This is the Full Version of the Extended Abstract that appears
in the Proceedings of the 16th International Conference on Practice and Theory in Public-Key Cryptography (PKC '13)
(26 February – 1 March 2013, Nara, Japan)
Kaoru Kurosawa Ed., Springer-Verlag, LNCS 7778, pages 292–311.

Tighter Reductions for Forward-Secure Signature Schemes

Michel Abdalla Fabrice Ben Hamouda David Pointcheval

Departement d'Informatique, École normale supérieure
45 Rue d'Ulm, 75230 Paris Cedex 05, France
{Michel.Abdalla,Fabrice.Ben.Hamouda,David.Pointcheval}@ens.fr
http://www.di.ens.fr/users/{mabdalla,fbenhamo,pointche}

Abstract

In this paper, we revisit the security of factoring-based signature schemes built via the Fiat-Shamir transform and show that they can admit tighter reductions to certain decisional complexity assumptions such as the quadratic-residuosity, the high-residuosity, and the ϕ-hiding assumptions. We do so by proving that the underlying identification schemes used in these schemes are a particular case of the lossy identification notion recently introduced by Abdalla *et al.* at Eurocrypt 2012. Next, we show how to extend these results to the forward-security setting based on ideas from the Itkis-Reyzin forward-secure signature scheme. Unlike the original Itkis-Reyzin scheme, our construction can be instantiated under different decisional complexity assumptions and has a much tighter security reduction. Finally, we show that the tighter security reductions provided by our proof methodology can result in concrete efficiency gains in practice, both in the standard and forward-security setting, as long as the use of stronger security assumptions is deemed acceptable. All of our results hold in the random oracle model.

1 Introduction

A common paradigm for constructing signature schemes is to apply the Fiat-Shamir transform [FS87] to a secure three-move canonical identification protocol. In these protocols, the prover first sends a commitment to the verifier, which in turn chooses a random string from the challenge space and sends it back to the prover. Upon receiving the challenge, the prover sends a response to the verifier, which decides whether or not to accept based on the conversation transcript and the public key. To obtain the corresponding signature scheme, one simply makes the signing and verification algorithms non-interactive by computing the challenge as the hash of the message and the commitment. As shown by Abdalla *et al.* in [AABN02], the resulting signature scheme can be proven secure in the random oracle model as long as the identification scheme is secure against passive adversaries and the commitment has large enough min-entropy. Unfortunately, the reduction to the security of the identification scheme is not tight and loses a factor q_h, where q_h denotes the number of queries to the random oracle.

If one assumes additional properties about the identification scheme, one can avoid impossibility results such as those in [GBL08, PV05, Seu12] and obtain a signature scheme with a tighter proof of security. For instance, in [MR02], Micali and Reyzin introduced a new method for converting identification schemes into signature schemes, known as the "swap method", in which they reverse the roles of the commitment and challenge. More precisely, in their transform, the challenge is chosen uniformly at random from the challenge space and the commitment is computed as the hash of the message and the challenge. Although they only provided a tight security proof for the modified version of Micali's signature scheme [Mic94], their method generalizes to any scheme in which the prover can compute the response given only the challenge and the commitment, such as the factoring-based schemes in [FFS88, FS87, GQ90, OO90, OS90]. This is due to the fact that the prover in these schemes possesses a trapdoor (such as the factorization of the modulus in the public key) which allows it to compute the response. On the other hand, their method does not apply to discrete-log-based identification schemes in which the prover needs to know the discrete log with respect to the commitment when computing the response, such as in [Sch90].

In 2003, Katz and Wang [KW03] showed that tighter security reductions can be obtained even with respect to the Fiat-Shamir transform, by relying on a proof of membership rather than a proof of knowledge. In particular, using this idea, they proposed a signature scheme with a tight security reduction to the hardness of the DDH problem. They also informally mentioned that one could obtain similar results based on the quadratic-residuosity problem by relying on a proof that shows that a set of elements in $\mathbb{Z}_N^*$ are all quadratic residues. This result was recently extended to other settings by Abdalla *et al.* [AFLT12], who presented three new signature schemes based on the hardness of the short exponent discrete log problem [PS98, vW96], on the worst-case hardness of the shortest vector problem in ideal lattices [LM06, PR06], and on the hardness of the Subset Sum problem [IN96, MM11]. Additionally, they also formalized the intuition in [KW03] by introducing the notion of lossy identification schemes and showing that any such schemes can be transformed into a signature scheme via the Fiat-Shamir transform while preserving the tightness of the reduction.

Tight security from lossy identification. In light of these recent results, we revisit in this paper the security of factoring-based signature schemes built via the Fiat-Shamir transform. Even though the swap method from [MR02] could be applied in this setting (resulting in a slightly different scheme), our first contribution is to show that these signature schemes admit tight security reductions to certain decisional complexity assumptions such as the quadratic-residuosity, the high-residuosity [Pai99], and the ϕ-hiding [CMS99] assumptions. We do so by showing that the underlying identification schemes used in these schemes are a particular case of a lossy identification scheme [AFLT12]. As shown in Section 4.1 in the case of the Guillou-Quisquater signature scheme [GQ90], our tighter security reduction can result in concrete efficiency gains with respect to the swap method. However, this comes at the cost of relying on a stronger security assumption, namely the ϕ-hiding [CMS99] assumption.

Tighter reductions for forward-secure signatures. Unlike the swap method of Micali and Reyzin, the prover in factoring-based signature schemes built via the Fiat-Shamir transform does not need to know the factorization of the modulus in order to be able to compute the response. Using this crucial fact, the second main contribution of this paper is to extend our results to the forward-security setting. To achieve this goal, we first introduce in Section 3 the notion of lossy key-evolving identification schemes and show how the latter can be turned into forward-secure signature schemes using a generalized version of the Fiat-Shamir transform. As in the case of standard signature schemes, this transformation does not incur a loss of factor of q_h in the security reduction. Nevertheless, we remark that the reduction is not entirely tight as we lose a factor T corresponding to the total number of time periods.

After introducing the notion of lossy key-evolving identification schemes, we show in Section 4.2 that a variant of the Itkis-Reyzin forward-secure signature scheme [IR01] (which can be seen as an extension of the Guillou-Quisquater scheme to the forward-security setting) admits a much tighter security reduction, albeit to a stronger assumption, namely the ϕ-hiding assumption.

Concrete security. As in the case of standard signature schemes, the tighter security reductions provided by our proof methodology can result in concrete efficiency gains in practice. More specifically, as we show in Section 5, our variant of the Itkis-Reyzin scheme outperforms the original scheme for most concrete choices of parameters.

Generic factoring-based signatures and forward-secure signatures. As an additional contribution, we show in Section 6 that all the above-mentioned schemes can be seen as straightforward instantiations of a generic factoring-based forward-secure signature scheme. This enables us to not only easily prove the security properties of these schemes, but to also design a new forward-secure scheme based on a new assumption, the 2^t-strong-residuosity.

Organization. After recalling some definitions in Section 2, we introduce the notion of key-evolving lossy identification scheme and show how to transform such a scheme into a forward-secure signature scheme in Section 3. Then, in Section 4, we apply our security proof methodology to two cases: the Guillou-Quisquater scheme and its extension to the forward-secure case (i.e., our variant of the Itkis-Reyzin scheme). In Section 5, we compare this second scheme with the original Itkis-Reyzin scheme and the MMM scheme by Malkin, Micciancio and Miner [MMM02]. Finally, we introduce our generic lossy key-evolving identification scheme and show various instantiations of it in Section 6.

2 Definitions

2.1 Notation and Conventions

Let $\mathbb{N}$ denote the set of natural numbers. If $n \in \mathbb{N}$, then $\{0,1\}^n$ denotes the set of n-bit strings, and $\{0,1\}^*$ is the set of all bit strings. The empty string is denoted $\perp$. If x is a string then $|x|$ denotes its length, and if S is a set then $|S|$ denotes its size. If S is finite, then $x \xleftarrow{\$} S$ denotes the assignment to x of an element chosen uniformly at random from S. If $\mathcal{A}$ is an algorithm, then $y \leftarrow \mathcal{A}(x)$ denotes the assignment to y of the output of $\mathcal{A}$ on input x, and if $\mathcal{A}$ is randomized, then $y \xleftarrow{\$} \mathcal{A}(x)$ denotes that the output of an execution of $\mathcal{A}(x)$ with fresh coins assigned to y. Unless otherwise indicated, an algorithm may be randomized. We denote by $k \in \mathbb{N}$ the security parameter. Let $\mathbb{P}$ denote the set of primes and $\mathbb{P}_{\ell_e}$ denote the set of primes of length ℓ_e. All our schemes are in the random oracle model [BR93].

2.2 Complexity Assumptions

The security of the signature schemes being analyzed in this paper will be based on decisional assumptions over composite-order groups: the e-residuosity assumption, the ϕ-hiding assumption and a new assumption called the strong-2^t-residuosity. We also need to recall the strong-RSA assumption to be able to compare our scheme with the Itkis-Reyzin scheme [IR01].

Let N be the product of distinct large primes p_1 and p_2. We call such N an RSA modulus. Informally, the **e-residuosity** assumption states that the problem of deciding whether a given element y in $\mathbb{Z}_N^*$ is an e-residue or not is intractable without knowing the factorization of N. Remember that an element $y \in \mathbb{Z}_N^*$ is said to be an e-residue if there exists an element $x \in \mathbb{Z}_N^*$ such that $y = x^e \bmod N$. If $e = 2$, this assumption is called the **quadratic-residuosity** assumption. Furthermore, if we extend it to $N = e^2$, with e an RSA modulus, this is called the **high-residuosity** assumption [Pai99]. Likewise, the **ϕ-hiding** assumption, introduced by Cachin, Micali, and Stadler in [CMS99], states that it is hard for an adversary to tell whether a prime number e divides the order of the group $\mathbb{Z}_N^*$ or not. Next, we introduce the **strong-2^t-residuosity** assumption that states that it is hard for an adversary to decide whether a given element y in $\mathbb{Z}_N^*$ is a 2^t-residue or is even not a 2-residue, when 2^t divides $p_1 - 1$ and $p_2 - 1$. Finally, the **strong-RSA** assumption states that, given an element $y \in \mathbb{Z}_N^*$, it is hard for an adversary to find an integer $e \geq 2$ and an element $x \in \mathbb{Z}_N^*$ such that $y = x^e \bmod N$.

For each of these assumptions, the underlying problem is said to be (t, ε)-hard, if no adversary running in time at most t is able to solve the problem with probability at least ε. Formal descriptions of the assumptions can be found in Appendix A.2.

2.3 Forward-Secure Signature Schemes

A forward-secure signature scheme is a key-evolving signature scheme in which the secret key is updated periodically while the public key remains the same throughout the lifetime of the scheme [BM99]. Each time period has a secret signing key associated with it, which can be used to sign messages with respect to that time period. The validity of these signatures can be checked with the help of a verification algorithm. At the end of each time period, the signer in possession of the current secret key can generate the secret key for the next time period via an update algorithm. Moreover, old secret keys are erased after a key update.

Formally, a key-evolving signature scheme is defined by a tuple of algorithms $\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$ and a message space $\mathcal{M}$, providing the following functionality. Via $(pk, sk) \xleftarrow{\$} \mathsf{KG}(1^k, 1^T)$, a user can run the probabilistic key generation algorithm KG to obtain a pair (pk, sk_1) of public and secret keys for a given security parameter k and a given total number of periods T. sk_1 is the secret key associated with time period 1. Via $sk_{i+1} \leftarrow \mathsf{Update}(sk_i)$, the user in possession of the secret key sk_i associated with time period $i \leq T$ can generate a secret key sk_{i+1} associated with time period $i + 1$. By convention, $sk_{T+1} = \perp$. Via $\langle \sigma, i \rangle \xleftarrow{\$} \mathsf{Sign}(sk_i, M)$, the user in possession of the secret key sk_i associated with time period $i \leq T$ can generate a signature $\langle \sigma, i \rangle$ for a message $M \in \mathcal{M}$ for period i. Finally, via $d \leftarrow \mathsf{Ver}(pk, \langle \sigma, i \rangle, M)$, one can run the deterministic verification algorithm to check if σ is a valid signature for a message $M \in \mathcal{M}$ for period i and public key pk, where $d = 1$ if the signature is correct and 0 otherwise. For correctness, it is required that for all honestly generated keys $(sk_1, \ldots, sk_T)$ and for all messages $M \in \mathcal{M}$, $\mathsf{Ver}(pk, \mathsf{Sign}(sk_i, M), M) = 1$ holds with all but negligible probability.

Informally, a key-evolving signature scheme is **existentially forward-secure** under adaptive chosen-message attack (EUF-CMA), if it is infeasible for an adversary —also called forger— to forge a signature σ^* on a message M^* for a time period i^*, even with access to the secret key for a period $i > i^*$ (and thus to all the subsequent secret keys; this period i is called the breakin period) and to signed messages of his choice for any period (via a signing oracle), as long as he has not requested a signature on M^* for period i^* to the signing oracle. This notion is a generalization of the existential unforgeability under adaptive chosen-message attacks (EUF-CMA for signature schemes) [GMR85] to key-evolving signature scheme and a slightly stronger variant of the definition in [BM99]. In particular, we do not restrict the adversary to only perform signing queries with respect to the current time period.

In the remainder of the paper, we also use a stronger notion: **forward security** (SUF-CMA). In this notion, the forger is allowed to produce a signature σ^* on a message M^* for a period i^*, such that the triple (M^*, i^*, σ^*) is different from all the triples produced by the signing oracle. More formally, a key-evolving signature scheme is $(t, q_h, q_s, \varepsilon)$-(existentially)-forward-secure if no adversary running in

time at most t and making at most q_h queries to the random oracle and q_s queries to the signing oracle can break the (existential) forward security with probability at least ε. All the formal security notions and the comparison with [BM99], together with other security notions (used for detailed comparisons), can be found in Appendices A.3 and A.4.

3 Lossy Key-Evolving Identification and Signature Schemes

In this section, we present a new notion, called lossy key-evolving identification scheme, which combines the notions of lossy identification schemes [AFLT12], which can be transformed to tightly secure signature scheme, and key-evolving identification schemes [BM99], which can be transformed to forward-secure signature via a generalized Fiat-Shamir transform (not necessarily tight, and under some conditions). Although this new primitive is not very useful for practical real-world applications, it is a tool that will enable us to construct forward-secure signatures with tight reductions, via the generalized Fiat-Shamir transform described in Section 3.2.

3.1 Lossy Key-Evolving Identification Scheme

The operation of a key-evolving identification scheme is divided into time periods $1, \ldots, T$, where a different secret is used in each time period, and such that the secret key for a period $i+1$ can be computed from the secret key for the period i. The public key remains the same in every time period. In this paper, a key-evolving identification scheme is a three-move protocol in which the prover first sends a **commitment** cmt to the verifier, then the verifier sends a **challenge** ch uniformly at random, and finally the prover answers by a **response** rsp. The verifier's final decision is a deterministic function of the conversation with the prover (the triple (cmt, ch, rsp)), of the public key, and of the index of the current time period.

Informally, a lossy key-evolving identification scheme has $T+1$ kinds of public keys: normal public keys, which are used in the real protocol, and i-lossy public keys, for $i \in \{1, \ldots, T\}$, which are such that no prover (even not computationally bounded) should be able to make the verifier accept for the period i with non-negligible probability. Furthermore, for each period i, it is possible to generate a i-lossy public key, such that the latter is indistinguishable from a normal public key even if the adversary is given access to any secret key for period $i' > i$.

More formally, a lossy key-evolving identification scheme $\mathcal{ID}$ is defined by a tuple $(\mathsf{KG}, \mathsf{LKG}, \mathsf{Update}, \mathsf{Prove}, \ell_c, \mathsf{Ver})$ such that:

- KG is the normal key generation algorithm which takes as input the security parameter k and the number of periods T and outputs a pair (pk, sk_1) containing the public key and the prover's secret key for the first period.
- LKG is the lossy key generation algorithm which takes as input the security parameter k and the number of periods T and a period i and outputs a pair (pk, sk_{i+1}) containing a i-lossy public key pk and a prover's secret key for period $i+1$ ($sk_{T+1} = \perp$).
- Update is the deterministic secret key update algorithm which takes as input a secret key sk_i for period i and outputs a secret key sk_{i+1} for period $i+1$ if sk_i is a secret key for some period $i < T$, and $\perp$ otherwise. We write Update^j the function Update composed j times with itself ($\mathsf{Update}^j(sk_i)$ is a secret key sk_{i+j} for period $i+j$, if $i+j \leq T$).
- Prove is the prover algorithm which takes as input the secret key for the current period, the current conversation transcript (and the current state st associated with it, if needed) and outputs the next message to be sent to the verifier, and the next state (if needed). We suppose that any secret key sk_i for period i always contains i, and so i is not an input of Prove.
- ℓ_c is a polynomial; $\ell_c(k)$ (often simply denoted ℓ_c) is the length of the challenge sent by the verifier.

- Ver is the deterministic verification algorithm which takes as input the conversation transcript and the period i and outputs 1 to indicate acceptance, and 0 otherwise.

A randomized transcript generation oracle $\mathsf{Tr}^{\mathcal{ID}}_{pk,sk_i,k}$ is associated to each $\mathcal{ID}$, k, and (pk, sk_i). $\mathsf{Tr}^{\mathcal{ID}}_{pk,sk_i,k}$ takes no inputs and returns a random transcript of an "honest" execution for period i. More precisely, the transcript generation oracle $\mathsf{Tr}^{\mathcal{ID}}_{pk,sk_i,k}$ is defined as follows:

function $\mathsf{Tr}^{\mathcal{ID}}_{pk,sk_i,k}$
$(cmt, st) \xleftarrow{\$} \mathsf{Prove}(sk_i)$; $ch \xleftarrow{\$} \{0,1\}^{\ell_c}$; $rsp \xleftarrow{\$} \mathsf{Prove}(sk_i, cmt, ch, st)$
return (cmt, ch, rsp)

An identification scheme is said to be lossy if it has the following properties:

(1) **Completeness of normal keys.** $\mathcal{ID}$ is said to be complete, if for every period i, every security parameter k and all honestly generated keys $(pk, sk_1) \xleftarrow{\$} \mathsf{KG}(1^k)$, $\mathsf{Ver}(pk, cmt, ch, rsp, i) = 1$ holds with probability 1 when $(cmt, ch, rsp) \xleftarrow{\$} \mathsf{Tr}^{\mathcal{ID}}_{pk,sk_i,k}()$, with $sk_i = \mathsf{Update}^{i-1}(sk_1)$.

(2) **Simulatability of transcripts.** Let (pk, sk_1) be the output of $\mathsf{KG}(1^k)$ for a security parameter k, and sk_i be the output of $\mathsf{Update}^{i-1}(sk_1)$. Then, $\mathcal{ID}$ is said to be ε-simulatable if there exists a probabilistic polynomial time algorithm $\widetilde{\mathsf{Tr}}^{\mathcal{ID}}_{pk,i,k}$ with no access to any secret key, which can generate transcripts $\{(cmt, ch, rsp)\}$ whose distribution is statistically indistinguishable from the transcripts output by $\mathsf{Tr}^{\mathcal{ID}}_{pk,sk_i,k}$, where ε is an upper-bound for the statistical distance. When $\varepsilon = 0$, then $\mathcal{ID}$ is said to be simulatable.

(3) **Indistinguishability of keys.** Consider the two following experiments $\mathbf{Exp}^{\text{ind-keys-real}}_{\mathcal{ID},k,i}(\mathsf{D}_i)$ and $\mathbf{Exp}^{\text{ind-keys-lossy}}_{\mathcal{ID},k,i}(\mathsf{D}_i)$ $(i \in \{1, \ldots, T\})$:

$\mathbf{Exp}^{\text{ind-keys-real}}_{\mathcal{ID},k,i}(\mathsf{D}_i)$	$\mathbf{Exp}^{\text{ind-keys-lossy}}_{\mathcal{ID},k,i}(\mathsf{D}_i)$
$(pk, sk_1) \xleftarrow{\$} \mathsf{KG}(1^k, 1^T)$; $sk_{i+1} \xleftarrow{\$} \mathsf{Update}^i(sk_1)$	$(pk, sk_{i+1}) \xleftarrow{\$} \mathsf{LKG}(1^k, 1^T, i)$
return $\mathsf{D}_i(pk, sk_{i+1})$	return $\mathsf{D}_i(pk, sk_{i+1})$

D is said to (t, ε)-solve the key-indistinguishability problem for period i if it runs in time t and $\left|\Pr\left[\mathbf{Exp}^{\text{ind-keys-real}}_{\mathcal{ID},k,i}(\mathsf{D}_i) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-keys-lossy}}_{\mathcal{ID},k,i}(\mathsf{D}_i) = 1\right]\right| \geq \varepsilon$. Furthermore, we say that $\mathcal{ID}$ is (t, ε)-key-indistinguishable, if, for any i, no algorithm (t, ε)-solves the key-indistinguishability problem for period i.

(4) **Lossiness.** Let I_i be an impersonator for period i $(i \in \{1, \ldots, T\})$, st be its state. We consider the experiment $\mathbf{Exp}^{\text{los-imp-pa}}_{\mathcal{ID},k,i}(\mathsf{I}_i)$ played between I_i and a hypothetical challenger:

$\mathbf{Exp}^{\text{los-imp-pa}}_{\mathcal{ID},k,i}(\mathsf{I}_i)$
$(pk, sk_{i+1}) \xleftarrow{\$} \mathsf{LKG}(1^k, 1^T, i)$; $(cmt, st) \xleftarrow{\$} \mathsf{I}_i(pk, sk_{i+1})$; $ch \xleftarrow{\$} \{0,1\}^{\ell_c}$; $rsp \xleftarrow{\$} \mathsf{I}_i(ch, st)$
return $\mathsf{Ver}(pk, cmt, ch, rsp, i)$

I_i is said to ε-solve the impersonation problem with respect to i-lossy public keys if $\Pr\left[\mathbf{Exp}^{\text{los-imp-pa}}_{\mathcal{ID},k,i}(\mathsf{I}_i) = 1\right] \geq \varepsilon$. Furthermore, $\mathcal{ID}$ is said to be ε-lossy if, for any period $i \in \{1, \ldots, T\}$, no (computationally unrestricted) algorithm ε-solves the impersonation problem with respect to i-lossy keys.

We remark that, for $T = 1$, a key-evolving lossy identification scheme becomes a standard lossy identification scheme[1], described in [AFLT12].

Finally, we say that $\mathcal{ID}$ is **response-unique** if for all normal public keys pk or for all lossy keys pk, for all periods $i \in \{1, \ldots, T\}$, for all messages M, for all bit strings cmt[2], and for all challenges ch, there exists at most one response rsp such that $\mathsf{Ver}(pk, cmt, ch, rsp, i) = 1$.

[1] Contrary to the definition of lossiness given in [AFLT12], the impersonator I_1 does not have access to an oracle $\widetilde{\mathsf{Tr}}^{\mathcal{ID}}_{pk,1,k}$ in $\mathbf{Exp}^{\text{los-imp-pa}}_{\mathcal{ID},k,1}(\mathsf{I}_1)$. However, we remark that this has no impact on the security definition as the execution of $\widetilde{\mathsf{Tr}}^{\mathcal{ID}}_{pk,1,k}$ does not require any secret information.

[2] Not necessarily a correctly generated commitment, but any bit string.

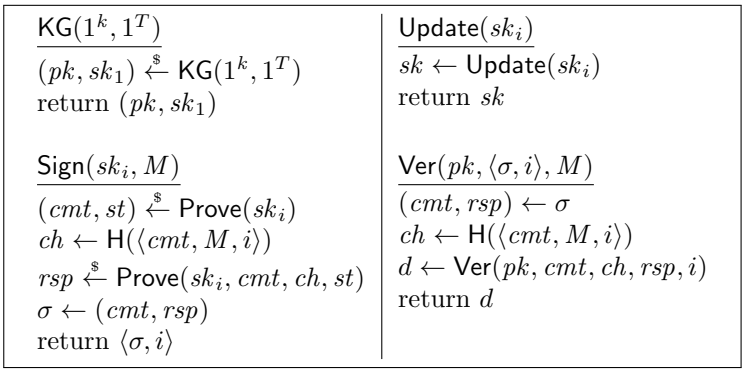

Figure 3.1: Generalized Fiat-Shamir transform for forward-secure signature

3.2 Generalized Fiat-Shamir Transform

The forward-secure signature schemes considered in this paper are built from a key-evolving identification scheme via a straightforward generalization of the Fiat-Shamir transform [FS87], depicted in Figure 3.1. More precisely, the signature for period i is just the signature obtained from a Fiat-Shamir transform with secret key $sk_i = \mathsf{Update}^{i-1}(sk_1)$ (with the period i included in the random oracle input).

Let $\mathcal{FS}[\mathcal{ID}] = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be the signature scheme obtained via this generalized Fiat-Shamir transform. The following theorem is a generalization of (a special case of) Theorem 1 in [AFLT12], where we assume perfect completeness.

Theorem 3.1 *Let $\mathcal{ID} = (\mathsf{KG}, \mathsf{LKG}, \mathsf{Update}, \mathsf{Prove}, \ell_c, \mathsf{Ver})$ be a key-evolving lossy identification scheme whose commitment space has min-entropy at least β (for every period i), let H be a random oracle, and let $\mathcal{FS}[\mathcal{ID}] = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be the signature scheme obtained via the generalized Fiat-Shamir transform. If $\mathcal{ID}$ is ε_s-simulatable, complete, (t', ε_k)-key-indistinguishable, and ε_ℓ-lossy, then $\mathcal{FS}[\mathcal{ID}]$ is $(t, q_h, q_s, \varepsilon)$-existentially-forward-secure in the random oracle model for:*

$$\varepsilon = T\ (\varepsilon_k + (q_h + 1)\varepsilon_\ell) + q_s\varepsilon_s + (q_h + 1)q_s/2^\beta$$
$$t \approx t' - (q_s\, t_{\mathsf{Sim-Sign}} + (T - 1)\, t_{\mathsf{Update}})$$

where $t_{\mathsf{Sim-Sign}}$ denotes the average time of a query to the simulated transcript function $\widetilde{\mathsf{Tr}}^{\mathcal{ID}}_{pk,i,k}$ and t_{Update} denotes the average time of a query to Update. *Furthermore, if $\mathcal{ID}$ is response-unique, $\mathcal{FS}[\mathcal{ID}]$ is also $(t, q_h, q_s, \varepsilon)$-forward-secure.*

Actually, if we choose $T = 1$ in the previous theorem, we get a slightly improved special case of Theorem 1 in [AFLT12], since the forward security for $T = 1$ is exactly the strong unforgeability for a signature scheme. The proof of this theorem can be found in Appendix C and is very similar to the proof in [AFLT12], except that we need to guess the period i^* of the signature output by the adversary, in order to choose the correct lossy key. That is why we lose a factor T in the reduction.

Remark 3.2 As in the standard Fiat-Shamir transform, the signature obtained via the generalized transform consists of a commitment-response pair. However, in all schemes proposed in this paper, the commitment can be recovered from the challenge and the response (as in the scheme depicted in Figure G.1). Hence, since the challenge is often shorter than the commitment, it is generally better to use the challenge-response pair as the signature in our schemes. Obviously, this change does not affect the security of our schemes.

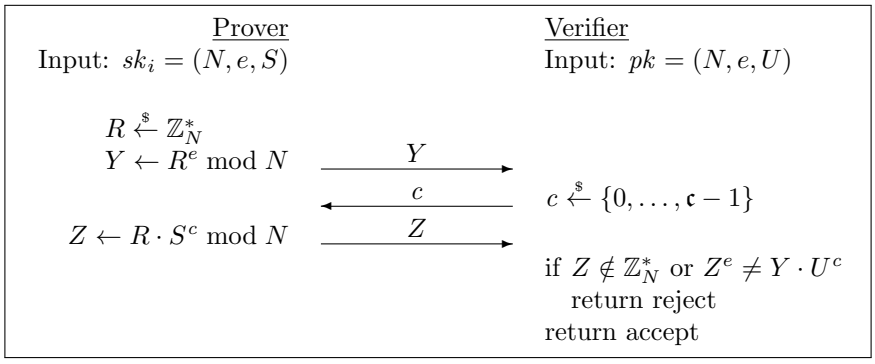

Figure 4.1: Description of the GQ identification scheme ($U = S^e \bmod N$).

4 Tighter Security Reductions for Guillou-Quisquater-like Schemes

In this section, we prove tighter security reductions for the Guillou-Quisquater scheme (GQ, [GQ90]) and for a slight variant of the Itkis-Reyzin scheme (IR, [IR01]), which can also be seen as a forward-secure extension of the GQ scheme. We analyze the practical performance of this new scheme in the next section of this article. Detailed proofs for these schemes are available in Appendix G.2.1[3].

4.1 Guillou-Quisquater Scheme

Let us describe the identification scheme corresponding to the GQ signature scheme, before presenting our tight reduction and comparing it with the swap method.

Scheme. Let N be a product of two distinct ℓ_N-bit primes p_1, p_2 and let e be a ℓ_e-bit prime, coprime with $\phi(N) = (p_1 - 1)(p_2 - 1)$, chosen uniformly at random. Let S be an element chosen uniformly at random in $\mathbb{Z}_N^*$ and let $U = S^e \bmod N$. Let $\mathfrak{c} = 2^{\ell_e}$. The public key is $pk = (N, e, U)$ and the secret key is $sk = (N, e, S)$.

The goal of the identification scheme is to prove U is a e-residue. The identification scheme is depicted in Figure 4.1 and works as follows. First, the prover chooses a random element $R \in \mathbb{Z}_N^*$, computes $Y \leftarrow R^e \bmod N$. It sends Y to the verifier, which in turn chooses $c \in \{0, \ldots, \mathfrak{c} - 1\}$ and returns it to the prover. Upon receiving c, the prover computes $Z \leftarrow R \cdot S^c \bmod N$ and sends this value to the verifier. Finally, the verifier checks whether $Z \in \mathbb{Z}_N^*$ and $Z^e = Y \cdot U^c$ and accepts only in this case[4].

Security. The previous proofs of the GQ schemes looses a factor q_h in the reduction. In this paragraph, we prove the previously described identification scheme $I\mathcal{D}$ is a lossy identification scheme, under the ϕ-hiding assumption. This yields a security proof of the strong unforgeability of the GQ scheme, with a tight reduction to this assumption.

The algorithm LKG chooses e and $N = p_1 p_2$ such that e divides $p_1 - 1$, instead of being coprime with $\phi(N)$, and chooses U uniformly at random among the non-e-residue modulo N. Proposition D.13 and Proposition D.14 show that if U is chosen uniformly at random in $\mathbb{Z}_N^*$, it is not an e-residue with probability $1 - 1/e$ and that it is possible to efficiently check whether U is an e-residue or not if the factorization of N is known: U is a e-residue if and only if, for any $k \in \{1, 2\}$, e does not divide $p_k - 1$ or $U^{(p_k-1)/e} = 1 \bmod p_k$.

The proof that $I\mathcal{D}$ is **complete** follows immediately from the fact that, if $U = S^e \bmod N$, an honest execution of the protocol will always result in acceptance as $Z^e = (R \cdot S^c)^e = R^e \cdot (S^e)^c = Y \cdot U^c$.

The **simulatability** of $I\mathcal{D}$ follows from the fact that, given $pk = (N, e, U)$, we can easily generate transcripts whose distribution is perfectly indistinguishable from the transcripts output by an honest

[3]However the proofs in this appendix use our generic factoring-based scheme described in Section 6.

[4]The test $Z \in \mathbb{Z}_N^*$ can be replaced by the less expensive test $Z \bmod N \neq 0$, as explained in Appendix G.1.

execution of the protocol. This is done by choosing Z uniformly at random in $\mathbb{Z}_N^*$ and c uniformly at random in $\{0, \ldots, \mathfrak{c} - 1\}$, and setting $Y = Z^e / U^c$.

Let us prove the **key indistinguishability**. The distribution of normal public keys is indistinguishable from the one where e divides $\phi(N)$ and U is chosen uniformly at random, according to the ϕ-hiding assumption. And in this latter distribution, U is not a e-residue with probability $1 - 1/e$, so this distribution is statistically close to the distribution of lossy keys. Therefore, $\mathcal{ID}$ is key indistinguishable.

To show that $\mathcal{ID}$ is **lossy**, we note that, when the public key is lossy, for every element Y chosen by the adversary, there exists only one value of $c \in \{0, \ldots, \mathfrak{c} - 1\}$ for which there exists a response Z which is considered valid by the verifier. To see why, assume for the sake of contradiction that there exist two different values c_1 and c_2 in $\{0, \ldots, \mathfrak{c} - 1\}$ for which there exists a valid response. Denote by Z_1 and Z_2 one of the valid responses in each case. Without loss of generality, assume that $c_1 < c_2$. Since $Z_1^e = Y \cdot U^{c_1}$ and $Z_2^e = Y \cdot U^{c_2}$, we have that $(Z_2/Z_1)^e = U^{c_2 - c_1}$. As $c_2 - c_1$ is a positive number smaller than 2^{ℓ_e}, it is coprime with e (since e is a prime and $e \geq 2^{\ell_e}$). Therefore, according to Bezout theorem, there exists two integers u, v such that: $ue + v(c_1 - c_2) = 1$. So:

$$U = U^{ue + v(c_1 - c_2)} = (U^u)^e (U^{c_2 - c_1})^v = (U^u (Z_2/Z_1)^v)^e$$

and U is a e-residue, which is impossible. This means that the probability that a valid response Z_i exists in the case where U is not a e-residue is at most $1/\mathfrak{c}$. It follows that $\mathcal{ID}$ is $1/\mathfrak{c}$-lossy.

Comparison with the swap method. Applying the swap method [MR02] to the GQ identification scheme can also provide a signature with a tight reduction, to the RSA problem. However, in this case, the signing algorithm needs to compute the e-root of the output of the random oracle modulo N. Therefore, instead of requiring two exponentiation modulo N with a ℓ_e-bit exponent, the signing algorithm requires one such exponentiation and one exponentiation modulo N with a ℓ_N-bit exponent. And our signing algorithm will be $\ell_N/(2\ell_e)$ faster, for the same parameters and the same security level, if we consider the ϕ-hiding problem is as hard as the RSA problem. Furthermore, the swap method cannot be directly extended to the forward-secure extension of the GQ scheme, described in the next section, because the prover has to know the factorization of N.

A slight variant of the scheme. We can also chooses e uniformly at random among the ℓ_e-bit primes (without forcing that e is coprime with $\phi(N)$ in KG), because, with high probability, such a prime number will be coprime with $\phi(N)$.

4.2 Variant of the Itkis-Reyzin Scheme

Scheme. The idea of this forward-secure extension of the GQ scheme consists in using a different e for each period. More precisely, let $e_1, \ldots, e_T$ be T distinct ℓ_e-bit primes chosen uniformly at random. Let $f_i = e_{i+1} \ldots e_T$, $f_T = 1$ and $E = e_1 \ldots e_T$. Let S be an element chosen uniformly at random in $\mathbb{Z}_N^*$ and let $U = S^E \bmod N$. Let $S_i = S^{E/e_i}$ and $S_i' = S^{E/f_i}$. Then the public key is $pk = (N, e_1, \ldots, e_T, U)$ and the secret key for period i is $sk_i = (N, e_i, \ldots, e_T, S_i, S_i')$. We remark we can easily compute sk_{i+1} from sk_i, since $S_{i+1} = {S'}_i^{f_{i+1}} \bmod N$ and ${S'}_{i+1} = {S'}_i^{e_{i+1}} \bmod N$.

For period i, the identification scheme works exactly as the previous one with public key $pk = (N, e_i, U)$ and secret key $sk = (N, e_i, S_i)$.

For the sake of simplicity, in this naive description of the scheme, we store the exponents $e_1, \ldots, e_T$ in the public key and in the secret key. Therefore, the keys are linear in T, the number of periods. It is possible to have constant-size key, either by using fixed exponents, or by computing the exponents using a random oracle. This will be discussed in Section 5.1.

Security. The security proof is similar to the one for the previous scheme, with the main difference being the description of the lossy key generation algorithm LKG. More precisely, on input $(1^k, 1^T, i)$, the algorithm LKG generates e_i and $N = p_1 p_2$ such that e_i divides $p_1 - 1$, instead of being coprime with $\phi(N)$,

Table 5.1: Choice of parameters

k	q_h	q_s	ℓ_e	ε_p	ℓ_N
80	2^{80}	2^{30}	123	2^{-80}	≥ 1248
128	2^{128}	2^{46}	171	2^{-128}	≥ 3248

and chooses U' uniformly at random among the non-e_i-residues modulo N. Then it chooses $T-1$ distinct random ℓ_e-bit primes $e_1, \ldots, e_{i-1}, e_{i+1}, \ldots, e_T$, and sets $U = U'^{e_{i+1} \cdots e_T} \bmod N$, $S_{i+1} = U'^{e_{i+2} \cdots e_T} \bmod N$ and $S'_{i+1} = U'^{e_{i+1}} \bmod N$. The public key is $pk = (N, e_1, \ldots, e_T, U)$ and the secret key for period $i+1$ is $sk_{i+1} = (N, e_{i+1}, \ldots, e_T, S_{i+1}, S'_{i+1})$ (or $\perp$ if $i = T$). We remark that, since U' is a non-e_i-residue, U is also a non-e_i-residue and so the public key pk is i-lossy.

5 Analysis of our Variant of the Itkis-Reyzin Scheme

In this section, we analyze our variant of the IR scheme and compare it with the original IR scheme [IR01] and the MMM scheme [MMM02].

5.1 Computation of the exponents $e_1, \ldots, e_T$

As explained before, storing the exponents $e_1, \ldots, e_T$ in the keys is not a good idea since the key size becomes linear in T. Since we need $e_1, \ldots, e_T$ to be random primes to be able to do the reduction of key indistinguishability to the ϕ-hiding assumption, we can use a second random oracle H' which outputs prime numbers of length ℓ_e, and set $e_i = \mathsf{H}'(i)$.

An implementation of a random oracle for prime numbers using a classical random oracle is presented in Appendix I.2. The construction is close to the construction of a PRF mapping to prime numbers in [HW09]. The idea is to hash the input value concatenated to a counter and to increment the counter until we get a prime number. One can prove that it behaves like a random oracle uniform over all primes, and that we can program it efficiently (property which is needed for the security reductions).

We finally remark that, we can always store e_i in the secret key for period i. The secret key length is increased only by a small amount and the signing algorithm becomes faster, since it does not need to recompute e_i.

5.2 Choice of Parameters

In order to be able to compare the original IR scheme with our scheme, we need to choose various parameters. In Table 5.1, we show our choice of parameters for two security levels: $k = 80$ bits and $k = 128$ bits. When choosing these parameters, we considered a value of $T = 2^{20}$, as it enables to update the key every hour for up to 120 years. In both cases, ε_p denotes the maximum error probability of the probabilistic primality test used in the random oracle for primes numbers H', whereas q_h and q_s specify the maximum number of queries to the random oracle and to the signing oracle, respectively, in the forward-security game.

Let us explain our choice for ℓ_e. As in [MR02], we supposed $T\varepsilon, \delta \geq 2^{-20} \approx 10^{-6}$ (we use $T\varepsilon$ instead of ε because of the way weak security notions are defined, see Remark B.4 to understand this choice. And we chose $\ell_e \approx k + 43$ to satisfy inequalities in security reductions in Theorem G.2 and Theorem I.1[5]. In the sequel, all the parameters are fixed except the length ℓ_N of the modulus.

5.3 Comparison with Existing Schemes

Comparison with the Itkis-Reyzin scheme. In this section, we compare the original IR scheme

[5]The constant 43 comes from $-\log_2 \varepsilon - \log_2 \delta + 3$ (for our scheme) and $-\log_2 \varepsilon + \log_2 T + 3$ (for the original IR scheme).

Table 5.2: Time of verification algorithm (using parameters of Table 5.1)

		exponentiation		prime generation		verification orig.[a]		verification new[b]	
k	ℓ_N	mul.[c]	ms[d]	mul.[c]	ms[d]	mul.[c]	ms[d]	mul.[c]	ms[d]
k	ℓ_N	$\frac{3}{2}\,\ell_e\,{\ell_N}^2$	n/a	$(\frac{3}{2}\,kp + 2\,\ell_e)\,{\ell_e}^3$	n/a	$3\,\ell_e\,{\ell_N}^2$	n/a	$3\,\ell_e\,{\ell_N}^2 + (\frac{3}{2}\,kp + 2\,\ell_e)\,{\ell_e}^3$	n/a
80	1248	$0.29 \cdot 10^9$	0.15	$0.68 \cdot 10^9$	0.26	$0.58 \cdot 10^9$	0.30	$1.26 \cdot 10^9$	0.56
80	1920	$0.68 \cdot 10^9$	0.34	$0.68 \cdot 10^9$	0.26	$1.36 \cdot 10^9$	**0.68**	$2.04 \cdot 10^9$	**0.94**
80	6848	$8.65 \cdot 10^9$	3.09	$0.68 \cdot 10^9$	0.26	$17.3 \cdot 10^9$	**6.18**	$1.26 \cdot 10^9$	6.44
128	3248	$2.71 \cdot 10^9$	1.19	$2.67 \cdot 10^9$	0.82	$5.42 \cdot 10^9$	2.38	$8.09 \cdot 10^9$	3.10

[a] verification time of the original scheme (also equal to the signature time for both schemes), estimated using the time of the two exponentiations.

[b] verification time of our scheme, estimated using the time of the two exponentiations and of the prime generation.

[c] approximate theoretical complexity (see Appendix I.2).

[d] time on an Intel Core i5 750 (2.67 GHz), using GMP version 5.0.4 (`http://gmplib.org`, a pseudo-random number generator is used as a random oracle.

without optimization with our scheme (in which e_i is stored in the secret key sk_i, as in the IR scheme). The original IR scheme is very close to our scheme. The only differences are that the IR scheme requires that the factors p_1 and p_2 of the modulus N are safe primes[6] and that IR signatures for period i contain the used exponent e_i. Therefore the IR verification algorithm does not need to recompute the exponent, and is faster. In order to prevent an adversary from using an exponent for the breakin period to sign messages for an older period, the exponent has to be in a different set for each period. The security of the scheme comes from the strong-RSA assumption. Unfortunately, we cannot use such an optimization with our security reduction for our scheme, because we need to know which exponent the adversary will use to make the key lossy for this exponent. However, we remark in Appendix I.3 that the other optimizations of the original IR scheme can also be applied to our scheme.

Let us now compare the two schemes with the same security parameters (k, ℓ_e, ℓ_N), before analyzing the exact security. We first remark that for the same security parameters, our key generation algorithm is slightly faster since it does not require safe primes; and our signing and key update algorithms are as fast as the IR ones. The key and signature lengths of the signatures are nearly the same as the IR ones (IR signatures are only ℓ_e-bits longer than our signatures). The real difference is the verification time since our verification algorithm needs to recompute the e_i, contrary to the IR scheme. Verification consists of two exponentiations (modulo N with a ℓ_e-bit exponent) for the original scheme and two exponentiations and an evaluation of the random prime oracle (roughly equivalent to a random prime generation) for our scheme.

Let us now focus on the exact security of the two schemes. As explained by Kakvi and Kiltz in [KK12], the best known attacks against the ϕ-hiding problems are the factorization of N. Let us also consider it is true for the strong RSA problem (since it just strengthens our result if it is not the case). According to Section G.3.1 and Section I.1, with our choice of parameters, if we want $k = 80$ bits of security, we need to choose a modulo length ℓ_N such that the factorization is $k + \log_2(T) = 100$-bit hard (for our scheme) and $k + \log_2(Tq_h) = 180$-bit hard (for the original scheme). This corresponds to about $\ell_N \approx 1920$ and $\ell_N \approx 6848$ respectively, according to Ecrypt II [ECR11]. In this case, according to Table 5.2, our verification algorithm is about 6 times faster (0.94ms vs 6.18ms) and our signing algorithm is about 9 times faster (0.68ms vs 6.18ms). And our scheme generates 3.5 times shorter signatures.

Comparison with the MMM scheme. The MMM scheme [MMM02] is one of the most efficient generic

[6] A safe prime p is an odd prime such that $(p-1)/2$ is also prime. This assumption is needed for the security reduction of the IR scheme.

constructions of forward-secure signatures (from any signature scheme), to the best of our knowledge. Furthermore, it does not require to fix the number of periods T. However, in the security proof, we have to bound the number of periods T the adversary can use (as query for the oracles **Sign** and **Breakin**). Its forward security can be reduced to the strong unforgeability of the underlying signature scheme with a loss of a factor T.

If we want to compare the MMM scheme with our variant of the IR scheme, the fairest solution is to instantiate the MMM scheme with the GQ scheme. Then we can use our tight reduction of the GQ scheme to the ϕ-hiding problem, to prove that the resulting MMM scheme is forward-secure with a relatively tight (losing only a factor T) reduction to the ϕ-hiding problem. In this setting, the MMM scheme and our scheme have approximatively the same proven security. And the comparison of the MMM scheme with our scheme is roughly the same as the comparison in [MMM02] between the IR scheme and the MMM scheme (which did not take into account the tightness of the reduction).

Very roughly, we can say that the MMM key generation and key update algorithms are faster (about T times faster). However, MMM private keys are longer. And, even if MMM public keys are shorter (more than 30 times for $k = 80, \ell_N = 1248$), in most cases, it is not really useful since signatures with the MMM scheme are about four times longer than signatures with our scheme ($4\ell_N + (\log(k) + \log T)k$ compared to $\ell_N + k$), and also about twice as long as the sum of the length of a public key of our scheme and a signature. Therefore, since the public key is used for verification, the total memory needed to store input data needed for the verification of a signature with the MMM scheme is still twice the amount of the one needed with our scheme. Furthermore, our scheme outperforms the MMM scheme with respect to verification time (considering Table 5.2, since the MMM verification algorithm verifies two classical GQ signatures). This means that, if verification time, signing time, and signature size are critical (for example, if verification or signing has to be performed on a smartcard), our scheme is better than the MMM scheme. And, even more generally, if key updates are not performed often and if T can be bounded by a reasonable constant (for example, if keys are updated each day and are expected to last 3 years, $T = 2^{10}$, and key update time is not really a problem), our scheme is also better than the MMM scheme.

6 Generic Factoring-Based Forward-Secure Signature Scheme

In this section, we show that all our previous results on the GQ scheme and its forward-secure extension can be generalized and applied to several other schemes. To do so, we first introduce a new generic factoring-based key-evolving lossy identification scheme and then show that several factoring-based signature and forward-secure signature schemes can be seen as simple instantiations of this generic scheme.

6.1 Generic Factoring-Based Forward-Secure Signature Scheme

Let ℓ be a security parameter, let N be a product of large primes, and let $e_1, \ldots, e_T$ be T integers and E be the least common multiple of $e_1, \ldots, e_T$. Let $S_1, \ldots, S_\ell$ be a set of elements in ${\mathbb{Z}_N^*}^\ell$ and let $U_1, \ldots, U_\ell \in {\mathbb{Z}_N^*}^\ell$ be the set of elements containing the corresponding E-powers. That is, for each $j \in \{1, \ldots, \ell\}$, $U_j = S_j^E \bmod N$. The public key is $pk = (N, e_1, \ldots, e_T, U_1, \ldots, U_\ell)$ (as for our variant of the IR scheme, we can use a random oracle to avoid storing the exponents in the keys, as explained in Section 5.1). Let f_i be the least common multiple of $e_{i+1}, \ldots, e_T$ for each $i \in \{1, \ldots, T\}$ ($f_T = 1$) and let $S_{j,i} = S_j^{E/e_i}$ and $S'_{j,i} = S_j^{E/f_i}$, for each $1 \leq i \leq T$ and each $1 \leq j \leq \ell$. Then, the secret key for period $1 \leq i \leq T$ is $sk_i = (i, N, e_i, \ldots, e_T, S_{1,i}, \ldots, S_{\ell,i}, S'_{1,i}, \ldots, S'_{\ell,i})$. We remark that it is possible to compute sk_{i+1} from sk_i by computing: $S_{j,i+1} = {S'_{j,i}}^{f_i/e_{i+1}} \bmod N$ and $S'_{j,i+1} = {S'_{j,i}}^{f_i/f_{i+1}} \bmod N$.

The identification scheme is depicted in Figure 6.1 and is a straightforward extension of the one of our variant of the IR scheme in Section 4.2. For period i, its goal is to prove that the elements $U_1, \ldots, U_\ell$ are all e_i-residues, and works as follows. First, the prover chooses an element $R_j \in \mathbb{Z}_N^*$ and computes $Y_j \leftarrow R_j^{e_i} \bmod N$, for $j \in \{1, \ldots, \ell\}$. It then sends $Y_1, \ldots, Y_\ell$ to the verifier, which in turn chooses

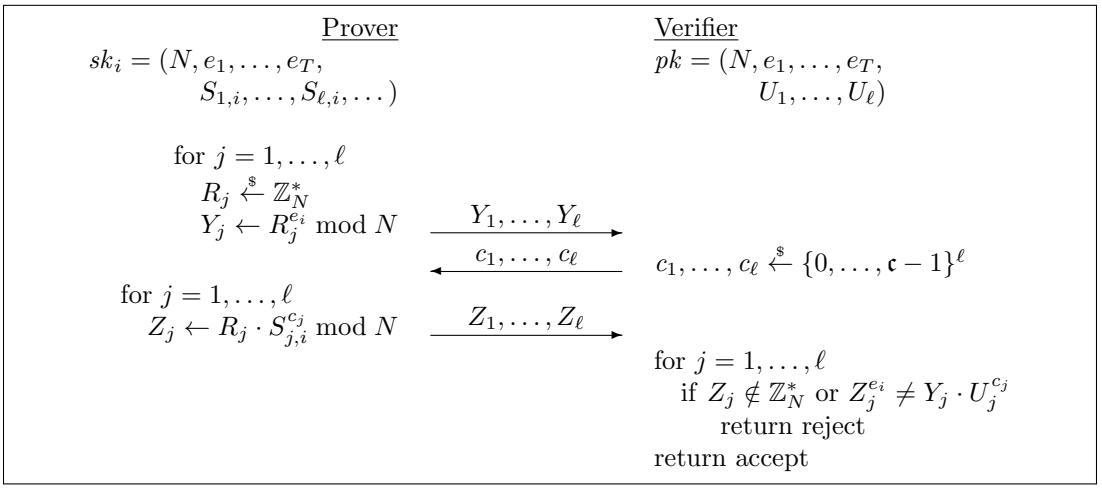

Figure 6.1: Description of the generic identification scheme $I\mathcal{D}$ for proving that the elements $U_1, \ldots, U_\ell$ in pk are all e_i-residues (for each $j \in \{1, \ldots, \ell\}$, $U_j = S_{j,i}^e \bmod N$).

$c_1, \ldots, c_\ell \in \{0, \ldots, \mathfrak{c} - 1\}^\ell$ and returns it to the prover. Upon receiving $c_1, \ldots, c_\ell$, the prover computes $Z_j \leftarrow R_j \cdot S_{j,i}^{c_j} \bmod N$ for $j \in \{1, \ldots, \ell\}$ and sends these values to the verifier. Finally, the verifier checks whether $Z_j \in \mathbb{Z}_N^*$ and $Z_j^{e_i} = Y_j \cdot U_j^{c_j}$ for $j \in \{1, \ldots, \ell\}$ and accepts only if this is the case. The corresponding factoring-based forward-secure signature scheme is depicted in Figure 6.2.

In Appendix E, we prove that the previous scheme is existentially forward-secure, under the following condition:

Condition 6.1 *There exists a normal key generation algorithm* KG *and a lossy key generation algorithm* LKG *which takes as input the security parameter and the period* i *and outputs a pair* (pk, sk'_{i+1}) *such that, for every* $i \in \{1, \ldots, T\}$*:*

- (pk, sk'_{i+1}) *is indistinguishable from a pair* (pk, sk_{i+1}) *generated by* KG *and* i *calls to* Update *(to get* sk_{i+1} *from* sk_1*);*
- *for all* $c \in \{0, \ldots, \mathfrak{c} - 1\}$*, none of* $U_1, \ldots, U_\ell$ *is a* $e'(e, c, N)$*-residue, where* $e'(e, c, N)$ *is:*

$$e'(e, c, N) = \gcd_{i \in \{1, \ldots, m\}} \frac{e \wedge (p_i^{k_i} - p_i^{k_i - 1})}{c \wedge e \wedge (p_i^{k_i} - p_i^{k_i - 1})} e'_i,$$

with $N = p_1^{k_1} \ldots p_m^{k_m}$ *the prime decomposition of* N *and* e'_i *the greatest divisor of* e *coprime with* $p_i^{k_i} - p_i^{k_i - 1}$*, and where* $a \wedge b$ *is the greatest common divisor (gcd) of* a *and* b*.*

The second part of the condition ensures that the scheme is $1/\mathfrak{c}^\ell$-lossy.

6.2 Some Instantiations

In addition to the GQ scheme and our variant of the IR scheme, there are other possible instantiations of our generic scheme.

Quadratic-Residuosity-Based Signature Scheme. The case where $e = \mathfrak{c} = 2$ and $T = 1$ is an important instantiation of the generic scheme as it coincides with the quadratic-residuosity-based scheme informally suggested by Katz and Wang in [KW03]. This scheme is existentially unforgeable based on the hardness of the quadratic-residuosity problem as long as ℓ is large enough to make the term $q_h/2^\ell$ negligible.

2^t-Root Signature Scheme by Ong and Schnorr. The case where $e = \mathfrak{c} = 2^t$, $\ell = 1$, and $T = 1$ coincides with the 2^t-root identification scheme by Ong and Schnorr [OS90]. If $N = p_1 p_2$ is an RSA

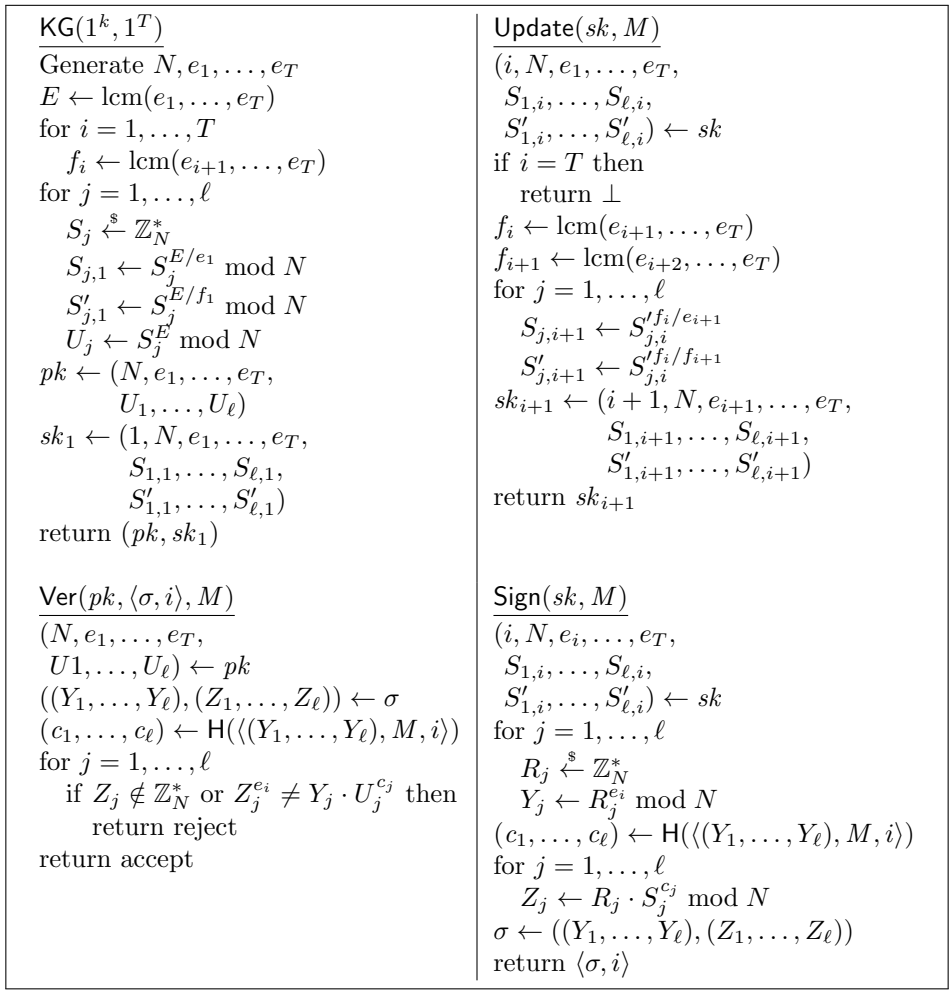

Figure 6.2: Factoring-based forward-secure signature scheme

modulus such that 2^t divides $p_1 - 1$ and $p_2 - 1$, this scheme is existentially unforgeable based on the hardness of the strong-2^t-residuosity problem as long as t is large enough to make the term $q_h/2^t$ negligible.

Paillier Signature Scheme. The case where $\ell = 1$, $T = 1$, and $e = p_1 p_2$ is an RSA modulus, $N = e^2 = p_1^2 p_2^2$ and $\mathfrak{c} \leq \min(p_1, p_2)$ coincides with the Paillier signature scheme [Pai99]. This scheme is existentially unforgeable based on the hardness of the high-residuosity problem of [Pai99].

2^t-Root Forward-Secure Signature Scheme. The case in which $e_i = 2^{t(T-i+1)}$ with t a positive integer and $\mathfrak{c} = 2^i$ is a generalization of the quadratic-residuosity-based scheme and the 2^t-root scheme. In this case, $f_i = e_i$, and we do not need to store $S'_{1,i}$. If $N = p_1 p_2$ is an RSA modulus such that 2^{tT} divides $p_1 - 1$ and $p_2 - 1$, this scheme is existentially forward-secure based on the hardness of a variant of the strong-2^{tT}-assumption, as long as the exponents t and ℓ are large enough to make the term $q_h/2^{t\ell}$ negligible. Although this scheme appears to be new, it is of limited interest as its public key and secret key sizes are linear in the number T of time periods.

Proof details for the above instantiations can be found in Appendix G.

Acknowledgments

We would like to thank Mihir Bellare and Eike Kiltz for their helpful comments on a preliminary version of this paper and the anonymous referees of PKC 2013 for their valuable input.

This work was supported in part by the French ANR-10-SEGI-015 PRINCE Project and in part by the

European Commission through the FP7-ICT-2011-EU-Brazil Program under Contract 288349 SecFuNet and the ICT Program under Contract ICT-2007-216676 ECRYPT II.

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 Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Berlin, Germany. (Cited on page 1.)

[AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 572–590, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany. (Cited on pages 1, 4, 5, and 6.)

[BM99] Mihir Bellare and Sara K. Miner. A forward-secure digital signature scheme. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 431–448, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Berlin, Germany. (Cited on pages 3, 4, and 20.)

[BNN07] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Unrestricted aggregate signatures. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *ICALP 2007*, volume 4596 of *LNCS*, pages 411–422, Wroclaw, Poland, July 9–13, 2007. Springer, Berlin, Germany. (Cited on page 22.)

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. (Cited on page 2.)

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416, Saragossa, Spain, May 12–16, 1996. Springer, Berlin, Germany. (Cited on page 22.)

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Berlin, Germany. (Cited on page 17.)

[BS96] Eric Bach and Jeffrey Shallit. *Algorithmic Number Theory*. MIT Press, August 1996. (Cited on pages 18 and 42.)

[CD95] Ronald Cramer and Ivan Damgård. Escure signature schemes based on interactive protocols. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 297–310, Santa Barbara, CA, USA, August 27–31, 1995. Springer, Berlin, Germany. (Cited on page 37.)

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 402–414, Prague, Czech Republic, May 2–6, 1999. Springer, Berlin, Germany. (Cited on pages 1, 3, and 17.)

[Dus98] P. Dusart. Autour de la fonction qui compte le nombre de nombres premiers. *These, Université de Limoges*, page 36, 1998. (Cited on page 31.)

[ECR11] ECRYPT II yearly report on algorithms and keysizes, 2011. (Cited on page 10.)

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988. (Cited on page 1.)

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Berlin, Germany. (Cited on pages 1 and 6.)

[GBL08] Sanjam Garg, Raghav Bhaskar, and Satyanarayana V. Lokam. Improved bounds on security reductions for discrete log based signatures. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 93–107, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany. (Cited on page 1.)

[GMR85] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A "paradoxical" solution to the signature problem (abstract) (impromptu talk). In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, page 467, Santa Barbara, CA, USA, August 19–23, 1985. Springer, Berlin, Germany. (Cited on page 3.)

[GQ90] Louis C. Guillou and Jean-Jacques Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 216–231, Santa Barbara, CA, USA, August 21–25, 1990. Springer, Berlin, Germany. (Cited on pages 1, 7, and 33.)

[HW09] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 654–670, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany. (Cited on pages 9 and 41.)

[IN96] Russell Impagliazzo and Moni Naor. Efficient cryptographic schemes provably as secure as subset sum. *Journal of Cryptology*, 9(4):199–216, 1996. (Cited on page 1.)

[IR01] Gene Itkis and Leonid Reyzin. Forward-secure signatures with optimal signing and verifying. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 332–354, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany. (Cited on pages 2, 7, 9, 18, 40, and 42.)

[KK12] Saqib A. Kakvi and Eike Kiltz. Optimal security proofs for full domain hash, revisited. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 537–553, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany. (Cited on page 10.)

[KOS10] Eike Kiltz, Adam O'Neill, and Adam Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 295–313, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Berlin, Germany. (Cited on page 17.)

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM CCS 03*, pages 155–164, Washington D.C., USA, October 27–30, 2003. ACM Press. (Cited on pages 1, 12, and 35.)

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact Knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 144–155, Venice, Italy, July 10–14, 2006. Springer, Berlin, Germany. (Cited on page 1.)

[Mic94] Silvio Micali. A secure and efficient digital signature algorithm. Technical Memo MIT/LCS/TM-501b, Massachusetts Institute of Technology, Laboratory for Computer Science, April 1994. (Cited on page 1.)

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Berlin, Germany. (Cited on page 1.)

[MMM02] Tal Malkin, Daniele Micciancio, and Sara K. Miner. Efficient generic forward-secure signatures with an unbounded number of time periods. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 400–417, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Berlin, Germany. (Cited on pages 2, 9, 10, and 11.)

[MR99] Silvio Micali and Leonid Reyzin. Improving the exact security of digital signature schemes. In Rainer Baumgart, editor, *CQRE'99*, volume 1740 of *LNCS*, pages 167–182, Düsseldorf, Germany, November 30 – December 2, 1999. Springer, Berlin, Germany. (Cited on page 16.)

[MR02] Silvio Micali and Leonid Reyzin. Improving the exact security of digital signature schemes. *Journal of Cryptology*, 15(1):1–18, 2002. Full version of [MR99]. (Cited on pages 1, 8, 9, 19, 20, 35, 37, 38, 39, and 40.)

[OO90] Kazuo Ohta and Tatsuaki Okamoto. A modification of the Fiat-Shamir scheme. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 232–243, Santa Barbara, CA, USA, August 21–25, 1990. Springer, Berlin, Germany. (Cited on page 1.)

[OS90] H. Ong and Claus-Peter Schnorr. Fast signature generation with a Fiat-Shamir-like scheme. In Ivan Damgård, editor, *EUROCRYPT'90*, volume 473 of *LNCS*, pages 432–440, Aarhus, Denmark, May 21–24, 1990. Springer, Berlin, Germany. (Cited on pages 1, 12, and 35.)

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238, Prague, Czech Republic, May 2–6, 1999. Springer, Berlin, Germany. (Cited on pages 1, 3, 13, 18, and 36.)

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 145–166, New York, NY, USA, March 4–7, 2006. Springer, Berlin, Germany. (Cited on page 1.)

[PS98] Sarvar Patel and Ganapathy S. Sundaram. An efficient discrete log pseudo random generator. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 304–317, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Berlin, Germany. (Cited on page 1.)

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000. (Cited on page 37.)

[PV05] Pascal Paillier and Damien Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 1–20, Chennai, India, December 4–8, 2005. Springer, Berlin, Germany. (Cited on page 1.)

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards (abstract) (rump session). In Jean-Jacques Quisquater and Joos Vandewalle, editors, *EUROCRYPT'89*, volume 434 of *LNCS*, pages 688–689, Houthalen, Belgium, April 10–13, 1990. Springer, Berlin, Germany. (Cited on page 1.)

[Seu12] Yannick Seurin. On the exact security of schnorr-type signatures in the random oracle model. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 554–571, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany. (Cited on page 1.)

[vW96] Paul C. van Oorschot and Michael J. Wiener. On Diffie-Hellman key agreement with short exponents. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 332–343, Saragossa, Spain, May 12–16, 1996. Springer, Berlin, Germany. (Cited on page 1.)

A Notations

A.1 Games

The definitions and proofs in this paper use code-based game-playing [BR06]. In such games, there exist procedures for initialization (**Initialize**) and finalization (**Finalize**) and procedures to respond to adversary oracle queries. A game G is executed with an adversary $\mathcal{A}$ as follows. First, **Initialize** executes and its outputs are the inputs to $\mathcal{A}$. Then $\mathcal{A}$ executes, its oracle queries being answered by the corresponding procedures of G. When $\mathcal{A}$ terminates, its output becomes the input to the **Finalize** procedure. The output of the latter, denoted $\mathrm{G}(\mathcal{A})$, is called the output of the game, and "$\mathrm{G}(\mathcal{A}) \Rightarrow y$" denotes the event that the output takes a value y. The running time of an adversary is the worst case time of the execution of the adversary with the game defining its security, so that the execution time of the called game procedures is included.

A.2 Computational Assumptions

ϕ-HIDING ASSUMPTION [CMS99, KOS10]. To define ϕ-hiding assumption more formally, we adopt the notation and formalization given in [KOS10]. Let k be a security parameter, let ℓ_N be a function of k and let RSA_{ℓ_N} denote the set of all tuples (N, p_1, p_2) such that $N = p_1 p_2$ is a ℓ_N-bit number which is the product of two distinct $\ell_N/2$-bit primes. We call such a number an RSA modulo. As in [KOS10], we denote by $(N, p_1, p_2) \xleftarrow{\$} \mathsf{RSA}_{\ell_N}$ the process of sampling (N, p_1, p_2) according to the uniform distribution on RSA_{ℓ_N}. Likewise, let R be a relation on p_1 and p_2. We denote by $\mathsf{RSA}_{\ell_N}[R]$ the subset of RSA_{ℓ_N} for which the relation R holds on p_1 and p_2 and by $(N, p_1, p_2) \xleftarrow{\$} \mathsf{RSA}_{\ell_N}[R]$ the process of sampling (N, p_1, p_2) according to the uniform distribution on $\mathsf{RSA}_{\ell_N}[R]$. Consider for instance the example given in [KOS10] where e is a prime and the relation is $p_1 = 1 \bmod e$. Then, $\mathsf{RSA}_{\ell_N}[p_1 = 1 \bmod e]$ is the set of all (N, p_1, p_2) for which $N = p_1 p_2$ is the product of two $\ell_N/2$-bit primes p_1 and p_2 with $p_1 = 1 \bmod e$. That means that the relation $R(p_1, p_2)$ is true for values of p_1 such that $p_1 = 1 \bmod e$ and for arbitrary values of p_2. Finally, we denote by $\mathsf{KG}_{\mathrm{rsa}}$ and $\mathsf{KG}_{\mathrm{rsa}}[R]$ an algorithm which samples (N, p_1, p_2) according to the uniform distribution on RSA_{ℓ_N} and $\mathsf{RSA}_{\ell_N}[R]$, respectively.

Let c be a public positive constant smaller than $1/4$ and denote by $e \xleftarrow{\$} \mathbb{P}_{c\ell_N}$ the process of sampling a prime e according to the uniform distribution on $\mathbb{P}_{c\ell_N}$, the set of primes of length $c\ell_N$. Let $\mathbf{Exp}_k^{\phi\text{H-}0}(\mathcal{A})$ denote the game in which the procedure **Initialize** samples $e \xleftarrow{\$} \mathbb{P}_{c\ell_N}$ and $(N, p_1, p_2) \xleftarrow{\$} \mathsf{RSA}_{\ell_N}[\gcd(e, \phi(N)) = 1]$ and returns (N, e) to the adversary $\mathcal{A}$ ($\phi(N)$ is the order of $\mathbb{Z}_N^*$). Let $\mathbf{Exp}_k^{\phi\text{H-}1}(\mathcal{A})$ denote the game in which the procedure **Initialize** samples $e \xleftarrow{\$} \mathbb{P}_{c\ell_N}$ and $(N, p_1, p_2) \xleftarrow{\$} \mathsf{RSA}_{\ell_N}[p_1 = 1 \bmod e]$ and returns (N, e) to $\mathcal{A}$. In both games, the adversary $\mathcal{A}$ eventually queries the procedure **Finalize** with a guess β, which becomes the output of the game. The advantage $\mathbf{Adv}_k^{\phi\mathrm{H}}(\mathcal{A})$ of an adversary $\mathcal{A}$ in solving the ϕ-hiding problem is then defined as the probability that $\mathbf{Exp}_k^{\phi\text{H-}0}(\mathcal{A})$ outputs 1 minus the probability that $\mathbf{Exp}_k^{\phi\text{H-}1}(\mathcal{A})$ outputs 1. We say that the ϕ-hiding problem is (t, ε)-hard if for all adversary $\mathcal{A}$ with running time at most t, $\mathbf{Adv}_k^{\phi\mathrm{H}}(\mathcal{A})$ is at most ε.

We remark that the procedure **Initialize** in $\mathbf{Exp}_{\mathcal{A}}^{\phi\text{H-0}}(k)$ and $\mathbf{Exp}_{\mathcal{A}}^{\phi\text{H-0}}(k)$ can be implemented efficiently if we assume the widely-accepted Extended Riemann Hypothesis (Conjecture 8.4.4 of [BS96]).

e-RESIDUOSITY. Let N be an RSA modulus and let e is an integer that divides $\phi(N)$. Let $\mathsf{HR}_N[e]$ denote the set of all e-residues modulo N. That is, $\mathsf{HR}_N[e] = \{g^e \text{ s.t. } g \in \mathbb{Z}_N^*\}$. Let $\mathsf{J}_N[e]$ be a subgroup of $\mathbb{Z}_N^*$ including $\mathsf{HR}_N[e]$ whose membership can be efficiently checked and for which we expect the problem of testing whether an element $x \in \mathsf{HR}_N[e]$ or $x \in \mathsf{J}_N[e] \setminus \mathsf{HR}_N[e]$ is hard. We refer to $\mathsf{J}_N[e] \setminus \mathsf{HR}_N[e]$ as the set of pseudo-e-residues modulo N. For instance, in the case where $e = 2$, we can define $\mathsf{J}_N[2]$ as the set of elements in $\mathbb{Z}_N^*$ with Jacobi symbol 1. The latter case is known as the **quadratic-residuosity** problem. Likewise, in the extended case where $e = p_1p_2$ and $N = p_1^2p_2^2$ for large primes p_1 and p_2, we have that $\mathsf{J}_N[e] = \mathbb{Z}_N^*$. The latter case was introduced by Paillier in [Pai99] and the problem of deciding whether $x \in \mathsf{HR}_N[e]$ or $x \in \mathsf{J}_N[e] \setminus \mathsf{HR}_N[e]$ when given (N, e, x) is known as the **high-residuosity** problem. Since the exact way in which N is generated will depend on the specific e-residuosity assumption on which we are relying, we assume that there exists an efficient parameter generation algorithm KG which on input k outputs (N, e). We remark that if e is a constant, generating random a random RSA modulus N such that e divides $\phi(N)$ can be done efficiently (in time linear in e), if we assume the widely-accepted Extended Riemann Hypothesis (Conjecture 8.4.4 of [BS96]).

To define the e-residuosity problem more precisely, where e is an integer that divides $\phi(N)$, we will define games $\mathbf{Exp}_k^{\text{hr-0}}(\mathcal{A})$ and $\mathbf{Exp}_k^{\text{hr-1}}(\mathcal{A})$. In game $\mathbf{Exp}_k^{\text{hr-0}}(\mathcal{A})$, the procedure **Initialize** samples $(N, e) \xleftarrow{\$} \mathsf{KG}(k)$, chooses $x \in \mathbb{Z}_N^*$ uniformly at random, computes $y = x^e \bmod N$, and returns (N, e, y) to the adversary $\mathcal{A}$. Clearly, we have $y \in \mathsf{HR}_N[e]$ in this case. In game $\mathbf{Exp}_k^{\text{hr-1}}(\mathcal{A})$, the procedure **Initialize** samples $(N, e) \xleftarrow{\$} \mathsf{KG}(k)$, chooses $y \in \mathsf{J}_N[e] \setminus \mathsf{HR}_N[e]$ uniformly at random, and returns (N, e, y) to $\mathcal{A}$. In both games, the adversary $\mathcal{A}$ eventually queries the procedure **Finalize** with a guess β, which becomes the output of the game. The advantage $\mathbf{Adv}_k^{\text{hr}}(\mathcal{A})$ of an adversary $\mathcal{A}$ in solving the e-residuosity problem is then defined as the probability that $\mathbf{Exp}_k^{\text{hr-0}}(\mathcal{A})$ outputs 1 minus the probability that $\mathbf{Exp}_k^{\text{hr-1}}(\mathcal{A})$ outputs 1. We say that the e-residuosity problem is (t, ε)-hard if for all adversary $\mathcal{A}$ with running time at most t, $\mathbf{Adv}_k^{\text{hr}}(\mathcal{A})$ is at most ε.

STRONG-2^t-RESIDUOSITY. This assumption is a slightly stronger assumption than the 2^t-residuosity because we force 2^t to divide $p_1 - 1$ and $p_2 - 1$ (and not only $\phi(N)$), and we want to distinguish a 2^t-residue from a non 2-residue (instead of a non 2^t-residue). However, we expect that, in practice, to solve the problem, the best algorithm is still roughly as efficient as factorizing N, as for all the other complexity assumptions made in this article.

To define the 2^t-residuosity problem more precisely, we will define two games: $\mathbf{Exp}_k^{\text{hr-0}}(\mathcal{A})$ and $\mathbf{Exp}_k^{\text{hr-1}}(\mathcal{A})$. Let $e = 2^t$. In game $\mathbf{Exp}_k^{\text{shr-0}}(\mathcal{A})$, the procedure **Initialize** samples $(N, p_1, p_2) \xleftarrow{\$} \mathsf{RSA}_{\ell_N}[p_1 = 1 \bmod e \wedge p_2 = 1 \bmod e]$, chooses $x \in \mathbb{Z}_N^*$ uniformly at random, computes $y = x^e \bmod N$, and returns (N, e, y) to the adversary $\mathcal{A}$. Clearly, we have $y \in \mathsf{HR}_N[e]$ in this case. In game $\mathbf{Exp}_k^{\text{shr-1}}(\mathcal{A})$, the procedure **Initialize** samples $(N, p_1, p_2) \xleftarrow{\$} \mathsf{RSA}_{\ell_N}[p_1 = 1 \bmod e \wedge p_2 = 1 \bmod e]$, chooses $y \in \mathsf{J}_N[2] \setminus \mathsf{HR}_N[2]$ uniformly at random, and returns (N, e, y) to $\mathcal{A}$. In both games, the adversary $\mathcal{A}$ eventually queries the procedure **Finalize** with a guess β, which becomes the output of the game. The advantage $\mathbf{Adv}_k^{\text{shr}}(\mathcal{A})$ of an adversary $\mathcal{A}$ in solving the strong-2^t-residuosity problem is then defined as the probability that $\mathbf{Exp}_k^{\text{shr-0}}(\mathcal{A})$ outputs 1 minus the probability that $\mathbf{Exp}_k^{\text{shr-1}}(\mathcal{A})$ outputs 1. We say that the strong-2^t-residuosity problem is (t, ε)-hard if for all adversary $\mathcal{A}$ with running time at most t, $\mathbf{Adv}_k^{\text{shr}}(\mathcal{A})$ is at most ε.

STRONG-RSA. We use the variant of the strong-RSA assumption described in [IR01]. More precisely, we fix a security parameter ℓ_e. Let $\mathbf{Exp}_k^{\text{SRSA}}(\mathcal{A})$ denote the game in which the procedure **Initialize** samples $(N, p_1, p_2) \xleftarrow{\$} \mathsf{RSA}_{\ell_N}[p_1 \text{ and } p_2 \text{ are safe}^7]$ and $y \xleftarrow{\$} \mathbb{Z}_N^*$ and returns (N, y) to the adversary $\mathcal{A}$. In the game, the adversary $\mathcal{A}$ eventually queries the procedure **Finalize** with a pair (x, e). $\mathbf{Exp}_k^{\text{SRSA}}(\mathcal{A})$ outputs

[7] A prime number p is safe if it can be written as $p = 2q + 1$ with q a prime number.

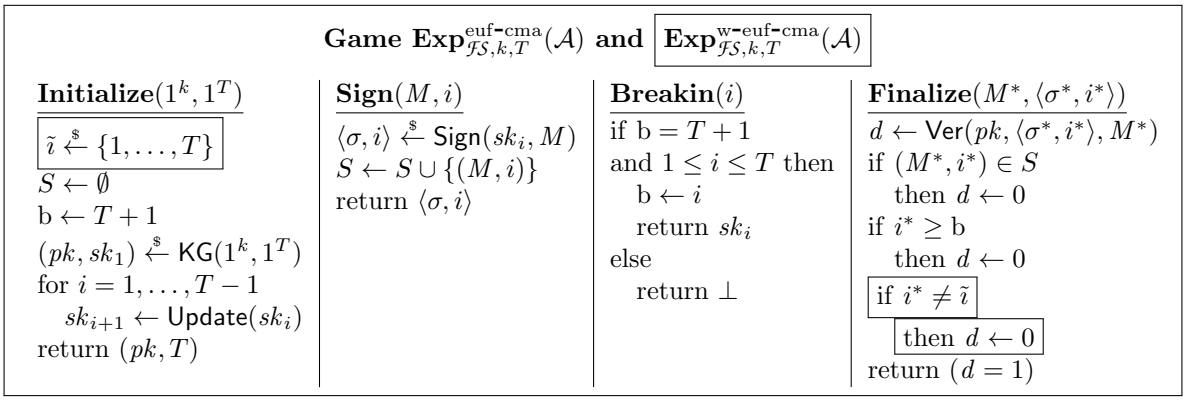

Figure A.1: Game $\mathbf{Exp}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ (and $\mathbf{Exp}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$) defining the EUF-CMA (and W-EUF-CMA) security of a key-evolving signature scheme $\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$. $\mathbf{Exp}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ includes the boxed codes in **Initialize** and in **Finalize** but $\mathbf{Exp}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ does not.

1 if and only if $e \le 2^{\ell_e}$ and $y = x^e \bmod N$. The advantage $\mathbf{Adv}^{\text{SRSA}}_k(\mathcal{A})$ of an adversary $\mathcal{A}$ in solving the strong-RSA problem is then defined as the probability that $\mathbf{Exp}^{\text{SRSA}}_k(\mathcal{A})$ outputs 1. We say that the strong-RSA problem is (t, ε)-hard if for all adversary $\mathcal{A}$ with running time at most t, $\mathbf{Adv}^{\text{SRSA}}_k(\mathcal{A})$ is at most ε.

A.3 Weak Security Notions

In addition to classical security notions defined in Section 2.3, we consider two even weaker notions: the **weak forward security** and the **weak existential forward security**. For these notions, the challenger of the adversary, picks a period $\tilde{\imath}$ uniformly at random at the beginning and reject the forged signature if it does not correspond to the period $\tilde{\imath}$. Then we say that a key-evolving signature scheme is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-(existentially)-forward-secure if there is no adversary (running in time at most t, doing at most q_h requests to the random oracle, and q_s requests of signatures), such that, with probability at least δ, the challenger chooses a period $\tilde{\imath}$ and a key pair (pk, sk_1), such that the adversary forges a correct signature for period $\tilde{\imath}$ with probability at least ε. The idea of this definition is to distinguish the probability from the choice of the period and of the key pair. This is actually a (not so straightforward) extension of the security definition of Micali and Reyzin in [MR02].

A.4 Formal Security Notions

This section gives the formal notions of security corresponding to the informal ones, introduced in Section 2.3 and Section A.3.

The precise definitions of existential forward security (EUF-CMA) and (strong) forward security (SUF-CMA) consider the games $\mathbf{Exp}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ and $\mathbf{Exp}^{\text{suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$, respectively, described in Figure A.1 and Figure A.2. $\mathbf{Exp}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ and $\mathbf{Exp}^{\text{suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ contain four procedures, which are executed with an adversary $\mathcal{A}$ as follows. The procedure **Initialize** generates a pair of public and secret keys $(pk, sk_1) \xleftarrow{\$} \mathsf{KG}(1^k, 1^T)$ and all the secret keys sk_i for $2 \le i \le T$, where sk_i is the secret key for time period i and T is the total number of time periods, and returns (pk, T) to $\mathcal{A}$. During the execution of the game, the adversary is allowed to make queries (M, i) to the **Sign** procedure, where $M \in \mathcal{M}$, and i is the time period of the requested signature. To answer it, the game $\mathbf{Exp}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ generates a signature $\sigma \xleftarrow{\$} \mathsf{Sign}(sk_i, M)$ and gives (σ, i) to $\mathcal{A}$. The adversary is also allowed to make one query i to the **Breakin** procedure, which returns the secret key sk_i of the period i. Eventually, the adversary ends the game by querying the **Finalize** procedure with a message-signature-period triple $(M^*, \langle \sigma^*, i^* \rangle)$. The advantage $\mathbf{Adv}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ ($\mathbf{Adv}^{\text{suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$) of the adversary $\mathcal{A}$ in breaking the EUF-CMA (SUF-CMA)

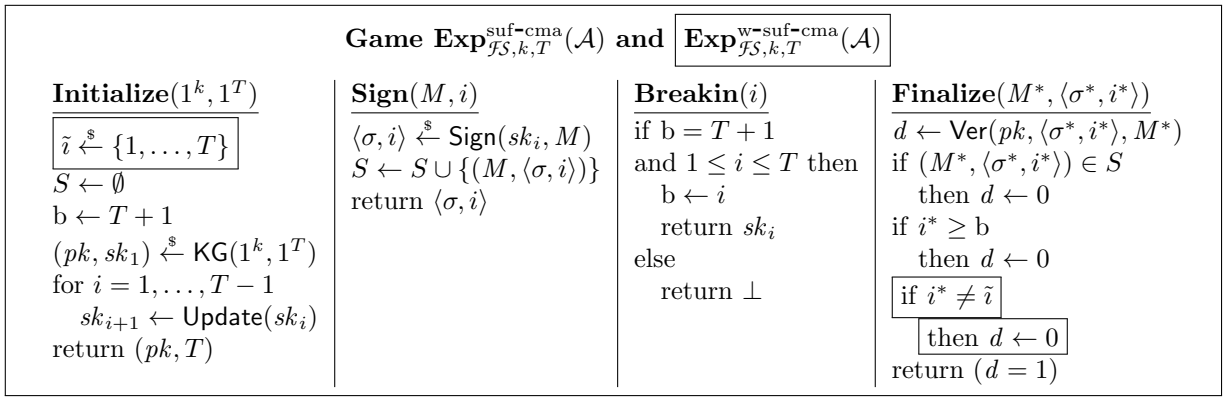

Figure A.2: Game $\mathbf{Exp}^{\text{suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ (and $\mathbf{Exp}^{\text{w-suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$) defining the SUF-CMA (and W-SUF-CMA) security of a key-evolving signature scheme $\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$. $\mathbf{Exp}^{\text{w-suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ includes the boxed code in **Initialize** and in **Finalize** but $\mathbf{Exp}^{\text{suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ does not.

security of $\mathcal{FS}$ is then defined as the probability that the game $\mathbf{Exp}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ ($\mathbf{Exp}^{\text{suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$) outputs true. If $\mathbf{Adv}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A}) \geq \varepsilon$, we say that $\mathcal{A}$ $(t, q_h, q_s, \varepsilon)$-breaks the existential forward security of $\mathcal{FS}$. If $\mathbf{Adv}^{\text{suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A}) \geq \varepsilon$, we say that $\mathcal{A}$ $(t, q_h, q_s, \varepsilon)$-breaks the forward security of $\mathcal{FS}$. Finally, we say that $\mathcal{FS}$ is $(t, q_h, q_s, \varepsilon)$-(existentially)-forward-secure, if no adversary $\mathcal{A}$ $(t, q_h, q_s, \varepsilon)$-breaks the (existential) forward security of $\mathcal{FS}$.

We remark that our definition of forward security is slightly stronger than the one proposed by [BM99]. Indeed, the adversary can choose the breakin period after seeing signatures for following periods. The advantage of this definition is that the game is somehow simpler than the one with the definition of Bellare and Miner in [BM99]. Anyway, it seems that most of the current schemes (maybe even all of them) also verify this stronger notion.

The precise definitions of weak existential forward security (W-EUF-CMA) and weak forward security (W-SUF-CMA) consider the games $\mathbf{Exp}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ and $\mathbf{Exp}^{\text{w-suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ described in Figure A.1 and Figure A.2. The only difference with the games $\mathbf{Exp}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ and $\mathbf{Exp}^{\text{suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ is that **Initialize** picks a random period $\tilde{\imath} \in \{1, \ldots, T\}$ and **Finalize** rejects any signature which period i^* is not $\tilde{\imath}$. An adversary $\mathcal{A}$ $(t, q_h, q_s, \varepsilon, \delta)$-weakly-breaks the (existential) forward security of $\mathcal{FS}$ if, with probability at least δ, **Initialize** picks $(pk, sk_1, \tilde{\imath})$, such that $\mathbf{Adv}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A}) \geq \varepsilon$ (for W-EUF-CMA and $\mathbf{Exp}^{\text{w-suf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ for W-SUF-CMA) for this $(pk, sk_1, \tilde{\imath})$, $\mathcal{A}$ runs in time at most t and does at most q_h queries to the random oracle and at most q_s queries to the signature oracle. We then say that $\mathcal{FS}$ is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-(existentially)-forward-secure if no adversary $(t, q_h, q_s, \varepsilon, \delta)$-weakly-breaks the (existential) forward security of $\mathcal{FS}$.

This notion is weaker than the previous one in the following way: if a scheme is $(t, q_h, q_s, T\varepsilon\delta)$-forward-secure, then it is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure, as proven in Appendix B. More details on relations between these two security notions can be found in Appendix B.

B Relations Between Security Notions

Micali and Reyzin introduces the $(t, q_h, q_s, \varepsilon, \delta)$-weak-security notion for signature in [MR02] but without explaining its relation with the standard $(t, q_h, q_s, \varepsilon)$-security notion. We generalize this weak notion to forward security in Section 2.3 (W-SUF-CMA and W-EUF-CMA). And, in this appendix, we present some propositions to try to understand the relation between the two notions. These propositions apply indifferently to weak forward security (W-SUF-CMA) and weak existential forward security (W-EUF-CMA). Therefore, for sake of clarity, we only write the proposition for weak forward security.

Firstly, we have the following straightforward proposition:

Proposition B.1 *Let $\varepsilon, \delta \in [0,1]^2$, such that $T\varepsilon\delta < 1$. A $(t, q_h, q_s, T\varepsilon\delta)$-forward-secure scheme is also $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure.*

Proof: Let $\mathcal{A}$ be an adversary which $(t, q_h, q_s, \varepsilon, \delta)$-weakly-breaks the scheme. Then, it is clear from the definitions that $\mathbf{Adv}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A}) \geq \varepsilon\delta$. But we also have $\mathbf{Adv}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A}) = \frac{1}{T}\mathbf{Adv}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$. Therefore $\mathbf{Adv}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A}) \geq T\varepsilon\delta$, and $\mathcal{A}$ $(t, q_h, q_s, T\varepsilon\delta)$-breaks the scheme. ▮

Unfortunately, the converse is not necessarily true. Let $\varepsilon, \delta, \eta \in]0,1]^3$ and $q_h \geq 1$. Let suppose there exists a $(t, q_h, q_s, 0)$-secure scheme[8] $\mathcal{FS}$. Then we can construct a $(t, q_h, q_s, \varepsilon, \delta)$-weakly-secure scheme $\mathcal{FS}'$ which is not $(t, q_h, q_s, (1-\eta)\delta)$-secure. The scheme $\mathcal{FS}'$ be the same as $\mathcal{FS}$ except that the key generation algorithm KG includes the secret key sk_1 in the public key pk with probability $\delta(1-\eta)$. There exists an adversary which $(t, q_h, q_s, (1-\eta)\delta)$-breaks $\mathcal{FS}'$, but no adversary can $(t, q_h, q_s, \varepsilon, \delta)$-weakly-breaks $\mathcal{FS}'$, since only a proportion $\delta(1-\eta)$ of the keys are breakable. A corollary of this is that if you only prove that a scheme is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-secure, no matter how small is ε, if $\delta' < \delta$, the scheme is not necessarily (t, q_h, q_s, δ')-secure.

We can also construct a $(t, q_h, q_s, \varepsilon, \delta)$-weakly-secure scheme $\mathcal{FS}'$ which is not $(t, q_h, q_s, (1-\eta)\varepsilon)$-secure, if there exists a scheme $\mathcal{FS}$ for which the best adversary $\mathcal{A}$ wins the game $\mathbf{Exp}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ with probability $(1-\eta)\varepsilon$ (independently of the choice of the period i, and of the key pair (pk, sk_1))[9]. A corollary of this is that if you only prove that a scheme is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-secure, no matter how small is δ, if $\varepsilon' < \varepsilon$, the scheme is not necessarily $(t, q_h, q_s, \varepsilon')$-secure.

After all these negative results, let us now present a positive result.

Proposition B.2 *Let suppose there are λ different key pairs (pk, sk_1, i), and that KG choose uniformly at random one of them[10]. Let $\mathcal{FS}$ be a key-evolving scheme. Let $\varepsilon' \in]0,1[$ and $\alpha = 1 + \log(\lambda T)$.*

If $\mathcal{FS}$ is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure for any $\varepsilon, \delta \in]0,1]^2$ such that $\varepsilon\delta \geq \varepsilon'/(\alpha T)$, then $\mathcal{FS}$ is $(t, q_h, q_s, \varepsilon')$-forward-secure.

Proof: Let $\mathcal{A}$ be an adversary which $(t, q_h, q_s, \varepsilon')$-breaks the scheme. Then we have $\mathbf{Adv}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A}) \geq \varepsilon'$, and as in the previous proof, we also have

$$\mathbf{Adv}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A}) = \mathbf{Adv}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})/T \geq \varepsilon'/T.$$

Let us consider the advantages $\mathbf{Adv}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ for each triple (pk, sk_1, i). Let us sort them in decreasing order and write them $\varepsilon'_1 \geq \cdots \geq \varepsilon'_{\lambda T}$. By hypothesis, all triples are equiprobable, therefore, we have

$$\frac{\varepsilon'}{T} \leq \mathbf{Adv}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A}) = \frac{\varepsilon'_1 + \cdots + \varepsilon'_{\lambda T}}{\lambda T}.$$

We remark that since $\varepsilon'_1 \geq \cdots \geq \varepsilon'_{\lambda T}$, if one of the j first triples is used in the game, the advantage $\mathbf{Adv}^{\text{w-euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ is at least ε'_j. Therefore, for any j, $\mathcal{A}$ $(t, q_h, q_s, \varepsilon'_j, j/(\lambda T))$-weakly-breaks $\mathcal{FS}$. So we just need to prove that for some j, $\varepsilon'_j j/(\lambda T) \geq \varepsilon'/(\alpha T)$. Let us suppose for all j, $\varepsilon'_j < (\lambda\varepsilon')/(\alpha j)$, by contrapositive. Then, we can sum these inequalities and we get

$$\lambda T\varepsilon' = \sum_{j=1}^{\lambda T} \varepsilon'_j < \frac{\lambda\varepsilon'}{\alpha} \sum_{j=1}^{\lambda T} \frac{1}{j} < \lambda\varepsilon'$$

(the right inequality comes from the well-known inequality for harmonic series $\sum_{j=1}^{\lambda T} \frac{1}{j} < \alpha$). This is contradictory. So we have proven the proposition. ▮

[8] We could use a $(t, q_h, q_s, \varepsilon')$-secure scheme with ε' small enough but this complicates the proof.

[9] By best, we mean that, for any fixed period i and key pair (pk, sk_1), no adversary can win the game with probability more than $(1-\eta)\varepsilon$.

[10] This is the case with most currently used schemes.

Corollary B.3 *Under the same assumptions as in Proposition B.2, if $\mathcal{FS}$ is $(t, q_h, q_s, \varepsilon'/(\alpha T), \varepsilon'/(\alpha T))$-weakly-forward-secure, then $\mathcal{FS}$ is $(t, q_h, q_s, \varepsilon')$-forward-secure.*

Proof: We just need to remark that if $\varepsilon\delta \geq \varepsilon'/(\alpha T)$, then, since $\delta, \varepsilon \leq 1$, $\varepsilon, \delta \geq \varepsilon'/(\alpha T)$ and so $\mathcal{FS}$ is also $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure. And we apply Proposition B.2. ▮

Remark B.4 However, we may consider that an attacker which can attack only a ridiculously small portion of the keys is quite strange, and do not consider it. In order to be able to compare different schemes, we think it is better to suppose any attacker which $(t, q_h, q_s, \varepsilon)$-breaks the forward security of a scheme also $(t, q_h, q_s, \varepsilon, 1/2)$-breaks it[11]. This means, we suppose that the $(t, q_h, q_s, \varepsilon, 1/2)$-weakly-forward security implies the $(t, q_h, q_s, T\varepsilon)$-forward-security. As you will see in Section 5.3, it enables us to do quite fair comparison, if we consider the security of a $(t, q_h, q_s, T\varepsilon)$-forward-secure scheme is $\log_2(t/(T\varepsilon))$ bits, which is the intuitive notion of security.

C Proof of Theorem 3.1 and of a Slight Variant

C.1 Recall on Code-Based Game-Playing Proofs

Our proof will use code-based game-playing [BR96]. We recall some background here. The boolean flag bad is assumed initialized to false. We say that games $\mathrm{G}_i, \mathrm{G}_j$ are identical until bad if their codes differ only in statements that follow the setting of bad to true. For examples, games $\mathrm{G}_0, \mathrm{G}_1$ of Figure C.1 are identical until bad.

This lemma was stated in [BR96].

Lemma C.1 ([BR96]) *Let $\mathrm{G}_i, \mathrm{G}_j$ be identical until bad games, and $\mathcal{A}$ an adversary. Then*

$$\Pr[\,\mathrm{G}_i(\mathcal{A}) \Rightarrow 1\,] - \Pr[\,\mathrm{G}_j(\mathcal{A}) \Rightarrow 1\,] \leq \Pr[\,\mathrm{G}_i \textit{ sets } \mathsf{bad}\,].$$

The following was stated in [BNN07] and its proof is implicit in [BR96].

Lemma C.2 ([BNN07]) *Let $\mathrm{G}_i, \mathrm{G}_j$ be identical until bad games, and $\mathcal{A}$ an adversary. Let $\mathsf{Good}_i, \mathsf{Good}_j$ be the events that bad is never set in games $\mathrm{G}_i, \mathrm{G}_j$, respectively. Then,*

$$\Pr[\,\mathrm{G}_i(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_i\,] = \Pr[\,\mathrm{G}_j(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_j\,].$$

C.2 Weak Key Indistinguishability

Before proving Theorem 3.1, we need to introduce a new notion: weak key indistinguishability and to prove that this notion is weaker than the key indistinguishability

Consider the experiments $\mathbf{Exp}^{\text{weak-ind-keys-real}}_{I\mathcal{D},k}(\mathsf{D})$ and $\mathbf{Exp}^{\text{weak-ind-keys-lossy}}_{I\mathcal{D},k}(\mathsf{D})$:

$\mathbf{Exp}^{\text{weak-ind-keys-real}}_{I\mathcal{D},k}(\mathsf{D})$	$\mathbf{Exp}^{\text{weak-ind-keys-lossy}}_{I\mathcal{D},k}(\mathsf{D})$
$\tilde{\imath} \xleftarrow{\$} \{1, \ldots, T\}$	$\tilde{\imath} \xleftarrow{\$} \{1, \ldots, T\}$
$(pk, sk_1) \xleftarrow{\$} \mathsf{KG}(1^k, 1^T)$; $sk_{\tilde{\imath}+1} \xleftarrow{\$} \mathsf{Update}^{\tilde{\imath}}(sk_1)$	$(pk, sk_{\tilde{\imath}+1}) \xleftarrow{\$} \mathsf{LKG}(1^k, 1^T, \tilde{\imath})$
return $\mathsf{D}(pk, sk_{\tilde{\imath}+1})$	return $\mathsf{D}(pk, sk_{\tilde{\imath}+1})$

D is said to (t, ε)-solve the weak key indistinguishability problem if it runs in time t and

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

Furthermore, we say that $I\mathcal{D}$ is (t, ε)-weak-key-indistinguishable, if no algorithm (t, ε)-solves the weak-key-indistinguishability problem.

[11] 1/2 is just an arbitrary constant. It can be any reasonable constant.

Lemma C.3 *If $I\mathcal{D}$ is (t,ε)-key-indistinguishable, then, $I\mathcal{D}$ is (t,ε)-weak-key-indistinguishable.*

Proof: Suppose $I\mathcal{D}$ is not (t,ε)-weak-key-indistinguishable. Let D be a distinguisher that (t,ε)-solve the weak key indistinguishability problem.

$$\begin{aligned}
\varepsilon &\leq \left|\Pr\left[\mathbf{Exp}^{\text{weak-ind-keys-real}}_{I\mathcal{D},k}(\mathsf{D})=1\right]-\Pr\left[\mathbf{Exp}^{\text{weak-ind-keys-lossy}}_{I\mathcal{D},k}(\mathsf{D})=1\right]\right| \\
&= \left|\sum_{i=1}^{T}\Pr\left[\mathbf{Exp}^{\text{weak-ind-keys-real}}_{I\mathcal{D},k}(\mathsf{D})=1\wedge i=\tilde{i}\right]-\Pr\left[\mathbf{Exp}^{\text{weak-ind-keys-lossy}}_{I\mathcal{D},k}(\mathsf{D})=1\wedge i=\tilde{i}\right]\right| \\
&= \frac{1}{T}\left|\sum_{i=1}^{T}\Pr\left[\mathbf{Exp}^{\text{weak-ind-keys-real}}_{I\mathcal{D},k}(\mathsf{D})=1\,|\,i=\tilde{i}\right]-\Pr\left[\mathbf{Exp}^{\text{weak-ind-keys-lossy}}_{I\mathcal{D},k}(\mathsf{D})=1\,|\,i=\tilde{i}\right]\right| \\
&= \frac{1}{T}\left|\sum_{i=1}^{T}\Pr\left[\mathbf{Exp}^{\text{ind-keys-real}}_{I\mathcal{D},k,i}(\mathsf{D})=1\right]-\Pr\left[\mathbf{Exp}^{\text{ind-keys-lossy}}_{I\mathcal{D},k,i}(\mathsf{D})=1\right]\right| \\
&\leq \frac{1}{T}\sum_{i=1}^{T}\left|\Pr\left[\mathbf{Exp}^{\text{ind-keys-real}}_{I\mathcal{D},k,i}(\mathsf{D})=1\right]-\Pr\left[\mathbf{Exp}^{\text{ind-keys-lossy}}_{I\mathcal{D},k,i}(\mathsf{D})=1\right]\right|
\end{aligned}$$

Therefore, there exists some $i^*\in\{1,\ldots,T\}$, such that:

$$\varepsilon \leq \left|\Pr\left[\mathbf{Exp}^{\text{ind-keys-real}}_{I\mathcal{D},k,i^*}(\mathsf{D})=1\right]-\Pr\left[\mathbf{Exp}^{\text{ind-keys-lossy}}_{I\mathcal{D},k,i^*}(\mathsf{D})=1\right]\right|$$

otherwise the previous inequality is false.

Then D is a distinguisher that (t,ε)-solve the weak key indistinguishability problem for period i^*, and so $I\mathcal{D}$ is not (t,ε)-key-indistinguishable. ∎

C.3 Proof of Theorem 3.1

Let us suppose there exists an adversary $\mathcal{A}$ which (t,q_h,q_s,ε)-breaks the existential forward security of $\mathcal{FS}$. Let us consider the games $\mathrm{G}_0,\ldots,\mathrm{G}_9$ of Figure C.1 and Figure C.2. The random oracle $\mathbf{H}$ is simulated using a table HT containing all the previous queries to the oracle and responses of the oracle.

Firstly, we will assume that the set of queries to the random oracle made by the adversary always contains the query $\langle cmt^*, M^*\rangle$. This is without loss of generality because, given any adversary, we can always create an adversary (with the same success probability and approximately the same running time) that performs this query before calling **Finalize**. It only increases the total amount of hash queries by 1.

G_0 corresponds to a slightly stronger game than the game $\mathbf{Exp}^{\text{euf-cma}}_{\mathcal{FS},k,T}(\mathcal{A})$ defining the existential forward security of a key-evolving signature built from a key-evolving scheme via generalized Fiat-Shamir transform. We force the forged signature by the adversary to be such that $\langle cmt^*, M^*, i^*\rangle$ (instead of $\langle M^*, i^*\rangle$) is different from all the previous queries to the signing oracle. We have inlined the code of the random oracle in the procedure Sign, and set bad when $\mathbf{H}(\langle cmt, M\rangle)$ is already defined. We have also modified the code of the random oracle $\mathbf{H}$ such that the $\mathrm{fp}^{\mathrm{th}}$ query is answered by ch^*, a random challenge chosen in **Initialize**. These modifications do not change the output of the original game.

To compute the probability $\Pr[\,\mathrm{G}_0 \text{ sets } \mathsf{bad}\,]$, we can assume the worst-case scenario where the q_h hash-queries are made before the q_s signing queries. For each signing query, the probability that there is a collision (i.e., bad is set for this query) is at most $(q_h+1)/2^{\beta}$. By summing over all j, we have

$$\Pr[\,\mathrm{G}_0 \text{ sets } \mathsf{bad}\,] \leq (q_h+1)q_s/2^{\beta}.$$

Initialize(k, T) Game $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \mathrm{G}_3, \mathrm{G}_4$

001 $\mathrm{hc} \leftarrow 1$; $\mathrm{b} \leftarrow T+1$
002 $\mathrm{fp} \xleftarrow{\$} \{1, \ldots, q_h + 1\}$; $ch^* \xleftarrow{\$} \{0,1\}^{\ell_c}$
003 $(pk, sk_1) \xleftarrow{\$} \mathsf{KG}(1^k, 1^T)$
004 for $i = 1, \ldots, T-1$
005 $\quad sk_{i+1} \leftarrow \mathsf{Update}(sk_i)$
006 return (pk, T)

H(x) Game $\mathrm{G}_0, \ldots, \mathrm{G}_9$

011 if $\mathsf{HT}(x) = \perp$ then
012 $\quad \mathsf{QT}(\mathrm{hc}) \leftarrow x$
013 $\quad$ if $\mathrm{hc} \neq \mathrm{fp}$ then
014 $\quad\quad \mathsf{HT}(x) \xleftarrow{\$} \{0,1\}^{\ell_c}$
015 $\quad$ else
016 $\quad\quad \mathsf{HT}(x) \xleftarrow{\$} ch^*$
017 $\mathrm{hc} \leftarrow \mathrm{hc} + 1$
018 return $\mathsf{HT}(x)$

Breakin(i) Game $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \mathrm{G}_3, \mathrm{G}_4$

021 if $\mathrm{b} = T+1$
022 and $1 \leq i \leq T$ then
023 $\quad \mathrm{b} \leftarrow i$
024 $\quad$ return sk_i
025 else
026 $\quad$ return $\perp$

Sign(M, i) Game G_0, $\boxed{\mathrm{G}_1}$

031 $(cmt, st) \xleftarrow{\$} \mathsf{Prove}(sk_i)$
032 if $S(cmt, M, i) \neq \perp$ then
033 $\quad \sigma \leftarrow (cmt, S(cmt, M, i))$
034 $\quad$ return $\langle \sigma, i \rangle$
035 if $\mathsf{HT}(\langle cmt, M, i \rangle) \neq \perp$ then
036 $\quad \mathsf{bad} \leftarrow \mathsf{true}$
037 $\quad \boxed{\mathsf{HT}(\langle cmt, M, i \rangle) \xleftarrow{\$} \{0,1\}^{\ell_c}}$
038 else
039 $\quad \mathsf{HT}(\langle cmt, M, i \rangle) \xleftarrow{\$} \{0,1\}^{\ell_c}$
040 $ch \xleftarrow{\$} \mathsf{HT}(\langle cmt, M, i \rangle)$
041 $rsp \xleftarrow{\$} \mathsf{Prove}(sk_i, cmt, ch, st)$
042 $\sigma \leftarrow (cmt, rsp)$
043 $S(\langle cmt, M, i \rangle) \leftarrow rsp$
044 return $\langle \sigma, i \rangle$

Sign(M, i) Game G_2

231 $(cmt, st) \xleftarrow{\$} \mathsf{Prove}(sk_i)$
232 if $S(cmt, M, i) \neq \perp$ then
233 $\quad \sigma \leftarrow (cmt, S(cmt, M, i))$
234 $\quad$ return $\langle \sigma, i \rangle$
235 $ch \xleftarrow{\$} \{0,1\}^{\ell_c}$
236 $\mathsf{HT}(\langle cmt, M, i \rangle) \leftarrow ch$
237 $rsp \xleftarrow{\$} \mathsf{Prove}(sk_i, cmt, ch, st)$
238 $\sigma \leftarrow (cmt, rsp)$
239 $S(\langle cmt, M, i \rangle) \leftarrow rsp$
240 return $\langle \sigma, i \rangle$

Sign(M, i) Game G_3

331 $(cmt, ch, rsp) \xleftarrow{\$} \mathsf{Tr}^{I\mathcal{D}}_{pk, sk_i, k}$
332 if $S(cmt, M, i) \neq \perp$ then
333 $\quad \sigma \leftarrow (cmt, S(cmt, M, i))$
334 $\quad$ return $\langle \sigma, i \rangle$
335 $\mathsf{HT}(\langle cmt, M, i \rangle) \leftarrow ch$
336 $\sigma \leftarrow (cmt, rsp)$
337 $S(\langle cmt, M, i \rangle) \leftarrow rsp$
338 return $\langle \sigma, i \rangle$

Sign(M, i) Game $\mathrm{G}_4, \ldots, \mathrm{G}_9$

431 $(cmt, ch, rsp) \xleftarrow{\$} \widetilde{\mathsf{Tr}}^{I\mathcal{D}}_{pk, i, k}$
432 if $S(cmt, M, i) \neq \perp$ then
433 $\quad \sigma \leftarrow (cmt, S(cmt, M, i))$
434 $\quad$ return $\langle \sigma, i \rangle$
435 $\mathsf{HT}(\langle cmt, M, i \rangle) \leftarrow ch$
436 $\sigma \leftarrow (cmt, rsp)$
437 $S(\langle cmt, M, i \rangle) \leftarrow rsp$
438 return $\langle \sigma, i \rangle$

Finalize$(M^*, \langle \sigma^*, i^* \rangle)$ Game $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \mathrm{G}_3, \mathrm{G}_4$

051 $d \leftarrow \mathsf{Ver}(pk, \langle \sigma^*, i^* \rangle, M^*)$
052 if $i^* \geq \mathrm{b}$ then
053 $\quad d \leftarrow 0$
054 $(cmt^*, rsp^*) \leftarrow \sigma^*$
055 if $S(\langle cmt^*, M^*, i^* \rangle) \neq \perp$ then
056 $\quad d \leftarrow 0$
057 return d

Figure C.1: Games $\mathrm{G}_0, \ldots, \mathrm{G}_4$ for proof of Theorem 3.1. G_1 includes the boxed code at line 037 but G_0 does not.

Initialize(k, T) Game G_5, G_6, G_7

501 hc $\leftarrow 1$; b $\leftarrow T+1$
502 fp $\xleftarrow{\$} \{1, \ldots, q_h + 1\}$; $ch^* \xleftarrow{\$} \{0,1\}^{\ell_c}$
503 $\tilde{\imath} \xleftarrow{\$} \{1, \ldots, T\}$
504 $(pk, sk_1) \xleftarrow{\$} \mathsf{KG}(1^k, 1^T)$
505 for $i = 1, \ldots, T-1$
506 $\quad sk_{i+1} \leftarrow \mathsf{Update}(sk_i)$
507 return (pk, T)

Initialize(k, T) Game G_8, G_9

801 hc $\leftarrow 1$; b $\leftarrow T+1$
802 fp $\xleftarrow{\$} \{1, \ldots, q_h + 1\}$; $ch^* \xleftarrow{\$} \{0,1\}^{\ell_c}$
803 $\tilde{\imath} \xleftarrow{\$} \{1, \ldots, T\}$
804 $(pk, sk_{\tilde{\imath}+1}) \xleftarrow{\$} \mathsf{LKG}(1^k, 1^T, \tilde{\imath})$
805 for $i = 1, \ldots, \tilde{\imath}$
806 $\quad sk_i \leftarrow \perp$
807 for $i = \tilde{\imath}+1, \ldots, T-1$
808 $\quad sk_{i+1} \leftarrow \mathsf{Update}(sk_i)$
809 return (pk, T)

H(x) Game $G_0, \ldots, G_9$

011 if $\mathsf{HT}(x) = \perp$ then
012 $\quad \mathsf{QT}(\text{hc}) \leftarrow x$
013 $\quad$ if hc $\neq$ fp then
014 $\quad\quad \mathsf{HT}(x) \xleftarrow{\$} \{0,1\}^{\ell_c}$
015 $\quad$ else
016 $\quad\quad \mathsf{HT}(x) \xleftarrow{\$} ch^*$
017 hc $\leftarrow$ hc $+ 1$
018 return $\mathsf{HT}(x)$

Breakin(i) Game $G_5, \boxed{G_6}$

521 if b $= T+1$
522 and $1 \le i \le T$ then
523 $\quad$ b $\leftarrow i$
524 $\quad$ if $i \le \tilde{\imath}$
525 $\quad\quad$ bad $\leftarrow$ true
526 $\quad\quad \boxed{\text{return } \perp}$
527 $\quad$ return sk_i
528 else
529 $\quad$ return $\perp$

Breakin(i) Game G_7, G_8, G_9

721 if b $= T+1$
722 and $1 \le i \le T$ then
723 $\quad$ b $\leftarrow i$
724 $\quad$ if $i \le \tilde{\imath}$
725 $\quad\quad$ return $\perp$
726 $\quad$ return sk_i
727 else
728 $\quad$ return $\perp$

Sign(M, i) Game $G_4, \ldots, G_9$

431 $(cmt, ch, rsp) \xleftarrow{\$} \widetilde{\mathsf{Tr}}^{ID}_{pk,i,k}$
432 if $S(cmt, M, i) \neq \perp$ then
433 $\quad \sigma \leftarrow (cmt, S(cmt, M, i))$
434 $\quad$ return $\langle \sigma, i \rangle$
435 $\mathsf{HT}(\langle cmt, M, i \rangle) \leftarrow ch$
436 $\sigma \leftarrow (cmt, rsp)$
437 $S(\langle cmt, M, i \rangle) \leftarrow rsp$
438 return $\langle \sigma, i \rangle$

Finalize$(M^*, \langle \sigma^*, i^* \rangle)$ Game $G_5, \boxed{G_6}$

551 $d \leftarrow \mathsf{Ver}(pk, \langle \sigma^*, i^* \rangle, M^*)$
552 if $i^* \ge$ b then
553 $\quad d \leftarrow 0$
554 if $i^* \neq \tilde{\imath}$ then
555 $\quad$ bad $\leftarrow$ true
556 $\quad \boxed{d \leftarrow 0}$
557 $(cmt^*, rsp^*) \leftarrow \sigma^*$
558 if $S(\langle cmt^*, M^*, i^* \rangle) \neq \perp$ then
559 $\quad d \leftarrow 0$
560 return d

Finalize$(M^*, \langle \sigma^*, i^* \rangle)$ Game $G_7, G_8, \boxed{G_9}$

751 $d \leftarrow \mathsf{Ver}(pk, \langle \sigma^*, i^* \rangle, M^*)$
752 if $i^* \ge$ b or $i^* \neq \tilde{\imath}$ then
753 $\quad d \leftarrow 0$
754 $(cmt^*, rsp^*) \leftarrow \sigma^*$
755 if $\mathsf{QT}(\text{fp}) \neq (cmt^*, M^*)$ then
756 $\quad$ bad $\leftarrow$ true
757 $\quad \boxed{d \leftarrow 0}$
758 if $S(\langle cmt^*, M^*, i^* \rangle) \neq \perp$ then
759 $\quad d \leftarrow 0$
760 return d

Figure C.2: Games $G_5, \ldots, G_9$ for proof of Theorem 3.1. G_6 includes the boxed code at lines 526 and 556 but G_5 does not; G_9 includes the boxed code at line 757 but G_7 and G_8 do not.

In G_1, when bad is set, a new random value for $\mathbf{H}(\langle cmt, M\rangle)$ is set in **Sign**. Since G_0 and G_1 are identical until bad, thanks to Lemma C.1, we have

$$\Pr[G_0 \Rightarrow 1] - \Pr[G_1 \Rightarrow 1] \leq \Pr[G_0 \text{ sets } \mathsf{bad}] \leq (q_s + q_h)q_s/2^{\beta}$$

In G_2, bad is no more set and the procedure **Sign** is rewritten in an equivalent way. Since the latter does not change the output of the game, we have $\Pr[G_1(\mathcal{A}) \Rightarrow 1] = \Pr[G_2(\mathcal{A}) \Rightarrow 1]$.

In G_3, the procedure **Sign** is changed such that the values (cmt, ch, rsp) are computed using the transcript generation function $\mathsf{Tr}^{I\mathcal{D}}_{pk,sk,k}$. Since the latter does not change the output of the game, we have $\Pr[G_2(\mathcal{A}) \Rightarrow 1] = \Pr[G_3(\mathcal{A}) \Rightarrow 1]$.

In G_4, the q_s calls to the transcript generation function $\mathsf{Tr}^{I\mathcal{D}}_{pk,sk_i,k}$ are replaced by q_s calls to the simulated transcript generation function $\widetilde{\mathsf{Tr}}^{I\mathcal{D}}_{pk,i,k}$. Since the statistical distance between the distributions output by $\mathsf{Tr}^{I\mathcal{D}}_{pk,sk_i,k}$ and by $\widetilde{\mathsf{Tr}}^{I\mathcal{D}}_{pk,i,k}$ is at most ε_s, we have

$$\Pr[G_3(\mathcal{A}) \Rightarrow 1] - \Pr[G_4(\mathcal{A}) \Rightarrow 1] \leq q_s\varepsilon_s.$$

In G_5, a period $\tilde{\imath} \in \{1, \ldots, T\}$ is chosen uniformly at random, and bad is set when the adversary queries **Breakin** (for the first time) with period $\mathrm{b} \leq \tilde{\imath}$ or when the adversary outputs a signature for a period $i^* \neq \tilde{\imath}$. Since if G_5 outputs 1, we have $i^* < \mathrm{b}$, "bad is never set and G_5 outputs 1" (event $G_5(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_5$) if and only if "$i^* = \tilde{\imath}$ and G_5 outputs 1". Therefore, we have:

$$\begin{aligned}
\Pr[G_5(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_5] &= \Pr[G_5(\mathcal{A}) \Rightarrow 1 \wedge i^* = \tilde{\imath}] \\
&= \sum_{i=1}^{T} \Pr[G_5(\mathcal{A}) \Rightarrow 1 \wedge i^* = i \wedge \tilde{\imath} = i] \\
&= \sum_{i=1}^{T} \Pr[G_5(\mathcal{A}) \Rightarrow 1 \wedge i^* = i]\Pr[\tilde{\imath} = i] \\
&= \frac{1}{T}\sum_{i=1}^{T} \Pr[G_5(\mathcal{A}) \Rightarrow 1 \wedge i^* = i] \;=\; \frac{1}{T}\Pr[G_5(\mathcal{A}) \Rightarrow 1].
\end{aligned}$$

So, we have

$$\Pr[G_4(\mathcal{A}) \Rightarrow 1] = \Pr[G_5(\mathcal{A}) \Rightarrow 1] = T\Pr[G_5(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_5].$$

In G_6, the empty string $\perp$ is returned if **Breakin** is queried with period $\mathrm{b} \leq \tilde{\imath}$, and the game outputs 0 if $i^* \neq \tilde{\imath}$. Since G_5 and G_6 are identical until bad, according to Lemma C.2, we have

$$\Pr[G_5(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_5] = \Pr[G_6(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_6] = \Pr[G_6(\mathcal{A}) \Rightarrow 1].$$

In G_7, some procedures have been rewritten in an equivalent way, and bad is now set when the query (cmt^*, M^*) is not the $\mathrm{fp}^{\mathrm{th}}$ query to the random oracle. Since the latter does not change the output of the experiment, we have $\Pr[G_6(\mathcal{A}) \Rightarrow 1] = \Pr[G_7(\mathcal{A}) \Rightarrow 1]$.

In G_8, the key is generated using the lossy key generation algorithm LKG for period $\tilde{\imath}$ instead of the normal key generation algorithm KG. From any adversary $\mathcal{A}$ to G_8, it is straightforward to construct an adversary which (t'', ε'')-solves the weak key indistinguishability problem with $t'' \approx t + (q_s t_{\mathsf{Sim-Sign}} + (T-1)t_{\mathsf{Update}})$ and $\varepsilon'' = |\Pr[G_7(\mathcal{A}) \Rightarrow 1] - \Pr[G_8(\mathcal{A}) \Rightarrow 1]|$. Therefore, thanks to the (t', ε')-key-indistinguishability of $I\mathcal{D}$, and thanks to Lemma C.3, if the adversary runs in time approximately at most $t' - (q_s t_{\mathsf{Sim-Sign}} + (T-1)t_{\mathsf{Update}})$:

$$\Pr[G_7(\mathcal{A}) \Rightarrow 1] - \Pr[G_8(\mathcal{A}) \Rightarrow 1] \leq \varepsilon_k. \tag{C.1}$$

In G_9, the game outputs 0 if the signature does not corresponds to the challenge ch^*. Since we have

$$\Pr[\,G_8(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_8\,] = \Pr[\,G_8(\mathcal{A}) \Rightarrow 1\,] \cdot \Pr[\,\mathsf{QT}(\mathrm{fp}) = (cmt^*, M^*)\,] = \frac{1}{1+q_h}\Pr[\,G_8(\mathcal{A}) \Rightarrow 1\,],$$

and $\Pr[\,G_9(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_9\,] = \Pr[\,G_9(\mathcal{A}) \Rightarrow 1\,]$, according to Lemma C.2, we have

$$\Pr[\,G_8(\mathcal{A}) \Rightarrow 1\,] = (1+q_h)\Pr[\,G_9(\mathcal{A}) \Rightarrow 1\,].$$

From any adversary $\mathcal{A}$ to G_9, it is straightforward to construct an adversary I (not necessarily computationally bounded) which ε'-solves the impersonation problem with $\varepsilon' = \Pr[\,G_9(\mathcal{A}) \Rightarrow 1\,]$. Therefore, we have

$$\Pr[\,G_9(\mathcal{A}) \Rightarrow 1\,] \leq \varepsilon_\ell.$$

From the previous equalities and inequalities, we deduce that, for any adversary $\mathcal{A}$ running in time approximately at most $t' - (q_s t_{\mathsf{Sim-Sign}} + (T-1)t_{\mathsf{Update}})$:

$$\varepsilon \leq \Pr[\,G_0(\mathcal{A})\,] \leq T\ (\varepsilon_k + (1+q_h)\varepsilon_\ell) + q_s\varepsilon_s + (q_h+1)q_s/2^\beta$$

Proof of forward security if ID is response-unique Let us now prove that $\mathcal{FS}$ is strongly forward-secure (with the same parameters) if ID is response-unique. We first remark that, if we replace line 055 of $\mathbb{G}_0$ in Figure C.1 by

if $S(\langle cmt^*, M^*, i^*\rangle) = rsp^*$ then

then we get exactly the game for forward security.

Therefore, if normal keys are response-unique, it is clear that this new game is equivalent to the original game $\mathbb{G}_0$, since if $S(\langle cmt^*, M^*, i^*\rangle)$ is defined, it is the only possible response rsp^*.

If lossy keys are response-unique, to prove forward security, it is sufficient to replace lines 055, 558 and 758 for games $\mathbb{G}_0, \ldots, \mathbb{G}_9$ in Figure C.1 and Figure C.2 by

if $S(\langle cmt^*, M^*, i^*\rangle) = rsp^*$ then

Then the probability the adversary wins the new game $\mathbb{G}_9$ is still bounded by ε_l since if $S(\langle cmt^*, M^*, i^*\rangle)$ is defined, it is the only possible response rsp^*.

C.4 A slight variant

The following theorem is a slight variant of Theorem 3.1 which makes easier the comparison between the various schemes:

Theorem C.4 *Let $ID = (\mathsf{KG}, \mathsf{LKG}, \mathsf{Update}, \mathsf{Prove}, \ell_c, \mathsf{Ver})$ be a key-evolving lossy identification scheme whose commitment space has min-entropy at least β (for every period i), let H be a random oracle, and let $\mathcal{FS}[ID] = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be the signature scheme obtained via the generalized Fiat-Shamir transform. If ID is ε_s-simulatable, complete, (t', ε')-key-indistinguishable, and ε_ℓ-lossy, then $\mathcal{FS}[ID]$ is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-existentially-forward-secure in the random oracle model for:*

$$t \approx \left(t' - (T-1)\,t_{\mathsf{Update}}\right) \cdot \left(\varepsilon - q_s\varepsilon_s - (q_h+1)q_s/2^\beta\right) - q_s t_{\mathsf{Sim-Sign}}$$

as long as

$$\varepsilon > q_s\varepsilon_s + (q_h+1)q_s/2^\beta \text{ and } \varepsilon' \leq \delta\left(1 - \frac{1}{\mathrm{e}}\right) - \frac{(1+q_h)\,\varepsilon_\ell}{\varepsilon - q_s\varepsilon_s - (q_h+1)q_s/2^\beta}$$

where $t_{\mathsf{Sim-Sign}}$ denotes the average time of a query to the simulated transcript function $\widetilde{\mathsf{Tr}}^{ID}_{pk,i,k}$ and t_{Update} denotes the average time of a query to Update. Furthermore, if ID is response-unique, $\mathcal{FS}[ID]$ is also $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure.

And here is a straightforward corollary:

Corollary C.5 *Under the same hypothesis of Theorem C.4, $\mathcal{FS}[\mathcal{ID}]$ is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-existentially-forward-secure in the random oracle model for:*

$$t \approx \frac{(t' - (T-1)\, t_{\mathsf{Update}}) \cdot \varepsilon}{2} - q_s t_{\mathsf{Sim-Sign}}$$

as long as

$$\varepsilon \geq 2\left(q_s\varepsilon_s + (q_h+1)q_s/2^\beta\right) \text{ and } \varepsilon' \leq \delta\left(1 - \frac{1}{\mathrm{e}}\right) - \frac{2\,(1+q_h)\,\varepsilon_\ell}{\varepsilon}.$$

Furthermore, if $\mathcal{ID}$ is response-unique, $\mathcal{FS}[\mathcal{ID}]$ is also $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure.

Proof of Corollary C.5 from Theorem C.4: It is a direct corollary of Theorem C.4. The condition

$$\varepsilon > 2\left(q_s\varepsilon_s + (q_h+1)q_s/2^\beta\right)$$

ensures that

$$\varepsilon - q_s\varepsilon_s - (q_h+1)q_s/2^\beta \geq \varepsilon/2.$$

∎

Proof of Theorem C.4: Let us suppose there exists an adversary $\mathcal{A}$ which $(t, q_h, q_s, \varepsilon, \delta)$-breaks $\mathcal{DS}$. In particular, $\mathcal{A}$ $(t, q_h, q_s, \varepsilon\delta)$-breaks $\mathcal{DS}$.

The proof of Theorem C.4 is very similar to the proof of Theorem 3.1. We use the same games, except for **Initialize** and **Finalize** of games $\mathrm{G}_1, \ldots, \mathrm{G}_5$ which are replaced by the ones of game G_6. Indeed, in the game of the weak-security, a period $\tilde{\imath}$ is chosen in **Initialize** and the adversary has to forge a signature for this period. Then the proof is identical except that

$$\Pr[\,\mathrm{G}_4(\mathcal{A}) \Rightarrow 1\,] = \Pr[\,\mathrm{G}_5(\mathcal{A}) \Rightarrow 1\,] = \Pr[\,\mathrm{G}_6(\mathcal{A}) \Rightarrow 1\,]$$

and except for the inequality of Equation (C.1).

We remark that, if we write $\gamma = \left(q_s\varepsilon_s + (q_h+1)q_s/2^\beta\right)$:

$$\Pr[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1\,] \geq \varepsilon - \gamma \text{ with probability at least } \delta \text{ over } (pk, sk_1, \tilde{\imath}) \tag{C.2}$$

$$\Pr[\,\mathrm{G}_8(\mathcal{A}) \Rightarrow 1\,] \leq (1+q_h)\,\varepsilon_\ell \tag{C.3}$$

Let us construct an adversary $\mathcal{B}$ which (t'', ε'')-breaks the key-indistinguishability with $t'' \approx \frac{t + q_s t_{\mathsf{Sim-Sign}}}{\varepsilon - \gamma} + (T-1)t_{\mathsf{Update}}$ and $\varepsilon'' \geq \delta\left(1 - \frac{1}{\mathrm{e}}\right) - \frac{1}{\varepsilon-\gamma}(1+q_h)\,\varepsilon_\ell$. $\mathcal{B}$ takes as input a period $\tilde{\imath}$, a public key pk and a secret key $sk_{\tilde{\imath}+1}$ for period $\tilde{\imath}+1$. It then runs $\mathcal{A}$ $\frac{1}{\varepsilon-\gamma}$ times and simulates the oracles as in game G_7 (or G_8 which is the same), except for **Initialize** where it uses directly its inputs $\tilde{\imath}$, pk and $sk_{\tilde{\imath}+1}$ (instead of picking them at random). If $\mathcal{A}$ outputs a correct forgery during one of its run, $\mathcal{B}$ outputs 1. Otherwise, it outputs 0.

Clearly $\mathcal{B}$ perfectly simulates the environment of $\mathcal{A}$ in the game G_7, if pk is not lossy, or in the game G_8, if pk is lossy. According to Equation (C.2), if pk is not lossy, we have

$$\begin{aligned}\Pr[\,\mathcal{B} \Rightarrow 1 \,|\, pk \text{ normal}\,] &\geq \delta \Pr[\,\mathcal{B} \Rightarrow 1 \,|\, pk \text{ normal and } \Pr[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1\,] \geq \varepsilon - \gamma\,] \\ &\geq \delta\left(1 - (1 - (\varepsilon-\gamma))^{\frac{1}{\varepsilon-\gamma}}\right) \geq \delta\left(1 - \frac{1}{\mathrm{e}}\right)\end{aligned}$$

and, according to Equation (C.3), if pk is lossy, we have

$$\Pr[\,\mathcal{B} \Rightarrow 1 \,|\, pk \text{ lossy}\,] \leq \frac{1}{\varepsilon-\gamma}\Pr[\,\mathrm{G}_8(\mathcal{A}) \Rightarrow 1\,] \leq \frac{1}{\varepsilon-\gamma}(1+q_h)\,\varepsilon_\ell.$$

Therefore, the advantage of $\mathcal{B}$ is $\varepsilon'' \geq \delta\left(1 - \frac{1}{\mathrm{e}}\right) - \frac{1}{\varepsilon-\gamma}(1+q_h)\,\varepsilon_\ell$. Its running time is $t'' \approx \frac{t + q_s t_{\mathsf{Sim-Sign}}}{\varepsilon - \gamma} + (T-1)t_{\mathsf{Update}}$. This proves the theorem. ∎

D Results on Residues

This appendix presents various results on multiples (i.e., residues for an additive law) in cyclic groups and then uses them to prove results on residues of $\mathbb{Z}_N^*$, with $N \geq 3$ an odd number.

D.1 Multiples in Cyclic Groups

Let n be an integer greater or equal to 2. Let a be an element of $\mathbb{Z}_n$.

D.1.1 Definition

Definition D.1 Let α be a positive integer. a is a **α-multiple** (modulo n) if and only if there exists $b \in \mathbb{Z}_n$, such that $a = \alpha b$.

D.1.2 Characterization of α-Multiples in $\mathbb{Z}_n$

Let $\gcd(u, v) = u \wedge v$ be the greatest common divisor of two integers u and v.

Remark D.2 If β is an integer which divides n, β divides ($a \bmod n$) if and only if it divides any $(a + ln)$ (l an integer). In this case, we say that β divides a.

Theorem D.3 *Let α be a positive integer. a is an α-multiple if and only if* $\gcd(\alpha, n)$ *divides a.*

Proof: If a is an α-multiple, there exists $b \in \mathbb{Z}_n$ such that $a = \alpha b \bmod n$, so there exists an integer m such that α divides $a - mn$. Therefore $\gcd(\alpha, n)$ divides $a - mn$ and $\gcd(\alpha, n)$ divides a.

Suppose $d = \gcd(\alpha, n)$ divides a. There exists b such that $a = db$. Thanks to Bezout theorem, there exists two integers u and v such that $u\alpha + vn = d$. Then, in $\mathbb{Z}_n$:

$$a = db = db - vnb = u\alpha b$$

and a is a α-multiple. ∎

Corollary D.4 *There are exactly $\frac{n}{\gcd(\alpha,n)}$ α-multiples modulo n. Furthermore for each α-multiples modulo n, there exists* $\gcd(\alpha, n)$ *elements $b \in \mathbb{Z}_n$, such that $a = \alpha b$.*

Proof: Thanks to the previous theorem, the α-multiples are the elements $\gcd(\alpha, n) \cdot a$, with $a \in \{0, \ldots, \frac{n}{\gcd(\alpha,n)} - 1\}$. And if a is an α-multiple, there exists b such that $a = \alpha b$, and then $a = \alpha \cdot (b + \frac{i\,n}{\gcd(\alpha,n)})$, for each $i \in \{0, \ldots, \gcd(\alpha, n) - 1\}$. ∎

In addition, we have the following corollary, which leads to an efficient way of checking e-residuosity in $\mathbb{Z}_N^*$ in the section after next:

Corollary D.5 *Let α be a positive integer. a is an α-multiple if and only if $\frac{n}{\gcd(\alpha,n)}a = 0$ (in $\mathbb{Z}_n$).*

Proof: Thanks to the previous theorem, it is sufficient to prove that $\frac{n}{\gcd(\alpha,n)}a = 0$ if and only if $\gcd(\alpha, n)$ divides a. If $\gcd(\alpha, n)$ divides a, clearly $\frac{n}{\gcd(\alpha,n)}a = 0$. Otherwise, let us write $a = \gcd(\alpha, n)q + r$, with $0 \leq r < \gcd(\alpha, n)$, then $\frac{n}{\gcd(\alpha,n)}a = \frac{nr}{\gcd(\alpha,n)} \neq 0$ in $\mathbb{Z}_n$. ∎

D.1.3 Main Theorem

Theorem D.6 *Let α, β, γ be three positive integers. Suppose γ is coprime with n. Then, βa is a α-multiple, if and only if a is a $\gamma\frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}$-multiple.*

Remark D.7 Let us choose $a = \frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}$. Then βa is divisible by $\alpha \wedge n$ and so is a α-multiple. But, for any divisor $\gamma \neq 1$ of n, which does not divides $\frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}$, a is not a γ-multiple. Hence, we can see the theorem as optimal.

Proof: If a is a $\gamma\frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}$-multiple, a is divisible by $\gcd(\gamma\frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}, n) = \frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}$ and so βa is divisible by $\alpha\wedge n$ and a is a α-multiple.

If βa is a α-multiple, βa is divisible by $\alpha \wedge n$. $\frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}$ is coprime with $\frac{\beta}{\alpha\wedge\beta\wedge n}$ and divides $\frac{\beta}{\alpha\wedge\beta\wedge n}a$. So, thanks to Gauss theorem, a is divisible by $\frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}$. Since $\gcd(\gamma\frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}, n) = \frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}$, a is a $\gamma\frac{\alpha\wedge n}{\alpha\wedge\beta\wedge n}$-multiple. ∎

D.2 Residues of $\mathbb{Z}_N^*$

We can then use the previous results to prove some results on residues of $\mathbb{Z}_N^*$.

Definition D.8 Let e be a positive integer. $A \in \mathbb{Z}_N^*$ is a **e-residue** modulo N if and only if there exists $B \in \mathbb{Z}_N^*$ such that $A = B^e$.

Remark D.9 Let p be an odd prime number and k a positive integer. It is well know there exists a group isomorphism ψ_{p^k} from $\mathbb{Z}_{p^k}^*$ to $\mathbb{Z}_{p^k-p^{k-1}}$. And we can see that, for any $A \in \mathbb{Z}_{p^k}^*$, A is a e-residue modulo p^k if and only if $\psi_{p^k}(A)$ is a e-multiple (in $\mathbb{Z}_{p^k-p^{k-1}}$).

Let $N = p_1^{k_1} \dots p_m^{k_m}$ be the prime decomposition of N.
The following lemma comes directly from the Chinese Remain Theorem:

Lemma D.10 *$A \in \mathbb{Z}_N^*$ is a e-residue modulo N if and only is it is an e-residue modulo $p_i^{k_i}$ for all i.*

And then, thanks to Theorem D.6, Remark D.9 and Lemma D.10, we have the following theorem:

Theorem D.11 *Let e, c be two positive integer, and U, Z two elements of Z_N^* such that $Z^e = U^c$ (i.e., U^c is a e-residue). Then U is a e'-residue, with e' the gcd of all $\frac{e\wedge(p_i^{k_i}-p_i^{k_i-1})}{c\wedge e\wedge(p_i^{k_i}-p_i^{k_i-1})}e_i$, with e_i the greatest divisor of e coprime with $p_i^{k_i} - p_i^{k_i-1}$.*

Remark D.12 The optimality of this theorem comes from the optimality of Theorem D.6.

Thanks to Remark D.9, Lemma D.10 and Corollary D.4, we also have the following proposition:

Proposition D.13 *The number of e-residues modulo N is:*

$$\phi(N, e) = \prod_{i=1}^{m} \frac{p^{k_i} - p^{k_i-1}}{e \wedge (p^{k_i} - p^{k_i-1})}$$

Furthermore, each e-residue modulo N has exactly $\phi(N)/\phi(N, e)$ roots.

And thanks to Remark D.9, Lemma D.10 and Corollary D.5, we also have the following proposition, which yields an efficient algorithm to know if an integer U is an e-residue modulo N or not (if the factorization of N is known):

Proposition D.14 *$U \in \mathbb{Z}_N^*$ is an e-residue modulo N if and only if, for all i:*

$$U^{\frac{p^{k_i}-p^{k_i-1}}{e\wedge(p^{k_i}-p^{k_i-1})}} = 1 \bmod p^{k_i}.$$

E Generic Forward-Secure Scheme Proofs

In this appendix, we give a detailed proof that our generic forward-secure scheme described in Section 6 is forward-secure under Condition 6.1. Let us prove it for the case $T = 1$ and forget indexes i to make the proof easier to understand. The key-evolving extension is trivial.

Informally, the condition 6.1 ensures that, in a lossy setting, given a commitment, there cannot be more than one challenge for which there exists a response. This follows from some arithmetical results on residues, described in Appendix D, and namely in Theorem D.11. Formally, we have to show that $\mathcal{ID}$ meets the simulatability, completeness, key indistinguishability, and lossiness conditions.

The proof that $\mathcal{ID}$ is **complete** follows immediately from the fact that, if $U_j = S_j^e \bmod N$ for $j \in \{1, \ldots, \ell\}$, an honest execution of the protocol will always result in acceptance as $Z_j^e = (R_j \cdot S_j^{c_j})^e = R_i^e \cdot (S_j^e)^{c_j} = Y_j \cdot U_j^{c_j}$.

The **simulatability** of $\mathcal{ID}$ follows from the fact that, given $pk = (N, e, U_1, \ldots, U_\ell)$, we can easily generate transcripts whose distribution is perfectly indistinguishable from the transcripts output by an honest execution of the protocol. This is done by choosing Z_j uniformly at random in $\mathbb{Z}_N^*$ and c_j uniformly at random in $\{0, \ldots, \mathfrak{c} - 1\}$, and setting $Y_j = Z_j^e / U_j^{c_j}$ for $j \in \{1, \ldots, \ell\}$.

The **key indistinguishability** directly follows from Condition 6.1.

To show that $\mathcal{ID}$ is **lossy**, we note that, when the public key is lossy, for every element Y_j chosen by the adversary, there exists only one value of $c_i \in \{0, \ldots, \mathfrak{c} - 1\}$ for which there exists a valid response Z_j which passes the test. To see why, assume for the sake of contradiction that there exist two different values $c_{j,1}$ and $c_{j,2}$ in $\{0, \ldots, \mathfrak{c} - 1\}$ for which there exists a valid response. Denote by $Z_{j,1}$ and $Z_{j,2}$ one of the valid responses in each case. Without loss of generality, assume that $c_{j,1} < c_{j,2}$. Since $Z_{j,1}^e = Y_j \cdot U_i^{c_{j,1}}$ and $Z_{j,2}^e = Y_j \cdot U_j^{c_{j,2}}$, we have that $(Z_{j,2}/Z_{j,1})^e = U_j^{c_{j,2} - c_{j,1}}$. As $c_{j,2} - c_{j,1}$ is a positive number smaller than $\mathfrak{c}$, this means that U_j is an $e'(e, c_{j,2} - c_{j,1}, N)$-residue, according to Theorem D.11, which is a contradiction. This means that the probability that a valid response Z_j exists in the case where U_j is pseudo-e-residue is at most $1/\mathfrak{c}$. Since there are ℓ challenges, it follows that $\mathcal{ID}$ is ε_ℓ-lossy, with $\varepsilon_\ell = 1/\mathfrak{c}^\ell$.

More formally, we have proven the following theorem:

Theorem E.1 *Under Condition 6.1, $\mathcal{ID}$ is complete, 1-simulatable, key indistinguishable and $1/\mathfrak{c}^\ell$-lossy. Furthermore, the min-entropy β of the commitment scheme is at least the minimum over $i \in \{1, \ldots, T\}$ of $\ell \log_2(\phi(N, e))$ where $\phi(N, e_i)$ is the number of e_i-residues modulo N.*

Therefore, thanks to Theorem 3.1, we can prove that our generic forward-secure signature scheme $\mathcal{FS}$ is existentially forward-secure.

F Some Propositions on Prime Numbers

This section shows some known results on primes numbers.

Let $\pi(x)$ be the number of primes not greater than x. The following lemma is a direct corollary of Theorem 1.10 in [Dus98]:

Lemma F.1 *For $x \geq 599$,*

$$\frac{x}{\log x}\left(1 + \frac{1}{\log x}\right) \leq \pi(x) \leq \frac{x}{\log x}\left(1 + \frac{1.28}{\log x}\right)$$

From this lemma, we can prove the following proposition:

Proposition F.2 *The number of primes of length ℓ_e is at least $2^{\ell_e - 1}/(\ell_e - 1)$, if $\ell_e \geq 11$.*

Proof: If $\ell_e \geq 11$, $2^{\ell_e-1} \geq 599$, so the previous lemma applies to $x = 2^{\ell_e}$ and $x = 2^{\ell_e-1}$.
The number of primes of length ℓ_e is

$$\begin{aligned}
&\pi(2^{\ell_e}) - \pi(2^{\ell_e-1}) \\
&\geq \frac{2^{\ell_e}}{\log 2^{\ell_e}}\left(1 + \frac{1}{\log 2^{\ell_e}}\right) - \frac{2^{\ell_e-1}}{\log 2^{\ell_e-1}}\left(1 + \frac{1.28}{\log 2^{\ell_e-1}}\right) \\
&= \frac{2^{\ell_e}-1}{\ell_e-1}\frac{\ell_e-1}{\log 2}\left(\frac{25\,\log(2)\,(\ell_e-1)^3 + 18\,(\ell_e-1)^2 - (25\,\log(2)+64)\,(\ell_e-1) - 32}{25\,\log(2)\,(\ell_e-1)^2\,{\ell_e}^2}\right)
\end{aligned}$$

The derivative of $\frac{\ell_e-1}{\log 2}\left(\frac{25\,\log(2)\,(\ell_e-1)^3+18\,(\ell_e-1)^2-(25\,\log(2)+64)\,(\ell_e-1)-32}{25\,\log(2)\,(\ell_e-1)^2\,{\ell_e}^2}\right)$ is

$$\frac{2(\ell_e-1)}{\log 2}\left(\frac{(25\,\log(2)-9)\,(\ell_e-1)^3 + (25\,\log(2)+73)\,(\ell_e-1)^2 + 48\,(\ell_e-1) + 16}{25\,\log(2)\,(\ell_e-1)^3\,{\ell_e}^3}\right)$$

Therefore, this function is increasing. Since its value in $\ell_e = 11$ is greater than 1.0, we have the following result

$$\pi(2^{\ell_e}) - \pi(2^{\ell_e-1}) \geq \frac{2^{\ell_e-1}}{\ell_e-1}$$

for any $\ell_e \geq 11$.

∎

Let us now introduce a new proposition useful to prove key indistinguishability in the GQ scheme (Section 4.1).

Proposition F.3 *Let ℓ_N, ℓ_e be two positive integers. Suppose $\ell_e < \ell_N$ and $\ell_N \geq 10$. Let N be a ℓ_N-bit integer. Let D_0 be the uniform distribution for ℓ_e-bit primes. Let D_1 be the uniform distribution for ℓ_e-bit primes, coprime with $\phi(N)$. The statistical distance between D_0 and D_1 is at most $2\,\frac{\ell_N+1}{2^{\ell_e-1}}$.*

Proof: Let N_0 be the number of ℓ_e-bit primes and N_1 be the number of ℓ_e-bit primes, coprime with $\phi(N)$. We remark that a ℓ_e-bit prime is at least 2^{ℓ_e-1}, and so there are at most $(\ell_N+1)/(\ell_e-1)$ such primes which divide $\phi(N)$. Otherwise, their product is greater than 2^{ℓ_N} and it divides $\phi(N) < 2^{k+1}$, which is impossible. Therefore, $N_1 \geq N_0 - (\ell_N+1)/(\ell_e-1)$. Furthermore, according to Proposition F.2: $N_0 \geq 2^{\ell_e-1}/(\ell_e-1)$.

The statistical distance is

$$\begin{aligned}
D &= \sum_{x\in\mathbb{P}_{\ell_e}} \left|\Pr_{e\xleftarrow{\$}\mathsf{D}_0}[\,e=x\,] - \Pr_{e\xleftarrow{\$}\mathsf{D}_1}[\,e=x\,]\right| \\
&= \sum_{\substack{x\in\mathbb{P}_{\ell_e}\\ \gcd(x,\phi(n))=1}} \left|\Pr_{e\xleftarrow{\$}\mathsf{D}_0}[\,e=x\,] - \Pr_{e\xleftarrow{\$}\mathsf{D}_1}[\,e=x\,]\right| \\
&\quad + \sum_{\substack{x\in\mathbb{P}_{\ell_e}\\ \gcd(x,\phi(n))=x}} \left|\Pr_{e\xleftarrow{\$}\mathsf{D}_0}[\,e=x\,] - \Pr_{e\xleftarrow{\$}\mathsf{D}_1}[\,e=x\,]\right| \\
&= \sum_{\substack{x\in\mathbb{P}_{\ell_e}\\ \gcd(x,\phi(n))=1}} \left|\frac{1}{N_0} - \frac{1}{N_1}\right| + \sum_{\substack{x\in\mathbb{P}_{\ell_e}\\ \gcd(x,\phi(n))=x}} \left|\frac{1}{N_0} - 0\right| \\
&= N_1\left|\frac{1}{N_0} - \frac{1}{N_1}\right| + (N_0-N_1)\left|\frac{1}{N_0} - 0\right| = 1 - \frac{N_1}{N_0} + 1 - \frac{N_1}{N_0} = 2\frac{N_0-N_1}{N_0}
\end{aligned}$$

We have $N_0 - N_1 \leq (\ell_N + 1)/(\ell_e - 1)$ and, according to Proposition F.2, $N_0 \geq 2^{\ell_e - 1}/(\ell_e - 1)$, so:

$$D \leq 2\,\frac{\ell_N + 1}{2^{\ell_e - 1}}$$

∎

G Instantiations of our Generic Factoring-Based Signature and Forward-Secure Signature Schemes

This section goes into the details of the GQ scheme (Section 4.1), our variant of the IR scheme (Section 4.2) and the schemes described in 6.2, after describing a slight optimization of the generic scheme for our cases.

G.1 An Optimization

Let us present an optimization of the generic scheme for our cases. We consider the case of a classical signature scheme ($T = 1$) for the sake of simplicity.

We can remark that if the factorization[12] of N is hard, then we can replace the test $Z_j \in \mathbb{Z}_N^*$ by $Z_j \bmod N = 0$, in the identification scheme depicted in Figure 6.1. We just need to remark that the (existential) forward-security (or unforgeability) game with the original verification and the one with the new verification are identical until the following bad event happens: one of the Z_j is not equal to 0 modulo N, nor coprime with N. But this bad event only happens with low probability, because it leads to factorizing N. More precisely, if the key-indistinguishability is easier than the factorization (meaning that knowing the factorization of N solves the key-indistinguishability problem with probability close to 1 — which is the case with all our instantiation), the security reductions of Theorem 3.1 and Theorem C.4 are still valid with our new scheme.

Let us prove this for Theorem 3.1, we use the same proofs except the games $\mathrm{G}_0, \ldots, \mathrm{G}_7$ use the new scheme (i.e., check only that $rsp^* \neq 0$ instead of $rsp^* \in \mathbb{Z}_N^*$) and the next ones use the old scheme. We just need to prove Equation (C.1) still holds. From adversary $\mathcal{A}$, we create an adversary $\mathcal{B}$ for the key indistinguishability, which simulates $\mathcal{A}$ in the environment of game G_7 or G_8. If $\mathcal{A}$ outputs a correct forgery for the old scheme ($rsp^* \in \mathbb{Z}_N^*$), it outputs 1; if $\mathcal{A}$ outputs a correct forgery for the new scheme but not the old scheme ($rsp^* \notin Z_N^*$), it factorizes N, solves the key-indistinguishability problem and outputs 1 if the key is normal, and 0 if the key is lossy; finally, if $\mathcal{A}$ does not output a forgery for the old scheme, it outputs 0. Therefore, if the key provided is normal, with probability at least $\Pr[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1\,]$, $\mathcal{B}$ will output 1, because, if $\mathrm{G}_7(\mathcal{A}) \Rightarrow 1$, either $rsp^* \in Z_N^*$, in which case $\mathcal{B}$ output 1, either $rsp^* \notin Z_N^*$, in which case $\mathcal{B}$ can solve the key-indistinguishability problem and output 1. And if the key provided is lossy, with probability at least $\Pr[\,\mathrm{G}_8(\mathcal{A}) \Rightarrow 0\,]$, $\mathcal{B}$ will output 0, because, if $\mathrm{G}_8(\mathcal{A}) \Rightarrow 0$, either $rsp^* \in Z_N^*$, in which case $\mathcal{B}$ output 0, either $rsp^* \notin Z_N^*$, in which case $\mathcal{B}$ can solve the key-indistinguishability problem and output 0. Therefore the advantage of $\mathcal{B}$ to solve the key-indistinguishability is at least

$$\Pr[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1\,] - (1 - \Pr[\,\mathrm{G}_8(\mathcal{A}) \Rightarrow 0\,]) = \Pr[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1\,] - \Pr[\,\mathrm{G}_8(\mathcal{A}) \Rightarrow 1\,].$$

This proves Equation (C.1). The same proof also applies to Theorem C.4.

G.2 Signature Schemes

G.2.1 Guillou-Quisquater Signature Scheme

This scheme, where e is a ℓ_e-bit prime number, with ℓ_e a security parameter, and $\ell = 1$ approximately coincides with the GQ identification scheme [GQ90]. A complete description of the scheme with some

[12] By factorization, we mean finding any non-trivial factor of N.

$\underline{\mathsf{KG}(1^k, 1^T)}$	$\underline{\mathsf{Sign}(sk, M)}$	$\underline{\mathsf{Ver}(pk, \sigma, M)}$

$(N, p_1, p_2) \xleftarrow{\$} \mathsf{RSA}_{\ell_N}$
$e \xleftarrow{\$} \mathbb{P}_{\ell_e}$
$S \xleftarrow{\$} \mathbb{Z}_N^*$
$U \leftarrow S^{-e}$
$pk \leftarrow (N, e, U)$
$sk \leftarrow (N, e, f, S)$
return (pk, sk)
	$(N, e_T, S) \leftarrow sk$
	$R \xleftarrow{\$} \mathbb{Z}_N^*$
	$Y \leftarrow R^e \bmod N$
	$c \leftarrow \mathsf{H}(\langle Y, M\rangle)$
	$Z \leftarrow RS^c \bmod N$
	$\sigma \leftarrow (c, Z)$
	return σ
		$(N, e, U) \leftarrow pk$
		$(c, Z) \leftarrow \sigma$
		if $Z \bmod N = 0$
		  return reject
		$Y \leftarrow Z^e U^c \bmod N$
		if $\mathsf{H}(\langle Y, M\rangle) = c$
		  return accept
		else
		  return reject

$\underline{\mathsf{KG}_{\mathrm{swap}}(1^k, 1^T)}$	$\underline{\mathsf{Sign}_{\mathrm{swap}}(sk, M)}$	$\underline{\mathsf{Ver}_{\mathrm{swap}}(pk, \sigma, M)}$

$(N, p_1, p_2) \xleftarrow{\$} \mathsf{RSA}_{\ell_N}$
$e \xleftarrow{\$} \mathbb{P}_{\ell_e}$
$f \leftarrow e^{-1} \bmod N$
$S \xleftarrow{\$} \mathbb{Z}_N^*$
$U \leftarrow S^{-e}$
$pk \leftarrow (N, e, U)$
$sk \leftarrow (N, e, S)$
return (pk, sk)
	$(N, e_T, S) \leftarrow sk$
	$c \xleftarrow{\$} \{1, \ldots, 2^{\ell_e - 1}\}$
	$Y \leftarrow \mathsf{H}(\langle c, M\rangle)$
	$R \leftarrow Y^f \bmod N$
	$Z \leftarrow RS^c \bmod N$
	$\sigma \leftarrow (c, Z)$
	return σ
		$(N, e, U) \leftarrow pk$
		$(c, Z) \leftarrow \sigma$
		if $Z \bmod N = 0$
		  return reject
		$Y \leftarrow Z^e U^c \bmod N$
		if $\mathsf{H}(\langle c, M\rangle) = Y$
		  return accept
		else
		  return reject

Figure G.1: GQ signature scheme

optimizations (using, in particular, the Remark 3.2 and the fact that the inversion is a permutation over $\mathbb{Z}_N^*$) is depicted in Figure G.1.

Formal Proof of Security. To formally prove the key indistinguishability of $I\mathcal{D}$, we first note that, the statistical distance between the distribution D_0 of (N, e, U) for a normal key and the distribution D_1 of (N, e, U) where e is coprime with $\phi(N)$ and U is still generated as $U = S^e$, is at most $2\,\frac{\ell_N+1}{2^{\ell_e-1}}$ according to Proposition F.3. Furthermore, in the distribution D_1, U is a random element of $\mathbb{Z}_N^*$ due to the fact that the function $f(x) = x^e \bmod N$ is a permutation over $\mathbb{Z}_N^*$ when $\gcd(e, \phi(N)) = 1$. If the ϕ-hiding problem is (t'', ε'')-hard, the distribution D_1 is indistinguishable from the distribution D_2 of (N, e, U), where U is a random element of Z_N^* and e divides $\phi(N)$. Then, the statistical distance between the distribution D_2 and the distribution D_3 of (N, e, U) for lossy keys is $2/e \leq 1/2^{\ell_e-2}$. The proof is straightforward and similar to the proof of Proposition F.3, since there are exactly $\phi(N)/e$ e-residues among the $\phi(N)$ elements of $\mathbb{Z}_N^*$, according to Proposition D.13. As a result, $I\mathcal{D}$ is (t', ε')-key-indistinguishable where $t' = t'' - O(1)$ and $\varepsilon' = \varepsilon'' + \frac{2\ell_N+3}{2^{\ell_e-2}}$.

In addition, $e'(e, c, N) = e$ for any $c \in \{1, \ldots, \mathfrak{c} - 1\}$. Indeed, if $e \wedge (p_i - 1) = e$, $e'_i = 1$; otherwise $e \wedge (p_i - 1) = 1$ and $e'_i = e$, because e is prime. Therefore $(e \wedge (p_i - 1))e'_i = e$ and $e \wedge c \wedge (p_i - 1) = 1$, for $i \in \{1, 2\}$. And, for any $j \in \{1, \ldots, \ell\}$, for lossy keys, U_j is not a e-residue. So, Condition 6.1 is verified.

Finally, the scheme is clearly response-unique for keys of D_1 (since the function $f(x) = x^e \bmod N$ is a permutation over $\mathbb{Z}_N^*$ when $\gcd(e, \phi(N)) = 1$). We can see that it is sufficient[13] to have the strong unforgeability (instead of the existential unforgeability). And we have the following theorem:

Theorem G.1 *If the ϕ-hiding problem is (t', ε')-hard, then the previous scheme is (approximately) $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure in the random oracle model for:*

$$t \approx \frac{t' \cdot \varepsilon}{2} - q_s t_{\mathsf{Sim-Sign}}$$

as long as

$$\varepsilon \geq 2\frac{(q_h + 1)q_s}{2^{\ell_N - 2\ell_e - 2}} \text{ and } \varepsilon' + \frac{2\ell_N + 3}{2^{\ell_e - 2}} \leq \delta\left(1 - \frac{1}{\mathrm{e}}\right) - \frac{2\,(1 + q_h)}{\varepsilon\, 2^{\ell_e - 1}}.$$

[13] Although D_1 is not exactly the distribution of normal keys, it is statistically very close to it.

The theorem is not exact since for normal keys, factorization of N enables to distinguish normal keys from lossy keys with a very high probability but not 1, due to the statistical distance between D_0 and D_1. For sake of simplicity we do not take into account this completely negligible fact.

Swap Method. Applying the swap method [MR02] to the Guillou Quisquater identification scheme can also provide a signature with a tight reduction, to the RSA problem. The corresponding algorithm is depicted in Figure G.1 (where we suppose H is a random oracle for elements in $\mathbb{Z}_N^*$, which can be roughly implemented by a random oracle for elements in $\{1,\ldots,N-1\}$[14]). We see that, this algorithm requires two exponentiations modulo N, one with a ℓ_e-bit exponent e and one with a ℓ_N-bit exponent f, whereas our signing algorithm only requires two exponentiations modulo N with a ℓ_e-bit exponent e. So our signing algorithm is $\ell_N/(2\ell_e)$ faster, for the same parameters and the same security level, if we consider the ϕ-hiding problem is as hard as the RSA problem, and if we disregard the small differences of the exact tightness of the reductions.

G.2.2 Quadratic-Residuosity-Based Scheme

This scheme, where $e = 2$, coincides with the quadratic-residuosity-based scheme informally suggested by Katz and Wang in [KW03].

Suppose $\mathfrak{c} = e$, $N = p_1p_2$ is an RSA modulus and the algorithm LKG chooses $U_1,\ldots,U_l$ uniformly at random from the set $\mathsf{J}_N[e] \setminus \mathsf{HR}_N[e]$. Let us prove that Condition 6.1 is verified.

To prove the key indistinguishability, we use the fact that the e-residuosity problem is random-self-reducible. That is, the distribution $(U_1,\ldots,U_\ell)$ where U_i is chosen uniformly at random from $\mathsf{HR}_N[e]$ is indistinguishable from the distribution $(U, U\alpha_2^e \bmod N,\ldots,U\alpha_\ell^e \bmod N)$ where U is chosen uniformly at random from $\mathsf{HR}_N[e]$ and α_i for $i \in \{2,\ldots,\ell\}$ is chosen uniformly at random from $\mathbb{Z}_N^*$. The latter distribution is clearly indistinguishable from the distribution $(U_1,\ldots,U_\ell)$ where U_i is chosen uniformly at random from $\mathsf{J}_N[e] \setminus \mathsf{HR}_N[e]$ due to the hardness of the e-residuosity problem. As a result, $\mathcal{ID}$ (t',ε)-key-indistinguishable where $t' = t - O((\ell-1)\cdot t_{\exp})$ where $t_{\exp}$ denotes the average time to compute an exponentiation with respect to exponent e.

Furthermore, $e'(e,c,N) = 2$ for any $c \in \{1,\ldots,\mathfrak{c}-1\}$ (i.e., $c = 1$), since $e \wedge (p_i-1) = 2$, $e \wedge c \wedge (p_i-1) = 1$ and $e'_i = 1$, for $i \in \{1,2\}$.

According to our security proof, this scheme is existentially unforgeable in the random oracle model based on the hardness of the quadratic-residuosity problem as long as ℓ is large enough to make the term $q_h/2^\ell$ negligible. And the reduction is tight.

G.2.3 2^t-Root Scheme by Ong and Schnorr

This scheme, where $e = 2^t$ is a t-power of 2 and $\ell = 1$, coincides with the 2^t-root identification scheme by Ong and Schnorr [OS90].

Suppose $\mathfrak{c} = e$, $N = p_1p_2$ is an RSA modulus such that 2^t divides $p_1 - 1$ and $p_2 - 1$ and the algorithm LKG chooses $U_1,\ldots,U_l$ uniformly at random from the set $\mathsf{J}_N[2] \setminus \mathsf{HR}_N[2]$. Let us prove that, if the strong-2^t-residuosity problem is hard, Condition 6.1 is verified.

Indeed, the key indistinguishability directly comes from the strong-2^t-residuosity. Furthermore, $e'(e,c,N)$ is a multiple of 2 for any $c \in \{1,\ldots,\mathfrak{c}-1\}$, since $e \wedge (p_i-1) = 2^t$, $e \wedge c \wedge (p_i-1)$ divides 2^{t-1} and $e'_i = 1$, for $i \in \{1,2\}$. So, Condition 6.1 is verified.

According to our security proof, this scheme is existentially unforgeable in the random oracle model based on the hardness of the strong-2^t-residuosity problem as long as t is large enough to make the term $q_h/2^t$ negligible. And the reduction is tight.

We can easily extend this scheme to $\ell > 1$. The self-reducibility of the strong-2^t-residuosity problem enables to prove the key indistinguishability. In this case, we only need the term $q_h/2^{\ell t}$ to be negligible.

[14] The probability that a random non-zero element of $\mathbb{Z}_N$ is non-invertible is negligible.

G.2.4 Paillier Signature Scheme

This scheme, where $\ell = 1$ and $e = p_1 p_2$ is an RSA modulus and $N = e^2 = p_1^2 p_2^2$, coincides with the Paillier signature scheme [Pai99]. Let us prove that Condition 6.1 is verified.

Suppose $\mathfrak{c} \leq \min(p_1, p_2)$ (we can choose for example, $\mathfrak{c} = \lfloor \sqrt{e}/2 \rfloor$, if $p_1, p_2 \geq \sqrt{e}/2$) and the algorithm LKG chooses $U_1, \ldots, U_l$ uniformly at random from the set $\mathsf{J}_N[e] \setminus \mathsf{HR}_N[e]$. The key indistinguishably is similar to the one of the above schemes.

In addition, $e'(e, c, N) = p_1 p_2 = e$ for any $c \in \{1, \ldots, \mathfrak{c} - 1\}$. Indeed, if $e \wedge (p_i - 1) = p_{3-i}$, $e \wedge (p_i^2 - p_i) = p_1 p_2$ and $e'_i = 1$; otherwise $e \wedge (p_i^2 - p_i) = p_i$ and $e'_i = p_{3-i}$. Therefore $(e \wedge (p_i^2 - p_i))e'_i = e$ and $e \wedge c \wedge (p_i - 1) = 1$, for $i \in \{1, 2\}$. So, Condition 6.1 is verified.

According to our security proof, the construction provides a signature scheme existentially unforgeable with a tight security reduction to the N-residuosity problem of [Pai99].

G.3 Forward-Secure Signature Schemes

G.3.1 Variant of the Itkis-Reyzin Scheme

This section goes into the detail of the security of our variant of the IR scheme presented in Section 4.2. We have the following theorem, which follows from the analysis, Corollary C.5 and Theorem E.1:

Theorem G.2 *If the ϕ-hiding problem is (t', ε')-hard, then our variant of the IR scheme is (approximately) $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure in the random oracle model for:*

$$t \approx \frac{(t' - (T-1)\, t_{\mathsf{Update}}) \cdot \varepsilon}{2} - q_s t_{\mathsf{Sim-Sign}}$$

as long as

$$\varepsilon \geq 2\frac{(q_h + 1)q_s}{2^{\ell_N - 2\ell_e - 2}} \text{ and } \varepsilon' \leq \delta\left(1 - \frac{1}{\mathrm{e}}\right) - \frac{2\,(1 + q_h)}{2^{\ell_e - 1}\,\varepsilon}.$$

For this theorem, for sake of simplicity, we did not take into account the statistical distance between D_0 and D_1, and D_1 and D_2, from Section 4.2. That is why the theorem is a (very) slight approximation.

Under the assumption of Remark B.4, if we suppose $t_{\mathsf{Sim-Sign}} = 0$ and $t_{\mathsf{Update}} = 0$, we can say that the scheme is about $(\frac{t\varepsilon}{2}, q_h, q_s, T\varepsilon)$-forward-secure if the ϕ-hiding problem is $(t, (1 - 1/e)/2)$-hard. This means roughly that if we want k-bits of security, the modulus has to correspond to a security level of $k' = k + \log_2(T)$ bits (k' being an approximate solution of $2^{k'} = \frac{2^k \varepsilon}{2} \frac{1}{T\varepsilon}$).

G.3.2 Power-of-2-Root Forward-Secure Scheme

For this scheme, $e_i = 2^{t(T-i+1)}$ with t a positive integer, $\mathfrak{c} = 2^i$. We remark that, in this case, $f_i = e_i$, and one can easily change the algorithm such that we only store $S_{j,i} = S_j^{e_i}$.

As for the 2^t-root signature scheme, we need to choose $N = p_1 p_2$ such that 2^{tT} divides $p_1 - 1$ and $p_2 - 1$. Unfortunately this means the keys and signatures have a length linear in T.

The proof that Condition 6.1 is verified, is quite similar to the proof in Section 6.2. To generate a lossy key for period $\tilde{\imath}$, LKG chooses $S_{1,\tilde{\imath}}, \ldots, S_{\ell,\tilde{\imath}}$ uniformly at random in $\mathsf{J}_N[2] \setminus \mathsf{HR}_N[2]$, $S_{j,i} = S_{j,\tilde{\imath}}^{e_{\tilde{\imath}}/e_i}$ for $i > \tilde{\imath}$, and $U_j = S_{j,\tilde{\imath}}^{e_{\tilde{\imath}}}$. We then remark that the key-indistinguishability for period $\tilde{\imath}$ can trivially be reduced to the key-indistinguishability problem for the 2^t-root scheme by Ong and Schnorr in Section 6.2, which itself can be reduced to the strong-$2^{t(T-\tilde{\imath}+1)}$-assumption. The lossiness can also be proven as for the 2^t-root scheme by Ong and Schnorr.

Therefore, this scheme is existentially forward-secure in the random oracle model based on the hardness of the strong-2^{ti}-assumption, for all $i \in \{1, \ldots, T\}$, as long as the exponents t and ℓ are large enough to make the term $q_h/2^{t\ell}$ negligible. And the reduction is relatively tight (we only lose a factor T).

H Generic Proofs of Security Based on the Forking Lemma for Key-Evolving Collision-Intractable Identification Schemes

In this appendix, we introduce generic proofs of security based on the forking lemma for signatures obtained from some particular key-evolving identification schemes via the generalized Fiat-Shamir transform described in Section 3.2. This is a generalization of [MR02], which itself is based on [PS00].

H.1 Key-Evolving Collision-Intractable Identification Schemes

In this section we extend the notion of collision-intractable identification scheme introduced in [CD95] to key-evolving identification scheme. Let $I\!D$ be a key-evolving identification scheme, as described in Section 3.1.

Informally, $I\!D$ is collision-intractable if an adversary cannot output two valid transcripts (cmt, ch, rsp) and (cmt, ch', rsp') with $ch \neq ch'$, for a period $\tilde{\imath}+1$, even with access to the public key pk and the secret key $sk_{\tilde{\imath}}$ for period $\tilde{\imath}$.

More formally, let $\mathcal{A}$ be an adversary and k be a security parameter. Let $\mathbf{Exp}^{\text{col-int}}_{I\!D,k}(\mathcal{A})$ be the following experiment played between $\mathcal{A}$ and a hypothetical challenger:

$\mathbf{Exp}^{\text{col-int}}_{I\!D,k}(\mathcal{A})$
$(pk, sk_1) \xleftarrow{\$} \mathsf{KG}(1^k)\,;\, \tilde{\imath} \xleftarrow{\$} \{1,\ldots,T\}\,;\, sk_{\tilde{\imath}+1} = \mathsf{Update}^{\tilde{\imath}}(sk_1)$
$(cmt, ch, rsp, ch', rsp') \xleftarrow{\$} \mathcal{A}(\tilde{\imath}, pk, sk_{\tilde{\imath}+1})$
$d = \mathsf{Ver}(pk, cmt, ch, rsp, i) \wedge \mathsf{Ver}(pk, cmt, ch', rsp', \tilde{\imath}) \wedge ch \neq ch'$
return d

$\mathcal{A}$ is said to (t,ε)-breaks the collision-intractability problem if $\mathcal{A}$ runs in time at most t and its probability of success is $\Pr\left[\mathbf{Exp}^{\text{col-int}}_{I\!D,k}(\mathcal{A}) = 1\right] \geq \varepsilon$. Furthermore, $I\!D$ is said to be (t,ε)-collision-intractable if no adversary (t,ε)-breaks the collision-intractability problem

H.2 Generalized Fiat-Shamir Transformation

Theorem H.1 *Let $I\!D = (\mathsf{KG}, \mathsf{Update}, \mathsf{Prove}, \ell_c, \mathsf{Ver})$ be a key-evolving identification scheme whose commitment space has min-entropy β, and whose challenge space has ℓ_c bits, let H be a random oracle, and let $\mathcal{FS}[I\!D] = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be the signature scheme obtained via the generalized Fiat-Shamir transform. If $I\!D$ is ε_s-simulatable, complete, (t',ε')-collision-intractable, then $\mathcal{FS}[I\!D]$ is $(t, q_h, q_s, \varepsilon, \delta)$-existentially-weakly-forward-secure in the random oracle model for:*

$$t \approx \frac{(t' - (T-1)\,t_{\mathsf{Update}}) \cdot \left(\varepsilon - q_s\,\varepsilon_s - (q_h+1)\,q_s/2^{\beta} - (q_h+1)/2^{\ell_c - 1}\right)}{2q_h + 3} - q_s\,t_{\mathsf{Sim-Sign}}$$

as long as

$$\varepsilon > q_s\varepsilon_s + \frac{(q_h+1)q_s}{2^{\beta}} + \frac{q_h+1}{2^{\ell_c-1}} \text{ and } \varepsilon' \leq \delta\left(1 - \frac{1}{\mathrm{e}}\right)^2$$

where t_{Sign} denotes the average time of a query to the simulated transcript function $\widetilde{\mathsf{Tr}}^{I\!D}_{pk,i,k}$ and t_{Update} denotes the average time of a query to Update. Furthermore, if $I\!D$ is response-unique (for normal keys), $\mathcal{FS}[I\!D]$ is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure.

Corollary H.2 *Under the same hypothesis of H.1, $\mathcal{FS}[I\!D]$ is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure in the random oracle model for:*

$$t \approx \frac{(t' - (T-1)\,t_{\mathsf{Update}}) \cdot \varepsilon}{4q_h + 6} - q_s t_{\mathsf{Sim-Sign}}$$

Initialize(k, T) Game $\mathrm{G}_7, \mathrm{G}'_8$

701 $\mathrm{hc} \leftarrow 1$; $\mathrm{b} \leftarrow T+1$
702 $\mathrm{fp} \xleftarrow{\$} \{1,\ldots,q_h+1\}$; $ch^* \xleftarrow{\$} \{0,1\}^{\ell_c}$
703 $\tilde{\imath} \xleftarrow{\$} \{1,\ldots,T\}$
704 $(pk, sk_1) \xleftarrow{\$} \mathsf{KG}(1^k, 1^T)$
705 for $i = 1,\ldots,T-1$
706 $\quad sk_{i+1} \leftarrow \mathsf{Update}(sk_i)$
707 return (pk, T)

H(x) Game $\mathrm{G}_7, \mathrm{G}'_8$

711 if $\mathsf{HT}(x) = \perp$ then
712 $\quad \mathsf{QT}(\mathrm{hc}) \leftarrow x$
713 $\quad$ if $\mathrm{hc} \neq \mathrm{fp}$ then
714 $\quad\quad \mathsf{HT}(x) \xleftarrow{\$} \{0,1\}^{\ell_c}$
715 $\quad$ else
716 $\quad\quad \mathsf{HT}(x) \xleftarrow{\$} ch^*$
717 $\mathrm{hc} \leftarrow \mathrm{hc}+1$
718 return $\mathsf{HT}(x)$

Breakin(i) Game $\mathrm{G}_7, \mathrm{G}'_8$

721 if $\mathrm{b} = T+1$
722 and $1 \le i \le T$ then
723 $\quad \mathrm{b} \leftarrow i$
724 $\quad$ if $i \le \tilde{\imath}$
725 $\quad\quad$ return $\perp$
726 $\quad$ return sk_i
727 else
728 $\quad$ return $\perp$

Sign(M, i) Game $\mathrm{G}_7, \mathrm{G}'_8$

731 $(cmt, ch, rsp) \xleftarrow{\$} \widetilde{\mathsf{Tr}}^{I\mathcal{D}}_{pk,i,k}$
732 if $S(cmt, M, i) \neq \perp$ then
733 $\quad \sigma \leftarrow (cmt, S(cmt, M, i))$
734 $\quad$ return $\langle \sigma, i \rangle$
735 $\mathsf{HT}(\langle cmt, M, i \rangle) \leftarrow ch$
736 $\sigma \leftarrow (cmt, rsp)$
737 $S(\langle cmt, M, i \rangle) \leftarrow rsp$
738 return $\langle \sigma, i \rangle$

Finalize$(M^*, \langle \sigma^*, i^* \rangle)$ Game $\mathrm{G}_7, \boxed{\mathrm{G}'_8}$

751 $d \leftarrow \mathsf{Ver}(pk, \langle \sigma^*, i^* \rangle, M^*)$
752 if $i^* \ge \mathrm{b}$ or $i^* \neq \tilde{\imath}$ then
753 $\quad d \leftarrow 0$
754 $(cmt^*, rsp^*) \leftarrow \sigma^*$
755 if $\mathsf{QT}(\mathrm{fp}) \neq (cmt^*, M^*)$ then
756 $\quad \mathsf{bad} \leftarrow \mathsf{true}$
757 $\quad \boxed{d \leftarrow 0}$
758 if $S(\langle cmt^*, M^*, i^* \rangle) \neq \perp$ then
759 $\quad d \leftarrow 0$
760 return $(d = 1)$

Figure H.1: Games $\mathrm{G}_7, \mathrm{G}'_8$ for proof of Theorem H.1. G'_8 includes the boxed code at line 757 but G_7 does not.

as long as

$$\varepsilon \ge 2\left(q_s\varepsilon_s + \frac{(q_h+1)q_s}{2^\beta} + \frac{q_h+1}{2^{\ell_c-1}}\right) \text{ and } \varepsilon' \le \delta\left(1-\frac{1}{\mathrm{e}}\right)^2.$$

Proof of Corollary H.2: It is a direct corollary of Theorem H.1. The condition

$$\varepsilon \ge 2\left(q_s\varepsilon_s + (q_h+1)q_s/2^\beta + (q_h+1)2^{-\ell_c+1}\right)$$

ensures that

$$\varepsilon - q_s\varepsilon_s - (q_h+1)q_s/2^\beta - (q_h+1)2^{-\ell_c+1} \ge \varepsilon/2.$$

∎

Proof of Theorem H.1: We use the same methods as Micali and Reyzin in [MR02]. Let us suppose there exists an adversary $\mathcal{A}$ which $(t, q_h, q_s, \varepsilon, \delta)$-breaks $\mathcal{DS}$. In particular, $\mathcal{A}$ $(t, q_h, q_s, \varepsilon\delta)$-breaks $\mathcal{DS}$. Let us consider the games $\mathrm{G}_0, \ldots, \mathrm{G}_7$ of Figure C.1 and Figure C.2, modified as for the proof of Theorem C.4 (using **Initialize** and **Finalize** of G_6 for games $\mathrm{G}_0, \ldots, \mathrm{G}_5$), and G'_8 of Figure H.1.

Assume $pk, sk_1, \tilde{\imath}$ are chosen such that $\Pr[\,\mathrm{G}_1 \Rightarrow 1\,] \ge \varepsilon - \gamma$ in G_7. It happens with probability at least δ. As in Section C.3, according to Equation (C.2), if we write $\gamma = q_s\varepsilon_s + (q_h+1)q_s/2^\beta$:

$$\Pr[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1\,] \ge \varepsilon - \gamma \text{ with probability at least } \delta \text{ over } (pk, sk_1, \tilde{\imath}).$$

In G'_8, the game outputs 0 if the signature does not corresponds to the challenge ch^*. Since we have

$$\Pr[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}_7\,] = \Pr[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1\,] \cdot \Pr[\,\mathsf{QT}(\mathrm{fp}) = (cmt^*, M^*)\,] = \frac{1}{1+q_h}\Pr[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1\,],$$

and $\Pr\left[\,\mathrm{G}'_8(\mathcal{A}) \Rightarrow 1 \wedge \mathsf{Good}'_8\,\right] = \Pr\left[\,\mathrm{G}'_8(\mathcal{A}) \Rightarrow 1\,\right]$, according to Lemma C.2, we have

$$\Pr\left[\,\mathrm{G}_7(\mathcal{A}) \Rightarrow 1\,\right] = (1+q_h)\Pr\left[\,\mathrm{G}'_8(\mathcal{A}) \Rightarrow 1\,\right].$$

And so

$$\Pr\left[\,\mathrm{G}'_8(\mathcal{A}) \Rightarrow 1\,\right] \geq \frac{\varepsilon-\gamma}{1+q_h} \text{ with probability at least } \delta \text{ over } (pk, sk_1, \tilde{\imath}).$$

Let $a = \frac{\varepsilon-\gamma}{1+q_h}$. Let us now construct an adversary $\mathcal{B}$ which breaks the collision-intractability problem. In the first part, $\mathcal{B}$ runs $\mathcal{A}$ α times and simulates the oracles as in G'_8, except for **Initialize** where it uses directly its inputs $\tilde{\imath}$, pk and $sk_{\tilde{\imath}+1}$ (instead of picking them uniformly at random). Every repetition starts completely anew, i.e., with a new random tape for $\mathcal{A}$ and new answers for the random oracle. With probability $1-(1-\alpha)^{\alpha^{-1}} \geq 1-1/\mathrm{e}$, the adversary $\mathcal{A}$ outputs at least a correct forgery accepted by **Finalize**, i.e., a forgery for period $i^* = \tilde{\imath}$ and with the challenge corresponding to the $\mathrm{fp}^{\mathrm{th}}$ query to the random oracle, fp being chosen uniformly at random at each run of $\mathcal{A}$. The adversary $\mathcal{B}$ stores the first correct output forgery $(M^*, \langle \sigma^*, i^* \rangle)$. Let $(cmt^*, rsp^*) = \sigma$ and $ch^* = \mathsf{H}(\langle cmt^*, M^* \rangle)$, such that (cmt^*, ch^*, rsp^*) is a correct transcript.

Now we can run again $\mathcal{A}$ a certain number of times with the same random tape s and the same answers for the random oracle queries up to the $\mathrm{fp}^{\mathrm{th}}$ query, and then use new uniform random answers.

Let us compute the probability that $\mathcal{A}$ will again output a correct forgery. Let ξ be the random variable $(s, \mathrm{fp}, h_1, \ldots, h_{\mathrm{fp}-1})$ with s the random tape of the adversary $\mathcal{A}$ and $h_1, \ldots, h_{\mathrm{fp}-1}$ be the answers to the $\mathrm{fp}-1$ first queries to the random oracle, for the first run where $\mathcal{A}$ managed to output again a correct forgery. Let E be the event that the adversary $\mathcal{A}$ outputs a correct forgery if it is simulated in the environment of game G'_8. For $\lambda = (s', \mathrm{fp}', h'_1, \ldots, h'_{\mathrm{fp}-1})$, let E_λ be the event that in such simulations, the random tape of $\mathcal{A}$ is $s = s'$, the fp chosen by **Initialize** is fp', and the answers to the $\mathrm{fp}-1$ first queries to the random oracle are $h'_1, \ldots, h'_{\mathrm{fp}}$. The events E_λ are disjoint, $\sum_\lambda \Pr\left[\,E_\lambda\,\right] = 1$, and, because of the choice of ξ, we also have $\Pr\left[\,\xi = \lambda\,\right] = \Pr\left[\,E_\lambda \,|\, E\,\right]$. In addition $\Pr\left[\,E\,\right] \geq \alpha$.

We can then apply the following lemma stated and proven in [MR02] (Lemma 3).

Lemma H.3 ([MR02]) *Let E be an event with probability α. Let Λ a finite set and let $(E_\lambda)_{\lambda\in\Lambda}$ be disjoint events such that $\sum_\lambda \Pr\left[\,E_\lambda\,\right] = 1$. Let ξ be a Λ-valued random variable with the following distribution $\Pr\left[\,\xi = \lambda\,\right] = \Pr\left[\,E_\lambda \,|\, E\,\right]$. Then*

$$\Pr_\xi\left[\,\Pr\left[\,E \,|\, E_\xi\,\right] \geq \frac{\alpha}{2}\,\right] \geq \frac{1}{2}.$$

Therefore, we have

$$\Pr_\xi\left[\,\Pr\left[\,E \,|\, E_\xi\,\right] \geq \frac{\alpha}{2}\,\right] \geq \frac{1}{2}.$$

which means that with probability $1/2$, the probability α' that the adversary $\mathcal{A}$ will do a forgery under condition ξ is at least $\alpha/2$. Assume ξ is such that $\alpha' \geq \alpha/2$. Then the probability that $\mathcal{A}$ will output a forgery corresponding to a "good" transcript (cmt^*, ch'^*, rsp'^*) with $ch'^* \neq ch^*$ is at least $\alpha/2 - 2^{-\ell_c}$.

So, in the second part, $\mathcal{B}$ runs $\mathcal{A}$ $(\alpha/2-2^{-\ell_c})^{-1}$ times under the condition ξ. The probability that $\mathcal{A}$ will output a forgery corresponding to a "good" transcript is

$$\left(1-\left(1-(\alpha/2-2^{-\ell_c})\right)^{(\alpha/2-2^{-\ell_c})^{-1}}\right) \geq 1-1/\mathrm{e}.$$

Therefore, with probability $(1-1/\mathrm{e})^2/2$, $\mathcal{B}$ gets two transcripts (cmt^*, ch^*, rsp^*) and (cmt^*, ch'^*, rsp'^*) with $ch^* \neq ch'^*$. This corresponds to the expected output for the game of collision-intractability.

We can now slightly improve the running time of $\mathcal{B}$. Instead of simulating the environment of G'_8, let $\mathcal{B}$ simulates the environment of G_7 when it runs $\mathcal{A}$, in the first part. The only difference is that $\mathcal{B}$ accepts

any forgery (in the first part) instead of accepting only forgeries for the $\mathrm{fp}^{\mathrm{th}}$ query to the random oracle, where fp is chosen uniformly at random. Therefore, $\mathcal{B}$ just needs to run $\mathcal{A}$ $\frac{1}{\varepsilon-\gamma}$ times instead of $\frac{1+q_h}{\varepsilon-\gamma}$, to have a forgery with probability at least $1-1/\mathrm{e}$. As explained in [MR02], the probability distribution of ξ is still the same and so it does not change the rest of the proof.

Let us now analyze the running time of $\mathcal{B}$. The first part takes $\frac{1}{\varepsilon-\gamma}\left(t+q_s t_{\mathsf{Sim-Sign}}\right)$ and the second part takes $\frac{1}{\alpha/2-2^{-\ell_c}}\left(t+q_s t_{\mathsf{Sim-Sign}}\right)$. Therefore $\mathcal{B}$ (t',ε')-breaks the collision intractability with

$$\begin{aligned} t' &\leq \left(\frac{1}{\varepsilon-\gamma}+\frac{1}{(\varepsilon-\gamma)/(2(q_h+1))-2^{-\ell_c}}\right)\left(t+q_s t_{\mathsf{Sim-Sign}}\right)+(T-1)\,t_{\mathsf{Update}} \\ &\leq \left(\frac{1}{\varepsilon-\gamma-\left(q_h+1\right)2^{-\ell_c+1}}+\frac{2\left(q_h+1\right)}{\varepsilon-\gamma-\left(q_h+1\right)2^{-\ell_c+1}}\right)\left(t+q_s t_{\mathsf{Sim-Sign}}\right)+(T-1)\,t_{\mathsf{Update}} \\ &\leq \frac{(2q_h+3)\left(t+q_s t_{\mathsf{Sim-Sign}}\right)}{\varepsilon-\gamma-\left(q_h+1\right)2^{-\ell_c+1}}+(T-1)\,t_{\mathsf{Update}} \end{aligned}$$

and

$$\varepsilon' = \delta\left(1-\frac{1}{\mathrm{e}}\right)^2 \text{ as long as } \varepsilon \geq \gamma+\left(q_h+1\right)2^{-\ell_c+1}.$$

We can remark this is exactly the bound of [MR02] (if $T_2=0$ in their paper, and $t_{\mathsf{Update}}=0$), which is quite normal since the existential weak forward security is very close to the classical existential unforgeability.

∎

I Analysis of our Variant of the Itkis-Reyzin Scheme

I.1 Security of the Itkis-Reyzin Scheme

In this section, we present another security analysis of the original Itkis-Reyzin scheme in [IR01], based on the forking lemma. We use a generalization of the method described in [MR02] to prove a security result more useful for a fair comparison. According to Appendix H, and more precisely to Corollary H.2, we just need to prove that the underlying identification key-evolving identification scheme is collision-intractable; informally, this means, it is hard for an adversary to find two correct transcripts (cmt, ch, rsp) and (cmt, ch', rsp') for a period $\tilde{\imath}$ such that $ch \neq ch'$, given the public key pk, the period $\tilde{\imath}$, and the secret key $sk_{\tilde{\imath}+1}$ for period $\tilde{\imath}+1$. For the IR scheme, it is straightforward to see that the identification scheme is (t',ε')-collision intractable if the strong-RSA problem is (t',ε')-hard. It is also response-unique exactly for the same reason as our scheme in Section G.3.1. Therefore, thanks to Corollary H.2, we have the following theorem:

Theorem I.1 *If the strong-RSA problem is (t',ε')-hard, then the previous scheme is $(t, q_h, q_s, \varepsilon, \delta)$-weakly-forward-secure in the random oracle model for:*

$$t \approx \frac{\left(t'-(T-1)\,t_{\mathsf{Update}}\right)\cdot\varepsilon}{4q_h+6} - q_s t_{\mathsf{Sim-Sign}}$$

as long as

$$\varepsilon \geq 2\left(\frac{(q_h+1)q_s}{2^{\ell_N-2\ell_e-2}}+\frac{(q_h+1)}{2^{\ell_e-1}}\right) \text{ and } \varepsilon' \leq \delta\left(1-\frac{1}{\mathrm{e}}\right)^2.$$

Under the assumption of Remark B.4, if we suppose $t_{\mathsf{Sim-Sign}}=0$ and $t_{\mathsf{Update}}=0$, we can say that the scheme is about $(\frac{t\varepsilon}{4q_h}, q_h, q_s, T\varepsilon)$-forward-secure if the strong-RSA problem is $(t,(1-1/e)^2/2)$-hard. This means roughly that if we want k-bits of security, and if we suppose strong-RSA is as hard as factorization, the modulus has to correspond to a security level of about $k' \approx k+\log_2(Tq_h)$ (k being an approximate solution of $2^{k'} = \frac{2^k\varepsilon}{4q_h}\frac{1}{T\varepsilon}$).

```
AlgH'(i)
001 found ← false
002 cpt ← 0
003 while found ≠ true
004    x ← H'(i‖cpt)
005    x ← x with bits (ℓ_e − 1) and 0 set
006    if isPrime(x) then
007       break
008    cpt ← cpt + 1
009 return x
```

```
AlgProgH''_{x*}(i‖cpt)
010 if HT'(i‖cpt) =⊥ then
011    y ←$ {0,1}^{ℓ_e}
012    y ← y with bits (ℓ_e − 1) and 0 set
013    if isPrime'(y) then
014       HT'(i‖cpt) ← x*(i)
015    else
016       HT'(i‖cpt) ← y
017 return HT'(i‖cpt)
```

Figure I.1: Algorithm AlgH' which simulates a random prime oracle H' using a classical random oracle H'' and algorithm $\mathsf{AlgProgH}''$ which simulates H'' such that the output of AlgH is $x^*(i)$ on input i. isPrime is a probabilistic or deterministic primality test and $\mathsf{isPrime}'$ is a deterministic one.

I.2 Random Oracle for Prime Numbers

As explained in Section 5.1, we need a random oracle for prime numbers in order to be able to generate the exponents e_i for our scheme. Here is a description of a construction of such a random oracle from a classical random oracle. This construction is close to the construction of a PRF mapping to prime numbers in [HW09].

If we have access to a classical random oracle H'' with output of length ℓ_e, we simulate H' using the algorithm AlgH' depicted in Figure I.1. It is clear that the outputs of such AlgH' is uniform over $\mathbb{P}_{\ell_e}$ the set of primes of length ℓ_e. Notice, we only force the output of H'' to be odd by setting bit 0 of the output. Furthermore, the algorithm $\mathsf{AlgProgH}''_{x^*}$ depicted in Figure I.1 simulates the random oracle H'' such that AlgH' outputs $x^*(i)$ on inputs but that H'' still has a random uniform distribution (as soon as $x^*(i)$ is uniform over $\mathbb{P}_{\ell_e}$).

Let write C the random variable equal to the number of primality tests needed in AlgH' (i.e., the final value of $cpt + 1$), if the primality tests are deterministic. According to Proposition F.2, C is a geometric random variable of parameter at least $1/(\ell_e - 1)$. So its expectation $\mathbf{E}[C]$, the average number of calls to isPrime is at most $\ell_e - 1$.

For efficiency purpose, it is necessary to use a probabilistic primality test for isPrime, such as Miller-Rabin. Let suppose the error probability of the test (i.e., the probability a composite number is considered prime) is $\varepsilon_p = 2^{-kp}$. In this case, the error probability of AlgH' for input i is at most

$$\begin{aligned}\varepsilon' &= \sum_{j=0}^{\infty} \Pr\left[\, C = j \wedge \mathsf{isPrime} \text{ has done an error on } \mathsf{H}'(i\|0), \mathsf{H}'(i\|1), \dots \text{ or } \mathsf{H}'(i\|j)\,\right] \\ &\leq \sum_{j=0}^{\infty} \Pr[\, C = j\,]\, j\, \varepsilon \;=\; \mathbf{E}[\, C\,]\, \varepsilon \;\leq\; (\ell_e - 1)\varepsilon.\end{aligned}$$

We can adapt the proof of security in Appendix C: we just replace isPrime in AlgH' in the verification in **Finalize** by a deterministic algorithm $\mathsf{isPrime}'$. This just add a term $(\ell_e - 1)\varepsilon_p$ to the final probability for the adversary to win the original game.

We can notice that it is now possible for the algorithms Sign and Update to output an incorrect value. But the probability is at most $(\ell_e - 1)\varepsilon_p$, which should be negligible.

Let us now analyze the performance of AlgH'. If we forget the probability of errors of the primality test and do not take into account the time to call H''[15], the average time of AlgH' is $(\mathbf{E}[\,C\,]-1)t_{\mathsf{isPrime-composite}} + t_{\mathsf{isPrime-prime}}$, where $t_{\mathsf{isPrime-composite}}$ is the average running time of isPrime if its input is a composite number, and $t_{\mathsf{isPrime-composite}}$ is the average running time of isPrime if its input is a prime.

[15] In practice, H'' will be implemented using a hash function which is hundred times faster than any primality test.

For Miller-Rabin test, if the input is prime, the algorithm roughly does $kp/2$ exponentiation modulo a ℓ_e-bit number with a ℓ_e-bit exponent. Otherwise, if the input is a composite number, it does fewer than 4/3 such exponentiation in average[16]. Therefore, $t_{\mathsf{isPrime-prime}} \approx kp\frac{3}{2}\,{\ell_e}^3$ and $t_{\mathsf{isPrime-composite}} \approx \frac{3}{2}\frac{4}{3}\,{\ell_e}^3$, therefore the total time is about $(\frac{3}{2}kp + 2\ell_e){\ell_e}^3$. In comparison, the time of a signature or a verification (if the e_i are stored) is the time of two exponentiation with a modulus of length ℓ_N and an exponent of length ℓ_e, that means about $2\ell_N k^2$. A practical comparison can be found in Table 5.2.

I.3 Optimizations

In this section, we analyze optimizations of the original IR scheme and see that they can be applied to our scheme too. We also propose a specific optimization for our scheme.

e_i POWER OF SMALL PRIMES. If we slightly change the e_i to be power of small primes ε_i: $e_i = \varepsilon_i^{\ell_e/\lfloor\log(\varepsilon_i)}$, we can make the generation of e_i faster since generating a small ${\ell_e}'$-bit prime ε_i is about $(\ell_e/{\ell_e}')^4$ faster than generating a ℓ_e-bit prime e_i. However, we need to change the ϕ-hiding assumption in order to be able to do the security reduction[17].

PEBBLING. We also remark that the pebbling mechanism described in [IR01] can directly be applied to our scheme.

STORING *cpt*. Another possible trade-off consists in storing the last *cpt* of AlgH' for each i, in the public and secret keys. Since $\mathbf{E}[C] \leq \ell_e - 1$, the expected size of *cpt* is $\log_2 \ell_e$ and storing them increase the size of the keys by $T\log_2 \ell_e$. For small values of T this can be useful, since this completely remove the necessity of isPrime in Sign, Ver and Update.

[16] Actually, a Miller-Rabin test should be a little faster than an exponentiation since we can stop the exponentiation before the end, in some cases.

[17] More precisely, we need e in the ϕ-assumption to be chosen as a power of a small prime number. The distribution of $N = p_1p_2$ such that e divides p_1 can be sampled in polynomial time, if we assume the extended Riemann hypothesis (Conjecture 8.4.4 of [BS96]), exactly as when e is a prime. But we are not sure the assumption actually still holds...