A Family of FDH Signature Schemes Based on the Quadratic Residuosity Assumption

Giuseppe Ateniese[1], Katharina Fech[2], and Bernardo Magri[2(✉)]

[1] Stevens Institute of Technology, Hoboken, USA
[2] Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
{katharina.fech,bernardo.magri}@fau.de

Abstract. Signature schemes are arguably the most crucial cryptographic primitive, and devising *tight* security proofs for signature schemes is an important endeavour, as it immediately impacts the feasibility of deployment in real world applications. Hash-then-sign signature schemes in the Random Oracle Model, such as RSA-FDH, and Rabin-Williams variants are among the fastest schemes to date, but that unfortunately do not enjoy tight security proofs based on the one-wayness of their trapdoor function; instead, all known tight proofs rely on variants of the (non-standard) Φ-Hiding assumption. As our main contribution, we introduce a family of hash-then-sign signature schemes, inspired by a lossy trapdoor function from Freeman et al. (JoC' 13), that is tightly secure under the Quadratic Residuosity assumption. Our first scheme has the property of having *unique* signatures, while the second scheme is deterministic with an extremely fast signature verification, requiring at most 3 modular multiplications.

Keywords: Digital signatures · Full domain hash
Tight security proof · Quadratic residuosity · Lossy trapdoor function

1 Introduction

After the beginning of public-key cryptography [13] many new computational problems were devised, and along with them came cryptographic schemes based on the difficulty of solving those problems. At first, asymptotic security analysis was enough to claim the robustness of a given scheme, but it was realized later that a more precise analysis was required to measure the security of a scheme under a realistic scenario. A security proof is built upon computational complexity theory, using polynomial-time reductions from a well established hard problem to the problem of solving (or breaking) the cryptographic scheme. If this reduction is possible, we can say that breaking the cryptographic scheme is as difficult as solving the well established hard problem (up to a polynomial). If this polynomial is of a high degree, it can degrade the security of the scheme

D. Chakraborty and T. Iwata (Eds.): INDOCRYPT 2018, LNCS 11356, pp. 248–262, 2018.
https://doi.org/10.1007/978-3-030-05378-9_14

considerably, even rendering it useless for practical applications. Bellare and Rogaway [4] started dealing with security reductions that explicitly stated the polynomial factors involved in those reductions, making it possible to build *tight* reductions, in which the polynomial is a small constant.

1.1 Hash-then-Sign Signature Schemes

In 1993, Bellare and Rogaway [3] introduced the Full Domain Hash (FDH) signature scheme based on RSA (RSA-FDH), where the message is hashed to the full domain of the underlying trapdoor function before being signed (also known as "hash-then-sign" schemes). The security proof presented in [3] for RSA-FDH was not tight, making the actual scheme potentially impractical for an acceptable level of security. Fortunately, probabilistic FDH (PFDH) schemes, which prepend a short random string to the message, already allow for tight proofs. In particular, Katz and Wang [22] showed that even a single bit of randomness is enough for achieving tight proofs.

Signature schemes that behave deterministically are usually more efficient and easier to implement, what makes them invaluable for practical applications. Moreover, it is a fact that signature schemes secure in the Random Oracle Model (ROM) are much more practical than schemes secure in the standard model [6,8,10,16,34], therefore, in this paper we only focus on FDH schemes with deterministic signatures in the ROM.

We mainly categorize signature schemes into four distinct classes, namely probabilistic, derandomized, deterministic and unique, that we describe next.

- Probabilistic schemes utilize randomness during the signing process; signatures are always different (with high probability) even if the same message is signed twice with the same signing key. Some examples of probabilistic schemes are PSS [4], Schnorr [28], El-Gamal [14], and Bitcoin's ECDSA.
- Derandomized schemes are probabilistic schemes that demonstrate a deterministic behavior but still requires an internal use of randomness. It is folklore that any randomized signature scheme can be turned into a deterministic one; merely generate the random coins used during the signing algorithm through a pseudo-random function (PRF) that takes the message as input. Then, the random coins used to sign a particular message will be fixed, therefore producing a deterministic signature for each message. Unfortunately, in some cases, the derandomization process can lead to several vulnerabilities [23]. Signature schemes in the derandomized category include the Derandomized Rabin-Williams (DRW) scheme, where the signature is a square root selected uniformly at random out of four possibilities, and returned systematically (by using the PRF "trick").
- Deterministic schemes always produce the same signature for each message without relying on randomness (or derandomization) for signing, but the verification algorithm accepts more than 1 valid signature per message (for each key pair).

– Lastly, unique schemes are deterministic schemes where the verification algorithm only accepts as valid the only signature ever produced by its signing algorithm (for each message and key). Schemes in this category are the Absolute Principal Rabin-Williams (APRW) scheme, and the RSA-FDH (since RSA [27] defines a permutation over $\mathbb{Z}_n^*$).

In Table 1 we show a quick comparison between FDH signature schemes.

Table 1. Comparison of different hash-then-sign signature schemes.

	Assumption	Derandomized?	Unique?	Tight?
DRW [5]	Factoring	✓	✗	✓
APRW [30]	2-Φ/4-Hiding	✗	✓	✓
RSA-FDH [20]	Φ-Hiding	✗	✓	✓
BLS [7]	EC-CDH	✗	✓	✗
Katz-Wang [22]	RSA	✓	✗	✓
Our scheme Π_{u} (Sect. 3.1)	Quadratic residuosity	✗	✓	✓
Our scheme Π_{d} (Sect. 3.2)	Quadratic residuosity	✗	✗	✓

1.2 Previous Work

A seminal impossibility result by Coron [12] states that any FDH signature scheme with *unique* signatures could not hope to have a tight security proof. Kakvi and Kiltz [20] clarified that Coron's impossibility result only holds when the trapdoor permutation is certified. They also presented a tight security proof for RSA-FDH based on the Φ-Hiding assumption [9].

Bernstein [5] studied all variants of Rabin-Williams signatures and devised an ingenious tight proof for the DRW scheme (which he calls "fixed unstructured"), where it releases systematically one of the four square roots that is initially selected at random. Bernstein also provides a non-tight security proof for APRW (the unique signature version of the scheme) and left as an open problem finding a tight proof for it. Seurin [30] first showed that the Rabin function is lossy and then presented a tight security proof for APRW, but under a new assumption dubbed 2-Φ/4-Hiding assumption.

Unique signatures received renewed attention lately, as Bader et al. [2] extended the seminal meta-reduction of Coron [12] by showing that any security proof for unique signatures based on static assumptions or in the security of the underlying trapdoor permutation must lose a factor of q_s in its security reduction, where q_s is the number of signature queries asked by the adversary. Later, Guo et al. [18] clarified that the authors of [2] implicitly assumed in their meta-reduction that the simulator is only allowed to extract information from the adversary's forgeries when trying to invert the underlying trapdoor

permutation; [18] circumvents the impossibility of [2] by allowing the simulator (in addition) to extract information from the adversary's hash queries. In [18] the authors present a unique signature scheme based on Computational Diffie-Hellman (CDH) with a tight security proof, with the drawback that the size of a signature is logarithmic in the number of hash queries asked by the adversary. Shacham [31] improves on the results of [18] and presents a version of the unique scheme of [18] with succinct signatures, where each signature consists of 2 group elements. Unfortunately, the scheme of [31] is still not as fast as RSA-FDH or any Rabin-Williams variant.

Thus, to summarize: All the unique schemes with tight security proofs from the assumption that the underlying trapdoor function (or permutation) is one-way are not efficient. On the other hand, efficient unique schemes such as RSA-FDH and APRW have a tight security proof that relies on the lossiness of the trapdoor function and are based on variants of the Φ-Hiding assumption. Seurin (cf. Theorem 5 in [30]) noted that it is very unlikely that FDH-RSA and APRW will have a tight security reduction from, respectively, inverting RSA or factoring. It is evident that the state of affairs is a bit confusing. FDH-RSA and Rabin-Williams signatures with non-tight proofs were criticized as being potentially impractical due to the large size of the parameters involved. Their tight proofs, however, rely on new assumptions that appear to be markedly stronger than factoring [19,29]. How should these results be interpreted in practice? Should we trust these new assumptions and keep parameters short or should we use large parameters to account for possible cryptanalytic attacks on these new assumptions?

1.3 State of Affairs

What is wrong with randomness? Generating cryptographically-strong random or pseudo-random numbers (RNG or PRNG) has always been a challenging endeavor. Several devices are even unable to generate random numbers that are good enough for cryptographic purposes. For instance, smart cards and sensors are not usually capable of collecting enough entropy. Some are susceptible to *reset* attacks where the PRNG is brought back to previous states. A reset attack can be devastating for signature schemes since it could be possible even to recover the signing keys of the user [26]. The same attack can be applied to virtualized systems where the adversary can take snapshots of a virtual machine and later replay them with distinct messages to recover the signing key. When possible, probabilistic schemes should be avoided in these circumstances.

What is wrong with derandomization? Despite showing a deterministic behavior, derandomized schemes still require randomness to sign messages. Therefore, it is crucial to have a sound derandomization process; otherwise, it can be a source of vulnerabilities [17,23]. For instance, a simple fault attack during the derandomization leads to a full key recovery attack in the derandomized Rabin-Williams scheme (by outputting 2 different square roots of the same message), while deterministic schemes are immune to such attacks.

What is wrong with the Φ-Hiding assumption? The Φ-Hiding assumption appears to be much stronger than factoring, and it does not hold in some cases, as shown in [19,29]. The RSA-FDH scheme is tightly secure under the Φ-Hiding assumption, while the APRW scheme is tightly secure under a new assumption dubbed 2-Φ/4-Hiding assumption [30]. As reported by Seurin [30], the 2-Φ/4-Hiding assumption is clearly stronger than quadratic residuosity (on which our schemes rely instead): When $n \equiv 1 \mod 4$, the 2-Φ/4-Hiding problem is equivalent to the problem of establishing whether -1 is a square in $\mathbb{Z}_n^*$; thus, it's enough to provide $y = -x^2 \mod n$, for a random $x \in \mathbb{Z}_n^*$, to a quadratic residuosity solver to violate the 2-Φ/4-Hiding assumption.

A Case for Unique Signatures. Ateniese et al. [1] shows a generic subversion attack against virtually all probabilistic and deterministic signature schemes that leads to the complete recovery of the signing key. The intuition behind the attack is that the adversary builds a subverted signing algorithm that leaks bits of the signing key through the produced signatures; this is only possible because the signature contains randomness that is used to "disguise" the parts of the signing key that is being leaked. Deterministic schemes are also susceptible to such attacks since the bits of the signing key can still be leaked through the choice of the signature that is returned among the possible options. On the other hand, [1] shows that unique signature schemes are *secure* against the class of subversion attacks that satisfies the verifiability condition[1]. When used in tandem with a cryptographic reverse firewall [25] unique signature schemes are secure against *all* classes of subversion attacks [1]. Therefore, unique signatures are recommended for settings where the generation of randomness is problematic, and subversion attacks are a concern.

1.4 Our Contribution

Our contribution is a family of FDH signature schemes in the ROM with tight security proofs to the Quadratic Residuosity (QR) assumption[2]. The family consists of a unique scheme and a deterministic scheme, both based on a variation of a lossy function from [15]. To argue tight security for the unique signature scheme, we leverage the results of Kakvi and Kiltz [20] that show a generic proof for any unique scheme based on a lossy trapdoor function. As far as we could ascertain, this is the first unique signature scheme tightly secure under the quadratic residuosity assumption (and non-tightly secure under factoring). Besides, the reduction is tighter than the one in [30], i.e., our unique scheme is closer to quadratic residuosity than principal Rabin-Williams is to the 2-Φ/4-Hiding assumption.

[1] The verifiability condition informally says that *all* signatures produced by the signing algorithm must be valid for the corresponding verification key.

[2] Arguably, the next best assumption after factoring is quadratic residuosity, which has been extensively studied, at least as much as the RSA assumption.

The efficiency of the schemes in our family is comparable to that of the Rabin-Williams family, which are considered the fastest (for signature verification) signature schemes ever devised [5]. The unique scheme does require the computation of a Jacobi symbol (as the unique variant of Rabin-Williams also does) but we believe such a computation carries an unfair stigma. In reality, computing Jacobi symbols can be performed very efficiently [24,32] (in particular in $O(n^2/\log n)$ as reported in [24]), and can be parallelized [24] to harness recent multicore and/or distributed platforms. Nevertheless, for applications where the verification process has to be even faster, we provide a deterministic signature scheme that does not require the computation of Jacobi symbols.

2 Preliminaries

2.1 Basic Notations

When A is a deterministic algorithm, we write $y := \mathsf{A}(x)$ to denote a run of A on input x and output y; if A is a randomized algorithm then $y \leftarrow \mathsf{A}(x;r)$ denotes a run of A on input x and randomness r; when it is clear from context we simply write $y \leftarrow \mathsf{A}(x)$. An algorithm A is probabilistic polynomial-time (PPT) if A is randomized and for any input $x, r \in \{0,1\}^*$ the computation of $\mathsf{A}(x;r)$ terminates in at most $poly(|x|)$ steps. We denote with $\kappa \in \mathbb{N}$ the security parameter. A function $\nu : \mathbb{N} \to [0,1]$ is negligible in the security parameter (or simply negligible) if it vanishes faster than the inverse of any polynomial in κ, i.e., $\nu(\kappa) = \kappa^{-\omega(1)}$. For a random variable $\mathbf{X}$, we write $\mathbb{P}[\mathbf{X} = x]$ for the probability that $\mathbf{X}$ takes on a particular value $x \in \mathcal{X}$ (where $\mathcal{X}$ is the set where $\mathbf{X}$ is defined).

2.2 Number Theory

We denote by $\mathbb{J}_n$ the set of all $x \in \mathbb{Z}_n^*$ with Jacobi symbol 1, by $\overline{\mathbb{J}}_n$ the set of all $x \in \mathbb{Z}_n^*$ with Jacobi symbol -1, and by $\mathbb{QR}_n$ the set of all quadratic residues of $\mathbb{Z}_n^*$. For $n \in \mathbb{Z}$, we call n a Williams integer if $n = pq$ for primes p and q of the form $p \equiv 3 \mod 8$ and $q \equiv 7 \mod 8$. Our results rely on the following lemmas from [15,33].

Lemma 1. *Let $n = pq$ be a Williams integer and let $x \in \mathbb{QR}_n$. The equation $x \equiv y^2 \mod n$ takes four distinct values, namely $\{\pm y_0, \pm y_1\}$, where*

(i) for $b \in \{0,1\}$, we have that y_b and $-y_b$ are both either in $\mathbb{J}_n$ or $\overline{\mathbb{J}}_n$,
(ii) $y_0 \in \mathbb{J}_n$ if and only if $y_1 \in \overline{\mathbb{J}}_n$.

Lemma 2. *Let $n = pq$ be a Williams integer, then $2 \in \overline{\mathbb{J}}_n$.*

Lemma 3. *For $n, x, y \in \mathbb{Z}$, where $x \not\equiv \pm y \mod n$, if $x^2 \equiv y^2 \mod n$ then $\gcd(n, x-y)$ gives a non-trivial factor of n.*

2.3 Signature Schemes

A signature scheme is a triple of algorithms $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ specified as follows:

- KGen takes as input the security parameter κ and outputs a verification/signing key pair $(vk, sk) \in \mathcal{VK} \times \mathcal{SK}$, where $\mathcal{VK} := \mathcal{VK}_\kappa$ and $\mathcal{SK} := \mathcal{SK}_\kappa$ denote the sets of all verification and secret keys produced by $\mathsf{KGen}(1^\kappa)$.
- Sign takes as input the signing key $sk \in \mathcal{SK}$, a message $m \in \mathcal{M}$ and random coins $r \in \mathcal{R}$, and outputs a signature $\sigma \in \Sigma$.
- Vrfy takes as input the verification key $vk \in \mathcal{VK}$ and a pair (m, σ), and outputs a decision bit that equals 1 iff σ is a valid signature for message m under the key vk.

The correctness of a signature scheme informally says that verifying honestly generated signatures always works.

Definition 1 (Correctness). *Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. We say that Π satisfies (perfect) correctness if for all (vk, sk) output by KGen, and all $m \in \mathcal{M}$,*

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m, \mathsf{Sign}(sk, m))) = 1\right] = 1,$$

where the probability is taken over the randomness of the signing algorithm.

The standard notion of security for a signature scheme demands that no PPT adversary given access to a signing oracle returning signatures for arbitrary messages, can forge a signature on a "fresh" message (not asked to the signing oracle).

Definition 2 (Existential unforgeability). *Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. We say that Π is (t, q, ε)-existentially unforgeable under chosen-message attacks if for all adversaries A running in time t it holds:*

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m^*, \sigma^*)) = 1 \wedge m^* \notin \mathcal{Q} : \begin{array}{c}(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa);\\ (m^*, \sigma^*) \leftarrow \mathsf{A}^{\mathsf{Sign}(sk, \cdot)}(vk)\end{array}\right] \leq \varepsilon,$$

where $\mathcal{Q} = \{m_1, \ldots, m_q\}$ denotes the set of queries to the signing oracle. If for all $t, q = poly(\kappa)$ there exists $\varepsilon(\kappa) = negl(\kappa)$ such that Π is (t, q, ε)-existentially unforgeable under chosen-message attacks (EUF-CMA for short), then we simply say Π is EUF-CMA.

We define the so-called *unique* signatures next. Informally, a signature scheme is unique if, for any message, there is only a single signature that verifies w.r.t. an honestly generated verification key.

Definition 3 (Uniqueness). *Let Π be a signature scheme. We say that Π satisfies uniqueness if for all vk output by KGen, and all $m \in \mathcal{M}$, there exists a single value $\sigma \in \Sigma$ such that $\mathsf{Vrfy}(vk, (m, \sigma)) = 1$.*

3 The Signature Scheme Family

In this section, we describe the two components of our hash-and-sign family of signature schemes. Our family is a variant of the Rabin-Williams family [5], and is inspired by a *lossy* trapdoor function from [15]. We first describe the unique signature scheme based on QR in Sect. 3.1, followed by the deterministic scheme based on QR in Sect. 3.2.

3.1 Unique Scheme Π_{u}

Let the functions $h, j : \mathbb{Z}_n \to \{0,1\}$ be defined as

$$h(x) = \begin{cases} 1, \text{ if } x > n/2, \\ 0, \text{ otherwise,} \end{cases}$$

$$j(x) = \begin{cases} 1, \text{ if } x \in \overline{\mathbb{J}}_n, \\ 0, \text{ otherwise.} \end{cases}$$

We build the unique signature scheme $\Pi_{\mathsf{u}} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ as follows:

- $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$: The key generation algorithm takes as input the security parameter 1^κ and produces a pair of corresponding verification and signing keys. The signing key sk is composed of two randomly sampled $\kappa/2$-bit primes p and q of the form $p \equiv 3 \mod 8$ and $q \equiv 7 \mod 8$. The verification key vk is defined by $n := pq$ and a randomly sampled parameter $s \in \mathbb{J}_n \setminus \mathbb{QR}_n$.
- $\sigma := \mathsf{Sign}(sk, m)$: Set $b := 0$ and hash the message m to obtain $x := H(m)$, where $H : \{0,1\}^* \to \mathbb{Z}_n^*$ is a collision-resistant hash function. Compute $x' := x \cdot 2^{j(x)} \mod n$ and iff $x' \notin \mathbb{QR}_n$ set $b := 1$ and compute $x' := x' \cdot s \mod n$ with the public parameter s. Now that $x' \in \mathbb{QR}_n$ we use the signing key to compute the four modular square roots of x' and select the single root y such that $j(y) = j(x)$ and $h(y) = b$ (according to Lemma 1); set $\sigma := y$ and output σ.
- $b := \mathsf{Vrfy}(vk, m, \sigma)$: If $\sigma \notin \{1, \ldots, n-1\}$ then output 0, otherwise output $H(m) = \sigma^2 \cdot 2^{-j(\sigma)} \cdot s^{-h(\sigma)} \mod n$.

On uniqueness. We note that for a signature scheme to be considered unique, it is necessary, but not sufficient, that the signing algorithm always returns the same signature when the same message is signed more than once. To fully characterize a unique signature scheme, the verification algorithm needs (for each verification key vk) to reject as invalid all the signatures for a particular message m, except the only signature for m that is ever returned by the signing algorithm. It is easy to see that the scheme Π_{u} above satisfies these requirements, as for each key a single signature σ is ever produced for some message m, and only σ is ever accepted as a signature for m.

3.2 Deterministic Scheme Π_{d}

In order to achieve even better efficiency, we construct additionally the deterministic variant Π_{d} of the previous signature scheme. We define the deterministic scheme $\Pi_{\mathsf{d}} = (\mathsf{KGen}', \mathsf{Sign}', \mathsf{Vrfy}')$, where the algorithms KGen' and Sign' are exactly the same as KGen and Sign in Π_{u}, and the verification algorithm Vrfy' is described below:

– $b := \mathsf{Vrfy}'(vk, m, \sigma)$: If $\sigma \notin \{1, \ldots, n-1\}$ then output 0, otherwise output $(H(m) = \sigma^2 \cdot s^{-h(\sigma)} \mod n) \vee (H(m) = \sigma^2 \cdot s^{-h(\sigma)} \cdot 2^{-1} \mod n)$.

Note that although the signing algorithm will always return a unique signature for each message, the verification algorithm does accept 2 different signatures for a message. The main advantage of the deterministic scheme over the unique scheme is efficiency; while the unique scheme requires computation of a Jacobi symbol in the signature verification, the deterministic scheme only needs to perform 3 modular multiplications (in the worst case).

4 Security Analysis

In this section, we analyze the security of the signature schemes presented in Sect. 3. We first present a security proof for Π_{u} based on the hardness of factoring, and then a *tight* security proof based on QR. To achieve the latter, we leverage the results of Kakvi and Kiltz [21] on unique signatures based on *lossy* functions. Later we also present a tight security proof for the Π_{d} signature scheme based on QR.

4.1 Security of Π_{u} Based on Factoring

Theorem 1. *If the Integer Factorization Problem (IFP) is (t, ε)-hard, then the unique signature scheme Π_{u} is $(t', q_h, q_s, \varepsilon')$-secure, with*

$$t = t' + (q_h + q_s + 1) \cdot O(\kappa^2) \quad \text{and} \quad \varepsilon = \frac{\varepsilon'}{4 \cdot (q_h + q_s + 1)}.$$

Proof. Let A be an adversary that $(t', q_h, q_s, \varepsilon')$-breaks Π_{u}. We build a reduction R that uses A as a subroutine and (t, ε)-breaks the IFP.

The reduction R receives a modulus $n = pq$ from the challenger, and its objective is to factor n. Instead of sampling $s \in \mathbb{J}_n \setminus \mathbb{QR}_n$, which R is not able to, it simply samples an $s \in \mathbb{J}_n$. When $s \in \mathbb{QR}_n$ the reduction aborts, what happens with probability 1/2. The reduction R sends $vk := (n, s)$ to A. We allow the adversary A to make two types of oracle queries, namely hash and sign queries, that R must answer with the same distribution as a real signing oracle would. The reduction R maintains a list $\mathcal{L} := \emptyset$ of hash queries and a counter i that

is initialized by 0. R chooses a random $\ell \in \{1, ..., q\}$, where $q := q_h + q_s$, and answers the queries as follows:

- **Hash queries:** Upon a hash query for message m check if $m \in \mathcal{L}$; if yes, then return x from the triple $(m, x, y) \in \mathcal{L}$, otherwise proceed as follows. Increment the counter i, and if $i \neq \ell$ the reduction R chooses a random $y_i \in \mathbb{Z}_n^*$ and sets $x_i = y_i^2 \cdot 2^{-j(y_i)} \cdot s^{-h(y_i)} \mod n$. However, when $i = \ell$, reduction R chooses random values $y_i \in \mathbb{Z}_n^*, \alpha, \beta \in \{0, 1\}$ and sets $x_i = y_i^2 \cdot 2^{-\alpha} \cdot s^{-\beta} \mod n$. Store the triple (m_i, x_i, y_i) in the list $\mathcal{L}$ and return x_i.
- **Sign queries:** When A makes a sign query for a message m, reduction R checks if there exists a triple $(m, x, y) \in \mathcal{L}$; if not, R simply makes the corresponding hash query itself. Return y as the signature of message m.

The adversary A eventually outputs a forgery (m_i, σ_i), and we assume wlog that $(m_i, x_i, y_i) \in \mathcal{L}$. If $i = l$ we have that both $y' = \sigma_i \cdot 2^{-j(\sigma)} \cdot s^{-h(\sigma)} \mod n$ and y_i are square roots of y_i^2. With probability 1/2, the roots y' and y_i are not the complement of each other, and in that case we can factor n by computing $\gcd(n, y' - y_i)$, due to Lemma 3. The running time for R is the running time of the adversary A plus all the oracle queries. □

The reduction R is required to answer all the oracle queries that A makes; in particular, R needs to produce valid signatures to all the messages queried by A without knowing the signing key. Before every signature query for message m is made, a corresponding hash query for m needs to be made to the reduction R; the reduction first samples a random $y \in \mathbb{Z}_n^*$ and returns $H(m) := y^2 \cdot 2^{-j(y)} \cdot s^{-h(y)} \mod n$ as the answer to the hash query. To answer a signature query for message m, the reduction R returns y as a valid signature for m.

In order to factor, R selects an index $\ell \in \{1, \ldots, q\}$ during initialization, and for the ℓ-th hash query made by A the reduction R replies with $x_\ell = y_\ell^2 \cdot 2^{-\alpha} \cdot s^{-\beta} \mod n$ for $y_\ell \in \mathbb{Z}_n^*, \alpha, \beta \in \{0, 1\}$. The reduction is then able to factor with probability 1/2 if A produces a pair (m_ℓ, σ_ℓ) as a forgery for the message m_ℓ.

We note that the above security reduction can be further improved to roughly $\varepsilon = \varepsilon'/4q_s$ by applying a technique by Coron [11].

4.2 Tight Security of Π_{u} Based on QR

The unique signature scheme Π_{u} of Sect. 3.1 is a variant of a lossy trapdoor function based on QR from [15]. In fact, the changes made to our scheme were carefully crafted so the scheme would still maintain its lossiness; the main difference is that n is a Williams integer so that $2 \in \overline{\mathbb{J}}_n$.

To instantiate the lossy version of our scheme, KGen needs to be modified to sample the public parameter $s \in \mathbb{QR}_n$, in contrast to the injective version, where $s \in \mathbb{J}_n \setminus \mathbb{QR}_n$. Note that the only difference between the lossy and the injective version of the scheme is the domain of s; in both cases $s \in \mathbb{J}_n$, but in the lossy version $s \in \mathbb{QR}_n$, while in the injective version $s \notin \mathbb{QR}_n$. Distinguishing among these two cases is precisely the QR assumption, so an adversary that is able to

distinguish must solve the QR problem. Since the lossy version of the scheme is 2-to-1 [15] and the injective version is a permutation in $\{1, \ldots, n\}$, the scheme has lossiness of 1-bit.

For the tight security proof of the scheme Π_{u} we leverage the generic result of Kakvi and Kiltz [21] for unique signatures based on lossy functions, that intuitively states that any unique signature scheme based on a lossy function has a tight security reduction based on the lossiness of the function. From that, we achieve the following result.

Theorem 2. *If the Quadratic Residuosity assumption is $(t_{QR}, \varepsilon_{QR})$-hard, then for any q_h, q_s the unique signature scheme Π_{u} is $(t, q_h, q_s, \varepsilon)$-EUF-CMA secure in the random oracle model with*

$$t = t_{QR} - q_h \cdot \mathcal{O}(\kappa^2) \quad \textit{and} \quad \varepsilon = 3 \cdot \varepsilon_{QR}.$$

4.3 Tight Security of Π_{d} Based on QR

In this section, we build a reduction from breaking the security of the Π_{d} scheme to breaking the security of the Π_{u} scheme. Since the Π_{u} scheme has tight security to the QR problem, then Π_{d} has also tight security to the QR problem.

Theorem 3. *If the Π_{u} scheme is $(t', q_h, q_s, \varepsilon')$-EUF-CMA secure, then the deterministic signature scheme Π_{d} is $(t, q_h, q_s, \varepsilon)$-EUF-CMA secure, with $t = t'$, and $\varepsilon = 2 \cdot \varepsilon'$.*

Proof. Assume there exists an adversary A that $(t, q_h, q_s, \varepsilon)$-breaks the security of Π_{d}. Then, we build another adversary A' that $(t', q_h, q_s, \varepsilon')$-breaks the security of Π_{u}.

Adversary A':

- Receive the verification key $vk := (n, s)$ from the challenger and send it to A.
- Upon any hash or signature query from A, forward the query to its corresponding oracle and send the reply to A.
- Eventually, receive a forgery (m, σ) from A. Sample a random bit b and return the pair $(m, \sigma \cdot 2^b)$ to the challenger as a forgery for message m.

For the analysis, we note that the simulation performed by A' is perfect since the hash and signature oracles from both schemes are exactly the same. By assumption, A produces a valid forgery (m, σ) with non-negligible probability, and in that case, $(m, \sigma \cdot 2^b)$ is a valid forgery for A' when $\sigma \cdot 2^b$ has the same Jacobi symbol as $H(m)$, what happens with probability $1/2$ when b is sampled at random. Therefore, if A breaks the security of Π_{d} with probability ε, then A' breaks the security of Π_{u} with probability $\varepsilon/2$. □

5 Performance

When the factors p and q are known, calculating the Jacobi symbol of an element is very efficient since it is enough to compute two Legendre symbols. In particular, for $x \in \mathbb{Z}_n^*$, $x \in \mathbb{J}_n$ if $x^{(p-1)/2} \mod p = x^{(q-1)/2} \mod q$, otherwise $x \in \overline{\mathbb{J}}_n$.[3] The signature σ is the unique square root y of the square x such that $j(y) = j(x)$ and $y > n/2$ iff $x > n/2$. Computing such a square root is very efficient thanks to the Chinese remainder theorem.

In general, when p and q are known, the computation of the Jacobi symbol and a square root share several calculations and can be optimized when performed simultaneously. Since computing Jacobi symbols when p and q are unknown is computationally more expensive than other modular operations, we recommend the deterministic version Π_{d} of our scheme for applications where unique signatures are not necessary.

While in the unique signature scheme the computation of a Jacobi symbol (for signature verification) is necessary, in the deterministic scheme it is enough to compute $t := \sigma^2 \cdot s^{-h(\sigma)} \mod n$ and then check whether any of $H(m) = t$ or $H(m) = t \cdot 2^{-1} \mod n$ holds to consider the signature σ as valid.

A note on efficiency. Our Π_{u} scheme has comparable speed to the unique signature scheme from the Rabin-Williams family, denoted by APRW* in [30]. The running time of the verification algorithm is dominated by the computation of a Jacobi symbol in both schemes. Our deterministic scheme Π_{d} is very efficient, requiring at most 3 modular multiplications for signature verification.

6 Conclusions

We presented a family of FDH signature schemes with tight security based on a standard assumption (QR). The schemes are as efficient as other variants of Rabin-Williams which hold the record for fastest signature verification schemes [5]. A tight security proof for the APRW scheme was presented only recently by Seurin [30], and his proof is based on the lossiness of the APRW function, which is based on a new assumption called 2-Φ/4-Hiding, that is a variation of the Φ-Hiding problem [9]. Unlike QR, the Φ-Hiding problem is a new and poorly understood assumption as remarked in [19,29].

In practice, since the security of our signature scheme is based on the QR assumption, in comparison to RSA-FDH and APRW, it is possible to safely employ smaller parameters for comparable levels of security, which leads to even better efficiency.

[3] We do not consider cases where the Jacobi or Legendre symbols are 0 since they happen with negligible probability.

References

1. Ateniese, G., Magri, B., Venturi, D.: Subversion-resilient signature schemes. In: Ray, I., Li, N., Kruegel, C. (eds.) 22nd Conference on Computer and Communications Security – ACM CCS 2015, pp. 364–375. ACM Press (2015)
2. Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 273–304. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_10
3. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 93: 1st Conference on Computer and Communications Security, pp. 62–73. ACM Press, November 1993
4. Bellare, M., Rogaway, P.: The exact security of digital signatures: how to sign with RSA and Rabin. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9_34
5. Bernstein, D.J.: Proving tight security for Rabin-Williams signatures. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 70–87. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_5
6. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_4
7. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. J. Cryptol. **17**(4), 297–319 (2004)
8. Boneh, D., Shen, E., Waters, B.: Strongly unforgeable signatures based on computational Diffie-Hellman. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 229–240. Springer, Heidelberg (2006). https://doi.org/10.1007/11745853_15
9. Cachin, C., Micali, S., Stadler, M.: Computationally Private Information Retrieval with Polylogarithmic Communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_28
10. Chevallier-Mames, B., Joye, M.: A practical and tightly secure signature scheme without hash function. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 339–356. Springer, Heidelberg (2006). https://doi.org/10.1007/11967668_22
11. Coron, J.-S.: On the exact security of full domain hash. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 229–235. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_14
12. Coron, J.-S.: Optimal security proofs for PSS and other signature schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_18
13. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
14. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**, 469–472 (1985)
15. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. J. Crypt. **26**(1), 39–74 (2013)
16. Gennaro, R., Halevi, S., Rabin, T.: Secure hash-and-sign signatures without the random oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_9

17. Granboulan, L.: How to repair ESIGN. In: Cimato, S., Persiano, G., Galdi, C. (eds.) SCN 2002. LNCS, vol. 2576, pp. 234–240. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36413-7_17
18. Guo, F., Chen, R., Susilo, W., Lai, J., Yang, G., Mu, Y.: Optimal security reductions for unique signatures: bypassing impossibilities with a counterexample. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10402, pp. 517–547. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_18
19. Herrmann, M.: Improved cryptanalysis of the multi-prime ϖ - hiding assumption. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 92–99. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21969-6_6
20. Kakvi, S.A., Kiltz, E.: Optimal security proofs for full domain hash, revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 537–553. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_32
21. Kakvi, S.A., Kiltz, E.: Optimal security proofs for full domain hash, revisited. J. Crypt. **31**(1), 276–306 (2018)
22. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Sushil, J., Vijayalakshmi, A., Trent, J. (eds.) ACM CCS 03: 10th Conference on Computer and Communications Security – ACM CCS 2003, pp. 155–164. ACM Press, October 2003
23. Leurent, G., Nguyen, P.Q.: How risky is the random-oracle model? In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 445–464. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_26
24. Eikenberry, S.M., Sorenson, J.P.: Efficient algorithms for computing the Jacobi symbol. J. Symb. Comput. **26**, 509–523 (1998)
25. Mironov, I., Stephens-Davidowitz, N.: Cryptographic reverse firewalls. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 657–686. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_22
26. Ristenpart, T., Yilek, S.: When good randomness goes bad: virtual machine reset vulnerabilities and hedging deployed cryptography. In: ISOC Network and Distributed System Security Symposium - NDSS 2010. The Internet Society, February/March 2010
27. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signature and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
28. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, Gilles (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_22
29. Schridde, C., Freisleben, B.: On the validity of the phi-hiding assumption in cryptographic protocols. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 344–354. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_21
30. Seurin, Y.: On the lossiness of the rabin trapdoor function. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 380–398. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_22
31. Shacham, H.: Short unique signatures from RSA with a tight security reduction (in the random oracle model). In: 22nd Financial Cryptography and Data Security (2018)

32. Shallit, J., Sorenson, J.: A binary algorithm for the Jacobi symbol. ACM SIGSAM Bull. **27**, 4–11 (1993)
33. Shoup, V.: A Computational Introduction to Number Theory and Algebra. Cambridge University Press, New York (2009)
34. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7