# Unique Signature with Short Output from CDH Assumption

Shiuan-Tzuo Shen, Amir Rezapour, and Wen-Guey Tzeng

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
{vink,rezapour,wgtzeng}@cs.nctu.edu.tw

**Abstract.** We give a simple and efficient construction of unique signature on groups equipped with bilinear map. In contrast to prior works, our proof of security is based on *computational Diffie-Hellman* problem in the random oracle model. Meanwhile, the resulting signature consists of only one group element. Due to its simplicity, security and efficiency, our scheme is suitable for those situations that require to overcome communication bottlenecks. Moreover, the unique signature is a building block for designing chosen-ciphertext secure cryptosystems and verifiable random functions, which have found many interesting applications in cryptographic protocol design.

**Keywords:** Unique signature, strongly unforgeable signature, verifiable unpredictable function, verifiable random function, bilinear map, random oracle model

## 1 Introduction

Since the invention of public key cryptography, various attempts have been made to design a provably secure cryptosystem. A remarkable proof of security is a polynomial time reduction from solving a *standard mathematical problem* (weak assumption) to the problem of breaking a cryptosystem in a standard model. For example, factoring big integers and computing discrete logarithms in prime order groups are two standard mathematical problems for cryptographic protocol design. Unlike traditional signature schemes, unique signature, a.k.a. verifiable unpredictable function (VUF), is a function from the message space to the signature space under the given public key. This particular property ensures that each message would have only "one" possible signature. From the security perspective, unique signature is not only existentially unforgeable against the chosen message attack, but also strongly unforgeable against the chosen message attack. The latter property assures that the adversary cannot even produce a valid signature for an earlier signed message.

Intuitively, unique signatures are very fascinating objects, as there is no reason to verify a signature on the same message twice. For instance, if one has verified a signature on one particular message, it is unnecessary to verify the message again unless the signature is changed. Another situation includes the signature scheme with a very efficient signer to generate many signatures for one

particular message. This may simply lead to overload a verifier to verify many signatures on the same message. Above all, Unique signatures are a building block for constructing an *adaptive CCA-secure* IBE encryption scheme from a *selective-identity CPA-secure* IBE scheme [3].

Unique signature has significant implication for constructing verifiable random functions (VRFs). VRF has found many interesting applications, such as non-interactive zero-knowledge proofs, micropayment schemes, verifiable transaction escrow schemes, compact e-cash, adaptive oblivious transfer protocols, and keyword search as discussed in [11].

### 1.1 Contribution

The primary objective of this study is to find a unique signature scheme with a weaker assumption and a signature of only one group element. This is always an appealing task for cryptographers. In contrast to earlier findings, we come up with a new provable scheme under the computational Diffie-Hellman (CDH) problem. In addition to using a weaker assumption, our unique signature consists of only one group element, which results in a fixed and small signature on any arbitrary input message length. Therefore, our unique signature enjoys a shorter signature than Boneh et al. [6] and Lysyanskaya [14].

Boneh et al. introduced the BLS signature [5], which enjoys a short signature and efficiency in both signature generation and verification algorithms. They proved existential unforgeability based on CDH assumption in the random oracle model. However, it is easy to see that their scheme also achieves strongly existential unforgeability. BLS signature outputs a signature of one element, and its signing key and verification key also consist of one element respectively. Therefore, it is relatively more efficient than our construction. Nevertheless, it still has some efficiency issues as the output length of the hash function grows. Informally, in the construction of BLS signature, the purpose of the hash function is to map a given message $m$ into a group element. However, to ensure the security of the hash function, we may need to employ an elliptic curve of larger group size[1]. This affects the performance of BLS signature and leads to a larger signature. In contrast to BLS signature, we hash a message to determine the signing key of the signature. Therefore, the group size in our construction is independent of the output length of the hash function. There is another difference between BLS signature and our unique signature. Although the two signatures are all provable in the random oracle model, they rely on different level of programmability of random oracle. BLS signature needs a random oracle for embedding the challenge instance of some hard problem, while our unique signature needs a random oracle for random outputs only.

*Strongly existential unforgeability.* Our construction is based on the result of Lysyanskaya [14], where the signature on an $n_0$-bit message consists of $\theta(n_0)$

[1] NIST [1] recommends SHA-256, SHA-384, and SHA-512 for minimum security of digital signatures, but the recommended group size of an elliptic curve is 256 bits.

group elements. Lysyanskaya proves the existential unforgeability based on the $l$-CDH problem. She embeds the challenge instance in $\ell$ independent indexes of the public key. Therefore, she can employ an error correcting code to bound the success probability of the reduction. In contrast with Lysyanskaya, we aim to give a signature of a single group element on an $n_0$-bit message, and prove the existential unforgeability based on the CDH problem. The challenging part is to give a non-negligible lower bound for the success probability of our reduction. Therefore, we cannot use the proof technique of Lysyanskaya because the error correcting code that we need does not exist. We use a different technique to bound the success probability. Our strategy is to inject variability into the signature. Therefore, we design a dynamic pattern for the signature, where the combination of secret exponents is determined by the hash output of the signed message. Meanwhile, the signature that contains the solution of the CDH problem has a specific pattern. Hence, we can reduce the failure probability and obtain a non-negligible lower bound for the success probability of our reduction.

*Malicious signer resistance.* Lysyanskaya [14] achieves malicious signer resistance. She implicitly represents each codeword symbol by an element of the signature. If two messages are different, then their signatures are also different. In contrast with Lysyanskaya, we compress our signature into one group element. It is possible that two distinct messages result in the same signature. To prove that our signature achieves malicious signer resistance, we first work on giving an upper bound for the number of hash outputs which result in the same signature. We propose the notion of the equivalent set for a signature and show that the size of an equivalent set is in a negligible proportion. Then, we can prove malicious signer resistance in the random oracle model. The next task is to relax the requirement of the random oracle due to the fact that a malicious signer may be able to choose the hash function. The difficulty is that most of the cryptographic protocols rely on a trusted source of randomness. An honest signer will choose his secret key randomly, but in contrast a malicious signer would not necessarily do that. We propose the H-F-H structure to resist a malicious signer, where H stands for a hash function and F stands for a one-way permutation. The H-F-H structure has the following properties:

- To evaluate an output of the H-F-H structure, a malicious signer has to decide his public key first. Thus, he cannot choose his secret key to force two hash outputs to be in the same equivalent set. This makes the malicious signer to guess a message that results in the same signature.
- The H-F-H structure is one-way. Therefore, a malicious signer cannot compute a message from an equivalent set. More precisely, the probability of finding an input of the F-layer for a given output is negligible. Even if the malicious signer can find an input of the F-layer, the input has to pass the verification of the $H$-layer.
- The design of double hash layers makes a malicious signer hard to find a candidate for the hash function. The two $H$-layers employ the same hash function. In addition, the output of the first $H$-layer will determine the input

of the second $H$-layer. Even if the malicious signer can find a candidate for first H-layer, the candidate has to pass the verification of the second $H$-layer or vise-versa.

The rest of the paper is organized as follows: In the next section, we review the related works. Some definitions are provided in Sect. 3. The unique signature scheme, its efficiency, and its applications are presented in Sect. 4. The security proofs are provided in Sect. 5. Finally, the conclusion is given in Sect. 6.

## 2 Related Work

Unique signature has been known to exist in the random oracle model. Until the result in [15], there was no construction for such schemes in the standard model. In addition to the seminal work of Micali et al. [15], which is based on the strong RSA assumption, there are few unique signatures in the standard model. Lysyanskaya in [14] proposed such a scheme based on the many-DH assumption. This differs from that of Micali et al., in both the underlying assumptions and signature size. In contrast, Lysyanskaya provided a signature of $n$ elements. Dodis proposed a unique signature scheme based on a much stronger assumption, sum-free $s$-decisional Diffie-Hellman assumption (SF-$l$-DDH). The key size is half of Lysyanskaya [14]'s. Dodis et al. [8] introduced another unique signature scheme based on $l$-Diffie-Hellman inversion assumption ($l$-DHI). The signature consists of only one element. Kuchta and Manulis [13] proposed a generic construction for unique aggregate signatures, which can be converted to distributed verifiable random functions. Jager [12] proposed a nearly identical unique signature scheme to that proposed by Lysyanskaya, but a new construction of VRF without having to resort to the Goldreich-Levin transformation [9]. The VRF presented in [12] is a relatively simple and efficient with large input space and full adaptive security using $q$-decisional Diffie-Hellman assumption in the standard model. Abdalla et al. [2] provided a methodology to construct a VRF by showing some connections to identity-based encryption. Moreover, they considered a few constructions without pairings in a more limited setting in which the number of queries was upper-bounded. They also showed that the Boneh-Franklin ID-KEM [4] can lead to a very efficient VRF in the random-oracle model. Boneh et al. [5] proposed the BLS signature, which produces a signature of only one group element. The signing key and verification key are also short. Their security proof is based on the *computational Diffie-Hellman* (CDH) problem in the random oracle model. Boneh et al. [6] constructed a strongly unforgeable signature based on the *computational Diffie-Hellman* (CDH) problem that produces signature of 2 group elements plus a short string. Unique signature schemes are clearly strongly unforgeable, but not all strongly unforgeable signature schemes are necessarily unique signatures. For instance, in the aforementioned work [6], as long as there is randomness in the signing algorithm, it is still possible for the *adversary* to produce a new valid signature on the previously signed message by choosing different randomness.

## 3 Definition

We first recall some standard notations and definitions that will be used throughout the paper. Let $k$ be a security parameter. We model the participants in the cryptographic model by probabilistic polynomial-time Turing machines (PPTMs), whose running time is at most polynomial in $k$. In the rest of the paper, complexity classes are with respect to $k$, unless there is an explicit specification.

We say that a function $\mu : \mathbb{N} \to \mathbb{R}^+$ is negligible if for every positive polynomial $P(\cdot)$ and all sufficiently large $k$, it holds that $0 < \mu(k) < \frac{1}{P(k)}$. For instance, $\mu(k) = 2^{-k}$ is a negligible function.

The number of elements in a set $\mathcal{X}$ is denoted as $|\mathcal{X}|$, and the bit length of an element $x \in \mathcal{X}$ is denoted as $|x|$. Choosing an element $x$ from set $\mathcal{X}$ randomly and uniformly is denoted as $x \in_R \mathcal{X}$. The value of $x$ mod $n$ is denoted as $[x]_n$.

A binary string of length $n$ consists of $n$ symbols, where each symbol has two possible values. The set of all binary strings of length $n$ is denoted as $\{0,1\}^n$, and the set of all binary strings of arbitrary length is denoted as $\{0,1\}^* = \bigcup_{n=0}^{\infty}\{0,1\}^n$. The $i$-th symbol of a string $x \in \{0,1\}^n$ is denoted as $x(i-1)$, where the index $i-1$ is between 0 and $n-1$. The concatenation of two strings $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ is denoted as $x\|y$.

### 3.1 Unique Signature

The notation of a unique signature was introduced by Goldwasser and Ostrovsky [10][2]. A unique signature must be a strongly unforgeable signature, but a strongly unforgeable signature [6] is not necessarily a unique signature. Unique signature is also known as verifiable unpredictable function. A verifiable unpredictable function may not be a verifiable random function, but a verifiable random function [15] must be a verifiable unpredictable function. A unique signature scheme consists of four polynomial-time algorithms $\mathsf{Setup}$, $\mathsf{KeyGen}$, $\mathsf{Sign}$, and $\mathsf{Verify}$, which are defined as follows:

- $\mathsf{Setup}(1^k) \to \pi$: It is a probabilistic algorithm run by the system manager. Algorithm $\mathsf{Setup}$ takes security parameter $k$ as the input, and outputs public parameter $\pi$.
- $\mathsf{KeyGen}(\pi) \to (sk, pk)$: It is a probabilistic algorithm run by a signer. Algorithm $\mathsf{KeyGen}$ takes public parameter $\pi$ as the input, and outputs secret key $sk$ and public key $pk$.
- $\mathsf{Sign}(\pi, sk, pk, m) \to \sigma$: It is a deterministic algorithm run by a signer. Algorithm $\mathsf{Sign}$ takes public parameter $\pi$, secret key $sk$, public key $pk$, and message $m$ as inputs, and outputs signature $\sigma$.
- $\mathsf{Verify}(\pi, pk, m, \sigma) \to \{\mathsf{Yes}, \mathsf{No}\}$: It is a deterministic algorithm run by a verifier. Algorithm $\mathsf{Verify}$ takes public parameter $\pi$, public key $pk$, message $m$, and signature $\sigma$ as inputs, and outputs the validity of $(m, \sigma)$ under $pk$.

These algorithms must satisfy the following requirements:

[2] Goldwasser and Ostrovsky called it invariant signature.

- **Consistency:** For every public parameter $\pi$ produced by algorithm $\mathsf{Setup}$, every key pair $(sk, pk)$ produced by algorithm $\mathsf{KeyGen}$, and every message $m$, we have that $\mathsf{Verify}(\pi, pk, m, \mathsf{Sign}(\pi, sk, pk, m)) = \mathsf{Yes}$.
- **Uniqueness:** For every public parameter $\pi$ produced by algorithm $\mathsf{Setup}$, every key pair $(sk, pk)$ produced by algorithm $\mathsf{KeyGen}$, every message $m$, and every pair of signatures $\sigma_1$ and $\sigma_2$, if we have $\mathsf{Verify}(\pi, pk, m, \sigma_1) = \mathsf{Verify}(\pi, pk, m, \sigma_2) = \mathsf{Yes}$, then it must imply $\sigma_1 = \sigma_2$.

**Strongly Existential Unforgeability.** Security for a unique signature scheme is defined as the security against strongly existential forgery under an adaptive chosen message attack. Strongly existential unforgeability is a stronger security property, comparing with existentially unforgeable signature schemes. In both cases, an adversary who is given a signature for some message of his choice might not be able to produce a valid signature for a new message. Nevertheless, the strongly existential unforgeability property ensures that the adversary cannot even produce a valid signature for a previously signed message. This notion is defined by the unforgeability game $\mathsf{Game}^{\mathsf{UF}}$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

- **Setup.** Challenger $\mathcal{C}$ runs algorithm $\mathsf{Setup}(1^k)$ to generate public parameter $\pi$. Then, $\mathcal{C}$ runs algorithm $\mathsf{KeyGen}(\pi)$ to generate secret-public key pair $(sk, pk)$. $\mathcal{C}$ holds $sk$ and gives $(\pi, pk)$ to adversary $\mathcal{A}$.
- **Query.** Adversary $\mathcal{A}$ queries $q$ messages $(m_1, m_2, \ldots, m_q)$ of his choice. Challenger $\mathcal{C}$ returns $q$ signatures $(\sigma_1, \sigma_2, \ldots, \sigma_q)$ to answer the queries. These queries are issued adaptively, namely, $\mathcal{A}$ can choose $m_i$ after seeing the signatures $(\sigma_1, \ldots, \sigma_{i-1})$.
- **Forgery.** Adversary $\mathcal{A}$ outputs a message-signature pair $(m^*, \sigma^*)$, where $m^*$ has not been queried in the query phase.

Adversary $\mathcal{A}$ wins the game if $m^* \notin \{m_1, m_2, \ldots, m_q\}$ and $\mathsf{Verify}(\pi, pk, m^*, \sigma^*) = \mathsf{Yes}$. The advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{UF}}$ is defined as the probability that $\mathcal{A}$ wins the game.

**Definition 1.** *A signature scheme achieves $(t, q, \epsilon)$ strongly existential unforgeability against adaptive chosen message attack if no adversary, who runs in time $t$ and queries at most $q$ messages, can win the unforgeability game with advantage over $\epsilon$.*

**Malicious Signer Resistance.** Besides strongly existential unforgeability, we study another important security property which is called *malicious signer resistance*. A malicious signer will try to find a specific setting of his secret key so that he can sign two different messages with the same signature. The signer obtains some benefit from this collision. He can sign a message first and then claim that the signature is for another message instead. The malicious signer resistance property ensures that an adversary cannot sign two distinct messages to the same signature even if the secret key is on his choice. This notion is defined by the malicious signer game $\mathsf{Game}^{\mathsf{MS}}$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

- **Setup.** Challenger $\mathcal{C}$ runs algorithm $\mathsf{Setup}(1^k)$ to generate public parameter $\pi$. Then, $\mathcal{C}$ gives $\pi$ to adversary $\mathcal{A}$.
- **Answer.** Adversary $\mathcal{A}$ outputs a public key $pk$, two messages $(m_1, m_2)$, and a signature $\sigma$, where $m_1$ and $m_2$ are distinct.

Adversary $\mathcal{A}$ wins the game if $m_1 \neq m_2$ and $\mathsf{Verify}(\pi, pk, m_1, \sigma) = \mathsf{Yes} = \mathsf{Verify}(\pi, pk, m_2, \sigma)$. The advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MS}}$ is defined as the probability that $\mathcal{A}$ wins the game.

**Definition 2.** *A signature scheme achieves $(t, \epsilon)$ malicious signer resistance if no adversary, who runs in time $t$, can win the malicious signer game with advantage over $\epsilon$.*

### 3.2 Cryptographic Primitive

*One-Way Permutation.* Let $\mathcal{X}$ be a space of exponent size. A owe-way permutation $F : \mathcal{X} \to \mathcal{X}$ is a bijective one-way function. Specifically, $F$ should satisfy the following properties:

- Computability. For all input $x \in \mathcal{X}$, there is a (deterministic) polynomial-time algorithm $A(\cdot)$ who can output $F(x)$. That is, we have $A(x) = F(x)$ for every input $x$.
- One-Wayness. For a random message $x \in_R \mathcal{X}$, there is no probabilistic polynomial-time adversary $\mathcal{A}$ who can output an inverse of $F(x)$ with non-negligible probability. That is, for every probabilistic polynomial-time adversary $\mathcal{A}$, every positive polynomial $P(\cdot)$, and all sufficiently large $k$, we have
$$\Pr_{x \in_R \mathcal{X}} \left[ x' \in F^{-1}(F(x)) : \mathcal{A}(F(x)) = x' \right] < \frac{1}{P(k)} \ .$$

**Definition 3.** *We say that a one-way permutation $F$ is $(t, \epsilon)$ one-way if no adversary can break the one-wayness of $F$ in time $t$ with probability over $\epsilon$.*

*Cryptographic Hash Function.* Let $\mathcal{M}$ be a message space of exponent size, and $\mathcal{D}$ be a digest space of exponent size. A cryptographic hash function $H : \mathcal{M} \to \mathcal{D}$ is a one-way function. Specifically, $H$ should satisfy the following properties:

- Computability. For all messages $m \in \mathcal{M}$, there is a (deterministic) polynomial-time algorithm $A(\cdot)$ who can output $H(m)$. That is, we have $A(m) = H(m)$ for every input $m$.
- Pre-image Resistance. For a random message $m \in_R \mathcal{M}$, there is no probabilistic polynomial-time adversary $\mathcal{A}$ who can output an inverse of $H(m)$ with non-negligible probability. That is, for every probabilistic polynomial-time adversary $\mathcal{A}$, every positive polynomial $P(\cdot)$, and all sufficiently large $k$, we have
$$\Pr_{m \in_R \mathcal{M}} \left[ m' \in H^{-1}(H(m)) : \mathcal{A}(H(m)) = m' \right] < \frac{1}{P(k)} \ .$$

– Second Pre-image Resistance. Given a random message $m \in_R \mathcal{M}$, there is no probabilistic polynomial-time adversary $\mathcal{A}$ who can output another inverse $m' \in \mathcal{M}$ of $H(m)$ with non-negligible probability. That is, for every probabilistic polynomial-time adversary $\mathcal{A}$, every positive polynomial $P(\cdot)$, and all sufficiently large $k$, we have

$$\Pr_{m \in_R \mathcal{M}} \left[ m' \in H^{-1}(H(m)) \wedge m \neq m' : \mathcal{A}(H, m) = m' \right] < \frac{1}{P(k)} \ .$$

– Collision Resistance. There is no probabilistic polynomial-time adversary $\mathcal{A}$ who can output two distinct messages $m \in \mathcal{M}$ and $m' \in \mathcal{M}$ such that $H(m) = H(m')$ with non-negligible probability. That is, for every probabilistic polynomial-time adversary $\mathcal{A}$, every positive polynomial $P(\cdot)$, and all sufficiently large $k$, we have

$$\Pr \left[ H(m) = H(m') \wedge m \neq m' : \mathcal{A}(H) = (m, m') \right] < \frac{1}{P(k)} \ .$$

**Definition 4.** *We say that a cryptographic hash function $H$ is $(t, \epsilon)$ pre-image resistant if no adversary can break the pre-image resistance of $H$ in time $t$ with probability over $\epsilon$.*

**Definition 5.** *We say that a cryptographic hash function $H$ is $(t, \epsilon)$ second pre-image resistant if no adversary can break the second pre-image resistance of $H$ in time $t$ with probability over $\epsilon$.*

**Definition 6.** *We say that a cryptographic hash function $H$ is $(t, \epsilon)$ collision resistant if no adversary can break the collision resistance of $H$ in time $t$ with probability over $\epsilon$.*

*Bilinear Map.* Let $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ be two multiplicative cyclic groups of prime order $q$. Let $g$ be a generator of $\mathbb{G}$. A map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ is called an admissible bilinear map if it satisfies the following properties:

– Bilinearity: for all $u, v \in \mathbb{G}$ and $x, y \in \mathbb{Z}_q$, we have $\hat{e}(u^x, v^y) = \hat{e}(u, v)^{xy}$.
– Non-degeneracy: we have $\hat{e}(g, g) \neq \mathbf{1}$, where $\mathbf{1}$ is the identity element of $\mathbb{G}_{\mathbb{T}}$.
– Computability: there is a polynomial-time algorithm to compute $\hat{e}(u, v)$ $\forall\ u, v \in \mathbb{G}$

### 3.3 Hardness Assumption

The security of our unique signature scheme will be reduced to hardness of the *computational Diffie-Hellman* (CDH) problem.

**Definition 7.** *Let $\mathbb{G}$ be a multiplicative cyclic group of prime order $q$. Let $g$ be a generator of $\mathbb{G}$. The CDH problem is to compute $g^{ab}$ when given $g, g^a, g^b \in \mathbb{G}$, where $a, b \in_R \mathbb{Z}_q$.*

**Definition 8.** *We say that the $(t, \epsilon)$-CDH assumption holds in the group $\mathbb{G}$ if no adversary can solve the CDH problem in $\mathbb{G}$ in time $t$ with probability over $\epsilon$.*

## 4 Unique Signature Scheme

In this section, we give a simple construction for unique signatures. Our construction is based on a result due to Lysyanskaya [14], where the signature on an $n_0$-bit message consists of $\theta(n_0)$ group elements. We show that our solution gives rise to a signature of a single group element on an $n_0$-bit message.

### 4.1 Construction

We use the cryptographic hash function, one-way permutation, and bilinear map to build our unique signature scheme. The construction is described as follows:

- $\mathsf{Setup}(1^k) \rightarrow \pi$. Let $k$ be the security parameter, and $n_0$ be the message length, where $n_0 = \mathsf{poly}(k)$. Let $n$ be $2^t + 1$, and $[x]$ denote $[x]_n = x \bmod n$, where $t \in \mathbb{N}$ and $n = \theta(n_0)$. Let $q$ be a $k$-bit prime, $\mathbb{G}$ and $\mathbb{G}_\mathbb{T}$ be two multiplicative cyclic groups of prime order $q$, and $g$ be a generator of $\mathbb{G}$. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\mathbb{T}$ be an admissible bilinear map, $H : \{0,1\}^* \rightarrow \{0,1\}^{n+t-1}$ be a cryptographic hash function, and $F : \{0,1\}^{n+t-1+n_0} \rightarrow \{0,1\}^{n+t-1+n_0}$ be a one-way permutation. The system manager publishes the public parameter

$$\pi = (k, n_0, n, q, \mathbb{G}, \mathbb{G}_\mathbb{T}, g, \hat{e}, H, F) \ .$$

- $\mathsf{KeyGen}(\pi) \rightarrow (sk, pk)$. A signer randomly chooses $2n$ exponents $a_{i,j} \in_R \mathbb{Z}_q^*$ and computes $A_{i,j} = g^{a_{i,j}}$, where $i \in \mathbb{Z}_n$ and $j \in \mathbb{Z}_2$. These exponents have to satisfy the two requirements:
  1. For every $i, i' \in \mathbb{Z}_n$ and every $j, j' \in \mathbb{Z}_2$, we have $a_{i,j} = a_{i',j'}$ iff. $(i,j) = (i',j')$. It can be verified without knowing the exponents by checking whether every $A_{i,j}$ is unique.
  2. For every $h \in \{1, 2, \ldots, \frac{n-1}{2}\}$, every $i \in \mathbb{Z}_n$, and every $j, j' \in \mathbb{Z}_2$, we have $a_{i,j} + a_{[i+2h],j'} \neq 0$. It can be verified without knowing the exponents by checking whether every $A_{i,j} \times A_{[i+2h],j'} \neq 1$.

  The signer stores his secret key

$$sk = \{(a_{0,0}, a_{0,1}), (a_{1,0}, a_{1,1}), \ldots, (a_{n-1,0}, a_{n-1,1})\}$$

  and publishes his public key

$$pk = \{(A_{0,0}, A_{0,1}), (A_{1,0}, A_{1,1}), \ldots, (A_{n-1,0}, A_{n-1,1})\} \ .$$

- $\mathsf{Sign}(\pi, sk, pk, m) \rightarrow \sigma$. To sign a message $m \in \{0,1\}^{n_0}$ of $n_0$ bits[3], a signer generates the signature $\sigma$ as follows:
  1. Use his public key $pk$ and the cryptographic hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{n+t-1}$ to compute $x = H(pk \| m)$.
  2. Use the one-way permutation $F : \{0,1\}^{n+t-1+n_0} \rightarrow \{0,1\}^{n+t-1+n_0}$ to compute $y = F(x \| m)$.

[3] A cryptographic hash function $H' : \{0,1\}^* \rightarrow \{0,1\}^{n_0}$ can be used to expand the message space.

3. Use the cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^{n+t-1}$ to compute $z = H(y)$.
4. Let $h = \mathsf{LSB}_{t-1}(z)+1$, where $\mathsf{LSB}_{t-1}(z)$ is the least $t-1$ significant bits of $z$. Use his secret key $sk = \{(a_{0,0}, a_{0,1}), (a_{1,0}, a_{1,1}), \ldots, (a_{n-1,0}, a_{n-1,1})\}$ to compute signature

$$\sigma = \prod_{i=0}^{n-1} g^{a_{i,z(i)} a_{[i+h],z([i+h])}} \quad .$$

– $\mathsf{Verify}(\pi, pk, m, \sigma) \to \{\mathsf{Yes}, \mathsf{No}\}$. Suppose that the signer's public key $pk$ is well-formed. A verifier verifies a message-signature pair $(m, \sigma)$ of the signer as follows:
  1. Use the cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^{n+t-1}$ and the signer's public key $pk$ to compute $x = H(pk\|m)$.
  2. Use the one-way permutation $F : \{0,1\}^{n+t-1+n_0} \to \{0,1\}^{n+t-1+n_0}$ to compute $y = F(x\|m)$.
  3. Use the cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^{n+t-1}$ to compute $z = H(y)$.
  4. Let $h = \mathsf{LSB}_{t-1}(z)+1$, where $\mathsf{LSB}_{t-1}(z)$ is the least $t-1$ significant bits of $z$. Use the bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\mathbb{T}}$ and the signer's public key $pk = \{(A_{0,0}, A_{0,1}), (A_{1,0}, A_{1,1}), \ldots, (A_{n-1,0}, A_{n-1,1})\}$ to check whether

$$\hat{e}(\sigma, g) = \prod_{i=0}^{n-1} \hat{e}\left(A_{i,z(i)}, A_{[i+h],z([i+h])}\right) \quad .$$

*Consistency.* If the signature $\sigma$ is well-formed, then we have

$$\begin{aligned}
\hat{e}(\sigma, g) &= \hat{e}\left(\prod_{i=0}^{n-1} g^{a_{i,z(i)} a_{[i+h],z([i+h])}}, g\right) \\
&= \prod_{i=0}^{n-1} \hat{e}\left(g^{a_{i,z(i)}}, g^{a_{[i+h],z([i+h])}}\right) \\
&= \prod_{i=0}^{n-1} \hat{e}\left(A_{i,z(i)}, A_{[i+h],z([i+h])}\right)
\end{aligned}$$

*Uniqueness.* If there are two signatures $(\sigma_1, \sigma_2)$ for the same message $m$ under a secret-public key pair $(sk, pk)$, then we have

$$\hat{e}(\sigma_1, g) = \prod_{i=0}^{n-1} \hat{e}\left(A_{i,z(i)}, A_{[i+h],z([i+h])}\right) = \hat{e}(\sigma_2, g)$$

because $\sigma_1$ and $\sigma_2$ share the same $x = H(pk\|m)$, $y = F(x\|m)$, $z = H(y)$, and $h = \mathsf{LSB}_{t-1}(z) + 1$. Thus, it must be $\sigma_1 = \sigma_2$ unless $g$ is not a generator.

### 4.2 Efficiency

Let $\mathsf{Hash}$ be an execution of hash function $H$, $\mathsf{Perm}$ be an execution of one-way permutation $F$, and $\mathsf{Pair}$ be an execution of bilinear map $\hat{e}$. Let $\mathsf{Add}_{\mathbb{Z}_q}$ be the operation of addition in $\mathbb{Z}_q$, $\mathsf{Mul}_{\mathbb{Z}_q}$ be the operation of multiplication in $\mathbb{Z}_q$, $\mathsf{Exp}_{\mathbb{G}}$ be the operation of scalar exponentiation in $\mathbb{G}$, and $\mathsf{Mul}_{\mathbb{G}_\mathbb{T}}$ be the operation of multiplication in $\mathbb{G}_\mathbb{T}$. The computational complexity of algorithm $\mathsf{Sign}$ is $2\mathsf{Hash}+\mathsf{Perm}+(n-1)\mathsf{Add}_{\mathbb{Z}_q}+n\mathsf{Mul}_{\mathbb{Z}_q}+\mathsf{Exp}_{\mathbb{G}}$. The computational complexity of algorithm $\mathsf{Verify}$ is $2\mathsf{Hash}+\mathsf{Perm}+(n+1)\mathsf{Pair}+(n-1)\mathsf{Mul}_{\mathbb{G}_\mathbb{T}}$.

We now compare our construction with the related works in Table 1. The schemes [7, 11] are verifiable random functions, the scheme [6] is a strongly unforgeable signature, and the others are verifiable unpredictable functions (unique signatures). Our unique signature scheme is based on the standard CDH assumption. Our key size is the same as [14], but our signature consists of only one group element instead of $n$ elements. This differs from that of Micali et al. [15], in the size of signature as discussed in [8]. For the RSA assumption to have the same security level as DDH-based assumptions for the same input size, the signature size will grow in the order of a few hundred kilobytes. There are number of important differences between our construction and Jager [12] in both signature size and underlying hard assumption. The BLS signature [5] enjoys shorter key size besides signature size. It is also based on the standard CDH assumption. As we mentioned earlier, its group size is dominated by the output length of the employed hash function. Our construction and Boneh et al.'s [6] differ not only in signature size and key size, but also in the type of signatures. Their construction is not a unique signature. The strongly existential unforgeability does not necessarily implies the uniqueness.

### 4.3 Applications

Our scheme produces a signature of only one group element. When such a signature scheme is used with arbitrary input message length, a better bandwidth is obtained, as a shorter signature needs to be transferred.

In addition to explicit applications of unique signatures for authenticity, integrity and non-repudiation of a message, there is a natural transformation from unique signatures to VRFs by an early work of Goldreich and Levin [9]. VRFs behave similarly to pseudorandom functions, namely, giving the adversary the oracle access of the VRF function to evaluate for some input of his choice. Eventually, the adversary should not be able to distinguish the output of a VRF function from a random source. Besides, the VRF has another additional property that, given the output of the VRF function to the verifier, it is easy for the prover to non-interactively convince the verifier that the given commitment is correct with respect to prover's public key. Due to these properties, VRFs found some significant applications such as resettable zero-knowledge proofs, adaptive oblivious transfer protocols, and verifiable transaction escrow schemes.

Unique signatures have also found an important application for building a secure cryptosystem. Boneh et al. [3] proposed a conversion from a selective-

**Table 1.** Comparison with Related Work

| Scheme | Type | Assumption | Secret Key (bits) | Public Key (bits) | Output (bits) |
|---|---|---|---|---|---|
| [15] | VUF | RSA | $k$ | $(2k^2+1)k+t$ | $k$ |
| [12] | VUF | $l$-CDH | $2nk$ | $(2n+2)\ell$ | $n\ell$ |
| [14] | VUF | $l$-CDH | $2nk$ | $2n\ell$ | $n\ell$ |
| [8] | VUF | $l$-DHI | $k$ | $\ell$ | $\ell$ |
| [7] | VRF | SF-$l$-DDH | $lk$ | $l\ell$ | $l\ell$ |
| [11] | VRF | $l$-DDHE | $(n+1)k+2\ell$ | $(n+3)\ell$ | $(n+1)\ell+\ell_T$ |
| [6] | SUS | CDH | $\ell$ | $(n+5)\ell$ | $2\ell+k$ |
| [5] | VUF | CDH | $k$ | $\ell$ | $\ell$ |
| Ours | VUF | CDH | $2nk$ | $2n\ell$ | $\ell$ |

- $k$ is the security parameter
- $\ell$ is the size of an element in $\mathbb{G}$
- $\ell_T$ is the size of an element in $\mathbb{G}_T$
- $l$ is the parameter of complexity assumption
- $t$ is the size of random coin
- $n$ is the equivalent size of a message
- SUS stands for strongly unforgeable signature
- VUF stands for verifiable unpredictable function
- VRF stands for verifiable random function

identity CPA-secure IBE cryptosystem to an adaptive CCA-secure IBE cryptosystem. In this conversion, they manipulate the selective-identity CPA-secure IBE encryption function to sign the ciphertext by a one-time strongly unforgeable signature scheme. Therefore, the sender of a message $m$, first evokes $\mathsf{KeyGen}(\pi)$ to obtain verification key $vk$ and secret key $sk$. The sender then encrypts the message $m$ using recipient $pk$ and also computes $\mathsf{Sign}_{\mathsf{sk}}(\pi, sk, vk, C)$. The final ciphertext is $\langle vk, C, \sigma_c \rangle$. The recipient, after receiving $\langle vk, C, \sigma_c \rangle$, checks if $\mathsf{Verify}(\pi, vk, C, \sigma_c) = \mathsf{Yes}$ holds and then decrypts the ciphertext to obtain the message $m$.

## 5 Security Proof

The unique signature scheme is provable to achieve strongly existential unforgeability and malicious signer resistance against any probabilistic polynomial-time adversary.

### 5.1 Strongly Existential Unfogeability

We give a proof of strongly existential unforgeability in the random oracle model. Theorem 1 states that if the CDH assumption holds, the unique signature scheme achieves strongly existential unforgeability.

**Theorem 1.** *Let $k$ be the security parameter. Let $\mathcal{O}_S$ be the signing oracle of the unique signature scheme. Suppose that an adversary queries at most $q_s$ messages*

*to $\mathcal{O}_S$, and each query is handled in time $t_s$. Let $\mathcal{O}_H$ be the random oracle of hash function $H : \{0,1\}^* \to \{0,1\}^{n+t-1}$, where $n = 2^t + 1 \in \mathsf{poly}(k)$ and $n \geq \frac{q_s+3}{2}$. Suppose that an adversary queries at most $q_h$ messages to $\mathcal{O}_H$, and each query is handled in time $t_h$. If the $(t,\epsilon)$-CDH assumption holds, the unique signature scheme achieves $(t - q_h t_h - q_s t_s, q_s, 2e(n-1)\epsilon)$ strongly existential unforgeability, where $e$ is the Euler's number.*

*Proof.* Assume that adversary $\mathcal{A}$ breaks $(t', q_s, \epsilon')$ strongly existential unforgeability of the unique signature scheme. We construct an algorithm $\mathcal{B}$ to solve the CDH problem as follows:

*Setup.* Let $\pi = (k, n_0, n, q, \mathbb{G}, \mathbb{G}_\mathbb{T}, g, \hat{e}, H, F)$ be the public parameter, where $n = 2^t + 1 \geq \frac{q_s+3}{2}$. Let $(g, g^a, g^b)$ be an instance of the CDH problem in $\mathbb{G}$. To generate public key $pk = \{(A_{0,0}, A_{0,1}), (A_{1,0}, A_{1,1}), \ldots, (A_{n-1,0}, A_{n-1,1})\}$, $\mathcal{B}$ randomly chooses $h^* \in_R \{1, 2, \ldots, \frac{n-1}{2}\}$, $i^* \in_R \mathbb{Z}_n$, and $b_{i^*}, b_{[i^*+h^*]} \in_R \mathbb{Z}_2$. Then, $\mathcal{B}$ embeds $(g^a, g^b)$ in $pk$ by setting $A_{i^*,b_{i^*}} = g^a$ and $A_{[i^*+h^*],b_{[i^*+h^*]}} = g^b$. If $A_{i^*,b_{i^*}} = A_{[i^*+h^*],b_{[i^*+h^*]}}$ or $A_{i^*,b_{i^*}} A_{[i^*+h^*],b_{[i^*+h^*]}} = 1$, $\mathcal{B}$ sets $A_{i^*,b_{i^*}} = g^{a+1}$ and $C = g^b$. Otherwise, $\mathcal{B}$ sets $C = 1$. For the rest of $A_{i,j}$, $\mathcal{B}$ randomly chooses $a_{i,j} \in_R \mathbb{Z}_q^*$ and computes $A_{i,j} = g^{a_{i,j}}$ such that the public key $pk$ satisfies the requirements in algorithm $\mathsf{KeyGen}$. $\mathcal{B}$ invokes $\mathcal{A}$ as a subroutine $\mathcal{A}^{\mathcal{O}_S, \mathcal{O}_H}(\pi, pk)$.

*Query.* $\mathcal{A}$ can query at most $q_s$ messages to the signing oracle $\mathcal{O}_S$ and at most $q_h$ messages to the random oracle $\mathcal{O}_H$. $\mathcal{B}$ simulates the signing oracle $\mathcal{O}_S$ and the random oracle $\mathcal{O}_H$ as follows:

- $\mathcal{O}_H$: $\mathcal{B}$ maintains a table $\mathcal{T}_H = \{(m, H(m))\}$ to record the $\mathcal{O}_H$-queries, where $m \in \{0,1\}^*$ and $H(m) \in \{0,1\}^{n+t-1}$. $\mathcal{B}$ takes a message $m \in \{0,1\}^*$ as the input. If record $(m, H(m)) \in \mathcal{T}_H$, then $\mathcal{B}$ outputs $H(m)$ for consistency. Otherwise, $\mathcal{B}$ randomly chooses $x \in_R \{0,1\}^{n+t-1}$ and inserts $(m, x)$ into $\mathcal{T}_H$. Finally, $\mathcal{B}$ outputs $H(m) = x$.
- $\mathcal{O}_S$: $\mathcal{B}$ takes a message $m_i \in \{0,1\}^{n_0}$ as the input. Then, $\mathcal{B}$ computes $x = H(pk \| m_i)$, $y = F(x \| m_i)$, and $z = H(y)$. Let $h = \mathsf{LSB}_{t-1}(z) + 1$. If $h = h^*$, $z(i^*) = b_{i^*}$, and $z([i^* + h^*]) = b_{[i^*+h^*]}$, $\mathcal{B}$ has to abort because computing the signature $\sigma_i$ of $m_i$ is equivalent to solving the CDH problem. If $h \neq h^*$, $z(i^*) = b_{i^*}$, and $z([i^* + h^*]) = b_{[i^*+h^*]}$, $\mathcal{B}$ can compute $\sigma_i$ as follows:

$$\begin{aligned}\sigma_i = &\left(A_{i^*,z(i^*)}\right)^{a_{[i^*-h],z([i^*-h])} + a_{[i^*+h],z([i^*+h])}} \\ &\times \left(A_{[i^*+h^*],z([i^*+h^*])}\right)^{a_{[i^*+h^*-h],z([i^*+h^*-h])} + a_{[i^*+h^*+h],z([i^*+h^*+h])}} \\ &\times \prod_{i \in \mathbb{Z}_n \setminus \{i^*-h, i^*, i^*+h^*-h, i^*+h^*\}} g^{a_{i,z(i)} a_{[i+h],z([i+h])}}\end{aligned}$$

  If $z(i^*) \neq b_{i^*}$ and $z([i^*+h^*]) \neq b_{[i^*+h^*]}$, $\mathcal{B}$ can compute $\sigma_i = \mathsf{Sign}(\pi, sk, pk, m_i)$ because all the needed exponents of $sk$ are chosen by $\mathcal{B}$. For the other cases, either $z(i^*) = b_{i^*}$ or $z([i^*+h^*]) = b_{[i^*+h^*]}$, let $i' \in \{i^*, [i^*+h^*]\}$ be the index

such that $z(i') = b_{i'}$. $\mathcal{B}$ can compute $\sigma_i$ as follows:

$$\sigma_i = \left(A_{i',z(i')}\right)^{a_{[i'-h],z([i'-h])}+a_{[i'+h],z([i'+h])}} \times \prod_{i\in\mathbb{Z}_n\setminus\{i'-h,i'\}} g^{a_{i,z(i)}a_{[i+h],z([i+h])}}$$

*Forgery.* If $\mathcal{A}$ outputs a forgery $(m^*, \sigma^*)$ such that $m^* \notin \{m_1, m_2, \ldots, m_{q_s}\}$ and $\mathsf{Verify}(\pi, pk, m^*, \sigma^*) = \mathsf{Yes}$, then $\mathcal{B}$ computes $x = H(pk\|m^*)$, $y = F(x\|m^*)$, and $z = H(y)$. Let $h = \mathsf{LSB}_{t-1}(z) + 1$. If $h = h^*$, $z(i^*) = b_{i^*}$, and $z([i^* + h^*]) = b_{[i^*+h^*]}$, $\mathcal{B}$ can solve the CDH problem by computing

$$\begin{aligned} g^{ab} &= \sigma^* \times C^{-1} \times \left(A_{i^*,z(i^*)}\right)^{-a_{[i^*-h^*],z([i^*-h])}} \\ &\times \left(A_{[i^*+h^*],z([i^*+h^*])}\right)^{-a_{[i^*+2h^*],z([i^*+2h^*])}} \\ &\times \prod_{i\in\mathbb{Z}_n\setminus\{i^*-h^*,i^*,i^*+h^*\}} g^{-a_{i,z(i)}a_{[i+h],z([i+h])}} \end{aligned}$$

The remaining work is to analyze the success probability and the execution time of $\mathcal{B}$. In the query phase, $\mathcal{B}$ has to abort if any queried message results in $(z, h)$ such that $h = h^*$, $z(i^*) = b_{i^*}$, and $z([i^* + h^*]) = b_{[i^*+h^*]}$. In the forgery phase, $\mathcal{B}$ succeeds if the forgery results in $(z, h)$ such that $h = h^*$, $z(i^*) = b_{i^*}$, and $z([i^* + h^*]) = b_{[i^*+h^*]}$. Therefore, the success probability of $\mathcal{B}$ is

$$\begin{aligned} &\Pr\left[\mathcal{B}^{\mathcal{A}^{\mathcal{O}_S,\mathcal{O}_H}}(g, g^a, g^b) = g^{ab}\right] \\ &= \left(1 - \frac{2}{n-1}\cdot\frac{1}{4}\right)^{q_s} \times \epsilon' \times \frac{2}{n-1}\cdot\frac{1}{4} \\ &\geq \left(1 - \frac{1}{q_s+1}\right)^{q_s} \times \epsilon' \times \frac{1}{2(n-1)} \\ &\geq e^{-1} \times \frac{\epsilon'}{2(n-1)} \end{aligned}$$

The execution time of $\mathcal{B}$ is $t' + q_h t_h + q_s t_s$. By choosing appropriate parameters $n, q_h, t_h, q_s, t_s \in \mathsf{poly}(k)$, we complete the proof: if the $(t, \epsilon)$-CDH assumption holds, the unique signature signature scheme achieves $(t', q_s, \epsilon')$ strongly existential unforgeability, where $t' = t - q_h t_h - q_s t_s$ and $\epsilon' = 2e(n-1)\epsilon$. □

### 5.2 Malicious Signer Resistance

We define the equivalent set of a signature to prove malicious signer resistance. Given a secret key $sk = \{(a_{0,0}, a_{0,1}), ((a_{1,0}, a_{1,1}), \ldots, ((a_{n-1,0}, a_{n-1,1})\}$ and a signature $\sigma$ of the unique signature scheme, the equivalent set $E_\sigma^{(sk)}$ of the signature $\sigma$ under the secret key $sk$ is the collection of hash outputs which result in the signature $\sigma$. That is,

$$E_\sigma^{(sk)} = \left\{ z \in \{0,1\}^{n+t-1} \;\middle|\; \begin{array}{l} h = \mathsf{LSB}_{t-1}(z) + 1 \\ \prod_{i=0}^{n-1} g^{a_{i,z(i)}a_{[i+h],z([i+h])}} = \sigma \end{array} \right\} .$$

We can partition the equivalent set $E_{\sigma}^{(sk)} = \bigcup_{h=1}^{(n-1)/2} E_{\sigma,h}^{(sk)}$, where

$$E_{\sigma,h}^{(sk)} = \left\{ z \in \{0,1\}^{n+t-1} \;\middle|\; \begin{array}{l} \mathsf{LSB}_{t-1}(z) + 1 = h \\ \prod_{i=0}^{n-1} g^{a_{i,z(i)} a_{[i+h],z([i+h])}} = \sigma \end{array} \right\} .$$

A malicious signer intends to choose a secret key $sk$ which maximizes the size of an equivalent set $E_{\sigma}^{(sk)}$. Thus, the malicious signer has the largest chance to find two messages which result in the same signature $\sigma$. We give an upper bound for the size of an equivalent set $E_{\sigma}^{(sk)}$ by analyzing the size of each partition $E_{\sigma,h}^{(sk)}$. Lemma 1 states the upper bound for the size of an equivalent set.

**Lemma 1.** *Suppose that secret key $sk$ consists of $2n$ secret exponents. The size of an equivalent set $E_{\sigma}^{(sk)}$ is at most $2^{n/3+t-1}$.*

*Proof.* Our analysis has three steps:

1. For every $z, z' \in E_{\sigma,h}^{(sk)}$, if $z \neq z'$, there are at least two indexes $i', i'' \in \mathbb{Z}_n$ such that $z(i') \neq z'(i')$ and $z(i'') \neq z'(i'')$.
   We prove it by contradiction. Assume that there is only one index $i' \in \mathbb{Z}_n$ such that $z(i') \neq z'(i')$. Then, we have

$$\begin{aligned} & \prod_{i=0}^{n-1} g^{a_{i,z(i)} a_{[i+h],z([i+h])}} = \prod_{i=0}^{n-1} g^{a_{i,z'(i)} a_{[i+h],z'([i+h])}} \\ \Rightarrow\; & a_{i',z(i')} \left( a_{[i'-h],z([i'-h])} + a_{[i'+h],z([i'+h])} \right) \\ & = a_{i',z'(i')} \left( a_{[i'-h],z([i'-h])} + a_{[i'+h],z([i'+h])} \right) \\ \Rightarrow\; & a_{[i'-h],z([i'-h])} + a_{[i'+h],z([i'+h])} = 0 \;\vee\; a_{i',z(i')} = a_{i',z'(i')} \;, \end{aligned}$$

   which violates the requirements in algorithm KeyGen. Thus, the above assumption is false.
2. The size of partition $E_{\sigma,n/3}^{(sk)}$ is at most $2^{n/3}$.
   Suppose that $n$ is divisible by three. Thus, partition $E_{\sigma,n/3}^{(sk)}$ has the maximized size. $E_{\sigma,n/3}^{(sk)}$ determines a pattern for the combinations of secret exponents. The pattern is

$$\sum_{i=0}^{n/3-1} \left( \begin{array}{c} a_{i,z(i)} a_{[i+\frac{n}{3}],z([i+\frac{n}{3}])} \\ + a_{[i+\frac{n}{3}],z([i+\frac{n}{3}])} a_{[i+\frac{2n}{3}],z([i+\frac{2n}{3}])} \\ + a_{[i+\frac{2n}{3}],z([i+\frac{2n}{3}])} a_{i,z(i)} \end{array} \right) ,$$

   which has $n/3$ circular chains of three secret exponents. By the result of the first step, a circular chain of three secret exponents (three indexes) can contribute two choices to construct a segment of a hash output. Therefore, there are $2^{n/3}$ hash outputs in $E_{\sigma,n/3}^{(sk)}$.

3. No partition have size greater than $2^{n/3}$.
Suppose that the pattern of partition $E_{\sigma,h}^{(sk)}$ has $n/\ell$ circular chains of $\ell$ secret exponents. Note that $\ell$ must be odd because $n$ is odd. The result of the first step gives a trivial construction for $\left(\frac{\ell+1}{2}\right)^{n/\ell}$ hash outputs. We give a systematic construction for more hash outputs when $\ell \geq 7$. The idea is that we separate a circular chain into many linear chains, and make two hash outputs have the same sum of each linear chain. Given a hash output $z \in E_{\sigma,h}^{(sk)}$, we consider another hash output $z' \in E_{\sigma,h}^{(sk)}$. Let $\mathcal{I} = z \oplus z'$ denote the difference between $z$ and $z'$. For each circular chain $a_{i,z(i)}a_{[i+h],z([i+h])} + a_{[i+h],z([i+h])}a_{[i+2h],z([i+2h])} + \cdots + a_{[i+(\ell-1)h],z([i+(\ell-1)h])}a_{i,z(i)}$, if we choose

$$\mathcal{I}([i+2h]) = \mathcal{I}([i+5h]) = \cdots = \mathcal{I}([i+(3\lfloor \ell/3 \rfloor - 1)h]) = 0 \ ,$$

then the circular chain is divided into $\lfloor \ell/3 \rfloor$ linear chains, where $\lfloor \ell/3 \rfloor - 1$ linear chains have two variable exponents and one fixed exponent, and one linear chain has $\ell - 3\lfloor \frac{\ell}{3} \rfloor + 2$ variable exponents and one fixed exponent. The reason is that if $a_{[i+2h],z([i+2h])}$ and $a_{[i+5h],z([i+5h])}$ are fixed, then the changes of $a_{[i+3h],z([i+3h])}$ and $a_{[i+4h],z([i+4h])}$ will not propagate to the other variable exponents. Thus, we obtain two linear chains $a_{[i+2h],z([i+2h])}a_{[i+3h],z([i+3h])} + \cdots + a_{[i+4h],z([i+4h])}a_{[i+5h],z([i+5h])}$ and $a_{[i+5h],z([i+5h])}a_{[i+6h],z([i+6h])} + \cdots + a_{[i+h],z([i+h])}a_{[i+2h],z([i+2h])}$. As a result, a circular chain of $\ell$ secret exponents can contribute $2^{\lfloor \ell/3 \rfloor}$ choices to construction a segment of a hash output if $\ell \not\equiv 2 \pmod 3$, and $2^{\lfloor \ell/3 \rfloor - 1} \times 3$ choices if $\ell \equiv 2 \pmod 3$. Therefore, there are $2^{\frac{n}{\ell}\lfloor \frac{\ell}{3} \rfloor}$ hash outputs in $E_{\sigma,h}^{(sk)}$ if $\ell \not\equiv 2 \pmod 3$, and $\left(2^{\lfloor \ell/3 \rfloor - 1} \times 3\right)^{n/\ell}$ hash outputs if $\ell \equiv 2 \pmod 3$. For $\ell \not\equiv 2 \pmod 3$, we have $2^{\frac{n}{\ell}\lfloor \frac{\ell}{3} \rfloor} \leq 2^{n/3}$ clearly. For $\ell \equiv 2 \pmod 3$, we have

$$\left(2^{\lfloor \ell/3 \rfloor - 1} \times 3\right)^{n/\ell} = \left(2^{\frac{\ell-2}{3} - 1 + \log_2 3}\right)^{n/\ell} < \left(2^{\ell/3}\right)^{n/\ell} = 2^{n/3} \ .$$

An equivalent set $E_{\sigma}^{(sk)}$ has $\frac{n-1}{2}$ partitions, and each partition $E_{\sigma,h}^{(sk)}$ has size at most $2^{n/3}$. Thus, the size of an equivalent set $E_{\sigma}^{(sk)}$ is at most $\frac{n-1}{2} \times 2^{n/3} = 2^{n/3+t-1}$. □

The unique signature scheme is provable to achieve malicious signer resistance in the random oracle model. Lemma 2 states the result of malicious signer resistance.

**Lemma 2.** *Let $k$ be the security parameter. Let $\mathcal{O}_H$ be the random oracle of hash function $H : \{0,1\}^* \to \{0,1\}^{n+t-1}$, where $n = 2^t + 1 \in \mathsf{poly}(k)$. If malicious signer $\mathcal{S}$ runs in time $t_{\mathcal{S}}$ and queries at most $q_h$ messages to $\mathcal{O}_H$, then the unique signature scheme achieves $\left(t_{\mathcal{S}}, \frac{q_h(q_h-1)}{2}\left(2^{-n-t+1} + 2^{-2n/3}\right) + 3q_h \times 2^{-n-t+1}\right)$ malicious signer resistance.*

The proof of Lemma 2 is similar to the proof of Theorem 2, where random oracle $\mathcal{O}_H$ returns a uniformly random answer for each fresh oracle query and is $\left(t_{\mathcal{S}}, \frac{q_h(q_h-1)}{2} \times 2^{-n-t+1}\right)$ collision resistant.

We give a proof of malicious signer resistance under a more relaxed condition. Theorem 2 states that if the hash function is collision resistant and the one-way permutation is indeed one-way, then the unique signature achieves malicious signer resistance.

**Theorem 2.** *Let $k$ be the security parameter. Let $c$ be a positive real number, where $1/3 < c < 1$. Let $t_{\mathcal{S}}$ be the execution time of a malicious signer $\mathcal{S}$, where $t_{\mathcal{S}} \in \mathsf{poly}(k)$. Suppose that hash function $H : \{0,1\}^* \to \{0,1\}^{n+t-1}$ is $(t_H, \epsilon_H)$ collision resistant, where $n = 2^t + 1 \in \mathsf{poly}(k)$. Suppose that one-way permutation $F : \{0,1\}^{n+t-1+n_0} \to \{0,1\}^{n+t-1+n_0}$ is $(t_F, \epsilon_F)$ one-way. If we choose $\epsilon_H \leq 1 - e^{-\frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{-cn-t+1}}$, the unique signature scheme achieves $\left(t_S, \epsilon_H + \frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{(1/3-c)n} + 2\epsilon_F + t_{\mathcal{S}} \times 2^{-cn-t+1}\right)$ malicious signer resistance.*

*Proof.* Our analysis has three steps:

1. If we choose $\epsilon_H \leq 1 - e^{-\frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{-cn-t+1}}$, the output distribution of hash function $H : \{0,1\}^* \to \{0,1\}^{n+t-1}$ has min-entropy $\delta \geq cn + t - 1$.

   Suppose that the output distribution of hash function $H$ has min-entropy $\delta$. Consider the birthday attack. The probability that $\mathcal{S}$ outputs $t_{\mathcal{S}}$ messages and finds a collision of $H$ is about $1 - e^{-\frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{-\delta}}$. The hash function $H$ is $(t_H, \epsilon_H)$ collision resistant. Thus, we have

$$
\begin{aligned}
& 1 - e^{-\frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{-\delta}} \leq \epsilon_H \\
\Rightarrow\ & e^{-\frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{-\delta}} \geq 1 - \epsilon_H \\
\Rightarrow\ & -\frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{-\delta} \geq \ln(1 - \epsilon_H) \\
\Rightarrow\ & 2^{-\delta} \leq \frac{2}{t_{\mathcal{S}}(t_{\mathcal{S}}-1)} \ln \frac{1}{(1-\epsilon_H)} \\
\Rightarrow\ & \delta \geq -\lg\left(\frac{2}{t_{\mathcal{S}}(t_{\mathcal{S}}-1)} \ln \frac{1}{(1-\epsilon_H)}\right)
\end{aligned}
$$

   If we choose $\epsilon_H \leq 1 - e^{-\frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{-cn-t+1}}$, we have $\delta \geq cn + t - 1$.

2. If the malicious signer $\mathcal{S}$ chooses his secret-public key pair $(sk, pk)$ and evaluates the output of the H-F-H structure, he has probability at most $\epsilon_H + \frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{(1/3-c)n}$ to find two messages $m, m' \in_R \{0,1\}^{n_0}$ such that $m \neq m'$ and $\mathsf{Sign}(\pi, sk, pk, m) = \mathsf{Sign}(\pi, sk, pk, m')$.

   Suppose that $\mathcal{S}$ outputs $t_{\mathcal{S}}$ messages $m_1, m_2, \ldots, m_{t_{\mathcal{S}}} \in_R \{0,1\}^{n_0}$. Let $x_i = H(pk \| m_i)$, $y_i = F(x_i \| m_i)$, and $z_i = H(y_i)$. If $m_i \neq m_j$, we have $y_i \neq y_j$. If $\mathsf{Sign}(\pi, sk, pk, m_i) = \sigma = \mathsf{Sign}(\pi, sk, pk, m_j)$, we have $z_i = z_j$ or $z_i, z_j \in$

$E_\sigma^{(sk)}$. Thus, the success probability of $\mathcal{S}$ is

$$\begin{aligned}
&\Pr\left[y_i \neq y_j \wedge z_i = z_j \bigvee z_i \neq z_j \wedge z_i, z_j \in E_\sigma^{(sk)}\right] \\
&\leq \Pr\left[y_i \neq y_j \wedge z_i = z_j\right] + \Pr\left[z_i \neq z_j \wedge z_i, z_j \in E_\sigma^{(sk)}\right] \\
&\leq \epsilon_H + \frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times \frac{2^{n/3+t-1}}{2^{cn+t-1}} \\
&= \epsilon_H + \frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{(1/3-c)n}
\end{aligned}$$

3. If the malicious signer $\mathcal{S}$ tries to invert the H-F-H structure, he has probability at most $2\epsilon_F + t_{\mathcal{S}} \times 2^{-cn-t+1}$ to find a secret-public key pair $(sk, pk)$ and two messages $m, m' \in \{0,1\}^{n_0}$ such that $\mathsf{Sign}(\pi, sk, pk, m) = \mathsf{Sign}(\pi, sk, pk, m')$. The H-F-H structure has three layers. The first H-layer computes $x_i = H(pk\|m_i)$, the F-layer computes $y_i = F(x_i\|m_i)$, and the last H-layer computes $z_i = H(y_i)$. The malicious signer $\mathcal{S}$ tries to invert the H-F-H structure from an equivalent set $E_\sigma^{(sk)}$ in time $t_{\mathcal{S}}$. Note that the secret-public key pair $(sk, pk)$ is determined once the equivalent set $E_\sigma^{(sk)}$ is chosen. The first method is to determine $(sk, pk)$ by $z_i$, invert the last two layers, and pass the verification of the first H-layer. The second method is to determine $(sk, pk)$ by $y_i$, invert the F-layer, and pass the verification of the first H-layer. The third method is to determine $(sk, pk)$ by $x_i\|m_i$ and pass the verification of the first H-layer.
We give an upper bound for the probability that $\mathcal{S}$ succeeds by the first method and the second method. The success probability is

$$\begin{aligned}
&\Pr\begin{bmatrix} \mathcal{S}(\pi, pk, z_i) = y_i \in H^{-1}(z_i) \\ \mathcal{S}(\pi, pk, y_i) = x_i\|m_i \in F^{-1}(y_i) \\ H(pk\|m_i) = x_i \end{bmatrix} + \Pr\begin{bmatrix} \mathcal{S}(\pi, pk, y_i) = x_i\|m_i \in F^{-1}(y_i) \\ H(pk\|m_i) = x_i \end{bmatrix} \\
&\leq \Pr\left[\mathcal{S}(\pi, pk, y_i) = x_i\|m_i \in F^{-1}(y_i)\right] + \Pr\left[\mathcal{S}(\pi, pk, y_i) = x_i\|m_i \in F^{-1}(y_i)\right] \\
&= 2\Pr\left[\mathcal{S}(\pi, pk, y_i) = x_i\|m_i \in F^{-1}(y_i)\right]
\end{aligned}$$

The one-way permutation $F$ is $(t_F, \epsilon_F)$ one-way, where the probability is on the random choice of an input. It implies that there are at most $\epsilon_F$-portion of inputs whose outputs are invertible in time $t_F$. The one-way permutation $F$ is bijective. Therefore, there are at most $\epsilon_F$-portion of outputs that are invertible in time $t_F$. Thus, we have

$$\Pr\left[\mathcal{S}(\pi, pk, y_i) = x_i\|m_i \in F^{-1}(y_i)\right] \leq \epsilon_F \ .$$

By the result of the first step, we give an upper bound for the probability that $\mathcal{S}$ succeeds by the third method. The success probability is

$$\Pr\left[H(pk\|m_i) = x_i : \begin{matrix} \mathcal{S}(\pi) = x_i\|m_i \\ F(x_i\|m_i) = y_i \\ H(y_i) = z_i \\ z_i \rightsquigarrow (sk, pk) \end{matrix}\right] \leq t_{\mathcal{S}} \times 2^{-cn-t+1} \ .$$

Putting the above analyses together, we prove that the unique signature scheme achieves $(t_{\mathcal{S}}, \epsilon_H + \frac{t_{\mathcal{S}}(t_{\mathcal{S}}-1)}{2} \times 2^{(1/3-c)n} + 2\epsilon_F + t_{\mathcal{S}} \times 2^{-cn-t+1})$ malicious signer resistance. □

## 6 Conclusion

We propose a unique signature scheme based on the *computational Diffie-Hellman* problem on groups equipped with bilinear map. The key feature of this study is its efficiency and signature size. Our unique signature scheme produces a signature of only one group element. The security of the proposed scheme is based on the computational Diffie-Hellman assumption in the random oracle model.

The strong unforgeability of unique signature ensures that the adversary cannot even produce a valid signature for a previously signed message. This is due to the fact that the unique signature scheme is a function from the message space to the signature space. As a result, every message has only one unique signature under given public key.

## References


1. NIST Report on Cryptographic Key Length and Cryptoperiod. Tech. rep., NIST (2012)
2. Abdalla, M., Catalano, D., Fiore, D.: Verifiable random functions: Relations to identity-based key encapsulation and new constructions. J. Cryptol. 27(3), 544–593 (Jul 2014), http://dx.doi.org/10.1007/s00145-013-9153-x
3. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. 36(5), 1301–1328 (Dec 2006), http://dx.doi.org/10.1137/S009753970544713X
4. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology. pp. 213–229. CRYPTO '01, Springer-Verlag, London, UK, UK (2001), http://dl.acm.org/citation.cfm?id=646766.704155
5. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. J. Cryptol. 17(4), 297–319 (Sep 2004), http://dx.doi.org/10.1007/s00145-004-0314-9
6. Boneh, D., Shen, E., Waters, B.: Strongly unforgeable signatures based on computational diffie-hellman. In: Proceedings of the 9th International Conference on Theory and Practice of Public-Key Cryptography. pp. 229–240. PKC'06, Springer-Verlag, Berlin, Heidelberg (2006), http://dx.doi.org/10.1007/11745853_15
7. Dodis, Y.: Efficient construction of (distributed) verifiable random functions. In: Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings. pp. 1–17 (2003), http://dx.doi.org/10.1007/3-540-36288-6_1
8. Dodis, Y., Yampolskiy, A.: A verifiable random function with short proofs and keys. In: Proceedings of the 8th International Conference on Theory and Practice in Public Key Cryptography. pp. 416–431. PKC'05, Springer-Verlag, Berlin, Heidelberg (2005), http://dx.doi.org/10.1007/978-3-540-30580-4_28

9. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing. pp. 25–32. STOC '89, ACM, New York, NY, USA (1989), http://doi.acm.org/10.1145/73007.73010
10. Goldwasser, S., Ostrovsky, R.: Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology. pp. 228–245. CRYPTO '92, Springer-Verlag, London, UK, UK (1993), http://dl.acm.org/citation.cfm?id=646757.705546
11. Hohenberger, S., Waters, B.: Constructing verifiable random functions with large input spaces. In: Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques. pp. 656–672. EUROCRYPT'10, Springer-Verlag, Berlin, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-13190-5_33
12. Jager, T.: Verifiable random functions from weaker assumptions. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II. pp. 121–143 (2015), http://dx.doi.org/10.1007/978-3-662-46497-7_5
13. Kuchta, V., Manulis, M.: Unique aggregate signatures with applications to distributed verifiable random functions. In: Cryptology and Network Security - 12th International Conference, CANS 2013, Paraty, Brazil, November 20-22. 2013. Proceedings. pp. 251–270 (2013), http://dx.doi.org/10.1007/978-3-319-02937-5_14
14. Lysyanskaya, A.: Unique signatures and verifiable random functions from the dh-ddh separation. In: Proceedings of the 22Nd Annual International Cryptology Conference on Advances in Cryptology. pp. 597–612. CRYPTO '02, Springer-Verlag, London, UK, UK (2002), http://dl.acm.org/citation.cfm?id=646767.704299
15. Micali, S., Rabin, M., Vadhan, S.: Verifiable random functions. In: Proceedings of the 40th Annual Symposium on Foundations of Computer Science. pp. 120–. FOCS '99, IEEE Computer Society, Washington, DC, USA (1999), http://dl.acm.org/citation.cfm?id=795665.796482