INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2020, VOL. 66, NO. 3, PP. 473-480
Manuscript received February 25, 2020; revised July, 2020. DOI: 10.24425/ijet.2020.131901

Improving Security of Existentially Unforgeable Signature Schemes

Mariusz Jurkiewicz

***Abstract*—In this paper we present a family of transforms that map existentially unforgeable signature schemes to signature schemes being strongly unforgeable. In spite of rising security, the transforms let us make a signature on a union of messages at once. The number of elements in this union depends on the signing algorithm of a scheme being transformed. In addition to that we define an existentially unforgeable signature scheme based on pairings, which satisfies all assumptions of the first part and is able to be subjected to transformation.**

***Keywords*—Digital signature scheme (DSS); strongly unforgeable DSS; pairing**

I. Introduction

THE notion od existentially unforgeable signature schemes was introduced in [3]. For these schemes, an adversary has almost unbounded access to the signing oracle of a scheme which is being attacked, and may adaptively interact with this oracle, requesting signatures on messages $m^1, \ldots, m^q$ generated by itself. Apart from that, it is assumed the adversary knows the public key all the time. In spite of having all this knowledge and control over oracle, it is able to output a valid forgery only with negligible probability. Here, the forgery means a positively verifiable pair $(m^\star, \sigma^\star)$ such that $m^\star \notin \{m_1, \ldots, m_q\}$.

The notion of existential unforgeability seems to be strong, but it turns out that it can be strengthen. This stronger version is termed as strong unforgeability (see [1]), and differs from existential unforgeability in one detail. Namely, if σ^i is a signature on m^i then it is negligibly likely to output a valid forgery $(m^\star, \sigma^\star) \notin \{(m^i, \sigma^i)\}$. In addition to the security given by existential unforgeability, this time a scheme is also prevented from a new signature on a message that has been already signed. This security is especially important to protect systems against some types of attacks, such as replay attacks for instance.

In this paper we show how to transform signature schemes that are existentially unforgeable and their signing algorithms satisfy some additional conditions into strongly unforgeable signature schemes in a classical model. We were inspired by the work [4] regarding an Identity-Based-Encryption scheme, which can be transformed into a signature scheme by using the generic Boneh-Franklin method described in [5].

We also construct a flexible signature scheme which is existentially unforgeable and is not strongly unforgeable, and belongs to the domain of our transform. This scheme can serve as an example of applying the first part of this paper.

M. Jurkiewicz is with Faculty of Cybernetics, Military University of Technology, Warsaw, Poland (e-mail: mariusz.jurkiewicz@wat.edu.pl).

II. Preliminaries

Before presenting the main results, we briefly review the definitions of regular and strong CMA-security for digital signature schemes. We additionally recall some facts about the notion of bilinear maps in finite groups.

A. Notation

Following the work of [3], we introduce some notation used in this paper. Let $\mathcal{A}$ be a probabilistic Turing machine. The notations $y \leftarrow \mathcal{A}(x)$ and $y \xleftarrow{\$} \mathcal{A}(x)$ mean that a random tape is selected with some distribution and uniformly at random, and then y is assigned the value $\mathcal{A}(x, \mathsf{tape})$. When $\mathcal{A}$ is deterministic we write $y = \mathcal{A}(x)$. Analogously, if A is a finite set, then $y \xleftarrow{\$} A$ denotes that y is assigned an element uniformly and independently chosen from A. For a positive integer k, we use the following notation $[k] := \{1, \ldots, k\}$.

B. Pairings

We first define two groups $\mathbb{G}, \mathbb{G}_T$ (each with multiplicative notation), all of prime order.

Definition 1: A map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ having the following properties:

1. bilinearity, i.e., for all $g_1, g_2 \in \mathbb{G}$ and $a, b \in \mathbb{Z}_r$ we have
$$\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab};$$
2. non-degeneracy, i.e., for $g_1, g_2 \neq 1_{\mathbb{G}}$,
$$\hat{e}(g_1, g_2) \neq 1_{\mathbb{G}_T};$$

is called a *pairing of type 1* (see [2]).

C. Security and strong security against adaptive CMA

Below we recall the formal definition of a signature scheme.

Definition 2: A signature scheme is composed of three algorithms $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ along with an associated message space $\mathcal{M} = \{M_n\}_n$ such that:

- Gen is a probabilistic polynomial time algorithm (PPT). It takes as input a security parameter 1^n and outputs a key (sk, pk), where sk and pk are called, the *secret key* and the *public key* respectively. We assume that both sk and pk depend on the security parameter n (i.e. $sk = sk(n)$ and $pk = pk(n)$).
- The *signing algorithm* Sign is a probabilistic polynomial time one. It takes as input a secret key sk and a message

$m \in M_n$ and outputs a signature σ. It is written as $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$.
- The *verification algorithm* Vrfy is a deterministic polynomial time one. It takes as input a public key pk, a message $m \in M_n$ and a (purported) signature σ. It outputs a single bit b, with $b = 1$ meaning *accept* and $b = 0$ meaning *reject*. It is written as $b = \mathsf{Vrfy}_{pk}(m, \sigma)$.

We now define security against an attack in which the adversary knows pk and is provided with full access to $\mathsf{Sign}_{sk}(\cdot)$. In this context, $\mathsf{Sign}_{sk}(\cdot)$, together with the fixed secret key is called a signing oracle. This oracle should be thought of as a „black box" that outputs a signature for a message given on input. The adversary itself is able to adaptively interact with the oracle by choosing messages to be signed, depending on both the public key as well as on any signatures it has previously obtained.

We start with the weaker version of intended security model. Toward the formal definition, consider the following experiment for a signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, an adversary $\mathcal{A}$ and a value n for the security parameter.

$\mathbf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Pi}(1^n)$:

1. Generate $(sk, pk) \leftarrow \mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ is given pk and access to an oracle $\mathsf{Sign}_{sk}(\cdot)$, requesting signatures on as many messages as it likes (it is denoted by $\mathcal{A}^{\mathsf{Sign}_{sk}(\cdot)}(pk)$). Let $\{m^i\}_{i=1}^q$ be the set of queries that $\mathcal{A}$ has asked the oracle.
3. Eventually, $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}_{sk}(\cdot)}(pk)$.
4. $\mathcal{A}$ succeeds if $\mathsf{Vrfy}_{pk}(m^*, \sigma^*) = 1 \wedge m^* \notin \{m^i\}_{i=1}^q$. In this case, the output of the experiment is defined to be 1. Otherwise, the experiment outputs 0.

We refer to such an adversary as an euf-cma adversary. The advantage of the adversary $\mathcal{A}$ in attacking the scheme Π is defined as

$$\mathbf{Adv}^{\mathsf{euf\text{-}cma}}_{\Pi}(\mathcal{A}, n) = \Pr[\mathbf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Pi}(1^n) = 1].$$

A signature scheme is secure if no efficient adversary can succeed in the above game with non-negligible probability.

Definition 3: A signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, is called to be *existentially unforgeable under a chosen-message attack* if for all efficient probabilistic, polynomial-time adversaries $\mathcal{A}$, there is a negligible function negl such that

$$\mathbf{Adv}^{\mathsf{euf\text{-}cma}}_{\Pi}(\mathcal{A}, n) \leqslant \mathsf{negl}(n).$$

Signature schemes which are existentially unforgeable under a chosen-message attack are often called euf-cma *secure*

Below, we introduce a stronger notion of security for signature schemes. In addition to a forgery made by a eu-cma adversary, here the adversary is said to produce a forgery even if it outputs a new and valid signature on a previously signed message. This means that a forgery occurs whenever the adversary outputs a valid pair $(m^*, \sigma^*) \notin \{(m^i, \sigma^i)\}_{i=1}^q$, where σ^i is a signature obtained from $\mathsf{Sign}_{sk}(\cdot)$ on input m^i.

$\mathbf{Exp}^{\mathsf{suf\text{-}cma}}_{\mathcal{A},\Pi}(1^n)$:

1. Generate $(sk, pk) \leftarrow \mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ is given pk and access to an oracle $\mathsf{Sign}_{sk}(\cdot)$, requesting signatures on as many messages as it likes. Let $\{(m^i, \sigma^i)\}_{i=1}^q$ be the set of all pairs of query/answer that $\mathcal{A}$ has asked and obtained from the oracle.
3. Eventually, $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}_{sk}(\cdot)}(pk)$.
4. $\mathcal{A}$ succeeds if $\mathsf{Vrfy}_{pk}(m^*, \sigma^*) = 1 \wedge (m^*, \sigma^*) \notin \{(m_i, \sigma_i)\}_{i=1}^q$. In this case the output of the experiment is defined to be 1. Otherwise, the experiment outputs 0.

An adversary like this is referred to as suf-cma adversary. Moreover, its advantage in attacking the scheme Π is defined as

$$\mathbf{Adv}^{\mathsf{suf\text{-}cma}}_{\Pi}(\mathcal{A}, n) = \Pr[\mathbf{Exp}^{\mathsf{suf\text{-}cma}}_{\mathcal{A},\Pi}(1^n) = 1].$$

Definition 4: A signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, is called to be *strongly unforgeable under a chosen-message attack* if for all efficient probabilistic, polynomial-time adversaries $\mathcal{A}$ there is a negligible function negl such that

$$\mathbf{Adv}^{\mathsf{suf\text{-}cma}}_{\Pi}(\mathcal{A}, n) \leqslant \mathsf{negl}(n).$$

Signature schemes which are strongly unforgeable under a chosen-message attack are also called suf-cma *secure*

D. Uniform distribution

In this subsection we formulate a formal definition of a uniform distribution in a finite set. Even though this topic is viewed as being trivial, it plays an important role herein.

If Ω is a finite and nonempty set, then the family of sets over Ω, i.e. 2^Ω is obviously a σ-field. Then, it is easily seen that the map $\mu : 2^\Omega \to [0, 1]$, given by the formula

$$\mu(A) = \frac{\#A}{\#\Omega},$$

is a probability measure.

Definition 5: The measure μ is called the *uniform distribution* in Ω.

Herein, we take either a finite group of prime order or a finite prime field as Ω. The next simple lemma turns out to be useful in explaining the details of picking random elements while conducting security proofs.

Lemma 6: Let $\Omega = \mathbb{F}_p$ with p being a prime. Then, for every $A \in 2^{\mathbb{F}_p}$ and fixed $\gamma \in \mathbb{F}_p$, we have the following equality

$$\mu(A) = \mu(\gamma + A).$$

Proof. Obviously $A + \gamma = \{\alpha + \gamma \mid \alpha \in A\}$ is an element of the σ-field $2^{\mathbb{F}_p}$ (as one knows $\alpha + \gamma$ must be viewed as $\alpha + \gamma \pmod p$). Furthermore, it is easily seen that the map $\varphi_\gamma(\alpha) = \alpha + \gamma$ is a bijection inside $\mathbb{F}_p$, therefore

$$\mu(A) = \frac{\#A}{\#\mathbb{F}_p} = \frac{\#\varphi_\gamma(A)}{\#\mathbb{F}_p} = \mu(\varphi_\gamma(A)).$$

This finishes the proof. □

III. Transforms improving security

A. A domain associated to a single transform

Let $\{\Theta_\xi\}_{\xi\in\mathbb{N}}$ denote a family of sets, where a single Θ_ξ consists of all signatures schemes such that for every $\Pi \in \Theta_\xi$, the following properties are fulfilled.

(i) Π is euf-cma secure;
(ii) Π is not suf-cma secure.
(iii) An associated message space $\mathcal{M} = \{M_n\}_n$ is such that $\{0,1\}^{|p|} \subseteq M_n$, where $|p| = \lfloor \lg p \rfloor + 1$ is a bit-length of p and p is taken from a set $\mathcal{P}(n)$, being combined of all primes depending on value n for the security parameter.
(iv) Sign can be viewed as a deterministic function $\Big((r_1)_{i=1}^{\xi}, sk, m\Big) \mapsto \Big(\varphi_1(r_1, sk), \ldots, \varphi_\xi(r_\xi, sk), \varphi_{\xi+1}((r_1)_{i=1}^{\xi}, sk, m)\Big)$, where $r_i \in \mathcal{R}_\Pi$ is a randomness, sk is such that $(sk, \cdot) \leftarrow \mathsf{Gen}(1^n)$ and $m \in M_n$. Moreover, a signature itself is generated in the following manner:
   1. **for** $i = 1$ to ξ **do**
   2. $r_i \xleftarrow{\chi} \mathcal{R}_\Pi$ and $\sigma_i{=}\varphi_i(r_i, sk)$
   3. **end for**
   4. $\sigma_{\xi+1}{=}\varphi_{\xi+1}\Big((r_i)_{i=1}^{\xi}, sk, m\Big)$
   5. **return** $\sigma = (\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1})$
(v) Let (sk, pk) be fixed. Then, for given $m \in M_n$ and $(\sigma_i)_{i=1}^{\xi}$, there exists at most one component $\sigma_{\xi+1}$, such that the pair $(m, (\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1}))$ is valid, i.e $\mathsf{Vrfy}_{pk}(m, \sigma) = 1$, where $\sigma = (\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1})$.

Note that the condition (iii) implies that for a suitable prime p, each element of $\mathbb{F}_p$ has a unique representation in an appropriate message space. Therefore, it is able to be signed by the signature algorithm of Π.

B. Construction of a single transform

In this section, we define a family of transforms $\{\mathcal{T}_\xi\}_{\xi\in\mathbb{N}}$ mapping Θ_ξ to Ξ_ξ, where the latter is a set of schemes with a message space $\hat{\mathcal{M}} = \{\hat{M}_n\}$, where $\hat{M}_n = \times_{i=1}^{\xi} M_{n,i}$.

Let ξ be fixed and n be a security parameter. Assume further that $\mathscr{G}_\mathcal{F}$ is an efficient algorithm that on input 1^n outputs $(\mathbb{G}, p, g, \mathcal{H})$, where:

- $\mathbb{G}$ is a cyclic group of prime order $p \in \mathcal{P}(n)$, where the group operation in $\mathbb{G}$ can be performed efficiently.
- g is chosen uniformly at random from the set of all generators of $\mathbb{G}$.
- $\mathcal{H} = \{H_k : \{0,1\}^* \to \{0,1\}^\ell \mid k \in K\}$ is a keyed family of collision resistant hash functions ([1]). In addition to collision resistance, we need the output length of the functions to be less than p, i.e. $2^\ell \leqslant p$. This allows us to equate values of the hash functions with elements of $\mathbb{F}_p$.

Let $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme belonging to Θ_ξ. We construct the signature scheme $\mathcal{T}_\xi(\Pi) = \Pi^* = (\mathsf{Gen}^*, \mathsf{Sign}^*, \mathsf{Vrfy}^*)$, whose message space is $\hat{\mathcal{M}}$, as follows:

Key generation: Algorithm $\mathsf{Gen}^*(1^n)$ is defined as follows:

1. Compute $\mathsf{params} := (\mathbb{G}, p, g, \mathcal{H}) \leftarrow \mathscr{G}_\mathcal{F}(1^n)$.
2. Choose g_1 uniformly at random from the set of all generators of $\mathbb{G}$.
3. Pick out $k \xleftarrow{\$} K$, being a key for a function from $\mathcal{H}$.
4. Run $\mathsf{Gen}(1^n)$ to obtain (sk, pk).
5. The secret key is $sk^* = sk$ and the public key is $pk^* = (\mathsf{params}, pk, g_1, k)$.

Signature generation: For $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_\xi) \in \hat{M}_n$, algorithm $\mathsf{Sign}^*_{sk*}(\mathbf{m})$ is defined as follows:

1. Select $u \xleftarrow{\$} \mathbb{F}_p^*$.
2. For each $i = 1, \ldots \xi$, choose $r_i \xleftarrow{\chi} \mathcal{R}_\Pi$ and compute $\sigma_i{\leftarrow}\varphi_i(r_i, sk)$.
3. Compute $h \leftarrow H_k(\mathbf{m}_1 \| \sigma_1 \| \cdots \| \mathbf{m}_\xi \| \sigma_\xi)$, where "$\|$" denotes concatenation; $h \in \{0,1\}^\ell$. Thus, according to the condition $2^l \leqslant p$, it is viewed as an element of $\mathbb{F}_p$.
4. Compute $m = g^h g_1^u$.
5. Compute $\sigma_{\xi+1}{=}\varphi_{\xi+1}\Big((r_i)_{i=1}^{\xi}, sk, m\Big)$.
6. Output the signature $\sigma = (\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1}, \sigma_{\xi+2})$, where $\sigma_{\xi+2} = u$.

Signature verification: Algorithm $\mathsf{Vrfy}^*_{pk*}(\mathbf{m}, \sigma)$ is defined as follows:

1. Parse pk^* as $(\mathsf{params}, pk, g_1, k)$ and σ^* as $(\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1}, \sigma_{\xi+2})$.
2. Compute $\overline{h} = H_k(\mathbf{m}_1 \| \sigma_1 \| \cdots \| \mathbf{m}_\xi \| \sigma_\xi)$.
3. Compute $\overline{m} = g^{\overline{h}} g_1^{\sigma_{\xi+2}}$
4. Output 1 if and only if $(\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1})$ is a valid signature on $\overline{m}$ (with respect to pk); i.e., output 1 if and only if $\mathsf{Vrfy}_{pk}(\overline{m}, (\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1})) \stackrel{?}{=} 1$. Otherwise, it outputs 0.

C. Security

As we base our proof on the hardness of solving the discrete logarithm problem (DLP) in finite groups, we briefly recall this notion, fitting it into the considered case.

Definition 7: The *DLP is hard relative to* $\mathscr{G}_\mathcal{F}$, if for all PPT adversaries $\mathcal{A}$ and for all polynomials $p = p(n)$ $(\in \mathcal{P}(n))$, there exists a negligible function negl, such that

$$\Pr\left[\begin{array}{l}(\mathbb{G}, p, g, \mathcal{H}) \leftarrow \mathscr{G}_\mathcal{F}(1^n); g_2 \xleftarrow{\$} \mathbb{G}; \\ \quad x \leftarrow \mathcal{A}(1^k, \mathbb{G}, p, g, \mathcal{H}, g_2)\end{array} \mid g^x = g_2\right] \leqslant \mathsf{negl}(n).$$

Analogously, we formulate the strict definition of collision resistant family of hash functions.

Definition 8: The *collision resistance* property is satisfied relative to $\mathscr{G}_\mathcal{F}$, if for all PPT adversaries $\mathcal{A}$ and for all polynomials $p = p(n)$ $(\in \mathcal{P}(n))$, the probability

$$\Pr\left[\begin{array}{l}(\mathbb{G}, p, g, \mathcal{H}) \leftarrow \mathscr{G}_\mathcal{F}(1^n); k \xleftarrow{\$} K; \\ (x_1, x_2) \leftarrow \mathcal{A}(1^k, \mathbb{G}, p, g, \mathcal{H}, k), \\ \quad x_1 \neq x_2\end{array} \mid H_k(x_1) = H_k(x_2)\right]$$

is negligible.

[1] Here the collision resistance is understood in the sense of Definition 8.

Now, we are ready to formulate the main result of this section.

Theorem 9: If both the discrete logarithm problem is hard and the collision resistance property is satisfied relative to $\mathscr{G}_{\mathcal{F}}$, then for every signature scheme $\Pi \in \Theta_\xi$, a signature scheme $\Pi^* = \mathcal{T}_\xi(\Pi)$ is $\mathsf{suf\text{-}cma}$ secure.

Proof. Turning to the formal proof, let $\mathcal{A}^*$ be a PPT adversary attacking Π^* and, as in Section II-C, denote by $(\mathbf{m}^\star, \sigma^\star) \leftarrow \mathbf{Exp}^{\mathsf{suf\text{-}cma}}_{\mathcal{A}^*,\Pi^*}(1^n)$ the experiment

$$(sk^*, pk^*) \leftarrow \mathsf{Gen}^*(1^n);\ (\mathbf{m}^\star, \sigma^\star) \leftarrow (\mathcal{A}^*)^{\mathsf{Sign}^*_{sk^*}(\cdot)}(pk^*)$$

Since $\mathcal{A}^*$ runs in polynomial time, a map $q = q(m)$ describing the maximum number of queries made by $\mathcal{A}^*$ to its signing oracle on security parameter n, is well-defined polynomial. Therefore, we can assume without loss of generality that $\mathcal{A}^*$ always makes exactly this number of queries. In a given execution of $\mathbf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A}^*,\Pi^*}(1^n)$, let $\mathbf{m}^i = (\mathbf{m}^i_1, \ldots, \mathbf{m}^i_\xi)$ denote the ith message submitted by $\mathcal{A}^*$ to its signing oracle, and let $\sigma^i = (\sigma^i_1, \ldots, \sigma^i_\xi, \sigma^i_{\xi+1}, \sigma^i_{\xi+2})$ denotes the ith signature obtained in return. Denote by $\mathbf{m}^\star = (\mathbf{m}^\star_1, \ldots, \mathbf{m}^\star_\xi)$ and $\sigma^\star = (\sigma^\star_1, \ldots, \sigma^\star_\xi, \sigma^\star_{\xi+1}, \sigma^\star_{\xi+2})$, a pair message-signature output by $\mathcal{A}^*$. Further, let m^i and $m^\star$ denote messages computed in step 4 of signature generation, for $\mathbf{m}^i$ and $\mathbf{m}^\star$ respectively. Similarly, let h^i and $h^\star$ denote, hashes computed in step 3 of signature generation, respectively, for $\mathbf{m}^i$ and $\mathbf{m}^\star$.

Let $\mathsf{Forge}_{\mathcal{A}^*}$ be the event that $\mathsf{Vrfy}^*_{pk^*}(\mathbf{m}^\star, \sigma^\star) = 1$ and $(\mathbf{m}^\star, \sigma^\star) \notin \{(\mathbf{m}^i, \sigma^i)\}_{i=1}^q$ Then the advantage of the adversary can be defined as follow:

$$\mathbf{Adv}^{\mathsf{euf\text{-}cma}}_{\Pi^*}(\mathcal{A}^*, n) = \Pr[(\mathbf{m}^\star, \sigma^\star) \leftarrow \mathbf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A}^*,\Pi^*}(1^n) \mid \mathsf{Forge}].$$

Note that the above is equivalent to the formulation of the advantage from Section II-C. Thus according to Definition 4, it is sufficient to show that $\mathbf{Adv}^{\mathsf{euf\text{-}cma}}_{\Pi^*}(\mathcal{A}^*, n)$ is negligible.

We split Forge into the three following events:

- Let $\mathsf{Forge\text{-}m}$ be the event that none of m^i equals $m^\star$, i.e. $m_i \neq m^*$, for all $i \in [q]$, and define

$$\mathbf{Succ}^{\mathsf{Forge\text{-}m}}_{\Pi^*,\mathcal{A}^*}(n) = \Pr[(\mathbf{m}^\star, \sigma^\star) \leftarrow \mathbf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A}^*,\Pi^*}(1^n) \mid \mathsf{Forge} \cap \mathsf{Forge\text{-}m}].$$

- Let $\mathsf{Forge\text{-}coll}$ be the event that $m^\star = m^i$ and $h^\star = h^i$ for some $i \in [q]$, and define

$$\mathbf{Succ}^{\mathsf{Forge\text{-}coll}}_{\Pi^*,\mathcal{A}^*}(n) = \Pr[(\mathbf{m}^\star, \sigma^\star) \leftarrow \mathbf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A}^*,\Pi^*}(1^n) \mid \mathsf{Forge} \cap \mathsf{Forge\text{-}coll}].$$

- Similarly, let $\mathsf{Forge\text{-}dlp}$ denote the event complementary to $\mathsf{Forge\text{-}coll}$, which means that $m^\star = m^i$ and $h^\star \neq h^i$ for some $i \in [q]$, and define

$$\mathbf{Succ}^{\mathsf{Forge\text{-}dlp}}_{\Pi^*,\mathcal{A}^*}(n) = \Pr[(\mathbf{m}^\star, \sigma^\star) \leftarrow \mathbf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A}^*,\Pi^*}(1^n) \mid \mathsf{Forge} \cap \mathsf{Forge\text{-}dlp}].$$

It is easily seen that the events $\mathsf{Forge\text{-}m}$, $\mathsf{Forge\text{-}coll}$, $\mathsf{Forge\text{-}dlp}$ are pairwise disjoint, and Forge is their union. Therefore, we have

$$\mathbf{Adv}^{\mathsf{euf\text{-}cma}}_{\Pi^*}(\mathcal{A}^*, n) = \mathbf{Succ}^{\mathsf{Forge\text{-}m}}_{\Pi^*,\mathcal{A}^*}(n) + \mathbf{Succ}^{\mathsf{Forge\text{-}coll}}_{\Pi^*,\mathcal{A}^*}(n) + \mathbf{Succ}^{\mathsf{Forge\text{-}dlp}}_{\Pi^*,\mathcal{A}^*}(n). \quad (1)$$

Claim 1. $\mathbf{Succ}^{\mathsf{Forge\text{-}m}}_{\Pi^*,\mathcal{A}^*}(n)$ is negligible.

Indeed, we construct a PPT adversary $\mathcal{A}^{\mathsf{m}}$ using $\mathcal{A}^*$ as a subroutine, which launches a strong existential chosen-message attack on the scheme Π, and has success probability exactly $\mathbf{Succ}^{\mathsf{Forge\text{-}m}}_{\Pi^*,\mathcal{A}^*}(n)$.

The adversary $\mathcal{A}^{\mathsf{m}}$ starts a game according to $\mathbf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Pi}(1^n)$. At the beginning, the challenger runs $\mathsf{Gen}(1^n)$ to get a pair of keys (sk, pk) and sends pk to $\mathcal{A}$, which can interact with the signing oracle Sign_{sk}, asking adaptively q queries. The algorithm $\mathcal{A}^{\mathsf{m}}$ conducts the following steps:

Algorithm $\mathcal{A}^{\mathsf{m}}$:
The algorithm is given the pair of keys (sk, pk).

1. Run $\mathscr{G}_{\mathcal{F}}(1^n)$ to generate parameters $(\mathbb{G}, p, g, \mathcal{H})$.
2. Pick out $a \xleftarrow{\$} \mathbb{F}^*_p$ and compute $g_1 = g^a$.
3. Choose $k \xleftarrow{\$} \mathcal{K}$.
4. Create a public key $pk^* = (\mathbb{G}, p, g, \mathcal{H}, pk, g_1, k)$ of Π^*, which is given to $\mathcal{A}^*$.
5. When $\mathcal{A}^*$ requests a signature on the ith message $\mathbf{m}^i = (\mathbf{m}^i_1, \ldots, \mathbf{m}^i_\xi)$, do:
   5.1. Pick $\gamma \leftarrow \mathbb{F}^*_p$ and compute $m^i = g^\gamma$.
   5.2. Send m^i to the signing oracle Sign_{sk} and receive in return a signature $\underline{\sigma}^i = (\sigma^i_1, \ldots, \sigma^i_\xi, \sigma^i_{\xi+1})$.
   5.3. Compute $h^i \leftarrow H_k(\mathbf{m}^i_1 \| \sigma^i_1 \| \cdots \| \mathbf{m}^i_\xi \| \sigma^i_\xi)$.
   5.4. In the body $\mathbb{F}_p$, compute $u^i = (\gamma - h^i) \cdot a^{-1}$.
   5.5. Return the signature $\sigma^i = (\sigma^i_1, \ldots, \sigma^i_\xi, \sigma^i_{\xi+1}, u^i)$ to $\mathcal{A}^*$
6. When the algorithm $\mathcal{A}^*$ outputs $(\mathbf{m}^\star, \sigma^\star)$, parse $\sigma^\star = (\sigma^\star_1, \ldots, \sigma^\star_\xi, \sigma^\star_{\xi+1}, \sigma^\star_{\xi+2})$.
7. Compute $h^\star = H_k(\mathbf{m}^\star_1 \| \sigma^\star_1 \| \cdots \| \mathbf{m}^\star_\xi \| \sigma^\star_\xi)$.
8. Compute $m^\star = g^{h^\star} g_1^{\sigma^\star_{\xi+2}}$.
9. Output $(m^\star, \underline{\sigma}^\star = (\sigma^\star_1, \ldots, \sigma^\star_\xi, \sigma^\star_{\xi+1}))$

Let A denote the event where $\mathcal{A}^{\mathsf{m}}$ is able to make a forgery. Observe first that if Forge occurs, then $(\mathbf{m}^\star, \sigma^\star)$, outputted by $\mathcal{A}^*$ in step 6, does not belong to the set $\{(\mathbf{m}^i, \sigma^i)\}_{i=1}^q$, and the pair $(m^\star, \underline{\sigma}^\star)$, returned by $\mathcal{A}^{\mathsf{m}}$ in step 9, is positively verified by Vrfy^*_{pk}. Similarly, if in addition to that, the event $\mathsf{Forge\text{-}m}$ occurs, we are assured that $m^\star$ is not an element of $\{m^i\}_{i=1}^q$. This implies that A occurs and that, in particular, $\mathsf{Forge} \cap \mathsf{Forge\text{-}m}$ can be viewed as a subset of A. Therefore

$$\mathbf{Succ}^{\mathsf{Forge\text{-}m}}_{\Pi^*,\mathcal{A}^*}(n) \leqslant \Pr[\mathsf{A}] \leqslant \mathsf{negl}(n),$$

according to the assumption that Π is $\mathsf{euf\text{-}cma}$ secure.

Claim 2. $\mathbf{Succ}^{\mathsf{Forge\text{-}coll}}_{\Pi^*,\mathcal{A}^*}(n)$ is negligible.

To prove this claim, we construct a PPT algorithm $\mathcal{A}^{\mathsf{coll}}$, using $\mathcal{A}^*$ as a subroutine and attacking collision resistance of $\mathcal{H}$.

The game starts and the challenger runs $\mathcal{G}_{\mathcal{F}}(1^n)$ to generate parameters $(\mathbb{G}, p, g, \mathcal{H})$, and next it chooses $k \in K$ uniformly at random.

Algorithm $\mathcal{A}^{\mathsf{coll}}$:
The algorithm is given the parameters $(\mathbb{G}, p, g, \mathcal{H})$ and the hash key $k \in K$.

1. Run $\mathsf{Gen}^*(1^n)$ to generate keys (sk^*, pk^*).
2. The public key is sent to $\mathcal{A}^*$, which caries out a strong adaptive chosen-message attack on Π^*, and eventually outputs $(\mathbf{m}^\star, \sigma^\star = (\sigma_1^\star, \ldots, \sigma_\xi^\star, \sigma_{\xi+1}^\star, \sigma_{\xi+2}^\star)$ to $\mathcal{A}^{\mathsf{coll}}$.
3. If$\mathcal{A}^*$ outputs a valid forgery $(\mathbf{m}^\star, \sigma^\star)$, then do:
   3.1 If there is $i \in [q]$ such that $m^\star = m^i$ and $h^\star = h^i$, then output a pair $(\mathbf{m}_1^\star \| \sigma_1^\star \| \cdots \| \mathbf{m}_\xi^\star \| \sigma_\xi^\star, \quad \mathbf{m}_1^i \| \sigma_1^i \| \cdots \| \mathbf{m}_\xi^i \| \sigma_\xi^i)$.
   3.2 Otherwise, abort.
4. Otherwise, abort.

Denote by B the event where $\mathcal{A}^{\mathsf{coll}}$ finds a collision. We see that if Forge occurs, then the condition from step 3 is satisfied and if, in addition to that, the event $\mathsf{Forge\text{-}coll}$ also occurs then, according to step 3.2, $\mathcal{A}^{\mathsf{coll}}$ outputs two points, namely $x_1 = \mathbf{m}_1^\star \| \sigma_1^\star \| \cdots \| \mathbf{m}_\xi^\star \| \sigma_\xi^\star$ and $x_2 = \mathbf{m}_1^i \| \sigma_1^i \| \cdots \| \mathbf{m}_\xi^i \| \sigma_\xi^i)$. Additionally, note that the event $\mathsf{Forge\text{-}coll}$ implies $m^\star = m^i$ and $h^\star = h^i$. The former means that $H_k(x_1) = H_k(x_2)$. Hence, if $x_1 \neq x_2$, then we would have a collision, and B would occur as well. Assuming this is the case, the union $\mathsf{Forge} \cap \mathsf{Forge\text{-}coll}$ can be viewed as a subset of B, we obtain

$$\mathbf{Succ}^{\mathsf{Forge\text{-}coll}}_{\Pi^*, \mathcal{A}^*}(n) \leqslant \Pr[\mathsf{B}].$$

Due to the assumption of collision resistance property of $\mathcal{H}$, the above probability is negligible. This implies that to complete the proof of Claim 2, it is sufficient to argue that x_1 and x_2 are different. Indeed, suppose to the contrary that $x_1 = x_2$. This leads us to the immediate conclusion

$$\mathbf{m}_j^\star = \mathbf{m}_j^i \text{ and } \sigma_j^\star = \sigma_j^i, \tag{2}$$

for all $j \in [\xi]$. The second equality of (2) together with the fact that $m^\star = m^i$, being the consequence of the occurrence of the event $\mathsf{Forge\text{-}coll}$, and with the assumed condition (v) for the elements of Θ_ξ, implies that $\sigma_{\xi+1}^\star = \sigma_{\xi+1}^i$. Further, the condition $m^\star = m^i$ can be written as $g^{h^\star} g_1^{u^\star} = g^{h^i} g_1^{u^i}$; using $h^\star = h^i$, following from the occurrence of the event $\mathsf{Forge\text{-}coll}$, we get $g^{h^\star} g_1^{u^\star} = g^{h^\star} g_1^{u^i}$. By making simple computations, we eventually obtain $u^\star = u^i$ in $\mathbb{F}_p$. This, in turn, means that $\sigma_{\xi+2}^\star = \sigma_{\xi+2}^i$. Wrapping up these considerations, we have proven that $\sigma^\star = \sigma^i$, which in conjunction with the first equality of (2), implies that $\mathbf{m}^\star = \mathbf{m}^i$, and leads us to the contradiction with the condition $(\mathbf{m}^\star, \sigma^\star) \notin \{(\mathbf{m}^i, \sigma^i)\}_{i=1}^q$, being a consequence of the assumed event Forge.

Claim 3. $\mathbf{Succ}^{\mathsf{Forge\text{-}dlp}}_{\Pi^*, \mathcal{A}^*}(n)$ is negligible.

In order to prove this claim, we construct a PPT algorithm $\mathcal{A}^{\mathsf{dlp}}$, using $\mathcal{A}^*$ as a subroutine and attacking the discrete logarithm problem in a group $\mathbb{G}$, generated by $\mathcal{G}_{\mathcal{F}}(1^n)$.

The game is launched by the challenger, which runs $\mathsf{Gen}(1^n)$ to obtain sk^* and pk^*, keeping in mind that the former consists of $\mathsf{params} = (\mathbb{G}, p, g, \mathcal{H})$. In addition to this, the challenger generates $g_1 \in \mathbb{G}$ uniformly at random.

Algorithm $\mathcal{A}^{\mathsf{dlp}}$:
The algorithm is given the keys sk^*, pk^* and g_1.

1. The public key is sent to $\mathcal{A}^*$, which carries on a strong adaptive chosen-message attack; here, $\mathcal{A}^{\mathsf{dlp}}$ serves as its oracle. $\mathcal{A}^*$ eventually outputs $(\mathbf{m}^\star, \sigma^\star)$.
2. If $(\mathbf{m}^\star, \sigma^\star)$ is a valid forgery, then do:
   2.1 If there is $i \in [q]$ such that $m^\star = m^i$ and $h^\star \neq h^i$, then output $(h^\star - h^i) \cdot (u^i - u^\star)^{-1}$, where the computations are made in $\mathbb{F}_p$.
   2.2 Otherwise, abort.
3. Otherwise, abort.

Let C denote the event where $\mathcal{A}^{\mathsf{dlp}}$ solves the underlying discrete logarithm problem. We see that if both Forge and $\mathsf{Forge\text{-}dlp}$ occur, then the need from steps 2 and 2.1 are fulfilled, and consequently, $\mathcal{A}^{\mathsf{dlp}}$ outputs an element of $\mathbb{F}_p$, being of the form $(h^\star - h^i) \cdot (u^i - u^\star)^{-1}$. We are intended to justify that it is actually $\log_g g_1$. Let $g^1 = g^x$ and note that, due to $\mathsf{Forge\text{-}dlp}$, we have $m^\star = m^i$ and ,thus, $g^{h^\star} g_1^{u^\star} = g^{h^i} g_1^{u^i}$, this in turn, can be written as $g^{h^\star} (g^x)^{u^\star} = g^{h^i} (g^x)^{u^i}$. The last equation in $\mathbb{G}$ is equivalent to the following equation in $\mathbb{F}_p$

$$x \cdot (u^i - u^\star) = h^\star - h^i. \tag{3}$$

Observe that if $u^i = u^\star$, then $g^{h^\star} g_1^{u^\star} = g^{h^i} g_1^{u^\star}$, and consequently $g^{h^\star - h^i} = 1$. This leads us to the condition $h^\star = h^i$, which contradicts the assumption that $\mathsf{Forge\text{-}dlp}$ occurs. Therefore, $u^i \neq u^\star$ and the equation (3) has a solution $x = (h^\star - h^i) \cdot (u^i - u^\star)^{-1}$, obviously $x = \log_g g_1$. Putting things together, we conclude that if Forge and $\mathsf{Forge\text{-}dlp}$ occur then C occurs as well, and hence, we get

$$\mathbf{Succ}^{\mathsf{Forge\text{-}dlp}}_{\Pi^*, \mathcal{A}^*}(n) \leqslant \Pr[\mathsf{C}] \leqslant \mathsf{negl}(n),$$

according to the assumption.

Now we use (1) in conjunction with Claims 1-3 to conclude that $\mathbf{Adv}^{\mathsf{euf\text{-}cma}}_{\Pi^*}(\mathcal{A}^*, n)$ is negligible. This finishes the proof. □

IV. Multifactorial and euf-cma secure digital signature scheme

In this section we present a construction of a signature scheme which is based on pairings. We prove that the scheme is euf-cma secure, whilst simultaneously not being suf-cma secure.

As usual, let $\mathcal{M} = \{M_n\}$ denote the message space associated with the scheme, and let us assume that $M_n = \{0,1\}^d$, where $d = d(n)$. Therefore, for the fixed value n of the security parameter, the messages which are being signed are

of the length d. Such messages are split into w-bits blocks, where w depends on a fixed integer parameter $\xi \geqslant 1$ and in a sense, each block is signed separately to finally mix them together, and the result combine with the appropriate secure key. A number of bits in a single block is computed as $w = \lceil d/\xi \rceil$, if $d \not\equiv 0 \pmod{\xi}$, then a message is padded with $(\lceil d/\xi \rceil \xi - d)$ zeros at the least significant bits positions, before splitting it up. Not wanting to multiply notations, a padded message is dented by the same symbol as the original one. To be more precise, if $m \in M_n$, then the symbol m describes a padded message and is addition $m = m_1 \| m_2 \| \cdots \| m_\xi$, where $m_i = ((m_i)_1, (m_i)_2, \ldots (m_i)_w)_2$ and $(m_i)_j \in \{0, 1\}$.

A. Construction of the signature scheme

Let us fix an integer $\xi \geqslant 1$ and let n be a security parameter. We stress that ξ is understood as indicating the scheme, and does not depend on n. Assume further that $\mathscr{G}$ is an efficient algorithm that on inputs 1^n and ξ outputs $(w, \mathbb{G}, \mathbb{G}_T, p, g, \hat{e})$, where:

- $\mathbb{G}$, $\mathbb{G}_T$ are two cyclic groups of prime order $p \in \mathcal{P}(n)$, where group operations in $\mathbb{G}$, $\mathbb{G}_T$ can be performed efficiently.
- $g \in \mathbb{G}$ is chosen uniformly at random from the set of all generators of $\mathbb{G}$.
- $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently computable pairing.
- $w = \lceil \frac{d}{\xi} \rceil$.

Key generation: Algorithm $\mathsf{Gen}(1^n)$ is defined as follows:

1. Compute $\mathsf{params} := (w, \mathbb{G}, \mathbb{G}_T, p, g, \hat{e}) \leftarrow \mathscr{G}(1^n, \xi)$.
2. Choose g_2 uniformly at random from the set of all generators of $\mathbb{G}$.
3. Choose $a \xleftarrow{\$} \mathbb{F}_p^*$ and compute $g_1 = g^a$.
4. Pick out $(u_{0,i})_{i=1}^{\xi}, u_1, \ldots, u_w \xleftarrow{\$} \mathbb{G}$.
5. The secret key is $sk = g_2^a$ and the public key is $pk = (\mathsf{params}, g_1, g_2, (u_{0,i})_{i=1}^{\xi}, u_1, \ldots, u_w)$.

Signature generation: For $m \in M_n$, the algorithm $\mathsf{Sign}_{sk}(m)$ is defined as follows:

1. The message m is padded to $m = m_1 \| m_2 \| \cdots \| m_\xi$, where $m_i = ((m_i)_1, (m_i)_2, \ldots (m_i)_w)_2$, as we have described above.
2. For each $i \in [\xi]$, choose $r_i \xleftarrow{\$} \mathbb{F}_p$.
3. Output the signature $\sigma = (\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1})$, where

$$\sigma_i = g^{r_i} \text{ for each } i \in [\xi],$$

and

$$\sigma_{\xi+1} = sk \prod_{i=1}^{\xi} U_i(m_i)^{r_i} = g_2^a \prod_{i=1}^{\xi} \left(u_{0,i} \prod_{k=1}^{w} u_k^{(m_i)_k} \right)^{r_i}.$$

We set $U_i(m_i) = u_{0,i} \prod_{k=1}^{w} u_k^{(m_i)_k}$, for $i \in [\xi]$.

Signature verification: Algorithm $\mathsf{Vrfy}_{pk}(m, \sigma)$ is defined as follows:

1. Parse pk as $(\mathsf{params}, g_1, g_2, (u_{0,i})_{i=1}^{\xi}, u_1, \ldots, u_w)$ and σ as $(\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1})$.
2. Pad and split m into ξ blocks of w-bits each, as described above, i.e. $m = m_1 \| m_2 \| \cdots \| m_\xi$, where $m_i = ((m_i)_1, (m_i)_2, \ldots (m_i)_w)_2$.
3. Output 1 if and only if the following equality holds

$$\hat{e}(g, \sigma_{\xi+1}) \stackrel{?}{=} \hat{e}(g_1, g_2) \prod_{i=1}^{\xi} \hat{e}(\sigma_i, U_i(m_i)). \tag{4}$$

Otherwise, output 0.

Firstly, we verify correctness. To do this, let us take a honestly generated signature σ on a message m, we get

$$\begin{aligned}
\hat{e}(g, \sigma_{\xi+1}) &= \hat{e}(g, g_2^a \prod_{i=1}^{\xi} U_i(m_i)^{r_i}) \\
&= \hat{e}(g, g_2^a) \prod_{i=1}^{\xi} \hat{e}(g, U_i(m_i)^{r_i}) \\
&= \hat{e}(g^a, g_2) \prod_{i=1}^{\xi} \hat{e}(g^{r_i}, U_i(m_i)) \\
&= \hat{e}(g_1, g_2) \prod_{i=1}^{\xi} \hat{e}(\sigma_i, U_i(m_i)).
\end{aligned}$$

B. Security of the signature scheme

Before we start justifying that the presented signature scheme is euf-cma, we need to formulate the same formal definitions. Although the first one is analogous to Definition 7, there is a need to adapt it to the new theoretical conditions.

Definition 10: The *DLP is hard relative to* $\mathscr{G}$, if for all PPT adversaries $\mathcal{A}$ and for all polynomials $p = p(n)$ $(\in \mathcal{P}(n))$, the probability

$$\Pr\left[\begin{array}{c} (w, \mathbb{G}, \mathbb{G}_T, p, g, \hat{e}) \leftarrow \mathscr{G}(1^n, \xi); g_2 \xleftarrow{\$} \mathbb{G}; \\ x \leftarrow \mathcal{A}(1^n, w, \xi, \mathbb{G}, \mathbb{G}_T, p, g, \hat{e}) \end{array} \mid g^x = g_2 \right].$$

is negligible.

Note that this definition implies hardness of the discrete logarithm problem in $\mathbb{G}_T$. If not, then it is easy to use $\hat{e}$ and find desired exponent, which contradicts the definition.

Definition 11: The *computational Diffie-Hellman* (CDH) *problem is hard relative to* $\mathscr{G}$ if DLP is hard relative to $\mathscr{G}$ and if for all PPT adversaries $\mathcal{A}$ and for all polynomials $p = p(n)$ $(\in \mathcal{P}(n))$, the following is negligible

$$\Pr\left[\begin{array}{c} \mathsf{params} \leftarrow \mathscr{G}(1^n, \xi); \\ a, b \xleftarrow{\$} \mathbb{F}_p \end{array} \mid \mathcal{A}(1^n, \mathsf{params}, g^a, g^b) = g^{ab} \right].$$

Theorem 12: If the CDH problem is hard relative to $\mathscr{G}$, then the signature scheme is euf-cma secure.

Proof. Let $\mathcal{A}$ be an adversary which attacks the signature scheme. Without loss of generality, we can assume that $\mathcal{A}$ makes q queries to the signing oracle and it succeeds with probability ϵ. Having this, we construct an adversary $\mathcal{A}^{\mathsf{cdh}}$ which solves the CDH problem with success probability $\mathcal{O}(\epsilon)$.

Algorithm $\mathcal{A}^{\mathsf{cdh}}$:
The algorithm is given $\mathsf{params} = (w, \mathbb{G}, \mathbb{G}_T, p, g, \hat{e})$ and g_1, g_2.

1. Set $\ell = 2q$ and assume $w\ell < p$.
2. For each $i \in [\xi]$, pick up $x_{0,i} \xleftarrow{\$} \{-w\ell, \ldots, 0\}$ and $y_{0,i} \xleftarrow{\$} \mathbb{F}_p$. Also choose $x_1, \ldots, x_w \xleftarrow{\$} \{0, \ldots, \ell\}$ and $y_1, \ldots, y_w \xleftarrow{\$} \mathbb{F}_p$.
3. For $i \in [\xi]$, set $u_{0,i} := g_2^{x_{0,i}} g^{y_{0,i}}$, similarly, for $i \in [w]$, set $u_i := g_2^{x_i} g^{y_i}$.
4. For $m = m_1 \| m_2 \| \cdots \| m_\xi$, define the functions

$$X_i(m_i) = x_{0,i} + \sum_{k=1}^{w} (m_i)_k x_k$$
$$Y_i(m_i) = y_{0,i} + \sum_{k=1}^{w} (m_i)_k y_k$$

4. Send the public key $(\mathsf{params}, g_1, g_2, (u_{0,i})_{i=1}^{\xi}, u_1, \ldots, u_w)$ to $\mathcal{A}$. When $\mathcal{A}$ requests a signature on the message $m = m_1 \| m_2 \| \cdots \| m_\xi$, do:
   4.1. If $X_i := X_i(m_i) = 0$ in $\mathbb{F}_p$, for some $i \in [\xi]$ then abort.
   4.2. Otherwise, set $Y_i := Y_i(m_i)$. For each $i \in [\xi]$, choose $r_i \xleftarrow{\$} \mathbb{F}_p$ and return to $\mathcal{A}^{\mathsf{cdh}}$, the desired signature σ having the form

$$\left(\left(g^{r_i} g_1^{-\xi^{-1} X_i^{-1}} \right)_{i=1}^{\xi}, \prod_{i=1}^{\xi} g_2^{X_i r_i} g^{Y_i r_i} g_1^{-\xi^{-1} X_i^{-1} Y_i} \right).$$

5. Eventually, $\mathcal{A}^{\mathsf{cdh}}$ outputs a pair $(m^\star, \sigma)$, $\sigma = (\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1})$. If it is a valid forgery, do
   5.1. If $X_i(m_i^\star) \neq 0$ in $\mathbb{F}_p$, for some $i \in [\xi]$, then abort.
   5.2. Otherwise, output $\sigma_{\xi+1} / \prod_{i=1}^{\xi} \sigma_i^{Y_i(m_i^\star)}$

Below, we argue the following three statements:

1) Due to Lemma 6, the public key sent to $\mathcal{A}$ has the proper distribution.
2) As long as $\mathcal{A}^{\mathsf{cdh}}$ does not abort, the signatures given to $\mathcal{A}$ are correctly distributed. Indeed, let $a, b \in \mathbb{F}_p$ be such that $g^a = g_1$ and $g^b = g_2$. As we saw in Section IV-A, the real signature is computed by signing algorithm as $\left(g^{\tau_1}, \ldots, g^{\tau_\xi}, g_2^a \prod_{i=1}^{\xi} U_i(m_i)^{\tau_i} \right)$ for τ_i selected uniformly at random from $\mathbb{F}_p$. Setting $\tau_i = r_i - \xi^{-1} a X^{-1}$ (keeping in mind that $\mathcal{A}^{\mathsf{cdh}}$ signs a message iff $X \neq 0$ in $\mathbb{F}_p$), we see by lemma 6 that r_i and τ_i are equally likely. Furthermore, observe that $U_i(m_i) = g_2^{X_i(m_i)} g^{Y_i(m_i)}$, we obtain

$$\begin{aligned} g_2^a \prod_{i=1}^{\xi} U_i(m_i)^{\tau_i} &= g_2^a \prod_{i=1}^{\xi} U_i(m_i)^{r_i - \xi^{-1} a X_i^{-1}} \\ &= g_2^a \prod_{i=1}^{\xi} (g_2^{X_i} g^{Y_i})^{r_i - \xi^{-1} a X_i^{-1}} \\ &= (g_2^{\xi^{-1}})^{\xi a} \prod_{i=1}^{\xi} g_2^{X_i r_i} g_2^{-\xi^{-1} a} g^{Y_i r_i} g^{-\xi^{-1} a X_i^{-1} Y_i} \\ &= \prod_{i=1}^{\xi} (g_2^{\xi^{-1}})^a g_2^{X_i r_i} ((g_2^{\xi^{-1}})^a)^{-1} g^{Y_i r_i} g^{-\xi^{-1} a X_i^{-1} Y_i} \\ &= \prod_{i=1}^{\xi} g_2^{X_i r_i} g^{Y_i r_i} g_1^{-\xi^{-1} X_i^{-1} Y_i}, \end{aligned}$$

and

$$g^{\tau_i} = g^{r_i - \xi^{-1} a X_i^{-1}} = g^{r_i} g^{-\xi^{-1} a X_i^{-1}} = g^{r_i} g_1^{-\xi^{-1} X_i^{-1}}.$$

Thus, putting things together, signatures generated by $\mathcal{A}^{\mathsf{cdh}}$ have the correct distribution.
3) As long as both $\mathcal{A}^{\mathsf{cdh}}$ and $\mathcal{A}$ do not abort, the output of $\mathcal{A}^{\mathsf{cdh}}$ is a correct solution to the given CDH instance. Indeed, let $\sigma = (\sigma_1, \ldots, \sigma_\xi, \sigma_{\xi+1})$ be a valid signature on $m^\star$, which has been outputted by $\mathcal{A}$, satisfying $X_i(m_i^\star) = 0$ in $\mathbb{F}_p$ for all $i \in [\xi]$. By (4), we have

$$\frac{\hat{e}(g, \sigma_{\xi+1})}{\prod_{i=1}^{\xi} \hat{e}(\sigma_i, U_i(m_i))} = \hat{e}(g_1, g_2) = \hat{e}(g, g^{ab}),$$

Hence,

$$\begin{aligned} \hat{e}(g, g^{ab}) &= \frac{\hat{e}(g, \sigma_{\xi+1})}{\prod_{i=1}^{\xi} \hat{e}\left(\sigma_i, u_{0,i} \cdot u_1^{(m_i^\star)_1} \cdots u_w^{(m_i^\star)_w}\right)} \\ &= \frac{\hat{e}(g, \sigma_{\xi+1})}{\prod_{i=1}^{\xi} \hat{e}\left(\sigma_i, g_2^{x_{0,i}} g^{y_{0,i}} \cdot (g_2^{x_1} g^{y_1})^{(m_i^\star)_1} \cdots (g_2^{x_w} g^{y_w})^{(m_i^\star)_w}\right)} \\ &= \frac{\hat{e}(g, \sigma_{\xi+1})}{\prod_{i=1}^{\xi} \hat{e}\left(\sigma_i, g_2^{X_i(m_i^\star)} g^{Y_i(m_i^\star)}\right)} = \frac{\hat{e}(g, \sigma_{\xi+1})}{\prod_{i=1}^{\xi} \hat{e}\left(\sigma_i, g_2^0 g^{Y_i(m_i^\star)}\right)} \\ &= \frac{\hat{e}(g, \sigma_{\xi+1})}{\prod_{i=1}^{\xi} \hat{e}\left(g, \sigma_i^{Y_i(m_i^\star)}\right)} = \frac{\hat{e}(g, \sigma_{\xi+1})}{\hat{e}\left(g, \prod_{i=1}^{\xi} \sigma_i^{Y_i(m_i^\star)}\right)}. \end{aligned}$$

This, together with basic properties of pairing, implies that $\sigma_{\xi+1} / \prod_{i=1}^{\xi} \sigma_i^{Y_i(m_i^\star)} = g^{ab}$, what solves CDH.

Studying carefully the algorithm $\mathcal{A}^{\mathsf{cdh}}$, we immediately find out that it solves the CDH problem, if and only if it does not abort, or in other words, if the conditions 4.2 and 5.2 are satisfied. Of course, we might suspect that they never are or that the probability of such an event is negligible. Obviously, in each of these cases our above considerations would be worthless. This means that to complete the proof we have to analyze how likely it is that $\mathcal{A}^{\mathsf{cdh}}$ does not abort and successfully completes the simulation.

As justified earlier, $\mathcal{A}^{\mathsf{cdh}}$ perfectly simulates the real signature oracle to $\mathcal{A}$, which, in particular means that $\{x_{0,i}, x_k\}$ chosen by $\mathcal{A}^{\mathsf{cdh}}$ are independent of pk given to $\mathcal{A}$. This means

that if $\mathcal{A}^{\mathsf{cdh}}$ adaptively requests making signatures on messages $m^1, \dots m^q$, then it outputs a forgery on $m^\star \notin \{m^1, \dots, m^q\}$ with probability ϵ.

We attempt to estimate the desire probability in more general matter. Namely, let us fix arbitrary messages $m^1, \dots, m^q$ and $m^\star \notin \{m^1, \dots m^q\}$, and let $\{x_{0,i}, x_k\}$ be chosen from the same distribution used by $\mathcal{A}^{\mathsf{cdh}}$. Of course, we can assume without loss of the generality that m_i^j and $m_i^\star$ has been padded. Therefore, we already know that $m^j = m_1^j \| m_2^j \| \cdots \| m_\xi^j$, where $j \in [q] \cup \{\star\}$. Let E_i^j and $E_i^\star$ denote the events such that $X_i(m_i^j) \neq 0$ and $X_i(m_i^\star) = 0$ respectively. Putting things together, we are intended to find the lower bound for the probability of the following event:

$$E := \bigcap_{j=1}^{q} \bigcap_{i=1}^{\xi} E_i^j \cap \bigcap_{i=1}^{\xi} E_i^\star = \bigcap_{i=1}^{\xi} \left(E_i^\star \cap \bigcap_{j=1}^{q} E_i^j \right). \tag{5}$$

The probability space Ω is a subfamily of the power set $P(\{0, \dots, \ell\})$, consisting of all these sets whose elements sum to an element of $\{0, \dots w\ell\}$. Due to the fact that we do not care for the order of the elements, there is a bijection between Ω and $\{0, \dots w\ell\}$. Keeping this in minds we note that for each block m_i of arbitrary (padded) message m and $x_1, \dots x_w$ (we take into account only coefficients standing by bits 1), there is exactly one choice of $x_{0,i}$ such that $X_i(m_i) = 0$. As $x_{0,i}$ are chosen independently, the events $F_i := E_i^\star \cap \bigcap_{j=1}^{q} E_i^j$ are independent as well. Therefore, by (5),

$$\Pr[E] = \prod_{i=1}^{\xi} \Pr[F_i]. \tag{6}$$

For the complement of F_i, we have the inclusion

$$\overline{F_i} \subset \overline{E_i^\star} \cup \bigcup_{j=1}^{q} \left(E_i^\star \cap \overline{E_i^j} \right).$$

Hence,

$$\begin{aligned} \Pr[\overline{F_i}] &\leqslant \Pr[\overline{E_i^\star}] + \sum_{j=1}^{q} \Pr[\overline{E_i^j} \cap E_i^\star] \\ &\leqslant \Pr[\overline{E_i^\star}] + \sum_{j=1}^{q} \Pr[\overline{E_i^j} | E_i^\star] \Pr[E_i^\star] \end{aligned} \tag{7}$$

It remains to estimate the above probability. To do this, we consider two cases:

(i) $\Pr[\overline{E_i^\star}]$. We have argued that for $x_1, \dots x_w$, there is exactly one choice of $x_{0,i}$, which satisfies $X_i(m_i^\star) = 0$. This observation leads us to the immediate conclusion

$$\Pr[\overline{E_i^\star}] = \Pr[X_i(m_i^\star) \neq 0] = \frac{w\ell}{w\ell + 1} \tag{8}$$

(ii) $\Pr[\overline{E_i^j} | E_i^\star] \Pr[E_i^\star]$. As by assumption $m^j \neq m^\star$, they differ in at least one bit, which belongs to the ith block with probability $1/\xi$. In this case, there is $k \in [w]$ such that $(m_i^j)_l = (m_i^\star)_l$, and we can assume without loss of generality that $(m_i^j)_l = 0$ and $(m_i^\star)_l = 1$. This yields

$$x_{0,i} = - \sum_{\substack{k=1 \\ k \neq l}}^{w} x_k \cdot (m_i^j)_k,$$

$$x_{0,i} + x_l = - \sum_{\substack{k=1 \\ k \neq l}}^{w} x_k \cdot (m_i^\star)_k.$$

As equality $X_i(m_i^\star) = 0$ indicates a unique element of Ω, the same element corresponds to the condition $X_i(m_i^j) = 0$, and this implies $x_l = 0$. Therefore, we obtain $\Pr[\overline{E_i^j} | E_i^\star] \leqslant 1/\xi(\ell + 1)$. Arguing as in (i), we have $\Pr[E_i^\star] = 1/w\ell + 1$. Finally,

$$\Pr[\overline{E_i^j} | E_i^\star] \Pr[E_i^\star] \leqslant \frac{1}{\xi(\ell + 1)(w\ell + 1)}$$

By (i), (ii) and (7),

$$\Pr[\overline{F_i}] \leqslant \frac{w\ell}{w\ell + 1} + \frac{q}{\xi(\ell + 1)(w\ell + 1)}$$

Having computed complementary probability of $\overline{F_i}$, we immediately get

$$\begin{aligned} \Pr[F_i] &\geqslant \frac{2q\xi + \xi - q}{(2wq + 1)(2q\xi + \xi)} \geqslant \frac{q\xi + \xi}{(2wq + 1)(2q\xi + \xi)} \\ &\geqslant \frac{q\xi + \xi}{(2wq + 1)(2q\xi + \xi)} \geqslant \frac{1}{4wq + 2} \end{aligned}$$

Combining this with (6), we have

$$\Pr[E] \geqslant \frac{1}{(4wq + 2)^\xi}$$

To sum up, we have shown that $\mathcal{A}^{cdh}$ succeeds with probability at least $\epsilon \cdot (4wq + 2)^{-\xi}$. On the other hand, CDH is hard relative to $\mathscr{G}$ and ξ does not depend on n, hence $\epsilon = \epsilon(n)$ must be negligible. This finishes the proof. □

ACKNOWLEDGMENT

The author would like to thank Krzysztof Mańk for giving helpful suggestions.

REFERENCES

[1] J. An, Y. Dodis, and T. Rabin, "On the security of joint signature and encryption", *Proceedings of Eurocrypt 2002*, vol. 2332 of LNCS, pages 83-107. Springer-Verlag, 2002.

[2] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers", *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[3] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks", *SIAM Journal on Computing*, 17(2):281–308, 1988.

[4] B. Waters, "Efficient identity-based encryption without random oracles", *Proceedings of Eurocrypt 2005*, vol. 3494 of LNCS, pages 114–127. Springer-Verlag, 2005.

[5] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing", *SIAM J. Comput.*, 32(3):586–615, 2003.