Confined Guessing: New Signatures From Standard Assumptions

Florian Böhl*[1], Dennis Hofheinz[1], Tibor Jager†[2], Jessica Koch‡[1], and Christoph Striecks[1]

[1]Karlsruhe Institute of Technology, Germany,
{florian.boehl,dennis.hofheinz,jessica.koch,christoph.striecks}@kit.edu
[2]Ruhr-Universität Bochum, Germany, tibor.jager@rub.de

Abstract

We put forward a new technique to construct very efficient and compact signature schemes. Our technique combines several instances of an only mildly secure signature scheme to obtain a fully secure scheme. Since the mild security notion we require is much easier to achieve than full security, we can combine our strategy with existing techniques to obtain a number of interesting new (stateless and fully secure) signature schemes. Concretely, we get:

- A scheme based on the computational Diffie-Hellman (CDH) assumption in pairing-friendly groups. Signatures contain $\mathbf{O}(1)$ and verification keys $\mathbf{O}(\log k)$ group elements, where k is the security parameter. Our scheme is the first CDH-based scheme with such compact verification keys.
- A scheme based on the (non-strong) RSA assumption in which both signatures and verification keys contain $\mathbf{O}(1)$ group elements. Our scheme is significantly more efficient than existing RSA-based schemes.
- A scheme based on the Short Integer Solutions (SIS) assumption. Signatures contain $\mathbf{O}(\log(k) \cdot m)$ and verification keys $\mathbf{O}(n \cdot m)$ $\mathbb{Z}_p$-elements, where p may be polynomial in k, and n, m denote the usual SIS matrix dimensions. Compared to state-of-the-art SIS-based schemes, this gives very small verification keys, at the price of slightly larger signatures.

In all cases, the involved constants are small, and the arising schemes provide significant improvements upon state-of-the-art schemes. The only price we pay is a rather large (polynomial) loss in the security reduction. However, this loss can be significantly reduced at the cost of an additive term in signature and verification key size.

Keywords: digital signatures, CDH assumption, pairing-friendly groups, RSA assumption, SIS assumption.

1 Introduction

Generic and non-generic signature schemes. Digital signature schemes can be built from any one-way function [24, 27, 28]. However, this generic construction is not particularly efficient. (For instance, each signature contains $\mathbf{O}(k^2)$ preimages.) One could hope that for *concrete* assumptions (such as the RSA or Diffie-Hellman-related assumptions), it is possible to derive much more efficient schemes.

Tree-based signatures. But while indeed there exists a variety of efficient signature schemes from concrete assumptions, there are surprisingly few different technical paradigms. For instance, early RSA-based signature schemes (such as [16, 11, 12]) follow a tree-based approach,

*Supported by MWK grant "MoSeS".

†Part of work performed while at KIT and supported by DFG grant GZ HO 4534/2-1.

‡Supported by BMBF project "KASTEL".

much like the generic construction from [27, 28]. Also later schemes (e.g., [13] and its variants [14, 22, 17], or [20]) can at least be seen as heavily inspired by earlier tree-based schemes. For instance, [13] can be seen as a more efficient variant of the scheme from [12] with an extremely flat tree, at the price of a stronger computational assumption. On the other hand, the scheme from [20] can be viewed as a tree-based scheme in which signatures are aggregated and thus become very compact.

Partitioning strategies. A second class of signature schemes tries to enable a "partitioning strategy" during the security proof (e.g., [9, 34, 17, 6]). (This means that during the security proof, the set of messages is partitioned into those that can be signed by the simulator, and those that cannot.) At least outside of the random oracle model, currently known instantiations of partitioning strategies rely on certain algebraic structures, and lead to comparatively large public keys.

More specific schemes. Finally, there are a number of very efficient signature schemes (e.g., [5, 33]) with specific requirements. For instance, the scheme of [5] relies on a specific (and somewhat nonstandard) computational assumption, and the scheme of [33] inherently requires a decisional (as opposed to a search) assumption. While we focus on the standard model, we note that there are also very efficient schemes with a heuristic proof, i.e., a proof in the random oracle model (e.g., [2]).

1.1 Our contribution

In this work, we present a new paradigm for the construction of efficient signature schemes from standard computational assumptions.

The technical difficulty. We believe that one of the main difficulties in constructing signature schemes is the following. Namely, in the standard security experiment for digital signatures, an adversary A wins if it generates a signature for a (fresh) message of his own choice. If we use A in a straightforward way as a problem-solver in a security reduction to a computational assumption, A itself may select which instance of the particular problem it is going to solve (by choosing the forgery message). Note that we cannot simply guess which instance A is going to solve, since there usually will be superpolynomially many possible messages (and thus problem instances).[1]

Our main idea. We now explain the main idea behind our approach. As a stepping stone, we introduce a very mild form of security for signature schemes that is much easier to achieve than full (i.e., standard) security. Intuitively, the mild security experiment forces an adversary to essentially commit to a crucial part of the forgery before even seeing the verification key. During a security reduction, this allows to embed a computational problem into the verification key that is tailored specifically to the adversary's forgery. In particular, we do not have to rely on strong assumptions to achieve mild security. Indeed, we present very efficient mildly secure schemes based on the computational Diffie-Hellman (CDH) assumption (in pairing-friendly groups), the RSA assumption, and the Short Integer Solutions (SIS) assumption. These constructions are basically stripped-down versions of known (fully secure) stateful [21] and stateless [6] schemes.

The heart of our work is a very simple and efficient construction of a fully secure signature scheme from $\log(k)$ instances of a mildly secure one. Note that we only use *logarithmically many* mildly secure instances. In contrast, the related – but technically very different – prefix-guessing technique of Hohenberger and Waters [20, 27, 7], uses k instances of a "less secure" scheme. Furthermore, the signature schemes that result from our transformation can often be optimized (e.g., using aggregation of signatures or verification keys). Concretely, if we use our

[1] There are more clever ways of embedding a computational problem into a signature scheme (e.g., partitioning [9, 34]). These techniques however usually require special algebraic features such as homomorphic properties or pairing-friendly groups. For instance, partitioning is not known to apply in the (standard-model) RSA setting.

transformation on the mildly secure schemes mentioned above, and optimize the result, we end up with extremely efficient new signature schemes from standard assumptions.

More on our techniques. We now explain our transformation in a little more detail. We start out with a *tag*-based signature scheme that satisfies only a very mild form of security. In a tag-based signature scheme, signatures carry a tag t that can be chosen freely during signature time. In our mild security experiment, the adversary A must initially (i.e., non-adaptively) specify all messages M_i it wants signed, along with corresponding *pairwise different* tags t_i. After receiving the verification key and the requested signatures, A is then expected to forge a signature for a fresh message M^*, but with respect to a "recycled" tag $t^* = t_i$ (that was already used for one of the initially signed messages).

Mild security thus forces A to choose the tag t^* of his forgery from a small set $\{t_i\}_i$ of possible tags. In a security proof, we can simply guess $t^* = t_i$ with significant probability, and embed a computational problem that is specifically tailored towards t^* into the verification key.[2] How this embedding is done depends on the specific setting; for instance, in the CDH setting, t^* can be used to program a Boneh-Boyen hash function [4]. In fact, Hohenberger and Waters [21] use such a programming to construct a very efficient *stateful* signature scheme.

A signature in our fully secure scheme consists of $\log(k)$ signatures $(\sigma_i)_{i=1}^{\log(k)}$ of a mildly secure scheme. (In some cases, these signatures can be aggregated.) The i-th signature component σ_i is created with tag chosen as uniform 2^i-bit string. Hence, tag-collisions (i.e., multiply used tags) are likely to occur after a few signatures in instances with small i, while instances with larger i will almost never have tag-collisions.

We will reduce the (full) security of the new scheme *generically* to the mild security of the underlying scheme. When reducing a concrete (full) adversary B to a mild adversary A, we will first single out an instance i^* such that (a) the set of all tags is polynomially small (so we can guess the i^*-th challenge tag $t^*_{i^*}$ in advance), and (b) tag-collisions occur only with sufficiently small (but possibly non-negligible) probability in an attack with A (so only a constant number of $t^*_{i^*}$-signatures will have to be generated for A). This instance i^* is the challenge instance, and all other instances are simulated by A for B.[3] Any valid forgery of B *must* contain a valid signature under instance i^* with 2^{i^*}-bit tag. Hence any B-forgery implies an A-forgery.

The bottleneck of our transformation is the security reduction. Concretely, if B makes q signature queries and forges with success probability ε, A will make up to $Q = \mathbf{O}(q^4/\varepsilon)$ signature queries and have success $\varepsilon/2$. (One "squaring" is caused by a birthday bound when hoping for few tag-collisions; another squaring may occur when we have to round i^* up to the next integer.) This security loss is annoying, but can be significantly reduced by using techniques from [17]. Namely, by first applying a suitable "weakly programmable" hash function to tags, we can allow m-fold tag-collisions in the signatures prepared for B, at the cost of an extra m group elements in verification key and signatures. Orthogonally, we can use c^i-bit states with $1 < c < 2$ (instead of 2^i-bit states) in the i-th mildly secure instance to get a "finer-grained" growth of possible tag sets. This costs a multiplicative factor of $1/\log_2(c)$ in verification key and signature size, of course unless aggregation is possible. With these measures, the security reduction improves considerably: A will make $Q = \mathbf{O}(q^{c+c/m}/\varepsilon^{c/m})$ signature queries and have success $\varepsilon/2$. Varying c and m gives thus an interesting tradeoff between efficiency and quality of the security reduction.

Efficiency comparison. The most efficient previous CDH-based signature scheme [34] has signatures and public keys of size $\mathbf{O}(k)$, resp. $\mathbf{O}(1)$ group elements. Our CDH-based scheme also has constant-sized signatures, and more compact public keys. Concretely, we can get public keys of $\mathbf{O}(\log(k))$ group elements at the price of a worse security reduction. Our RSA-based

[2] Since we guess t^* from a small set of possible tags, we call our technique "confined guessing."

[3] This neglects a subtlety: A must specify the messages to be signed for the i^*-th instance in advance, while B expects to make adaptive signing queries. This difference can be handled using standard techniques (i.e., chameleon hashing [23]).

scheme has similar key and signature sizes as existing RSA-based schemes [20, 18], but requires significantly fewer (i.e., only $\mathbf{O}(\log(k))$ instead of $\mathbf{O}(k)$, resp. $\mathbf{O}(k/\log(k))$ many) generations of large primes. Again, this improvement is bought with a worse security reduction. Our SIS-based scheme offers an alternative to the existing scheme of [6]. Concretely, our scheme has larger (by a factor of $\log(k)$) signatures and a worse security reduction, but significantly smaller (by a factor of $k/\log(k)$) public keys.

On a related result by Seo. In an independent work, Seo [30] constructs essentially the same CDH-based scheme, however with a very different security analysis. His analysis is much tighter than ours, and for concrete security parameters, his scheme is more efficient. At the same time, Seo only proves a bounded form of security in which the number of adversarial signature queries has to be known at the time of key generation. The merged paper [3] presents both his and our own analysis.

2 Preliminaries

Notation. For $n \in \mathbb{R}$, let $[n] := \{1, \ldots, \lfloor n \rfloor\}$. Throughout the paper, $k \in \mathbb{N}$ denotes the security parameter. For a finite set $\mathcal{S}$, we denote by $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For a probabilistic algorithm A, we write $y \leftarrow A(x)$ for the process of running A on input x with uniformly chosen random coins, and assigning y the result. If A's running time is polynomial in k, then A is called *probabilistic polynomial-time* (PPT). A function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible* if it vanishes faster than the inverse of any polynomial (i.e., if $\forall c \exists k_0 \forall k \geq k_0 : f(k) \leq 1/k^c$). On the other hand, f is *significant* if it dominates the inverse of some polynomial (i.e., if $\exists c, k_0 \forall k \geq k_0 : f(k) \geq 1/k^c$).

Definition 2.1 (Signature scheme)**.** *A signature scheme* SIG *with message space* $\mathcal{M}_k$ *consists of three PPT algorithms:*

Setup. *The setup algorithm* $\mathsf{Gen}(1^k)$*, given the security parameter* 1^k *in unary, outputs a public key* pk *and a secret key* sk*.*

Sign. *The signing algorithm* $\mathsf{Sig}(sk, M)$*, given the secret key* sk *and a message* $M \in \mathcal{M}_k$*, outputs a signature* σ*.*

Verify. *Given the public key* pk*, a message* M*, and a signature* σ*,* $\mathsf{Ver}(pk, M, \sigma)$ *outputs a bit* $b \in \{0, 1\}$*. (The case* $b = 1$ *corresponds to a valid signature on the message, and the case* $b = 0$ *means invalid.)*

For correctness, we require for any $k \in \mathbb{N}$*, all* $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$*, all* $M \in \mathcal{M}_k$*, and all* $\sigma \leftarrow \mathsf{Sig}(sk, M)$ *that* $\mathsf{Ver}(pk, M, \sigma) = 1$*.*

Definition 2.2 (Existential unforgeability under (non-adaptive) chosen-message attacks)**.** *We say a signature scheme is existential unforgeable under chosen-message attacks (EUF-CMA) or existential unforgeable under non-adaptive chosen-message attacks (EUF-naCMA) iff*

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},F}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},F}(k) = 1\right] \text{ and } \mathsf{Adv}^{\mathsf{euf\text{-}nacma}}_{\mathsf{SIG},F}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}nacma}}_{\mathsf{SIG},F}(k) = 1\right],$$

respectively, are negligible for any PPT adversary F*, where* $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},F}(k)$ *and* $\mathsf{Exp}^{\mathsf{euf\text{-}nacma}}_{\mathsf{SIG_t},F}(k)$*, respectively, are defined in Figure 1.*

Pseudorandom functions. For any set $\mathcal{S}$ a *pseudorandom function (PRF) with range* $\mathcal{S}$ is an efficiently computable function $\mathsf{PRF}^{\mathcal{S}} : \{0,1\}^k \times \{0,1\}^* \to \mathcal{S}$. We may also write $\mathsf{PRF}^{\mathcal{S}}_{\kappa}(x)$ for $\mathsf{PRF}^{\mathcal{S}}(\kappa, x)$ with key $\kappa \in \{0,1\}^k$. Additionally we require that

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF}^{\mathcal{S}},A}(k) := \left|\Pr\left[A^{\mathsf{PRF}^{\mathcal{S}}_{\kappa}(\cdot)} = 1 \text{ for } \kappa \leftarrow \{0,1\}^k\right] - \Pr\left[A^{\mathcal{U}_{\mathcal{S}}(\cdot)} = 1\right]\right|$$

is negligible in k where $\mathcal{U}_{\mathcal{S}}$ is a truly uniform function to $\mathcal{S}$. Note that for any efficiently samplable set $\mathcal{S}$ with uniform sampling algorithm Samp we can generically construct a PRF

```
Experiment $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},F}(\mathrm{k})$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
  $(M^*, \sigma^*) \leftarrow F^{\mathsf{Sig}(sk,\cdot)}(pk)$
  if $\mathsf{Ver}(pk, M^*, \sigma^*) = 0$
    or $F$ has queried $\mathsf{Sig}(sk, M^*)$
  then return 0, else return 1
```

```
Experiment $\mathsf{Exp}^{\mathsf{euf\text{-}nacma}}_{\mathsf{SIG},F}(\mathrm{k})$
  $(M_1, \dots, M_q) \leftarrow F(k)$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
  $\sigma_i \leftarrow \mathsf{Sig}(sk, M_i)$, for all $i \in [q]$
  $(M^*, \sigma^*) \leftarrow F(pk, \sigma_1, \dots, \sigma_q)$
  if $\mathsf{Ver}(pk, M^*, \sigma^*) = 0$
    or $M^* \in \{M_1, \dots, M_q\}$
  then return 0, else return 1
```

Figure 1: EUF-CMA and EUF-naCMA experiment for signature schemes.

with range $\mathcal{S}$ from a PRF $\mathsf{PRF}^{\{0,1\}^k}$ by using the output of $\mathsf{PRF}^{\{0,1\}^k}_{\kappa}$ as random coins for Samp. Following this principle we can construct $(\mathsf{PRF}^{\mathcal{S}_i})_{i\in[n]}$ for a family of sets $(\mathcal{S}_i)_{i\in[n]}$ from a single PRF $\mathsf{PRF}^{\{0,1\}^k}$ with sufficiently long output (hence we need only one key κ).

Chameleon hashing. A *chameleon hash scheme* consists of two PPT algorithms $(\mathsf{CHGen}, \mathsf{CHTrapColl})$. $\mathsf{CHGen}(1^k)$ outputs a tuple (CH, τ) where CH is the description of a efficiently computable *chameleon hash function* $\mathsf{CH} : \mathcal{M} \times \mathcal{R} \to \mathcal{N}$ which maps a message M and randomness r to a hash value $\mathsf{CH}(M, r)$. We require collision-resistance in the sense that it is infeasible to find $(M, r) \neq (M', r')$ with $\mathsf{CH}(M, r) = \mathsf{CH}(M', r')$. However, the trapdoor τ allows to produce collisions in the following sense: given arbitrary M, r, M', $\mathsf{CHTrapColl}(\tau, M, r, M')$ finds r' with $\mathsf{CH}(M, r) = \mathsf{CH}(M', r')$. We require that the distribution of r' is uniform given only CH and M'.

3 Tag-based Signatures: From Mild to Full Security

We now describe our main result: a generic transformation from mildly secure tag-based signature schemes to fully secure schemes. Let us first define the notion of tag-based signature schemes.

Definition 3.1. *A tag-based signature scheme* $\mathsf{SIG_t} = (\mathsf{Gen_t}, \mathsf{Sig_t}, \mathsf{Ver_t})$ *with message space* $\mathcal{M}_k$ *and tag space* $\mathcal{T}_k$ *consists of three PPT algorithms. The key generation algorithm* $(pk, sk) \leftarrow \mathsf{Gen_t}(1^k)$ *takes as input a security parameter and outputs a key pair* (pk, sk)*. The signing algorithm* $\sigma \leftarrow \mathsf{Sig_t}(sk, M, t)$ *computes a signature* σ *on input a secret key sk, message M, and tag t. The verification algorithm* $\mathsf{Ver_t}(pk, M, \sigma, t) \in \{0, 1\}$ *takes as input a public key pk, message M, signature* σ*, and a tag t, and outputs a bit. For correctness, we require for any* $k \in \mathbb{N}$*, all* $(pk, sk) \leftarrow \mathsf{Gen_t}(1^k)$*, all* $M \in \mathcal{M}_k$*, all* $t \in \mathcal{T}_k$*, and all* $\sigma \leftarrow \mathsf{Sig_t}(sk, M, t)$ *that* $\mathsf{Ver_t}(pk, M, \sigma, t) = 1$*.*

We define a mild security notion for tag-based schemes, dubbed EUF-naCMA$^*_m$ security, which requires an adversary F to initially specify all messages M_i it wants signed, along with corresponding tags t_i. Only then, F gets to see a public key, and is subsequently expected to produce a forgery for an arbitrary fresh message M^*, but with respect to an already used tag $t^* \in \{t_i\}_i$. As a slightly technical (but crucial) requirement, we only allow F to initially specify at most m messages M_i with tag $t_i = t^*$. We call m the *tag-collision parameter*; it influences key and signature sizes, and the security reduction.

Definition 3.2 (EUF-naCMA$^*_m$)**.** *Let* $m \in \mathbb{N}$*. A tag-based signature scheme* $\mathsf{SIG_t}$ *is existentially unforgeable under non-adaptive chosen-message attacks with m-fold tag-collisions (short: EUF-naCMA$^*_m$ secure) iff the function* $\mathsf{Adv}^{\mathsf{euf\text{-}nacma}^*_m}_{\mathsf{SIG_t},F}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}nacma}^*_m}_{\mathsf{SIG_t},F}(k) = 1\right]$ *is negligible for any PPT adversary F. Here, experiment* $\mathsf{Exp}^{\mathsf{euf\text{-}nacma}^*_m}_{\mathsf{SIG_t},F}(k)$ *is defined in Figure 2.*

```
$\mathsf{Gen}(1^k)$
  $(pk', sk) \leftarrow \mathsf{Gen_t}(1^k)$
  $\kappa \leftarrow \{0,1\}^k$
  $pk := (pk', \kappa)$
  return $(pk, sk)$
```

```
$\mathsf{Sig}(sk, M)$
  $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M)$ for $i \in [l]$
  $\sigma_i \leftarrow \mathsf{Sig_t}(sk, M, t_i)$
  return $\sigma := (\sigma_i)_{i=1}^l$
```

```
$\mathsf{Ver}((pk', \kappa), \sigma = (\sigma_i)_{i=1}^l, M)$
  $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M)$ for $i \in [l]$
  return $\bigwedge_{i=1}^{l} \mathsf{Ver_t}(pk', M, \sigma_i, t_i)$
```

Figure 3: Our EUF-naCMA secure signature scheme.

In this section, we will show how to use a EUF-naCMA$^*_m$ secure scheme $\mathsf{SIG_t}$ to build an EUF-naCMA secure scheme SIG. (Full EUF-CMA security can then be achieved using a chameleon hash function [23].)

To this end, we separate the tag space $\mathcal{T}_k$ into $l := \lfloor \log_c(k) \rfloor$ *pairwise disjoint* sets $\mathcal{T}_i'$, such that $|\mathcal{T}_i'| = 2^{\lceil c^i \rceil}$. Here $c > 1$ is a *granularity parameter* that will affect key and signature sizes, and the security reduction. For instance, if $c = 2$ and $\mathcal{T}_k = \{0,1\}^k$, then we may set $\mathcal{T}_i' := \{0,1\}^{2^i}$. The constructed signature scheme SIG assigns to each message M a vector of tags $(t_1, \ldots, t_l)$, where each tag is derived from the message M by applying a pseudorandom function as $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i'}(M)$. The PRF seed κ is part of SIG's public key.[4]

```
Experiment $\mathsf{Exp}_{\mathsf{SIG_t},F}^{\mathsf{euf\text{-}nacma}^*_\mathsf{m}}(k)$
  $(M_j, t_j)_{j=1}^{q(k)} \leftarrow F(1^k)$
  $(pk, sk) \leftarrow \mathsf{Gen_t}(1^k)$
  $\sigma_j \leftarrow \mathsf{Sig_t}(sk, M_j, t_j)$ for $j \in [q(k)]$
  $(M^*, \sigma^*, t^*) \leftarrow F(pk, (\sigma_j)_{j=1}^{q(k)})$
  if $\mathsf{Ver_t}(pk, M^*, \sigma^*, t^*) = 0$
    or $M \in \{M_j\}_{j=1}^{q(k)}$
    or $|\{j \in [q(k)] : t_j = t^*\}| > m$
    or $t^* \notin \{t_j\}_{j=1}^{q(k)}$
  then return 0, else return 1
```

Figure 2: EUF-naCMA$^*_m$ experiment for tag-based signature schemes.

A SIG-signature is of the form $\sigma = (\sigma_i)_{i=1}^l$, where each $\sigma_i \leftarrow \mathsf{Sig_t}(sk, M, t_i)$ is a signature according to $\mathsf{SIG_t}$ with message M and tag t_i. This signature is considered valid if all σ_i are valid w.r.t. $\mathsf{SIG_t}$.

The crucial idea is to define the sets $\mathcal{T}_i'$ of allowed tags as sets quickly growing in i. This means that $(m+1)$-tag-collisions (i.e., the same tag t_i being chosen for $m+1$ different signed messages) are very likely for small i, but become quickly less likely for larger i.

Concretely, let $\mathsf{SIG_t} = (\mathsf{Gen_t}, \mathsf{Sig_t}, \mathsf{Ver_t})$ be a tag-based signature scheme with tag space $\mathcal{T}_k = \bigcup_{i=1}^l \mathcal{T}_i'$, let $m \in \mathbb{N}$ and $c > 1$, and let PRF be a PRF. SIG is described in Figure 3.

It is straightforward to verify SIG's correctness. Before turning to the formal proof, we first give an intuition why SIG is EUF-naCMA secure. We will map an adversary F on SIG's EUF-naCMA security to an adversary F' on $\mathsf{SIG_t}$'s EUF-naCMA$^*_m$ security. Intuitively, F' will internally simulate the EUF-naCMA security experiment for F and embed its own $\mathsf{SIG_t}$-instance (with public key pk') in the SIG-instance of F by setting $pk := (pk', \kappa)$. Additionally, the seed κ for PRF is chosen internally by F'.

Say that F makes $q = q(k)$ (non-adaptive) signing requests for messages $M_1, \ldots, M_q$. To answer these q requests, F' can obtain signatures under pk' from its own EUF-naCMA$^*_m$ experiment. The corresponding tags are chosen as in SIG, as $t_i^{(j)} = \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M_j)$. Once F produces a forgery $\sigma^* = (\sigma_i^*)_{i=1}^l$, F' will try to use $\sigma_{i^*}^*$ (with tag $t_{i^*}^* = \mathsf{PRF}_\kappa^{\mathcal{T}_{i^*}}(M^*)$ for some appropiate $i^* \in [l]$) as its own forgery.

Indeed, $\sigma_{i^*}^*$ will be a valid $\mathsf{SIG_t}$-forgery (in the EUF-naCMA$^*_m$ experiment) if (a) F' did not initially request signatures for more than m messages for the forgery tag $t_{i^*}^*$, and (b) $t_{i^*}^*$ already appears in one of F''s initial signature requests. Our technical handle to make this event likely

[4] It will become clear in the security proof that actually a function with weaker security properties than a fully-secure PRF is sufficient for our application. However, we stick to standard PRF security for simplicity. Thanks to an anonymous reviewer for pointing this out.

will be a suitable choice of i^*. First, recall that the i-th signature σ_i uses $\lceil c^i \rceil$-bit tags. We will hence choose i^* such that

(i) the probability of an $(m+1)$-tag-collision among the $t_{i^*}^{(j)}$ is significantly lower than F's success probability (so F will sometimes have to forge signatures when no $(m+1)$-tag collision occurs), and

(ii) $|\mathcal{T}'_{i^*}| = 2^{\lceil c^{i^*} \rceil}$ is polynomially small (so all tags in $\mathcal{T}'_{i^*}$ can be initially queried by F').

We turn to a formal proof:

Theorem 3.3. *If* PRF *is a PRF and* $\mathsf{SIG_t}$ *is an EUF-naCMA*$^*_m$ *secure tag-based signature scheme, then* SIG *is EUF-naCMA secure. Concretely, let F be an EUF-naCMA forger on* SIG *with non-negligible advantage* $\varepsilon := \mathsf{Adv}^{\mathsf{euf\text{-}nacma}}_{\mathsf{SIG},F}(k)$ *and making $q = q(k)$ signature queries. Then* $\varepsilon(k) > \frac{1}{p(k)}$ *for some polynomial p and $k \in K$ for an infinite set $K \subseteq \mathbb{N}$. For $k \in K$ there exists a EUF-naCMA*$^*_m$ *forger F' on* $\mathsf{SIG_t}$ *with advantage* $\varepsilon' := \mathsf{Adv}^{\mathsf{euf\text{-}nacma}^*_m}_{\mathsf{SIG_t},F'}(k)$ *and making* $q'(k) \leq 2 \cdot \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)}\right)^{c/m}$ *signature queries, such that*

$$\varepsilon' \geq \varepsilon/2 - \varepsilon_{\mathsf{PRF}} - \frac{1}{|\mathcal{M}_k|},$$

where $\varepsilon_{\mathsf{PRF}}$ *is the advantage of a suitable PRF distinguisher on* PRF *and* $\mathcal{M}_k$ *the message space.*

Proof. **Setup and sign.** First, F' receives messages $M_1, \ldots, M_q$ from F. F' chooses the challenge instance i^* such that the probability of an $(m+1)$-tag collision is at most $\varepsilon(k)/2$, i.e.,

$$\Pr\left[\exists \{j_0, \ldots, j_m\} \subseteq [q] : t_{i^*}^{(j_0)} = \cdots = t_{i^*}^{(j_m)}; t_{i^*}^{(j)} \leftarrow \mathcal{T}_{i^*}, 1 \leq j \leq q\right] \leq \frac{\varepsilon(k)}{2}, \qquad (1)$$

and such that $|\mathcal{T}'_{i^*}|$ is polynomial in k. Concretely,

$$i^* := \left\lceil \log_c \left(\log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m} \right) \right\rceil$$

is an index that fulfills these conditions. (See Lemma 3.5 for a complete analysis.) F' then chooses a PRF key $\kappa \leftarrow \{0,1\}^k$.

Recall that a signature $\sigma = (\sigma_1, \ldots, \sigma_l)$ of SIG consists of l signatures of $\mathsf{SIG_t}$. In the sequel we write $\sigma^{(j)} = (\sigma_1^{(j)}, \ldots, \sigma_l^{(j)})$ to denote the SIG-signature for message M_j, for all $j \in \{1, \ldots, q\}$. Adversary F' uses its signing oracle provided from the $\mathsf{SIG_t}$-security experiment to simulate these SIG-signatures. To this end, it proceeds as follows.

In order to simulate all signatures $\sigma_i^{(j)}$ with $i \neq i^*$, F' computes $t_i^{(j)} := \mathsf{PRF}_\kappa^{\mathcal{T}'_i}(M_j)$ and defines message-tag pair $(M_j, t_i^{(j)})$. F' will later request signatures for these message-tag pairs from its EUF-naCMA$^*_m$-challenger. Note that $t_i^{(j)} \notin \mathcal{T}'_{i^*}$ for all $i \neq i^*$, since the sets $\mathcal{T}'_1, \ldots, \mathcal{T}'_l$ are pairwise disjoint.

To compute the i^*-th $\mathsf{SIG_t}$-signature $\sigma_{i^*}^{(j)}$ contained in $\sigma^{(j)}$, F' proceeds as follows. First it computes $t_{i^*}^{(j)} := \mathsf{PRF}_\kappa^{\mathcal{T}'_{i^*}}(M_j)$ for all $j \in \{1, \ldots, q\}$. If a $(m+1)$-fold tag-collision occurs, then F' aborts. This defines q message-tag-pairs (M_j, t_j) for $j \in \{1, \ldots, q\}$. Note that the list $(t_{i^*}^{(1)}, \ldots, t_{i^*}^{(q)})$ need not contain all elements of $\mathcal{T}'_{i^*}$, that is, it might hold that $\mathcal{T}'_{i^*} \setminus \{t_{i^*}^{(1)}, \ldots, t_{i^*}^{(q)}\} \neq \emptyset$. If this happens, then F' chooses a dummy message $M \leftarrow \mathcal{M}_k$ uniformly at random and associates it with all tags $t \in \mathcal{T}'_{i^*} \setminus \{t_{i^*}^{(1)}, \ldots, t_{i^*}^{(q)}\}$ that are not contained in $\{t_{i^*}^{(1)}, \ldots, t_{i^*}^{(q)}\}$. This defines further message-tag-pairs (M, t) for each $t \in \mathcal{T}'_{i^*} \setminus \{t_{i^*}^{(1)}, \ldots, t_{i^*}^{(q)}\}$. We do this since F' has to re-use an already queried tag for a valid forgery later and F' does not know at this point which tag F is going to use in his forgery later.

Finally F' requests signatures for all message-tag-pairs from its challenger, and receives in return signatures $\sigma_{i^*}^{(j)}$ for all j, as well as a public key pk'.

F' defines $pk := (pk', \kappa)$ and hands $(pk, \sigma^{(1)}, \dots, \sigma^{(q)})$ to F. Note that each $\sigma^{(j)}$ is a valid SIG-signature for message M_j.

Extraction. F eventually forges a signature $\sigma^* = (\sigma_i^*)_{i=1}^l$ for a fresh message $M^* \notin \{M_1, \dots, M_q\}$. If $M^* = M$, then F' aborts. Otherwise it forwards $(M^*, \sigma_{i^*}^*, \mathsf{PRF}_\kappa^{\mathcal{T}'_{i^*}}(M^*))$ to the challenger of the EUF-CMA$_m^*$ security experiment.

This concludes the description of F'.

Analysis. We now turn to F''s analysis. Let $\mathsf{bad}_{\mathsf{abort}}$ be the event that F' aborts. It is clear that F' successfully forges a signature whenever F does so, and $\mathsf{bad}_{\mathsf{abort}}$ does not occur. Note that message M is independent of the view of F, thus we have $\Pr[M = M^*] \leq 1/|\mathcal{M}_k|$. Hence, to prove our theorem, it suffices to show that $\Pr[\mathsf{bad}_{\mathsf{abort}}] \leq \varepsilon/2 + \varepsilon_{\mathsf{PRF}} + 1/|\mathcal{M}_k|$ since this leaves a non-negligible advantage for F'.

First note that the probability of an $(m+1)$-tag collision would be at most $\varepsilon/2$ by (1) if the tags $t_{i^*}^{(j)}$ were chosen truly uniformly from $\mathcal{T}'_{i^*}$. Now recall that the actual choice of the $t_{i^*}^{(j)} = \mathsf{PRF}_\kappa^{\mathcal{T}'_{i^*}}(M_j)$ was performed in a way that uses PRF only in a black-box way. Hence, if $(m+1)$-tag collisions (and thus $\mathsf{bad}_{\mathsf{abort}}$) occurred significantly more often than with truly uniform tags, we had a contradiction to PRF's pseudorandomness. Concretely, a PRF distinguisher that simulates F' until the decision to abort is made shows $\Pr[\mathsf{bad}_{\mathsf{abort}}] \leq \varepsilon/2 + \varepsilon_{\mathsf{PRF}} + 1/|\mathcal{M}_k|$, and thus the theorem. □

In order to obtain a fully EUF-CMA secure signature scheme, one may combine our EUF-naCMA-secure scheme with a suitable chameleon hash function or a one-time signature scheme. This is a very efficient standard construction, see for instance [20, Lemma 2.3] for details.

We now turn to the analysis of selecting the challenge index. For this, the following Lemma 3.4 is helpful.

Lemma 3.4 ([18], Lemma 2.3)**.** *Let A be a set with $|A| = a$. Let $X_1, \dots, X_q$ be q independent random variables, taking uniformly random values from A. Then the probability that there exist $m+1$ pairwise distinct indices $i_1, \dots, i_{m+1}$ such that $X_{i_1} = \cdots = X_{i_{m+1}}$ is upper bounded by $\frac{q^{m+1}}{a^m}$.*

Lemma 3.5. *Given the situation in Theorem 3.3, then*

$$i^* := \left\lceil \log_c \left(\log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m} \right) \right\rceil$$

is an index that meets the conditions required in the proof of Theorem 3.3 (small probability for tag collisions and polynimial tag space size) for all $k \in K$.

Proof. For the granularity parameter $c > 1$, the number of instances $l = \lfloor \log_c(k) \rfloor$, the tag sets $(\mathcal{T}_1, \dots, \mathcal{T}_l)$, the number of signing queries q, the forger advantage $\varepsilon(k)$, and for an index i^* it should hold in Theorem 3.3 that

(a) the probability of an $(m+1)$-tag-collision is smaller than or equal to $\varepsilon(k)/2$, i.e.,

$$\Pr\left[\exists \{j_0, \dots, j_m\} \subseteq [q] : t_{i^*}^{(j_0)} = \cdots = t_{i^*}^{(j_m)}; t_{i^*}^{(j)} \leftarrow \mathcal{T}_{i^*}, 1 \leq j \leq q\right] \leq \frac{\varepsilon(k)}{2}, \text{ and}$$

(b) $|\mathcal{T}_{i^*}|$ is polynomial in k.

We start with (b). To show that, for such an i^* above, $|\mathcal{T}_{i^*}|$ is polynomial in k, we derive by definition of the tag sets in Section 3,

$$|\mathcal{T}_{i^*}| = 2^{\lceil c^{i^*} \rceil} \leq 2^{c^{i^*}+1}$$

$$= 2^{c^{\left\lceil \log_c \left(\log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m} \right) \right\rceil}+1} \leq 2^{c^{\log_c \left(\log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m} \right)+1}+1}$$

$$= 2^{\log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m} \cdot c+1} = \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{c/m} \cdot 2.$$

Now, since $\varepsilon(k)$ is significant in k, i.e., $\varepsilon(k) \geq 1/k^{c'}$ for a constant c', for $\mathcal{T}_{i^*}$ with index i^* as above, we have

$$|\mathcal{T}_{i^*}| \leq 2 \cdot \left(k^{c'} \cdot 2 \cdot q^{m+1} \right)^{c/m}$$

where $|\mathcal{T}_{i^*}|$ is the upper bounded by a polynomial in k. This fulfills (b).

We now turn to (a). We start with such i^* as above and derive

$$i^* = \left\lceil \log_c \left(\log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m} \right) \right\rceil \geq \log_c \left(\log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m} \right)$$

which, since $c > 1$, is equivalent to

$$c^{i^*} \geq \log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m}.$$

Now, we may derive

$$\lceil c^{i^*} \rceil \geq c^{i^*} \geq \log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m}$$

$$\iff \quad 2^{\lceil c^{i^*} \rceil} \geq \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m}$$

$$\iff \quad \frac{\varepsilon(k)}{2} \geq \frac{q^{m+1}}{2^{\lceil c^{i^*} \rceil \cdot m}}$$

By the definition of the tag sets in Section 3, i.e., $|\mathcal{T}_{i^*}| = 2^{\lceil c^{i^*} \rceil}$, we have

$$\frac{\varepsilon(k)}{2} \geq \frac{q^{m+1}}{|\mathcal{T}_{i^*}|^m}. \qquad (2)$$

Now, applying the generalized birthday bound from Lemma 3.4 it holds that

$$\Pr \left[\exists \{j_0, \ldots, j_m\} \subseteq [q] : t_{i^*}^{(j_0)} = \cdots = t_{i^*}^{(j_m)}; t_{i^*}^{(j)} \leftarrow \mathcal{T}_{i^*}, 1 \leq j \leq q \right] \leq \frac{q^{m+1}}{|\mathcal{T}_{i^*}|^m}. \qquad (3)$$

Equation 2 and Equation 3 shows that the index i^* fulfills (a). Hence, index i^* meets our requirements which concludes our proof. □

```
$\mathsf{Gen}_\mathsf{t}(1^k)$
    Choose $\mathbb{G}$ s.t. $p := |\mathbb{G}| > 2^k$
    $a \leftarrow \mathbb{Z}_p$
    $g, u_0, \ldots, u_m, z, h \leftarrow \mathbb{G}$
    $sk := a$
    $pk := (g, g^a, u_0, \ldots, u_m, z, h)$
    return $(pk, sk)$
```

```
$\mathsf{Sig}_\mathsf{t}(sk, M, t)$
    $s \leftarrow \mathbb{Z}_p$
    $\mathbf{u}^M := \prod_{i=0}^{m} u_i^{M^i}$
    $\tilde{\sigma}_1 := (\mathbf{u}^M)^a (z^t h)^s$
    $\tilde{\sigma}_2 := g^s$
    return $(\tilde{\sigma}_1, \tilde{\sigma}_2)$
```

```
$\mathsf{Ver}_\mathsf{t}(pk, M, \sigma = (\tilde{\sigma}_1, \tilde{\sigma}_2), t)$
    if $t \notin \mathcal{T}_k$
        return 0
    if $e(\tilde{\sigma}_1, g) \neq e(\mathbf{u}^M, g^a) e(\tilde{\sigma}_2, z^t h)$
        return 0
    else
        return 1
```

Figure 4: The modified Hohenberger-Waters CDH-based signature scheme $\mathsf{SIG}^{\mathsf{CDH}}$ [21].

4 Our CDH-based scheme

In this section we construct a fully EUF-CMA-secure signature scheme based on the CDH assumption. We start with constructing a tag-based scheme, which is derived from the stateful CDH-based scheme of Hohenberger and Waters [21], and prove it EUF-naCMA$^*_m$-secure. Then we can apply our generic transformation from Section 3 to achieve full EUF-CMA security. Finally, we illustrate some optimizations that allow us to reduce the size of public keys and signatures, for instance by aggregation. Our scheme is the first CDH-based signature scheme with such compact public keys.

Definition 4.1 (CDH assumption). *We say that the* Computational Diffie-Hellman (CDH) assumption *holds in a group $\mathbb{G}$ of order p iff $\mathsf{Adv}^{\mathsf{cdh}}_{\mathbb{G},F}(k) := \Pr\left[F(1^k, g, g^a, g^b) = g^{ab}\right]$ is negligible for any PPT adversary F, where $g \leftarrow \mathbb{G}$ and $a, b \leftarrow \mathbb{Z}_p$ are uniformly chosen.*

CDH Construction. The signature scheme $\mathsf{SIG}^{\mathsf{CDH}}$ described in Figure 4 is derived from the CDH-based scheme of [21], but with two modifications. First, we substitute the implicit chameleon hash function $u^m v^r$ used in [21] with a product $\mathbf{u}^M = \prod_{i=0}^m u_i^{M^i}$. Second, we omit the $w^{\lceil \log(t) \rceil}$-factor in the "Boneh-Boyen hash function", which simplifies this part to $(z^t h)^s$. From now on we assume that $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map, i.e., $e(g, g) \neq 1$ for $g \neq 1$ and $e(g^a, g^b) = e(g, g)^{ab}$ for $a, b \in \mathbb{Z}$.

Theorem 4.2. *If the CDH assumption holds in $\mathbb{G}$, then $\mathsf{SIG}^{\mathsf{CDH}}$ from Figure 4 is EUF-naCMA$^*_m$-secure. Let F be a PPT adversary with advantage $\varepsilon := \varepsilon(k) := \mathsf{Adv}^{\mathsf{euf\text{-}nacma}^*_\mathsf{m}}_{\mathsf{SIG}^{\mathsf{CDH}},F}(k)$ asking for $q := q(k)$ signatures, then it can be used to solve a CDH challenge with probability $(\varepsilon \cdot m)/q$.*

Proof. **Public key setup.** The simulation receives a CDH-Challenge (g, g^a, g^b) and pairs $(M_i, t_i)_{i \in [q]}$ for which the adversary F asks for signatures. We first guess an index $i^* \leftarrow [q]$ for which we suppose F will forge a signature on a new message $M^* \neq M_{i^*}$, but with $t^* = t_{i^*}$. W.l.o.g., we assume that the adversary F queries $q = q' \cdot m > 0$ signatures for messages with tags where q' is the number of distinct tags $(t_i)_{i \in [q']}$ and m the number of messages queried for each tag. During the simulation we define by $M_j^*, j = 1, \ldots, m$, the corresponding messages to t_{i^*}. Using these we construct a polynomial $f(X) := \prod_{i=1}^m (X - M_i^*) = \sum_{i=0}^m d_i X^i \in \mathbb{Z}_p[X]$ for some appropriate coefficients $d_1, \ldots, d_m \in \mathbb{Z}_p$.

Next, we set up the public key for F by first choosing random $r_0, \ldots, r_m, x_z, x_h \in \mathbb{Z}_p$ and then set $u_i := (g^b)^{d_i} g^{r_i}, i = 0, \ldots, m$, $z := g^b g^{x_z}$, and $h := g^{-bt_{i^*}} g^{x_h}$. The simulation outputs $pk := (g, g^a, u_0, \ldots, u_m, z, h)$ and implicitly sets the secret key as $sk := a$. We set $r(X) := \sum_{i=0}^m r_i X^i$ so we can write $\mathbf{u}^M = g^{bf(M)+r(M)}$.

Signing. Now there are two cases we have to consider when F asks for a signature on (M_i, t_i). If $t_i = t_{i^*}$ and thus $M_i = M_j^*$ for some $j = 1, \ldots, m$, then we can compute a valid signature as follows: We choose a random $s_i \leftarrow \mathbb{Z}_p$ and set $\sigma_i := (\tilde{\sigma}_{1,i}, \tilde{\sigma}_{2,i})$ where

$$\tilde{\sigma}_{1,i} = (g^a)^{r(M_j^*)} \cdot (z^{t_{i^*}} h)^{s_i} \ , \ \tilde{\sigma}_{2,i} = g^{s_i}.$$

We verify this by using that $f(M_j^*) = 0$, i.e.,

$$\tilde{\sigma}_{1,i} = (g^a)^{r(M_j^*)} \cdot (z^{t_{i^*}} h)^{s_i} = (g^{bf(M_j^*)} g^{r(M_j^*)})^a \cdot (z^{t_{i^*}} h)^{s_i} = (\mathbf{u}^{M_j^*})^a \cdot (z^{t_{i^*}} h)^{s_i}.$$

If $t_i \neq t_{i^*}$, then we consider the Boneh-Boyen simulation as in the original scheme, choose a random $s_i' \leftarrow \mathbb{Z}_p$ and set $S_i := g^{s_i'}/(g^a)^{f(M_i)/(t_i - t_{i^*})} = g^{s_i' - af(M_i)/(t_i - t_{i^*})}$. We compute our valid signature $\sigma_i := (\tilde{\sigma}_{1,i}, \tilde{\sigma}_{2,i})$ as follows:

$$\tilde{\sigma}_{1,i} = (g^a)^{r(M_i)} \cdot S_i^{x_z t_i + x_h} \cdot (g^b)^{s_i'(t_i - t_{i^*})} \ , \ \tilde{\sigma}_{2,i} = S_i.$$

Thus, implicitly, we set the randomness $s_i = s_i' - af(M_i)/(t_i - t_{i^*})$ and obtain $S_i = g^{s_i}$. We verify by showing that

$$\begin{aligned}
\tilde{\sigma}_{1,i} &= (g^a)^{r(M_i)} \cdot S_i^{x_z t_i + x_h} \cdot (g^b)^{s_i'(t_i - t_{i^*})} \\
&= (g^{r(M_i)})^a \cdot (g^{x_z t_i} g^{x_h})^{s_i} \cdot (g^b)^{s_i'(t_i - t_{i^*})} \\
&= (g^{ab})^{f(M_i)} \cdot (g^{r(M_i)})^a \cdot (g^{x_z t_i} g^{x_h})^{s_i} \cdot (g^b)^{s_i'(t_i - t_{i^*})} \cdot (g^{-ab})^{f(M_i)} \\
&= ((g^{bf(M_i) + r(M_i)})^a \cdot (g^{x_z t_i} g^{x_h})^{s_i} \cdot (g^{b(t_i - t_{i^*})})^{s_i} \\
&= (\mathbf{u}^{M_i})^a \cdot ((g^{b + x_z})^{t_i} \cdot (g^{-bt_{i^*} + x_h}))^{s_i} \\
&= (\mathbf{u}^{M_i})^a \cdot (z^{t_i} h)^{s_i}.
\end{aligned}$$

Extract from forgery. The adversary F responds with (M^*, σ^*, t^*) for some tag $t^* \in \{t_1, \ldots, t_q\}$ and $\sigma^* = (\tilde{\sigma}_1^*, \tilde{\sigma}_2^*)$. We abort if σ^* is not a valid forgery. Otherwise, since the verification equation holds we have

$$\tilde{\sigma}_1^* = (\mathbf{u}^{M^*})^a (z^{t^*} h)^{s^*} \ , \ \tilde{\sigma}_2^* = g^{s^*}$$

If $t_{i^*} \neq t^*$ abort, otherwise our guess was correct and it holds that

$$\begin{aligned}
\tilde{\sigma}_1^* &= ((g^b)^{f(M^*)} (g^{r(M^*)}))^a ((g^{b + x_z})^{t^*} (g^{-bt_{i^*} + x_h}))^{s^*} \\
&= g^{abf(M^*)} g^{ar(M^*)} (g^{x_z t^*} g^{x_h})^{s^*} \\
&= g^{abf(M^*)} g^{ar(M^*)} g^{s^*(x_z t^* + x_h)}.
\end{aligned}$$

Since $M^* \neq M_j^*$ we have $f(M^*) \neq 0$ and the simulator can compute

$$(\tilde{\sigma}_1^* / (g^{ar(M^*)} \tilde{\sigma}_2^{*(x_z t^* + x_h)}))^{1/f(M^*)} = g^{ab}$$

Analysis. We show that the adversary F cannot distinguish effectively between the experiment and the simulation. We denote by ε the advantage of the adversary F in the experiment and by $\mathsf{success}$ the event, that the simulation outputs a solution g^{ab}. The simulator does not pick $(u_i)_{i=0}^m$, z, and h at random but sets them as described above. Since the r_i, x_z and x_h are randomly chosen this yields the correct distribution, so the view of the adversary is still the same as in the experiment. The simulator is successful if it does not abort, that means if F is successful and it guesses t^* correctly. So we have $\Pr[\mathsf{success}] = \frac{\varepsilon \cdot m}{q}$. □

```
$\mathsf{Gen}(1^k)$
  Choose $\mathbb{G}$ s.t. $p := |\mathbb{G}| > 2^k$
  $a \leftarrow \mathbb{Z}_p$
  $g, w, h, u_0, \ldots, u_m,$
  $z_1, \ldots, z_l \leftarrow \mathbb{G}$
  $\kappa \leftarrow \{0,1\}^k$
  $sk := (g, w, a)$
  $pk := (g, w, g^a, (u_j)_{j=0}^m,$
          $(z_i)_{i=1}^l, h, \kappa)$
  return $(pk, sk)$
```

```
$\mathsf{Sig}(sk, M)$
  $s, r \leftarrow \mathbb{Z}_p$
  $x := \mathsf{CH}_{(g,w)}(M, r)$
  $\mathbf{u}^x := \prod_{i=0}^m u_i^{x_i}$
  for $i := 1$ to $l$ do
    $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x)$
  $z := \prod_{i=1}^l z_i^{t_i}$
  $\tilde{\sigma}_1 := (\mathbf{u}^x)^a (z \cdot h)^s$
  $\tilde{\sigma}_2 := g^s$
  return $(\tilde{\sigma}_1, \tilde{\sigma}_2, r)$
```

```
$\mathsf{Ver}(pk, M, \sigma = (\tilde{\sigma}_1, \tilde{\sigma}_2, r))$
  $x := \mathsf{CH}_{(g,w)}(M, r)$
  for $i := 1$ to $l$ do
    $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x)$
  if $e(\tilde{\sigma}_1, g) \neq e(\mathbf{u}^x, g^a) e(\tilde{\sigma}_2, h \prod_{i=1}^{l} z_i^{t_i})$
    return 0
  else
    return 1
```

Figure 5: The optimized CDH-based signature scheme $\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{opt}}$.

4.1 Optimizations

Now, with this result and our generic transformation from Section 3 we can construct a stateless signature scheme, which is proven EUF-naCMA secure by Theorem 3.3. Then we add an explicit chameleon hash function $\mathsf{CH}_{(g,w)}(M, r) := g^M w^r$ in each instance $i = 1, \ldots, \lfloor \log_c(k) \rfloor$, to achieve a fully EUF-CMA-secure signature scheme. This signature scheme does have a constant size public key but signatures consist of $\mathbf{O}(\log k)$ group elements, i.e. $\sigma = (\sigma_1, \ldots, \sigma_{\log k})$ where $\sigma_i = (\tilde{\sigma}_{1,i}, \tilde{\sigma}_{2,i}, r_i)$.

Now, we concentrate on how we can improve this and achieve constant size signatures. This will be done by aggregation, essentially by multiplying the signatures of each instance similar to [25]. We re-use $u_0, \ldots, u_m$, one $sk := a$ and one randomness s for all instances i (see Figure 5). Unfortunately, we need additional elements in the public key for the aggregation to work. In this sense our optimization is rather a tradeoff: We prefer constant-size signatures with public keys of logarithmic length over logarithmic-length signatures with constant-size public keys.

Theorem 4.3. *If the CDH assumption holds in $\mathbb{G}$ then the optimized CDH-based signature scheme in Figure 5 is a EUF-CMA secure signature scheme. Let F be a PPT adversary with advantage $\varepsilon := \varepsilon(k) := \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{opt}},F}(k)$ asking for $q := q(k)$ signatures, then it can be used to solve a CDH challenge with probability at least $\frac{\varepsilon^{c/m+1}}{2^{2+c/m}.q^{c+c/m}} - \varepsilon_{\mathsf{PRF}} - \varepsilon_{\mathsf{CH}}$, where $\varepsilon_{\mathsf{PRF}}$ and $\varepsilon_{\mathsf{CH}}$ correspond to the advantages for breaking PRF and CH respectively.*

Proof. We only sketch the proof here because it is almost a combination of the proofs from Theorem 4.2 and Theorem 3.3. We emphasize the differences and important parts. We have to deal with $l = \lfloor \log_c(k) \rfloor$ instances and obtain our signature by generic aggregation.
Public key setup. First we select an index i^* and guess a tag t_{i^*} from the corresponding set $\mathcal{T}_{i^*}$. Next we pick a random $\kappa \leftarrow \{0,1\}^k$ and random $M, r_1, ..., r_q \leftarrow \mathbb{Z}_p$ and compute $x_j = \mathsf{CH}(M, r_j), j \in [q]$. We derive from that a tag $t_i^{(j)} := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x_j)$ for each instance $i \in [l]$. Then we consider the set $J = \{\, j \in [q] \mid t_{i^*}^{(j)} = t_{i^*}\}$. W.l.o.g. we assume that $|J| = m$. With that we can, similar to Theorem 4.2, compute a polynomial f s.t. $f(x_j) = 0$ for $j \in J$. We set up $u_0, .., u_m, h, z_{i^*}$ as before in Theorem 4.2 to embed our challenge here and choose random z_i for $i \in [q] \neq i^*$ by choosing random dlogs x_{z_i} for them.
Signing. The adversary will send us messages $M_1, ..., M_q$ and we have to compute a valid signature $\sigma^{(j)} = (\tilde{\sigma}_{1,j}\, \tilde{\sigma}_{2,j}, r'_j)$ for each of them. We compute r'_j s.t. $x_j = \mathsf{CH}(M_j, r'_j)$. Thus $t_i^{(j)}$ belongs to M_j, for $i = 1, ..., l$. Now we consider the instance $i = i^*$:

Again, we have two cases, either $t_{i^*}^{(j)} = t_{i^*}$ or $t_{i^*}^{(j)} \neq t_{i^*}$. In both cases we can apply the same techniques as in Theorem 4.2 to obtain a valid $(\mathbf{u}^{x_j})^a (z_{i^*}^{t_{i^*}^{(j)}}\, h)^{s_j}$.

For the other instances $i \neq i^*$ we can compute $(z_i^{t_i^{(j)}})^{s_j} = (g^{x_{z_i} t_i^{(j)}})^{s_j} = (g^{s_j})^{x_{z_i} t_i^{(j)}}$ due to the fact that we know the dlogs x_{z_i} and then can aggregate them to get

$$\sigma_j := ((\mathbf{u}^{x_j})^a (h \prod_{i=1}^{l} z_i^{t_i^{(j)}})^{s_j}, g^{s_j}, r'_j)$$

Extract from forgery. The adversary F responds with (M^*, σ^*) where $\sigma^* = (\tilde{\sigma}_1^*, \tilde{\sigma}_2^*, r^*)$ and $M^* \neq M_j$ for $j \in [q]$. Abort if σ^* is not valid. We can assume that F has not produced a collision $x^* = \mathsf{CH}(M^*, r^*) = x_j$ for some $j \in [q]$ due to the collision resistance of our chameleon hash function. Thus we have $f(x^*) \neq 0$. Now, we compute $t_i^* = \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x^*)$ for each instance $i \in [l]$. If $t_{i^*}^* \neq t_{i^*}$ abort, otherwise it holds

$$\tilde{\sigma}_1^* = (\mathbf{u}^{x^*})^a (h \prod_{i=1}^{l} z_i^{t_i^*})^{s^*}, \ \tilde{\sigma}_2^* = g^{s^*}.$$

We can compute $(g^{s^*})^{x_{z_i} t_i^*} = (z_i^{t_i^*})^{s^*}$ for $i \neq i^*$ and obtain

$$\sigma_1^* / \prod_{i \neq i^*}^{l} (z_i^{t_i^*})^{s^*} = (\mathbf{u}^{x^*})^a (z_{i^*}^{t_{i^*}^*} h)^{s^*}.$$

Therefore we apply the same method as in Theorem 4.2 since $f(x^*) \neq 0$ and $t_{i^*}^* = t_{i^*}$ to extract a solution g^{ab}.

Analysis. The analysis is similar to Theorem 3.3 and Theorem 4.2:

$$\Pr[\mathsf{success}] \geq \frac{1}{|\mathcal{T}_{i^*}|} \frac{\varepsilon}{2} - (\varepsilon_{\mathsf{PRF}} + \varepsilon_{\mathsf{CH}}) \overset{*}{\geq} \frac{\varepsilon^{c/m+1}}{2^{2+c/m} \cdot q^{c+c/m}} - (\varepsilon_{\mathsf{PRF}} + \varepsilon_{\mathsf{CH}})$$

(*) since by Lemma 3.5 we have $|\mathcal{T}_{i^*}| \leq 2 \cdot \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{c/m}$. Here, $\varepsilon_{\mathsf{PRF}}$ is the advantage for a suitable adversary on PRF and $\varepsilon_{\mathsf{CH}}$ is the advantage to produce a collision for CH, both negligible in the security parameter. □

5 Our RSA-based scheme

In this section we construct a stateless signature scheme $\mathsf{SIG}_{\mathsf{opt}}^{\mathsf{RSA}}$ secure under the RSA assumption. The result is the most efficient RSA-based scheme currently known.

The prototype for our construction is the stateful RSA-based scheme of Hohenberger and Waters [21] which we reference as $\mathsf{SIG}_{\mathsf{HW09}}^{\mathsf{RSA}}$ from now on. We first show that a stripped-to-the-basics variation of their scheme (which is tag-based but stateless), denoted $\mathsf{SIG}^{\mathsf{RSA}}$, is mildly secure, i.e., EUF-naCMA$_m^*$ secure. Subsequently, we apply our generic transformation from Section 3 and add a chameleon hash to construct a fully secure stateless scheme. Finally we apply common aggregation techniques which yield the optimized scheme $\mathsf{SIG}_{\mathsf{opt}}^{\mathsf{RSA}}$.

5.1 Preliminaries

Definition 5.1 (RSA assumption). *Let $N \in \mathbb{N}$ be the product of two distinct safe primes P and Q with $2^{\frac{k}{2}} \leq P, Q \leq 2^{\frac{k}{2}+1} - 1$. Let e be a randomly chosen positive integer less than and relatively prime to $\varphi(N) = (P-1)(Q-1)$. For $y \leftarrow \mathbb{Z}_N^\times$ we call the triple (N, e, y)* RSA challenge. *The* RSA assumption *holds if for every PPT algorithm A the probability*

$$\Pr\left[A(N, e, y) = x \wedge x^e \equiv y \ (mod\ N)\right]$$

is negligible for a uniformly chosen RSA challenge (N, e, y).

Lemma 5.2 (Shamir's trick [31, 10]). *Given $x, y \in \mathbb{Z}_N$ together with $a, b \in \mathbb{Z}$ such that $x^a = y^b$ and $\gcd(a, b) = 1$, there is an efficient algorithm for computing $z \in \mathbb{Z}_N$ such that $z^a = y$.*

Lemma 5.3. *Let $\pi(n)$ denote the number of primes $p \leq n$. For $n \in \mathbb{N}, n \geq 2^{21}$ we have*

$$\frac{n}{\log_2(n)} \leq \pi(n) \leq \frac{2n}{\log_2(n)}$$

Proof. This Lemma is just a variation of the well-known prime bound $\frac{n}{\log(n)+2} \leq \pi(n) \leq \frac{n}{\log(n)-4}$ for $n \geq 55$ [29]. We find

- $\frac{n}{\log_2(n)} \leq \frac{n}{\log(n)+2}$ since $\log_2(n) \geq \log(n) + 2$ for $n \geq 92 \geq e^{2/(\log_2(e)-1)}$ and
- $\frac{n}{\log(n)-4} \leq \frac{2n}{\log_2(n)}$ since $\log(n) - 4 \geq \frac{1}{2}\log_2(n)$ for $n \geq 2^{21} \geq e^{8/(2-\log_2(e))}$

which proves the Lemma. □

Lemma 5.4. *For $k \in \mathbb{N}$ we define $\mathbb{P}_k^* := \{p \text{ prime } | 2^{\frac{k}{2}} \leq p \leq 2^k\}$. It holds $|\mathbb{P}_k^*| > \frac{2^k}{5k}$.*

Proof.

$$|\mathbb{P}_k^*| = \pi(2^k) - \pi(2^{\frac{k}{2}}) \geq \pi(2^k) - \pi(2^{\lfloor \frac{k}{2} \rfloor}) = \sum_{i=1}^{\lceil k/2 \rceil} (\pi(2^{i+\lfloor \frac{k}{2} \rfloor}) - \pi(2^{i+\lfloor \frac{k}{2} \rfloor - 1}))$$

$$\overset{*}{>} \sum_{i=1}^{\lceil k/2 \rceil} \frac{2^{i+\lfloor \frac{k}{2} \rfloor - 1}}{3\log(2)(i + \lfloor \frac{k}{2} \rfloor)} = \frac{1}{6\log(2)} \sum_{i=1}^{\lceil k/2 \rceil} \frac{2^{i+\lfloor \frac{k}{2} \rfloor}}{i + \lfloor \frac{k}{2} \rfloor} > \frac{1}{6\log(2)} \frac{2^k}{k} > \frac{2^k}{5k}$$

(*) by [32], Theorem 5.8 (Betrand's postulate): $\pi(2^l) - \pi(2^{l-1}) \geq \frac{2^{l-1}}{3\log(2)l}$ □

Lemma 5.5. *For $\ell, k \in \mathbb{N}$ we have that for all sets $\mathcal{X} \subseteq [2^k]$ with $|\mathcal{X}| \leq 2^\ell$*

$$\Pr[p \leftarrow \mathbb{P}_k^* : \exists x \in \mathcal{X} \textit{ such that } p|x] \leq \frac{10k}{2^{k-\ell}}$$

where $\mathbb{P}_k^*$ is the set of primes p with $2^{\frac{k}{2}} \leq p \leq 2^k$.

Proof. For any $x \in [2^k]$ we have $|\{p \in \mathbb{P}_k^* : p|x\}| \leq 2$ since the product of any three elements of $\mathbb{P}_k^*$ is bigger than 2^k. Hence

$$\Pr[p \leftarrow \mathbb{P}_k^* : \exists x \in \mathcal{X} \text{ such that } p|x] \overset{(1)}{\leq} \frac{2|\mathcal{X}|}{|\mathbb{P}_k^*|} \overset{(2)}{\leq} \frac{2 \cdot 2^l}{|\mathbb{P}_k^*|} \overset{(3)}{\leq} \frac{2 \cdot 2^\ell}{\frac{2^k}{5k}} = \frac{10k}{2^{k-\ell}}$$

(1) by the union bound, (2) by assumption $|\mathcal{X}| \leq 2^\ell$, (3) by Lemma 5.4. □

5.2 EUF-naCMA$_m^*$-Secure Signature Scheme

The basic scheme $\mathsf{SIG}^{\mathsf{RSA}}$. Let $N = PQ$ be an RSA modulus consistent with the RSA assumption (Definition 5.1). Basically, a $\mathsf{SIG}^{\mathsf{RSA}}$ signature for a message-tag-pair (M, t) is a tuple $((\mathbf{u}^M)^{\frac{1}{p}} \bmod N, t)$ where p is a prime derived from the tag t. Analogously to our CDH scheme (Section 4), we define $\mathbf{u}^M := \prod_{i=0}^m u_i^{M^i}$ using quadratic residues $(u_i)_{i=0}^m$ to allow for the signing of up to m messages with the same tag. The message space is $\{0,1\}^\ell$ where we pick $\ell = k/2$ for our realization – we will need later that $\frac{1}{2^{k-\ell}}$ is negligible. To construct a mapping from tags to primes we use a technique from [20] and [18]: For a PRF $\mathsf{PRF}^{\{0,1\}^k}$, a corresponding key $\kappa \leftarrow \{0,1\}^k$, and a random bitstring $b \leftarrow \{0,1\}^k$, we define

$$\mathsf{P}_{(\kappa,b)}(t) := \mathsf{PRF}_\kappa^{\{0,1\}^k}(t||\mu_t) \oplus b$$

```
$\mathsf{Gen}_\mathsf{t}(1^k)$
  Pick modulus $N = PQ$
  $u_i \leftarrow QR_N \ (i \in \{0, \dots, m\})$
  $\kappa \leftarrow \{0,1\}^k$
  $b \leftarrow \{0,1\}^k$
  $pk := (N, (u_i)_{i=0}^m, \kappa, b)$
  $sk := (P, Q)$
  return $(pk, sk)$
```

```
$\mathsf{Sig}_\mathsf{t}(sk, M, t)$
  $p := \mathsf{P}_{(\kappa,b)}(t)$
  $\hat{\sigma} := (\prod_{i=0}^m u_i^{M^i})^{\frac{1}{p}} \bmod N$
  return $(\hat{\sigma}, t)$
```

```
$\mathsf{Ver}_\mathsf{t}(pk, M, \sigma = (\hat{\sigma}, t))$
  if $t \notin \mathcal{T}$
    return 0
  $p := \mathsf{P}_{(\kappa,b)}(t)$
  if $\hat{\sigma}^p \not\equiv \prod_{i=0}^m u_i^{M^i} \bmod N$
    return 0
  else
    return 1
```

Figure 6: The tag-based RSA scheme $\mathsf{SIG}^{\mathsf{RSA}}$.

where $\mu_t := \min\{\mu \in \mathbb{N} : \mathsf{PRF}_\kappa^{\{0,1\}^k}(t||\mu) \oplus b \text{ is prime}\}$ and $||$ denotes the concatenation of bitstrings.[5] We call μ_t the *resolving index* of t. The complete scheme $\mathsf{SIG}^{\mathsf{RSA}}$ is depicted in Figure 6.

Differences to $\mathsf{SIG}_{\mathsf{HW09}}^{\mathsf{RSA}}$. For readers acquainted with the stateful Hohenberger-Waters construction [21], also known as HW09a, we give a quick overview how $\mathsf{SIG}^{\mathsf{RSA}}$ relates to its prototype $\mathsf{SIG}_{\mathsf{HW09}}^{\mathsf{RSA}}$. To have the least amount of overhead, we first removed all components from $\mathsf{SIG}_{\mathsf{HW09}}^{\mathsf{RSA}}$ that are not required to prove the scheme EUF-naCMA$_m^*$ secure. This includes the chameleon hash (we are in a non-adaptive setting) and logarithm-of-tag-construction in the exponent (we guess from a small set of tags only). Our setup of $\mathsf{P}_{(\kappa,b)}$ slightly differs from the one in $\mathsf{SIG}_{\mathsf{HW09}}^{\mathsf{RSA}}$ since we do need that *every* tag is mapped to a prime.

5.3 EUF-naCMA$_m^*$ security

Theorem 5.6. *If F is a PPT EUF-naCMA$_m^*$-adversary for $\mathsf{SIG}^{\mathsf{RSA}}$ with advantage $\varepsilon := \mathsf{Adv}_{\mathsf{SIG}^{\mathsf{RSA}},F}^{\mathsf{euf\text{-}nacma}_m^*}(k)$ asking for $q := q(k)$ signatures, then it can be used to efficiently solve an RSA challenge according to Definition 5.1 with probability at least*

$$\frac{\varepsilon}{q'k^2} - \frac{\varepsilon_{\mathsf{PRF}}}{q'k^2} - \frac{\varepsilon_{\mathsf{PRF}}'}{q'} - \mathbf{O}\left(\frac{1}{2^{k/2}}\right)$$

where q' denotes the number of distinct tags queried by F and $\varepsilon_{\mathsf{PRF}}$ and $\varepsilon_{\mathsf{PRF}}'$ are the advantages of suitable distinguishers for the PRF.

Proof. We first describe the simulation and subsequently provide a detailed analysis. In the following (N, e^*, y) denotes the RSA challenge given to the simulator. W.l.o.g. we assume that the adversary F queries exactly $q = q' \cdot m > 0$ signatures where q' is the number of distinct tags $(t_i)_{i \in [q']}$ and m the number of messages queried for each tag.

If e^* is not an odd prime with $\log_2(e^*) \geq \frac{k}{2}$, we abort. Otherwise we proceed as follows:

Public key setup. We guess an index $i^* \leftarrow [q']$ and write $t^* := t_{i^*}$ for the corresponding tag. Intuitively, t^* is the tag we will use to embed our challenge. $(M_i^*)_{i \in [m]}$ denotes the list of messages queried for tag t^*. Remember that e^* is an odd prime. We pick κ and $\mu_t^* \leftarrow [k^2]$ at random and compute $b := \mathsf{PRF}_\kappa^{\{0,1\}^k}(t^*||\mu_t^*) \oplus e^*$. If $\mathsf{P}_{(\kappa,b)}(t^*) \neq e^*$ or if $\mathsf{P}_{(\kappa,b)}(t_i) = e^*$ for any $t_i \neq t^*$ $(i \in [q'])$, we abort. We compute $p_i := \mathsf{P}_{(\kappa,b)}(t_i)$ for $i \in [q']$ and set $p^* := p_{i^*}$ (note that $p^* = e^*$). Next we define two polynomials over $\mathbb{Z}[X]$. The first one, $f(X) := \prod_{i=1}^m (X - M_i^*)$, is determined by the messages queried for the challenge tag t^*. We have $f(X) = \sum_{i=0}^m \alpha_i X^i$

[5] $\mathsf{P}_{(\kappa,b)}(t)$ can be computed in expected polynomial time but not in strict polynomial time. However, one can simply pick an upper bound $\overline{\mu}$ and set $\mathsf{P}_{(\kappa,b)}(t) = p$ for some arbitrary but fix prime p if $\mu_t > \overline{\mu}$ for the resolving index of t μ_t. For a proper $\overline{\mu}$ the event $\mu_t > \overline{\mu}$ will only occur with negligible probability (see Theorem 5.6, Game 2).

for some $\alpha_i \in \mathbb{Z}$. For the second one we pick $\beta_i \leftarrow \mathbb{Z}_{N/4}$ for $i \in \{0, \dots, m\}$ and set $g(X) := \sum_{i=0}^{m} \beta_i X^i$. For a more comprehensive notation we define $\phi_I := \prod_{i \in ([q'] \setminus I)} p_i$ for any set of indexes $I \subseteq [q']$, $\phi := \phi_\emptyset$ and $\mathbf{u}^M := \prod_{i=0}^{m} u_i^{M^i}$. We set

$$u_i := (y^{\phi_{\{i^*\}}\alpha_i + \phi\beta_i})^2$$

Note that for any message M

$$\mathbf{u}^M = (y^{\phi_{\{i^*\}} f(M) + \phi g(M)})^2$$

We send the public key $(N, (u_i)_{i=0}^{m}, \kappa, b)$ to the adversary.

Signing. For each query (M, t) we compute the corresponding signature σ as follows. If $t = t^*$, σ is $(\hat{\sigma}, t)$ where

$$\hat{\sigma} := (y^{\phi_{\{i^*\}} g(M)})^2$$

We use the fact that $M = M_i^*$ for some $i \in [m]$ and hence $f(M) = 0$ to verify

$$\left(\mathbf{u}^M\right)^{\frac{1}{p^*}} \equiv ((y^{\phi_{\{i^*\}} f(M) + \phi g(M)})^2)^{\frac{1}{p^*}} \equiv (y^{\phi_{\{i^*\}} g(M)})^2 \equiv \hat{\sigma}$$

If $t = t_i \neq t^*$ $(i \in [q'])$, the corresponding signature σ is $(\hat{\sigma}, t)$ where

$$\hat{\sigma} := (y^{\phi_{\{i,i^*\}} f(M) + \phi_{\{i\}} g(M)})^2$$

We verify

$$\left(\mathbf{u}^M\right)^{\frac{1}{p_i}} \equiv ((y^{\phi_{\{i^*\}} f(M) + \phi g(M)})^2)^{\frac{1}{p_i}} \equiv (y^{\phi_{\{i,i^*\}} f(M) + \phi_{\{i\}} g(M)})^2 \equiv \hat{\sigma}$$

Finally, we send all signatures to the adversary.

Extract from forgery. The adversary responds with (M, σ) where $\sigma = (\hat{\sigma}, t)$ for some state $t \in \{t_1, \dots, t_{q'}\}$. If σ is not a valid forgery, we abort. Otherwise, since the verification equation holds, we have

$$\sigma^{p^*} \equiv \mathbf{u}^M \equiv (y^{\phi_{\{i^*\}} f(M) + \phi g(M)})^2 \equiv y^{2\phi_{\{i^*\}} f(M)} y^{2\phi g(M)}$$

and hence

$$(\sigma / y^{2\phi_{\{i^*\}} g(M)})^{p^*} \equiv y^{2\phi_{\{i^*\}} f(M)}$$

Note that $f(M) \neq 0$ since $M \neq M_i^*$ for $i \in [m]$. Clearly, p^* does not divide $2\phi_{\{i^*\}}$. If $\gcd(p^*, f(M)) \neq 1$, we abort. Otherwise we use Shamir's trick (Lemma 5.2) to compute x such that $x^{p^*} \equiv y \ (mod\ N)$. Since $p^* = e^*$ by construction, x is the output of the simulator and a solution to the RSA challenge (N, e^*, y) it was given.

Analysis. We show that the adversary F cannot distinguish effectively between the experiment and the simulation. Let X_i denote the event that the adversary is successful in Game i.

Game 0. In Game 0 the simulator runs the original EUF-naCMA$_m^*$ experiment and hence we have $\Pr[X_0] = \varepsilon$.

Game 1. In Game 1 the simulator aborts if e^* is not an odd prime with $\log_2(e^*) \geq \frac{k}{2}$. $\mathsf{abort}_{\mathsf{challenge}}$ denotes the corresponding event.

$$\Pr[\mathsf{abort}_{\mathsf{challenge}}] = \frac{\pi(2^{\frac{k}{2}})}{\pi(2^k)} \overset{(*)}{\leq} \frac{2 \cdot 2^{\frac{k}{2}} \cdot k}{\frac{k}{2} 2^k} = \frac{4}{2^{\frac{k}{2}}}$$

Here, $(*)$ follows by Lemma 5.3. Hence

$$\Pr[X_1] \geq \Pr[X_0] - \frac{4}{2^{\frac{k}{2}}}$$

Game 2. In Game 2 we set up $\mathsf{P}_{(\kappa,b)}$ as described in the simulation above. Concretely, we choose $\mu_t^* \leftarrow [k^2]$ and set $b := \mathsf{PRF}_\kappa^{\{0,1\}^k}(t^*||\mu_t^*) \oplus e^*$. If μ_t^* is not the resolving index of $\mathsf{P}_{(\kappa,b)}(t^*)$, we abort. For a detailed analysis of this Game see [19], Proof of Theorem 4.1, Games 8-10. We have

$$\Pr[X_2] \geq \frac{1}{k^2}\left(\Pr[X_1] - \varepsilon_{\mathsf{PRF}} - \frac{1}{2^k}\right)$$

where $\varepsilon_{\mathsf{PRF}}$ is the advantage for distinguishing the PRF $\mathsf{PRF}^{\{0,1\}^k}$ from a truly random function.

Game 3. In Game 3 we abort if $\mathsf{P}_{(\kappa,b)}(t_i) = e^*$ for some $t_i \neq t^*$ $(i \in [q'])$ $\mathsf{abort}_{\mathsf{coll}}$ denotes the corresponding event. If PRF is a truly random function, then the output of $\mathsf{P}_{(\kappa,b)}(t_i)$ is a uniform k-bit prime. By Lemma 5.3 there are at least $\frac{2^k}{k}$ such primes. Hence the probability of a collision with e^* is at most $\left(\frac{k}{2^k}\right)^{q'-1}$. Using this, we can construct an adversary that distinguishes PRF from a truly random function with advantage $\varepsilon_{\mathsf{PRF}}' \geq \Pr[\mathsf{abort}_{\mathsf{coll}}] - \left(\frac{k}{2^k}\right)^{q'-1}$. We have

$$\Pr[X_3] \geq \Pr[X_2] - \varepsilon_{\mathsf{PRF}}' - \left(\frac{k}{2^k}\right)^{q'-1}$$

Game 4. In Game 4 we do not pick $(u_i)_{i=0}^m$ at random but set them as described above. Since the β_i are randomly chosen and blind the α_i known to the adversary this yields a correct distribution.

$$\Pr[X_4] = \Pr[X_3]$$

The view of the adversary in this Game is exactly the view in the simulation described above.

Let now $\mathsf{abort}_{\mathsf{gcd}}$ denote the event that $\gcd(p^*, f(M)) \neq 1$ or $t \neq t^*$. Remember that messages are bitstrings of length ℓ. The range of f is a set $\mathcal{X} \subseteq [2^k]$ containing 2^ℓ elements at most. Therefore, by Lemma 5.5 and with $\ell = k/2$, we have

$$\Pr[\mathsf{abort}_{\mathsf{gcd}}] \leq \frac{10k}{2^{k-\ell}} = \frac{10k}{2^{k/2}}$$

Finally, we summarize and see that if the simulator guesses t^* correctly, it is successful with probability at least

$$\frac{1}{q'k^2}\left(\varepsilon - \varepsilon_{\mathsf{PRF}} - k^2\varepsilon_{\mathsf{PRF}}'\right) - \mathbf{O}\left(\frac{1}{2^{k/2}}\right)$$

□

Now, by Theorem 5.6, our generic transformation from Section 3 applied to $\mathsf{SIG}^{\mathsf{RSA}}$ yields an EUF-naCMA-secure signature scheme. Finally, we use chameleon hashing [23] to generically construct the fully secure scheme $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{gen}}$, like for instance the RSA-based chameleon hash from [20, Appendix C].

5.4 Optimizations

The resulting signature scheme of the previous section $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{gen}}$ may be EUF-CMA-secure but is not very compact yet. In addition to parameters for the chameleon hash, a signature of $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{gen}}$ consists of $l = \lfloor \log_c(k) \rfloor$ $\mathsf{SIG}^{\mathsf{RSA}}$ signatures. This can be improved considerably to constant size signatures by generic aggregation.

Figure 7 depicts the resulting scheme $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{opt}}$ for the two parameters l (which implicitly contains the granularity parameter c) and m. We still use l tags (intuitively representing the l instances of the original scheme) for signing and verification. However, the public key's size depends only on m (which is a fixed parameter) and the signature size is constant: We need one group element and randomness for the chameleon hash (which is typically also about the size

```
$\mathsf{Gen}(1^k)$
  Pick modulus $N = PQ$
  $u_i \leftarrow QR_N$ $(i \in \{0, \dots, m\})$
  $\kappa \leftarrow \{0,1\}^k$
  $b \leftarrow \{0,1\}^k$
  $(\mathsf{CH}, \tau) \leftarrow \mathsf{CHGen}(1^k)$
  $pk := (N, (u_i)_{i=0}^m, \kappa, b, \mathsf{CH})$
  $sk := (P, Q)$
  return $(pk, sk)$
```

```
$\mathsf{Sig}(sk, M)$
  Pick uniform $r$ for $\mathsf{CH}$
  $x := \mathsf{CH}(M, r)$
  for $i := 1$ to $l$ do
    $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x)$
    $p_i := \mathsf{P}_{(\kappa,b)}(t_i)$
  $p := \prod_{i \in [l]} p_i$
  $\hat{\sigma} := (\prod_{i=0}^m u_i^{x^i})^{\frac{1}{p}} \bmod N$
  return $(\hat{\sigma}, r)$
```

```
$\mathsf{Ver}(pk, M, (\hat{\sigma}, r))$
  $x := \mathsf{CH}(M, r)$
  for $i := 1$ to $l$ do
    $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x))$
    $p_i := \mathsf{P}_{(\kappa,b)}(t_i)$
  $p := \prod_{i \in [l]} p_i$
  if $\hat{\sigma}^p \not\equiv \prod_{i=0}^m u_i^{x^i} \bmod N$
    return 0
  else
    return 1
```

Figure 7: The optimized RSA-based signature scheme $\mathsf{SIG}_{\mathsf{opt}}^{\mathsf{RSA}}$.

of a group element). Additionally to $\mathsf{PRF}^{\{0,1\}^k}$ we now need functions $(\mathsf{PRF}^{\mathcal{T}_i})_{i \in [l]}$ to generate the tags for a signature. We can construct all of these functions from a single PRF $\mathsf{PRF}^{\{0,1\}^*}$ with sufficiently long output and use $\kappa \in \{0,1\}^k$ as its key.

Theorem 5.7. *Let F be a PPT EUF-CMA adversary against $\mathsf{SIG}_{\mathsf{opt}}^{\mathsf{RSA}}$ with advantage $\varepsilon := \mathsf{Adv}_{\mathsf{SIG}^{\mathsf{RSA}},F}^{\mathsf{euf\text{-}cma}}(k)$ asking for $q := q(k)$ signatures (at most). Then it can be used to efficiently solve an RSA challenge according to Definition 5.1 with probability at least*

$$\frac{\varepsilon^{c/m+1}}{k^2 \cdot 2^{c/m+1} \cdot q^{c+c/m}} - \frac{\varepsilon}{2} - \varepsilon_{\mathsf{PRF}} - \varepsilon_{\mathsf{CH}} - \mathbf{O}\left(\frac{1}{2^{k/2}}\right)$$

where $\varepsilon_{\mathsf{PRF}}$ and $\varepsilon_{\mathsf{CH}}$ are the success probabilities for breaking PRF and CH respectively.

Proof. Since the proof is very similar to the combination of proofs Theorem 3.3 and Theorem 5.6 we only describe the interesting parts. Additionally, we omit the chameleon hash here and prove the EUF-naCMA security of the corresponding modified scheme (with $x := M$ instead of $x := \mathsf{CH}(M, r)$ and no randomness for the chameleon hash in the signature). The chameleon hash is then added at the end using generic arguments [23].

Again, w.l.o.g. we assume that the adversary will ask for exactly $q > 0$ signatures.

Setup. Analogously to Theorem 3.3 we use the select algorithm to pick an i^*. Intuitively, i^* represents one of the l instances and $\mathcal{T}_{i^*}$ is the set of tags used for this instance. We will embed the challenge only in this instance and simulate all the others.

We start off by picking a random key κ for the PRF PRF. For each queried message M_j $(j \in [q])$ we compute the tags $t_j^{(i)} := \mathsf{Samp}(\mathcal{T}_i, \mathsf{PRF}_\kappa(M_j))$. Subsequently, we guess a tag $t^* \leftarrow \mathcal{T}_{i^*}$ and abort if the list of tags for the i^*th instance $(t_j^{(i^*)})_{j \in [q]}$ contains any tag more than m times.

Next we embed the challenge exponent. Like in Theorem 5.6 we set up $\mathsf{P}_{(\kappa,b)}$ such that $\mathsf{P}_{(\kappa,b)}(t^*) = e^* =: p^*$ and abort if $\mathsf{P}_{(\kappa,b)}(t_j^{(i)}) = p^*$ for any $t_j^{(i)} \neq t^*$. Afterwards we compute primes $p_j^{(i)} := \mathsf{P}_{(\kappa,b)}(t_j^{(i)})$ corresponding to the instance tags for $i \in l, j \in [q]$.

Finally and analogously to Theorem 3.3 we define two polynomials f and g. For the construction of f we use the messages M_j with $t_j^{(i^*)} = t^*$. If there are no such messages we define $f(M) := 1$. For $i \in [m]$ we set

$$u_i := (y^{\pi_f \alpha_i + \pi_g \beta_i})^2$$

where π_f is the product of all the primes for all instances computed above omitting occurrences of $p^*$and $\pi_g := \pi_f \cdot p^*$.

Signing. Signing works exactly like in Theorem 3.3: For the signature of M_j we use the tags $(t_j^{(i)})_{i \in [l]}$. If $t_j^{(i^*)} = t^*$, we make use of the fact that $f(M_j) = 0$ and sign by omitting the primes

$(p_j^{(i)})_{i\in[l]}$ in π_g. Otherwise, if $t_j^{(i^*)} \neq t^*$, and hence $p_j^{(i)} \neq p^*$ for $i \in [l]$, we can sign by omitting the corresponding factors in π_f and π_g.

Extract from forgery. We receive a message M and a forgery σ. If $\mathsf{Samp}(\mathcal{T}_i, \mathsf{PRF}_\kappa(M)) \neq t^*$, we abort. Otherwise, if the adversary was successful, the verification equation holds and we can, analogously to Theorem 5.6, compute

$$(\sigma/y^{2\pi_f g(M)})^{p^*} \equiv y^{2\pi_f f(M)}$$

Again we need $\gcd(p^*, 2\pi_f f(M)) = 1$ as a prerequisite for Shamir's trick (Lemma 5.2). We can then compute a solution for the given RSA challenge.

Analysis. The analysis is very similar to that of Theorem 5.6. The following differences occur

- More primes are computed using $\mathsf{P}_{(\kappa,b)}$. All of these $q \cdot l$ primes must be distinct from p^*. Hence instead of q' in Theorem 5.6, Game 3 we have $q \cdot l$. However, the relevant probability is still negligible $(\left(\frac{k}{2^k}\right)^{lq-1}$ instead of $\left(\frac{k}{2^k}\right)^{q'-1})$.
- There is an additional abort if more than m of the queried messages have the same tag in instance i^*. Analogously to Theorem 3.3, we lose $\varepsilon_{\mathsf{PRF}} - \frac{\varepsilon}{2}$ here.
- The chance for the simulator to guess the tag used for the forgery correctly is $\frac{1}{|\mathcal{T}_{i^*}|}$ instead of $\frac{1}{q'}$.

Finally, we generically add a chameleon hash with the techniques of [23] to reach full security which reduces the success negligibly by the $\varepsilon_{\mathsf{CH}}$ (the success of an adversary to produce a collision for the chameleon hash). Hence the simulator is successful with probability at least

$$\frac{1}{|\mathcal{T}_{i^*}|}\frac{\varepsilon(k)}{k^2} - \varepsilon_{\mathsf{PRF}} - \frac{\varepsilon(k)}{2} - \varepsilon_{\mathsf{CH}} - \mathbf{O}\left(\frac{1}{2^{k/2}}\right) \overset{*}{\geq} 2^{-1}\left(\frac{\varepsilon(k)}{2\cdot q^{m+1}}\right)^{c/m}\frac{\varepsilon(k)}{k^2} - \frac{\varepsilon(k)}{2} - \varepsilon_{\mathsf{PRF}} - \varepsilon_{\mathsf{CH}} - \mathbf{O}\left(\frac{1}{2^{k/2}}\right)$$

(*) since by Lemma 3.5 we have $|\mathcal{T}_{i^*}| \leq 2 \cdot \left(\frac{2\cdot q^{m+1}}{\varepsilon(k)}\right)^{c/m}$. □

6 Our SIS-based Scheme

Let us now describe our SIS-based signature scheme. Again we start with constructing a tag-based signature scheme and prove EUF-naCMA$^*_m$-security, but only for $m = 1$. This scheme can be converted into a fully EUF-CMA secure signature scheme by applying the generic transformation from Section 3.

Note that in the previous chapters we have used the character m to denote the number of repeating tags in the EUF-naCMA$^*_m$ security experiment. Unfortunately, the same character is commonly used in lattice-based cryptography to denote the dimension of a matrix $\mathbb{Z}_p^{n\times m}$. In order to be consistent with the literature, and since we consider only EUF-naCMA$^*_1$-security in the sequel, we will from now on use m to denote the dimension of matrices.

6.1 Preliminaries

In this section we summarize some known facts about lattices, as far as relevant for our signature scheme and its security analysis. The reader familiar with lattice-based cryptography, in particular with [15, 1, 8, 6], can safely skip this section.

6.1.1 Lattices and SIS

For positive integers p, m, n and $A \in \mathbb{Z}_p^{n\times m}$, the m-dimensional integer lattices $\Lambda_p^\perp(A)$ and Λ_p^u are defined as

$$\Lambda_p^\perp(A) := \{e \in \mathbb{Z}^m : Ae = 0 \bmod p\}$$
$$\Lambda_p^u(A) := \{e \in \mathbb{Z}^m : Ae = u \bmod p\}$$

Definition 6.1. *The* (p,n,m,β)-small integer solution (SIS) *problem (in* ℓ_2*-norm, denoted* $\|\cdot\|$*) is: given* $p \in \mathbb{N}$, $A \in \mathbb{Z}_p^{n\times m}$, *and* $\beta \in \mathbb{R}$, *find a non-zero vector* $e \in \mathbb{Z}^m$ *such that* $Ae = 0 \bmod p$ *and* $\|e\| \le \beta$.

Fact 1 (Theorem 4 of [1]). *Let* $p \ge 3$ *be odd and let* $m \ge 6n\log p$. *There exists a probabilistic polynomial-time algorithm* TrapGen *that, on input* (p,n), *outputs a matrix* $A \in \mathbb{Z}_q^{n\times m}$ *which is statistically close to uniform, and a basis* T_A *of* $\Lambda_p^{\perp}(A)$ *such that* $\|\tilde{T}_A\| \le \mathbf{O}(\sqrt{m})$, *where* $\tilde{T}_A$ *denotes the Gram-Schmidt orthogonalization of* T_A, *and* $\|T_A\| \le \mathbf{O}(m)$, *with all but negligible (in* n*) probability.*

Fact 2 (Theorem 4.1 of [15]). *Let* $\mathcal{D}_{\Lambda,\gamma,c}$ *denote the discrete Gaussian distribution over* Λ *with center* c *and parameter* γ. *Let* T_A *be any basis of* $\Lambda_p^{\perp}(A)$. *There exists a probabilistic polynomial-time algorithm that takes as input* (A, T_A, c, γ) *with* $\gamma \ge \|\tilde{T}_A\| \cdot \omega(\sqrt{\log m})$, *and whose output distribution is identical to* $\mathcal{D}_{\Lambda_p^{\perp}(A),\gamma,c}$ *up to a negligible statistical distance.*

To simplify our notation we write $\mathcal{D}_{\Lambda_p^{\perp}(A),\gamma}$ for $\mathcal{D}_{\Lambda,\gamma,c}$ if $c = 0$.

Fact 3 (Lemma 4.4 of [26]). *Let* $e \leftarrow \mathcal{D}_{\Lambda_p^{\perp}(A),\gamma}$. *Then the probability that* $\|e\| > \gamma\sqrt{m}$ *is negligible (in* n*).*

Fact 4 (Algorithm SampleLeft of [1]). *Let*

- $A \in \mathbb{Z}_p^{n\times m}$ *be a matrix of rank* n, *and* T_A *be a (short) basis* $\Lambda_p^{\perp}(A)$,
- $F \in \mathbb{Z}_p^{n\times m}$,
- $u \in \mathbb{Z}_p^n$ *be a vector, and*
- $\gamma \ge \|\tilde{T}_A\| \cdot \omega(\sqrt{\log m})$ *be a Gaussian parameter.*

There exists an efficient algorithm SmpL *that takes as input* (A, T_A, F, u, γ), *and outputs* $e \in \mathbb{Z}_p^{2m}$ *such that the distribution of* e *is statistically close to* $\mathcal{D}_{\Lambda_p^u(A|F),\gamma}$.

Fact 5 (Algorithm SampleRight of [1]). *Let*

- $A, B \in \mathbb{Z}_p^{n\times m}$ *be matrices, where* B *has rank* n *and* T_B *is a (short) basis of* $\Lambda_p^{\perp}(B)$,
- $\Delta \in \mathbb{Z}_p^{n\times n}$ *be of full rank* n,
- $R \in \{-1,1\}^{m\times m}$ *be a random matrix, and* $s_R := \|R\| = \sup_{\|x\|=1}\|Rx\|$ *(note that for random* $R \in \{-1,1\}^{m\times m}$ *we have* $s_R \le \mathbf{O}(\sqrt{\log m})$ *with overwhelming probability),*
- $u \in \mathbb{Z}_p^n$ *be a vector, and*
- $\gamma \ge \|\tilde{T}_B\| \cdot s_R \cdot \omega(\sqrt{\log m})$ *be a Gaussian parameter.*

There exists an efficient algorithm SmpR *that takes as input* $(A, B, T_B, \Delta, R, u, \gamma)$, *and outputs* $e \in \mathbb{Z}_p^{2m}$ *such that the distribution of* e *is statistically close to* $\mathcal{D}_{\Lambda_p^u(A|F),\gamma}$, *where* $F := AR + \Delta B$.

Fact 6 (Lemma 13 of [1]). *Let* $p \in \mathbb{N}$ *be an odd prime and let* $m > (n+1)\log p + \omega(\log n)$. *Let* $R \leftarrow \{-1,1\}^{m\times m}$ *and* $A, A' \leftarrow \mathbb{Z}_p^{n\times m}$ *be uniformly random. Then the distribution of* (A, AR) *is statistically close to the distribution of* (A, A').

Fact 7 (Corollary 5.4 of [15], Fact 14 of [6]). *Let* p *be a prime and let* n, m *be integers such that* $m \ge 2n\log p$. *Let* $e \leftarrow \mathcal{D}_{\mathbb{Z}^m,\gamma}$, *where* $\gamma \ge \omega(\sqrt{\log m})$. *Then for all but at most a* $2p^{-n}$ *fraction of all matrices* $A \in \mathbb{Z}_p^{n\times m}$ *the distribution of the syndrome* $u := Ae \bmod p$ *is statistically close to uniform over* $\mathbb{Z}_p^n$. *Furthermore, the conditional distribution of* e, *given* u, *is* $\mathcal{D}_{\Lambda_p^u(A),\gamma}$.

6.1.2 Full-Rank Difference Hashing

We will need a map $H : \mathcal{T} \to \mathbb{Z}_p^{n\times n}$ that allows to map tags from $\mathcal{T}$ to matrices in $\mathbb{Z}_p^{n\times n}$, with the property that the difference matrix $\Delta := H(t) - H(t')$ has full rank for all $t, t' \in \mathcal{T}$ with $t \neq t'$.

```
$\mathsf{Gen_t}(1^k)$
    $(A, T_A) \leftarrow \mathsf{TrapGen}(p, n)$
    $Z, Y \leftarrow \mathbb{Z}_q^{n \times m}$
    $U \leftarrow \mathbb{Z}_q^{n \times \ell}$
    $v \leftarrow \mathbb{Z}_p^n$
    $sk := T_A$
    $pk := (U, A, Z, Y, v)$
    return $(sk, pk)$
```

```
$\mathsf{Sig_t}(sk, M, t)$
    $G_t := Z + H(t)Y \bmod p$
    $u := UM + v$
    $e \leftarrow \mathsf{SmpL}(A, T_A, G_t, u, \gamma)$
    return $(e, t) \in \mathbb{Z}_p^{2m} \times \mathcal{T}$
```

```
$\mathsf{Ver_t}(pk, M, \sigma = (e, t))$
    if $t \notin \mathcal{T}$ or $M \notin \{0,1\}^\ell$
        return 0
    if $e \leq 0$ or $\|e\| > \sqrt{2m} \cdot \gamma$
        return 0
    $G_t := Z + H(t)Y \in \mathbb{Z}_p^{n \times 2m}$
    if $(A|G_t)e = UM + v \bmod p$
        return 1
    else return 0
```

Figure 8: The tag-based SIS scheme.

Definition 6.2. *Let p be prime, $n \in \mathbb{N}$ be a positive integer, and let $\mathcal{T}$ be a set. We say that a hash function $H : \mathcal{T} \to \mathbb{Z}_p^{n \times n}$ is a* full-rank difference *hash function, if H is efficiently computable, and for all distinct $t, t' \in \mathcal{T}$ holds that the difference matrix $\Delta := H(t) - H(t')$ has full rank.*

The notion of *full-rank difference* hash functions was introduced in [1], together with a simple and elegant construction of such hash functions with domain $\mathcal{T} = \mathbb{Z}_p^n$, which is suitable for our purposes.

6.2 EUF-naCMA$_1^*$-Secure Signature Scheme

Our tag-based signature scheme $\mathsf{SIG_t^{SIS}}$ is described in Figure 8. The scheme uses a *full-rank difference* hash function $H : \mathcal{T} \to \mathbb{Z}_p^{n \times n}$, and the tag space is an arbitrary set $\mathcal{T}$ such that there exists a such a hash function (e.g. $\mathcal{T} := \mathbb{Z}_p^n$, as in [1]). We use parameters $p, n, m \in \mathbb{N}$, where p is prime, and $\gamma \in \mathbb{R}$, whose choice partially depends on our security analysis and is therefore deferred to Section 6.3. Correctness of this scheme follows from Fact 3.

Theorem 6.3. *Suppose there exists an efficient adversary F breaking the EUF-naCMA$_1^*$-security of $\mathsf{SIG_t^{SIS}}$ as described in Figure 8. Then there exists an efficient algorithm* Sim *solving the (p, n, m, β)-SIS problem with $\beta = \mathbf{O}(\gamma m)$.*

Scheme $\mathsf{SIG_t^{SIS}}$ exhibits many similarities to the identity-key generation algorithm of the IBE scheme from [1], where we use tags correspond to identities of [1]. In the security proof we simulate signatures in a way very similar to the identity-key generation in [1], by embedding an additional trapdoor in the matrix G_t that allows to simulate signatures for arbitrary messages and for all tags except for one tag $t_i = t^*$ which equals the tag from the forgery (M^*, t^*) output by the forgery (since the tag-space is polynomially bounded, we can guess t^* with non-negligible probability). To this end, in the simulation matrix G_t is defined such that the additional trapdoor "vanishes" exactly for tag $t_i = t^*$.

The difference to the proof from [1] is that we must also be able to issue one signature for message-tag-pair (M_i, t_i) with $t_i = t^*$, but without knowing any trapdoor. To simulate a signature for this message-tag pair, we define the vector v contained in the public key as $v := G_{t_i} e_i - UM_i$ for a random short vector $e_i \leftarrow \mathcal{D}_{\mathbb{Z}^{2m}, \gamma}$. Note that this defines v such that e_i is a valid signature for message-tag-pair (M_i, t_i).

A successful forger F has to produce a forgery e^* for a message $M^* \neq M_i$, from which we obtain an equation $G_{t_i} e_i - UM_i = G_{t_i} e^* - UM^*$. By an adequate set-up of matrices G_{t_i} and U this equation allows us to extract a solution to the given SIS problem instance with high probability.

Proof. For simplicity let us assume a message length ℓ with $\ell = m$. The security proofs works identically for any $\ell \in [1, m]$.

Sim receives as input an SIS-challenge $A \in \mathbb{Z}_p^{n \times m}$, and runs F as a subroutine by simulating the EUF-naCMA$_1^*$-experiment for F. To this end, it proceeds as follows.

Start. Sim starts $F(1^k)$ to receive a list $(M_1, t_1), \ldots, (M_q, t_q)$ of q chosen message-tag-pairs, where $t_i \neq t_j$ for all $i \neq j$.

Setup of the public key. To create a public key, Sim chooses a full-rank difference hash function $H : \mathcal{T} \to \mathbb{Z}_p^{n \times m}$. Then it runs the algorithm of Fact 1 to generate a matrix $B \in \mathbb{Z}_p^{n \times m}$ together with a short basis $T_B \subset \Lambda_p^{\perp}(B)$ with $\|\tilde{T_B}\| \leq L$. Furthermore, it samples two random matrices $R_U, R_Z \leftarrow \{0,1\}^{m \times m}$, and defines matrices $U, Z, Y \in \mathbb{Z}_p^{n \times m}$ and vector $v \in \mathbb{Z}_p^n$ as

$$U := AR_U, \qquad Z := AR_Z - H(t_{i^*})B, \qquad Y := B \in \mathbb{Z}_p^{n \times m}, \qquad v := G_{t_{i^*}} e_{i^*} - UM_{i^*}$$

where all operations are performed modulo p, $i^* \leftarrow [q]$ is chosen uniformly random, and $e_{i^*} \leftarrow \mathcal{D}_{\mathbb{Z}^{2m},\gamma}$.

The public key is defined as (U, A, Z, Y, v). Note that matrices U, Z, Y are statistically close to uniform over $\mathbb{Z}_p^{n \times m}$ (due to Fact 6 and Fact 1), and v is statistically close to uniform (due to Fact 7), thus this is a correctly distributed public key (up to a negligibly small statistical distance).

Simulating signatures. A signature σ_i for message-tag-pair (M_i, t_i), $i \in [q]$, is computed as follows.

Case $i \neq i^*$. In this case we have

$$G_{t_i} = AR_Z - H(t_{i^*})B + H(t_i)B = AR_Z + \Delta B,$$

where $\Delta := H(t_i) - H(t_{i^*})$ is a full-rank matrix, since H is a full-rank difference hash function.

By running the algorithm from Fact 5 on input $(A, B, T_B, \Delta, R_Z, u, \gamma)$, where $u := UM_i + v$, Sim computes a low-norm non-zero vector $e_i \leftarrow \mathcal{D}_{\Lambda_p^u,\gamma}$ satisfying

$$(A|G_{t_i})e_i = UM_i + v,$$

and sets $\sigma_i := (e_i, t_i)$, which thus is a valid signature.

Case $i = i^*$. Now we have

$$G_{t_{i^*}} = AR_Z - H(t_{i^*})B + H(t_{i^*})B = AR_Z,$$

thus Sim is not able to use the trapdoor T_B to simulate a signature, since B "vanishes".

However, in this case Sim can set $\sigma_{i^*} := (e_{i^*}, t_{i^*})$. Recall that we have defined $v := G_{s_{i^*}} t_{i^*} - UM_{i^*}$ in the setup phase, thus

$$G_{t_{i^*}} e_{i^*} = UM_{i^*} + v \qquad \Longleftrightarrow \qquad G_{t_{i^*}} e_{i^*} - UM_{i^*} = v.$$

Note that e_{i^*} is correctly distributed due to Fact 7, and we have $t_i \neq t_j$ for all $i \neq j$, thus e_{i^*} is contained in exactly one signature.

Note also that this is the case where our construction and proof differ from [6], since in [6] it is never necessary to simulate a signature in the case where matrix B "vanishes", due to a different construction and security experiment.

In either case Sim is able to compute a valid and correctly distributed signature for each $i \in [q]$, and thus simulates the EUF-naCMA$_1^*$ security experiment properly. By assumption, F will thus output $(M^*, (e^*, t^*))$, where $t^* = t_i$ for some $i \in [q]$ and (e^*, t^*) is a valid signature for $M^* \notin \{M_1, \ldots, M_q\}$, with non-negligible probability.

Extracting the SIS solution. Suppose that $i = i^*$, which happens with probability $1/q$. Note that in this case it holds that

$$\begin{gathered} G_{t^*}e_{i^*} - UM_{i^*} = v = G_{t^*}e^* - UM^* \\ \Longleftrightarrow \\ (A|AR_Z)e_{i^*} - AR_UM_{i^*} = v = (A|AR_Z)e^* - AR_UM^* \\ \Longleftrightarrow \\ (A|AR_Z)(e_{i^*} - e^*) - AR_U(M_{i^*} - M^*) = 0. \end{gathered}$$

Let us write vector $\hat{e} := (e_{i^*} - e^*) \in \mathbb{Z}_p^{2m}$ as $\hat{e}^\top = (\hat{e}_1^\top, \hat{e}_2^\top)$ for two vectors $\hat{e}_1, \hat{e}_2 \in \mathbb{Z}_p^m$, and let us write $\hat{M} := (M^* - M_{i^*}) \in \{-1, 0, 1\}^m$. Then the above equation is equivalent to

$$A\hat{e}_1 + AR_Z\hat{e}_2 + AR_U\hat{M} = 0 \iff A(\hat{e}_1 + R_Z\hat{e}_2 + R_U\hat{M}) = 0.$$

Algorithm Sim computes and outputs $e := (\hat{e}_1 + R_Z\hat{e}_2 + R_U\hat{M})$ as solution to the given SIS challenge. It remains to show that e is sufficiently short and non-zero with high probability.

Note that we have $\|\hat{e}_1\| \leq \|\hat{e}\| \leq 2 \cdot \gamma\sqrt{m}$, and similarly $\|\hat{e}_2\| \leq 2 \cdot \gamma\sqrt{m}$. Note furthermore that $\|\hat{M}\| \leq \sqrt{m}$. By [1, Lemma 15] it furthermore holds that $\|R_U\| \leq 12\sqrt{2m}$ and $\|R_Z\| \leq 12\sqrt{2m}$, except for a negligibly small probability. In summary we thus have

$$\begin{aligned} \|e\| = \|\hat{e}_1 + R_Z\hat{e}_2 + R_U\hat{M}\| &\leq \|\hat{e}_1\| + \|R_Z\| \cdot \|\hat{e}_2\| + \|R_U\| \cdot \|\hat{M}\| \\ &\leq 2\gamma\sqrt{m} + 24 \cdot \sqrt{2} \cdot m\gamma + 24 \cdot \sqrt{2} \cdot m \\ &= \mathbf{O}(m\gamma). \end{aligned}$$

Finally let us show that $e \neq 0$ with high probability. Note that we must have $\hat{M} = M_{i^*} - M^* \neq 0 \in \{-1, 0, 1\}^m$, since $M_{i^*} \neq M^*$. Note furthermore that F does not receive R_Z and R_U explicitly as input, but only implicitly as (A, AR_Z, AR_U).

We will show a slightly stronger result than necessary, namely that even any *unbounded* algorithm Γ, that receives as input (A, R_Z, AR_U) (i.e. R_Z in explicit form), will output $\hat{e}_1, \hat{e}_2, \hat{M}$ such that $e := \hat{e}_1 + R_Z\hat{e}_s + R_U\hat{M} \neq 0$ with significant probability. Since R_Z is given explicitly, we may simplify this to

$$R_U\hat{M} = \delta \in \mathbb{Z}_p^n,$$

where $\hat{M} \neq 0$ and $\delta := -\hat{e}_1 - R_Z\hat{e}_2$ are chosen by Γ. Writing $R_U = (r_1, \ldots, r_m) \in \{0, 1\}^{m\times m}$ for vectors $r_i \in \{0, 1\}^m$ and $\hat{M} = (\hat{M}_1, \ldots, \hat{M}_m)^\top$, we can write this equivalently as

$$\delta = R_U\hat{M} = \sum_{i=1}^{m} r_i\hat{M}_i = r_j\hat{M}_j + \sum_{i=1, i\neq j}^{m} r_i\hat{M}_i \in \mathbb{Z}_p^n,$$

where $\hat{M}_j \in \{-1, 1\}$ is an arbitrary non-zero component of $\hat{M}$ (note that there must be at least one non-zero component, since $\hat{M} \neq 0 \in \mathbb{Z}_p^n$).

Recall that Γ receives only implicit information about R_U, in form of AR_U. If we can show that there exist two possible choices $r_j, r'_j \in \{-1, 1\}^m$ such that $r_j \neq r'_j$ which are equally likely in the view of Γ, then clearly we must have $\Pr[e \neq 0] \geq 1/2$, because

$$r_j \neq r'_j \quad \Longrightarrow \quad r_j\hat{M}_j + \sum_{i=1, i\neq j}^{m} r_i\hat{M}_i \neq r'_j\hat{M}_j + \sum_{i=1, i\neq j}^{m} r_i\hat{M}_i.$$

Note that $AR_U = (Ar_1|\cdots|Ar_m)$. Thus we need to show that with overwhelming probability there exists $r_j, r'_j \in \{-1, 1\}^m$ with

$$Ar_j = Ar'_j \in \mathbb{Z}_p^n.$$

```
$\mathsf{Gen}(1^k)$
  $(A, T_A) \leftarrow \mathsf{TrapGen}(p, n)$
  $Z, Y \leftarrow \mathbb{Z}_q^{n \times m}$
  $U \leftarrow \mathbb{Z}_q^{n \times \ell}$
  $v \leftarrow \mathbb{Z}_p^n$
  $\kappa \leftarrow \{0,1\}^k$
  $sk := T_{A_1}$
  $pk :=\ (U, A, Z, Y, v, \kappa)$
  return $(sk, pk)$
```

```
$\mathsf{Sig}(sk, M)$
  $u := UM + v$
  For $i \in [l]$ do
    $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M)$
    $G_{t_i} := Z + H(t_i)Y \bmod p$
  $e_i \leftarrow \mathsf{SmpL}(A, T_A, G_{t_i}, u, \gamma)$
  return $(e_i)_{i \in [l]}$
```

```
$\mathsf{Ver}(pk, M, (e_i)_{i \in [l]})$
  if $M \notin \{0,1\}^\ell$ return 0
  For $i \in [l]$ do
    if $e_i \leq 0$ or $\|e_i\| > \gamma\sqrt{2m}$
      return 0
    $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M)$
    $G_{t_i} := Z + H(t_i)Y$
    if $(A|G_{t_i})e_i \neq UM + v$
      return 0
  return 1
```

Figure 9: The EUF-naCMA-secure SIS scheme.

Let $f_A(r) : \{-1,1\}^m \to \mathbb{Z}_p^n$ be the map $r \mapsto Ar$. By the pigeonhole principle there are at most $p^n - 1$ vectors r in $\{-1,1\}^m$ such that the value $f_A(r) = Ar \in \mathbb{Z}_p^n$ has a unique preimage. Since r_j is chosen uniformly random from $\{-1,1\}^m$, the probability that r_j is one of those vectors is at most $(p^n - 1)/2^m \leq 2^{-m+n\log p}$, which is negligible in n if $m \geq 2n \log p$.

Thus, with overwhelming probability there exist at least two vectors r_j, r'_j with $r_j \neq r'_j$ that are consistent with the view of Γ, and thus equally likely, and therefore any algorithm Γ will output $(\hat{e}_1, \hat{e}_2, \hat{M})$ with $\hat{e}_1 + R_Z\hat{e}_s + R_U\hat{M} \neq 0$ with probability at least $1/2 - 2^{-m+n\log p}$. □

6.3 Selection of Parameters

For the scheme to work correctly, we set $n := k$, where k is the security parameter. Furthermore we need to ensure that

- TrapGen can operate, that is, we have $m \geq 6n \log p$,
- That γ is chosen such that the sampling algorithms from Facts 2, 4, and 5 produce the required distribution, and that Fact 7 applies, i.e., that $\gamma \geq m \cdot \omega(\sqrt{m})$,
- that the worst-case to average-case reductions for SIS [26, 15] apply, that is, we have $p \geq \beta \cdot \omega(n \log n)$,
- and that the SIS solutions produced in the reduction are sufficiently short, that is, $\beta \geq \mathbf{O}(m\gamma)$.

6.4 EUF-CMA-Secure Scheme

By applying the generic transformation from Section 3 to our lattice-based EUF-naCMA$_1^*$-secure signature scheme, we obtain EUF-naCMA-secure signatures. Concretely, suppose we use message space $\{0,1\}^\ell$ with $\ell = m$. Then the resulting EUF-naCMA-secure signature scheme has public keys consisting of $4nm + n$ elements of $\mathbb{Z}_p$ plus a key κ for the PRF. Signatures consist of l low-norm vectors in $\mathbb{Z}_p^n$, where $l = \lfloor \log_c(k) \rfloor = \mathbf{O}(\log k)$ is defined as in Section 3. The resulting scheme is depicted in Figure 9.

Unfortunately we are not able to aggregate signatures, like we did for the optimized CDH- and RSA-based constructions, due to the lack of signature aggregation techniques for lattice-based signatures. We leave this as an interesting open problem.

To obtain a fully EUF-CMA-secure signature scheme, it suffices to combine the scheme from Figure 9 with a suitable chameleon hash function, like for instance the SIS-based construction from [8, Section 4.1]. This chameleon hash adds another $2mn$ elements of $\mathbb{Z}_p$ to the public key, plus one additional low-norm vector $e \in \mathbb{Z}_p^m$ to each signature.

Acknowledgements

We thank Ronald Cramer for helpful comments, in particular on the presentation of our results.

References

[1] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.

[2] 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.

[3] Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, Jae Hong Seo, and Christoph Striecks. Practical signatures from standard assumptions. In *EUROCRYPT*, 2013.

[4] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Berlin, Germany.

[5] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, April 2008.

[6] Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 499–517, Paris, France, May 26–28, 2010. Springer, Berlin, Germany.

[7] Zvika Brakerski and Yael Tauman Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive, Report 2010/086, 2010. `http://eprint.iacr.org/`.

[8] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.

[9] Jean-Sébastien Coron. On the exact security of full domain hash. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 229–235, Santa Barbara, CA, USA, August 20–24, 2000. Springer, Berlin, Germany.

[10] R. Cramer and V. Shoup. Signature schemes based on the strong rsa assumption. *ACM Transactions on Information and System Security (TISSEC)*, 3(3):161–185, 2000.

[11] Ronald Cramer and Ivan Damgård. Secure 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.

[12] Ronald Cramer and Ivan Damgård. New generation of secure and practical RSA-based signatures. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 173–185, Santa Barbara, CA, USA, August 18–22, 1996. Springer, Berlin, Germany.

[13] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. In *ACM CCS 99*, pages 46–51, Kent Ridge Digital Labs, Singapore, November 1–4, 1999. ACM Press.

[14] Marc Fischlin. The Cramer-Shoup strong-RSA signature scheme revisited. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 116–129, Miami, USA, January 6–8, 2003. Springer, Berlin, Germany.

[15] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press.

[16] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.

[17] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 21–38, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany.

[18] Dennis Hofheinz, Tibor Jager, and Eike Kiltz. Short signatures from weaker assumptions. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 647–666, Seoul, South Korea, December 4–8, 2011. Springer, Berlin, Germany.

[19] Dennis Hofheinz, Tibor Jager, and Eike Kiltz. Short signatures from weaker assumptions. Cryptology ePrint Archive, Report 2011/296, 2011. `http://eprint.iacr.org/`.

[20] 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.

[21] Susan Hohenberger and Brent Waters. Realizing hash-and-sign signatures under standard assumptions. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 333–350, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.

[22] Marc Joye. An efficient on-line/off-line signature scheme without random oracles. In Matthew K. Franklin, Lucas Chi Kwong Hui, and Duncan S. Wong, editors, *CANS 08*, volume 5339 of *LNCS*, pages 98–107, Hong-Kong, China, December 2–4, 2008. Springer, Berlin, Germany.

[23] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS 2000*, San Diego, California, USA, February 2–4, 2000. The Internet Society.

[24] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979.

[25] Steve Lu, Rafail Ostrovsky, Amit Sahai, Hovav Shacham, and Brent Waters. Sequential aggregate signatures and multisignatures without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 465–485, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Berlin, Germany.

[26] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press.

[27] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*, pages 33–43, Seattle, Washington, USA, May 15–17, 1989. ACM Press.

[28] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.

[29] B. Rosser. Explicit bounds for some functions of prime numbers. *American Journal of Mathematics*, 63(1):211–232, 1941.

[30] Jae Hong Seo. Short signatures from diffie-hellman: Realizing short public key. Cryptology ePrint Archive, Report 2012/480, 2012. `http://eprint.iacr.org/`.

[31] A. Shamir. On the generation of cryptographically strong pseudorandom sequences. *ACM Transactions on Computer Systems (TOCS)*, 1(1):38–44, 1983.

[32] V. Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, 2008.

[33] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany.

[34] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer, Berlin, Germany.