Short Signatures from Diffie-Hellman, Revisited: Sublinear Public Key, CMA Security, and Tighter Reduction

Jae Hong Seo

Myongji University
jaehongseo@mju.ac.kr

Abstract. Designing efficient signature scheme based on the standard assumption such as the Computational Diffie-Hellman (CDH) assumption is important both from a practical and a theoretical point of view. Currently, there are only three standard model CDH-based signature schemes with short signatures due to Waters (EUROCRYPT 2005), and Seo and Böhl *et al.* (the merged paper in EUROCRYPT 2013). The Waters signature scheme achieves the *Existentail UnForgeability against Chosen Message Attack (EUF-CMA)* with nearly optimal reduction. However, this scheme suffers from large public keys. To shorten public key size, Seo and Böhl *et al.* proposed new approaches, respectively, but each approach has a weak point rather than the Waters signature scheme; Seo's approach could prove only a rather weak security, called the bounded CMA security, and Böhl *et al.*'s approach inherently accompanies a loose reduction.

In this paper, we aim at stepping towards efficient CDH-based EUF-CMA secure signature scheme with tighter reduction. To this end, we revisit the Seo signature scheme and devise an alternative security proof. The resulting security proof leads

1. *asymptotically* (almost) compact parameters; short signatures (two group elements and one exponent) and $\omega(1)$ public keys (e.g., $\log\log\lambda$), where λ is the security parameter, and
2. the standard EUF-CMA security with tighter reduction; $O(\lambda q)$ reduction loss, when ignoring negligible factors, which is less than $O(\sqrt{\frac{\lambda}{\log}}\lambda q)$ of the original security proof and almost the same as that of the Water signature scheme.

1 Introduction

Designing practical signature scheme based on reliable assumptions is very important both from a theoretical and a practical standpoint. In particular, it is desirable to design a signature scheme that is secure under the standard assumption such as the Computational Diffie-Hellman (CDH) assumption and the RSA assumption since such the standard assumptions have been analyzed for a long time and so those are stable and relatively reliable. But while there has been proposed a lot of signature schemes from the standard assumptions, there are only few candidates satisfying what the practitioners have come to expect; digital signature schemes can be designed from the general assumption [26, 1, 28, 31], or from general and concrete assumptions [19, 18, 27, 12, 9, 10]. These constructions are categorized in the so-called tree-based approach and are relatively inefficient in comparison with the other approach, called the hash-and-sign. While the most of practical signature schemes are in the category of the hash-and-sign approach, almost all hash-and-sign signature schemes require the heuristic random oracles [13, 32, 29, 2, 7, 17, 16] or the strong assumptions (Strong RSA assumption [15, 11, 14], non-static assumptions [5, 30, 20, 22], interactive assumption [8]). These assumptions are relatively less analyzed than the standard assumptions such as CDH and RSA assumptions.

Currently, there are only three standard model CDH-based signature schemes with short signatures due to Waters [35], Seo [33], and Böhl *et al.* [4].[1] The Waters signature scheme achieves the standard security notion, called *Existential UnForgeability against the Chosen-Message-Attack (EUF-CMA)* [19]. Furthermore, his analysis achieves nearly optimal reduction loss [21], where the reduction loss means a ratio between the success probability of the adversary and that of the reduction algorithm. However, the Waters signature scheme suffers from large public keys. Seo signature scheme yields not only asymptotically sublinear public keys, but also practical parameters for concrete security parameters. However, the security theorem guarantees only a rather weak (non-standard) security, called the bounded CMA security; the (polynomial) bound of allowable signing oracles should be fixed at the parameter generating time in the bounded CMA security. Hence, the given proof does not guarantee any security when one signs more than the pre-determined (polynomially many) times, and this is an undesirable property in practice. Böhl *et al.*'s signature scheme (BHJKS) achieves the asymptotically shortest public keys and the standard EUF-CMA security via developing a new proof technique, called the confined-guessing, which is widely applicable.[2] However, the advantage of the BHJKS signature scheme over the other schemes in the asymptotic efficiency inherently accompanies a loose reduction. In particular, the reduction loss in [4] depends on the adversarial success probability; and thus, the reduction loss increases when the adversary has smaller success probability.[3] Therefore, the reduction loss in [4] will increase when we choose parameters to cover all possible adversaries in practice. The tightness of security proof is very important in practice since a scheme with a tighter reduction has short parameters and so all operations such as group exponentiations and pairing operations will be much cheaper than those of a scheme with a loose reduction. Although the Seo signature scheme and the BHJKS signature scheme have an advantage in public key size over the Waters signature scheme, both also have disadvantages in the security argument (weak security and loose reduction, respectively). In this paper, we aim at stepping towards efficient CDH-based EUF-CMA secure signatures with tighter reduction.

Note that the above three CDH-based signature schemes due to Waters, Seo, and Böhl *et al.* are designed over bilinear groups and it is still open to construct a short and standard model CDH-based signature scheme without using pairings. We provide a comparison among the short and standard model CDH-based signature schemes using bilinear groups in Table 1.

Our approach toward CDH-based CMA secure scheme with tighter reduction. Even if the Seo signature scheme does not achieve the CMA security, its reduction loss is comparable with that of the Waters signatures. For our goal, we revisit the Seo signature scheme, and then devise a new analysis on the Seo signatures. Surprisingly, we find that our new security reduction can yield asymptotically (almost) compact public keys and at the same time achieve the standard EUF-CMA security with tighter reduction. The reduction loss of the proposed scheme is $O(\lambda q)$ and

[1] The merged paper of [33] and [4] is published in [3]. Each paper is based on totally different approach, though both the resulting schemes look the same except the tag sizes. Thus, we separately mention and cite them.

[2] Böhl *et al.* [4] proposed two more signature schemes based on the standard assumption such as RSA and SIS using the confined-guessing technique.

[3] For the reduction loss of the scheme in [4], we take the result of Theorem 4.3 in [4]. In particular, the main theorem [4, Theorem 3.3] about general methodology for EUF-CMA secure signature scheme implies that what the simulator has to do (that is, the number of queries by the simulator for constructing EUF-naCMA$^*_m$ attacker) increases according to the adversarial success probability.

Scheme	PK Size	Sig. Size	Reduction Loss	Sec. Model
Waters [35]	$O(\lambda)\tau_{\mathbb{G}}$	$2\tau_{\mathbb{G}}$	$O(\lambda q)$	EUF-CMA
Seo [33] (tag-free)	$O(\sqrt{\frac{\lambda}{\log\lambda}})\tau_{\mathbb{G}}$	$2\tau_{\mathbb{G}}+2\tau_{\mathbb{F}_p}$ $(2\tau_{\mathbb{G}}+1\tau_{\mathbb{F}_p})$	$O(\sqrt{\frac{\lambda}{\log\lambda}}\lambda q)$	EUF-q-CMA
BHJKS [4]	$O(\log_d \lambda)\tau_{\mathbb{G}}$	$2\tau_{\mathbb{G}}+1\tau_{\mathbb{F}_p}$	$O(\frac{2^{2+\frac{d}{m'}} q^{\frac{d}{m'}+d}}{\varepsilon^{\frac{d}{m'}}})$	EUF-CMA
This paper (tag-free)	$\omega(1)\tau_{\mathbb{G}}$	$2\tau_{\mathbb{G}}+2\tau_{\mathbb{F}_p}$ $(2\tau_{\mathbb{G}}+1\tau_{\mathbb{F}_p})$	$O(\lambda q)$	EUF-CMA

Table 1. CDH-based signature schemes: λ is the security parameter, $\tau_{\mathbb{G}}$ is the size of group element, $\tau_{\mathbb{F}_p}$ is the size of the exponent, q is the maximum bound of the signing queries, d are arbitrary constants satisfying $c \geq 1$ and $d > 1$. Note that for our result, we can set $c = 1$ so that the reduction loss is $O(\lambda q)$. m' in [4] is a constant, and ε in [4] is the success probability of the adversary. In PK size, $\omega(1)$ means any strictly increasing function in λ; e.g., $\log\log\lambda$.

it is smaller than $O(\sqrt{\frac{\lambda}{\log}}\lambda q)$ of the original security proof [33] and almost the same as that of the Water signature scheme.

Signature Scheme in [33], Revisited. In [33], the *asymmetric trade* using "the generalized version of the generalized birthday lemma" is focused to explain the reason why the resulting scheme achieved sublinear public keys. We find that the essential idea behind the construction and the analysis in [33] can be interpreted as the *prefix-guessing* technique and we give a new security proof basing on an alternative and simple prefix-guessing, which is not fully relying on the generalized version of the generalized birthday lemma unlike the original proof in [33]. The prefix-guessing is a proof technique introduced by Hohenberger and Waters to design weakly-secure signatures [24]. The goal of this technique is to guess a prefix of the message such that the adversary will forge on it and it is also not used in signing queries, and then to embed the challenge into the public parameters by using the knowledge of this prefix. This technique is useful in the weak CMA security model, in which the adversary should send all signing queries to the challenger before receiving public parameters. In the weak CMA security model, the simulator knows all message in advance, which will be used in signing queries, and that the message on which the adversary will forge should be different from all messages used in signing queries. Hence, the simulator can use this information to restrict the domain for prefix-guessing to be a polynomial, and so the simulator can correctly guess the prefix with a non-negligible probability (before generating public parameters). By using the standard technique for transformation from weakly-secure signatures to EUF-CMA secure signatures, we can obtain EUF-CMA secure signature scheme.

The analysis in [33] follows the basic flow of using the prefix-guess technique by Hohenberger and Waters, but the details how to use additional information to restrict the domain for prefix-guessing is quite different. The Seo signatures contain random tag vectors. In the security proof, the prefix-guessing technique is applied to random tag vectors instead of messages so that we cannot expect that the tag vector in the forgery is distinct from all tag vectors used in signing queries. Hence, a new way to restrict domain of prefix-guessing, which is a different way from that originally used by Hohenberger and Waters[4], is devised in [33], and the analysis is relying on the generalized version of the generalized birthday lemma. We will explain the details in the body of this paper.

[4] We can consider the prefix-guessing technique used by Hohenberger and Waters [24] as a kind of *partitioning* technique [35] since the simulator divides the space of prefixes of messages into the signable space and the unsignable space. For the prefix-guessing used in [33], however, the space of prefixes of tag vectors is not strictly divided into the signable space and the unsignable space.

In the construction, the bound q of allowable signing queries is used as a part of the public parameter. In the security proof, the reduction algorithm loses q factor for prefix-guessing, and so q should be kept as a polynomial for polynomial time reduction. Consequently, only the q-bounded CMA security of the Seo signature scheme is proved for a polynomial q. However, we find that the essential reason why the approach can achieve the sublinear public keys is the prefix-guessing (in particular, the way to restrict domain) and this idea can be independent from an undesirable relation between the bound q and the public parameters. Basing on this intuition, we revisit [33] and prove the CMA security of the Seo signatures with even better reduction efficiency.

Outline. We first give preliminaries and definitions in the next section. Next, we revisit the Seo signatures in Section 3, and then provide a new analysis on the Seo signatures in Section 4. In Section 5, we discuss two natural extensions of the Seo signatures.

2 Preliminaries and Definitions

Notation. For an algorithm Alg, $Alg(x) \to a$ means that Alg outputs a on input x. If the input of Alg is clear from the context, we sometimes omit it and simply write $Alg \to a$. For a set S, $s \xleftarrow{\$} S$ denotes that the element s is uniformly chosen from S. A negligible function is a function $\mu(\lambda) : \mathbb{N} \to \mathbb{R}$ such that for every positive polynomial $poly(\cdot)$, there exists a positive integer N_{poly} such that for all $\lambda > N_{poly}$, $|\mu(\lambda)| < \frac{1}{poly(\lambda)}$. For two functions a and b in λ, $a \sim b$ means that $|a-b|$ is a negligible function in λ.

2.1 Syntax and Security of Signature Scheme

Signature Scheme. A signature scheme consists of three algorithms, KeyGen, Sign, and Verify.

- $\mathsf{KeyGen}(\lambda)$: It takes the security parameter λ and outputs a keypair (PK,SK).
- Sign(PK,M,SK): It takes the public key PK, the secret key SK, and a message M and outputs a signature σ.
- Verify(PK,M,σ): It takes the public key PK, a message M, and a signature σ and returns 1 if the signature is valid; otherwise, 0.

Existential UnForgeability. The standard security notion for signature schemes, called *Existential UnForgeability with respect to Chosen-Message Attacks* (EUF-CMA), is formalized by Goldwasser, Micali, and Rivest [19]. There is a slightly weaker model called *Existential UnForgeability with respect to weak Chosen-Message Attacks* (EUF-wCMA). The adversary in both security models is given the public key and access to a signing oracle, and wins if she can produce a valid pair of a signature and a message on which the adversary did not query to the signing oracle. In the EUF-CMA security model, the adversary is allowed to query any time before she outputs a forgery. However, the adversary in the EUF-wCMA model should send the challenger the entire list of messages she wants to query before receiving the public key; thus, we sometimes say the adversary in the EUF-wCMA model the non-adaptive adversary. We provide the formal definition of EUF-CMA secure signature scheme and EUF-wCMA secure signature scheme. Let $\mathsf{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$

be a signature scheme. We consider two following experiments.

$\mathbf{Exp}^{\text{EUF-CMA}}_{\mathsf{SIG},\mathcal{A}}(\lambda)$
$(PK, SK) \leftarrow \mathsf{KeyGen}(\lambda)$;
$(M, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}(\cdot)}(PK)$;
Define L as the set of all messages queried by the adversary;
Return $\begin{cases} 1 \text{ if } M \notin L \\ \quad \text{and } \mathsf{Verify}(PK, M, \sigma) = 1, \\ 0 \text{ Otherwise.} \end{cases}$

$\mathbf{Exp}^{\text{EUF-wCMA}}_{\mathsf{SIG},\mathcal{A}}(\lambda)$
$(M_1, \ldots, M_q, st) \leftarrow \mathcal{A}(st)$;
$(PK, SK) \leftarrow \mathsf{KeyGen}(\lambda)$;
For $\forall i \in [1, q]$, $\sigma_i \leftarrow \mathsf{Sign}(PK, M_i, SK)$;
$(M, \sigma) \leftarrow \mathcal{A}(PK, \sigma_1, \ldots, \sigma_q, st)$;
Return $\begin{cases} 1 \text{ if for } \forall i \in [1, q],\ M \neq M_i \\ \quad \text{and } \mathsf{Verify}(PK, M, \sigma) = 1, \\ 0 \text{ Otherwise.} \end{cases}$

We define

$$\mathbf{Adv}^{\text{EUF-CMA}}_{\mathsf{SIG},\mathcal{A}}(\lambda) = \Pr\left[\mathbf{Exp}^{\text{EUF-CMA}}_{\mathsf{SIG},\mathcal{A}}(\lambda) = \mathbf{1}\right] \text{ and } \mathbf{Adv}^{\text{EUF-wCMA}}_{\mathsf{SIG},\mathcal{A}}(\lambda) = \Pr\left[\mathbf{Exp}^{\text{EUF-wCMA}}_{\mathsf{SIG},\mathcal{A}}(\lambda) = \mathbf{1}\right].$$

Definition 1 *Let* SIG *be a signature scheme. If for any probabilistic polynomial-time adversary* $\mathcal{A}$, $\mathbf{Adv}^{EUF\text{-}CMA}_{\mathsf{SIG},\mathcal{A}}(\lambda)$ *(*$\mathbf{Adv}^{EUF\text{-}wCMA}_{\mathsf{SIG},\mathcal{A}}(\lambda)$*, respectively) is a negligible function in* λ*, we say that the signature scheme* SIG *is EUF-CMA secure (EUF-wCMA secure, respectively).*

Generic Transformation from EUF-wCMA secure scheme to EUF-CMA secure scheme. There is a well-known standard technique to transform from a EUF-wCMA secure signature scheme to a EUF-CMA secure scheme by using the chameleon hashes [25, 34, 5, 23, 24, 33, 4, 3]. Krawczyk and Rabin formalized the chameleon hash function and they provided a simple construction based on the DL assumption in the standard model [25]. Krawczyk-Rabin chameleon hash function applies to the Seo signatures, and so in the body of the paper, we prove only the EUF-wCMA security of the Seo signature scheme and its variant. Note that the generic transformation using the chameleon hashes is efficient; it is sufficient to add the description of the chameleon hashes (two group elements for the DL-based chameleon hashes) in the public key and one exponent in the signatures for the transformed EUF-CMA secure scheme. We omit the description of the chameleon hashes and the generic transformation from EUF-wCMA secure scheme to EUF-CMA secure scheme since those are quite standard and there are several good references (we suggest to see [24, 33]).

2.2 Background of Group and Assumption

In this paper, we use groups with bilinear pairings and the computational Diffie-Hellman (CDH) assumption in the bilinear group setting.

Definition 2 (Bilinear Groups) *The bilinear group generator* $\mathcal{G}$ *is an algorithm that takes as input a security parameter* λ *and outputs a bilinear group* $(p, \mathbb{G}, \mathbb{G}_t, e)$*, where* p *is a prime of size* 2λ*,* $\mathbb{G}$ *and* $\mathbb{G}_t$ *are cyclic groups of order* p*, and* e *is an efficiently computable bilinear map* $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ *satisfying the following two properties:*

1. *(Bilinearity) For all* $u, u', v, v' \in \mathbb{G}$*,* $e(uu', v) = e(u, v)e(u', v)$ *and* $e(u, vv') = e(u, v)e(u, v')$ *hold.*
2. *(Non-degeneracy) For a generator* g *of* $\mathbb{G}$*,* $e(g, g) \neq 1_{\mathbb{G}_t}$*, where* $1_{\mathbb{G}_t}$ *is identity element in* $\mathbb{G}_t$*.*

Definition 3 (Computational Diffie-Hellman Assumption) *Let* $\mathcal{G}$ *be a bilinear group generator. We say that* $\mathcal{G}$ *satisfies the CDH assumption if for any polynomial time probabilistic algorithm* $\mathcal{A}$ *the following advantage* $\mathbf{Adv}^{\mathbf{CDH}}_{\mathcal{A}}$ *is negligible function in the security parameter* λ*.*

$$\mathbf{Adv}^{\mathbf{CDH}}_{\mathcal{G},\mathcal{A}}(\lambda) = \Pr\left[\mathcal{A}(p, \mathbb{G}, \mathbb{G}_t, e, g, g^a, g^b) \to g^{ab} \middle| \mathcal{G}(\lambda) \to (p, \mathbb{G}, \mathbb{G}_t, e), a, b \xleftarrow{\$} \mathbb{Z}_p, g \xleftarrow{\$} \mathbb{G}\right].$$

3 Practical Signatures from Diffie-Hellman

We review the signature scheme in [33] and its analysis given in the same paper. In the signature scheme, a signature contains a random tag vector consisting of the same size tag components.

- $\mathsf{KeyGen}(\lambda)$: Run a bilinear group generator $\mathcal{G}$ and obtain $(p, \mathbb{G}, \mathbb{G}_t, e)$. Choose group elements v, u_1, ..., u_m, g_1, ..., g_k, h, g from $\mathbb{G}$ and an integer α from $\mathbb{Z}_p$ at random. Choose Q that is polynomial in λ. Define public key $PK = \{Q, v, u_1, \ldots, u_m, g_1, \ldots, g_k, h, g, g^\alpha\}$ and secret key $SK = \{\alpha\}$, and then publish PK and keep SK in secret.

- $\mathsf{Sign}(PK, M, SK)$: Randomly choose an integer $r \xleftarrow{\$} \mathbb{Z}_p$ and a tag vector $\overrightarrow{tag} = (tag_1, \ldots, tag_k) \xleftarrow{\$} [1, Q]^k$ (k times canonical product set). Compute $\sigma_1 = (v \prod_{i=1}^m u_i^{M^i})^\alpha (h \prod_{i=1}^k g_i^{tag_i})^r$ and $\sigma_2 = g^{-r}$. Here M^i means M to the power of i (mod p). Output the signature $\sigma = (\sigma_1, \sigma_2, \overrightarrow{tag})$.

- $\mathsf{Verify}(PK, M, \sigma)$: Parse σ to $(\sigma_1, \sigma_2, \overrightarrow{tag})$. If $\overrightarrow{tag} \notin [1, Q]^k$, then output 0. If the equality $e(\sigma_1, g)e(\sigma_2, h \prod_{i=1}^k g_i^{tag_i}) = e(g^\alpha, v \prod_{i=1}^m u_i^{M^i})$ holds, then output 1; otherwise, 0.

The public key size is $O(m + k)$. When Q^k is smaller than p, we can consider a tag vector is an element in $\mathbb{Z}_p$ so that the signature size is two group elements and one exponent. In Section 5, we will explain the tag-free variants; by adding constant factor in public keys, we can remove tag vectors from signatures so that we obtain shorter signatures.

3.1 Look at q-Bounded CMA Security in [33]

The above scheme satisfies the q-bounded CMA security, which is slightly weaker notion of the standard CMA security, where $Q = \Omega(q)$; q-bounded CMA security is the almost same as the standard CMA security except the fact that the maximum number of allowable signing is fixed at the parameter generating time. To achieve the CMA security, we first try to completely understand the original proof strategy for q-bound CMA security in [33]. Here we only focus on the weak unforgeability since there are standard generic transformations to the full unforgeability using the chameleon hashes (even in the q-bounded model).

Basically, the proof strategy (for constructing a simulator solving the CDH instance by using the adversary) is to efficiently guess a prefix of the tag vector of the forgery, say the target tag vector. Here, the word *efficiently* means that with non-negligible probability. We will explain the next step after the prefix-guessing later.

For efficient prefix-guessing, the simulator divides adversarial types according to a relation between the target tag vector and a set of tag vectors used in signing queries. Since a tag vector in each signature is chosen uniformly and independently, the simulator can choose tag vectors in advance, which will be used in signing queries so that the simulator can use the relation between the target tag vector and the random tag vectors. The essential idea behind the analysis in [33] is to apply a generalization [33] of the generalized birthday lemma [20, 22], which is given below.

Lemma 1 *[33, Lemma 2] Let $\mathcal{T}$ and $\mathcal{T}^i$ be sets $[1, Q]$ and $[1, Q]^i$, respectively. For $\overrightarrow{tag} \in \mathcal{T}^k$, let $\overrightarrow{tag}^{(i)} \in \mathcal{T}^i$ be the first i entries of $\overrightarrow{tag}$. For given $\{\overrightarrow{tag}_j\}_{j \in [1,q]}$, we define the set $S_i \subset \mathcal{T}^i$ as*

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

Then, we have an inequality $\Pr_{\overrightarrow{tag}_1,\ldots,\overrightarrow{tag}_q \xleftarrow{\$} \mathcal{T}^k}[|S_i| \geq \ell] < (\frac{q^{m+1}}{(m+1)!Q^{im}})^{\ell}$.

The above lemma with parameter selection of $Q \geq q$, $m = k = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ directly implies the following two inequalities with overwhelming probability, where the probability goes over the choice of random tag vectors that are used in signing queries.

1. $|S_1| < \lambda$ (Set $\ell = \lambda$ in Lemma 1)
2. $|S_k| < 1$ (Set $\ell = 1$ in Lemma 1)

(The right hand side of the inequality in Lemma 1 is negligible by the parameter selection of Q, m, and k.) From now, we set $Q = \Omega(q)$ and $m = k = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$. For $j \in [1, q]$, let $\overrightarrow{tag}_j \in \mathcal{T}^k$ be the random tag vector used in the j-th signature query. Let $\overrightarrow{tag}^* = (tag_1^*, \ldots, tag_k^*) \in \mathcal{T}^k$ be the target tag vector. The adversarial type is divided according to $\overrightarrow{tag}^*$ as follows.

$$\begin{aligned}
\text{Type-1} &: \overrightarrow{tag}^{*(1)} \notin S_1. \\
\text{Type-2} &: \overrightarrow{tag}^{*(1)} \in S_1, \text{ and } \overrightarrow{tag}^{*(2)} \notin S_2. \\
&\vdots \\
\text{Type-}i &: \overrightarrow{tag}^{*(i-1)} \in S_{i-1}, \text{ and } \overrightarrow{tag}^{*(i)} \notin S_i. \\
&\vdots \\
\text{Type-}n &: \overrightarrow{tag}^{*(n-1)} \in S_{n-1}, \text{ and } \overrightarrow{tag}^{*(n)} \notin S_n. \\
\text{Type-}(n+1) &: \overrightarrow{tag}^{*(n)} \in S_n.
\end{aligned}$$

Since $|S_k| < 1$, we know that there exists the above n that is less than k. We can check that every adversary should be only one type among the above $n + 1$ types.

Then, we are ready to explain how the simulator guesses a prefix of the target tag vector. First, the simulator guesses the adversarial type with at least probability $\frac{1}{k}$. Next, for the type-i adversary, the simulator can guess $\overrightarrow{tag}^{*(i)}$ with probability $\frac{1}{|S_{i-1}| \cdot |\mathcal{T}|}$; it guesses $\overrightarrow{tag}^{*(i-1)}$ with $\frac{1}{|S_{i-1}|} \geq \frac{1}{\lambda}$ (from the inequality $|S_{i-1}| \leq |S_1| < \lambda$) and tag_i^* with $\frac{1}{Q}$. Overall, the simulator can guess a prefix of the target tag vector with at least probability $\frac{1}{k\lambda Q}$

If the simulator correctly guess a prefix of the target tag vector, then it uses two techniques of selectively-secure signature scheme [5] and programmable hashes [22][5]; that is, the simulator first guess a prefix of the target tag vector $\overrightarrow{tag}^{*(i)}$, and then responds signing queries as follows.

$$\begin{cases} \text{for } \overrightarrow{tag}^{(i)} \neq \overrightarrow{tag}^{*(i)}, \text{ it uses the proof technique of selectively-secure scheme.} \\ \text{for } \overrightarrow{tag}^{(i)} = \overrightarrow{tag}^{*(i)}, \text{ it uses the programmability of the weak programmable hash functions.} \end{cases}$$

Since $\overrightarrow{tag}^{*(i)} \notin S_i$, there are at most m tag vectors same as $\overrightarrow{tag}^{*(i)}$ so that the simulator can response m signature queries with tag vector $\overrightarrow{tag}^{*(i)}$ using the programmability of $(v \prod_{i=1}^{m} u_i^{M^i})$.

Here, the reduction looses $k\lambda Q$ factor. From the above proof strategy, the following theorem is obtained.

[5] This explanation is very technical. Since our security analysis also follows the same line of proof strategy (except prefix-guessing), readers may find the details from our analysis in Section 4.

Theorem 1 *[33, Theorem 1] If there is an adversary outputting a forgery of the Seo signature scheme with ϵ probability in time T after issuing q signing queries, then we can construct a simulator that solves the CDH problem with ϵ' probability in time $T' \approx T$, where*

$$\epsilon' = \frac{1}{k\lambda Q}(\epsilon - \frac{q^{m+1}}{(m+1)!Q^{km}} - \frac{q}{p} - (\frac{q^{m+1}}{(m+1)!Q^m})^\lambda).$$

If we set the parameters by $Q = \Omega(q)$ and $m = k = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$, then $\epsilon' \sim \frac{\epsilon}{k\lambda Q}$. Since the public parameter Q should be larger than q, we can interpret the above theorem as the proof of the q-bounded CMA security.

Why q-Bounded CMA Security is Inevitable? Let us reconsider the analysis in [33]. In the analysis, the public parameter Q should be larger than q so that q should be fixed at the parameter generating time of the signature scheme; the simulator first guesses the adversarial type, say i-type, and then guesses $\overrightarrow{tag}^{*(i)}$ with at least $\frac{1}{\lambda Q}$; $\overrightarrow{tag}^{*(i-1)}$ with $\frac{1}{|S_{i-1}|} \geq \frac{1}{\lambda}$ (from the inequality $|S_{i-1}| \leq |S_1| < \lambda$) and tag_i^* with $\frac{1}{Q}$. Here, we need the inequality $|S_1| < \lambda$. In fact, S_1 is the set of $m+1$ collisions among q random integers from $[1, Q]$. Intuitively, q integers are chosen from the set $[1, Q]$ so that if $q > Q$, then we cannot expect any meaningful upper bound of $|S_1|$. Lemma 1 just shows this intuition with the exact quantity; $\Pr_{\overrightarrow{tag}_1,\ldots,\overrightarrow{tag}_q \xleftarrow{\$} \mathcal{T}^k}[|S_1| \geq \lambda] < (\frac{q^{m+1}}{(m+1)!Q^m})^\lambda$ and so if $q > Q$, we cannot expect the right hand side is even smaller than 1. Therefore, Q should be set as an integer larger than q.

One may think that if Q is sufficiently large, e.g., $Q = 2^\lambda$, then the Seo signature scheme satisfies the standard CMA security. However, in the analysis, the reduction algorithm loses Q factor (to guess tag_i^*) in comparison to the adversary's success probability so that Q should be a polynomial for the reduction algorithm being a polynomial time reduction.

In summary, Q should be set as a polynomial (for polynomial time reduction), but if the adversary obtains signatures of polynomial numbers, which is larger than Q, then the analysis does not guarantee any security. Therefore, the security analysis in [33] failed to show the standard CMA security.

4 New Analysis: Achieving CMA Security with Tighter Reduction

In this section, we aim at removing an undesirable relation between the size of the tag vector space and the bound of the maximum number q of signing queries of the Seo signature scheme.

First, we try to understand the essential reason why the Seo signature scheme can achieve sub-linear pubic key size (though it is secure against q-Bounded CMA attackers only). Even though the prefix-guessing proof strategy is already used to prove the CMA security of Waters signature scheme, only linear public key size could be achieved [24]. There is a big difference between the proof in [33] and the proof in [24][6]; The goal of the security proof (that is, the goal of the simulator) in [24] is to guess a prefix of the message that satisfies the following two conditions.

1. The message will be used in the forged signature.
2. The prefix of the message is different from the prefixes of all messages used in signing queries.

[6] In [24], there are two schemes. In this paper, we are interested in the (variant of) Waters signature scheme only.

(If the simulator has the prefix of the message satisfying the above two condition, then it can use the well-known technique for the selectively-secure scheme [5].)

On the contrary, the goal of the security proof in [33] is to guess a prefix of the target tag vector that satisfies the following two conditions.

1. The tag vector will be used in the forged signature.
2. The prefix of the target tag vector is different from the prefixes of all tag vectors used in signing queries, except for at most m-collisions; that is, there could be at most m tag vectors used in signing queries such that those prefixes (with the same length) are all equal to the prefix of the target vector.

(Again, if the simulator correctly guesses a prefix of the target tag vector, then it combines two techniques of selectively-secure signature scheme [5] and programmable hashes [22].) To make the second condition be meaningful, the necessary condition for the public parameters is that the length of the tag vectors should satisfy that there are no m-collisions among q random tag vectors. (If there are m-collisions among q tag vectors used in signing queries and the adversary uses such a tag vector as the target tag vector, then the second condition cannot be hold.); tag vectors are chosen from $[1, Q]^k$, and so if we set $Q = poly(\lambda)$ and m and k are *sublinear-but-strictly-increasing* function in λ, then the necessary condition is satisfied (by Lemma 1 with $i = k$, $\ell = 1$) so that we can achieve sublinear public key size, where $poly(\cdot)$ is an arbitrary polynomial. From here, we can see that this line of proof strategy in [33] is definitely unassociated with the relation between Q (public parameter) and q (the number of signing queries). Recall that the undesirable relation between Q and q was inevitable once the simulator follows the way to guess a prefix of the target tag vector in the proof in [33]. To prove the CMA security of the Seo signature scheme, we basically follow the same line of the proof strategy in [33], but we change the way to guess a prefix of the target tag vector. In the next subsection, we will explain our way of prefix-guessing in details.

As a result, we construct a reduction algorithm solving the CDH problem by using the weak CMA attacker of the Seo signature scheme, with $O(Qq)$ reduction loss. Here, the reason why our result leads the weak CMA security, unlike the original security proof, is that we can set Q and q be independent; more precisely, we can set Q be a fixed polynomial in λ (e.g., $Q = \lambda$.) Our security analysis is comparable with the proof in [33] in the sense that ours is as tighter as that of the original analysis, even better.

By combining with the generic transformation from EUF-wCMA secure signatures to EUF-CMA secure signatures, we can obtain the CMA secure signature scheme. Therefore, we concentrate on proving the EUF-wCMA security for the Seo signatures in this section.

4.1 *Alternative*, *Simpler* Prefix-Guessing

We follow the big picture of the original security analysis in [33]; that is, we also use the prefix-guessing strategy, and then we use the combination of the technique of selectively secure scheme and the programmable hashes for signing queries. In the original security proof, Lemma 1 is essentially used to guess a prefix of the target tag vector, but the restricted version (the generalized birthday lemma [20, 22]) of Lemma 1 is sufficient for our purpose. Thus, our security proof is much simpler than the original proof in [33].

We provide our main theorem that proves the weak CMA security of the Seo signature scheme.

Theorem 2 *If there is an adversary breaking the EUF-wCMA security of the Seo signature scheme with ϵ success probability and T running time, then we can construct the CDH problem solver $\mathcal{B}$ with ϵ' success probability and T' running time, where*

$$\epsilon' \geq \frac{m+1}{kqQ}(\epsilon - \frac{q^{m+1}}{(m+1)!Q^{km}} - \frac{q}{p}) \text{ and } T \approx T'.$$

Proof. We use the same notation as in Section 3.1. The goal of the proof is to construct a simulator $\mathcal{B}$ that solves the CDH problem with running an EUF-wCMA attacker $\mathcal{A}$ of the Seo signature scheme.

Simulation Description.

$\mathcal{B}$ first takes an uniform instance of the CDH problem, $(g, g^a, g^b) \in \mathbb{G}^3$, and the bilinear group description $(\mathbb{G}, \mathbb{G}_t, e)$ over which the CDH instance is defined. For the sake of simplicity, let $A = g^a$ and $B = g^b$. Next, $\mathcal{B}$ receives a list L of q messages $M_1, \ldots, M_q$ from $\mathcal{A}$.

Adversarial Types: For $i \in [1, q]$, $\mathcal{B}$ uniformly generates random tag vectors $\overrightarrow{tag}_i$ in advance that will be used in the i-th signing query on M_i. We define the adversarial type similar to Section 3.1; that is,

$$\begin{aligned}
\text{Type-1} &: \overrightarrow{tag}^{*(1)} \notin S_1. \\
\text{Type-2} &: \overrightarrow{tag}^{*(1)} \in S_1, \text{ and } \overrightarrow{tag}^{*(2)} \notin S_2. \\
&\vdots \\
\text{Type-}i &: \overrightarrow{tag}^{*(i-1)} \in S_{i-1}, \text{ and } \overrightarrow{tag}^{*(i)} \notin S_i. \\
&\vdots \\
\text{Type-}k &: \overrightarrow{tag}^{*(k-1)} \in S_{k-1}, \text{ and } \overrightarrow{tag}^{*(k)} \notin S_k. \\
\text{Type-}(k+1) &: \overrightarrow{tag}^{*(k)} \in S_k.
\end{aligned}$$

If $|S_k| \geq 1$, then the simulator aborts. For this case, we say that an event E_1 occurs. Otherwise (that is, $|S_k| < 1$), we know that there is an integer $n < k$ such that every adversary should be only one of $n + 1$ types. Note that we do not require the condition on the public parameter $Q \geq q$ as in the original analysis in [33].

(Alternative, Simpler) Prefix-Guessing: Then, the simulator guesses a prefix of the target tag vector as follows: it guesses adversary type, say type-i, with at least $\frac{1}{k}$. Next step of the simulator is to guess $\overrightarrow{tag}^{*(i)}$. To this end, the simulator guesses $\overrightarrow{tag}^{*(i-1)}$ (if $i > 1$ only) and tag_i^*, respectively. For $\overrightarrow{tag}^{*(i-1)}$, the simulator randomly chooses a tag vector in the set $\{\overrightarrow{tag}_j\}_{j\in[1,q]}$ and then sets its $(i-1)$-th prefix as the simulator's guess $\overrightarrow{tag}^{*(i-1)}$; that is, the simulator guesses $\overrightarrow{tag}^{*(i-1)}$ as a randomly chosen vector from $\{\overrightarrow{tag}_j^{(i-1)}\}_{j\in[1,q]}$. For tag_i^*, the simulator uniformly guesses it from its domain $[1, Q]$. We will argue that the simulator's guess of $\overrightarrow{tag}^{*(i)}$ is correct with at least $\frac{m+1}{qQ}$ later. In the following description, let us assume that $\mathcal{B}$'s guess for $\overrightarrow{tag}^{*(i)}$ is correct.

Let us briefly explain the remaining part of the simulation. The important property of the prefix-guessing $\mathcal{B}$ did above is that there are at most m tag vectors $\overrightarrow{tag}_j$'s such that their i-th prefixes are equal to the i-th prefix of the target tag vector, that is, $\overrightarrow{tag}_j^{(i)} = \overrightarrow{tag}^{*(i)}$ since for the type-i adversary, $\overrightarrow{tag}^{*(i)} \notin S_i$. (For the type-$(n+1)$ adversary, $S_{n+1} = \emptyset$.) By using this property, we can simulate public key and all signatures on M_i's; for signatures with tag vectors having prefixes

different from $\overrightarrow{tag}^{*(i)}$, we will use the technique for selectively-secure signature scheme (e.g., Boneh-Boyen signatures [6]), and for signatures with tag vectors (at most m) having the same prefix as $\overrightarrow{tag}^{*(i)}$, we will use the (weak) programmable hashes [22] in simulation.

KeyGen: We know that the i-prefix of the target tag vector, $\overrightarrow{tag}^{*(i)}$ is not contained in S_i, and so there exist at most m distinct tag vectors whose i-prefix is equal to $\overrightarrow{tag}^{*(i)}$ among q tag vectors for signing queries. Let I be the set of indexes for such tag vectors. Then, $|I| \leq m$. We first define a polynomial $f(X)$ having as roots messages M_i for $i \in I$; that is, $f(X) := \prod_{i \in I}(X - M_i)$. If I is an empty set, we just define $f(X) = 1$. We can rewrite $f(X)$ by $\sum_{i=0}^{m} x_i X^i$ for some coefficients $x_0, \ldots, x_m \in \mathbb{Z}_p$. Note that $x_i = 0$ for $i > |I|$. Next, $\mathcal{B}$ uniformly chooses integers $y_0, \ldots, y_m, z_0, \ldots, z_\ell, w_0, \ldots, w_k \xleftarrow{\$} \mathbb{Z}_p$. Lastly, $\mathcal{B}$ generates a public key $PK = \{v, u_1, \ldots, u_m, g_1, \ldots, g_k, h, g, g^\alpha\}$ as follows.

$v = A^{x_0} g^{y_0}$,	$u_j = A^{x_j} g^{y_j}$ for $j \in [1, m]$,	$h = A^{-\sum_{j=1}^{i} tag_j^* z_j} g^{w_0}$
$g_j = \begin{cases} A^{z_j} g^{w_j} & \text{for } j \in [1, i] \\ g^{w_j} & \text{for } j \in [i+1, k] \end{cases}$,	$g = g$,	$g^\alpha = B$,

where $\overrightarrow{tag}^{*(i)} = (tag_1^*, \ldots, tag_i^*)$. Then, b is the corresponding secret key and is unknown to $\mathcal{B}$.

Sign: $\mathcal{B}$ generates signatures on $M_1, \ldots, M_q$ as follows. For the j-th signing query, $\mathcal{B}$ first checks whether $j \in I$, and then $\mathcal{B}$ separately behaves as follows:

If $j \in [1, q] \setminus I$, then $\mathcal{B}$ checks whether the equality $\sum_{t=1}^{i}(tag_{jt} - tag_t^*)z_t = 0$ holds, where $\overrightarrow{tag}_j = (tag_{j1}, \ldots, tag_{jk})$. (Recall that $\mathcal{B}$ already chose all tag vectors $\overrightarrow{tag}_1, \ldots, \overrightarrow{tag}_q$ in advance before the *prefix-guessing* phase.) If the equality holds, then $\mathcal{B}$ aborts the simulation and outputs a random element. For this case, we say that an event E_2 occurs. Otherwise, $\mathcal{B}$ chooses a random integer $r' \xleftarrow{\$} \mathbb{Z}_p$ and computes a signature as follows.

$$\sigma_{j1} = B^{(\sum_{t=0}^{m} y_t M_j^t) - (w_0 + \sum_{t=1}^{k} tag_{jt} w_t) \frac{(\sum_{t=0}^{m} x_t M_j^t)}{(\sum_{t=1}^{i} (tag_{jt} - tag_t^*) z_t)}} \cdot \left(A^{(\sum_{t=1}^{i} (tag_{jt} - tag_t^*) z_t)} g^{w_0 + \sum_{t=1}^{k} tag_{jt} w_t}\right)^{r'}$$

$$\text{and } \sigma_{j2} = B^{\frac{(\sum_{t=0}^{m} x_t M_j^t)}{(\sum_{t=1}^{i} (tag_{jt} - tag_t^*) z_t)}} \cdot g^{-r'}.$$

If $j \in I$, $\mathcal{B}$ chooses $r \xleftarrow{\$} \mathbb{Z}_p$ and computes a signature as follows.

$$\sigma_{j1} = B^{(\sum_{t=0}^{m} y_t M_j^t)} (g^{w_0 + \sum_{t=1}^{k} tag_{jt} w_t})^r \text{ and } \sigma_{j2} = g^{-r}.$$

Lastly, $\mathcal{B}$ defines the j-th signature σ_j on M_j by $(\sigma_{j1}, \sigma_{j2}, \overrightarrow{tag}_j)$.

Response: $\mathcal{B}$ sends $\mathcal{A}$ the public key $PK = (v, u_1, \ldots, u_m, h, g_1, \ldots, g_k, g, g^\alpha)$ along with signatures $\sigma_1, \ldots, \sigma_q$.

Extraction from Forgery: At the end of interaction, $\mathcal{B}$ receives a message M^* along with a forgery $\sigma^* = (\sigma_1^*, \sigma_2^*, \overrightarrow{tag}^*)$ on M^* from $\mathcal{A}$ such that $M^* \notin L$. If $\mathsf{Verify}(PK, M^*, \sigma^*) = 0$, $\mathcal{B}$ aborts. Otherwise, $f(M^*) = \sum_{i=0}^{m} x_i (M^*)^i \neq 0$ since $M^* \notin L$ and all f's roots are contained in L. Finally, $\mathcal{B}$ outputs $(\sigma_1^* \cdot B^{-\sum_{t=0}^{m} y_t (M^*)^t} \cdot (\sigma_2^*)^{w_0 + \sum_{t=1}^{k} tag_t^* w_t})^{\frac{1}{f(M^*)}}$ as the solution of the CDH instance (g, g^a, g^b).

Analysis of the Reduction Algorithm $\mathcal{B}$.

Simulation Halt. By setting $i = k$ and $\ell = 1$ in Lemma 1, we obtain that the probability $\Pr[E_1] = \Pr_{\overrightarrow{tag}_1,\ldots,\overrightarrow{tag}_q \xleftarrow{\$} \mathcal{T}^k}[|S_k| \geq 1]$ is less than $\frac{q^{m+1}}{(m+1)!Q^{km}}$.[7] For $\Pr[E_2]$, we obtain that $\Pr[E_2] = \Pr_{z_t}[\forall j \in [1,q], \Sigma_{t=1}^{i}(tag_{jt} - tag_t^*)z_t = 0] < \frac{q}{p}$ from the union bound. $\Pr[E_2]$ is independent from the adversarial behaviours since all z_t's are hidden from the adversarial view, and also independent from the bound for the simulator's guess, which is given below.

Simulator's guess. We show that the simulator's guess is correct with at least $\frac{m+1}{kqQ}$ probability. The simulator's prefix-guessing consists of three *independent* steps. First, it guesses the adversarial types with $\frac{1}{k}$ probability; since this guess is completely independent from all other process of the simulator and is also hidden from the adversarial view, it is correct with $\frac{1}{k}$ probability. Next, we consider the conditional probability that the simulator's guess of $\overrightarrow{tag}^{*(i-1)}$ is correct once its guess of the adversarial type is correct as i. If $i > 1$, then $\overrightarrow{tag}^{*(i-1)} \in S_{i-1}$ so that there are at least $m+1$ tag vectors in $\{\overrightarrow{tag}_j\}_{j\in[1,q]}$ such that $\overrightarrow{tag}_j^{(i-1)} = \overrightarrow{tag}^{*(i-1)}$. Hence, the probability that the simulator chooses such a tag vector $\overrightarrow{tag}_j$ satisfying the equality $\overrightarrow{tag}_j^{(i-1)} = \overrightarrow{tag}^{*(i-1)}$ is more than $\frac{m+1}{q}$. Finally, the simulator can guess tag_i^* with $\frac{1}{Q}$. Since all probabilities are independent, the overall probability to correctly guess the prefix of the target tag vector is at least $\frac{m+1}{kqQ}$.

Distribution of simulation. We show that the simulated transcript (public key and signing queries) between $\mathcal{A}$ and $\mathcal{B}$ are indistinguishable from the real transcript on the condition that the simulator does not abort and its guess is correct; since $y_0, \ldots, y_m$, and $w_0, \ldots, w_k$ are uniformly chosen from $\mathbb{Z}_p$ and the CDH instance is also uniformly generated, the public key simulated by $\mathcal{B}$ is identical to those of the output of the KeyGen algorithm. Next, we consider the distribution of simulated signatures for signing queries. The tag vectors chosen before the *prefix-guessing* phase are distributed uniformly and independently, so that we focus on the other parts in signatures except for tag vectors.

For $j \in [1,q] \setminus I$, we argue that the randomness r used in the j-th signature query is distributed as if $r = -\frac{(\sum_{t=0}^{m} x_t M_j^t b)}{(\sum_{t=1}^{i}(tag_{jt} - tag_t^*)z_t)} + r'$;

$$\begin{aligned}
\sigma_{j1} &= B^{(\sum_{t=0}^{m} y_t M_j^t) - (w_0 + \sum_{t=1}^{k} tag_{jt} w_t)\frac{(\sum_{t=0}^{m} x_t M_j^t)}{(\sum_{t=1}^{i}(tag_{jt} - tag_t^*)z_t)}} \cdot \left(A^{(\sum_{t=1}^{i}(tag_{jt} - tag_t^*)z_t)} g^{w_0 + \sum_{t=1}^{k} tag_{jt} w_t}\right)^{r'} \\
&= B^{(\sum_{t=0}^{m} y_t M_j^t)} (g^{ab})^{\sum_{t=0}^{m} x_t M_j^t} \cdot \left(A^{\sum_{t=1}^{i}(tag_{jt} - tag_t^*)z_t} g^{w_0 + \sum_{t=1}^{k} tag_{jt} w_t}\right)^{\frac{(-\sum_{t=0}^{m} x_t M_j^t b)}{(\sum_{t=1}^{i}(tag_{jt} - tag_t^*)z_t)}} \\
&\quad \cdot \left(A^{\sum_{t=1}^{i}(tag_{jt} - tag_t^*)z_t} g^{w_0 + \sum_{t=1}^{k} tag_{jt} w_t}\right)^{r'} \\
&= \left(\prod_{t=0}^{m} (A^{x_t} g^{y_t})^{M_j^t}\right)^b \cdot \left(A^{\sum_{t=1}^{i}(tag_{jt} - tag_t^*)z_t} g^{w_0 + \sum_{t=1}^{k} tag_{jt} w_t}\right)^r \\
&= \left(\prod_{t=0}^{m} (A^{x_t} g^{y_t})^{M_j^t}\right)^b \cdot \left(A^{-\sum_{t=1}^{i} tag_t^* z_t} g^{w_0} \cdot \prod_{t=1}^{i} (A^{z_t} g^{w_t})^{tag_{jt}} \cdot \prod_{t=i+1}^{k} (g^{w_t})^{tag_{jt}}\right)^r \\
&= \left(v \prod_{t=1}^{m} u_t^{M_j^t}\right)^b \left(h \prod_{t=1}^{k} g_t^{tag_{jt}}\right)^r
\end{aligned}$$

and $\sigma_{j2} = B^{\frac{(\sum_{t=0}^{m} x_t M_j^t)}{(\sum_{t=1}^{i}(tag_{jt} - tag_t^*)z_t)}} \cdot g^{-r'} = g^{-r}$.

[7] Note that here we can use the generalized birthday lemma [20, 22] instead of Lemma 1. Since we already stated Lemma 1, here we use Lemma 1. But, we note that our proof does not require a full advantage of Lemma 1 like the proof in [33].

We can see that r is uniformly distributed according to r' since r' is uniformly and independently chosen from $\mathbb{Z}_p$. Consequently, we showed that the simulated distribution of σ_{j1} and σ_{j2} is identical to that of the output of Sign algorithm.

For $j \in I$, we argue that the simulated signature is distributed with the randomness r;

$$\begin{aligned}
\sigma_{j1} &= B^{(\sum_{t=0}^{m} y_t M_j^t)}(g^{w_0+\sum_{t=1}^{k} tag_{jt} w_t})^r \\
&= B^{(\sum_{t=0}^{m} y_t M_j^t)}(g^{ab})^{\sum_{t=0}^{m} x_t M_j^t} \cdot (A^{\sum_{t=1}^{i}(tag_{jt}-tag_t^*)z_t} g^{w_0+\sum_{t=1}^{k} tag_{jt} w_t})^r \\
&= (\prod_{t=0}^{m}(A^{x_t} g^{y_t})^{M_j^t})^b \cdot (A^{-\sum_{t=1}^{i} tag_t^* z_t} g^{w_0} \prod_{t=1}^{i}(A^{z_t} g^{w_t})^{tag_{jt}} \prod_{i=i+1}^{k}(g^{w_t})^{tag_{jt}})^r \\
&= (v \prod_{t=1}^{m} u_t^{M_j^t})^b (h \prod_{t=1}^{k} g_t^{tag_{jt}})^r.
\end{aligned}$$

In the second equality, we used the fact that $\sum_{t=0}^{m} x_t M_j^t = 0$ and $tag_{jt} = tag_t^*$ for $j \in I$ and $t \in [1, i]$. We know that $\sigma_{j2} = g^{-r}$. Since r is an uniformly chosen integer, the simulated distribution of σ_{j1} and σ_{j2} is identical to that of the output of Sign algorithm.

CDH Solution Extraction. Finally, we show that the CDH solution $\mathcal{B}$ outputs is valid on the condition that two event E_1 and $_2$ do not occur and the simulator's guess is correct. On the same condition, we already showed that the simulated transcript is identical to those of the real transcript of EUF-wCMA security game. If $\mathcal{A}$ outputs a valid forgery $(\sigma_1^*, \sigma_2^*, \overrightarrow{tag}^*)$, then it satisfies the verification equation so that it is of the form $\sigma_1^* = (v \prod_{t=1}^{m} u_t^{M^{*t}})^{\alpha}(h \prod_{t=1}^{k} g_t^{tag_t^*})^r$ and $\sigma_2^* = g^{-r}$ for some r. From the simulator's public key setting, we know that σ_1^* is equal to $(g^{ab})^{f(M^*)}(g^b)^{\sum_{t=0}^{m} y_t (M^*)^t}(g^r)^{w_0+\sum_{t=1}^{k} tag_t^* w_t}$ so that $\mathcal{B}$ outputs

$$(\sigma_1^* \cdot B^{-\sum_{t=0}^{m} y_t (M^*)^t} \cdot (\sigma_2^*)^{w_0+\sum_{t=1}^{k} tag_t^* w_t})^{\frac{1}{f(M^*)}} = g^{ab}.$$

Success Probability. For the success probability $\Pr[S_{\mathcal{A}}] = \epsilon$, we can bound $\mathcal{B}$'s success probability $\Pr[S_{\mathcal{B}}]$ as follows.

$$\begin{aligned}
\Pr[S_{\mathcal{B}}] &= \tfrac{m+1}{kqQ} \Pr[S_{\mathcal{A}} \wedge \neg E_1 \wedge \neg E_2] \\
&\geq \tfrac{m+1}{kqQ}(\Pr[S_{\mathcal{A}}] - \Pr[E_1 \vee E_2]) \\
&= \tfrac{m+1}{kqQ}(\epsilon - \tfrac{q^{m+1}}{(m+1)!Q^{km}} - \tfrac{q}{p})
\end{aligned}$$

□

4.2 Parameter Selection

If we choose $Q = poly(\lambda)$ and $m = k = \omega(1)$, where *poly* is an arbitrary polynomial, then we can show that the simulator's success probability

$$\frac{m+1}{kqQ}(\epsilon - \frac{q^{m+1}}{(m+1)!Q^{km}} - \frac{q}{p})$$

is non-negligible, where ϵ is non-negligible and q is the maximum number of allowed signing queries, which is polynomial in the security parameter. For example, if $Q = \lambda$, $m = k = \log\log\lambda$, then $\frac{q^{m+1}}{Q^{km}} = \frac{q^{1+\log\log\lambda}}{\lambda^{(\log\log\lambda)^2}}$ is clearly a negligible function in λ, under the condition that q is polynomial in λ. Therefore, we obtain the asymptotic result $\Pr[S_{\mathcal{B}}] \sim \frac{1}{q\lambda}\epsilon$, with the parameter selection $m = k = \omega(1)$ and $Q = \lambda$.

Although $\frac{q^{m+1}}{(m+1)!Q^{km}}$ could be negligible with $m = k = \omega(1)$ and $Q = poly(\lambda)$, it is not exponentially small. In practice, one may want to consider sub-exponential or exponential-time adversaries. To this end, we can choose $m = k = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ and $Q = \lambda$ so that $\frac{q^{m+1}}{(m+1)!Q^{km}}$ is exponentially small in λ.[8] In this case, we still have sublinear public key $\Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ with preserving tighter reduction.

In particular, for concrete security parameter we can yield reasonably short public keys satisfying the above inequality; e.g., $\lambda = 80$, we can set $m = k = 15$ and $Q = \lambda$ so that the probability $\frac{q^{m+1}}{(m+1)!Q^{km}}$ is much less than $\frac{1}{2^{80}}$ for any $q \leq 2^{80}$. Therefore, we have an EUF-CMA secure signature scheme with public key size of 34 group elements for 80-bit security, which is much shorter than the public key size (164 group elements) of the Waters signature scheme for the same security parameter, where the reduction loss of both schemes is the same $O(\lambda q)$.

5 Extensions

In [33], two extensions are considered. First, a tag-free scheme using a pseudorandom function PRF is given[9]; the signer chooses a random key K for the pseudorandom function and publishes it along with other public parameters of signature scheme. Whenever the signer needs to generate a tag vector $\overrightarrow{tag}$ for signing a message M, she define $\overrightarrow{tag} = PRF_K(M)$. The main advantage of the tag-free scheme is shorter signatures since the verifier can generate tag vectors from messages and public parameters so that the signer can remove tag vectors from signatures. This technique applies to our scheme, and so we can obtain shorter signatures; that is, the EUF-wCMA secure signatures consist of two group elements and the EUF-CMA secure signature consist of two group elements and one exponent. Second, it is shown that the Seo signature scheme could be constructed and proven secure in asymmetric bilinear group setting. We note that such an extension also apply to our scheme.

References

1. M. Bellare and S. Micali. How to sign given any trapdoor function. In *CRYPTO 1988*, volume 403 of *LNCS*, pages 200–215. Springer, 1988.
2. M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with rsa and rabin. In *EUROCRYPT 1996*, volume 1070 of *LNCS*, pages 399–416. Springer, 1996.
3. F. Böhl, D. Hofheinz, T. Jager, J. Koch, J. H. Seo, and C. Striecks. Practical signatures from standard assumptions. In *EUROCRYPT2013*, volume 7881 of *LNCS*. Springer, 2013.
4. F. Böhl, D. Hofheinz, T. Jager, J. Koch, and C. Striecks. Confined guessing: New signatures from standard assumptions. In *Cryptology ePrint Archive (http://eprint.iacr.org/2013/171)*, 2013.
5. D. Boneh and X. Boyen. Short signatures without random oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 382–400. Springer, 2004.
6. D. Boneh and X. Boyen. Efficient selective identity-based encryption without random oracles. *J. Cryptology*, 24(4):659–693, 2011.
7. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *Journal of Cryptology*, volume 17, pages 297–319. Springer, 2004.
8. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilnear maps. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, 2004.
9. R. Cramer and I. Damgård. Secure signature schemes based on interactive protocols. In *CRYPTO 1995*, volume 936 of *LNCS*, pages 297–310. Springer, 1995.

[8] $\frac{q}{p}$ is already exponentially small in λ even when $q = 2^\lambda$ since p is usually set to be larger than $2^{2\lambda}$.

[9] Such a technique using PRF is previously used in the RSA signatures for compressing random prime numbers [23, 24, 20, 36].

10. R. Cramer and I. Damgård. New generation of secure and practical rsa-based signatures. In *CRYPTO 1996*, volume 1109 of *LNCS*, pages 173–185. Springer, 1996.
11. R. Cramer and V. Shoup. Signature schemes based on the strong rsa assumption. In *ACM CCS 1999*, pages 46–51. ACM Press, 1999.
12. C. Dwork and M. Naor. An efficient existentially unforgeable signature scheme and its applications. In *CRYPTO 1994*, volume 839 of *LNCS*, pages 234–246. Springer, 1994.
13. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakely and D. Chaum, editors, *CRYPTO 1984*, volume 196 of *LNCS*, pages 10–18. Springer-Verlag, 1984.
14. M. Fischlin. The cramer-shoup strong-rsa signature scheme revisited. In *PKC 2003*, volume 2567 of *LNCS*, pages 116–129. Springer, 2003.
15. R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In *EUROCRYPT 1999*, volume 1592 of *LNCS*, pages 123–139. Springer, 1999.
16. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *ACM STOC 2008*, pages 197–206, 2008.
17. E.-J. Goh, S. Jarecki, J. Katz, and N. Wang. Efficient signature schemes with tight reductions to the diffie-hellman problems. In *Journal of Cryptology*, volume 20, pages 493–514. Springer, 2007.
18. O. Goldreich. Two remarks concerning the goldwasser-micali-rivest signature scheme. In *CRYPTO 1986*, volume 263 of *LNCS*, pages 104–110. Springer, 1987.
19. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. In *SIAM J. Comput.*, volume 17, pages 281–308, 1988.
20. D. Hofheinz, T. Jager, and E. Kiltz. Short signatures from weaker assumptions. In *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 647–666. Springer, 2011.
21. D. Hofheinz, T. Jager, and E. Knapp. Waters signatures with optimal security reduction. In *PKC 2012*, volume 7293 of *LNCS*, pages 66–83. Springer, 2012.
22. D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. In *Journal of Cryptology*, volume 25, pages 484–527. Springer, 2012.
23. S. Hohenberger and B. Waters. Realizing hash-and-sign signatures under standard assumptions. In *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 333–350. Springer, 2009.
24. S. Hohenberger and B. Waters. Short and stateless signatures from the rsa assumption. In *CRYPTO 2009*, volume 5677 of *LNCS*, pages 654–670. Springer, 2009.
25. H. Krawczyk and T. Rabin. Chameleon signatures. In *NDSS 2000*. The Internet Society, 2000.
26. L. Lamport. Constructing digital signatures from a one-way function. In *Technical Report SRI-CSL-98*. SRI International Computer Science Laboratory, 1979.
27. R. C. Merkle. A certified digital signature. In *CRYPTO 1989*, volume 435 of *LNCS*, pages 300–317. Springer, 1990.
28. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*. ACM Press, 1989.
29. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO 1992*, volume 740 of *LNCS*, pages 31–53. Springer, 1992.
30. T. Okamoto. Efficient blind and partially blind signatures without random oracles. In *TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, 2006.
31. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, 1990.
32. C. P. Schnorr. Efficient signature generation for smart cards. In *Journal of Cryptology*, volume 4, pages 239–252. Springer, 1991.
33. J. H. Seo. Short signature from Diffie-Hellman: Realizing short public key. In *Cryptology ePrint Archive (http://eprint.iacr.org/2012/480)*, 2012.
34. A. Shamir and Y. Tauman. Improved online/offline signature schemes. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 355–367. Springer, 2001.
35. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer-Verlag, 2005.
36. S. Yamada, G. Hanaoka, and N. Kunihiro. Space efficient signature schemes from the RSA assumption. In *PKC 2012*, volume 7293 of *LNCS*, pages 102–119. Springer, 2012.