# Practical Signatures From Standard Assumptions

Florian Böhl[1], Dennis Hofheinz[1], Tibor Jager[2], Jessica Koch[1],
Jae Hong Seo[3], and Christoph Striecks[1]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
[2] Ruhr-Universität Bochum, Bochum, Germany
[3] Myongji University, Yongin, Republic of Korea

**Abstract.** We put forward new techniques for designing signature schemes. As a result, we present practical signature schemes based on the CDH, the RSA, and the SIS assumptions. Our schemes compare favorably with existing schemes based on these assumptions.

Our core idea is the use of *tag-based* signatures. Concretely, each signatures contains a tag which is uniformly chosen from a suitable tag set. Intuitively, the tag provides a way to embed instances of computational problems. Indeed, carefully choosing these tag spaces provides new ways to partition the set of possible message-tag pairs into "signable" and "unsignable" pairs. In our security proof, we will thus be able to sign all adversarially requested messages, and at the same time use an adversarially generated forgery with suitably large probability.

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

## 1 Introduction

**On the difficulty of constructing digital signature schemes.** From a purely theoretical point of view, digital signatures turn out to be a weaker primitive than public-key encryption (PKE): digital signature schemes are equivalent to one-way functions [29], while PKE appears to be a stronger primitive [24]. However, somewhat surprisingly, it seems much harder to construct *practical* signature schemes than PKE schemes. For instance, there exist practical and even chosen-ciphertext secure PKE schemes from a variety of assumptions (e.g., DDH [12], CDH [9], DCR [14], factoring [18], or LPN [28]), while it seems much harder to construct practical signature schemes from any of the above assumptions.[4] Indeed, the most efficient known schemes (e.g., [13, 15, 25, 5, 19]) are based on what [20] call "strong" assumptions.[5] Intuitively, to contradict a strong assumption, it suffices to solve one out of many possible problem instances given by the challenge. For instance, the strong RSA assumption demands that finding *any* $e \geq 2$ and $C^{1/e} \bmod N$ when given $N$ and $C \in \mathbb{Z}_N$ is hard.

[4] We ignore here schemes based on random oracles (e.g., full-domain hash [3]), since these come only with heuristic proofs.

[5] There are also practical schemes based on standard, non-strong assumptions, e.g., [6, 33, 22, 20]; these however suffer from large keys or signatures, or from a comparatively inefficient signing process. A notable exception are dual systems and dual form signatures [32, 17], which are however based on decisional assumptions like DLIN.

We believe that this reliance on strong assumptions is very natural for a signature scheme. 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. This gives $A$ much more freedom (by choosing signature *and* message freely) than, e.g., an adversary in an encryption security experiment. Thus, 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).[6]

**Our approach: tag-based signatures.** In this work, we explore *tag-based signature schemes* as a means to enable security reductions to standard computational assumptions. In a tag-based signature scheme, each signature carries a tag $t$ that can be chosen freely during signature time. Intuitively, the tag further parameterizes the problem instance we want to let $A$ solve during a security reduction. The benefit of this additional parameterization becomes apparent when one considers tags from a small domain: if there are only few (i.e., polynomially many) tags, we could try to guess the tag $t^*$ used in $A$'s forgery in advance. Our security reduction could then set up things such that precisely signatures with tags $t \neq t^*$ can be generated, and any signature with tag $t^*$ can be used to solve an underlying computational problem. (For now, let us assume that $A$ never reuses a tag from a previously signed message in his forgery.)

**Showcase: compact CDH-based signatures.** To showcase our ideas, we first consider a tag-based signature scheme in pairing-friendly groups. The scheme itself can be seen as a variant of the stateful signature scheme of Hohenberger and Waters [23], with tags (chosen from a suitably-sized tag space) in place of states. At this point, our work forks up into two directions: first, we show that this scheme achieves a *bounded* form of security in which an upper bound on the number of adversarial signature queries is known prior to setting the scheme's parameters. This yields an extremely compact and efficient scheme for reasonable security parameters. Next, we show how to achieve full security by employing a different, somewhat more generic proof strategy. This yields an *asymptotically* more efficient scheme, at the cost of a qualitatively worse security reduction. (With "qualitatively worse", we mean that the reduction *loss* depends – in a polynomial way – on the adversary's runtime and success.) In both cases, we prove security under the computational Diffie-Hellman (CDH) assumption.

**More on the bounded security of our scheme.** First, consider an adversary $A$ that makes only a fixed, a-priori bounded number $q$ of signing queries. Our tag space will then consist of *vectors* $\overrightarrow{t}$ over a polynomial domain whose size depends on $q$. In the security reduction, we will guess a suitable vector prefix $\overrightarrow{t}^{(i)}$ of the challenge tag that is different from all tag prefixes that arise during the signature generation for $A$. This allows us to embed a CDH challenge into the prefix $\overrightarrow{t}^{(i)}$ during the security proof.

[6] There are more clever ways of embedding a computational problem into a signature scheme (e.g., partitioning [11, 33]). 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.

On a technical level, this strategy opens the door to an interesting tradeoff between public key and signature size. Namely, using a partitioning argument, we can allow a very limited number of tag-prefix-collisions (in the sense that tags with the prefix $\overrightarrow{t}^{(i)}$ used in the forgery occur in the signatures generated for $A$). This yields smaller public keys, at the cost of additional group elements in the signatures. For reasonable security parameters (and assuming $q \leq 2^{30}$ signature queries), we thus obtain a scheme with $4$ and $15$ group elements per signature, resp. public key.

**Confined guessing.** To prove our scheme fully secure, we rely on a new technique we call "confined guessing". Concretely, we view our scheme as the combination of several instances of a mildly secure tag-based scheme, each instance with a different tag size. Here, mild security means that an adversary has to commit in advance on the tag $t^*$ of his forgery. The basic version of our CDH-based scheme can be proven mildly secure, since the CDH challenge can be embedded precisely in signatures with tag $t^*$. (In particular for small tag sets, it may be necessary to generate a *constant* number of signatures with tag $t^*$ for $A$. We can solve this problem using a partitioning technique, which – since the number of required $t^*$-signatures is small – can be very efficient.)

A signature in our fully secure scheme consists of $\log(\lambda)$ signatures $(\sigma_i)_{i=1}^{\log(\lambda)}$ of a mildly secure scheme. (In the CDH case, these signatures can be aggregated.) The $i$-th signature component $\sigma_i$ 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 $B$. 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. This leads to a very compact scheme (e.g., in the CDH case, with $O(1)$ and $O(\log(\lambda))$ group elements per signature, resp. public key). However, the *loss* in the security reduction depends (polynomially) on an adversary's success and runtime.

**Other applications.** We also show how to generalize our confined guessing paradigm to other computational settings. In particular, we construct mildly secure schemes from the RSA and SIS assumptions. Combining this with our generic transformation, this gives compact and very efficient new fully secure signature schemes.

**Efficiency comparison.** The most efficient previous CDH-based signature scheme [33] has signatures and public keys of size $O(\lambda)$, resp. $O(1)$ group elements. Our CDH-based scheme also has constant-sized signatures, and more compact public keys. Concretely, we can get public keys of $O(\sqrt{\lambda/\log(\lambda)})$ group elements when only aiming at bounded security (see Table 1 for exact figures). Besides, we can get public keys of $O(\log(\lambda))$ group elements at the price of a worse security reduction. Our RSA-based scheme has similar key and signature sizes as existing RSA-based schemes [22, 20], but requires significantly fewer (i.e., only $O(\log(\lambda))$ instead of $O(\lambda)$, resp. $O(\lambda/\log(\lambda))$

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 [7]. Concretely, our scheme has larger (by a factor of $\log(\lambda)$) signatures and a worse security reduction, but significantly smaller (by a factor of $\lambda/\log(\lambda)$) public keys.

**Note on the history of this paper.** This paper is the result of a merge of two papers submitted to Eurocrypt. Both submissions contained essentially the same CDH-based scheme. One submission, by Seo, contained its bounded security analysis. The other, by Böhl, Hofheinz, Jager, Koch, and Striecks, contained the confined guessing strategy. In this merged paper, the results of Seo, resp. BHJKS, are contained in Section 4, resp. 5. During the merge, Seo acted as corresponding author.

**Acknowledgement.** Jae Hong Seo thanks members of Shin-Akarui-Angou-Benkyou-Kai for their helpful comments. Part of his work was done while he was in NICT, Japan and it was made public through [1]. Böhl was supported by MWK grant "MoSeS". Koch was supported by BMBF project "KASTEL". Hofheinz and Jager were supported by DFG grant GZ HO 4534/2-1. Part of this work was done while Jager was with KIT.

## 2 Preliminaries

**Notation.** For $n \in \mathbb{R}$, let $[n] := \{1, \ldots, \lfloor n \rfloor\}$. We write $[a, b]$ to denote the set of integers $\{a, \ldots, b\}$. Throughout the paper, $\lambda \in \mathbb{N}$ 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 $\lambda$, then $A$ is called probabilistic polynomial-time (PPT). A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if it vanishes faster than the inverse of any polynomial (i.e., if $\forall c \exists \lambda_0 \forall \lambda \geq \lambda_0 : |f(\lambda)| \leq 1/\lambda^c$). On the other hand, $f$ is significant if it dominates the inverse of some polynomial (i.e., if $\exists c, \lambda_0 \forall \lambda \geq \lambda_0 : f(\lambda) \geq 1/\lambda^c$).

**Signature schemes.** A signature scheme $\mathsf{SIG}$ consists of three PPT algorithms ($\mathsf{Gen}$, $\mathsf{Sig}$, $\mathsf{Ver}$). The key generation algorithm $\mathsf{Gen}(1^\lambda)$ outputs a public key $pk$ and a secret key $sk$. The signing algorithm $\mathsf{Sig}(sk, M)$, given the secret key $sk$ and a message $M$, outputs a signature $\sigma$. Given the public key $pk$, a message $M$, and a signature $\sigma$, $\mathsf{Ver}(pk, M, \sigma)$ outputs a verdict $b \in \{0, 1\}$. For correctness, we require for any $\lambda \in \mathbb{N}$, all $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, all $M$, and all $\sigma \leftarrow \mathsf{Sig}(sk, M)$ that $\mathsf{Ver}(pk, M, \sigma) = 1$.

**EUF-(na)CMA security.** A signature scheme $\mathsf{SIG}$ is existential unforgeable under adaptive chosen-message attacks (EUF-CMA) iff for any PPT forger $F$ in the following experiment the probability to win is negligible. $F$ receives a public key $pk$ generated as $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, and has access to a signing oracle $\mathsf{Sig}(sk, \cdot)$. $F$ wins if it outputs a valid signature for a message $M$ such that it has never queried $\mathsf{Sig}(sk, M)$. In the non-adaptive (EUF-naCMA) case the adversary is forced to output messages $M_1, \ldots, M_q$ it wants to see signed before obtaining the public key $pk$. $F$ wins if it outputs a valid signature for a message $M \neq M_j \ \forall j \in [q]$. We define $\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},F}(\lambda)$ and $\mathsf{Adv}^{\mathsf{euf\text{-}nacma}}_{\mathsf{SIG},F}(\lambda)$ to be $F$'s winning probability in the adaptive, resp. non-adaptive case.

**EUF-$q$-(na)CMA security.** In addition to the previous notions, we define two weaker security notions for signatures. The notions existential unforgeability with respect to $q$-

bounded adaptive chosen-message-attacks (EUF-$q$-CMA), resp. non-adaptive chosen-message-attacks (EUF-$q$-naCMA) are exacly the same as the EUF-CMA, resp. EUF-naCMA security notions above, except that the adversary is restricted to at most $q$ signature queries. We define $\mathsf{Adv}^{\mathsf{euf\text{-}q\text{-}cma}}_{\mathsf{SIG},F}(\lambda)$ and $\mathsf{Adv}^{\mathsf{euf\text{-}q\text{-}nacma}}_{\mathsf{SIG},F}(\lambda)$ to be $F$'s winning probability in the adaptive, resp. non-adaptive case. Concretely, if for any PPT algorithm $F$, which issues at most (polynomial) $q$ signing queries and runs in time $T$, $\mathsf{Adv}^{\mathsf{euf\text{-}q\text{-}cma}}_{\mathsf{SIG},F}(\lambda) < \varepsilon$ and $\mathsf{Adv}^{\mathsf{euf\text{-}q\text{-}nacma}}_{\mathsf{SIG},F}(\lambda) < \varepsilon$, then we say that $\mathsf{SIG}$ is $(q, \varepsilon, T)$- EUF-CMA secure, resp. $(q, \varepsilon, T)$-EUF-naCMA secure.

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

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

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 $\mathsf{Samp}$ we can generically construct a PRF with range $\mathcal{S}$ from a PRF $\mathsf{PRF}^{\{0,1\}^{\lambda}}$ by using the output of $\mathsf{PRF}^{\{0,1\}^{\lambda}}_{\kappa}$ as random coins for $\mathsf{Samp}$. (We can assume without loss of generality that $\mathsf{Samp}$ requires only $\lambda$ bits of randomness.) 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 $\mathsf{PRF}^{\{0,1\}^{\lambda}}$.

**Chameleon hashing.** A chameleon hash scheme $\mathsf{CHS}$ consists of two PPT algorithms $(\mathsf{CHGen}, \mathsf{CHTrapColl})$. $\mathsf{CHGen}(1^{\lambda})$ outputs a tuple $(\mathsf{CH}, \tau)$ where $\mathsf{CH}$ is the description of an efficiently computable function $\mathsf{CH} : \mathcal{M} \times \mathcal{R} \to \mathcal{N}$ which maps a message $M$ and randomness $r$ to a hash value $\mathsf{CH}(M, r)$. The trapdoor $\tau$ 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 $\mathsf{CH}$ and $M'$. We say that $\mathsf{CH}$ is collision-resistant iff for $(\mathsf{CH}, \tau) \leftarrow \mathsf{CHGen}(1^{\lambda})$ and any PPT adversary $C$, which receives as input $\mathsf{CH}$, the probability $\mathsf{Adv}^{\mathsf{cr}}_{\mathsf{CH},C}(\lambda)$ to find $(M, r) \neq (M', r')$ with $\mathsf{CH}(M, r) = \mathsf{CH}(M', r')$ is negligible in $\lambda$.

**Generic transformation from non-adaptive to adaptive secure signatures.** There is a well known generic transformation from EUF-naCMA secure signatures to EUF-CMA secure signatures which was used in many previously proposed signature schemes (e.g., [26, 31, 4, 23, 22]). Analogously, we can construct a generic transformation from EUF-$q$-naCMA secure signatures to EUF-$q$-CMA secure signatures.

**Lemma 1.** *If* $\mathsf{SIG}$ *is* $(q, \varepsilon, T)$*-EUF-naCMA secure signature scheme and* $\mathsf{CHS}$ *is a chameleon hash scheme, then there is a generic transformation taking* $\mathsf{SIG}$ *and* $\mathsf{CHS}$ *as input and outputting* $(q, 2(\varepsilon + \varepsilon_{ch}), T')$*-EUF-CMA secure signature scheme, where the chameleon hash function satisfies* $(\varepsilon_{ch}, T_{ch})$*-collision resistance and* $T' \approx T \approx T_{ch}$.

We provide the details of the generic transformation in [30].

## 3 Our CDH-based Signature Scheme

**Overview.** Our starting point is an interpretation of the stateful signature scheme of Hohenberger and Waters [23] as a tag-based scheme. In this section, we will only de-

scribe the scheme (and a natural generalization); the next two sections will present two surprisingly different analyses of this scheme. But first, we start with a few preparatory definitions.

**Bilinear Groups.** We say that $\mathcal{G}$ is a bilinear group generator if on inputting the security parameter $\lambda$, it outputs a tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e)$, where $p$ is a $(2\lambda+1)$-bit prime, $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_t$ are finite abelian groups of order $p$, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$ is a non-degenerate bilinear map, that is, (bilinearity) for all $a, b \in \mathbb{Z}_p$ and $g \in \mathbb{G}_1$, $g' \in \mathbb{G}_2$, $e(g^a, g'^b) = e(g, g')^{ab}$ and (non-degeneracy) for generators $g \in \mathbb{G}_1$ and $g' \in \mathbb{G}_2$, $e(g, g') \neq 1$. If $\mathbb{G}_1 = \mathbb{G}_2$, then we use a notation $\mathbb{G}$ to denote $\mathbb{G}_1 = \mathbb{G}_2$ and we say that $e$ is a type-1 pairing. If $\mathbb{G}_1 \neq \mathbb{G}_2$ but there is an efficiently computable homomorphism $\phi : \mathbb{G}_2 \to \mathbb{G}_1$, then we say that $e$ is a type-2 pairing. Otherwise (that is, $\mathbb{G}_1 \neq \mathbb{G}_2$ and there are no efficiently computable homomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$), we say that $e$ is a type-3 pairing.

**CDH assumption.** Let $\mathcal{G}$ be a bilinear group generator. We say that $\mathcal{G}$ satisfies the $(\epsilon_{dh}, T_{dh})$-DH assumption if for any $T_{dh}$- time probabilistic algorithm $B$ the following advantage $\mathsf{Adv}^{dh}_{\mathcal{G},B}$ is less than $\epsilon_{dh}$:

$$\mathsf{Adv}^{dh}_{\mathcal{G},B} = \Pr\Big[B(p, \mathbb{G}, \mathbb{G}_t, e, g, g^a, g^b) \to g^{ab} : \mathcal{G}(\lambda) \to (p, \mathbb{G}, \mathbb{G}_t, e), a, b \leftarrow \mathbb{Z}_p, g \leftarrow \mathbb{G}\Big].$$

**Tag-based signature schemes.** Our basic scheme will be tag-based; that means that signature and verification take an additional *tag* as input. More formally, a tag-based signature scheme $\mathsf{SIG}_{\mathsf{t}} = (\mathsf{Gen}_{\mathsf{t}}, \mathsf{Sig}_{\mathsf{t}}, \mathsf{Ver}_{\mathsf{t}})$ with message space $\mathcal{M}_\lambda$ and tag space $\mathcal{T} = \mathcal{T}_\lambda$ consists of three PPT algorithms. Key generation $(pk, sk) \leftarrow \mathsf{Gen}_{\mathsf{t}}(1^\lambda)$ takes as input a security parameter and outputs a key pair $(pk, sk)$. Signing $\sigma \leftarrow \mathsf{Sig}_{\mathsf{t}}(sk, M, t)$ computes a signature $\sigma$ on input a secret key $sk$, message $M$, and tag $t$. Verification $\mathsf{Ver}_{\mathsf{t}}(pk, M, \sigma, t) \in \{0, 1\}$ takes a public key $pk$, message $M$, signature $\sigma$, and a tag $t$, and outputs a bit. For correctness, we require for any $\lambda \in \mathbb{N}$, all $(pk, sk) \leftarrow \mathsf{Gen}_{\mathsf{t}}(1^\lambda)$, all $M \in \mathcal{M}_\lambda$, all $t \in \mathcal{T}$, and all $\sigma \leftarrow \mathsf{Sig}_{\mathsf{t}}(sk, M, t)$ that $\mathsf{Ver}_{\mathsf{t}}(pk, M, \sigma, t) = 1$.

**The basic (tag-based) scheme.** The signature scheme $\mathsf{SIG}^{\mathsf{CDH}}$ from Figure 1 is derived from the stateful CDH-based scheme of [23], but with states interpreted as tags, and with two additional modifications. First, we substitute the implicit chameleon hash function $u^M v^r$ used in [23] with a product $\mathbf{u}^M = \prod_{i=0}^{m} u_i^{M^i}$. (The parameter $m$ will be fixed later.) Second, we omit the $w^{\lceil \log(t) \rceil}$-factor in the "Boneh-Boyen hash function", which simplifies this part to $(z^t h)^s$.

**The generalized (non-tag-based) scheme.** We also provide a natural generalization of $\mathsf{SIG}^{\mathsf{CDH}}$, which we call $\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{gen}}$ (see Figure 2). Compared to $\mathsf{SIG}^{\mathsf{CDH}}$, $\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{gen}}$ first hashes the message to be signed (using a chameleon hash with images in $\mathbb{Z}_p$); besides, $\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{gen}}$ uses a pseudorandom function to derive $l$ tags $t_i$ that are incorporated into the mentioned "Boneh-Boyen hash function". The number $l$ of tags and the respective sets $\mathcal{T}_i$ from which the $t_i$ are chosen which will be defined in our respective analyses.

$\mathsf{Gen}_{\mathsf{t}}(1^\lambda)$
$(p, \mathbb{G}, \mathbb{G}_t, e) \leftarrow \mathcal{G}(\lambda)$
$\alpha \leftarrow \mathbb{Z}_p$
$g, u_0, \ldots, u_m, z, h \leftarrow \mathbb{G}$
$sk := \alpha$
$pk :=$
$(g, g^\alpha, u_0, \ldots, u_m, z, h)$
return $(pk, sk)$

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

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

**Fig. 1.** The modified Hohenberger-Waters CDH-based signature scheme $\mathsf{SIG}^{\mathsf{CDH}}$ [23].

$\mathsf{Gen}(1^\lambda)$
$(p, \mathbb{G}, \mathbb{G}_t, e) \leftarrow \mathcal{G}(\lambda)$
$(\mathsf{CH}, \tau) \leftarrow \mathsf{CHGen}(1^\lambda)$
$\alpha \leftarrow \mathbb{Z}_p$
$g, h, u_0, \ldots, u_m,$
$z_1, \ldots, z_l \leftarrow \mathbb{G}$
$\kappa \leftarrow \{0,1\}^\lambda$
$sk := (g, \alpha, \mathsf{CH})$
$pk := (g, g^\alpha, (u_j)_{j=0}^m,$
$(z_i)_{i=1}^l, h, \mathsf{CH}, \kappa)$
return $(pk, sk)$

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

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

**Fig. 2.** The generalized CDH-based signature scheme $\mathsf{SIG}_{\mathsf{gen}}^{\mathsf{CDH}}$.

## 4 Bounded CMA Security

In this section, we first analyze the security of the basic tag-based signature scheme so that it is a EUF-$q$-naCMA secure signature scheme with somewhat short public key. Then, we prove that the generalized tag-based signature scheme is a EUF-$q$-naCMA secure signature scheme with short public key. The proposed signature scheme is for fixed length messages, but we note that we can easily modify it for arbitrary length messages by using collision resistant hash functions; first, compute a hash value of a long message, and then use it as a message for the signature scheme.

### 4.1 Combining Two Techniques: 'Somewhat' Short Public Key.

We begin with exploring two techniques for obtaining short signatures in the standard model. In the simulation of the EUF-$q$-naCMA model, the simulator should give a set of signatures on messages queried by the adversary, but the simulator should not be able to create signatures on all messages other than those queried by the adversary. If the simulator can create signatures on all messages, then the simulator does not need help from the adversary to obtain the forgery since the simulator can sign on all messages himself; hence, we cannot extract the solution of the DH problem from the output of the adversary. We can use programmable hash functions [19] to allow the simulator to produce only signatures on messages queried by the adversary. In particular, we use weak programmable hash functions [20] to construct EUF-$q$-naCMA secure short

signatures. For a $2\lambda$-bit message $M$, we consider $M$ as an element of $\mathbb{Z}_p$. $(\prod_{i=0}^{m} u_i^{M^i})$ is a weak programmable hash function on input $M$ that, in the EUF-$q$-naCMA model, allows the simulator to sign on at most $m$ messages, which are given by the adversary before generating the public key.[7] Furthermore, we can construct a simulator that extracts the solution of $g^{ab}$ from the forgery by imbedding $g^a$ in $u_i$ and setting $g$ and $g^\alpha$ by $g$ and $g^b$, respectively.

There is the other technique that obtains short signatures with short public key by maintaining the index counter in the signer side [23]. The idea of this technique is first to restrict the adversary to attack one of the polynomially many indexes and then uses the technique for selectively-secure signatures such as that used in the Boneh-Boyen signature scheme [5]. We can combine this technique with programmable hash functions. Since our aim is a stateless signature scheme, we should modify this technique so that the signer does not maintain the current index but randomly chooses it from some fixed set for each signature. Then, we obtain a short signature with somewhat short public key, which is our basic tag-based scheme in Figure 1, where $t$ is uniformly chosen from $t^* = [1, Q]$ for another parameter $Q \geq q$. For the basic scheme, we set $Q$ to be polynomial in $\lambda$. The strategy of the simulation in the EUF-$q$-naCMA model is as follows: The simulator guesses $t^*$, the tag of the forgery, (with non-negligible $\frac{1}{Q}$ probability) and uses the technique for the selectively-secure signature scheme of the Boneh-Boyen signatures. For each signature, the tag is randomly chosen so that there may exist several signatures containing the same tag as $t^*$ among the resulting signatures of singing queries. Under normal circumstances, the simulator cannot produce signatures with tag $t^*$ (since we use technique for selectively-secure scheme). We can resolve this by using the weak programmable hash functions. If we uniformly choose a tag from $[1, Q]$ at most $q$ times for polynomial $Q \geq q$, there are at most $\Theta(\frac{\lambda}{\log \lambda})$ same tags as the tag of the forgery with overwhelming probability. Therefore, we can set $m = \Theta(\frac{\lambda}{\log \lambda})$ and the simulator can create $m$ signatures, which has the same tag as that of the forgery. Since our main scheme in the next subsection is a generalization of the scheme in Figure 1, we omit the detailed security analysis of the scheme in Figure 1.

*Remark.* We used the combination of the two techniques in this section for signature schemes based on the DH assumption. There is similar approach for signature schemes based on the RSA assumption and $q$-DH assumption [20]. Note that our original contribution is explained in Section 4.2.

### 4.2 Asymmetric Trade: Realizing Short Public Key

Let $Q$, $m$, and $l$ be functions in $\lambda$. For readers who want to see the specific parameters a little early, we give an example parameter below. We will explain about selecting parameters in the last part of this subsection.

*Example Parameter 1.* $Q = 2^3 q$, $m = \left\lceil \sqrt{\frac{\lambda}{\log \lambda}} \right\rceil$, and $l = \left\lceil \sqrt{\frac{\lambda}{\log \lambda}} \right\rceil$.

We first describe a signature scheme which is EUF-q-naCMA secure under the DH assumption in Figure 3. By applying standard techniques using the chameleon

[7] $M^i$ is not the $i$-th bit of $M$, but the $i$ times product of $M$.

hashes and the pseudo-random functions, we can obtain the generalized (non-tag-based) scheme in Figure 2.

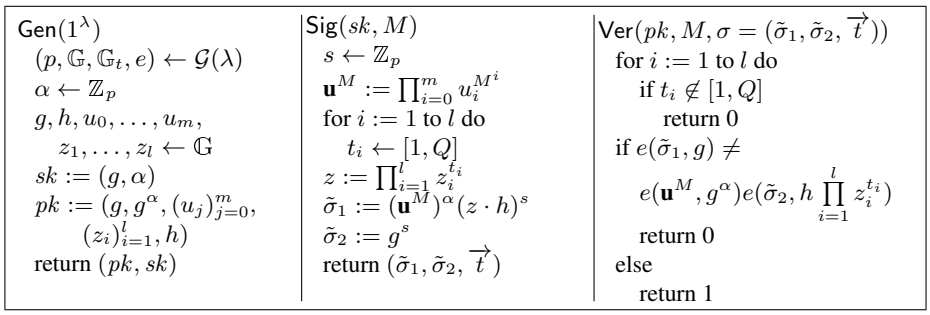


**Fig. 3.** EUF-$q$-naCMA secure signature with short public key.

For each signature $\sigma = (\tilde{\sigma}_1, \tilde{\sigma}_2, \overrightarrow{t})$, we call $\overrightarrow{t}$ tag vector. In contrast to the basic tag-based scheme, we use a vector $\overrightarrow{t}$ instead of an integer $t_1$ in signatures. Roughly speaking, our analysis shows that the signature scheme in Figure 3 with $\mathcal{T}_i := [1, Q]$ for $i \in \{1, \ldots, l\}$ satisfies non-adaptive unforgeability (against bounded CMA) when $ml = \Omega(\frac{\lambda}{\log \lambda})$ (this result contains the signatures with somewhat short public key in Figure 1). In addition (roughly speaking again), since the public key size is $\Theta(m + l)$ group elements, we can attain the minimal public key size when $m$ and $l$ are nearly equal. On the other hand, the size of signatures will increase when the parameter $l$ increases. However, each $t_i$ is a $\log Q$-bit integer, and so $\overrightarrow{t}$ is asymptotically much shorter than $\Theta(\lambda)$-bit (if we set $Q$ as a polynomial in $\lambda$). This is an *asymmetric trade* between the public key and tag vectors. When we apply the example parameter 1, the signature size will be bounded by two group and a field element, that is, the signature size is $\Theta(\lambda)$ bits. We give precise analysis of the efficiency of the proposed signature scheme in Section 4.2.

Our construction of the short signatures with short public key in Figure 3 is a simple generalization of the basic tag-based scheme (short signatures with somewhat short public key) in Figure 1. However, the analysis of the security in the EUF-$q$-naCMA model is more challenging than the construct itself. The basic strategy of the simulator in the EUF-$q$-naCMA model of the signature scheme in Figure 1 is guessing the tag $t^*$ of the forgery and then using the programmability of the weak programmable hash function $(\prod_{i=0}^m u_i^{M^i})$ to sign for the signature with the same tag. We cannot naively apply this proof strategy to the generalized construction. To obtain short public key, we should set $l$ sufficiently large (but not too much). However, if $l$ is large, then the simulator cannot guess the tag vector of the forgery, $t^* \in [1, Q]^l$, with non-negligible probability. That is, we would fail to construct a polynomial-time reduction. We developed a proof technique to resolve this problem.

**Our proof strategy** We now explain our proof strategy for polynomial-time reduction from solving the DH problem to the breaking the non-adaptive unforgeability of the

proposed signature scheme. In particular, we explain the method to guess the tag vector $t^*$ of the forgery with non-negligible probability. In fact, we cannot guess all the bits of $t^*$, but only part of $t^*$ with non-negligible probability. This is sufficient for our proof strategy.

We begin with defining notations for efficient explanation. Let $T$ and $T^i$ be sets $[1, Q]$ and $[1, Q]^i$ ($i$ times canonical product set), respectively. For $j \in [1, q]$, let $t_j \in T^l$ be the tag vector (randomly chosen by the simulator) of the signature on the $j$th message (queried by the adversary). Let $\overrightarrow{t}^* = (t_1^*, \ldots, t_l^*) \in T^l$ be the tag vector of the forgery output by the adversary. For $\overrightarrow{t} \in T^l$ and $i \leq l$, let $\overrightarrow{t}^{(i)} \in T^i$ be the first $i$ entries of $\overrightarrow{t}$ (e.g., $\overrightarrow{t} = (t_1, \ldots, t_l)$ and $\overrightarrow{t}^{(i)} = (t_1, \ldots, t_i)$). We separate the adversaries into several types according to the relations between $\overrightarrow{t}^*$ and $\{\overrightarrow{t}_i\}_{i\in[1,q]}$. To this end, for fixed $\{\overrightarrow{t}_i\}_{i\in[1,q]}$, we first define the set $T_i$ as

$$\{\hat{t} \in T^i \mid \exists \text{ at least } (m+1) \text{ distinct } j_1, \ldots, j_{m+1} \in [1, q] \text{ such that } \hat{t} = \overrightarrow{t}^{(i)}_{j_1} = \ldots = \overrightarrow{t}^{(i)}_{j_{m+1}}\}.$$

Let us consider an example to help the readers understand the definition of $T_i$.

*Example.* Suppose that

$$\begin{array}{ll} \overrightarrow{t}^{(i)}_1 = \ldots = \overrightarrow{t}^{(i)}_{m+2} \neq \overrightarrow{t}^{(i)}_j & \text{for } j \in [m+3, q], \\ \overrightarrow{t}^{(i+1)}_1 = \ldots = \overrightarrow{t}^{(i+1)}_{m+1} \neq \overrightarrow{t}^{(i+1)}_j & \text{for } j \in [m+2, q], \end{array}$$

and $\overrightarrow{t}^{(i)}_{m+3}, \ldots, \overrightarrow{t}^{(i)}_q$ are distinct. Then,

$$\begin{cases} \overrightarrow{t}^{(i)}_j \in \mathcal{T}_i \text{ for } j \in [1, m+2] \\ \overrightarrow{t}^{(i)}_j \notin \mathcal{T}_i \text{ for } j \in [m+3, q], \end{cases}, \quad \begin{cases} \overrightarrow{t}^{(i+1)}_j \in \mathcal{T}_{i+1} \text{ for } j \in [1, m+1] \\ \overrightarrow{t}^{(i+1)}_j \notin \mathcal{T}_{i+1} \text{ for } j \in [m+2, q], \end{cases}$$

and $|T_i| = |T_{i+1}| = 1$. □

We can easily see that $|T_{i+1}| \leq |T_i|$. Let $n$ be the largest integer in $[1, l]$ such that $T_n \neq \emptyset$. If we choose $m$, $l$, and $Q$ appropriately, we then obtain the following two properties with overwhelming probability, where the probability is taken over the choice of $\{\overrightarrow{t}_i\}_{i\in[1,q]}$.

1. $|\mathcal{T}_1| < \lambda$
2. $n < l$ (equivalently $T_l = \emptyset$, that is, $|T_l| < 1$)

When $Q \geq q$, the following lemma implies the above two properties. (e.g., we obtain the above properties when we apply the example parameter 1 to Lemma 2.)

**Lemma 2.** $\Pr_{\overrightarrow{t}_1, \ldots, \overrightarrow{t}_q \leftarrow T_l}[|T_i| \geq j] < (\frac{q^{m+1}}{(m+1)!Q^{im}})^j$.

*Proof.* Let $F$ be the set of all functions from $[1, q]$ to $S^i$. For $\overrightarrow{y} \in S^i$ and $f \in F$, let $|f^{-1}(\overrightarrow{y})|$ be the number of the distinct pre-images of $\overrightarrow{y}$. Let $T_f$ be the set of all $\overrightarrow{y} \in Im(f)$ such that $|f^{-1}(\overrightarrow{y})| \geq m+1$, where $Im(f)$ means the set of all images of $f$. Then, we can consider $\Pr_{\overrightarrow{s}_1, \ldots, \overrightarrow{s}_q \leftarrow S^k}[|S_i| \geq j]$ as

$$\Pr_{f \leftarrow F}[|T_f| \geq j].$$

To compute $\Pr_{f \leftarrow F}[|T_f| \geq j]$, we count all functions $f$ such that $|T_f| \geq j$, then divide the result by $|S^i|^q$ (the number of all elements in $F$). In fact, we count the number of $f$ such that $|T_f| \geq j$, allowing duplications, so that we compute the upper bound of $\Pr_{f \leftarrow F}[|T_f| \geq j]$. To define an $f$, we choose $j$ distinct subsets $A_1, \ldots, A_j$ of size $m+1$ from $[1, q]$ and $j$ distinct vectors $\overrightarrow{y}_1, \ldots, \overrightarrow{y}_j$ from $S^i$, and then set $f(a) = \overrightarrow{y}_t$ for all $a \in A_t$ and $t \in [1, j]$. For other integers $a \in [1, q] \setminus (A_1 \cup \ldots \cup A_j)$, we arbitrarily define $f(a)$. This way of defining a function covers all $f$ such that $|T_f| \geq j$. We count all $f$ that are defined as above. Then, the number of such $f$ is bounded by

$$\Big(\prod_{t=0}^{j-1} \binom{q - t(m+1)}{m+1} \cdot (|S^i| - t)\Big) \cdot (|S^i|)^{(q-j(m+1))},$$

where the notation $\binom{\cdot}{\cdot}$ denotes the binomial coefficient.

Therefore, we can obtain the desired result as follows:

$$\begin{aligned}
\Pr_{\overrightarrow{s}_1, \ldots, \overrightarrow{s}_q \leftarrow S^k}[|S_i| \geq j] &= \Pr_{f \leftarrow F}[|T_f| \geq j] \\
&< \frac{\Big(\prod_{t=0}^{j-1} \binom{q-t(m+1)}{m+1} \cdot (Q^i - t)\Big) \cdot (Q^i)^{(q-j(m+1))}}{|S^i|^q} \\
&< \frac{\left(\frac{q^{m+1}}{(m+1)!}\right)^j Q^{ij + i(q-j(m+1))}}{Q^{iq}} \\
&= \left(\frac{q^{m+1}}{(m+1)! Q^{im}}\right)^j.
\end{aligned}$$

□

For now, let us assume that we have $m$, $l$, and $Q$ such that the above two properties hold. We separate the types of adversaries according to $\overrightarrow{t}^*$ as follows.

$$\begin{aligned}
\text{Type-1}: &\ \overrightarrow{t}^{*(1)} \notin T_1. \\
\text{Type-2}: &\ \overrightarrow{t}^{*(1)} \in T_1, \text{ and } \overrightarrow{t}^{*(2)} \notin T_2. \\
&\ \vdots \\
\text{Type-}i: &\ \overrightarrow{t}^{*(i-1)} \in T_{i-1}, \text{ and } \overrightarrow{t}^{*(i)} \notin T_i. \\
&\ \vdots \\
\text{Type-}n: &\ \overrightarrow{t}^{*(n-1)} \in T_{n-1}, \text{ and } \overrightarrow{t}^{*(n)} \notin T_n. \\
\text{Type-}(n+1): &\ \overrightarrow{t}^{*(n)} \in T_n.
\end{aligned}$$

Here, $\overrightarrow{t}^{*(i-1)} \in T_{i-1}$ implies that $\overrightarrow{t}^{*(j)} \in T_j$ for all $j \in [1, i-1]$. Therefore, we can see that the above $n+1$ types of adversaries are pairwise disjoint and cover all possible adversaries. For the type-$i$ adversary, the simulator can guess $\overrightarrow{t}^{*(i)}$ with probability $\frac{1}{|\mathcal{T}_{i-1}| \cdot |T|}$; it guesses $\overrightarrow{t}^{*(i-1)}$ with $\frac{1}{|T_{i-1}|}$ and $t_i^*$ with $\frac{1}{|T|}$ (we use the second property for the case $i = n+1$). Since the simulator can guess the type of the adversary with probability $\frac{1}{l}$, it can guess the tag vector of the forgery with at least probability $\frac{1}{l\lambda Q}$ (we use the first property for the inequality $|T_{i-1}| \leq |T_1| < \lambda$).

The other parts of the proof strategy are similar to the strategy for the short signatures with somewhat short public key in Figure 1 as we mentioned in Section 4.1; (1) guess $\overrightarrow{t}^{*(i)}$, (2) use the proof technique for the reduction from solving the DH problem to breaking the selectively secure signatures, and (3) generate for the signature with the same tag vector as $t^{*(i)}$, using the programmability of the weak programmable hash functions. Since $\overrightarrow{t}^{*(i)} \notin T_i$ (for $i = n+1$, $\overrightarrow{t}^{*(i)} \notin T_i = \emptyset$) implies that there are at most $m$ tag vectors same as $\overrightarrow{t}^{*(i)}$, the simulator can response $m$ signatures with tag vector $\overrightarrow{t}^{*(i)}$ using the programmability of $(\prod_{i=0}^{m} u_i^{M^i})$. If $l\lambda Q$ is bounded by a polynomial in $\lambda$, then we obtain the polynomial-time reduction.

By applying the above strategy, we give the following theorem.

**Theorem 1.** *The signature scheme in Figure 3 is $(q, \varepsilon, T)$-EUF-naCMA secure assuming the $(\varepsilon_{dh}, T_{dh})$-DH assumption holds such that*

$$\varepsilon_{dh} = \frac{1}{l\lambda Q}(\varepsilon - \frac{q^{m+1}}{(m+1)!Q^{lm}} - \frac{q}{p} - (\frac{q^{m+1}}{(m+1)!Q^m})^{\lambda}) \quad \textit{and} \quad T \approx T_{dh}.$$

Because of space constraints, we relegate the proof Theorem 1 in [30].

To derive a meaningful result about the asymptotic security from Theorem 1, we need the following three conditions.

Condition 1. $l\lambda Q$ is polynomially bounded in $\lambda$.
Condition 2. $\frac{q^{m+1}}{(m+1)!Q^{lm}}$ is a negligible function in $\lambda$.
Condition 3. $(\frac{q^{m+1}}{(m+1)!Q^m})^{\lambda}$ is a negligible function in $\lambda$.

We give asymptotic values of $m$, $l$, and $Q$ for satisfying the above conditions and short public key in Section 4.2. For such parameters (e.g., example parameter 1), we obtain the following corollary by applying the generic transformation using the chameleon hashes.

**Corollary 1.** *Let* SIG *be the signature scheme resulting from the generic transformation on the signature scheme in Figure 3 and* CHSetup*. Then,* SIG *is $(q, 2(\varepsilon + \varepsilon_{\mathsf{CH}}), T)$-EUF-CMA secure assuming $(\frac{\varepsilon}{l\lambda Q} - neg(\lambda), T_{dh})$-DH assumption holds, where $\varepsilon_{\mathsf{CH}}$ is the advantage for breaking the collision resistance of* CHSetup*, $T \approx T_{dh}$.*

**Tag-Free Scheme by Pseudorandom Functions** We apply a trick for tag-free scheme using (non-adaptive) pseudorandom functions (PRF). Note that similar techniques are used in the RSA-based signatures [23, 22, 20, 34] to generate random prime numbers used in each signature. If we use this trick (and the chameleon hashes), we can obtain the generalized (tag-based) signature scheme in Figure 2, where $\forall \mathcal{T}_i = [1, Q]$; each signature has a tag vector that is uniformly chosen from its domain. Thus, a signer can use pseudorandom functions (PRF) mapping from messages to tag vectors, and publishes the PRF the signer used along with its key. Even though the signer publishes the PRF key, (in the non-adaptive security model) we can use the fact that the distribution of tag vectors is indistinguishable from the uniform distribution. The resulting signature size is reduced by eliminating tag vectors from signatures but augmenting signing/verification costs and adding constant factor in public key size (that is, public key size is still $\Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ group elements);

**Corollary 2.** *Let* SIG *be the signature scheme in Figure 2 with* $\forall \mathcal{T}_i = [1, Q]$. *Then,* SIG *is* $(q, 2(\varepsilon + \varepsilon_{\mathsf{CH}} + \varepsilon_{\mathsf{PRF}}), T)$*-EUF-CMA secure assuming* $(\frac{\varepsilon}{l\lambda Q} - neg(\lambda), T_{dh})$*-DH assumption holds, where* $\varepsilon_{\mathsf{CH}}$ *and* $\varepsilon_{\mathsf{PRF}}$ *are advantages for breaking the collision resistance of the chameleon hash functions and the pseudo-randomness of PRF, respectively,* $T \approx T_{dh}$*, and* $neg(\lambda)$ *is a negligible function in* $\lambda$*.*

**Parameter Selection for Short Public Key** The description of our construction did not explain how to choose $m$, $l$, and $Q$. We show how to minimize public key size. From Theorem 1, we obtained three conditions for polynomial-time reduction to the DH problem. First, $l\lambda Q$ should be polynomially bounded in $\lambda$. Second, $\frac{q^{m+1}}{(m+1)!Q^{lm}}$ and $(\frac{q^{m+1}}{(m+1)!Q^m})^\lambda$ should be a negligible function in $\lambda$. For simple analysis, we assume that $Q = Cq$ for small constant $C > 1$ and compute conditions for $m$ and $l$ when $\frac{q^{m+1}}{(m+1)!Q^{lm}}$ and $(\frac{q^{m+1}}{(m+1)!Q^m})^\lambda$ are smaller than $\frac{1}{2^\lambda}$. We compute asymptotically minimal values of $m$ and $l$ for short public key size, and then provide practical parameters with reasonable reduction loss, which is comparable to that of Waters signature in [33].

Condition 1. $l\lambda q$ is polynomially bounded in $\lambda$.
Condition 2. $\frac{q^{m+1}}{(m+1)!Q^{lm}} < \frac{1}{2^\lambda}$.
Condition 3. $(\frac{q^{m+1}}{(m+1)!Q^m})^\lambda < \frac{1}{2^\lambda}$.

From the condition 2, at least the denominator should be larger than $2^\lambda$. Since $Q = Cq$ and $(m+1)! \approx \sqrt{2\pi(m+1)}(\frac{m+1}{e})^{m+1}$ (by Stirling's approximation), where $e$ is the Euler's number, $lm = \Omega(\frac{\lambda}{\log \lambda})$ or $m = \Omega(\frac{\lambda}{\log \lambda})$. For minimizing public key size, we should minimize $m + l$ since the size of public key is $\Theta(m + l)$. Therefore, $m = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ and $l = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ are (asymptotically) minimal parameters for minimal public key size. In fact, if we set $m = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ and $l = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$, then the condition 1 and 3 also hold.

Next, we provide practical parameters for $\lambda \in \{80, 256\}$ and $q \in \{2^{30}, 2^{40}\}$, where $\lambda$ is the security parameter and $q$ is the bound for adversarial signing queries. If the above condition 2 and condition 3 hold, our security proof loses $4l\lambda Q$ factor in the simulation (when we ignore negligible factors), which is asymptotically larger than Waters signature scheme's reduction loss. However, for practical choices of $\lambda$ and $q$, $l$ is a small constant (at most 3 in our example parameters) and $Q$ is $Cq$ with small constant ($C = 2^3$ in our example parameters); and thus, the reduction loss in our example parameters is at most $96\lambda q$, which is comparable to that given in [33][8]. The example practical parameters are given in the table 1. To get the table 1, we firstly set $Q = 2^3 q$ and $q \in \{2^{30}, 2^{40}\}$, and then find small $m$ and $l$ satisfying the above three conditions. The size of a tag vector is a $l\lceil \log Q \rceil$-bit string, which is asymptotically smaller than

[8] Hofheinz et al. proposed a variant of Waters signatures using a special encoding for optimal security reduction $\Theta(\frac{1}{q})$ [21]. In [21], however, they do not provide a concrete constant factor of $\Theta$ notation for practical security parameters, but only asymptotic analysis for optimal security reduction.

| Security Parameter $\lambda$ | | $q$ | $m$ | $l$ | PK size | Sig. size |
|---|---|---|---|---|---|---|
| 80 | tag-based | $2^{30}$ | 7 | 2 | $12\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$ | $2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$ |
| | scheme | $2^{40}$ | 8 | 2 | $13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$ | $2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$ |
| | tag-free | $2^{30}$ | 7 | 2 | $12\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \lvert K\rvert$ | $2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$ |
| | scheme | $2^{40}$ | 8 | 2 | $13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \lvert K\rvert$ | $2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$ |
| 256 | tag-based | $2^{30}$ | 7 | 3 | $13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$ | $2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$ |
| | scheme | $2^{40}$ | 8 | 2 | $13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$ | $2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$ |
| | tag-free | $2^{30}$ | 7 | 3 | $13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \lvert K\rvert$ | $2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$ |
| | scheme | $2^{40}$ | 8 | 2 | $13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \lvert K\rvert$ | $2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$ |

**Table 1. Practical parameters for EUF-$q$-CMA secure signature scheme:** '$\tau_{\mathbb{G}_1}$' and '$\tau_{\mathbb{G}_2}$' are the bit-lengths to represent elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. For type-1 pairings, $\tau_{\mathbb{G}_1} = \tau_{\mathbb{G}_2}$. '$\tau_{\mathbb{Z}_p}$' is the size of prime $p$ that is order of cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_t$. '$\tau_{\mathbb{G}_{ch}}$' is the bit length to represent an element in the group $\mathbb{G}_{ch}$ (of order $p' \leq p$), over which the chameleon hashes defined. '$|K|$' is a size of a PRF key.

$2\lambda$ if $l = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ and $Q$ is a polynomial in $\lambda$; and thus, we can assume that a tag vector is a field element of $\mathbb{Z}_p$. In particular, when we apply practical parameters in the table 1, the size of a tag vector is still smaller than $2\lambda$ (e.g., a tag vector is 129-bit string when $l = 3$ and $Q = 2^{43}$).

**Instantiation using Asymmetric Pairings** Although we described our construction using type-1 pairings, we can easily modify our construction to be instantiated using type-2 pairings or type-3 parings. The scheme using type-1 pairings and its security proof does not use the symmetry property; Our main idea to achieve sub-linear public key is to divide adversarial types according to tag vectors in the security proof, and this technique is independent of pairing's type. We provide the details of the scheme using type-2 or type-3 parings in [30].

# 5 Confined Guessing

**Overview.** In this section, we will explain the schemes from Section 3 as arising from a more general transformation. The final schemes (and in particular $\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{gen}}$) are fully EUF-CMA secure (in contrast to EUF-$q$-CMA security), and have asymptotically very short public keys. The downside of this analysis is that the security reduction is qualitatively worse than the one from Section 4. Namely, the reduction *loss* depends on the adversary's success. Nonetheless, the reduction loss is always polynomial for polynomially bounded adversaries.

## 5.1 From Mild to Full Security

We start with a completely generic transformation from mildly secure tag-based signature schemes to fully secure schemes. We first 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 1 (EUF-naCMA$^*_m$ security).** *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 $\mathsf{Adv}^{\mathsf{euf\text{-}nacma^*_m}}_{\mathsf{SIG_t},F}(\lambda) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}nacma^*_m}}_{\mathsf{SIG_t},F}(\lambda) = 1\right]$ is negligible for any PPT adversary $F$. Here, experiment $\mathsf{Exp}^{\mathsf{euf\text{-}nacma^*_m}}_{\mathsf{SIG_t},F}(\lambda)$ is defined in Figure 4.*

In this subsection, we will show how to use a EUF-naCMA$^*_m$ secure scheme $\mathsf{SIG_t}$ to build an EUF-naCMA secure scheme $\mathsf{SIG}$. (Full EUF-CMA security can then be achieved using chameleon hashing [26].)

To this end, we separate the tag space $\mathcal{T}_\lambda$ into $l := \lfloor \log_c(\lambda) \rfloor$ *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 $\mathcal{T}_\lambda = \{0,1\}^\lambda$, then we may set $\mathcal{T}'_i := \{0,1\}^i$. The constructed signature scheme $\mathsf{SIG}$ 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 $t_i := \mathsf{PRF}^{\mathcal{T}'_i}_\kappa(M)$. The PRF seed $\kappa$ is part of $\mathsf{SIG}$'s public key.[9]

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

**Fig. 4.** The EUF-naCMA$^*_m$ experiment for tag-based signature schemes.

A $\mathsf{SIG}$-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 $\sigma_i$ 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 $\mathcal{T}_\lambda = \bigcup_{i=1}^l \mathcal{T}'_i$, let $m \in \mathbb{N}$ and $c > 1$, and let $\mathsf{PRF}$ be a PRF. $\mathsf{SIG}$ is described in Figure 5.

It is straightforward to verify $\mathsf{SIG}$'s correctness. Before turning to the formal proof, we first give an intuition why $\mathsf{SIG}$ is EUF-naCMA secure. We will map an adversary $F$

[9] 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.

| $\mathsf{Gen}(1^\lambda)$ | $\mathsf{Sig}(sk, M)$ | $\mathsf{Ver}((pk', \kappa), M, \sigma = (\sigma_i)_{i=1}^l)$ |
|---|---|---|
| $(pk', sk) \leftarrow \mathsf{Gen}_\mathsf{t}(1^\lambda)$ | $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M)$ for $i \in [l]$ | $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M)$ for $i \in [l]$ |
| $\kappa \leftarrow \{0,1\}^\lambda$ | $\sigma_i \leftarrow \mathsf{Sig}_\mathsf{t}(sk, M, t_i)$ | return $\bigwedge_{i=1}^{l} \mathsf{Ver}_\mathsf{t}(pk', M, \sigma_i, t_i)$ |
| $pk := (pk', \kappa)$ | return $\sigma := (\sigma_i)_{i=1}^l$ | |
| return $(pk, sk)$ | | |

**Fig. 5.** Our EUF-naCMA secure signature scheme.

on $\mathsf{SIG}$'s EUF-naCMA security to an adversary $F'$ on $\mathsf{SIG}_\mathsf{t}$'s EUF-naCMA$^*_m$ security. Intuitively, $F'$ will internally simulate the EUF-naCMA security experiment for $F$ and embed its own $\mathsf{SIG}_\mathsf{t}$-instance (with public key $pk'$) in the $\mathsf{SIG}$-instance of $F$ by setting $pk := (pk', \kappa)$. Additionally, the seed $\kappa$ for $\mathsf{PRF}$ is chosen internally by $F'$.

Say that $F$ makes $q = q(\lambda)$ (non-adaptive) signing requests for messages $M_j$, for all $j \in [q]$. 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 $\mathsf{SIG}$, 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 $\mathsf{SIG}_\mathsf{t}$-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 will be a suitable choice of $i^*$. First, recall that the $i$-th $\mathsf{SIG}_\mathsf{t}$-instance in $\mathsf{SIG}$ 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 queried by $F'$).

We turn to a formal proof:

**Theorem 2.** *If $\mathsf{PRF}$ is a PRF and $\mathsf{SIG}_\mathsf{t}$ is an EUF-naCMA$^*_m$ secure tag-based signature scheme, then $\mathsf{SIG}$ is EUF-naCMA secure. Concretely, let $F$ be an EUF-naCMA forger on $\mathsf{SIG}$ with non-negligible advantage $\varepsilon := \mathsf{Adv}^{\mathsf{euf\text{-}nacma}}_{\mathsf{SIG},F}(\lambda)$ and making $q = q(\lambda)$ signature queries. Then $\varepsilon(\lambda) > \frac{1}{p(\lambda)}$ for some polynomial $p$ and $\lambda \in K$ for an infinite set $K \subseteq \mathbb{N}$. For $\lambda \in K$ there exists a EUF-naCMA$^*_m$ forger $F'$ on $\mathsf{SIG}_\mathsf{t}$ with advantage $\varepsilon' := \mathsf{Adv}^{\mathsf{euf\text{-}nacma}^*_\mathsf{m}}_{\mathsf{SIG}_\mathsf{t},F'}(\lambda)$ and making $q'(\lambda) \leq \left(\frac{q^{m+1}}{\varepsilon(\lambda)}\right)^{c/m}$ signature queries, such that $\varepsilon' \geq \varepsilon/2 - \varepsilon_{\mathsf{PRF}} - \frac{1}{|\mathcal{M}_\lambda|}$, where $\varepsilon_{\mathsf{PRF}}$ is the advantage of a suitable PRF distinguisher on $\mathsf{PRF}$ and $\mathcal{M}_\lambda$ the message space.*

*Proof.* First, $F'$ receives messages $M_1, \ldots, M_q$ from $F$. Let $\varepsilon(\lambda)$ be $F$'s advantage in the EUF-naCMA experiment. $F'$ chooses the challenge instance $i^*$ such that the probability of an $(m+1)$-tag collision is at most $\varepsilon(\lambda)/2$, i.e.,

$$\Pr\left[\exists\, \{j_0, \ldots, j_m\} \subseteq [q] : t_{i^*}^{(j_0)} = \ldots = t_{i^*}^{(j_m)} \mid \forall j \in [q] : t_{i^*}^{(j)} \leftarrow \mathcal{T}_{i^*}\right] \leq \frac{\varepsilon(\lambda)}{2}, \quad (1)$$

and such that $|\mathcal{T}'_{i^*}|$ is polynomial in $\lambda$. Concretely, $i^* := \lceil \log_c(\log_2((\frac{q^{m+1}}{\varepsilon(\lambda)})^{1/m})) \rceil$ is an index that fulfills these conditions. (See [8, Lemma 3.5] for a complete analysis.) $F'$ then chooses a PRF-key $\kappa \leftarrow \{0,1\}^\lambda$.

Recall that a signature $\sigma = (\sigma_1, \ldots, \sigma_l)$ of $\mathsf{SIG}$ consists of $l$ signatures of $\mathsf{SIG}_\mathsf{t}$. In the sequel we write $\sigma^{(j)} = (\sigma_1^{(j)}, \ldots, \sigma_l^{(j)})$ to denote the $\mathsf{SIG}$-signature for message $M_j$, for all $j \in \{1, \ldots, q\}$. Adversary $F'$ uses its signing oracle provided from the $\mathsf{SIG}_\mathsf{t}$-security experiment to simulate these $\mathsf{SIG}$-signatures. To this end, it proceeds as follows.

**Simulation of signatures.** 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 $\mathsf{SIG}_\mathsf{t}$-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 $M \leftarrow \mathcal{M}_\lambda$ 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 $(pk, \sigma^{(1)}, \ldots, \sigma^{(q)})$ to $F$. Note that each $\sigma^{(j)}$ is a valid $\mathsf{SIG}$- signature for message $M_j$.

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

This concludes the description of $F'$.

**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 $\Pr[M = M^*] \leq 1/|\mathcal{M}_\lambda|$. Hence, to prove our theorem, it suffices to show that $\Pr[\mathsf{bad}_\mathsf{abort}] \leq \varepsilon/2 + \varepsilon_\mathsf{PRF} + 1/|\mathcal{M}_\lambda|$ 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 $\mathsf{PRF}$ 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 $\mathsf{PRF}$'s pseudorandomness. Concretely, a PRF distinguisher that simulates $F'$ until the decision to abort is made shows $\Pr[\mathsf{bad}_\mathsf{abort}] \leq \varepsilon/2 + \varepsilon_\mathsf{PRF} + 1/|\mathcal{M}_\lambda|$, 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 [22, Lemma 2.3] for details.

### 5.2 The CDH-based scheme

In this subsection we explain the schemes from Section 3 in the our confined guessing framework. We start with with the tag-based scheme $\mathsf{SIG}^{\mathsf{CDH}}$ (Figure 1) and prove it EUF-naCMA$^*_m$-secure. Then we can apply our generic transformation from Section 5.1 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. The result is essentially the generic scheme $\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{gen}}$.

**Theorem 3.** *If CDH holds in $\mathbb{G}$, then $\mathsf{SIG}^{\mathsf{CDH}}$ from Figure 1 is EUF-naCMA$^*_m$-secure. Let $F$ be a PPT adversary with advantage $\varepsilon := \varepsilon(\lambda) := \mathsf{Adv}^{\mathsf{euf\text{-}nacma}^*_{\mathsf{m}}}_{\mathsf{SIG}^{\mathsf{CDH}},F}(\lambda)$ and runtime $T$ asking for $q = q(\lambda)$ signatures, then it can be used to solve a CDH challenge with probability $\epsilon_{dh} = (\varepsilon \cdot m)/q$ and $T_{dh} \approx T$.*

*Proof sketch.* Because of space constraints we will only sketch the proof here. The complete proof can be found in [8].

**Public key setup.** The simulation receives a CDH challenge $(g, g^a, g^b)$ and signature queries $(M_i, t_i)_{i\in[q]}$.We guess an index $i^* \leftarrow [q]$ for which we expect $F$ to forge a signature on a new message $M^* \neq M_{i^*}$, but with $t^* = t_{i^*}$. Let $M^*_j$ (for $j \in [m]$) denote the corresponding messages to $t_{i^*}$. We set up a polynomial $f(X) := \prod_{i=1}^{m}(X - M^*_i) = \sum_{i=0}^{m} d_i X^i \in \mathbb{Z}_p[X]$ and choose random exponents $r_0, \ldots, r_m, x_z, x_h \in \mathbb{Z}_p$. Write $r(X) := \sum_{i=0}^{m} r_i X^i$, so $\mathbf{u}^M = g^{bf(M)+r(M)}$. We embed our challenge, the coefficients $d_i$, $t_{i^*}$ and the random exponents in the public key as follows: we set $pk := (g, g^a, u_0, \ldots, u_m, z, h)$ for $u_i := (g^b)^{d_i} g^{r_i}$, $z := g^{b+x_z}$, and $h := g^{-bt_{i^*}} g^{x_h}$. (Observe that this implicitly sets $sk := a$.)

**Signing.** We have to consider two cases. If $t_i = t_{i^*}$ and thus $M_i = M^*_j$ for some $j$, we choose a random $s_i \leftarrow \mathbb{Z}_p$ and set $\tilde{\sigma}_{1,i} = (g^a)^{r(M^*_j)} \cdot (z^{t_{i^*}} h)^{s_i}$ , $\tilde{\sigma}_{2,i} = g^{s_i}$. Since $f(M^*_j) = 0$, this signature is valid:

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

If $t_i \neq t_{i^*}$, let $s'_i \leftarrow \mathbb{Z}_p$ and $S_i := g^{s'_i}/(g^a)^{f(M_i)/(t_i - t_{i^*})} = g^{s'_i - af(M_i)/(t_i - t_{i^*})}$. A valid signature $\sigma_i := (\tilde{\sigma}_{1,i}, \tilde{\sigma}_{2,i})$ can then be computed as follows: $\tilde{\sigma}_{1,i} = (g^a)^{r(M_i)} \cdot S_i^{x_z t_i + x_h} \cdot (g^b)^{s'_i(t_i - t_{i^*})}$ and $\tilde{\sigma}_{2,i} = S_i$.

**Extract from forgery.** If $F$ generates a valid signature $(M^*, t^*, \sigma^*)$ for $t^* = t_{i^*}$, then $\tilde{\sigma}^*_1 = ((g^b)^{f(M^*)}(g^{r(M^*)}))^a((g^{b+x_z})^{t^*}(g^{x_h - bt_{i^*}}))^{s^*} = g^{abf(M^*)} g^{ar(M^*)} g^{s^*(x_z t^* + x_h)}$
As $M^* \neq M^*_j$, we have $f(M^*) \neq 0$, so $(\tilde{\sigma}^*_1/(g^{ar(M^*)} \tilde{\sigma}^{*(x_z t^* + x_h)}_2))^{1/f(M^*)} = g^{ab}$.

**Analysis.** We denote by $\varepsilon$ the advantage of the adversary $F$ in the experiment and by success the event that the simulation outputs a solution $g^{ab}$. Since the exponents are randomly chosen this yields the correct distribution for $F$. The simulator is successful if $F$ is successful and it guesses $t^*$ correctly. So we have $\Pr[\mathsf{success}] = \frac{\varepsilon \cdot m}{q}$. □

**Optimizations.** Now, with this result and our generic transformation from Section 5.1 we can construct a stateless signature scheme, which is proven EUF-naCMA secure by Theorem 2. By applying a chameleon hash function $\mathsf{CH}$, we obtain a fully EUF-CMA-secure signature scheme. This signature scheme does have a constant size public key but signatures consist of $O(\log(\lambda))$ group elements.

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 [27]. We re-use $u_0, \ldots, u_m$, one $sk := \alpha$ and one randomness $s$ for all instances $i$ (see Figure 2). 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. The result is scheme $\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{gen}}$ (Figure 2). The proof of the following theorem can be found in [8].

**Theorem 4.** *If the CDH assumption holds in $\mathbb{G}$ and $\mathsf{CH}$ is a chameleon hash function, then $\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{gen}}$ (Figure 2) is EUF-CMA secure. Let $F$ be a PPT adversary with advantage $\varepsilon := \varepsilon(\lambda) := \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG}^{\mathsf{CDH}}_{\mathsf{gen}},F}(\lambda)$ and runtime $T$ asking for $q := q(\lambda)$ signatures, then it can be used to solve a CDH challenge with probability at least $\epsilon_{dh} = \frac{\varepsilon^{c/m+1}}{2^{c/m+1} \cdot q^{c(m+1)/m}} - \varepsilon_{\mathsf{PRF}} - \varepsilon_{\mathsf{CH}}$, where $\varepsilon_{\mathsf{PRF}}$ and $\varepsilon_{\mathsf{CH}}$ correspond to the advantages for breaking the PRF and the chameleon hash respectively. $T_{dh} \approx T$.*

### 5.3 Our RSA-based scheme

In this subsection we construct a stateless signature scheme $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{opt}}$ that is EUF-CMA-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 [23], to which we refer as $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{HW09}}$ 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 5.1 and add a chameleon hash to construct a fully secure stateless scheme. Finally we apply common aggregation techniques which yields the optimized scheme $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{opt}}$.

**Definition 2 (RSA assumption).** *Let $N \in \mathbb{N}$ be the product of two distinct safe primes $P$ and $Q$ with $2^{\frac{\lambda}{2}} \leq P, Q \leq 2^{\frac{\lambda}{2}+1} - 1$. 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*

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

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

**EUF-naCMA$^*_m$-Secure Signature Scheme.** We start with the basic scheme $\mathsf{SIG}^{\mathsf{RSA}}$. Let $N = PQ$ be an RSA modulus consistent with the RSA assumption (Definition 2). 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 3), 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 $\ell = \lambda/2$ for our realization – we will need later that $\frac{1}{2^{\lambda-\ell}}$ is negligible. To construct a mapping from tags to primes we use a technique from [22] and [20]: For a PRF $\mathsf{PRF}^{\{0,1\}^\lambda}$, a corresponding key $\kappa \leftarrow \{0,1\}^\lambda$, and a random bitstring $b \leftarrow \{0,1\}^\lambda$, we define

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

where $\mu_t := \min\{\mu \in \mathbb{N} : \mathsf{PRF}_\kappa^{\{0,1\}^\lambda}(t||\mu) \oplus b \text{ is prime}\}$ and $||$ denotes the concatenation of bitstrings.[10] We call $\mu_t$ the *resolving index* of $t$. The complete scheme $\mathsf{SIG}^{\mathsf{RSA}}$ is depicted in Figure 6.

| $\mathsf{Gen}_\mathsf{t}(1^\lambda)$ | $\mathsf{Sig}_\mathsf{t}(sk, M, t)$ | $\mathsf{Ver}_\mathsf{t}(pk, M, \sigma = (\hat{\sigma}, t))$ |
|---|---|---|
| Pick modulus $N = PQ$, | $p := \mathsf{P}_{(\kappa,b)}(t)$ | if $t \notin \mathcal{T}$ |
| $u_i \leftarrow QR_N$ | $\hat{\sigma} :=$ | return 0 |
| $(i \in \{0, \ldots, m\})$ | $(\prod_{i=0}^{m} u_i^{M^i})^{\frac{1}{p}} \bmod N$ | $p := \mathsf{P}_{(\kappa,b)}(t)$ |
| $\kappa \leftarrow \{0,1\}^\lambda$ | return $(\hat{\sigma}, t)$ | if $\hat{\sigma}^p \not\equiv \prod_{i=0}^{m} u_i^{M^i} \bmod N$ |
| $b \leftarrow \{0,1\}^\lambda$ | | return 0 |
| $pk := (N, (u_i)_{i=0}^m, \kappa, b)$ | | else |
| $sk := (P, Q)$ | | return 1 |
| return $(pk, sk)$ | | |

**Fig. 6.** The tag-based RSA scheme $\mathsf{SIG}^{\mathsf{RSA}}$.

**Differences to $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{HW09}}$.** We give a brief overview how $\mathsf{SIG}^{\mathsf{RSA}}$ relates to $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{HW09}}$. To reduce overhead, we first removed all components from $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{HW09}}$ that are not required to prove EUF-naCMA$^*_m$-security. This includes the chameleon hash (we are in a non-adaptive setting) and the logarithm-of-tag-construction (we guess from a small set of tags only). Our setup of $\mathsf{P}_{(\kappa,b)}$ slightly differs from the one in $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{HW09}}$ since we do need that *every* tag is mapped to a prime.

**Theorem 5.** *If $F$ is a PPT EUF-naCMA$^*_m$-adversary for $\mathsf{SIG}^{\mathsf{RSA}}$ with advantage $\varepsilon := \mathsf{Adv}^{\mathsf{euf\text{-}nacma}^*_\mathsf{m}}_{\mathsf{SIG}^{\mathsf{RSA}},F}(\lambda)$ asking for $q := q(\lambda)$ signatures, then it can be used to efficiently solve an RSA challenge according to Definition 2 with probability at least*

$$\frac{\varepsilon}{q'\lambda^2} - \lambda^2 \varepsilon_{\mathsf{PRF}} - O(\frac{1}{2^{\lambda/2}})$$

*where $q'$ denotes the number of distinct tags queried by $F$ and $\varepsilon_{\mathsf{PRF}}$ is the advantage of a suitable distinguisher for the PRF.*

The proof for Theorem 5 can be found in [8]. The main idea is exactly that of the proof for Theorem 3. One additional challenges is that we have to map every tag

[10] $\mathsf{P}_{(\kappa,b)}(t)$ 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 $t$ $\mu_t$. For a proper $\overline{\mu}$ the event $\mu_t > \overline{\mu}$ will only occur with negligible probability (see [8], Theorem 5.6).

to a prime (realized by $\mathsf{P}_{(\kappa,b)}$). Furthermore, in the extraction phase, we need to use Shamir's trick here to perform a division in the exponent – this requires coprime exponents. Now, by Theorem 5, our generic transformation from Section 5.1 applied to $\mathsf{SIG}^{\mathsf{RSA}}$ yields an EUF-naCMA-secure signature scheme. Finally, we use chameleon hashing [26] to generically construct the fully secure scheme $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{gen}}$, like for instance the RSA-based chameleon hash from [22, Appendix C].

**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 $l = \lfloor \log_c(\lambda) \rfloor$ $\mathsf{SIG}^{\mathsf{RSA}}$ 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 of a group element). As mentioned, the functions $(\mathsf{PRF}^{\mathcal{T}_i})_{i\in[l]}$ necessary to generate the tags for a signature can be generically constructed from $\mathsf{PRF}^{\{0,1\}^\lambda}$.

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

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

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

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

**Theorem 6.** *Let $F$ be a PPT EUF-CMA adversary against $\mathsf{SIG}^{\mathsf{RSA}}_{\mathsf{opt}}$ with advantage $\varepsilon := \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG}^{\mathsf{RSA}},F}(\lambda)$ asking for $q := q(\lambda)$ signatures (at most). Then it can be used to efficiently solve an RSA challenge according to Definition 2 with probability at least*

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

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

The proof for Theorem 6 can be found in [8] and is analogous to the proof for Theorem 4. Again, we have the additional challenges of mapping tags to primes and meeting the prerequisite for Shamir's trick in the extraction phase.

### 5.4 Our SIS-based scheme

Let us now sketch the SIS-based signature scheme. Due to space limitations, this sketch is extremely brief. We refer to [8] for details.

In 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.

Again we construct a tag-based signature scheme first and prove EUF-naCMA$^*_1$-security. The scheme is described in Figure 8. It is based on techniques from [16, 2, 10, 7], in particular it exhibits many similarities to the identity-key generation algorithm of the IBE scheme from [2], where our tags correspond to identities of [2].

Figure 8 uses two algorithms $\mathsf{TrapGen}$ and $\mathsf{SampleLeft}$, which we unfortunately can not describe in detail here. Essentially, $\mathsf{TrapGen}$ computes a matrix $A \in \mathbb{Z}_q^{n\times m}$ together with a short basis $T_A$ of $\Lambda_p^\perp(A)$, and $\mathsf{SampleLeft}$ samples a short vector $e \in \mathbb{Z}_p^{2m}$ statistically close to $\mathcal{D}_{\Lambda_p^u(A|G_t),\gamma}$. A difference to [2] is that we must be able to issue one signature for message-tag-pair $(M_i, t_i)$ with $t_i = t^*$, but without knowing any trapdoor. This is resolved by a suitable set-up of the vector $v$ contained in the public key in the proof. See [8] for details.

$\mathsf{Gen_t}(1^\lambda)$
- $(A, T_A) \leftarrow \mathsf{TrapGen}(p, n)$
- $Z, Y \leftarrow \mathbb{Z}_q^{n\times m}$
- $U \leftarrow \mathbb{Z}_q^{n\times \ell}$
- $v \leftarrow \mathbb{Z}_p^n$
- $sk := T_A$
- $pk := (U, A, Z, Y, v)$
- 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 $e \le 0$ 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

**Fig. 8.** The EUF-naCMA$^*_1$-secure SIS scheme.

**EUF-CMA-Secure Scheme.** By applying the generic transformation from Section 5.1 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 $\kappa$ for the $\mathsf{PRF}$. Signatures consist of $l$ low-norm vectors in $\mathbb{Z}_p^n$, where $l = \lfloor \log_c(\lambda) \rfloor = O(\log \lambda)$ is defined as in Section 5.1. 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, one can combine this EUF-naCMA-secure scheme with a suitable chameleon hash function, like for instance the SIS-based construction from [10, Section 4.1]. This 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.

## References


[1] Memoirs of the 6th Cryptology Paper Contest, arranged by Korea Communications Commission.

[2] 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.

[3] 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.

[4] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73, Interlaken, Switzerland, May 2–6, 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] Dan Boneh, Ilya Mironov, and Victor Shoup. A secure signature scheme from bilinear maps. In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 98–110, San Francisco, CA, USA, April 13–17, 2003. Springer, Berlin, Germany.

[7] 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.

[8] Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, and Christoph Striecks. Confined guessing: New signatures from standard assumptions. Cryptology ePrint Archive, Report 2013/171, 2013. `http://eprint.iacr.org/`.

[9] David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 127–145, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.

[10] 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.

[11] 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.

[12] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25, Santa Barbara, CA, USA, August 23–27, 1998. 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] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Berlin, Germany.

[15] 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.

[16] 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.

[17] Michael Gerbush, Allison B. Lewko, Adam O'Neill, and Brent Waters. Dual form signatures: An approach for proving security from static assumptions. In *ASIACRYPT*, pages 25–42, 2012.

[18] Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.

[19] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. *Journal of Cryptology*, 25(3):484–527, 2012.

[20] 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.

[21] Dennis Hofheinz, Tibor Jager, and Edward Knapp. Waters signatures with optimal security reduction. In *PKC 2012*, volume 7293 of *LNCS*, pages 66–83. Springer, 2012.

[22] 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.

[23] 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.

[24] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61, Seattle, Washington, USA, May 15–17, 1989. ACM Press.

[25] 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.

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

[27] 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.

[28] Jörn Müller-Quade Nico Döttling and Anderson C. A. Nascimento. IND-CCA secure cryptography based on a variant of the LPN problem. In *ASIACRYPT 2012*, LNCS. Springer-Verlag, 2012. Appears.

[29] 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.

[30] Jae Hong Seo. Short signature from Diffie-Hellman: Realizing short public key. In *http://eprint.iacr.org/2012/480*, 2012.

[31] Adi Shamir and Yael Tauman. Improved online/offline signature schemes. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 355–367. Springer, 2001.

[32] 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.

[33] 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.

[34] Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Space efficient signature schemes from the RSA assumption. In *PKC 2012*, volume 7293 of *LNCS*, pages 102–119. Springer, 2012.