc + 1$
3: $\quad \textsc{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$
4: $\quad H(\textsc{Cmt}, m) \leftarrow \textsc{Ch}$
5: **return** $H(\textsc{Cmt}, m)$

**Finalize**$(m^*, \sigma^*)$:
1: Parse $\sigma^*$ as $\textsc{Cmt}^*, \textsc{Rsp}^*$
2: $\textsc{Ch}^* \leftarrow H(\textsc{Cmt}^*, m^*)$
3: $\mathrm{Dec} \leftarrow \mathsf{V}(pk, \textsc{Cmt}^*||\textsc{Ch}^*||\textsc{Rsp}^*)$
4: **return** $\mathrm{Dec} \wedge (m^* \notin \mathcal{M})$

and $\mathcal{I}$ outputs $\text{RSP}^*$ to the verifier as the last step of the identification protocol. If $(\text{CMT}^*, m^*) = (\text{CMT}_{fp}, m_{fp})$, then the probability that $\mathbf{Exp}_7$ outputs one is the that $\mathbf{Exp}^{\text{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}}(k)$ outputs one. We get $\Pr[\delta_7] \leq (q_H+1)\mathbf{Adv}^{\text{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}}(k)$. Putting everything together and we get the expected result.

□

### 3.3 Result from Lossy Identification Scheme

To prove the security of the underlying identification scheme we can either use a decision hard problem, or a search hard problem. By using a decision hard problem, we can replace the public key $pk$ by a "lossy" version in the impersonation experiment like in [AFLT12]. Then if the identification is lossy (see definition in part 2.2), we can add another step to the proof of Theorem 1 and we get the following result:

**Theorem 2.** *Let $\mathcal{ID} = (\mathsf{KeyGen}, \mathsf{LossyKeyGen}, \mathsf{P}, \mathsf{V}, c, g_v, f)$ be a lossy identification scheme using rejection sampling whose commitment space has min-entropy $\beta(k)$, let $H : \{0,1\}^* \to \{0,1\}^{c(k)}$ be a hash function modeled as a random oracle, and let $\mathcal{DS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be the associated signature as in Def. 6. If $\mathcal{ID}$ is $\varepsilon_{rs}$-perfect naHVZK, has correctness error $\varepsilon_c$, is $(t', \varepsilon_k)$-key-indistinguishable, and is secure against impersonation under passive attacks with respect to lossy keys then $\mathcal{DS}$ is existentially unforgeable secure against adaptive chosen-message attack in the random oracle model such that:*

$$\boldsymbol{Adv}^{\textit{uf-cma}}_{\mathcal{DS},\mathcal{F}}(k) \leq (q_H+1)\boldsymbol{Adv}^{\textit{los-imp-pa}}_{\mathcal{ID},\mathcal{I}}(k) + \varepsilon_k + q_S\varepsilon_{rs} + \frac{l(q_S+q_H+1)q_S}{2^\beta} + q_S\varepsilon_c^l.$$

*Furthermore, $\mathcal{F}$ runs in times $t = t' - O(q_S t_{\text{Sign}})$.*

*Proof.* The beginning of the proof is the same as for Thm. 1, we use the same sequence of experiments $\mathbf{Exp}_0, \cdots, \mathbf{Exp}_7$ plus another experiment $\mathbf{Exp}_8$.

$\mathbf{Exp}_8$. In this experiment, the challenger generates the public key thanks to the $\mathsf{LossyKeyGen}(1^k)$ instead of $\mathsf{KeyGen}(1^k)$. Distinguishing these two experiments corresponds to the key-indistinguishability property of $\mathcal{ID}$, we get $|\Pr[\delta_8] - \Pr[\delta_7]| \leq \varepsilon_k$. To conclude the proof, as in the proof of Thm. 1, we show that we can use the forger from $\mathbf{Exp}_8$ to construct an impersonator $\mathcal{I}$ playing the experiment $\mathbf{Exp}^{\text{los-imp-pa}}_{\mathcal{ID},\mathcal{I}}(k)$.

Fig. 6: $\mathbf{Exp}_7$ and $\boxed{\mathbf{Exp}_8}$

**Initialize**:

1: $\begin{cases} (pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k) \\ \boxed{pk \xleftarrow{\$} \mathsf{LossyKeyGen}(1^k)} \end{cases}$
2: $hc \leftarrow 0$, $sc \leftarrow 0$, $\mathcal{M} \leftarrow \{\,\}$
3: **for** $i = 1, \cdots, q_S$ **do**
4: $\quad ctr \leftarrow 0$
5: $\quad$ **while** $\text{Rsp}_i = \perp$ and $ctr \leq l$ **do**
6: $\quad\quad (\text{Cmt}_i, \text{Ch}_i, \text{Rsp}_i) \xleftarrow{\$} \tilde{\text{Tr}}^{\mathcal{ID}}_{pk,k}$
7: **return** $pk$

**On Sign-query** $m$:

1: $sc \leftarrow sc + 1$, $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
2: **return** $\sigma = (\text{Cmt}_{sc}, \text{Rsp}_{sc})$

**On Hash-query** $\text{Cmt}, m$:

1: **if** $H(\text{Cmt}, m) = \perp$ **then**
2: $\quad hc \leftarrow hc + 1$
3: $\quad \text{Ch} \xleftarrow{\$} \{0,1\}^{c(k)}$
4: $\quad H(\text{Cmt}, m) \leftarrow \text{Ch}$
5: **return** $H(\text{Cmt}, m)$

**Finalize**$(m^*, \sigma^*)$:

1: Parse $\sigma^*$ as $\text{Cmt}^*, \text{Rsp}^*$
2: $\text{Ch}^* \leftarrow H(\text{Cmt}^*, m^*)$
3: $\text{Dec} \leftarrow \mathsf{V}(pk, \text{Cmt}^* || \text{Ch}^* || \text{Rsp}^*)$
4: **return** $\text{Dec} \wedge (m^* \notin \mathcal{M})$

□

## 4 Application to the BLISS Signature

In this part, we apply our result from Theorem 1 to the non-lossy identification scheme of the BLISS signature. We describe this identification scheme, its properties and we compare our result to the original BLISS proof.

### 4.1 Description of the BLISS Identification and Signature Schemes

The BLISS signature was introduced by Ducas et al. [DDLL13] follows directly from the work of [Lyu12], where the authors improved the rejection sampling by taking a bimodal Gaussian instead of a shifted Gaussian. The secret key $sk$ is a short matrix $\mathbf{S} \in \mathbb{Z}_{2q}^{m \times n}$ and the public key is a matrix $\mathbf{A} \in \mathbb{Z}_{2q}^{n \times m}$ such that $\mathbf{AS} = q\mathbf{I}_n \mod 2q$. The challenge set is the set of binary vectors of length $n$ and weight $\kappa$, $\mathcal{C} = \{\mathbf{v} : \mathbf{v} \in \{0,1\}^n, \|\mathbf{v}\|_1 \leq \kappa\}$. The hash function $H : \{0,1\}^* \to \mathcal{C}$ outputs uniform elements in the challenge set $\mathcal{C}$. The underlying identification scheme of the BLISS signature works as follows:

| $\mathsf{P}_{sk=(\mathbf{A},\mathbf{S})}$ | | $\mathsf{V}_{pk=\mathbf{A}}$ |
|---|---|---|
| $\mathbf{y} \xleftarrow{\$} D_\sigma^m$, $\mathbf{u} \leftarrow \mathbf{Ay} \mod 2q$ | $\xrightarrow{\mathbf{u}}$ | |
| $b \xleftarrow{\$} \{0,1\}$, $\mathbf{z} \leftarrow (-1)^b\mathbf{Sc} + \mathbf{y}$ | $\xleftarrow{\mathbf{c}}$ | $\mathbf{c} \xleftarrow{\$} \mathcal{C}$ |
| **Output z** with probability $1 \Big/ \left(M \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{Sc} \rangle}{\sigma^2}\right)\right)$ otherwise **output z** $\leftarrow \perp$ | $\xrightarrow{\mathbf{z}}$ | **Output** 1 iff $\|\mathbf{z}\|_\infty < q/4$, $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$, and $\mathbf{Az} + q\mathbf{c} = \mathbf{u} \mod 2q$. |

### 4.2 Properties of the Identification Scheme

We give an high level overview of the properties achieve by the BLISS identification scheme (for more details, see [DDLL13]).

*Perfect Rejection Sampling.* The target output distribution of the prover responses is $f(\mathbf{z}) = D_\sigma^m$. The responses in the above identification scheme follow the family of distribution $g_{\mathbf{Sc}}(\mathbf{z}) = \frac{1}{2}D^m_{\mathbf{Sc},\sigma}(\mathbf{z}) + \frac{1}{2}D^m_{-\mathbf{Sc},\sigma}(\mathbf{z}) = f(\mathbf{z}) \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{Sc}, \mathbf{z}\rangle}{\sigma^2}\right)$. To ensure that $M \cdot g_{\mathbf{Sc}}(\mathbf{z}) \geq f(\mathbf{z})$ for all $z$, the authors of [DDLL13] choose $M = \exp\left(\frac{1}{2\alpha^2}\right)$ where $\alpha$ is such that $\sigma \geq \alpha\|\mathbf{Sc}\|$.

*naHVZK.* We describe the transcript generation oracle $\mathrm{Tr}^{\mathcal{ID}}_{pk,sk,k}$, and the simulated one $\tilde{\mathrm{Tr}}^{\mathcal{ID}}_{sk,k}$. Thanks to Lemma 1 the outputs of these to algorithm are statistically closed.

$\mathrm{Tr}^{\mathcal{ID}}_{pk,sk,k}$ :
$\mathbf{y} \leftarrow D_\sigma^m$, $\mathbf{u} \leftarrow \mathbf{Ay} \mod 2q$
$\mathbf{c} \xleftarrow{\$} \mathcal{C}$, $b \xleftarrow{\$} \{0,1\}$, $\mathbf{z} \leftarrow (-1)^b\mathbf{Sc} + \mathbf{y}$
**return** $(\mathbf{u}, \mathbf{c}, \mathbf{z})$ with probability $1 \Big/ \left(M \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{Sc}\rangle}{\sigma^2}\right)\right)$, otherwise **return** $(\perp, \perp, \perp)$.

$\tilde{\mathrm{Tr}}^{\mathcal{ID}}_{pk,k}$ :
$\mathbf{z} \leftarrow D_\sigma^m$
$\mathbf{c} \xleftarrow{\$} \mathcal{C}$
$\mathbf{u} \leftarrow \mathbf{Az} + q\mathbf{c} \mod 2q$
**return** $(\mathbf{u}, \mathbf{c}, \mathbf{z})$ with probability $\frac{1}{M}$, otherwise **return** $(\perp, \perp, \perp)$

*Correctness Error.* The prover uses a rejection sampling technique to ensure that its response $\mathbf{z}$ is independent from its secret key, and by Lemma 1, we know that the prover outputs $\mathbf{z} \neq \perp$ with probability at least $1 - 1/M$. If the prover outputs a valid response, we have $\mathbf{Az} - q\mathbf{c} = \mathbf{A}((-1)^b\mathbf{Sc} + \mathbf{y}) + q\mathbf{c} = \mathbf{u}$ and by Lemma 1, $\mathbf{z}$ is distributed according to $D_\sigma^m$ and hence has norm $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ with high probability.

*Impersonation/Soundness.* We want here to have an idea of the advantage of an impersonator $\mathcal{I}$ playing the experiment $\mathbf{Exp}^{\text{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}}(k)$, where the impersonator $\mathcal{I}$ has access to the real public key of the scheme $pk = \mathbf{A}$ and to the simulated transcript generation oracle $\tilde{\mathrm{Tr}}^{\mathcal{ID}}_{pk,k}$ described above. If we apply the Reset Lemma 2, the advantage of $\mathcal{I}$ corresponds to the probability acc and we get two valid transcripts on a same commitment $(\mathbf{u}, \mathbf{c}, \mathbf{z})$ and $(\mathbf{u}, \mathbf{c}', \mathbf{z}')$ with probability frk. With these two transcripts, we get $\mathbf{Az} + q\mathbf{c} = \mathbf{Az}' + q\mathbf{c}' \mod 2q$ which gives $\mathbf{A}(\mathbf{z} - \mathbf{z}') = 0 \mod q$. Then $\mathbf{z} - \mathbf{z}'$ is a solution of norm at most $\leq 2\eta\sigma\sqrt{m}$ of a SIS instance of parameters $n$, $m$, $q$, and $\beta = 2\eta\sigma\sqrt{m}$. If $\mathbf{Adv}_{\text{SIS}}$ denotes the advantage against such SIS instance, we finally get $\mathbf{Adv}^{\text{imp-pa-sim}}_{\mathcal{ID},\mathcal{I}} \leq \frac{1}{|\mathcal{C}|} + \sqrt{\mathbf{Adv}_{\text{SIS}}}$.

*Min-Entropy of commitments.* To get an idea of the min-entropy, we consider the probability that a commitment takes a particular value,

$$\Pr\left[\mathbf{Ay} = \mathbf{u};\ \mathbf{y} \leftarrow D_\sigma^m\right] \leq 2^{-n}.$$

*Conclusion.* Applying our result from Thm. 1 to the BLISS signature, we get

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},\mathcal{F}} \leq (q_H+1)\left(\frac{1}{|\mathcal{C}|}+\sqrt{\mathbf{Adv}_{\text{SIS}}}\right)+l(q_S+q_H+1)q_S 2^{-n}+q_S(1-1/M)^l.$$

### 4.3 Original BLISS proof

The original BLISS proof is summarized in [DDLL13, Thm 3.3] but proved through two lemmas. The first lemma [DDLL13, Lem. 3.4], states that the advantage in distinguishing the actual signing algorithm from an hybrid one constructed using the rejection sampling is at most $q_S(q_S+q_H)2^{-n}$. And the second lemma [DDLL13, Lem. 3.5] is a direct application of the General Forking Lemma of [BN06], which says that the advantage against the SIS problem with parameters $n$, $m$, $q$, and $\beta = 2\eta\sigma\sqrt{m}$ is at least $\mathbf{Adv}_{\text{SIS}} \geq acc \cdot \left(\frac{acc}{q_S+q_H} - \frac{1}{|\mathcal{C}|}\right)$ where $acc = \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},\mathcal{F}} - \frac{1}{|\mathcal{C}|}$. Thanks to the General Forking Lemma, we can rewrite this equation as $acc \leq \frac{q_H+q_S}{|\mathcal{C}|} + \sqrt{(q_H+q_S)\mathbf{Adv}_{\text{SIS}}}$. Putting the two lemmas together and we get

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},\mathcal{F}} \leq \frac{q_S+q_H+1}{|\mathcal{C}|} + \sqrt{(q_H+q_S)\mathbf{Adv}_{\text{SIS}}} + q_S(q_S+q_H)2^{-n}.$$

*Comparison.* In the original BLISS paper, the proof does the identification scheme only once during the signing algorithm instead of repeating the identification scheme until the response is non-valid. So we need to add a factor $l$ to the term $q_S(q_S+q_H)2^{-n}$ and add the term $q_S(1-1/M)^l$ in the previous equation to better fit the Fiat-Shamir transform from Def. 6. For concrete parameters, we would have $q_H \gg q_S$, for example in the NIST submission $q_H = 2^{128}$ and $q_S = 2^{64}$, and our reduction looses a factor roughly $\sqrt{q_H}$ (we looses at least $q_H$ in the reduction but gains roughly $\sqrt{q_H}$ by applying the Reset Lemma instead of the Forking Lemma).

**Acknowledgments.** Pauline Bert is funded by the Direction Générale de l'Armement (Pôle de Recherche CYBER). This work has received a French government support granted to the CominLabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01.

## References

AABN02. Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security. In *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 418–433. Springer, 2002.

ABB+17. Erdem Alkim, Nina Bindel, Johannes A. Buchmann, Özgür Dagdelen, Edward Eaton, Gus Gutoski, Juliane Krämer, and Filip Pawlega. Revisiting TESLA in the quantum random oracle model. In *PQCrypto*, volume 10346 of *Lecture Notes in Computer Science*, pages 143–162. Springer, 2017.

AFLT12. Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 572–590. Springer, 2012.

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108. ACM, 1996.

BN06. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *ACM Conference on Computer and Communications Security*, pages 390–399. ACM, 2006.

BP02. Mihir Bellare and Adriana Palacio. GQ and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 162–177. Springer, 2002.

DDLL13. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *CRYPTO (1)*, volume 8042 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2013.

FS86. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

GLP12. Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, volume 7428 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2012.

KLS17. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. *IACR Cryptology ePrint Archive*, 2017:916, 2017.

KW03. Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In *ACM Conference on Computer and Communications Security*, pages 155–164. ACM, 2003.

Lyu08. Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2008.

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, 2012.

PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.