Tatra Mt. Math. Publ. **47** (2010), 15–29
DOI: 10.2478/v10127-010-0029-2

TATRA MOUNTAINS
Mathematical Publications

SECURITY OF SIGNATURE SCHEMES IN THE PRESENCE OF KEY-DEPENDENT MESSAGES

Madeline González Muñiz — Rainer Steinwandt

ABSTRACT. In recent years, quite some progress has been made in understanding the security of encryption schemes in the presence of key-dependent plaintexts. Here, we motivate and explore the security of a setting, where an adversary against a signature scheme can access signatures on key-dependent messages.

We propose a way to formalize the security of signature schemes in the presence of *key-dependent signatures* (KDS). It turns out that the situation is quite different from key-dependent encryption: already to achieve KDS-security under non-adaptive chosen message attacks, the use of a stateful signing algorithm is inevitable—even in the random oracle model. After discussing the connection between key-dependent signing and forward security, we present a compiler to lift any EUF-CMA secure one-time signature scheme to a forward secure signature scheme offering KDS-CMA security.

1. Introduction

Established security notions for encryption schemes like IND-CCA refer to scenarios where encrypted plaintexts do not depend on the secret key. For some scenarios—like encrypting a hard disk storing the secret decryption key—such a security model is inadequate. Here the question of secure encryption in the presence of key-dependent messages naturally arises, and in recent years, significant progress in understanding such scenarios has been made (see [BRS03], [BPS07], [HK07], [BHHO08], [HH08], [HU08], [ACPS09] for instance).

For signature schemes, scenarios with key-dependent messages seem much less understood. Although perhaps being less obvious than for key-dependent encryption, a scenario where an adversary may have access to signatures on key-dependent messages is not that far-fetched: if we grant an adversary access to the signature of a (possibly encrypted) backup of a hard disk containing the secret signing key, then this is a scenario not covered by EUF-CMA security. A natural question arises about how to combine the security definitions of

2010 Mathematics Subject Classification: 94A60.
Keywords: signature scheme, key-dependent message, forward security.

key-dependent encryption and signing to come up with a signcryption scheme that is secure in the presence of key-dependent messages, and this is explored in [Gon09]. Key-dependent signing seems also interesting in connection with *combined public key schemes* as discussed by Haber and Pinkas [HP01] or González Vasco et al. [VHS09]: here keys used for decrypting and for signing are not necessarily independent, and signing a message derived from output of the decryption algorithm may actually imply signing a key-dependent message.

Our contribution. Following the notion of key dependent message (KDM) security proposed by Black et al. [BRS03], we propose a formalization of security in the presence of key dependent signatures (KDS). As discussed in Section 3.1, for stateless signers, a natural definition—where an adversay can obtain signatures on chosen key-dependent messages—allows no secure realization, even in the random oracle model. A compiler is presented which transforms any EUF-CMA secure one-time signature scheme into a (necessarily stateful) KDS-CMA secure signature scheme, offering also forward security. In Section 4 we show that KDS-security and forward security are related, but independent security goals.

Further related work. In addition to research on forward secure signature schemes and on encryption in the presence of key-dependent messages, also, research on leakage resilient cryptography can be mentioned here. Specifically, Katz [Kat09] explores *signature schemes with bounded leakage resilience*, where an adversary has limited access to information on the secret signing key. In a sense, the focus of [Kat09] is dual to ours: The work in [Kat09] focuses on a stateless signing algorithm, i.e., the secret key is not updated. To cope with such a scenario, a bound on the total leakage is imposed. In the discussion below, the adversary could in principle expose the complete secret key bit by bit, and we need a stateful signing algorithm to prevent such attacks. In terms of modeling adversarial capabilities, we decided to allow key-dependent queries to a signing oracle, rather than a sequence of leakage functions. As here we do not aim at modeling attacks at the implementation level, like side-channel attacks, this seems a viable model.

In Faust et al.'s independent work [FKPR09] on *Leakage-Resilient Signatures*, side-channel attacks are a central motivation. Faust et al. focus on a scenario with *bounded leakage* per invocation, respectively, leakage functions with bounded range. Like in the next section, stateful signature schemes are considered, and the resulting security notion is called UF-CMLA. Faust et al. present a compiler that lifts a 3-time signature scheme to an UF-CMLA secure signature scheme that can sign a prespecified number of messages.

2. Preliminaries and definitions

As already indicated and as will be detailed below, for our purposes it is crucial to allow a stateful signing algorithm, and the subsequent definitions take this into account.

2.1. Signature schemes and existential unforgeability

We formalize a signature scheme similarly as in [GMR88]. Technically, the main difference from [GMR88] is that we consider the secret key as part of the signer's state, instead of allowing auxiliary input to the signing algorithm. Moreover, we also allow the signing algorithm to output an error symbol.

Definition 1 (Signature scheme)**.** A signature scheme S is a triple of polynomial time algorithms $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$:

- $\mathcal{K}$ is a probabilistic *key generation algorithm* which on input the security parameter 1^k returns a pair (sk, pk) of keys—a public verification key pk with matching secret signing key $sk \in \{0,1\}^*$. In case of a stateful signer, we interpret sk as initial state of the signer, i. e., all secret information of the signer is part of its state.
- $\mathcal{S}$ is a probabilistic *signing algorithm* which on input a message $M \in \{0,1\}^*$ and state sk—which in case of a stateless signer is just the secret key––returns a signature $\sigma \in \{0,1\}^*$ on M or an error symbol $\perp$. Moreover, the state value sk is updated.
- $\mathcal{V}$ is a deterministic *verification algorithm* which on input a public key pk, a message M, and a candidate signature σ for M returns true or false, indicating whether σ is a valid signature for M under the public key pk.

For pairs (sk, pk) output by $\mathcal{K}$ we require that with overwhelming probability the obvious correctness condition holds: for all messages M we have

$$\mathcal{V}_{pk}\big(M, \mathcal{S}_{sk}(M)\big) = \mathsf{true}.$$

The standard security requirement for signature schemes is EUF-CMA which stands for *existential unforgeability under adaptive chosen message attack* (cf. [GMR88]):

Definition 2 (EUF-CMA)**.** Let $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a signature scheme, and let $\mathcal{A}^{\mathrm{euf}}$ be a probabilistic polynomial time algorithm. Consider the following attack scenario:

1. Compute a key pair $(sk, pk) \xleftarrow{\$} \mathcal{K}(1^k)$, and hand pk as input to $\mathcal{A}^{\mathrm{euf}}$.
2. The adversary $\mathcal{A}^{\mathrm{euf}}$ is given unrestricted access to a signing oracle $\mathcal{O}_{\mathcal{S}}$ to run $\mathcal{S}_{sk}(\cdot)$.
3. Eventually, $\mathcal{A}^{\mathrm{euf}}$ outputs a message M and a signature σ.

Let QueriedEarlier be the event that $\mathcal{A}^{\text{euf}}$ outputs a message M that has already been queried to the signing oracle $\mathcal{O}_{\mathcal{S}}$. The *success probability* $\text{Succ}^{\text{euf}}_{\mathcal{A}} = \text{Succ}_{\mathcal{A}^{\text{euf}}}(k)$ of $\mathcal{A}^{\text{euf}}$ is defined as

$$\text{Succ}_{\mathcal{A}^{\text{euf}}} := \Pr\big[\mathcal{V}_{pk}(M,\sigma) = \mathsf{true} \text{ and } \neg\, \mathsf{QueriedEarlier}\big],$$

and we call the signature scheme S *secure in the sense of* EUF-CMA if $\text{Succ}_{\mathcal{A}^{\text{euf}}}$ is negligible for all probabilistic polynomial time adversaries $\mathcal{A}^{\text{euf}}$.

Remark 1. The above definition of EUF-CMA security carries over to one-time signature schemes in the obvious way—the only modification being that $\mathcal{A}^{\text{euf}}$ can query the signing oracle $\mathcal{O}_{\mathcal{S}}$ only once.

In particular, security in the sense of EUF-CMA does not allow an adversary to obtain signatures on key-dependent messages—like a signature on the complete secret key (state) sk. In fact, given an EUF-CMA secure signature scheme, it is easy to come up with a signature scheme that is still EUF-CMA secure, but where a single key-dependent message query breaks the security of the scheme.

2.2. Security in the presence of key-dependent signatures

Informally, a signature scheme $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ is referred to as KDS-CMA secure if it is secure despite a forger's ability to obtain signatures on arbitrary (efficiently computable) functions g of the signer's state sk. In particular, g has access to the secret key stored at the time of signing.

Definition 3 (KDS-CMA)**.** Let the triple $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a signature scheme, and let $\mathcal{A}^{\text{kds}}$ be a probabilistic polynomial time algorithm. Consider the following attack scenario:

1. Compute a key pair $(sk, pk) \xleftarrow{\$} \mathcal{K}(1^k)$, and hand pk as input to $\mathcal{A}^{\text{kds}}$.
2. The adversary $\mathcal{A}^{\text{kds}}$ is given unrestricted access to a signing oracle $\widehat{\mathcal{O}}_{\mathcal{S}}$. The oracle $\widehat{\mathcal{O}}_{\mathcal{S}}$ accepts as input a function g, represented as a boolean circuit of polynomial size, and executes the signing algorithm $\mathcal{S}$ with the current state sk and the message $g(sk)$ as input.[1]
3. Eventually, $\mathcal{A}^{\text{kds}}$ outputs a message $M \in \{0,1\}^*$ and a signature σ.

Let QueriedEarlier be the event that $\mathcal{A}^{\text{kds}}$ outputs a message M such that one of $\mathcal{A}^{\text{kds}}$'s queries g to the signing oracle $\widehat{\mathcal{O}}_{\mathcal{S}}$ evaluated to $g(sk) = M$. Then the *success probability* $\text{Succ}_{\mathcal{A}^{\text{kds}}} = \text{Succ}_{\mathcal{A}^{\text{kds}}}(k)$ of $\mathcal{A}^{\text{kds}}$ is defined as

$$\text{Succ}_{\mathcal{A}^{\text{kds}}} := \Pr\big[\mathcal{V}_{pk}(M,\sigma) = \mathsf{true} \text{ and } \neg\, \mathsf{QueriedEarlier}\big],$$

and we call the signature scheme S *secure in the sense of* KDS-CMA if $\text{Succ}_{\mathcal{A}^{\text{kds}}}$ is negligible for all probabilistic polynomial time adversaries $\mathcal{A}^{\text{kds}}$.

[1] In the random oracle model, g may invoke the random oracle.

3. Achieving KDS-CMA security

By definition, security in the sense of KDS-CMA implies security in the sense of EUF-CMA, and the question arises whether/how security in the sense of Definition 3 can be achieved.

3.1. Impossibility of KDS-CMA with a stateless signing algorithm

As a first (negative) result, we note that no signature scheme with a stateless signing algorithm can meet the security goal of KDS-CMA security.

Remark 2. Let $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a signature scheme with a stateless signing algorithm $\mathcal{S}$, i. e., the secret signing key sk is not changed by executing $\mathcal{S}$. Then the signature scheme S is not secure in the sense of KDS-CMA.

P r o o f. Let $sk = b_0, \ldots, b_{\ell-1} \in \{0,1\}^\ell$ be the bit representation of the secret key and fix $i \in \{0, \ldots, \ell - 1\}$ arbitrary. Then the adversary $\mathcal{A}$ may query $\widehat{\mathcal{O}}_{\mathcal{S}}$ for a signature on b_i and use the public verification algorithm $\mathcal{V}$ to determine if the returned signature σ satisfies $\mathcal{V}_{pk}(0, \sigma) = \mathsf{true}$ or $V_{pk}(1, \sigma) = \mathsf{true}$. Thus ℓ queries to $\widehat{\mathcal{O}}_{\mathcal{S}}$ are sufficient to extract the complete secret signing key sk, and hereafter creating a forgery is trivial. □

Despite its simplicity, the attack in the proof of Remark 2 is quite devastating, and it might not be obvious if KDS-CMA security can be achieved at all. In the next section we show that, in the random oracle model, allowing the signing algorithm to be stateful enables the derivation of a KDS-CMA secure signature scheme from any one-time EUF-CMA secure one.

3.2. From one-time EUF-CMA to KDS-CMA: a compiler

The compiler in Figure 1 uses a random oracle $H\colon \{0,1\}^* \longrightarrow \{0,1\}^k$ to transform any one-time EUF-CMA secure signature scheme into one that is KDS-CMA secure (in the random oracle model). While we do not expect this construction to be optimal from an efficiency point of view, it provides a tool to systematically construct KDS-CMA secure signature schemes.

Proposition 1. *Let $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a one-time signature scheme that is secure in the sense of* EUF-CMA. *Then the signature scheme $\widehat{S} = (\widehat{\mathcal{K}}, \widehat{\mathcal{S}}, \widehat{\mathcal{V}})$ obtained from the compiler in Figure 1 is secure in the sense of* KDS-CMA *in the random oracle model.*

P r o o f. Let $\mathcal{A}^{\mathrm{kdm}}$ be an adversary in the sense of KDS-CMA, having a non-negligible success probability in creating a forgery for the signature scheme $\widehat{S}$. Then we can construct an adversary $\mathcal{A}^{\mathrm{euf}}$ that violates EUF-CMA security of the underlying one-time signature scheme S. For doing so, we start with $\mathcal{A}^{\mathrm{euf}}$ running a simulation of $\mathcal{A}^{\mathrm{kds}}$, including a simulation of all oracles. We modify $\mathcal{A}^{\mathrm{euf}}$'s

$\widehat{\mathcal{K}}$**:** Create a key pair $(sk_0^{\mathrm{crt}}, pk_0^{\mathrm{crt}}) \xleftarrow{\$} \mathcal{K}(1^k)$, return pk_0^{crt} as public verification key and use $sk := (sk_0^{\mathrm{crt}}, \lambda, [])$ as initial state, where $[]$ is an empty list and λ the empty string.

$\widehat{\mathcal{S}}$**:** To sign the ith $(1 \leq i)$ message $M \in \{0,1\}^*$ proceed as follows:

- Create two fresh key pairs $(sk_i^{\mathrm{crt}}, pk_i^{\mathrm{crt}}) \xleftarrow{\$} \mathcal{K}(1^k), (sk_i^{\mathrm{msg}}, pk_i^{\mathrm{msg}}) \xleftarrow{\$} \mathcal{K}(1^k)$.
- Compute $\mathrm{Cert}_i := pk_i^{\mathrm{crt}} \,\|\, pk_i^{\mathrm{msg}} \,\|\, \sigma_i$ with $\sigma_i \xleftarrow{\$} \mathcal{S}_{sk_{i-1}^{\mathrm{crt}}}(pk_i^{\mathrm{crt}} \,\|\, pk_i^{\mathrm{msg}})$.
- Update the internal state $sk = \big(sk_{i-1}^{\mathrm{crt}}, sk_{i-1}^{\mathrm{msg}}, [\mathrm{Cert}_\mu]_{1 \leq \mu \leq i-1}\big)$ to $sk \leftarrow \big(sk_i^{\mathrm{crt}}, sk_i^{\mathrm{msg}}, [\mathrm{Cert}_\mu]_{1 \leq \mu \leq i}\big)$.
- Compute $s \xleftarrow{\$} \mathcal{S}_{sk_i^{\mathrm{msg}}}\big(r \parallel H(M \parallel r)\big)$, where $r \xleftarrow{\$} \{0,1\}^k$ is chosen uniformly at random.
- Return the signature $\big(r, s, [\mathrm{Cert}_\mu]_{1 \leq \mu \leq i}\big)$.

$\widehat{\mathcal{V}}$**:** On input a message M and a candidate signature $\big(r, s, [\mathrm{Cert}_\mu]_{1 \leq \mu \leq i}\big)$, output true if all of the following conditions hold. Otherwise output false:

- $\mathcal{V}_{pk_i^{\mathrm{msg}}}\big(r \parallel H(M \parallel r), s\big) = \mathsf{true}$,
- $\mathcal{V}_{pk_{\mu-1}^{\mathrm{crt}}}(pk_\mu^{\mathrm{crt}} \parallel pk_\mu^{\mathrm{msg}}, \sigma_\mu) = \mathsf{true}$ for all $1 \leq \mu \leq i$, where $\mathrm{Cert}_\mu = pk_\mu^{\mathrm{crt}} \parallel pk_\mu^{\mathrm{msg}} \parallel \sigma_\mu$.

FIGURE 1. Deriving a KDS-CMA secure signature scheme, where it is assumed that public keys pk_i^{crt}, pk_i^{msg} are represented with a fixed length encoding.

simulation strategy through a short sequence of games, the last one yielding an attack on the EUF-CMA security of the underlying one-time signature scheme S.

Game 0. This is a trivial simulation of the original attack game played by $\mathcal{A}^{\mathrm{kds}}$: The public verification key and initial secret state of the challenge for $\mathcal{A}^{\mathrm{kds}}$ are fixed by $\mathcal{A}^{\mathrm{euf}}$ by running the key generation algorithm $\widehat{\mathcal{K}}$. From here on, all needed oracles for $\mathcal{A}^{\mathrm{kds}}$ can be simulated faithfully:

Random oracle: For simulating $\mathcal{A}^{\mathrm{kds}}$'s random oracle, $\mathcal{A}^{\mathrm{euf}}$ creates an empty list L^{RO}. Then, whenever $\mathcal{A}^{\mathrm{kds}}$ queries its random oracle with a message x such that L^{RO} contains no entry of the form $(x, \cdot)$, $\mathcal{A}^{\mathrm{euf}}$ chooses a value $r_x \in \{0,1\}^k$ uniformly at random, appends the pair (x, r_x) to L^{RO} and sends r_x to $\mathcal{A}^{\mathrm{kds}}$. In case $\mathcal{A}^{\mathrm{kds}}$ queries L^{RO} a second time with the same value x, $\mathcal{A}^{\mathrm{euf}}$ returns the stored random value r_x.

Signing oracle: Knowing the initial secret key, $\mathcal{A}^{\mathrm{euf}}$ can faithfully answer queries to $\widehat{\mathcal{O}}_{\mathcal{S}}$ by simply executing $\widehat{\mathcal{S}}$ with the appropriate input and using the above simulation of the random oracle H.

Game 1. Let Collision be the event that during the simulation, $\mathcal{A}^{\mathrm{euf}}$ stores pairs (x, r_x) and $(x', r_{x'})$ in L^{RO}, where $x \neq x'$ and $r_x = r_{x'}$. Whenever the event Collision occurs, $\mathcal{A}^{\mathrm{euf}}$ gives up, without creating a successful forgery. As $\mathcal{A}^{\mathrm{kds}}$ is polynomially bounded, Collision occurs with negligible probability only, and subsequently we may assume that the event Collision does not occur.

Game 2. Let q_s be a polynomial upper bound for the number of signing queries made by $\mathcal{A}^{\mathrm{kds}}$, and let g_i be the ith function/message submitted to $\widehat{\mathcal{O}}_{\mathcal{S}}$ by $\mathcal{A}^{\mathrm{kds}}$. By pk^* we denote the public key to be attacked by $\mathcal{A}^{\mathrm{euf}}$ in the definition of EUF-CMA security. In this game $\mathcal{A}^{\mathrm{euf}}$ chooses an index $i^* \in \{0, \ldots, q_s\}$ and then, if $i^* \neq 0$, a flag $\Gamma \in \{\mathrm{crt}, \mathrm{msg}\}$ uniformly at random—for $i^* = 0$ we always set $\Gamma := \mathrm{crt}$. Now, in the simulation $\mathcal{A}^{\mathrm{euf}}$ replaces the public key $pk_{i^*}^{\Gamma}$ with the challenge public key pk^*. In case that $\mathcal{A}^{\mathrm{kds}}$ does not submit any signature queries to $\widehat{\mathcal{O}}_{\mathcal{S}}$, this modification is not detectable for $\mathcal{A}^{\mathrm{kds}}$. Similarly, answering signature queries g_i with $i < i^*$ is still possible, as the secret key sk^* associated to the challenge key pk^* is not needed here. Moreover, $\mathcal{A}^{\mathrm{euf}}$ can compute Cert_{i^*+1}:

- If $\Gamma = \mathrm{crt}$, then $\mathcal{A}^{\mathrm{euf}}$ can use its signing oracle to compute Cert_{i^*+1}.
- If $\Gamma = \mathrm{msg}$, then $\mathcal{A}^{\mathrm{euf}}$ knows $sk_{i^*}^{\mathrm{crt}}$.

Consequently, $\mathcal{A}^{\mathrm{euf}}$ is able to correctly answer all signature queries g_i with $i > i^*$, too. For the only "critical" query g_{i^*}, we consider two cases:

- If the value

$$\begin{cases} g_{i^*}\big(\big(sk^*, sk_{i^*}^{\mathrm{msg}}, [\mathrm{Cert}_\mu]_{1 \leq \mu \leq i^*}\big)\big) & \text{if } \Gamma = \mathrm{crt}, \\ g_{i^*}\big(\big(sk_{i^*}^{\mathrm{crt}}, sk^*, [\mathrm{Cert}_\mu]_{1 \leq \mu \leq i^*}\big)\big) & \text{if } \Gamma = \mathrm{msg} \end{cases} \tag{1}$$

  can be predicted (in the sense specified in Remark 3) by $\mathcal{A}^{\mathrm{kds}}$, we modify $\mathcal{A}^{\mathrm{kds}}$ to make such a prediction, therewith replacing the potentially key--dependent query g_{i^*} with a key-independent query g_{i^*}. By construction, the success probability of $\mathcal{A}^{\mathrm{kds}}$ remains non-negligible, provided it was non-negligible before.

  Remark 3. Let σ be a signature on the value output above in (1). We say that $\mathcal{A}^{\mathrm{kds}}$ can *predict* the value (1) if there exists a probabilistic polynomial time (extractor) algorithm $\mathcal{E}$ which on input the state of $\mathcal{A}^{\mathrm{kds}}$ and g_{i^*} outputs a message M such that M equals the value in (1) with non-negligible probability.

– If the value (1) can be predicted with negligible probability only, $\mathcal{A}^{\text{euf}}$ creates a key pair $(sk', pk') \xleftarrow{\$} \mathcal{K}(1^k)$, a random $r' \xleftarrow{\$} \{0,1\}^k$ and queries

$$M' \parallel r' := \begin{cases} g_{i^*}\big(\big(sk', sk_{i^*}^{\text{msg}}, [\text{Cert}_\mu]_{1\le\mu\le i^*}\big)\big) \parallel r' & \text{if } \Gamma = \text{crt}, \\ g_{i^*}\big(\big(sk_{i^*}^{\text{crt}}, sk', [\text{Cert}_\mu]_{1\le\mu\le i^*}\big)\big) \parallel r' & \text{if } \Gamma = \text{msg} \end{cases}$$

to its simulation of the random oracle. The use of sk' instead of sk^* in the evaluation of g_{i^*} cannot be noticed by $\mathcal{A}^{\text{kds}}$ unless the event Collision occurs.

The value $\mathcal{S}_{sk^*}\big(r' \parallel H(M' \parallel r')\big)$ is handed to $\mathcal{A}^{\text{kds}}$ as s-component of the signature. If $\Gamma = \text{msg}$, this value can be obtained from $\mathcal{A}^{\text{euf}}$'s signing oracle, otherwise $\mathcal{A}^{\text{euf}}$ can compute this value itself.

Note that the adversary is only forced to predict during the critical query g_{i^*}. For $i \neq i^*$, $\mathcal{A}^{\text{euf}}$ can faithfully answer all key-dependent queries since it has generated the keys during the simulation. Where this is not the case, the adversary $\mathcal{A}^{\text{kds}}$ may have to predict a polynomial number of values, which in turn may make its success probability negligible, depending on the correctness of the predictions.

Game 3. Let $\big(M, (r, s, [\text{Cert}_\mu]_{1\le\mu\le i})\big)$ be a successful forgery returned by $\mathcal{A}^{\text{kds}}$. With probability $\geq 1/(2q_s+1)$ (minus some negligible function), this forgery includes a signature on a message that can be verified successfully with $pk_{i^*}^{\Gamma} = pk^*$ and one of the following holds—in all other cases the simulation of $\mathcal{A}^{\text{kds}}$ needs to be restarted.

– The list $[\text{Cert}_\mu]_{1\le\mu\le i}$ contains a $\text{Cert}_\mu = pk_\mu^{\text{crt}} \parallel pk_\mu^{\text{msg}} \parallel \sigma_\mu$, where $\mathcal{V}_{pk^*}(pk_\mu^{\text{crt}} \parallel pk_\mu^{\text{msg}}, \sigma_\mu) = \mathsf{true}$ and $pk_\mu^{\text{crt}} \parallel pk_\mu^{\text{msg}}$ has not been submitted to $\mathcal{A}^{\text{euf}}$'s signing oracle. Consequently, $\mathcal{A}^{\text{euf}}$ has created a valid forgery.
– We have $\mathcal{V}_{pk^*}\big(r \parallel H(M \parallel r), s\big) = \mathsf{true}$ and $r \parallel H(M \parallel r)$ has not been submitted to $\mathcal{A}^{\text{euf}}$'s signing oracle. Consequently, $\mathcal{A}^{\text{euf}}$ has created a valid forgery.

Summarizing, we see that if $\mathcal{A}^{\text{kds}}$'s forgery is valid with non-negligible probability, the same holds for $\mathcal{A}^{\text{euf}}$'s forgery. □

4. KDS-CMA and forward security

In forward security, so-called *key-evolving signature schemes* are considered, and compromise of the current secret key does not enable an adversary to forge signatures pertaining to the past. Signatures for messages signed in the past under a fixed public key are valid even if the current secret key is exposed. Furthermore, the adversary cannot forge signatures with a "date" prior to key

exposure. In this section we discuss connections between KDS-CMA and forward security—to the latter, we will refer to as FWD-CMA.

4.1. Key-evolving signature schemes and forward security

We adopt some terminology from Bellare and Miner [BM99a], [BM99b], starting by defining a key-evolving signature scheme.

Definition 4 (Key-evolving signature scheme)**.** A key-evolving signature scheme S^f is a quadruple of polynomial time algorithms $S = (\mathcal{K}^f, \mathcal{U}^f, \mathcal{S}^f, \mathcal{V}^f)$:

1. $\mathcal{K}^f$ is a probabilistic *key generation algorithm* which on input the security parameter 1^k, the total number of time periods $T \in \mathbb{N}$ (and possibly other parameters) returns a pair (sk_0, pk) of keys—a public verification key pk with matching (base) secret signing key sk_0.
2. $\mathcal{U}^f$ is a deterministic *secret key update algorithm* which takes as input the secret signing key sk_{j-1} of the previous time period $j-1$ and returns the secret signing key sk_j for time period j.
3. $\mathcal{S}^f$ is a probabilistic *signing algorithm* that on input a message $M \in \{0,1\}^*$ and the secret signing key sk_j of the current time period j returns a signature $\langle j, \zeta \rangle \xleftarrow{\$} \mathcal{S}^f_{sk_j}(M)$ for M for time period $j \in \mathbb{N}$ or returns an error symbol $\perp$.
4. $\mathcal{V}^f$ is a deterministic *verification algorithm* which on input a public key pk, a message M, and a signature $\langle j, \zeta \rangle$ returns true or false, indicating whether the signature is accepted or rejected, respectively.

We may assume that sk_j stores the value j itself for period $j \in \{1, \ldots, T\}$ as well as the total number T of time periods. Further on, we adopt the convention that sk_{T+1} is the empty string and that $\mathcal{U}^f(sk_T)$ returns sk_{T+1}. Both the current time period j and the total number of time periods T are publicly known and accessible to an adversary $\mathcal{A}^{\text{fwd}}$ along with the attacked public key pk. The actual attack game used to define forward security of a key-evolving signature scheme involves three stages: the chosen message attack phase (*cma*), the break-in phase (*breakin*), and the forgery phase (*forge*).

Definition 5 (FWD-CMA)**.** Let $S^f = (\mathcal{K}^f, \mathcal{U}^f, \mathcal{S}^f, \mathcal{V}^f)$ be a key-evolving signature scheme, and let $\mathcal{A}^{\text{fwd}}$ be a probabilistic polynomial time algorithm. Consider the following attack scenario:

1. ***CMA phase***
   Set $j \leftarrow 0$, and generate a key pair $(sk_0, pk) \xleftarrow{\$} \mathcal{K}^f(1^k, \ldots, T)$.[2]
   repeat
   $j \leftarrow j + 1$; $sk_j \leftarrow \mathcal{U}^f(sk_{j-1})$

[2]Here, '...' indicates that further auxiliary input parameters may be present.

$d \xleftarrow{\$} \mathcal{A}^{\mathrm{fwd}\,\mathcal{O}^j_{S^f}}(\mathsf{cma}, pk)$
until $(d = \mathsf{breakin})$ or $(j = T)$
if $d \neq \mathsf{breakin}$ and $j = T$
then $j = T + 1$
end if

2. ***Breakin phase***
   The adversary $\mathcal{A}^{\mathrm{fwd}}$ is handed the current secret key sk_j.
3. ***Forge phase***
   Eventually, $\mathcal{A}^{\mathrm{fwd}}$ outputs a message M and a signature $\langle b, \zeta\rangle$ with $b < j$.

Let QueriedEarlier be the event that $\mathcal{A}^{\mathrm{fwd}}$ outputs a message M that has already been queried to a signing oracle $\mathcal{O}^j_{S^f}$. The *success probability* $\mathrm{Succ}_{\mathcal{A}^{\mathrm{fwd}}} = \mathrm{Succ}_{\mathcal{A}^{\mathrm{fwd}}}(1^k, \ldots, T)$ of $\mathcal{A}^{\mathrm{fwd}}$ is defined as

$$\mathrm{Succ}_{\mathcal{A}^{\mathrm{fwd}}} := \Pr\Big[\mathcal{V}^f_{pk}\big(M, \langle b, \zeta\rangle\big) = \mathsf{true} \ \text{ and } \ \neg\,\mathsf{QueriedEarlier}\Big],$$

and we call the signature scheme S^f *forward-secure* if $\mathrm{Succ}_{\mathcal{A}^{\mathrm{fwd}}}$ is negligible (in k) for all probabilistic polynomial time adversaries $\mathcal{A}^{\mathrm{fwd}}$.

The process in Definition 5 is strictly ordered in that once an adversary gives up the signing oracle for sk_j, it cannot obtain access to that oracle again. At some point, the adversary $\mathcal{A}^{\mathrm{fwd}}$ decides to use its break-in privilege and is returned the current secret key sk_j. To be successful, $\mathcal{A}^{\mathrm{fwd}}$ must forge a signature under sk_b for some $b < j$ and new message M.

Remark 4. By definition, a FWD-CMA secure scheme allows an adversary $\mathcal{A}^{\mathrm{fwd}}$ to submit a polynomial number of queries to its signing oracle within a single time period j. Thus, in the presence of key-dependent messages, an attack as presented in the proof of Remark 2 may reveal the complete secret key, before an update of the secret key occurs. In other words, security in the sense of FWD-CMA does not imply strong security guarantees in the presence of key--dependent messages.

Contrasting the above negative statement, after applying some technical modifications to obtain a syntactically correct key-evolving signature scheme, the compiler in Figure 1 (which was designed to achieve KDS-CMA security) can be used to lift an EUF-CMA secure one-time signature scheme S to a forward secure key-evolving signature scheme S^f.

4.2. The compiler revisited: from one-time EUF-CMA to FWD-CMA

Figure 2 summarizes the necessary small changes to the compiler in Figure 1. The time periods are included in the state, the signature, and the certificates.

Using Definitions 4 and 5, we can show that the scheme derived in Figure 2 is FWD-CMA secure by adapting the proof of Proposition 1 accordingly:

$\mathcal{K}^f$**:** Create a key pair $(sk_{-1}^{\text{crt}}, pk_{-1}^{\text{crt}}) \xleftarrow{\$} \mathcal{K}(1^k)$, return pk_{-1}^{crt} as public verification key and use $sk := (0, sk_{-1}^{\text{crt}}, \lambda, [])$ as initial state, where [] is an empty list and λ the empty string.

$\mathcal{U}^f$**:** On input of the state $sk = \big(j, sk_i^{\text{crt}}, sk_i^{\text{msg}}, [\text{Cert}_\mu]_{0\le\mu\le i}\big)$, if $j = -1$, then we create two fresh key pairs $(sk_0^{\text{crt}}, pk_0^{\text{crt}}) \xleftarrow{\$} \mathcal{K}(1^k)$, $(sk_0^{\text{msg}}, pk_0^{\text{msg}}) \xleftarrow{\$} \mathcal{K}(1^k)$ and update the internal state to $sk \leftarrow \big(0, sk_0^{\text{crt}}, sk_0^{\text{msg}}, [\text{Cert}_0]\big)$. Else, the internal state becomes $sk \leftarrow \big(j+1, sk_i^{\text{crt}}, sk_i^{\text{msg}}, [\text{Cert}_\mu]_{0\le\mu\le i}\big)$ leaving the secret keys and certificates the same.

$\mathcal{S}^f$**:** To sign the ith $(1 \le i)$ message $M \in \{0,1\}^*$ proceed as follows:

- Create two fresh key pairs $(sk_i^{\text{crt}}, pk_i^{\text{crt}}) \xleftarrow{\$} \mathcal{K}(1^k)$, $(sk_i^{\text{msg}}, pk_i^{\text{msg}}) \xleftarrow{\$} \mathcal{K}(1^k)$.
- Set $\text{Cert}_i := j \parallel pk_i^{\text{crt}} \parallel pk_i^{\text{msg}} \parallel \zeta_i$ with $\zeta_i \xleftarrow{\$} \mathcal{S}_{sk_{i-1}^{\text{crt}}}(j \parallel pk_i^{\text{crt}} \parallel pk_i^{\text{msg}})$, where j is the current time period.
- Update the internal state $sk = \big(j, sk_{i-1}^{\text{crt}}, sk_{i-1}^{\text{msg}}, [\text{Cert}_\mu]_{0\le\mu\le i-1}\big)$ to $sk \leftarrow \big(j, sk_i^{\text{crt}}, sk_i^{\text{msg}}, [\text{Cert}_\mu]_{0\le\mu\le i}\big)$.
- Compute $s \xleftarrow{\$} \mathcal{S}_{sk_i^{\text{msg}}}\big(r \parallel H(M \parallel r)\big)$, where $r \xleftarrow{\$} \{0,1\}^k$ is chosen uniformly at random.
- Return the signature $\big\langle j, \big(r, s, [\text{Cert}_\mu]_{0\le\mu\le i}\big)\big\rangle$.

$\mathcal{V}^f$**:** On input a message M and a candidate signature $\big\langle j, \big(r, s, [\text{Cert}_\mu]_{0\le\mu\le i}\big)\big\rangle$, output true if all of the following conditions hold. Otherwise output false:

- $\mathcal{V}_{pk_i^{\text{msg}}}\big(r \parallel H(M \parallel r), s\big) = \text{true}$.
- $\mathcal{V}_{pk_{\mu-1}^{\text{crt}}}(j \parallel pk_\mu^{\text{crt}} \parallel pk_\mu^{\text{msg}}, \zeta_\mu) = \text{true}$ for all $0 \le \mu \le i$, where $\text{Cert}_\mu = j \parallel pk_\mu^{\text{crt}} \parallel pk_\mu^{\text{msg}} \parallel \zeta_\mu$.

FIGURE 2. Forward-secure modification of the KDS-CMA compiler from Figure 1, with time periods $j \in \{1, \ldots, T\}$ being understood as being represented with a fixed length encoding.

PROPOSITION 2. *Let $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a one-time signature scheme that is secure in the sense of* EUF-CMA. *Then the key-evolving signature scheme $S^f = (\mathcal{K}^f, \mathcal{U}^f, \mathcal{S}^f, \mathcal{V}^f)$ obtained from the compiler in Figure 2 is secure in the sense of* FWD-CMA *in the random oracle model.*

Proof. Let $\mathcal{A}^{\text{fwd}}$ be an adversary in the sense of FWD-CMA, having a non-negligible success probability in creating a forgery for the signature scheme S^f given in Figure 2. Then we can construct an adversary $\mathcal{A}^{\text{euf}}$ that violates EUF-CMA security of the underlying one-time signature scheme S.

CMA phase. We start with $\mathcal{A}^{\text{euf}}$ running a simulation of $\mathcal{A}^{\text{fwd}}$, and adapt *Game 0* in the proof of Proposition 1 in the obvious way: replace adversary $\mathcal{A}^{\text{kds}}$ with $\mathcal{A}^{\text{fwd}}$, signature scheme $\widehat{S}$ with S^f, and signing oracle $\widehat{\mathcal{O}}_{\mathcal{S}}$ with $\mathcal{O}^j_{\mathcal{S}^f}$. The public verification key and initial secret key of the challenge for $\mathcal{A}^{\text{fwd}}$ are fixed by $\mathcal{A}^{\text{euf}}$ by running $\mathcal{K}^f$. Likewise, we replace adversary $\mathcal{A}^{\text{kds}}$ with $\mathcal{A}^{\text{fwd}}$ in *Game 1*, and denote by q_s a polynomial upper bound for the number of signing queries made by $\mathcal{A}^{\text{fwd}}$ and by $\Gamma \in \{\text{crt}, \text{msg}\}$ a randomly chosen flag.

Let M_i be the ith message submitted to $\mathcal{O}^j_{\mathcal{S}^f}$ by $\mathcal{A}^{\text{fwd}}$. By pk^* we denote the public key to be attacked by $\mathcal{A}^{\text{euf}}$ in the definition of EUF-CMA security. Analogously as in *Game 2*, $\mathcal{A}^{\text{euf}}$ selects an index $i^* \in \{-1, \ldots, q_s\}$ uniformly at random, and in the simulation replaces the public key $pk^{\Gamma}_{i^*}$ with the challenge public key pk^*. Answering signature queries M_i with $i \neq i^*$ is possible, as the relevant secret keys $sk^{\text{crt}}_i, sk^{\text{msg}}_i$ are known to $\mathcal{A}^{\text{euf}}$. Moreover, $\mathcal{A}^{\text{euf}}$ can use its own signing oracle to compute a valid signature of M_{i^*}. This enables $\mathcal{A}^{\text{euf}}$ to correctly answer all signature queries M_i for all time periods j.

Breaking phase. At any time interval j, $\mathcal{A}^{\text{fwd}}$ can output a special value breakin and obtain the current secret key $\big(j, sk^{\text{crt}}_i, sk^{\text{msg}}_i, [\text{Cert}_\mu]_{0\leq\mu\leq i}\big)$, but must create a forgery using an index $b < j$ to be successful. When $i \neq i^*$, $\mathcal{A}^{\text{euf}}$ can correctly reveal the secret key sk_i. In case $i = i^*$, however, $\mathcal{A}^{\text{euf}}$ gives up without creating a forgery. The probability that $\mathcal{A}^{\text{fwd}}$ chooses $pk^{\Gamma}_{i^*}$ at target of its forgery—and consequently is not handed $sk^{\Gamma}_{i^*}$—is $\geq 1/(2q_s + 3)$, and therefore non-negligible.

Forge phase. Since the probability that $i = i^*$ is at least $1/(2q_s+3)$, if $\mathcal{A}^{\text{fwd}}$ can forge with non-negligible probability p, then $p/(2q_s + 3)$ is still non-negligible, since q_s is polynomial in the security parameter. If $\mathcal{A}^{\text{fwd}}$ does not use $pk^{\Gamma}_{i^*}$ in its forgery, the simulation needs to be restarted. Otherwise, $\mathcal{A}^{\text{fwd}}$ outputs message M with verifiable signature $\big\langle b, (r, s, [\text{Cert}_\mu]_{0\leq\mu\leq i})\big\rangle$ and one of the following cases holds:

- The list $[\text{Cert}_\mu]_{0\leq\mu\leq i}$ contains a $\text{Cert}_\mu = b \parallel pk^{\text{crt}}_\mu \parallel pk^{\text{msg}}_\mu \parallel \sigma_\mu$, where $\mathcal{V}_{pk^*}(b \parallel pk^{\text{crt}}_\mu \parallel pk^{\text{msg}}_\mu, \sigma_\mu) = \text{true}$ and $b \parallel pk^{\text{crt}}_\mu \parallel pk^{\text{msg}}_\mu$ has not been submitted to $\mathcal{A}^{\text{euf}}$'s signing oracle. Consequently, $\mathcal{A}^{\text{euf}}$ has created a valid forgery.
- We have $\mathcal{V}_{pk^*}\big(r \parallel H(M \parallel r), s\big) = \text{true}$ and $r \parallel H(M \parallel r)$ has, with overwhelming probability, not been submitted to $\mathcal{A}^{\text{euf}}$'s signing oracle. Consequently, $\mathcal{A}^{\text{euf}}$ has created a valid forgery.

Hence, with non-negligible probability $\mathcal{A}^{\text{fwd}}$'s forgery is also valid for $\mathcal{A}^{\text{euf}}$. □

On a simple long signature solution to achieve FWD-CMA. In [BM99b, Section 3.3] Bellare and Miner describe a construction to achieve forward security given an EUF-CMA secure signature. The resulting signature scheme

$S^\ell = (\mathcal{K}^\ell, \mathcal{U}^\ell, \mathcal{S}^\ell, \mathcal{V}^\ell)$ can be summarized as in Figure 3. We note that this scheme is not secure in the sense of FWD-CMA, however:

$\mathcal{K}^\ell$**:** Create a key pair $(sk_0, pk_0) \xleftarrow{\$} \mathcal{K}$, return $pk := pk_0$ as public verification key and use $sk := (sk_0, [])$ as initial state, where [] is an empty list.

$\mathcal{U}^\ell$**:** To update the secret state sk_{i-1} for $1 \le i \le T$ proceed as follows:

- Create a fresh key pair $(sk_i, pk_i) \xleftarrow{\$} \mathcal{K}(1^k)$.
- Compute $\mathrm{Cert}_i := (pk_i, \sigma_i)$, where $\sigma_i \xleftarrow{\$} \mathcal{S}_{sk_{i-1}}(i \parallel pk_i)$.
- Update the internal state $sk = \big(sk_{i-1}, [\mathrm{Cert}_\mu]_{1 \le \mu \le i-1}\big)$ to $sk \leftarrow \big(sk_i, [\mathrm{Cert}_\mu]_{1 \le \mu \le i}\big)$.

$\mathcal{S}^\ell$**:** To sign during the ith $(1 \le i \le T)$ time period a message $M \in \{0,1\}^*$, proceed as follows:

- Compute $s \xleftarrow{\$} \mathcal{S}_{sk_i}(M)$.
- Return the signature $\big(s, [\mathrm{Cert}_\mu]_{1 \le \mu \le i}\big)$.

$\mathcal{V}^\ell$**:** On input a message M and a candidate signature $\big(s, [\mathrm{Cert}_\mu]_{1 \le \mu \le i}\big)$, output true if all of the following conditions hold. Otherwise output false:

- $\mathcal{V}_{pk_i}(M, s) = \mathsf{true}$
- $\mathcal{V}_{pk_{\mu-1}}(i \parallel pk_\mu, \sigma_\mu) = \mathsf{true}$ for all $1 \le \mu \le i$, where $\mathrm{Cert}_\mu = (pk_\mu, \sigma_\mu)$.

FIGURE 3. Long signatures.

Remark 5. Let $S^\ell = (\mathcal{K}^\ell, \mathcal{U}^\ell, \mathcal{S}^\ell, \mathcal{V}^\ell)$ be the signature scheme described in Figure 3. Then S^ℓ is not secure in the sense of FWD-CMA.

P r o o f. Running the key generation algorithm $\mathcal{K}^\ell$, our adversary $\mathcal{A}^{\mathrm{fwd}}$ creates a key pair (sk^*, pk^*) and requests a signature on the *message* $i \parallel p^*$ during period $i-1$. The signature is stored by $\mathcal{A}^{\mathrm{fwd}}$ as σ_i, where $\mathrm{Cert}_i := (pk_i, \sigma_i)$. For an arbitrary message M, now $\mathcal{A}^{\mathrm{fwd}}$ can use $sk_i = sk^*$ to compute $s \xleftarrow{\$} \mathcal{S}_{sk_i}(M)$. Hence, $\big(s, [\mathrm{Cert}_\mu]_{1 \le \mu \le i}\big)$ is a valid forged signature on M as verified by $\mathcal{V}^\ell$. □

The adversary $\mathcal{A}^{\mathrm{fwd}}$ succeeds because the signatures requested can be used later on as certificates in the forgery. Therefore, we can avoid this attack if we append a 1 in front of any public key pk_i to be used in a certificate, as well as append a 0 in front of any message M to be signed. More specifically, in Figure 3, we would write $\sigma_i \xleftarrow{\$}, \mathcal{S}_{sk_{i-1}}(1 \parallel i \parallel pk_i)$ in the *secret key update algorithm* $\mathcal{U}^\ell$, and $s \xleftarrow{\$} \mathcal{S}_{sk_i}(0 \parallel M)$ in the *signing algorithm* $\mathcal{S}^\ell$—with the analogous modifications in the verification algorithm.

5. Conclusion

Given an existentially unforgeable one-time signature scheme, the construction we presented yields a signature scheme offering strong guarantees in the presence of key-dependent messages. Especially, if we are willing to make stronger assumptions than the availability of a one-time signature, the efficiency of our compiler is not completely satisfying, and exploring alternative constructions seems worthwhile. For instance, techniques as used for aggregate signatures might be attractive to reduce the size of signatures. Also, the feasibility of tree-based constructions (as in [BM99b], [FKPR09]) lends itself as a natural topic for further research. Finally, from a practical point of view, it appears desirable to explore in more detail which guarantees *can* be achieved with a "stateless" signing algorithm, i.e., if the signer's secret key cannot be updated.

Acknowledgements. We thank María Isabel González Vasco for valuable discussions and an anonymous reviewer for helpful comments.

REFERENCES

[ACPS09] APPLEBAUM, B.—CASH, D.—PEIKERT, C.—SAHAI, A.: *Fast cryptographic primitives and circular-secure encryption based on hard learning problems*, in: Advances in Cryptology—CRYPTO '09 (S. Halevi, ed.), 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, 2009. Lecture Notes in Comput. Sci., Vol. 5677, Springer, Berlin, 2009, pp. 595–618.

[BHHO08] BONEH, D.—HALEVI, S.—HAMBURG, M.—OSTROVSKY, R.: *Circular-secure encryption from decision Diffie-Hellman*, in: Advances in Cryptology––CRYPTO '08 (D. Wagner, ed.), 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, 2008, Lecture Notes in Comput. Sci., Vol. 5157, Springer, Berlin, 2008, pp. 108–125.

[BM99a] BELLARE, M.—MINER, S. K.: *A forward-secure digital signature scheme*, in: Advances in Cryptology—CRYPTO '99, Lecture Notes in Comput. Sci., Vol. 1666, Springer, Berlin, 1999, pp. 431–448.

[BM99b] BELLARE, M.—MINER, S. K.: *A forward-secure digital signature scheme*, `http://cseweb.ucsd.edu/~mihir/papers/fsig.html`, July, 1999, Full version of [BM99a].

[BPS07] BACKES, M.—PFITZMANN, B.—SCEDROV, A.: *Key-dependent message security under active attacks–BRSIM/UC-soundness of symbolic encryption with key cycles*, in: CSF '07, Proc. of the 20th IEEE Computer Security Foundations Symposium, IEEE Computer Society, Washington, DC, USA, 2007, pp. 112–124, `http://dx.doi.org/10.1109/CSF.2007.23`.

[BRS03] BLACK, J.—ROGAWAY, P.—SHRIMPTON, T.: *Encryption-scheme security in the presence of key-dependent messages*, in: SAC '02—Selected Areas in Cryptography (K. Nyberg et al., eds.), 9th Annual International Workshop, St. John's, Newfoundland, Canada, 2002, Lecture Notes in Comput. Sci., Vol. 2595, Springer, Berlin, 2003, pp. 62–75.

[FKPR09] FAUST, S.—KILTZ, E.—PIETRZAK, K.—ROTHBLUM, G.: *Leakage-resilient signatures*, Cryptology ePrint Archive: Report 2009/282, June, 2009, http://eprint.iacr.org/2009/282.

[GMR88] GOLDWASSER, S.—MICALI, S.—RIVEST, R. L.: *A digital signature scheme secure against adaptive chosen-message attacks*, SIAM J. Comput. **17** (1988), 281–308.

[Gon09] GONZALEZ, M: *Cryptography in the Presence of Key-Dependent Messages*. Ph.D. Thesis, Florida Atlantic University, December 2009, http://brain.math.fau.edu/Gonzalez/dissertation.pdf.

[HH08] HAITNER, I.—HOLENSTEINY, T.: *On the (im)possibility of key dependent encryption*, in: TCC '09—Theory of Cryptography (O. Reingold, ed.), 6th Theory of Cryptography Conference, San Francisco, CA, USA, 2009, Lecture Notes in Comput. Sci., Vol. 5444, Springer, Berlin, 2009, pp. 202–219.

[HK07] HALEVI, S.—KRAWCZYK, H.: *Security under key-dependent inputs*, in: Proc. of the 14th ACM Conference on Computer and Communications Security—CCS '07 (P. Ning et al., eds.), Alexandria, Virginia, USA, 2007, ACM, New York, NY, USA, 2007, pp. 466–475, http://doi.acm.org/10.1145/1315245.1315303.

[HP01] HABER, S.—PINKAS, B.: *Securely combining public-key cryptosystems*, in: CCS '01—Computer and Communications Security (P. Samarati, ed.), 8th ACM Conference, Philadelphia, PA, USA, 2001, ACM, New York, 2001, pp. 215–224.

[HU08] HOFHEINZ, D.—UNRUH, D.: *Towards key-dependent message security in the standard model*, in: EUROCRYPT '08—Advances in Cryptology (N. Smart, ed.), 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, 2008, Lecture Notes in Comput. Sci., Vol. 4965, Springer, Berlin, 2008, pp. 108–126.

[Kat09] KATZ, J.: *Signature schemes with bounded leakage resilience*, Cryptology ePrint Archive: Report 2009/220, May, 2009, http://eprint.iacr.org/2009/220.

[VHS09] GONZÁLEZ VASCO, M. I.—HESS, F.—STEINWANDT, R.: *Combined (identity--based) public key schemes*, Cryptology ePrint Archive: Report 2008/466, February, 2009, http://eprint.iacr.org/2008/466.

Received August 28, 2009

Madeline González Muñiz
Cybernetica AS
Akadeemia tee 21
EE–12618 Tallinn
ESTONIA
E-mail: madeline.gonzalez@cyber.ee

Rainer Steinwandt
Department of Mathematical Sciences
Florida Atlantic University
777 Glades Road
Boca Raton, FL 33431
U.S.A.
E-mail: rsteinwa@fau.edu