Short Signatures From Diffie-Hellman: Realizing Short Public Key

Jae Hong Seo

National Institute of Information and Communications Technology,
4-2-1, Nukui-kitamachi, Koganei, Tokyo, 184-8795, Japan
jaehong@nict.go.jp

Abstract. In EUROCRYPT 2005, Waters [42] proposed a signature scheme based on the computational Diffie-Hellman (DH) assumption without random oracles. His scheme is the first and sole signature scheme in the category of (hash-and-sign) signature schemes secure under the DH assumption in the standard model and has also been applied to the design of numerous protocols in the various cryptographic areas. However, the Waters signature scheme suffered from a large public key of $\Theta(\lambda)$ group elements, where λ is the security parameter. Realizing standard model DH-based signature scheme, in which both the signature and the public key are short, has been an open problem.

We propose short signatures from the DH assumption, which has a sublinear size public key. More precisely, our proposal produces a public key of $\Theta(\sqrt{\frac{\lambda}{\log\lambda}})$ group elements. Our construction is inspired from two techniques for short signatures such as using programmable hashes [26] and using tags [27]. From two previous techniques, we first derive a signature scheme with a somewhat short public key of $\Theta(\frac{\lambda}{\log\lambda})$, and then we developed a new technique for *asymmetric trade* between the public key size and the signature size. In particular, by adding one field element in each signature, we can reduce the public key size to $O(\sqrt{\frac{\lambda}{\log\lambda}})$ group elements, so that the resulting signature size is two group elements and two field elements.

We also propose a variant by applying a technique for compressing tag vectors so that the resulting signatures has a shorter signature size (two group elements and one field element) by 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). Note that we limit ourselves to dealing with only polynomial-time reductions in all security proofs.

1 Introduction

Digital signature scheme is one of fundamental primitives of modern cryptography and is used in many cryptographic protocols as an important building block. Many recent studies have constructed efficient digital signature schemes secure under the widely accepted cryptographic assumptions such as the Diffie-Hellman (DH) and RSA assumptions. Most of them follow a hash-and-sign paradigm (rather than a tree-based approach [18, 15]) to obtain efficient signature scheme, in particular short signatures; collision-resistant hash functions mapping from an arbitrarily long message to a short bit-string are first used, then the hashed message is signed onto. However, most (hash-and-sign) short signature schemes require strong assumptions (e.g., random oracles [20, 39, 36, 2, 7, 23, 22], strong RSA assumptions [21, 16, 19], q-strong type assumptions [5, 37, 25, 26], or LRSW assumption [9]), inefficient signing/verification processes [28, 25], or long public parameters [42].

Our Results. We propose a *practical* signature scheme, in which the both public key and signatures are short. We also explain the security analysis of the proposed signature scheme such that the proposed construction is proven secure under the DH assumption in the standard model. Note that realizing standard model DH-based signatures, in which both the signature and the public key are short, has been an open problem posed by Hohenberger and Waters in CRYPTO 2009 [28].

Security Parameter λ		$q = 2^{30}$				$q = 2^{40}$			
		m	k	PK size	Sig. size	m	k	PK size	Sig. size
80	w/o tag comp.	7	2	$12\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$	$2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$	8	2	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$	$2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$
	w/ tag comp.	7	2	$12\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \|K\|$	$2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$	8	2	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \|K\|$	$2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$
128	w/o tag comp.	7	2	$12\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$	$2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$	8	2	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$	$2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$
	w/ tag comp.	7	2	$12\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \|K\|$	$2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$	8	2	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \|K\|$	$2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$
192	w/o tag comp.	7	2	$12\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$	$2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$	8	2	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$	$2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$
	w/ tag comp.	7	2	$12\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \|K\|$	$2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$	8	2	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \|K\|$	$2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$
256	w/o tag comp.	7	3	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$	$2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$	8	2	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}}$	$2\tau_{\mathbb{G}_1} + 2\tau_{\mathbb{Z}_p}$
	w/ tag comp.	7	3	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \|K\|$	$2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$	8	2	$13\tau_{\mathbb{G}_2} + \tau_{\mathbb{G}_1} + 2\tau_{\mathbb{G}_{ch}} + \|K\|$	$2\tau_{\mathbb{G}_1} + \tau_{\mathbb{Z}_p}$

Table 1. Practical parameters for (fully secure) signature scheme: m, k, and Q are parameters used in the description of the main construction for signatures scheme, and q is the maximum of the number of signatures that an adversary can get. In this table, we set $Q = 2^3 q$, so that the reduction loss during the simulation is at most $96\lambda q$, which is comparable to that of Waters signature scheme in [42]. 'w/o tag comp.' means the signature scheme without applying tag compression technique in the figure 3 and 'w/ tag comp.' means the signature scheme with tag compression technique in the figure 4 and figure 5. '$\tau_{\mathbb{G}_1}$' 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.

Prior to the proposed signature scheme, there was the sole standard-model stateless DH-based signature scheme, called the Waters signature scheme [42]. The Waters signatures consist of two group elements, but suffer from a large public key of $\Theta(\lambda)$ group elements, where λ is the security parameter, e.g., public key is 164 group elements when $\lambda = 80$ and 516 group elements when $\lambda = 256$ (that is, $2\lambda + 4$). The proposed signatures consist of two group elements and two field elements (but, if we apply tag compression technique, we can reduce the signature size to two group elements and one field elements) and the public key is $\Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ group elements (regardless of applying tag compression technique), e.g., public key is at most 16 group elements when $\lambda \in [80, 256]$ and $q \in \{2^{30}, 2^{40}\}$, where q is the maximum of the number of signing queries issued by adversary and the reduction loss of the proposed scheme is comparable with that of Waters signatures in [42]. Table 1 shows efficiency of our constructions. See Section 4.3 for detailed efficiency analysis.

The signer of the proposed signature scheme does not need to maintain certain states [27] so that our construction is a stateless signature scheme. Naccache [34] proposed the variant of Waters signatures that offers a trade-off between the public key size and the concrete security level, but the asymptotic behavior is the same as the Waters signatures. Note that the variant, in which Naccache's trade-off is applied, requires a super-polynomial-time reduction to obtain (asymptotically) the same public key size as ours. In this paper, we limit ourselves to dealing with only polynomial-time reductions in the security proof.

Our Strategy. The proposed signature scheme is inspired from two techniques of obtaining DH-based short signatures, which are weakly secure under the DH assumption in the standard model. In contrast to the full security model, the adversary should send all messages for signing queries before receiving the public key in the weak security model. There is an efficient generic transformation of the weakly secure signatures to fully secure signatures using the chameleon hashes [30], which is secure under the Discrete Logarithm (DL) assumption. Since the description of the DL-based chameleon hashes consists of two group elements and the DL assumption is weaker than the DH assumption, the generic transformation using chameleon hashes does not inflict a loss in the security and the asymptotic efficiency of the proposed signature scheme.

We start with exploring two different techniques of obtaining short signatures. One technique is to use so-called programmable hash functions [26] so that one can obtain weakly secure (stateless) DH-based signatures with a large public key. The other technique is to use tags and so in the security proof the simulator can restrict the adversaries to forging in the polynomial number of tags by maintaining states so that one can obtain stateful DH-based signatures with both a short public key and signatures [27]. The naive combination of these two different techniques allows us to obtain short and stateless DH-based signatures with a *somewhat* short public key of $\Theta(\frac{\lambda}{\log \lambda})$ group elements. We then developed a new technique for *asymmetric trade* between the public key size and the signature size. In particular, by adding one field element in each signature, we can reduce the public key size to $O(\sqrt{\frac{\lambda}{\log \lambda}})$ group elements, so that the resulting signature size is two group elements and two field elements. Note that we limit ourselves to dealing with only polynomial-time reductions in the security proof. Our proof technique is of independent interest.

Furthermore, if we apply a technique for compressing tag vectors, which is used in RSA-based signatures [27, 28, 25, 43], then we can reduce the signature size (that is, the signature consists of two group elements and one field element) by 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); 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 weak security model) we can use the fact that the distribution of tag vectors is indistinguishable from the uniform distribution.

Applications. Digital signature schemes can be used in many applications such as electronic commerce [29] to provide authentication, integrity, and non-repudiation properties, that is, digital signatures can be used to authenticate electronic information, to guarantee the integrity of message transmission, and to preclude the signer from denying the signed message.

Moreover, techniques used to design digital signatures can be used to build 'functional' signature schemes such as blind signatures [12], group signatures [13], ring signatures [38], and aggregate signatures [6].

Related Works (Hash-and-Sign Signatures). By relying on the random oracle heuristic, many constructions for efficient signature schemes in several settings have been proposed (e.g., DL setting [20, 39, 36, 7, 23], RSA setting [2], and Lattice setting [22]). However, there have been some studies showing the limitations of the random oracles [10, 17, 31].

In the DL setting, Boneh and Boyen [5] proposed the first signature scheme in the standard model where signature size is comparable with that of BLS signature scheme [7] in the random oracle model. Okamoto [37] proposed a signature scheme which is more effective in many applications such as blind signatures, group signatures, and anonymous credentials. Recently, Hofheinz, Jager, and Kiltz [25, 26] proposed short signatures using programmable hash functions where the signature size is a bit shorter than previous schemes. However, the security of all these signature schemes are proven under non-static q-type assumptions. The size of the problem instance of q-type assumptions is (linearly) increased according to the number of signing queries. There are analyses for the q-type assumptions [8, 14]. Camenisch and Lysyanskaya [9] proposed a signature scheme which can be used to construct efficient group signatures, identity escrow schemes, and anonymous credential systems. They proved security of their signature scheme under interactive assumption, called the LRSW assumption [32]. Interactive assumptions are non-falsifiable and there is a criticism for non-falsifiable assumptions [35].

To the best of our knowledge, there are only two signature scheme based on the standard DH assumption in the standard model [42, 27]. However, the signer of the signature scheme in [27] needs to maintain certain states, i.e., stateful signature scheme. Therefore, there is only one construction for stateless signatures [42] that is proven secure under the standard DH assumption in the standard model. However, the signature scheme in [42] has a large public key $\Theta(\lambda)$ as compared with all aforementioned signature schemes based on the strong assumptions.

In the RSA setting, the first standard model construction was developed by Gennaro, Halevi, and Rabin [21]. Subsequently, Cramer and Shoup [16] and Fischlin [19] proposed more efficient signature scheme. However, these schemes are proven secure under the strong RSA assumption. Hohenberger and Waters proposed the first hash-and-sign signatures [27] from the RSA assumption. However, their scheme requires the signer to maintain states, i.e., stateful signature scheme. In the same year they proposed first stateless RSA-based signatures [28]. Subsequently, Hofheinz, Jager, and Kiltz [25] and Yamada, Hanaoka, and Kunihiro [43] improved its efficiency. However, all these signature schemes based on the RSA assumption require a large number of primality tests at signing and verifying.

Outline. In the next section, we give preliminaries to explain our results. In Section 3, we explain our intuition behind our construction, a new proof strategy, and the proposed signature scheme. Section 4 analyze the security and efficiency of the proposed scheme. In Section 5, we give several extensions, and in Section 6, we give our conclusion and end with open problems.

2 Background on Signatures, Bilinear Groups, and Diffie-Hellman Assumption

We define the following notations. We use $[a, b]$ to denote a set of integers between two integers, a and b. For a set S, $s \xleftarrow{\$} S$ denotes that the element s is uniformly chosen from S. For an algorithm Alg, $Alg(x) \rightarrow a$ 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 $Alg \rightarrow a$.

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 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 if the signature is valid; otherwise, 0.

Existential Unforgeability. We use the standard security notion for signatures, called *existential unforgeability with respect to chosen-message attacks* (EU-CMA), formalized by Goldwasser, Micali, and Rivest [24], and a slightly weaker model called *existential unforgeability with respect to weak chosen-message attacks* (EU-wCMA). The adversary in both security model 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 EU-CMA security model, the adversary is allowed to query any time before he outputs a forgery. However, the adversary in the EU-wCMA model should send the challenger the entire list of messages he wants to query before receiving the public key. We provide the formal definition of EU-CMA and EU-wCMA. Let $\mathsf{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be a signature scheme. We consider two following experiments.

```
$\mathbf{Exp}^{\text{EU-CMA}}_{\mathsf{SIG},\mathcal{A}}(\lambda)$
    $(PK, SK) \leftarrow \mathsf{KeyGen}(\lambda)$;
    $(M, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}(\cdot)}(PK)$;
    Let $List$ be the set of messages queried
     by the adversary;
    If $M \notin List$ and $\mathsf{Verify}(PK, M, \sigma) = 1$ return 1;
    Otherwise, return 0.
```

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

We define

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

Definition 1 *Let* SIG *be a signature scheme. If for any probabilistic polynomial-time adversary* $\mathcal{A}$, $\mathbf{Adv}^{EU\text{-}CMA}_{\mathsf{SIG},\mathcal{A}}(\lambda)$ ($\mathbf{Adv}^{EU\text{-}wCMA}_{\mathsf{SIG},\mathcal{A}}(\lambda)$, *respectively) is a negligible function in* λ, *we say that the signature scheme* SIG *is EU-CMA secure (EU-wCMA secure, respectively). More precisely, for any algorithm* $\mathcal{A}$ *with issuing at most* q *signing queries and running time* T, $\mathbf{Adv}^{EU\text{-}CMA}_{\mathsf{SIG},\mathcal{A}}(\lambda) < \epsilon$ ($\mathbf{Adv}^{EU\text{-}wCMA}_{\mathsf{SIG},\mathcal{A}}(\lambda) < \epsilon$, *respectively), then we say that* SIG *is* (q, ϵ, T)*-EU-CMA secure* ((q, ϵ, T)*-EU-wCMA secure, respectively).*

Chameleon Hash Functions and Generic Transformation. Krawczyk and Rabin [30] formalized the notion of the chameleon hash function and provided a simple construction based on the DL assumption in the standard model. A chameleon hash function H takes two inputs m (message) and r (randomness) and outputs the hash value $H(m;r)$. It satisfies three properties, collision-resistance, trapdoor collisions, and uniformity. The collision-resistance property states that it is infeasible (for the polynomial-time adversary) to find two distinct messages m and m' and randomness r and r' such that $H(m;r) = H(m';r')$. The uniformity means that for each message m, $H(m;r)$ has the same probability distribution where r is chosen uniformly at random. The trapdoor collisions property states that, given some trapdoor information, any pair m, r and any additional message m', it is possible to efficiently find a randomness r' such that $H(m;r) = H(m';r')$.

We review the chameleon hashes based on the DL assumption [30].[1] Let $\mathcal{G}_{ch}$ is a group generator that takes security parameter λ as input and outputs a cyclic group of prime order p' of 2λ-bits (e.g., an elliptic curve group generator).

$\mathsf{CHSetup}(\lambda)$: $\mathcal{G}_{ch}(\lambda) \to \mathbb{G}_{ch}$.
Choose $g_{ch} \xleftarrow{\$} \mathbb{G}_{ch}$ and $\beta \xleftarrow{\$} \mathbb{Z}_{p'}$ and compute $h_{ch} = g_{ch}^{\beta}$.
Output $\{g_{ch}, h_{ch}\}$, as the description of $H(\cdot;\cdot)$, and trapdoor $tr = \{\beta\}$, where $H(\cdot;\cdot) : \mathbb{Z}_{p'} \times \mathbb{Z}_{p'} \to \mathbb{G}_{ch}$ is defined by $(x, r) \mapsto g_{ch}^{x} h_{ch}^{r}$.

$\mathsf{Trapdoor\ collision}(tr, x, r, x')$: Solve the equation $x + \beta r = x' + \beta r$ with a variable r'.
Output r'

We can easily check that the above scheme satisfies three properties of chameleon hashes. In particular, the collision resistance of the above scheme tightly comes from the DL assumption on $\mathbb{G}_{ch}$; that is, if there exists an adversary finding collisions of the above chameleon hashes with ϵ_{ch} probability in time T_{ch}, then we can construct an algorithm solving the DL problem with ϵ_{ch} probability in time T'_{ch} such that $T'_{ch} \approx T_{ch}$.

The generic transformation from the EU-wCMA secure signature to the EU-CMA secure signature was used in many previously proposed signature schemes (e.g., [30, 41, 5, 27, 28]). Let $(\mathsf{G}, \mathsf{S}, \mathsf{V})$ be a EU-wCMA secure signature scheme for arbitrary length messages and $\mathsf{CHSetup}$ is a generator for description of a chameleon hash based on the DL assumption. Then, the following scheme is a EU-CMA secure signature scheme for fixed length messages.[2]

$\mathsf{KeyGen}(\lambda)$: Run $\mathsf{CHSetup}(\lambda) \to (H(\cdot;\cdot), tr)$ and $\mathsf{G}(\lambda) \to (\mathsf{pk}, \mathsf{sk})$, publish $PK = (pk, H)$, and then keep $SK = \{sk\}$.

$\mathsf{Sign}(PK, M, SK)$: Pick a random $r \in \mathbb{Z}_p$, compute $y = H(M;r)$, run $\mathsf{S}(pk, y, sk) \to \sigma'$, and then output the signature $\sigma = (\sigma', r)$.

$\mathsf{Verify}(PK, M, \sigma)$: Parse σ as (σ', r), compute $y = H(M;r)$, and then output $\mathsf{V}(pk, y, \sigma')$.

Lemma 1 ([28]) *If $(\mathsf{G}, \mathsf{S}, \mathsf{V})$ is (q, ϵ, T)-EU-wCMA secure and $\mathsf{CHSetup}$ is a generator for secure chameleon hashes, then the above scheme is $(q, 2(\epsilon + \epsilon_{ch}), T')$-EU-CMA secure signature scheme, where the chameleon hash satisfies (ϵ_{ch}, T_{ch})-collision resistance and $T' \approx T \approx T_{ch}$.*[3]

Bilinear Groups. We define bilinear groups of prime order. The proposed signature scheme essentially uses a bilinear map.

Definition 2 *We say that $\mathcal{G}$ is a bilinear group generator, if on inputting the security parameter λ, 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 $g' \in \mathbb{G}_2$, $e(g, g') \neq 1$.*

[1] In [30], the chameleon hashes is constructed over a multiplicative subgroup of a finite field. We can easily generalize it to the chameleon hashes over any cyclic groups, in which the DL assumption holds.

[2] For signing arbitrary length messages, the signer can first apply collision-resistance hash functions.

[3] In [28], a generic transformation is given for EU-wCMA secure signature scheme for fixed length messages. Note that we can easily generalize it and prove an analogous lemma for arbitrary length messages. In fact, if the message size of EU-wCMA secure signature scheme is larger than or equal to the size of the representation of the chameleon hash values, then the generic transformation derives EU-CMA secure signature scheme.

```
KeyGen(λ)          : Run $\mathcal{G} \to (p, \mathbb{G}, \mathbb{G}_t, e)$ and choose $v, u_1, \dots, u_q, g \xleftarrow{\$} \mathbb{G}$ and $\alpha \xleftarrow{\$} \mathbb{Z}_p$.
                     Output $PK = \{v, u_1, \dots, u_q, g, g^\alpha\}$ and $SK = \{\alpha\}$.
Sign(PK, M, SK)    : Compute $\sigma = (v \prod_{i=1}^{q} u_i^{M^i})^\alpha$, and output $\sigma$.
Verify(PK, M, σ)   : Check whether $e(\sigma, g) \overset{?}{=} e(g^\alpha, v \prod_{i=1}^{q} u_i^{M^i})$
                   : Output 1 if the equation holds; otherwise, 0.
```

Fig. 1. Short signatures with a large public key

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.

(Computational) Diffie-Hellman Assumption. We define the (computational) DH assumption in the bilinear group setting.

Definition 3 *Let $\mathcal{G}$ be a bilinear group generator. We say that $\mathcal{G}$ satisfies the (ϵ_{dh}, T_{dh})-DH assumption if for any T_{dh}-time probabilistic algorithm $\mathcal{B}$ the following advantage $\mathbf{Adv}_{\mathcal{B}}^{\mathbf{DH}}$ is less than ϵ_{dh}:*

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

3 Short Signatures with Short Public Key

In this section, we first propose a EU-wCMA secure signature scheme with a somewhat short public key by using two techniques in [25, 27], and then propose a EU-wCMA secure signature scheme with a 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.

We use λ to denote the security parameter and q to denote the maximum number of signing queries. Since we restrict the adversary to be computationally bounded (that is, we only consider the probabilistic polynomial time adversary), q is a polynomial in λ.

3.1 Combining Two Techniques: 'Somewhat' Short Public Key

We begin with exploring two techniques for obtaining short signature schemes in the standard model. In the simulation of the EU-wCMA 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 [26] to allow the simulator to produce only signatures on messages queried by the adversary. In particular, we use weak programmable hash functions [25] to construct EU-wCMA secure short signatures. We describe the short signature scheme with a polynomial size public key in Figure 1.

We assume that the public key contains the bilinear group description. For a 2λ-bit message M, we consider M as an element of $\mathbb{Z}_p$. $(v \prod_{i=1}^{q} u_i^{M^i})$ is a weak programmable hash function on input M that, in the EU-wCMA model, allows the simulator to sign on at most q messages, which are given by the adversary before generating the public key.[4] Furthermore, we can construct a simulator that extracts the solution of g^{ab} from the forgery by imbedding g^a in v and u_i and setting g and g^α by g and g^b, respectively.

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

```
$\mathsf{KeyGen}(\lambda)$        : Run $\mathcal{G} \to (p, \mathbb{G}, \mathbb{G}_t, e)$ and choose $v, u_1, \ldots, u_m, g_1, h, g \xleftarrow{\$} \mathbb{G}, \alpha \xleftarrow{\$} \mathbb{Z}_p$.
                                    Output $PK = \{v, u_1, \ldots, u_m, g_1, h, g, g^\alpha\}$ and $SK = \{\alpha\}$.
$\mathsf{Sign}(PK, M, SK)$ : Choose $t \xleftarrow{\$} \mathbb{Z}_p$ and $s_1 \xleftarrow{\$} [1, Q]$.
                                    Compute $\sigma_1 = (v \prod_{i=1}^{m} u_i^{M^i})^\alpha (h g_1^{s_1})^t$ and $\sigma_2 = g^{-t}$.
                                    Output $\sigma = (\sigma_1, \sigma_2, s_1)$.
$\mathsf{Verify}(PK, M, \sigma)$ : Parse $\sigma$ to $(\sigma_1, \sigma_2, s_1)$.
                                    Check whether $s_1 \in [1, Q]$. If not, abort and output 0.
                                    Check whether $e(\sigma_1, g)e(\sigma_2, h g_1^{s_1}) \overset{?}{=} e(g^\alpha, v \prod_{i=1}^{m} u_i^{M^i})$.
                                    Output 1 if the equation holds; otherwise, 0.
```

Fig. 2. Construction for a somewhat short public key

There is the other technique that obtains short signatures with a short public key by maintaining the index counter in the signer side [27]. 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 [4]. 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 a somewhat short public key and give the description of the scheme in Figure 2. In Figure 2, s_1 plays the role of index in [27] and we call s_1 as tag.

In Figure 2, we set Q to be polynomial in λ. The strategy of the simulation in the EU-wCMA model is as follows: The simulator guesses s_1^*, 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 s_1^* among the resulting signatures of singing queries. Under normal circumstances, the simulator cannot produce signatures with tag s_1^* (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, 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.

Remark. We used the combination of 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 [25]. Note that our original contribution is explained in Section 3.2.

3.2 Asymmetric Trade: Realizing Short Public Key

Let Q, m, and k be functions in λ. 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 Section 4.3.

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

We describe our main signature scheme in Figure 3. For each signature $\sigma = (\sigma_1, \sigma_2, \overrightarrow{s})$, we call $\overrightarrow{s}$ tag vector. In contrast to the scheme discussed in the previous section, we use a vector $\overrightarrow{s}$ instead of an integer s_1 in signatures. Roughly speaking, our analysis shows that the signature scheme in Figure 3 satisfies weak unforgeability when $mk = \Omega(\frac{\lambda}{\log \lambda})$ (this result contains the signatures with a somewhat short public key in Figure 2). In addition (roughly speaking again), since the public key size is $\Theta(m + k)$ group elements, we can attain the optimal public key size when m and k are nearly equal. On the other hand, the size of signatures will increase when the parameter k increases. However, each s_i is a $\log Q$-bit integer, and so $\overrightarrow{s}$ is asymptotically much shorter than $\Theta(\lambda)$-bit (if we set Q as a polynomial in λ). This is an *asymmetric trade*

```
KeyGen($\lambda$)         : Run $\mathcal{G} \to (p, \mathbb{G}, \mathbb{G}_t, e)$.
                            Choose $v, u_1, \ldots, u_m, g_1, \ldots, g_k, h, g \xleftarrow{\$} \mathbb{G}$,   $\alpha \xleftarrow{\$} \mathbb{Z}_p$.
                            Output $PK = \{v, u_1, \ldots, u_m, g_1, \ldots, g_k, h, g, g^{\alpha}\}$ and $SK = \{\alpha\}$.
Sign($PK, M, SK$)         : Uniformly choose $t \xleftarrow{\$} \mathbb{Z}_p$ and $s_1, \ldots, s_k \xleftarrow{\$} [1, Q]$.
                            Compute $\sigma_1 = (v \prod_{i=1}^{m} u_i^{M^i})^{\alpha} (h \prod_{i=1}^{k} g_i^{s_i})^t$ and $\sigma_2 = g^{-t}$.
                            Output $\sigma = (\sigma_1, \sigma_2, \overrightarrow{s})$, where $\overrightarrow{s} = (s_1, \ldots, s_k) \in [1, Q]^k$.
Verify($PK, M, \sigma$)   : Parse $\sigma$ to $(\sigma_1, \sigma_2, \overrightarrow{s})$.
                            Check whether $\overrightarrow{s} \in [1, Q]^k$. If not, abort and output 0.
                            Check whether $e(\sigma_1, g) e(\sigma_2, h \prod_{i=1}^{k} g_i^{s_i}) \stackrel{?}{=} e(g^{\alpha}, v \prod_{i=1}^{m} u_i^{M^i})$.
                            Output 1 if the above equation holds; otherwise, 0.
```

Fig. 3. Main construction for a short public key

between the public key and signatures. 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 4.3.

Our construction of the short signatures with a short public key in Figure 3 is a simple generalization of the short signatures with a somewhat short public key in Figure 2. However, the analysis of the security in the EU-wCMA model is more challenging than the construct itself. The basic strategy of the simulator in the EU-wCMA model of the signature scheme in Figure 2 is guessing the tag s_1^* of the forgery and then using the programmability of the weak programmable hash function $(v \prod_{i=1}^{m} u_i^{M^i})$ to sign for the signature with the same tag. We cannot naively apply this proof strategy to the generalized construction. To obtain a short public key, we should set k sufficiently large (but not too much). However, if k is large, then the simulator cannot guess the tag vector of the forgery, $\overrightarrow{s}^* \in [1, Q]^k$, 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 weak unforgeability of the proposed signature scheme. In particular, we explain the method to guess the tag vector $\overrightarrow{s}^*$ of the forgery with non-negligible probability. In fact, we cannot guess all the bits of $\overrightarrow{s}^*$, but only part of $\overrightarrow{s}^*$ with non-negligible probability. This is sufficient for our proof strategy.

We begin with defining notations for efficient explanation. Let S and S^i be sets $[1, Q]$ and $[1, Q]^i$ (i times canonical product set), respectively. For $j \in [1, q]$, let $\overrightarrow{s}_j \in S^k$ be the tag vector (randomly chosen by the simulator) of the signature on the jth message (queried by the adversary). Let $\overrightarrow{s}^* = (s_1^*, \ldots, s_k^*) \in S^k$ be the tag vector of the forgery output by the adversary. For $\overrightarrow{s} \in S^k$ and $i \leq k$, let $\overrightarrow{s}^{(i)} \in S^i$ be the first i entries of $\overrightarrow{s}$ (e.g., $\overrightarrow{s} = (s_1, \ldots, s_k)$ and $\overrightarrow{s}^{(i)} = (s_1, \ldots, s_i)$). We separate the adversaries into several types according to the relations between $\overrightarrow{s}^*$ and $\{\overrightarrow{s}_i\}_{i \in [1,q]}$. To this end, for fixed $\{\overrightarrow{s}_i\}_{i \in [1,q]}$, we first define the set S_i as

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

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

Example. Suppose that

$$\overrightarrow{s}_1^{(i)} = \ldots = \overrightarrow{s}_{m+2}^{(i)} \neq \overrightarrow{s}_j^{(i)} \text{ for } j \in [m+3, q],$$

$$\overrightarrow{s}_1^{(i+1)} = \ldots = \overrightarrow{s}_{m+1}^{(i+1)} \neq \overrightarrow{s}_j^{(i+1)} \text{ for } j \in [m+2, q],$$

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

$$\begin{cases} \overrightarrow{s}_j^{(i)} \in S_i \text{ for } j \in [1, m+2] \\ \overrightarrow{s}_j^{(i)} \notin S_i \text{ for } j \in [m+3, q], \end{cases}$$

$$\begin{cases} \overrightarrow{s}_j^{(i+1)} \in S_{i+1} \text{ for } j \in [1, m+1] \\ \overrightarrow{s}_j^{(i+1)} \notin S_{i+1} \text{ for } j \in [m+2, q], \end{cases}$$

and $|S_i| = |S_{i+1}| = 1$. □

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

1. $|S_1| < \lambda$
2. $n < k$ (equivalently $S_k = \emptyset$, that is, $|S_k| < 1$)

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 $\Pr_{\overrightarrow{s}_1, \ldots, \overrightarrow{s}_q \overset{\$}{\leftarrow} S^k}[|S_i| \geq j] < (\frac{q^{m+1}}{(m+1)! Q^{im}})^j$.

We prove Lemma 2 in Appendix 4.1.

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

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

Here, $\overrightarrow{s}^{*(i-1)} \in S_{i-1}$ implies that $\overrightarrow{s}^{*(j)} \in S_j$ 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 $\overrightarrow{s}^{*(i)}$ with probability $\frac{1}{|S_{i-1}| \cdot |S|}$; it guesses $\overrightarrow{s}^{*(i-1)}$ with $\frac{1}{|S_{i-1}|}$ and s_i^* with $\frac{1}{|S|}$ (we use the second property for the case $i = n+1$). Since the simulator can guess the type of the adversary with probability $\frac{1}{k}$, it can guess the tag vector of the forgery with at least probability $\frac{1}{k\lambda Q}$ (we use the first property for the inequality $|S_{i-1}| \leq |S_1| < \lambda$).

The other parts of the proof strategy are similar to the strategy for the short signatures with a somewhat short public key in Figure 2 as we mentioned in Section 3.1; (1) guess $\overrightarrow{s}^*$, (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 $\overrightarrow{s}^*$, using the programmability of the weak programmable hash functions. Since $\overrightarrow{s}^{*(i)} \notin S_i$ (for $i = n+1$, $\overrightarrow{s}^{*(i)} \notin S_i = \emptyset$) implies that there are at most m tag vectors same as $\overrightarrow{s}^{*(i)}$, the simulator can response m signatures with tag vector $\overrightarrow{s}^{*(i)}$ using the programmability of $(v \prod_{i=1}^{m} u_i^{M^i})$. If $k\lambda Q$ is bounded by a polynomial in λ, 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, ϵ, T)-EU-wCMA secure assuming the (ϵ', T')-DH assumption holds such that*

$$\epsilon' = \frac{1}{k\lambda Q}(\epsilon - \frac{q^{m+1}}{(m+1)! Q^{km}} - \frac{q}{p} - (\frac{q^{m+1}}{(m+1)! Q^m})^\lambda) \quad \textit{and} \quad T \approx T'.$$

We relegate the concrete proof of Theorem 1 to Appendix 4.2.

For a EU-CMA secure signature scheme, we can apply a generic transformation from a EU-wCMA secure scheme to a EU-CMA secure scheme given in Section 2. (Note that if the representation of a chameleon hash value needs more than 2λ-bit, then we first apply $(\epsilon_{crh}, T_{crh})$-collision resistant hash function to the EU-wCMA secure signature scheme for arbitrary length messages since the chameleon hash values are messages of EU-wCMA secure signatures.)

From Theorem 1 and Lemma 1, we obtain the following corollary (by using a (ϵ_{ch}, T_{ch})-collision resistant hash function if need be).[5]

Corollary 1. *Let* SIG *be the signature scheme for fixed length messages described in Figure 3 and* SIG' *be the signature scheme for arbitrary length messages obtained from* SIG *and a* (ϵ_{ch}, T_{ch})*-collision resistant hash function. Let* $\mathsf{CHSetup}$ *is a generator for the chameleon hashes based on the DL assumption. Let* SIG'' *be the signature scheme resulting from the generic transformation in Section 2 on* SIG' *and* $\mathsf{CHSetup}$*. If* $\mathcal{G}$ *satisfies* $(\frac{1}{k\lambda Q}(\frac{\epsilon}{4} - \frac{q^{m+1}}{(m+1)!Q^{km}} - \frac{q}{p} - (\frac{q^{m+1}}{(m+1)!Q^{m}})^{\lambda}), T_{dh})$*-DH assumption and* $\mathcal{G}_{ch}$ *satisfies the* (ϵ_{dl}, T_{dl})*-DL assumption, then* SIG *is* $(q, \epsilon + 4\epsilon_{crh} + 2\epsilon_{dl}, T)$*-EU-CMA secure, where* $\mathcal{G}$ *and* $\mathcal{G}_{ch}$ *are group generators used in* SIG *and* $\mathsf{CHSetup}$*, respectively,* q *is the maximum number of signatures the adversary can obtain, and* $T_{dh} \approx T_{dl} \approx T$*.*

If the collision resistant hash function is not used (that is, the representation of the output of chameleon hashes needs less than or equal to 2λ*-bit), and so* SIG'' *is the signature scheme resulting from the generic transformation on* SIG *and* $\mathsf{CHSetup}$*, then* SIG'' *is* $(q, \frac{\epsilon}{2} + 2\epsilon_{dl}, T)$*-EU-CMA secure.*

Proof. By Theorem 1, SIG is $(q, \frac{\epsilon}{4}, T)$-EU-wCMA secure. By the standard hybrid argument, we can show that SIG' is $(q, 2(\frac{\epsilon}{4} + \epsilon_{crh}), T)$-EU-wCMA secure. Lastly, by Lemma 1, SIG'' is $(q, 2(2(\frac{\epsilon}{4} + \epsilon_{crh}) + \epsilon_{dl}), T)$-EU-CMA secure since the chameleon hash function based on the (ϵ_{dl}, T_{dl})-DL assumption has (ϵ_{dl}, T_{dl})-collision resistance. If the collision resistant hash function is not used, then Theorem 1 and Lemma 1 directly imply that SIG'' is $(q, 2(\frac{\epsilon}{4} + \epsilon_{dl}), T)$-EU-CMA secure. $\square$

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

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

We give asymptotic values of m, k, and Q for satisfying the above conditions and short public key for $\lambda \in \{80, 128, 192, 256\}$ and $q \in \{2^{30}, 2^{40}\}$ in Section 4.3. For such parameters (e.g., example parameter 1), we obtain the following corollary.

Corollary 2. *Let* SIG'' *be a signature scheme in given in Corollary 1 with parameter* m*,* k*, and* Q *satisfying the above three conditions. Then,* SIG'' *is* $(q, \epsilon + 4\epsilon_{crh} + 2\epsilon_{dl}, T)$*-EU-CMA secure assuming* $(\frac{\epsilon}{4k\lambda Q} - neg(\lambda), T_{dh})$*-DH assumption holds, where* $neg(\lambda)$ *is a negligible function in* λ *and* $T \approx T_{dh}$*.*

If SIG'' *is the signature scheme resulting from the generic transformation on* SIG *and* $\mathsf{CHSetup}$ *(that is, the collision resistant hash function is not used), then* SIG'' *is* $(q, \frac{\epsilon}{2} + 2\epsilon_{dl}, T)$*-EU-CMA secure assuming* $(\frac{\epsilon}{4k\lambda Q} - neg(\lambda), T_{dh})$*-DH assumption holds, where* $neg(\lambda)$ *is a negligible function in* λ *and* $T \approx T_{dh}$*.*

[5] If we use elliptic curve groups as the base group $\mathbb{G}_{ch}$, over which the chameleon hashes are defined, then the representation of the chameleon hash values are sufficiently short to be used as messages for EU-wCMA secure signatures without firstly applying collision-resistance hash functions. However, if we use multiplicative subgroups of finite fields as $\mathbb{G}_{ch}$, then we need collision-resistance hash functions to map the chameleon hash values into short messages for EU-wCMA secure signatures.

4 Analysis

4.1 Proof of Lemma 2

In this subsection, we prove Lemma 2. 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 $\Pr_{\overrightarrow{s}_1,\ldots,\overrightarrow{s}_q \overset{\$}{\leftarrow} S^k}[|S_i| \geq j]$ as

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

To compute $\Pr_{f \overset{\$}{\leftarrow} F}[|T_f| \geq j]$, 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 $\Pr_{f \overset{\$}{\leftarrow} F}[|T_f| \geq j]$. 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:

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

Remark. The result in Lemma 2 is similar as the lemma given in [25] called 'generalized birthday bound'. Note that Lemma 2 is more general than 'generalized birthday bound'; e.g., if we set $i = 1$ and $j = 1$, then the result in Lemma 2 provides a more tighter upper bound than 'generalized birthday bound' given in [25].

4.2 Proof of Theorem 1

In this subsection, we prove Theorem 1. Suppose that there exists a probabilistic (polynomially bounded) adversary $\mathcal{A}$ which makes at most q signing queries and outputs a valid forgery with probability ϵ. We construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ as an internal sub-algorithm and breaks the DH assumption.

Simulation Description. We now describe an algorithm $\mathcal{B}$ on input (g, g^a, g^b), the uniform instance of the DH problem. Let $A = g^a$ and $B = g^b$. We use the notations S, S^i, and S_i defined in 3.2.

KeyGen: Without loss of generality, we assume that $\mathcal{A}$ makes the maximum q queries for signing. First, $\mathcal{B}$ receives a list L of messages $M_1, \ldots, M_q$ on which $\mathcal{A}$ requests signatures. $\mathcal{B}$ uniformly generates random vectors $\overrightarrow{s}_1, \ldots, \overrightarrow{s}_q \overset{\$}{\leftarrow} S^k$ such that $\overrightarrow{s}_j$ will be used in the j-th signing query on M_j. $\mathcal{B}$ separately behaves according to whether $n = k$. If $n = k$, we say that the event $E1$ occurs. When $E1$ occurs, $\mathcal{B}$ acts as a real challenger by running the KeyGen algorithm. When $E1$ does not occur, $\mathcal{B}$ guesses the tag vector of the forgery; $\mathcal{B}$ uniformly chooses ℓ from $[1, n+1]$, $(s'_1, \ldots, s'_{\ell-1})$ from $S_{\ell-1}$ except for the case $\ell = 1$, and s'_ℓ from S such that $(s'_1, \ldots, s'_\ell) \notin S_\ell$. Since $|S_\ell| \leq \frac{Q}{m+1} < Q = |S|$, there always exists such a s'_ℓ. We use notation $\overrightarrow{s}'^{(\ell)}$ to denote $(s'_1, \ldots, s'_\ell)$.

Next, $\mathcal{B}$ generates a public key. Since $\overrightarrow{s}'^{(\ell)} \notin S_\ell$, there exist at most m distinct integers $j_1, \ldots, j_m \in [1, q]$ such that $\overrightarrow{s}'^{(\ell)} = \overrightarrow{s}^{(\ell)}_{j_1} = \ldots = \overrightarrow{s}^{(\ell)}_{j_m}$. Let J be a set of such integers, so that $|J| \leq m$. Let $f(x)$ be a polynomial defined as

$$f(x) = \begin{cases} \prod_{i \in J}(X - M_i) & \text{if } |J| \geq 1, \\ 1 & \text{otherwise.} \end{cases}$$

Let $x_0, \ldots, x_m$ be coefficients of $f(x)$ such that $f(x) = \sum_{i=0}^m x_i X^i$ (for $i > |J|$, set $x_i = 0$). $\mathcal{B}$ uniformly chooses integers $y_0, \ldots, y_m, z_0, \ldots, z_\ell, w_0, \ldots, w_k \xleftarrow{\$} \mathbb{Z}_p$. If for some $j \in [1, q] \setminus J$, $\sum_{i=1}^\ell (s_{ji} - s'_i) z_i = 0$, where $\overrightarrow{s}_j = (s_{j1}, \ldots, s_{jk})$, then we say that the event $E2$ occurs. If $E2$ occurs, $\mathcal{B}$ behaves like a real challenger running the KeyGen algorithm and using $\overrightarrow{s}_j$ in the j-th signing query. Otherwise ($E2$ does not occur), $\mathcal{B}$ generates a public key as follows.

$$\begin{aligned}
v &:= A^{x_0} g^{y_0}, \\
u_i &:= A^{x_i} g^{y_i} \text{ for } i \in [1, m], \\
h &:= A^{-\sum_{i=1}^\ell s'_i z_i} g^{w_0}, \\
g_i &:= \begin{cases} A^{z_i} g^{w_i} & \text{for } i \in [1, \ell] \\ g^{w_i} & \text{for } i \in [\ell+1, k] \end{cases}, \\
g &:= g, \\
g^\alpha &:= B
\end{aligned}$$

The secret key is b, which is unknown to $\mathcal{B}$.

Sign: When $E1$ or $E2$ occurs, $\mathcal{B}$ behaves like a real challenger by running the Sign algorithm except for using $\overrightarrow{s}_1, \ldots, \overrightarrow{s}_q$ generated in the *KeyGen* phase.

Otherwise (both $E1$ and $E2$ do not occur), $\mathcal{B}$ generates signatures on $M_1, \ldots, M_q$ as follows. For the j-th signing query, $\mathcal{B}$ separately signs on M_j according to whether $j \in J$.

Case $j \in [1, q] \setminus J$: Since $E2$ does not occur, $\sum_{i=1}^\ell (s_{j1} - s'_i) z_i \neq 0$. Choose $t' \xleftarrow{\$} \mathbb{Z}_p$ and compute

$$\begin{aligned}
\sigma_{j1} &= B^{(\sum_{i=0}^m y_i M_j^i) - (w_0 + \sum_{i=1}^k s_{ji} w_i) \frac{(\sum_{i=0}^m x_i M_j^i)}{(\sum_{i=1}^\ell (s_{j1} - s'_i) z_i)}} \cdot (A^{(\sum_{i=1}^\ell (s_{j1} - s'_i) z_i)} g^{w_0 + \sum_{i=1}^k s_{ji} w_i})^{t'} \\
\text{and } \sigma_{j2} &= B^{(\sum_{i=0}^m x_i M_j^i)/(\sum_{i=1}^\ell (s_{j1} - s'_i) z_i)} \cdot g^{-t'}.
\end{aligned}$$

Case $j \in J$: Choose $t \xleftarrow{\$} \mathbb{Z}_p$ and compute

$$\sigma_{j1} = B^{(\sum_{i=0}^m y_i M_j^i)} (g^{w_0 + \sum_{i=1}^k s_{ji} w_i})^t \text{ and } \sigma_{j2} = g^{-t}.$$

For both cases, we define the j-th signature $\sigma_{(j)}$ on M_j as $(\sigma_{j1}, \sigma_{j2}, \overrightarrow{s}_j)$.

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 $\sigma_{(1)}, \ldots, \sigma_{(q)}$.

Extract solution of DH problem from forgery: When $E1$ or $E2$ occurs, $\mathcal{B}$ outputs a random group element as the answer of the DH problem. Otherwise, do the following.

$\mathcal{B}$ receives a message M^* along with a forgery $\sigma^* = (\sigma_1^*, \sigma_2^*, \overrightarrow{s}^*)$ on M^* from $\mathcal{A}$ such that $M^* \notin L$. If $\mathsf{Verify}(PK, M^*, \sigma^*) = 0$, $\mathcal{B}$ aborts. If $\overrightarrow{s}^{*(\ell)} \neq \overrightarrow{s}'^{(\ell)}$, $\mathcal{B}$ aborts. Otherwise, $M^* \notin L$, so that $f(M^*) = \sum_{i=0}^m x_i (M^*)^i \neq 0$. $\mathcal{B}$ outputs $(\sigma_1^* \cdot B^{-\sum_{i=0}^m y_i (M^*)^i} \cdot (\sigma_2^*)^{w_0 + \sum_{i=1}^k s_i^* w_i})^{\frac{1}{f(M^*)}}$.

During an interaction between $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{B}$ randomly chooses a group element from $\mathbb{G}$ and outputs it if $\mathcal{B}$ aborts or $\mathcal{A}$ quits the interaction.

Analysis (Distribution of Simulated Transcript). First, we argue that the distribution of the public key and signatures on $M_1, \dots, M_q$ generated by $\mathcal{B}$ are identical to that of the real challenger. $\overrightarrow{s}_1, \dots, \overrightarrow{s}_q$ are uniformly distributed. Even if $E1$ and $E2$ are determined by $\overrightarrow{s}_1, \dots, \overrightarrow{s}_q$, the tag vectors $\overrightarrow{s}_1, \dots, \overrightarrow{s}_q$ are used no matter whether $E1$ or $E2$ occurs. Therefore, in the view of $\mathcal{A}$, $\overrightarrow{s}_1, \dots, \overrightarrow{s}_q$ are uniformly distributed, so we only focus on the other randomness used in the public key and signatures.

When the events $E1$ or $E2$ occurs, $\mathcal{B}$ acts as the real challenger, so that the distributions generated by $\mathcal{B}$ and the real challenger are identical. Otherwise, $y_0, \dots, y_m$, and $w_0, \dots, w_k$ are uniformly chosen from $\mathbb{Z}_p$ and are independent from $E1$ and $E2$. Furthermore, the DH instance is also uniformly generated. Therefore, the distributions of the public key generated by $\mathcal{B}$ are identical to those of the output of the KeyGen algorithm. Next, we consider the distribution of the j-th signature on M_j when $E1$ and $E2$ do not occur (that is, $\sum_{i=1}^{\ell}(s_{j1} - s'_i)z_i \neq 0$). If $j \in [1, q] \setminus J$, randomness t is distributed as if $t = -\frac{(\sum_{i=0}^{m} x_i M_j^i b)}{(\sum_{i=1}^{\ell}(s_{j1}-s'_i)z_i)} + t'$. More precisely, we can check the following equations from the equality $t = -\frac{(\sum_{i=0}^{m} x_i M_j^i b)}{(\sum_{i=1}^{\ell}(s_{j1}-s'_i)z_i)} + t'$.

$$\begin{aligned}
&\sigma_{j1}\\
&= B^{(\sum_{i=0}^{m} y_i M_j^i)-(w_0+\sum_{i=1}^{k} s_{ji}w_i)\frac{(\sum_{i=0}^{m} x_i M_j^i)}{(\sum_{i=1}^{\ell}(s_{ji}-s'_i)z_i)}} \cdot (A^{\sum_{i=1}^{\ell}(s_{j1}-s'_i)z_i} g^{w_0+\sum_{i=1}^{k} s_{ji}w_i})^{t'}\\
&= B^{(\sum_{i=0}^{m} y_i M_j^i)} (g^{ab})^{\sum_{i=0}^{m} x_i M_j^i} \cdot (A^{\sum_{i=1}^{\ell}(s_{ji}-s'_i)z_i} g^{w_0+\sum_{i=1}^{k} s_{ji}w_i})^{\frac{(-\sum_{i=0}^{m} x_i M_j^i b)}{(\sum_{i=1}^{\ell}(s_{ji}-s'_i)z_i)}} \cdot (A^{\sum_{i=1}^{\ell}(s_{ji}-s'_i)z_i} g^{w_0+\sum_{i=1}^{k} s_{ji}w_i})^{t'}\\
&= (\prod_{i=0}^{m}(A^{x_i} g^{y_i})^{M_j^i})^b \cdot (A^{-\sum_{i=1}^{\ell} s'_i z_i} g^{w_0} \cdot \prod_{i=1}^{\ell}(A^{z_i} g^{w_i})^{s_{ji}} \cdot \prod_{i=\ell+1}^{k}(g^{w_i})^{s_{ji}})^t\\
&= (v \prod_{i=1}^{m} u_i^{M_j^i})^b (h \prod_{i=1}^{k} g_i^{s_{ji}})^t
\end{aligned}$$

and

$$\sigma_{j2} = B^{(\sum_{i=0}^{m} x_i M_j^i)/(\sum_{i=1}^{\ell}(s_{j1}-s'_i)z_i)} \cdot g^{-t'} = g^{-t}.$$

Since t' is uniformly chosen from $\mathbb{Z}_p$, t is also uniformly distributed. Therefore, the distribution of σ_{j1} and σ_{j2} is identical to that of the output of Sign algorithm.

If $j \in J$, t is uniformly distributed, that is,

$$\begin{aligned}
&\sigma_{j1}\\
&= B^{(\sum_{i=0}^{m} y_i M_j^i)} (g^{w_0+\sum_{i=1}^{k} s_{ji}w_i})^t\\
&= B^{(\sum_{i=0}^{m} y_i M_j^i)} (g^{ab})^{\sum_{i=0}^{m} x_i M_j^i} \cdot (A^{\sum_{i=1}^{\ell}(s_{ji}-s'_i)z_i} g^{w_0+\sum_{i=1}^{k} s_{ji}w_i})^t\\
&= (\prod_{i=0}^{m}(A^{x_i} g^{y_i})^{M_j^i})^b \cdot (A^{-\sum_{i=1}^{\ell} s'_i z_i} g^{w_0} \prod_{i=1}^{\ell}(A^{z_i} g^{w_i})^{s_{ji}} \prod_{i=\ell+1}^{k}(g^{w_i})^{s_{ji}})^t\\
&= (v \prod_{i=1}^{m} u_i^{M_j^i})^b (h \prod_{i=1}^{k} g_i^{s_{ji}})^t
\end{aligned}$$

By the definition of J and $f(x)$, $f(M_j) = (\sum_{i=0}^{m} x_i M_j^i) = 0$ and $s_{ji} = s'_i$ for all $j \in J$ and $i \in [1, \ell]$. This implies that the second equality holds. Since $\sigma_{j2} = g^{-t}$, σ_{j1} and σ_{j2} are distributed as the output of Sign algorithm.

Analysis (Success Probability). We now show that the success probability of $\mathcal{B}$ breaking the DH assumption. We separate the types of adversaries according to $\overrightarrow{s}^*$ as follows.

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

As we mentioned in Section 3.2, we can see that the above $n+1$ types of the adversaries are pairwise disjoint and cover all possible adversaries.

We define several events. Let ℓ^* be the type of adversary and C be the event such that $\ell^* = \ell$ and $\overrightarrow{s}^{*(\ell)} = \overrightarrow{s}'^{(\ell)}$. Let D be the event such that $|S_1| \geq \lambda$, F be the event such that $\mathcal{A}$ successfully (weakly) forges a signature, and S be the event such that $\mathcal{B}$ outputs g^{ab}. From the definition of events, we derive the followings.

$$\begin{aligned}
&\Pr[S] \\
&> \Pr[S \wedge C \wedge \neg E1 \wedge \neg E2 \wedge \neg D \wedge F] \\
&= \Pr[S \wedge C | \neg E1 \wedge \neg E2 \wedge \neg D \wedge F] \cdot \Pr[\neg E1 \wedge \neg E2 \wedge \neg D \wedge F] \\
&= \Pr[S | \neg E1 \wedge \neg E2 \wedge \neg D \wedge F \wedge C] \cdot \Pr[C | \neg E1 \wedge \neg E2 \wedge \neg D \wedge F] \cdot \Pr[\neg E1 \wedge \neg E2 \wedge \neg D \wedge F],
\end{aligned}$$

where the probability is over all randomness used by $\mathcal{A}$ and $\mathcal{B}$ in the simulation.

We first show that the probability

$$\Pr[S | \neg E1 \wedge \neg E2 \wedge \neg D \wedge F \wedge C] = 1.$$

Suppose that the event $(\neg E1 \wedge \neg E2 \wedge \neg D \wedge F \wedge C)$ occurs. Since $\mathcal{A}$ outputs a valid forgery such that σ_1^* and σ_2^* satisfy the verification equation (event F), we can see that σ_1^* and σ_2^* are $(v \prod_{i=1}^{m} u_i^{M^{*i}})^{\alpha}(h \prod_{i=1}^{k} g_i^{s_i^*})^t$ and g^{-t} for some t, respectively. From the hypothesis $\overrightarrow{s}^{*(\ell)} = \overrightarrow{s}'^{(\ell)}$ (event C), we know

$$\sigma_1^* = (g^{ab})^{f(M^*)} (g^b)^{\sum_{i=0}^{m} y_i M^{*i}} (g^t)^{w_0 + \sum_{i=1}^{k} s_i^* w_i}.$$

We can show that the output of $\mathcal{B}$ is

$$(\sigma_1^* \cdot B^{-\sum_{i=0}^{m} y_i (M^*)^i} \cdot (\sigma_2^*)^{w_0 + \sum_{i=1}^{k} s_i^* w_i})^{\frac{1}{f(M^*)}} = g^{ab}.$$

That is, $\Pr[S | \neg E1 \wedge \neg E2 \wedge \neg D \wedge F \wedge C] = 1$.

Next, we show that

$$\Pr[C | \neg E1 \wedge \neg E2 \wedge \neg D \wedge F] \geq \frac{1}{k} \cdot \frac{1}{\lambda} \cdot \frac{1}{Q}.$$

Event E_2 is independent from the other events since it is determined by independent variables z_i's, which are perfectly hidden from adversarial view, so that $\Pr[C | \neg E1 \wedge \neg D \wedge F] = \Pr[C | \neg E1 \wedge \neg E2 \wedge \neg D \wedge F]$. Since ℓ^* should be in $[1, n+1]$ and ℓ is perfectly hidden from adversarial view, $\Pr[\ell = \ell^* | \neg E1 \wedge \neg D \wedge F] = \frac{1}{n+1} \geq \frac{1}{k}$ (ℓ is chosen independently from other values used in the simulation so that it is independent from the other events). Suppose that $\ell^* = \ell$. Then, $\overrightarrow{s}^{*(\ell-1)} \in S_{\ell-1}$. The event $\overrightarrow{s}^{*(\ell)} = \overrightarrow{s}'^{(\ell)}$ is equal to $(\overrightarrow{s}^{*(\ell-1)} = \overrightarrow{s}'^{(\ell-1)}) \wedge (s_\ell^* = s_\ell')$ (except for the case $\ell = 1$). Since $\overrightarrow{s}'^{\ell-1}$ and s_ℓ' are uniformly chosen from $S_{\ell-1}$ and S, respectively and $\overrightarrow{s}'^{\ell}$ is perfectly hidden from adversarial view,

$$\Pr[\overrightarrow{s}^{*(\ell)} = \overrightarrow{s}'^{(\ell)} | (\ell = \ell^*) \wedge \neg E1 \wedge \neg D \wedge F] \; = \; \begin{cases} \frac{1}{Q} & \text{if } \ell = 1 \\ \frac{1}{|S_{\ell-1}|} \cdot \frac{1}{Q} & \text{otherwise.} \end{cases}$$

From the condition $\neg D$, we know that $|S_{\ell-1}| \leq |S_1| < \lambda$. Therefore, $\Pr[\overrightarrow{s}^{*(\ell)} = \overrightarrow{s}'^{(\ell)} | (\ell = \ell^*) \wedge \neg E1 \wedge \neg D \wedge F] > \frac{1}{\lambda Q}$; thus,

$$\begin{aligned}
&\Pr[C | \neg E1 \wedge \neg E2 \wedge \neg D \wedge F] \\
&= \Pr[C | \neg E1 \wedge \neg D \wedge F] \\
&= \Pr[\overrightarrow{s}^{*(\ell)} = \overrightarrow{s}'^{(\ell)} | (\ell = \ell^*) \wedge \neg E1 \wedge \neg D \wedge F] \cdot \Pr[\ell = \ell^* | \neg E1 \wedge \neg D \wedge F] \\
&> \tfrac{1}{k\lambda Q}.
\end{aligned}$$

Next, we show that the probability

$$\Pr[\neg E1 \wedge \neg E2 \wedge \neg D \wedge F] = \epsilon - (\tfrac{q^{m+1}}{(m+1)!Q^{km}} + \tfrac{q}{p} + (\tfrac{q^{m+1}}{(m+1)!Q^{m}})^{\lambda}).$$

By Lemma 2, we see that

$$\Pr[E1] = \Pr[|S_k| \geq 1] < \frac{q^{m+1}}{(m+1)!Q^{km}}$$

and

$$\Pr[D] = \Pr[|S_1| \geq \lambda] < (\frac{q^{m+1}}{(m+1)!Q^{m}})^{\lambda}.$$

The $E2$ is the event such that for some $j \in [1,q] \setminus J$, $\sum_{i=1}^{\ell}(s_{ji} - s'_i)z_i = 0$. For each $j \in [1,q] \setminus J$, there always exists a s_{ji} such that $(s_{ji} - s'_i) \neq 0$ for some i (by the definition of J); and hence, for each $j \in [1,q] \setminus J$ we obtain

$$\Pr[\sum_{i=1}^{\ell}(s_{ji} - s'_i)z_i = 0] = \frac{1}{p},$$

where the probability goes over the choice of $z_1, \ldots, z_\ell$. By the union bound, we obtain

$$\Pr[E2] < \frac{q}{p}.$$

We can easily verify the following equation.

$$\begin{aligned} &\Pr[\neg E1 \wedge \neg E2 \wedge \neg D \wedge F] \\ &= \Pr[F] - \Pr[(E1 \vee E2 \vee D) \wedge F] \\ &> \Pr[F] - (\Pr[E1] + \Pr[E2] + \Pr[D]). \end{aligned}$$

Therefore, we obtain the desired result

$$\Pr[\neg E1 \wedge \neg E2 \wedge \neg D \wedge F] \ = \ \epsilon - (\tfrac{q^{m+1}}{(m+1)!Q^{km}} + \tfrac{q}{p} + (\tfrac{q^{m+1}}{(m+1)!Q^{m}})^{\lambda}).$$

Putting it all together, we obtain the lower bound of the success probability of $\mathcal{B}$

$$\Pr[S] > \frac{1}{k\lambda Q}(\epsilon - \frac{q^{m+1}}{(m+1)!Q^{km}} - \frac{q}{p} - (\frac{q^{m+1}}{(m+1)!Q^{m}})^{\lambda}).$$

4.3 Parameter Selection for Short Public Key

The description of our construction did not explain how we choose m, k, and Q. In this subsection, we show that how we can optimize public key size. From Theorem 1, we obtain several conditions for polynomial-time reduction to the DH problem. First, $k\lambda Q$ should be polynomially bounded in λ. Second, $\frac{q^{m+1}}{(m+1)!Q^{km}}$ and $(\frac{q^{m+1}}{(m+1)!Q^{m}})^{\lambda}$ should be negligible function in λ. Therefore, Q should be a polynomial in λ. For simple analysis, we assume that $Q = \Theta(q)$, more precisely, $Q = Cq$ for small constant $C > 1$, and $\frac{q^{m+1}}{(m+1)!Q^{km}}$ and $(\frac{q^{m+1}}{(m+1)!Q^{m}})^{\lambda}$ are smaller than $\frac{1}{2^{\lambda}}$. We compute asymptotically optimal values of m and k for short public key size, and then provide practical parameters with reasonable reduction loss, which is comparable to that of Waters signature in [42].

Condition 1. $k\lambda q$ is polynomially bounded in λ.
Condition 2. $\frac{q^{m+1}}{(m+1)!Q^{km}} < \frac{1}{2^{\lambda}}$.
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^{λ}. Since $Q = \Theta(q)$ 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, $mk = \Omega(\frac{\lambda}{\log\lambda})$ or $m = \Omega(\frac{\lambda}{\log\lambda})$. For optimizing public key size, we should minimize $m+k$ since the size of public key is $\Theta(m+k)$. Therefore, $m = \Theta(\sqrt{\frac{\lambda}{\log\lambda}})$ and $k = \Theta(\sqrt{\frac{\lambda}{\log\lambda}})$ are (asymptotically) optimal parameter for optimal

```
KeyGen($\lambda$)      : Run $\mathcal{G} \to (p, \mathbb{G}, \mathbb{G}_t, e)$.
                         Choose $v, u_1, \ldots, u_m, g_1, \ldots, g_k, h, g \xleftarrow{\$} \mathbb{G}$,  $\alpha \xleftarrow{\$} \mathbb{Z}_p$, and pick a PRF key $K$ at random.
                         Output $PK = \{v, u_1, \ldots, u_m, g_1, \ldots, g_k, h, g, g^\alpha, K\}$ and $SK = \{\alpha\}$,
                            where $PRF : \{0,1\}^* \to \{0,1\}^{k\lceil \log Q \rceil}$ be a pseudorandom function family.
Sign($PK, M, SK$)  : Uniformly choose $t \xleftarrow{\$} \mathbb{Z}_p$.
                         Compute $PRF_K(M) = (s_1, \ldots, s_k)$, $\sigma_1 = (v \prod_{i=1}^m u_i^{M^i})^\alpha (h \prod_{i=1}^k g_i^{s_i})^t$, and $\sigma_2 = g^{-t}$.
                         Output $\sigma = (\sigma_1, \sigma_2)$.
Verify($PK, M, \sigma$) : Parse $\sigma$ to $(\sigma_1, \sigma_2)$.
                         Compute $PRF_K(M) = (s_1, \ldots, s_k)$.
                         Check whether $e(\sigma_1, g) e(\sigma_2, h \prod_{i=1}^k g_i^{s_i}) \overset{?}{=} e(g^\alpha, v \prod_{i=1}^m u_i^{M^i})$.
                         Output 1 if the above equation holds; otherwise, 0.
```

Fig. 4. Signatures with tag compression

public key size. In fact, if we set $m = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ and $k = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$, then the above three conditions also hold.

Next, we provide practical parameters for $\lambda \in \{80, 128, 192, 256\}$ and $q \in \{2^{30}, 2^{40}\}$, where λ is the security parameter and q is the maximum bound for adversarial signing queries. if the above condition 2 and condition 3 hold, our security proof loses $4k\lambda Q$ 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 λ and q, k is a small constant (at most 3 in our example parameters) and Q is Cq with small constant $C > 1$ ($C = 2^3$ 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 [42], where we assume that the message size is 2λ.[6] The example practical parameters are given in the table 1, where we set $Q = 2^3 q$ and thus all reduction loss of signature scheme with parameters in the table 1 are less than or equal to $96\lambda q$. 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 k satisfying the above three conditions. The size of a tag vector is a $k\lceil \log Q \rceil$-bit string, which is asymptotically smaller than 2λ if $k = \Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ and Q is a polynomial in λ; 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 2λ (e.g., a tag vector is 129-bit string when $k = 3$ and $Q = 2^{43}$).

5 Extensions

In this section, we provide two extensions of the main construction in Figure 3. We give a (EU-wCMA secure) variant of the main scheme that has shorter signatures (two group elements). Next, an instantiation using asymmetric pairings (type-2 pairings or type-3 pairings) instead of symmetric pairings (type-1 pairings) is considered.

5.1 Tag Compression using Pseudorandom Functions

In this subsection, we introduce a trick for tag compression using (non-adaptive) pseudorandom functions (PRF). Note that similar techniques are used in the RSA-based signatures [27, 28, 25, 43] to compress random prime numbers used in each signature. If we use this trick, we can reduce the signature size of EU-wCMA secure scheme to two group elements, but we should add constant size terms in the public parameters. (This modification does not affect the asymptotic size of public key.) In some application, short signatures are

[6] For arbitrarily large message, both our scheme and Waters' scheme take collision resistant hash function value, and then run signing algorithms. To prevent the birthday attack, the output of hash function should be larger than 2λ.

```
$\mathsf{KeyGen}(\lambda)$          : Run $\mathcal{G} \to (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e)$.
                     Choose $\mathsf{v}, \mathsf{u}_1, \ldots, \mathsf{u}_\mathsf{m}, \mathsf{g}_1, \ldots, \mathsf{g}_\mathsf{k}, \mathsf{h}, \alpha \xleftarrow{\$} \mathbb{Z}_p, g \xleftarrow{\$} \mathbb{G}_2, g' \xleftarrow{\$} \mathbb{G}_1,$ and pick a PRF key $K$ at random.
                     Compute $v = g^{\mathsf{v}}, u_i = g^{\mathsf{u}_i}, g_i = g^{\mathsf{g}_i}, h = g^{\mathsf{h}}$.
                     Output $PK = \{v, u_1, \ldots, u_m, g_1, \ldots, g_k, h, g \in \mathbb{G}_2, g'^{\alpha} \in \mathbb{G}_1, K\}$
                         and $SK = \{\mathsf{v}, \mathsf{u}_1, \ldots, \mathsf{u}_m, \mathsf{g}_1, \ldots, \mathsf{g}_k, \mathsf{h}, \alpha \in \mathbb{Z}_p, g' \in \mathbb{G}_1\}$,
                         where $PRF : \{0,1\}^* \to \{0,1\}^{k\lceil \log Q \rceil}$ be a pseudorandom function family.
$\mathsf{Sign}(PK, M, SK)$ : Uniformly choose $t \xleftarrow{\$} \mathbb{Z}_p$
                     Compute $PRF_K(M) = (s_1, \ldots, s_k), \sigma_1 = g'^{(\mathsf{v} + \sum_{i=1}^m \mathsf{u}_i M^i)\alpha + (\mathsf{h} + \sum_{i=1}^k \mathsf{g}_i s_i)t}$ and $\sigma_2 = g'^{-t} \in \mathbb{G}_1$.
                     Output $\sigma = (\sigma_1, \sigma_2)$.
$\mathsf{Verify}(PK, M, \sigma)$  : Parse $\sigma$ to $(\sigma_1, \sigma_2)$.
                     Compute $PRF_K(M) = (s_1, \ldots, s_k)$.
                     Check whether $e(\sigma_1, g) e(\sigma_2, h \prod_{i=1}^k g_i^{s_i}) \overset{?}{=} e(g'^{\alpha}, v \prod_{i=1}^m u_i^{M^i})$.
                     Output 1 if the above equation holds; otherwise, 0.
```

Fig. 5. Instantiation using asymmetric pairings

important even though public key size and signing/verifying costs increase. Then, the signature scheme with tag compression technique is appropriate in such applications.

Let $PRF : \{0,1\}^* \to \{0,1\}^{m\lceil \log Q \rceil}$ be a pseudorandom function family. The signer randomly chooses a PRF key K, uses $PRF_K(M)$ as the tag vector associated with the signature of M, and publishes the description of PRF and the key K of PRF as a part of PK. Then, we can remove tag vectors from signatures since everyone who knows a message and PK can compute the corresponding tag vector. Hence, the resulting scheme can have shorter signatures (two group elements). The signature scheme with tag compression technique is given in Figure 4.

This simple modified signature scheme is also EU-wCMA secure; in the security proof, we need the uniformity of all tag vectors used in signing queries, which is guaranteed by the proof of Lemma 2. More precisely, as we mentioned in Section 3.2, two properties $|S_1| < \lambda$ and $|S_k| < 1$ are essentially used in the security proof. If each tag vector is computed by $\overrightarrow{s}_i = R(M_i)$, where $R(\cdot)$ is a random function, then we obtain the above two properties since Lemma 2 with an appropriate parameter selection of m, k, and Q implies that $\Pr[|S_1| \geq \lambda]$ and $\Pr[|S_k| \geq 1]$ are negligible functions in λ. (In our choices of practical parameters, both probability are less than $\frac{1}{2^\lambda}$.) Therefore, if we change a random function R to a pseudorandom function PRF with randomly chosen key K, then we also obtain the same result such that $|S_1| < \lambda$ and $|S_k| < 1$. (If not (that is, $|S_1| \geq \lambda$ or $|S_k| \geq 1$), we can construct a distinguisher, which distinguish the output of PRF_K from random strings, where K is randomly chosen. Note that for this case, it is no matter whether the key K is published in the public key of the signature scheme or not.) By using these two properties $|S_1| < \lambda$ and $|S_k| < 1$, we can prove the security of a signature scheme with tag compression since other parts are essentially same to the proof of Theorem 1.

5.2 Instantiation using Asymmetric Pairings

Although we described our construction using type-1 pairings in Section 3, we can easily modify our construction to be instantiated using type-2 pairings or type-3 parings as in Figure 5. The scheme using type-1 parings and its security proof does not use the symmetry property; Our main idea to achieve sublinear public key is to divide adversarial types according to tag vectors in the security proof, and this technique is independent of pairing's types. Therefore, we can prove the security of the instantiation using asymmetric pairings by following the same proof strategy used for the scheme with type-1 pairings. For type-2 pairings, the security can be reduced to the co-DHP: given $\phi : \mathbb{G}_2 \to \mathbb{G}_1$ and $g', g^b \in \mathbb{G}_1, g, g^a \in \mathbb{G}_2$ such that $g' = \phi(g)$, compute g'^{ab}. For type-3 pairings, the security of the proposed DH-based signature scheme can be reduced to the co-DHP*: given $g', g'^a, g'^b \in \mathbb{G}_1, g, g^a \in \mathbb{G}_2$, compute g'^{ab}. Note that the security of the Waters signature scheme using asymmetric pairings is also based on the same problems [1, 11].

6 Conclusion and Open Problems

We developed a simple construction combining two techniques for short signatures and then proposed its generalization and a variant using tag compression technique. The new idea of this research is the analysis of the security and efficiency of the proposed construction. Our analysis showed that the proposed signature scheme has short signatures and a short public key ($\Theta(\sqrt{\frac{\lambda}{\log \lambda}})$ group elements) and is as secure as the DH assumption by giving polynomial-time reduction. Even if the signature size of the proposed signature scheme is larger than that of the Waters signature scheme, it is still shorter than RSA-based signatures [28, 25].

There are a number of interesting open problems. The proposed signature scheme is not publicly re-randomizable since the randomly chosen tag vectors and the randomness used for the chameleon hashes cannot be publicly re-randomized. Re-randomizable signatures (e.g., Camenisch-Lysyanskaya signatures [9] and Waters signatures [42]) are useful in some applications (e.g., group signatures [3] and blind signatures [33, 40]). It would be interesting to find a practical re-randomizable signature scheme from the standard assumption in the standard model, in particular, with short signatures and a short public key. Furthermore, we ask for standard-model (re-randomizable) signatures under the standard assumptions that attain similar asymptotic efficiency as practical signature schemes under strong assumptions or schemes in the random oracle model.

References

1. M. Bellare and T. Ristenpart. Simulation without the artificial abort: simplified proof and improved concrete security for Waters' IBE scheme. In *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424. Springer, 2009.
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. P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi. Get shorty via group signatures without encryption. In *SCN 2010*, volume 6280 of *LNCS*, pages 381–398. Springer, 2010.
4. D. Boneh and X. Boyen. Efficient selective-id identity based encryption without random oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer-Verlag, 2004.
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, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signature from bilinear maps. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432. Springer, 2003.
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. D. Brown and R. Gallant. The static diffie–hellman problem. Available in http://eprint.iacr.org/2004/306.
9. 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.
10. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *ACM STOC 2003*, pages 209–218, 2003.
11. S. Chatterjee, D. Hankerson, E. Knapp, and A. Menezes. Comparing two pairing-based aggregate signature schemes. In *Designs, Codes and Cryptography*, volume 55, pages 141–167. Springer, 2010.
12. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO 1982*, pages 199–203, 1982.
13. D. Chaum and E. van Heyst. Group signatures. In *CRYPTO 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
14. J. H. Cheon. Discrete logarithm problems with auxiliary inputs. In *Journal of Cryptology*, volume 23, pages 457–476, 2010.
15. 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.
16. R. Cramer and V. Shoup. Signature schemes based on the strong rsa assumption. In *ACM CCS 1999*, pages 46–51. ACM Press, 1999.

17. Y. Dodis, R. Oliveira, and K. Pietrzak. On the generic insecurity of the full domain hash. In *CRYPTO 2005*, volume 3621 of *LNCS*, pages 449–466. Springer, 2005.
18. 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.
19. M. Fischlin. The cramer-shoup strong-rsa signature scheme revisited. In *PKC 2003*, volume 2567 of *LNCS*, pages 116–129. Springer, 2003.
20. T. E. Gamal. 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.
21. 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.
22. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *ACM STOC 2008*, pages 197–206, 2008.
23. 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.
24. 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.
25. D. Hofheinz, T. Jager, and E. Kiltz. Short signatures from weaker assumptions. In *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 647–666. Springer, 2011.
26. D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. In *Journal of Cryptology*, volume 25, pages 484–527. Springer, 2012.
27. 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.
28. 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.
29. R. Kalakota and A. B. Whinston. *Electronic cermmerce: A manager's guide.* Addison-Wesley, 1997.
30. H. Krawczyk and T. Rabin. Chameleon signatures. In *NDSS 2000.* The Internet Society, 2000.
31. G. Leurent and P. Q. Nguyen. How risky is the random-oracle model? In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 445–464. Springer, 2009.
32. A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In *SAC 1999*, volume 1758 of *LNCS*, pages 184–199. Springer, 1999.
33. S. Meiklejohn, H. Shacham, and D. M. Freeman. Limitations on transformations from composite-order to prime-order groups: The case of round-optimal blind signatures. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 519–538. Springer, 2010.
34. D. Naccache. Secure and *practical* identity-based encryption. In *IACR Cryptology ePrint Archive 2005:369*, 2005.
35. M. Naor. On cryptographic assumptions and challenges. In *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, 2003.
36. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO 1992*, volume 740 of *LNCS*, pages 31–53. Springer, 1992.
37. T. Okamoto. Efficient blind and partially blind signatures without random oracles. In *TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, 2006.
38. R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, 2001.
39. C. P. Schnorr. Efficient signature generation for smart cards. In *Journal of Cryptology*, volume 4, pages 239–252. Springer, 1991.
40. J. H. Seo and J. H. Cheon. Beyond the limitation of prime-order groups, and round optimal blind signatures. In *TCC 2012*, volume 7194 of *LNCS*, pages 133–150. Springer, 2012.
41. A. Shamir and Y. Tauman. Improved online/offline signature schemes. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 355–367. Springer, 2001.
42. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer-Verlag, 2005.
43. 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.