Efficient Leakage-Resilient Signature Schemes in the Generic Bilinear Group Model

Fei Tang, Hongda Li, Qihua Niu, and Bei Liang

State Key Laboratory of Information Security
Institute of Information Engineering, Chinese Academy of Sciences (CAS)
No. 89 Minzhuang Road, Haidian District, Beijing 100093, China
Email: *tangfei127@163.com, hdli@ucas.ac.cn, {niuqihua, liangbei}@iie.ac.cn*

Abstract. We extend the techniques of Kiltz et al. (in ASIACRYPT 2010) and Galindo et al. (in SAC 2012) to construct two efficient leakage-resilient signature schemes. Our schemes based on Boneh-Lynn-Shacham (BLS) short signature and Waters signature schemes, respectively. Both of them are more efficient than Galindo et al.'s scheme, and can tolerate leakage of $(1 - o(1))/2$ of the secret key at every signature invocation. The security of the proposed schemes are proved in the generic bilinear group model (additionally, in our first scheme which based on the BLS short signature, a random oracle is needed for the proof).

Keywords: digital signature, leakage-resilient cryptography, generic bilinear group model.

1 Introduction

In the traditional security proof of the cryptographic schemes, there has a basic assumption that the secret state is completely hidden to the adversary. However, it is very hard to realize this assumption in the real world. Many cryptographic engineers have designed some side-channel attacks which can detect some leakage information about the secret state. For example, power consumption [25], fault attacks [5, 9], and timing attacks [8], etc.

Leakage-resilient cryptography is a countermeasure to against the side-channel attacks with some algorithmic techniques. Which means that designing algorithms such that their description already provides security against those attacks. Leakage-resilient cryptography is an increasingly active area in recent years and many leakage models have been proposed, such as only computation leaks information (OCLI) [19, 21, 27, 24], memory leakage [1, 17], bounded retrieval [2, 3, 14], and auxiliary input models [15, 21, 20, 34], etc. In this work, we design leakage-resilient signature schemes based on the following two leakage models:

- **OCLI model:** leakage is assumed to only occur on values that currently accessed during the computation.
- **Continual leakage model:** the amount of leakage is assumed to be bounded only in between any two successive key refreshes but the overall amount can be unbounded.

Bounded leakage model [23, 7] is a weaker notion, corresponding to the continual leakage model, which means that the amount of the leakage information is bounded with a fixed value throughout the lifetime of the system. Obviously, the continual leakage model is more closer to the real scenarios.

Note that in the continual leakage setting, the secret state should be stateful, i.e., the secret state should be updated after (or before) every round of the invocation of the secret state. Otherwise, the entire secret state will be completely leaked after plenty of invocations.

In the past few years, many leakage-resilient cryptographic schemes or protocols have been proposed based on various leakage models. For example, stream ciphers [30], zero knowledge proof [22], PKE [24, 29], IBE [14, 34], signatures [2, 7, 19, 23, 28], etc. All of these schemes could be proved secure in the standard model without any idealized assumption (e.g., random oracle, generic groups) and thus have a more persuasive security result, but on the other hand, they are not yet quite efficient to be used in practice. Such as the signature schemes [23, 18, 6, 7, 28, 20, 35], all of them utilized a non-interactive witness indistinguishable proof or zero-knowledge proof system, and even a complicated PKE scheme. Such schemes mainly exist in the field of theoretical research for the reason of the poor efficiency.

Kiltz and Pietrzak [24] constructed a leakage-resilient PKE scheme which is a bilinear version of the ElGamal key encapsulation mechanism and it is secure even in the presence of continual leakage in the *generic bilinear group* (GBG) model [11]. It is more important that their scheme is very efficient, just less than a little time slower than the standard ElGamal scheme. Galindo and Vivek [21] then adapted their techniques (i.e., blinding the secret state) to construct a practical signature scheme based on the Boneh-Boyen IBE scheme [10]. Its efficiency is close to the non leakage-resilient one and it tolerates leakage of almost half of the bits of the secret key at every signature invocation.

In this paper, we follow the techniques by Kiltz et al. [24] and Galindo et al. [21], construct two leakage-resilient signature schemes based on the OCLI and continual leakage models. They are provable leakage-resilience in the GBG model (the BLS-based one needs an additional random oracle for the proof). Our first scheme is based on the BLS short signature scheme [13] which signing algorithm is deterministic, we adapt it to a probabilistic one and the resulting scheme tolerates leakage of $(1 - o(1))/2$ of the secret key at every signature invocation. Our second scheme is based on the Waters signature scheme [33], the resulting scheme also tolerates leakage of $(1-o(1))/2$ of the secret key at every signature invocation. Both of them are more efficient than Galindo and Vivek's signature scheme, more precisely, one exponentiation is decreased in the signing and verification algorithm, respectively.

2 Preliminaries

In this section, we present some basic notions and preliminaries for this paper: bilinear groups and two intractability assumptions CDH and DBDH, generic bilinear groups model, entropy, and Schwartz-Zippel lemma.

The following notations will be used in this paper. Let $\mathbb{Z}$ be the set of integers and $\mathbb{Z}_p$ be the ring modulo p. 1^k denotes the string of k ones for $k \in \mathbb{N}$. $|x|$ denotes the length of the bit string x. $s \xleftarrow{\$} S$ means randomly choosing an element s form the set S. $[n]$ is a shorthand for the set $\{1, 2, \ldots, n\}$. We write $y \leftarrow \mathsf{A}(x)$ to indicate that running the algorithm A with input x and then outputs y, $y \xleftarrow{\$} \mathsf{A}(x)$ has the same indication except that A is a probabilistic algorithm, and if we

want to explicitly denote the randomness r used during the computation we write it $y \xleftarrow{r} \mathsf{A}(x)$. Lastly we write PPT for the probabilistic polynomial time.

2.1 Bilinear Groups

Let G_1 and G_2 be two multiplicative cyclic groups with a same prime order p, and g be an arbitrary generator of G_1. We say that $\hat{e} : G_1 \times G_1 \to G_2$ be an admissible bilinear mapping if it satisfies the the following properties:

- *Bilinearity:* $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_p$.
- *Non-degeneracy:* $\hat{e}(g, g) \neq 1_{G_2}$.
- *Computability:* there exists efficient algorithm to calculate $\hat{e}(g^a, g^b)$ for all $a, b \in \mathbb{Z}_p$.

We assume that $\mathsf{BGen}(1^k)$ be a PPT algorithm which generates parameters $(G_1, G_2, p, g, \hat{e})$ to satisfy the above properties with a input of security parameter k. The group G_1 is said to be a bilinear group, and it is also called the base group and G_2 be the target group.

Definition 1 (CDH Assumption). *For any PPT adversary $\mathcal{A}$, any polynomial $p(\cdot)$, and all sufficiently large $k \in \mathbb{N}$,*

$$\mathbf{Pr}\left[\begin{array}{ll}(G_1, G_2, p, g, \hat{e}) \leftarrow \mathsf{BGen}(1^k); & \\ a, b \xleftarrow{\$} \mathbb{Z}_p; & : \mathsf{v} = g^{ab} \\ \mathsf{v} \leftarrow \mathcal{A}(G_1, p, g, g^a, g^b) & \end{array}\right] < \frac{1}{p(k)}.$$

Definition 2 (DBDH Assumption). *For any PPT adversary $\mathcal{A}$, any polynomial $p(\cdot)$, and all sufficiently large $k \in \mathbb{N}$,*

$$\left|\mathbf{Pr}\left[\begin{array}{ll}(G_1, G_2, p, g, \hat{e}) \leftarrow \mathsf{BGen}(1^k); & \\ a, b, c \xleftarrow{\$} \mathbb{Z}_p; & : \mathsf{d} = 1 \\ \mathsf{d} \leftarrow \mathcal{A}(\mathbb{P}, g^a, g^b, g^c, \hat{e}(g, g)^{abc}) & \end{array}\right] - \right.$$

$$\left.\mathbf{Pr}\left[\begin{array}{ll}(G_1, G_2, p, g, \hat{e}) \leftarrow \mathsf{BGen}(1^k); & \\ a, b, c, r \xleftarrow{\$} \mathbb{Z}_p; & : \mathsf{d} = 1 \\ \mathsf{d} \leftarrow \mathcal{A}(\mathbb{P}, g^a, g^b, g^c, \hat{e}(g, g)^r) & \end{array}\right]\right| < \frac{1}{p(k)}.$$

2.2 Generic Bilinear Group Model

In the generic group model [31], the elements of the group encoded by unique but randomly chosen strings, and thus the only property that can be tested by adversary is equality. Boneh et al. [11] extended it to a generic bilinear group (GBG) model. In the GBG model, the encoding is given by randomly chosen injective functions $\xi_1 : \mathbb{Z}_p \to \Xi_1$ and $\xi_2 : \mathbb{Z}_p \to \Xi_2$ which are the representations of the elements of the base group G_1 and target group G_2, respectively (w.l.o.g., we assume that $\Xi_1 \cap \Xi_2 = \emptyset$). The operations of the groups and the bilinear map are performed by three public oracles $\mathcal{O}_1, \mathcal{O}_2$, and $\mathcal{O}_{\hat{e}}$, respectively. For any $a, b \in \mathbb{Z}_p$

- $\mathcal{O}_1(\xi_1(a), \xi_1(b)) \to \xi_1(a + b \mod p)$

- $\mathcal{O}_2(\xi_2(a), \xi_2(b)) \to \xi_2(a+b \mod p)$
- $\mathcal{O}_{\hat{e}}(\xi_1(a), \xi_1(b)) \to \xi_2(ab \mod p)$

For a fixed generator g of G_1 satisfies $g = \xi_1(1)$ and $g_T = \hat{e}(g, g) = \xi_2(1)$.

2.3 Entropy

Let X be a finite random variable. The *min-entropy* of X defined as:

$$\mathbf{H}_\infty(X) \stackrel{def}{=} -\log_2(\max_x \mathbf{Pr}[X = x]).$$

The *average conditional min-entropy* of X given a random variable Y defined as:

$$\widetilde{\mathbf{H}}_\infty(X|Y) \stackrel{def}{=} -\log_2(\mathbb{E}_{y \leftarrow Y}[\max_x \mathbf{Pr}[X = x|Y = y]).$$

The following lemma is based on the work of [16].

Lemma 1. *Let $f : X \to \{0,1\}^\Delta$ be a function on X. Then $\widetilde{\mathbf{H}}_\infty(X|f(X)) \geq \mathbf{H}_\infty(X) - \Delta$.*

2.4 Schwartz-Zippel Lemma

We follow the result of [21, 31], it is a simple variant of the Schwartz-Zippel lemma [32, 36].

Lemma 2. *Let $F \in \mathbb{Z}_p[X_1, \ldots, X_n]$ be a non-zero polynomial of total degree at most d. Let $P_i (i = 1, \ldots, n)$ be probability distributions on $\mathbb{Z}_p$ such that $\mathbf{H}_\infty(P_i) \geq \log p - \Delta$, where $0 \leq \Delta \leq \log p$. If $x_i \stackrel{P_i}{\leftarrow} \mathbb{Z}_p (i = 1, \ldots, n)$ are chosen independently, then $\mathbf{Pr}[F(x_1, \ldots, x_n) = 0] \leq \frac{d}{p} 2^\Delta$.*

This lemma can be proved by mathematical induction (cf. paper [21, 31] for detailed description). Based on this lemma, we can get the following result directly.

Corollary 1. *If $\Delta = (1 - o(1)) \log p$ in Lemma 2, then $\mathbf{Pr}[F(x_1, \ldots, x_n) = 0]$ is negligible (in $\log p$).*

3 Definitions

3.1 Signature Scheme

A signature scheme Σ generally consists of three algorithms, key generation, signing, and verification, denoted by KGen, Sign, and Vrfy, respectively.

Definition 3. *Σ=(KGen, Sign, Vrfy) is a signature scheme if it satisfies:*

- KGen *is a PPT algorithm takes as input a security parameter k, then outputs the signer's public key pk and secret key sk. We write it $(pk, sk) \stackrel{\$}{\leftarrow} \mathsf{KGen}(1^k)$.*
- Sign *is a PPT algorithm run by the signer who takes as input its secret key sk and a message m_i, then outputs a signature σ_i. We write it $\sigma_i \stackrel{\$}{\leftarrow} \mathsf{Sign}(sk, m_i)$.*
- Vrfy *is a deterministic algorithm run by the verifier who takes as input the signer's public key pk, the signed message m_i, and the corresponding signature σ_i, then outputs 1 if it is valid, else outputs 0. We write it $1/0 \leftarrow \mathsf{Vrfy}(pk, m_i, \sigma_i)$.*

  For any index i, we require that $1 \leftarrow \mathsf{Vrfy}(pk, m_i, \mathsf{Sign}(sk, m_i))$.

We say that a signature scheme is stateful if its signing algorithm is stateful, which means that the secret key will be updated before (or after) each signing algorithm invocation while the public key remains fixed.

3.2 Security

The notion of existential unforgeability against adaptive chosen message attack (EUF-CMA) for the signature scheme is defined by the following game $\mathcal{G}^{euf-cma}_{\Sigma,\mathcal{A}}$.

Game $\mathcal{G}^{euf-cma}_{\Sigma,\mathcal{A}}(1^k)$	**Oracle** $\mathcal{O}_{\mathsf{Sign}}(m_i)$
$(pk, sk) \xleftarrow{\$} \mathsf{KGen}(1^k)$	$\sigma_i \xleftarrow{\$} \mathsf{Sign}(sk, m_i)$
$(m^*, \sigma^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}}(pk)$	return σ_i and set $i \leftarrow i + 1$
if $1 \leftarrow \mathsf{Vrfy}(pk, m^*, \sigma^*)$ and $m^* \notin \{m_1, \ldots, m_i\}$	
then output 1 else output 0	

Adversary $\mathcal{A}$ wants to give a forgery (m^*, σ^*) by means of adaptive query to the signing oracle $\mathcal{O}_{\mathsf{Sign}}$. We denote the advantage of $\mathcal{A}$ wins the above game by $\mathbf{Adv}^{euf-cma}_{\Sigma,\mathcal{A}}$.

Definition 4. *The signature scheme Σ is* EUF-CMA *secure if have no polynomial-bounded adversary can win the above game with a non-negligible advantage.*

3.3 Security in the Presence of Leakage

Following the techniques of the papers [21, 24], we split the singing key into two parts which stored in different parts of the memory. Then the signing process be divided into corresponding two phases. However, the input/output behavior will exactly the same as in the original one.

Formally, $\Sigma^* = (\mathsf{KGen}^*, \mathsf{Sign}^*, \mathsf{Vrfy}^*)$ be a stateful signature scheme, in the KGen^* algorithm, the secret key sk is split into two initial states S_0 and S'_0, correspondingly, the signing algorithm be processed with a sequence of two phases $\mathsf{Sign}^* = (\mathsf{Sign}^*_{Phase1}, \mathsf{Sign}^*_{Phase2})$. The i^{th} invocation of signing (with secret state (S_{i-1}, S'_{i-1})) is computed as

$$(S_i, w_i) \xleftarrow{r_i} \mathsf{Sign}^*_{Phase1}(S_{i-1}, m_i); (S'_i, \sigma_i) \xleftarrow{r'_i} \mathsf{Sign}^*_{Phase2}(S'_{i-1}, w_i). \tag{1}$$

Where the parameter w_i is some state information passed from Sign^*_{Phase1} to Sign^*_{Phase2}. After this round of signing, the secret state will be updated to (S_i, S'_i).

In the presence of leakage, an adversary $\mathcal{A}^*$ can obtain some leakage information in addition to the signatures for some messages of its choice. To modeling thus scenario, we define a Sign&Leak oracle $\mathcal{O}^{\mathsf{Leak}}_{\mathsf{Sign}}$. In this oracle, besides the messages chosen by $\mathcal{A}^*$ to the signing oracle, it also allowed to specify two leakage functions f_i and h_i with bounded range $\{0,1\}^\lambda$ (where λ be the leakage parameter). The leakage functions defined as

$$\Lambda_i = f_i(S_{i-1}, r_i); \Lambda'_i = h_i(S'_{i-1}, r'_i, w_i). \tag{2}$$

We define the security notion of existential unforgeability under adaptive chosen message and leakage attacks (EUF-CMLA) through the following game $\mathcal{G}^{euf-cmla}_{\Sigma^*,\mathcal{A}^*}$, where $|f_i|$ denotes the length of the output of f_i.

Game $\mathcal{G}^{euf-cmla}_{\Sigma^*,\mathcal{A}^*}(1^k)$	**Oracle** $\mathcal{O}^{\mathsf{Leak}}_{\mathsf{Sign}}(m_i, f_i, h_i)$
$(pk, (S_0, S'_0)) \xleftarrow{\$} \mathsf{KGen}^*(1^k), i \leftarrow 1$	if $\lvert f_i\rvert \neq \lambda$ or $\lvert h_i\rvert \neq \lambda$, return $\perp$
$(m^*, \sigma^*) \xleftarrow{\$} \mathcal{A}^{*\mathcal{O}^{\mathsf{Leak}}_{\mathsf{Sign}}}(pk)$	$(S_i, w_i) \xleftarrow{r_i} \mathsf{Sign}^*_{Phase1}(S_{i-1}, m_i)$
if $1 \leftarrow \mathsf{Vrfy}^*(pk, m^*, \sigma^*)$ and $m^* \notin \{m_1, \ldots, m_i\}$	$(S'_i, \sigma_i) \xleftarrow{r'_i} \mathsf{Sign}^*_{Phase2}(S'_{i-1}, w_i)$
then output 1 else output 0	$\Lambda_i = f_i(S_{i-1}, r_i)$
	$\Lambda'_i = h_i(S'_{i-1}, r'_i, w_i)$
	return $(\sigma_i, \Lambda_i, \Lambda'_i)$ and set $i \leftarrow i + 1$

Adversary $\mathcal{A}^*$ wants to give a forgery (m^*, σ^*) by means of adaptive query to the Sign&Leakage oracle $\mathcal{O}^{\mathsf{Leak}}_{\mathsf{Sign}}$. We denote the advantage of $\mathcal{A}^*$ wins the above game by $\mathbf{Adv}^{euf-cmla}_{\Sigma^*,\mathcal{A}^*}$.

Definition 5. *The signature scheme Σ^* is* EUF-CMLA *secure if have no polynomial-bounded adversary can win the above game with a non-negligible advantage.*

4 Boneh-Lynn-Shacham Signature Scheme

In ASIACRYPT 2001, Boneh, Lynn, and Shacham [13] proposed a very efficient short signature scheme. It has been received great attention and adopted to construct many more complicated cryptographic schemes, e.g., aggregate signature [12]. The BLS signature scheme $\Sigma_{\mathsf{BLS}} = (\mathsf{KGen}_{\mathsf{BLS}}, \mathsf{Sign}_{\mathsf{BLS}}, \mathsf{Vrfy}_{\mathsf{BLS}})$ constructed as follows:

- $\mathsf{KGen}_{\mathsf{BLS}}(1^k)$:
  - $(G_1, G_2, p, g, \hat{e}) \xleftarrow{\$} \mathsf{BGen}(1^k)$ and choose a cryptographic hash function $H : \{0,1\}^* \to G_1$, then set the system public parameter as $\mathbb{P} = (G_1, G_2, p, g, \hat{e}, H)$.
  - Choose random $x \xleftarrow{\$} \mathbb{Z}_p$ and compute $v = g^x \in G_1$.
  - Output $pk = v$ and $sk = x$.
- $\mathsf{Sign}_{\mathsf{BLS}}(\mathbb{P}, sk, m)$: Compute and output the signature $\sigma = H(m)^x$.
- $\mathsf{Vrfy}_{\mathsf{BLS}}(\mathbb{P}, pk, m, \sigma)$: Check whether $\frac{\hat{e}(\sigma, g)}{\hat{e}(H(m), v)} \stackrel{?}{=} 1_{G_2}$.

Theorem 1 ([13]). *The BLS signature scheme Σ_{BLS} is* EUF-CMA *secure in the random oracle model based on the CDH assumption.*

4.1 Probabilistic BLS Signature Scheme

We adapt the deterministic BLS signature to a probabilistic scheme, which denoted by $\Sigma_{\mathsf{pBLS}} = (\mathsf{KGen}_{\mathsf{pBLS}}, \mathsf{Sign}_{\mathsf{pBLS}}, \mathsf{Vrfy}_{\mathsf{pBLS}})$ constructed as follows:

- $\mathsf{KGen}_{\mathsf{pBLS}}(1^k)$:
  - $(G_1, G_2, p, g, \hat{e}) \xleftarrow{\$} \mathsf{BGen}(1^k)$ and choose a cryptographic hash function $H : \{0,1\}^* \to G_1$, then set the system public parameter as $\mathbb{P} = (G_1, G_2, p, g, \hat{e}, H)$.

  - Choose random $x \xleftarrow{\$} \mathbb{Z}_p$, then compute $X = g^x \in G_1$ and $X_T = \hat{e}(X, g) = \hat{e}(g, g)^x \in G_2$.
  - Output $pk = X_T$ and $sk = X$.
- $\mathsf{Sign}_{\mathsf{pBLS}}(\mathbb{P}, sk, m)$:
  - Choose random $r \xleftarrow{\$} \mathbb{Z}_p$.
  - Compute and output $\sigma = (\sigma_1, \sigma_2) = (X \cdot H(m)^r, g^r)$.
- $\mathsf{Vrfy}_{\mathsf{pBLS}}(\mathbb{P}, pk, m, \sigma)$: Check whether $\frac{\hat{e}(\sigma_1, g)}{\hat{e}(\sigma_2, H(m))} \stackrel{?}{=} X_T$.

In fact, the probabilistic BLS signature scheme is parallels to the Galindo et al.'s basic signature scheme (cf. Section 3 of paper [21]). In their scheme, $\sigma_1 = X \cdot (X_0 \cdot X_1^m)^r$, which can be regarded as a design without random oracle model. However, we construct the probabilistic BLS signature mainly based on the considerations of the efficiency and the length of the public key. Furthermore, the message space of our scheme is more flexible than theirs.

Similar to the Galindo et al.'s basic scheme (which cannot proved in the standard model), unfortunately, we cannot prove the security of the probabilistic BLS scheme even in the random oracle model as the BLS scheme does, but we can prove it in the combinational models of the random oracle and generic bilinear group. Which means that in the generic bilinear groups model, the hash function H is treated as a random oracle. For space reasons, here we only give the security result and the complete proof is presented in the Appendix A.

Theorem 2. *The probabilistic BLS signature scheme Σ_{pBLS} is* EUF-CMA *secure w.r.t. the Definition 4 in the combinational models of random oracle and generic bilinear group. The advantage of a q-query adversary is $O(\frac{q^2}{p})$.*

4.2 Leakage-Resilient Probabilistic BLS Signature Scheme

We now adapt the probabilistic BLS signature scheme to the leakage resilient setting, i.e., leakage-resilient probabilistic BLS signature scheme $\Sigma^*_{\mathsf{pBLS}} = (\mathsf{KGen}^*_{\mathsf{pBLS}}, \mathsf{Sign}^*_{\mathsf{pBLS}}, \mathsf{Vrfy}^*_{\mathsf{pBLS}})$ which constructed as follows:

- $\mathsf{KGen}^*_{\mathsf{pBLS}}(1^k)$:
  - $(G_1, G_2, p, g, \hat{e}) \xleftarrow{\$} \mathsf{BGen}(1^k)$ and choose a cryptographic hash function $H : \{0,1\}^* \to G_1$, then set the system public parameter as $\mathbb{P} = (G_1, G_2, p, g, \hat{e}, H)$.
  - Choose random $x \xleftarrow{\$} \mathbb{Z}_p$, then compute $X = g^x \in G_1$ and $X_T = \hat{e}(X, g) = \hat{e}(g, g)^x \in G_2$.
  - Choose random $l_0 \xleftarrow{\$} \mathbb{Z}_p$ and set $(S_0, S_0') = (g^{l_0}, g^{x - l_0})$.
  - Output $pk = X_T$ and $sk_0 = (S_0, S_0')$.
- $\mathsf{Sign}^*_{\mathsf{pBLS}}(\mathbb{P}, sk_{i-1}, m)$:
  - *Phase 1* $(\mathbb{P}, S_{i-1}, m)$:
    - Choose random $l_i \xleftarrow{\$} \mathbb{Z}_p$ and compute $S_i = S_{i-1} \cdot g^{l_i}$.
    - Choose random $r \xleftarrow{\$} \mathbb{Z}_p$ and compute $(\sigma_1', \sigma_2') = (S_i \cdot H(m)^r, g^r)$.
    - Output $w_i = (l_i, \sigma_1', \sigma_2')$.

- *Phase 2* $(\mathbb{P}, S'_{i-1}, w_i)$:
  - Compute $S'_i = S'_{i-1} \cdot g^{-l_i}$.
  - Compute and output $\sigma = (\sigma_1, \sigma_2) = (S'_i \cdot \sigma'_1, \sigma'_2)$.

– $\mathsf{Vrfy}^*_{\mathsf{pBLS}}(\mathbb{P}, pk, m, \sigma)$: Check whether $\frac{\hat{e}(\sigma_1, g)}{\hat{e}(\sigma_2, H(m))} \stackrel{?}{=} X_T$.

At the beginning of the each signing phase, the partial secret key will be re-randomized, however, for any i, let $L_i := \sum_{j=0}^{i} l_j$, then $S_i \cdot S'_i = g^{L_i} \cdot g^{\alpha - L_i} = X$. Hence, actually, the signatures of the scheme Σ^*_{pBLS} identical to that from scheme Σ_{pBLS}. However, precisely because of the re-randomized process, adversary cannot collect enough leakage information of the "fresh" secret state to recover the real secret key X.

In the first phase of the the signing algorithm, it requires three exponentiations, and no exponentiation in the second phase if we see g^{-l} as the inverse element of g^l which has been calculated in the first phase. Hence it requires three exponentiations in every signature calculation. In addition, it requires two pairing operations in the verification algorithm.

Because of the values of the input and output of the schemes Σ^*_{pBLS} and Σ_{pBLS} are identical, the security of Σ^*_{pBLS} in the non-leakage setting is obviously. That is to say,

Lemma 3. *The probabilistic BLS signature scheme Σ^*_{pBLS} is* EUF-CMA *secure w.r.t. the Definition 4 in the combinational models of random oracle and generic bilinear group. The advantage of a q-query adversary is $O(\frac{q^2}{p})$.*

Considering the security in the leakage-resilient setting, for space reasons, here we only give the security result and the complete proof is presented in the Appendix B.

Theorem 3. *The probabilistic BLS signature scheme Σ^*_{pBLS} is* EUF-CMLA *secure w.r.t. the Definition 5 in the combinational models of random oracle and generic bilinear group. The advantage of a q-query adversary who gets at most λ bits of leakage per each invocation of Sign^*_{Phase1} or Sign^*_{Phase2} is $O(\frac{q^2}{p} 2^{2\lambda})$.*

5 Waters Signature Scheme

The Waters signature scheme [33], $\Sigma_{\mathsf{W}} = (\mathsf{KGen}_{\mathsf{W}}, \mathsf{Sign}_{\mathsf{W}}, \mathsf{Vrfy}_{\mathsf{W}})$, constructed as follows:

– $\mathsf{KGen}_{\mathsf{W}}(\mathbb{P})$:
- $\mathbb{P} = (G_1, G_2, p, g, \hat{e}) \stackrel{\$}{\leftarrow} \mathsf{BGen}(1^k)$.
- Choose random $x_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and compute $X_1 = g^{x_1}$.
- Choose random $X_2 \stackrel{\$}{\leftarrow} G_1$.
- Choose random $U_i \stackrel{\$}{\leftarrow} G_1, i \in [0, n]$ and set $\mathcal{U} = \{U_i\}_{i \in [0,n]}$.
- Output $pk = (X_1, X_2, \mathcal{U})$ and $sk = X = X_2^{x_1}$.

– $\mathsf{Sign}_{\mathsf{W}}(\mathbb{P}, sk, m)$:
- $m = m_1 m_2 \cdots m_n$, where m_i denotes the i^{th} bit of the m.

- Choose random $r \xleftarrow{\$} \mathbb{Z}_p$.
- Compute $(\sigma_1, \sigma_2) = (X \cdot (U_0 \prod_{i \in \mathcal{M}} U_i)^r, g^r)$, where $\mathcal{M}$ is the set of all i such that $m_i = 1$.
- Output $\sigma = (\sigma_1, \sigma_2)$.

– $\mathsf{Vrfy}_{\mathsf{W}}(\mathbb{P}, pk, m, \sigma)$: Check whether $\frac{\hat{e}(\sigma_1, g)}{\hat{e}(\sigma_2, U_0 \prod_{i \in \mathcal{M}} U_i)} \stackrel{?}{=} \hat{e}(X_1, X_2)$.

Theorem 4 ([33]). *The Waters signature scheme Σ_{W} is* EUF-CMA *secure in the standard model based on the DBDH assumption.*

5.1 Leakage-Resilient Waters Signature Scheme

We now adapt the Waters signature scheme to the leakage-resilient setting, i.e., leakage-resilient Waters signature scheme $\Sigma^*_{\mathsf{W}} = (\mathsf{KGen}^*_{\mathsf{W}}, \mathsf{Sign}^*_{\mathsf{W}}, \mathsf{Vrfy}^*_{\mathsf{W}})$ which constructed as follows:

– $\mathsf{KGen}^*_{\mathsf{W}}(\mathbb{P})$:
- $\mathbb{P} = (G_1, G_2, p, g, \hat{e}) \xleftarrow{\$} \mathsf{BGen}(1^k)$.
- Choose random $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p$ and compute $(X_1, X_2) = (g^{x_1}, g^{x_2})$, then set $X_T = \hat{e}(X_1, X_2)$.
- Choose random $u_i \xleftarrow{\$} \mathbb{Z}_p, i \in [0, n]$ and compute $U_i = g^{u_i} \in G_1$, then set $\mathcal{U} = \{U_i\}_{i \in [0,n]}$.
- Choose random $l_0 \xleftarrow{\$} \mathbb{Z}_p$ and set $(S_0, S'_0) = (X_2^{l_0}, X_2^{x_1 - l_0})$.
- Output $pk = (X_T, \mathcal{U})$ and $sk_0 = (S_0, S'_0)$.

– $\mathsf{Sign}^*_{\mathsf{W}}(\mathbb{P}, sk_{i-1}, m)$:
- *Phase 1* $(\mathbb{P}, S_{i-1}, m)$:
  - Let $m = m_1 m_2 \cdots m_n$, where m_i denotes the i^{th} bit of the m.
  - Choose random $l_i \xleftarrow{\$} \mathbb{Z}_p$ and compute $S_i = S_{i-1} \cdot X_2^{l_i}$.
  - Choose random $r \xleftarrow{\$} \mathbb{Z}_p$ and compute $(\sigma'_1, \sigma'_2) = (S_i \cdot (U_0 \prod_{i \in \mathcal{M}} U_i)^r, g^r)$.
  - Output $w_i = (l_i, \sigma'_1, \sigma'_2)$.
- *Phase 2* $(\mathbb{P}, S'_{i-1}, w_i)$:
  - Compute $S'_i = S'_{i-1} \cdot X_2^{-l_i}$.
  - Compute and output $\sigma = (\sigma_1, \sigma_2) = (S'_i \cdot \sigma'_1, \sigma'_2)$.

– $\mathsf{Vrfy}^*_{\mathsf{W}}(\mathbb{P}, pk, m, \sigma)$: Check whether $\frac{\hat{e}(\sigma_1, g)}{\hat{e}(\sigma_2, U_0 \prod_{i \in \mathcal{M}} U_i)} \stackrel{?}{=} X_T$.

In the key generation algorithm, we change the strategies that choosing the random elements X_2 and $U_i (i = 1, \ldots, n)$ from G_1 to choosing random integers x_2 and $u_i (i = 1, \ldots, n)$ from $\mathbb{Z}_p$, and then compute $X_2 = g^{x_2}$ and $U_i = g^{u_i} (i = 1, \ldots, n)$, respectively. However, the new strategies is indistinguishable from the old ones to anyone, and it also cannot provide more information to the potential adversary even in the presence of leakage if we restrict the leakage functions to apply only to those values used by the signer after the key generation phase, i.e., the secret key and all state variables used to sign, but not the randomness used to generate the secret-public key.

At the beginning of the each signing phase, the partial secret key will be re-randomized, however, for any i, let $L_i := \sum_{j=0}^{i} l_j$, then $S_i \cdot S'_i = X_2^{L_i} \cdot X_2^{x_1 - L_i} = X$. Hence, actually, the signature of this scheme Σ^*_{W} identical to that from scheme Σ_{W}.

In the first phase of the the signing algorithm, it requires three exponentiations, and no exponentiation in the second phase if we see X_2^{-l} as the inverse element of X_2^l which has been calculated in the first phase. Hence it requires three exponentiations in every signature calculation. In addition, it requires two pairing operations in the verification algorithm.

Because of the values of the input and output of the schemes Σ^*_{W} and Σ_{W} are identical, the security of Σ^*_{W} in the non-leakage setting is obviously. That is to say,

Lemma 4. *The signature scheme Σ^*_{W} is* EUF-CMA *secure in standard model based on the DBDH assumption.*

The security of the scheme Σ^*_{W} in the leakage-resilient setting be proved in the generic bilinear group model. For space reasons, here we only give the security result and the complete proof is presented in the Appendix C.

Theorem 5. *The signature scheme Σ^*_{W} is* EUF-CMLA *secure w.r.t. the Definition 5 in the generic bilinear group model. The advantage of a q-query adversary who gets at most λ bits of leakage per each invocation of Sign^*_{Phase1} or Sign^*_{Phase2} is $O(\frac{q^2}{p}2^{2\lambda})$.*

6 Comparison

We compare our schemes $\Sigma^*_{\mathsf{pBLS}}, \Sigma^*_{\mathsf{W}}$ to Galindo and Vivek's scheme Σ^*_{BB} [21]. All of them have similar security results: if allowing λ bits of leakage at every signing process then the security of the schemes decreased by at most a factor $2^{2\lambda}$, and thus they can tolerate $\frac{1-o(1)}{2}\log p$ bits per each signing invocation. We now compare them from the aspects of their length of public key, signing cost, and verification cost. The results of the comparison in the table below. Where $|G_1|, |G_2|$ denote the length of the element in group G_1 and G_2, respectively, $\mathbf{e}$ denotes an exponentiation computation and $\mathbf{p}$ denotes a pairing computation.

Scheme	Length of public key	Signing cost	Verification cost
Σ^*_{BB}	$2\|G_1\| + \|G_2\|$	$4\mathbf{e}$	$\mathbf{e} + 2\mathbf{p}$
Σ^*_{pBLS}	$\|G_2\|$	$3\mathbf{e}$	$2\mathbf{p}$
Σ^*_{W}	$(n+1)\|G_1\| + \|G_2\|$	$3\mathbf{e}$	$2\mathbf{p}$

Table 1. Comparing the three schemes.

From the above table we can see that both of our two schemes are more efficient than the scheme Σ^*_{BB}, especially the scheme Σ^*_{pBLS} not only has a low computation cost, but also has a short public key. The public key of the scheme Σ^*_{W} is long, however, its security can be guaranteed in the standard model in the black-box model which without any information leakage (both Σ^*_{BB} and Σ^*_{pBLS} do not have this property), and from this point we can see that proving the cryptographic scheme's security in the leakage-resilient setting is more intractable than in the traditional black-box model.

Finally, we may note that because of the similarity of the structure, the blinding technique also can be used to convert the Lewko-Waters signature (which can be constructed from their IBE scheme [26]) to the leakage-resilient setting. However, the result scheme, i.e., leakage-resilient LW

signature scheme, cannot improve the computational efficiency relative to the leakage-resilient BB scheme Σ^*_{BB}. Hence, we omit the analysis of this scheme in this paper.

References

1. A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography aginst memory attacks. In *TCC'09*, LNCS 5444, Springer-Verlag, pp. 474–495, 2009.
2. J. Alwen, Y. Dodis, and D. Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *Advances in Cryptology-CRYPTO'09*, LNCS 5677, Springer-Verlag, pp. 36–54.
3. J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, and D. Wichs. Public-key encryption in the bounded-retrieval model. In *Advances in Cryptology-EUROCRYPT'10*, LNCS 6110, Springer-Verlag, pp. 113–134, 2010.
4. D. Aggarwal and U. Maurer. The leakage-resilience limit of a computational problem is equal to its unpredictability entropy. In *Advances in Cryptology-ASIACRYPT'11*, LNCS 7073, Springer-Verlag, pp. 686–701, 2011.
5. E. Biham, Y. Carmeli, and A. Shamir. Bug attacks. In *Advances in Cryptology-CRYPTO'08*, LNCS 5157, Springer-Verlag, pp. 221–240, 2008.
6. Z. Brakerski, Y.T. Kalai, J. Katz, and V. Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *FOCS'10*, full version: http://eprint.iacr.org/2010/278.
7. E. Boyle, G. Segev, and D. Wichs. Fully leakage-resilient signatures. In *Advances in Cryptology-EUROCRYPT'11*, LNCS 6632, Springer-Verlag, pp. 89–108, 2011.
8. D. Boneh, and D. Brumley. Remote timing attacks are practical. In *Computer Networks*, vol. 48(5), pp. 701–716, 2005.
9. D. Boneh, R.A. Demillo, and R.J. Lipton. On the importance of checking cryptographic protocols for faults. In *Advances in Cryptology-EUROCRYPT'97*, LNCS 1233, Springer-Verlag, pp. 37–51, 1997.
10. D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Advances in Cryptology-EUROCRYPT'04*, LNCS 3027, Springer-Verlag, pp. 223–238, 2004.
11. D. Boneh, X. Boyen, and E.J. Goh. Hierarchical identity-based encryption with constant size ciphertext. In *Advances in Cryptology-EUROCRYPT'05*, LNCS 3494, Springer-Verlag, pp. 440–456, 2005.
12. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology-EUROCRYPT'03*, LNCS 2656, Springer-Verlag, pp. 416–432, 2003.
13. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Advances in Cryptology-ASIACRYPT'01*, LNCS 2248, Springer-Verlag, pp. 514-532, 2001.
14. S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters. Practical leakage-resilient identity-based encryption from simple assumptions. In *CCS'10*, ACM, pp. 152–161, 2010.
15. Y. Dodis, Y. Kalai, and S. Lovett. On cryptography with auxiliary input. In *STOC'09*, ACM, pp. 621–630, 2009.
16. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: how tow generate strong keys from biometrics and other noisy data. In *SIAM J. Comput.* 38(1), pp. 97–139, 2008.
17. Y. Dodis, K. Haralambiev, A. Lopez-Alt, and D. Wichs. Cryptography against continuous memory attacks. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pp. 511–520, 2010.
18. Y. Dodis, K. Haralambiev, A. Lopez-Alt, and D. Wichs. Efficient public-key cryptography in the presence of key leakage. In *Advances in Crytology-ASIACRYPT'10*, LNCS 6477, Springer-Verlag, pp. 613–631, 2010.
19. S. Faust, E. Kiltz, K. Pietrzak, and G. N. Rothblum. Leakage-resilient signatures. In *TCC'10*, LNCS 5978, Springer-Verlag, pp. 343–360, 2010.
20. S. Faust, C. Hazay, J.B. Nielsen, P.S. Nordholt, and A. Zottarel. Signature schemes secure against hard-to-invert leakage. In *Advances in Cryptology-ASIACRYPT'12*, LNCS 7658, Springer-Verlag, pp. 98–115, 2012.
21. D. Galindo and S. Vivek. A practical leakage-resilient signature scheme in the generic group model. In *SAC'12*, LNCS 7707, Springer-Verlag, pp. 50–65, 2013.
22. S. Garg, A. Jain, and A. Sahai. Leakage-resilient zero knowledge. In *Advances in Cryptology-CRYPTO'11*, Springer-Verlag, pp. 297–315, 2011.
23. J. Katz, and V. Vaikuntanathan. Signature schemes with bounded leakage resilience. In *Advances in Cryptology-ASIACRYPT'09*, LNCS 5912, Spinger-Verlag, pp. 703–720, 2009.

24. E. Kiltz and K. Pietrzak. Leakage resilient ElGamal encryption. In *Advances in Cryptology-AISACRYPT'10*, LNCS 6477, Springer-Verlag, pp. 595–612, 2010.
25. P.C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Advances in Cryptology-CRYPTO'99*, LNCS 1666, Springer-Verlag, pp. 388–397, 1999.
26. A. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC'10*, LNCS 5978, Springer-Verlag, pp. 455–479, 2010.
27. S. Micali, and L. Reyzin. Physically observable cryptography. In *TCC'04*, LNCS 2951, Springer-Verlag, pp. 278–296, 2004.
28. T. Malkin, I. Teranishi, Y. Vahlis, and M. Yung. Signatures resilient to continual leakage on memory and computation. In *TCC'11*, LNCS 6597, Springer-Verlag, pp. 89–106, 2011.
29. M. Naor, and G. Segev. Public-key cryptosystems resilient to key leakage. In *Advances in Cryptology-CRYPTO'09*, LNCS 5677, Springer-Verlag, pp. 18–35, 2009.
30. K. Pietrzak. A leakage-resilient mode of operation. In *Advances in Cryptology-EUROCRYPT'09*, LNCS 5479, Springer-Verlag, pp. 462–482, 2009.
31. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology-EUROCRYPT'97*, LNCS 1233, Springer-Verlag, pp. 256–266, 1997.
32. J.T. Schwartz. Fast probabilistic algorithms for verfication of polynomial identities. In *J. ACM* 27(4), pp. 701–717, 1980.
33. B. Waters. Efficient identity-based encryption without random oracle models. In *Advances in Cryptology-EUROCRYPT'05*, LNCS 3494, Springer-Verlag, pp. 114–127, 2005.
34. T.H. Yuen, S.S.M. Chow, Y. Zhang, and S.M. Yiu. Identity-based encryption resilient to continual auxiliary leakage. In *Advances in Cryptology-EUROCRYPT'12*, LNCS 7237, Springer-Verlag, pp. 117–134, 2012.
35. T.H. Yuen, S.M. Yiu, and L.C.K. Hui. Fully leakage-resilient signatures with auxiliary inputs. In *ACISP'12*, LNCS 7372, Springer-Verlag, pp. 294–307, 2012.
36. R. Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM'79*, LNCS 72, Springer-Verlag, pp. 216–226, 1979.

A Proof of Theorem 2

Let $\mathcal{A}$ be an adversary can break the security of the scheme Σ_{pBLS}. Without loss of generality, we assume that $\mathcal{A}$ is allowed to make totally at most q queries, which contains q_g group oracles $(\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_{\hat{e}})$ queries, q_h random oracle $(\mathcal{O}_H)$ queries, and q_s signing oracle $(\mathcal{O}_{\mathsf{Sign}})$ queries, i.e., $q_g + q_h + q_s \leq q$. We bound the advantage of $\mathcal{A}$ against Σ_{pBLS} in the following game $\mathcal{G}$ (cf. papers [11, 21, 31]). $\mathcal{A}$ plays the game $\mathcal{G}$ with a simulator $\mathcal{S}$ as follows.

`Game` $\mathcal{G}$: Let $X, \{H_i : i \in [q_h]\}, \{R_i : i \in [q_s]\}, \{Y_i : i \in [q_{g_1}], q_{g_1} \in [0, 2q_g + 2]\}$, and $\{Z_i : i \in [q_{g_2}], q_{g_2} \in [0, 2q_g]\}$ be indeterminates. They are correspond to randomly chosen group elements in the scheme Σ_{pBLS}, or more precisely their discrete logarithms. That is to say, X corresponds to x. R_i corresponds to the randomness r_i that used in the signature invocation. Besides that, $\mathcal{A}$ may query the group oracles with some bit strings that not previously obtained from the group oracles. In order to record thus values we introduce indeterminates Y_i and Z_i which correspond to the discrete logarithm of the elements of G_1 and G_2, respectively. H_i corresponds to the discrete logarithms of the random and independent elements chosen from G_1 which correspond to the hash values of the messages, this is why we need to see the hash function H as a random oracle. Without loss of generality, we assume that the first q_s queries of the $\mathcal{O}_H$, i.e., $\{H_i : i \in [q_s]\}$, correspond to the hash values of messages $\{m_i : i \in [q_s]\}$ that chosen by $\mathcal{A}$ used to query to the signing oracle, and the $(q_s + 1)^{th}$ query, i.e., H_{q_s+1}, corresponds to message m^* that also chosen by $\mathcal{A}$ as the message of its forgery. For simplicity sake, we denote them by $\{R\}, \{Y\}, \{Z\}$, and $\{H\}$, respectively.

$\mathcal{S}$ maintains the following two lists of polynomial-string pair to answer and record $\mathcal{A}$'s queries

$$\mathcal{L}_1 = \{(F_{1,i}, \xi_{1,i}) : i \in [\tau_1]\}, \tag{3}$$

$$\mathcal{L}_2 = \{(F_{2,i}, \xi_{2,i}) : i \in [\tau_2]\}, \tag{4}$$

where $F_{1,i} \in \mathbb{Z}_p[X, \{H\}, \{R\}, \{Y\}], F_{2,i} \in \mathbb{Z}_p[X, \{H\}, \{R\}, \{Y\}, \{Z\}]$ and $\xi_{1,i}, \xi_{2,i}$ are bit strings from the encoding sets Ξ_1 (of group G_1) and Ξ_2 (of group G_2), respectively.

Initially, i.e., at step $\tau = 0$ and $\tau_1 = 2q_s + q_h + q_{g_1} + 1, \tau_2 = q_{g_2} + 1$, $\mathcal{S}$ creates the following lists

$$\begin{aligned}
\mathcal{L}_1 = \Big\{ & (1, \xi_{1,1}), \{(H_i, \xi_{1,i+1}) : i \in [q_h]\}, \{(Y_i, \xi_{1,i+q_h+1}) : i \in [q_{g_1}]\}, \\
& \{(X + R_i H_i, \xi_{1,2i+q_h+q_{g_1}}), (R_i, \xi_{1,2i+q_h+q_{g_1}+1}) : i \in [q_s]\} \Big\}, \\
\mathcal{L}_2 = \Big\{ & (X, \xi_{2,1}), \{(Z_i, \xi_{2,i+1}) : i \in [q_{g_2}]\} \Big\},
\end{aligned}$$

where $\xi_{1,i}, \xi_{2,i}$ are chosen randomly and distinctly from Ξ_1 and Ξ_2, respectively. We assume that the entries in the sets Ξ_1 and Ξ_2 are recorded in order, and thus given a string $\xi_{1,i}$ or $\xi_{2,i}$, it is able to determine its index in the lists if it exists. Similarly, the entries $\{(H_i, \xi_{1,i+1}) : i \in [q_h]\}$ has an ordering and thus given a message m, then it is able to determine its index in these entries if it exists. At step $\tau \in [0, q_g + q_h]$ of the game,

$$\tau_1 + \tau_2 = \tau + 2q_s + q_h + q_{g_1} + q_{g_2} + 2. \tag{5}$$

For the initial entries of the two lists, they are correspond to the group elements of the public parameters and the signatures on the corresponding messages which chosen by $\mathcal{A}$. $\{Y\}$ and $\{Z\}$ correspond to the group elements that $\mathcal{A}$ will guess in the actual interaction. In the game, $\mathcal{A}$ can query the group oracles with at most two new (guessed) elements and it also will output two new elements from G_1 as the forgery, hence $q_{g_1} + q_{g_2} \leq 2q_g + 2$. Therefore, from the equation (5) we have (w.l.o.g., assuming $q_h + q_s \geq 4$)

$$\tau_1 + \tau_2 \leq q_g + q_h + 2q_s + q_h + 2q_g + 2 + 2 \leq 3(q_g + q_h + q_s) \leq 3q. \tag{6}$$

`Random Oracle` $\mathcal{O}_H$: $\mathcal{A}$ queries the random oracle $\mathcal{O}_H$ with input m, $\mathcal{S}$ searches the entries $\{(H_i, \xi_{1,i}) : i \in [q_h]\}$ in $\mathcal{L}_1$, if there exists entry $\{(H_k, \xi_{1,k})\}$ for $k \leq q_h$ corresponds to m, then $\mathcal{S}$ returns $\xi_{1,k}$ to $\mathcal{A}$. Otherwise, it first increments the counter $\tau_1 := \tau_1 + 1$ and $\tau := \tau + 1$, then returns a random string $\xi_{1,i}$ distinct from those already contained in $\mathcal{L}_1$ to $\mathcal{A}$. Finally, adding $(H_i, \xi_{1,i})$ to $\mathcal{L}_1$.

`Group Oracles` $\mathcal{O}_1, \mathcal{O}_2$: For the group operations in group G_1, $\mathcal{A}$ takes as input two elements $\xi_{1,i}, \xi_{1,j} (i, j \in [\tau_1])$ in $\mathcal{L}_1$ and specifies whether to multiply or divide them. $\mathcal{S}$ first increments the counters $\tau_1 := \tau_1 + 1$ and $\tau := \tau + 1$, then computes $F_{1,\tau_1} = F_{1,i} + F_{1,j}$ if $\mathcal{A}$ calls for multiplication, or else $F_{1,\tau_1} = F_{1,i} - F_{1,j}$. If there exists $k < \tau_1$ such that $F_{1,\tau_1} = F_{1,k}$, then sets $\xi_{1,\tau_1} := \xi_{1,k}$. Otherwise, $\mathcal{S}$ chooses a random string ξ_{1,τ_1} that distinct from those already contained in $\mathcal{L}_1$. Finally, $\mathcal{S}$ adds the entry $(F_{1,\tau_1}, \xi_{1,\tau_1})$ to $\mathcal{L}_1$. Note that the degree of the polynomials $F_{1,i}$ in $\mathcal{L}$ is at most *two*. Similarly, $\mathcal{S}$ answers $\mathcal{A}$'s queries to the oracle $\mathcal{O}_2$, updates the list $\mathcal{L}_2$ and the counters τ_2 and τ.

`Pairing Oracle` $\mathcal{O}_{\hat{e}}$: $\mathcal{A}$ takes as input two elements $\xi_{1,i}, \xi_{1,j} (i, j \in [\tau_1])$ from the list $\mathcal{L}_1$ to this oracle. $\mathcal{S}$ first increments the counters $\tau_2 := \tau_2 + 1$ and $\tau := \tau + 1$, and then computes $F_{2,\tau_2} = F_{1,i} \cdot F_{1,j}$.

If there exists $k < \tau_2$ such that $F_{2,\tau_1} = F_{2,k}$, then sets $\xi_{2,\tau_1} := \xi_{2,k}$. Otherwise, $\mathcal{S}$ chooses random string ξ_{2,τ_2} that distinct from those already contained in $\mathcal{L}_2$. Finally, $\mathcal{S}$ adds the entry $(F_{2,\tau_2}, \xi_{2,\tau_2})$ to the list $\mathcal{L}_2$. The degree of the polynomials $F_{2,i} \in \mathcal{L}_2$ is at most *four*.

`Output`: After finishing the queries, $\mathcal{A}$ outputs $(m^*, (\xi_{1,\sigma_1}, \xi_{1,\sigma_2})) \in \mathbb{Z}_p \times \mathcal{L}_1 \times \mathcal{L}_1(\sigma_1, \sigma_2 \in [\tau_1])$. Which corresponds to the forgery outputted by $\mathcal{A}$ in the actual interaction and m^* was not taken as input to the signing oracle. Let F_{1,σ_1} and F_{1,σ_2} be the polynomials which correspond to ξ_{1,σ_1} and ξ_{1,σ_2} in $\mathcal{L}_1$, respectively. $\mathcal{S}$ computes the polynomial

$$F_{1,\sigma} = X + F_{1,\sigma_2} H_{q_s+1} - F_{1,\sigma_1}. \tag{7}$$

The degree of $F_{1,\sigma}$ is at most *three*. Then $\mathcal{S}$ chooses random and independent values $x, \{h\}, \{r\}, \{y\}$ and $\{z\}$ from $\mathbb{Z}_p$ and evaluates the polynomials in the lists $\mathcal{L}_1$ and $\mathcal{L}_2$. We say that $\mathcal{A}$ wins the game $\mathcal{G}$ if one of the following cases occurs:

- *Case 1:* $F_{1,i}(x, \{h\}, \{r\}, \{y\}) = F_{1,j}(x, \{h\}, \{r\}, \{y\})$ in $\mathbb{Z}_p$, for some two polynomials $F_{1,i} \neq F_{1,j}$ in list $\mathcal{L}_1$.
- *Case 2:* $F_{2,i}(x, \{h\}, \{r\}, \{y\}, \{z\}) = F_{2,j}(x, \{h\}, \{r\}, \{y\}, \{z\})$ in $\mathbb{Z}_p$, for some two polynomials $F_{2,i} \neq F_{2,j}$ in list $\mathcal{L}_2$.
- *Case 3:* $F_{1,\sigma}(x, \{h\}, \{r\}, \{y\}) = 0$ in $\mathbb{Z}_p$.

`Game` $\mathcal{G}$ *`vs.`* `actual EUF-CMA game`: The success probability of $\mathcal{A}$ in the actual EUF-CMA game is bounded by its success probability in the above game $\mathcal{G}$ with a negligible probability gap. The reasons are as follows:

- Case 1 means that $\mathcal{A}$ can provoke collisions among group elements of G_1, i.e., $F_{1,i} \neq F_{1,j}$ but $g^{F_{1,i}(x,\{h\},\{r\},\{y\})} = g^{F_{1,j}(x,\{h\},\{r\},\{y\})}$ in the actual EUF-CMA game with a fixed generator g of the group G_1, which can be used to solve the discrete logarithm problem of the group G_1 (cf. Lemma 1 of the full version of the paper [24]). Similarly, case 2 means that $\mathcal{A}$ can provoke collisions among group elements of G_2. As long as these two cases do not occur, then the view of $\mathcal{A}$ is identical that in the game $\mathcal{G}$ and in the actual interaction. Therefore, if $\mathcal{A}$ cannot provoke collisions, then its adaptive strategies are no more powerful than non-adaptive ones (for more details, we refer to [31]).
- Case 3 means that $(\xi_{1,\sigma_1}, \xi_{1,\sigma_2})$ is a valid forgery on a new message m^*.

`Advantage`: We now analyze the advantage of $\mathcal{A}$ in the game $\mathcal{G}$. Since $F_{1,i} \neq F_{1,j} \Leftrightarrow F' = F_{1,i} - F_{1,j} \neq 0$, and the degree of the polynomials in the list $\mathcal{L}_1$ at most two. According to the Schwartz-Zippel lemma (with $\Delta = 0$), $\mathbf{Pr}[F'(x, \{h\}, \{r\}, \{y\}) = 0] \leq \frac{2}{p}$. The list $\mathcal{L}_1$ has τ_1 entries, so there exists at most $\binom{\tau_1}{2}$ distinct pairs $(F_{1,i}, F_{1,j})$, the probability of the case 1 occurs is at most $\binom{\tau_1}{2} \cdot \frac{2}{p}$. Similarly, the probability of the case 2 occurs is at most $\binom{\tau_2}{2} \cdot \frac{4}{p}$. The degree of the polynomial $F_{1,\sigma}$ is at most three, so if it is non-zero (proved in Lemma 5 below), then we can use the Schwartz-Zippel Lemma to compute the probability of the case 3 occurs is at most $\frac{3}{p}$. In conclusion, the advantage of $\mathcal{A}$ wins the game $\mathcal{G}$ is

$$\mathbf{Adv}^{euf-cma}_{\Sigma_{\mathsf{pBLS}},\mathcal{A}} \leq \binom{\tau_1}{2} \cdot \frac{2}{p} + \binom{\tau_2}{2} \cdot \frac{4}{p} + \frac{3}{p} \leq \frac{2}{p}(\tau_1 + \tau_2)^2 \leq \frac{18q^2}{p} = O(\frac{q^2}{p}). \tag{8}$$

Therefore, let $q = poly(\log p)$, then $\mathbf{Adv}^{euf-cma}_{\Sigma_{\mathsf{pBLS}},\mathcal{A}}$ is negligible.

Lemma 5. *$F_{1,\sigma}$ is a non-zero polynomial in $\mathbb{Z}_p[X, \{H\}, \{R\}, \{Y\}]$.*

Proof. From the design of the group oracles and the initial elements of the list $\mathcal{L}_1$, we can see that any polynomial in $\mathcal{L}_1$ is computed by either adding or subtracting two polynomials previously existing in the list. Therefore, w.l.o.g., we can write the forgery $(F_{1,\sigma_1}, F_{1,\sigma_2})$ as follows

$$F_{1,\sigma_1} = c_1 + \sum_{i=1}^{q_h} c_{2,i} H_i + \sum_{i=1}^{q_s} c_{3,i} R_i + \sum_{i=1}^{q_{g_1}} c_{4,i} Y_i + \sum_{i=1}^{q_s} c_{5,i}(X + R_i H_i), \tag{9}$$

$$F_{1,\sigma_2} = d_1 + \sum_{i=1}^{q_h} d_{2,i} H_i + \sum_{i=1}^{q_s} d_{3,i} R_i + \sum_{i=1}^{q_{g_1}} d_{4,i} Y_i + \sum_{i=1}^{q_s} d_{5,i}(X + R_i H_i), \tag{10}$$

where $c_1, d_1, c_{j,i}, d_{j,i} (j = 2, 3, 4, 5) \in \mathbb{Z}_p$ are chosen by $\mathcal{A}$. We divide them into two cases:

- *Case 1:* $c_{5,i} = d_{5,i} = 0$, for $\forall i \in [q_s]$.
  In this case, both F_{1,σ_1} and F_{1,σ_2} do not contain the indeterminate X. Hence the polynomial $F_{1,\sigma_2} H_{q_s+1} - F_{1,\sigma_1}$ in (7) also does not contain the determinate X. Therefore, in the polynomial $F_{1,\sigma}$, the coefficient of the term X is non-zero, and thus $F_{1,\sigma}$ is non-zero.
- *Case 2:* $c_{5,k} \neq 0$ or $d_{5,k} \neq 0$, for $\exists k \in [q_s]$.
  - If $d_{5,k} \neq 0$, then the coefficient of the term $R_k H_k H_{q_s+1}$ is non-zero, and thus $F_{1,\sigma}$ is non-zero.
  - If $c_{5,k} \neq 0$, then the coefficient of the term $R_k H_k$ is non-zero, and thus $F_{1,\sigma}$ is non-zero.

Therefore, the polynomial $F_{1,\sigma}$ is non-zero. This completes the proof of Theorem 2. □

B Proof of Theorem 3

Let $\mathcal{A}^*$ be an adversary can break the security of the scheme Σ^*_{pBLS}. Without loss of generality, we assume that $\mathcal{A}^*$ is allowed to make totally at most q queries, which contains q_g group oracles $(\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_{\hat{e}})$ queries, q_h random oracle $(\mathcal{O}_H)$ queries, and q_s signing oracle $(\mathcal{O}_{\mathsf{Sign}})$ queries, i.e., $q_g + q_h + q_s \leq q$. In the count q_g, the group oracle queries by leakage functions f_i, h_i specified by $\mathcal{A}^*$ are also included.

We define the following events:

- $\mathbf{E}$: $\mathcal{A}^*$ computes or guesses the secret key $X = g^x$ by any of the leakage functions f_i or $h_i (i \in [q_s])$.
- $\overline{\mathbf{E}}$: the complement of the event $\mathbf{E}$.
- $\mathbf{F}$: $\mathcal{A}^*$ outputs a forgery on a new message.

Hence, we have

$$\mathbf{Adv}^{euf-cmla}_{\Sigma^*_{\mathsf{pBLS}}, \mathcal{A}^*} = \mathbf{Pr}[\mathbf{F}|\mathbf{E}]\mathbf{Pr}[\mathbf{E}] + \mathbf{Pr}[\mathbf{F}|\overline{\mathbf{E}}]\mathbf{Pr}[\overline{\mathbf{E}}] \leq \mathbf{Pr}[\mathbf{E}] + \mathbf{Pr}[\mathbf{F}|\overline{\mathbf{E}}]. \tag{11}$$

We now bound the right two terms of the inequality, respectively.

Lemma 6. $\mathbf{Pr}[\mathbf{E}] \leq O(\frac{q^2}{p} 2^{2\lambda})$.

Proof. $\mathcal{A}^*$ plays the following game $\mathcal{G}^*$ with the simulator $\mathcal{S}$. Naturally, game $\mathcal{G}^*$ is similar to the game $\mathcal{G}$ in Theorem 2, and thus we use the notations introduced in the game $\mathcal{G}$ and only briefly describe $\mathcal{G}^*$. Let $\{L_i : i \in [0, q_s]\}$ be indeterminates which correspond to the values l_i in Σ^*_{pBLS}.
`Game` $\mathcal{G}^*$: For every signing process, $\mathcal{A}^*$ specifies two leakage functions f_i and h_i correspond to the signing phase 1 and 2, respectively, on the internal state which have involved in the computation, i.e., $f_i(S_{i-1}, (l_i, r_i))$ and $h_i(S'_{i-1}, (l_i, \sigma'_{i,1}, \sigma'_{i,2}))$ (for $|f_i| = |h_i| = \lambda$), respectively. In the mean time, $\mathcal{A}^*$ maintains two lists $\mathcal{L}_1^{f_i}$ and $\mathcal{L}_1^{h_i}$. These two lists contain polynomial-string pairs, where the polynomials are from $\mathbb{Z}_p[X, \{H\}, \{R\}, \{Y\}, \{L\}]$ and the strings are from the encoding set ξ_1 of G_1. Intuitively, the polynomials in the lists $\mathcal{L}_1^{f_i}$ and $\mathcal{L}_1^{h_i}$ correspond to the elements of group G_1 that can be computed by f_i and h_i, respectively. The polynomials in $\mathcal{L}_1^{f_i}$ are of the form

$$c_{1,i} \sum_{j=0}^{i-1} L_j + c_{2,i} L_i + c_{3,i} V_i, \tag{12}$$

where $c_{1,i}, c_{2,i}, c_{3,i} \in \mathbb{Z}_p$ are chosen by $\mathcal{A}^*$ and $V_i \in \mathcal{L}_1$. The polynomials in $\mathcal{L}_1^{h_i}$ are of the form

$$d_{1,i}(X - \sum_{j=0}^{i-1} L_j) + d_{2,i} L_i + d_{3,i}(\sum_{j=0}^{i} L_j + R_i \cdot H_i) + d_{4,i} W_i, \tag{13}$$

where $d_{1,i}, d_{2,i}, d_{3,i}, d_{4,i} \in \mathbb{Z}_p$ are also chosen by $\mathcal{A}^*$ and $W_i \in \mathcal{L}_1$.

When finishing the game, $\mathcal{A}^*$ outputs a polynomial F from the list $\mathcal{L}_1^{f_i}$ or $\mathcal{L}_1^{h_i}$. That is to say, the polynomial F is corresponds to $\mathcal{A}^*$'s guess of the secret key X. We say that $\mathcal{A}^*$ wins the game $\mathcal{G}^*$ if one of the following cases occurs:

- *Case 1*: $F - X = 0$ in $\mathbb{Z}_p$.
- *Case 2*: There exists collision in any of the lists $\mathcal{L}_1^{f_i}$ or $\mathcal{L}_1^{h_i}$, for some $i \in [q_s]$.

Since $|f_i| + |h_i| = 2\lambda$ and the result of the Lemma 1, the polynomials are now evaluated with values chosen from independent distributions with min-entropy $\log p - 2\lambda$.

Technically speaking, $\mathcal{A}^*$ should also maintain lists $\mathcal{L}_2^{f_i}$ and $\mathcal{L}_2^{h_i} (0 \le i \le q_s)$ which correspond to the elements of the group G_2 that can be computed by f_i and h_i. However, similar arguments of $\mathcal{L}_1^{f_i}$ and $\mathcal{L}_1^{h_i}$ also applied to the lists $\mathcal{L}_2^{f_i}$ and $\mathcal{L}_2^{h_i}$, respectively, hence the probability $\mathbf{Pr}[\mathbf{E}]$ only up to a constant factor.

For similar reasons as given in the proof of Theorem 2, $\mathbf{Pr}[\mathbf{E}]$ is bounded above by the success probability of $\mathcal{A}^*$ in the game $\mathcal{G}^*$, and even in the presence of leakage adaptive strategies are no more powerful than non-adaptive ones [4].
`Advantage`: We first prove that $F - X$ is a non-zero polynomial. If $c_{1,i} = c_{2,i} = 0$ in equation (13), then the lists $L_1^{f_i}$ do not contain the polynomial X. If $c_{1,i} \neq 0$ or $c_{2,i} \neq 0$, then the polynomials in (12) will contain polynomial L_i or L_{i-1}, or both. Hence the polynomial X cannot appear in any of the lists $\mathcal{L}_1^{f_i}$. Similarly, the lists $\mathcal{L}_1^{h_i}$ also cannot contain X. Therefore, $F - X$ is a non-zero polynomial of degree at most *two*. Hence $\mathbf{Pr}[Case1] \le \frac{2}{p} 2^{2\lambda}$.

In order to compute the probability of the case 2, we evaluate the polynomials in equations (12) and (13) by randomly and independently choose values from $\mathbb{Z}_p$ according to distributions with

min-entropy at least $\log p - 2\lambda$. The total length of the lists $\mathcal{L}_1^{f_i}, \mathcal{L}_1^{h_i}$ is at most $O(q_h + q_o + q_s) = O(q)$, and thus there can be at most $O(q^2)$ pairs of distinct polynomials of degree at most *two* evaluated to the same value. According to the Schwartz-Zippel Lemma, $\mathbf{Pr}[Case2] \leq O(\frac{q^2}{p}2^{2\lambda})$. Therefore, $\mathbf{Pr}[\mathbf{E}] \leq O(\frac{q^2}{p}2^{2\lambda})$.

Lemma 7. $\mathbf{Pr}[\mathbf{F}|\overline{\mathbf{E}}] \leq \frac{18q^2}{p}2^{\lambda}$.

Proof. If the event $\mathbf{E}$ has not occurred which means that $\mathcal{A}^*$ has not compute or guess the secret key X, the only meaningful leakage for $\mathcal{A}^*$ is that of $r_i (i = 1, \ldots, q_s)$. Since at most λ bits of r_i will be leaked by f_i, from the view point of $\mathcal{A}^*$ the values r_i have min-entropy at least $\log p - \lambda$. According to the Schwartz-Zippel lemma and the analysis of the cases 1, 2, 3 in the Theorem 2, $\mathbf{Pr}[\mathbf{F}|\overline{\mathbf{E}}] \leq \frac{18q^2}{p}2^{\lambda}$.

In conclusion, we have $\mathbf{Adv}^{euf-cmla}_{\Sigma^*_{\mathsf{pBLS}},\mathcal{A}^*} \leq O(\frac{q^2}{p}2^{2\lambda})$. From the point of the length of the leakage, it can be leaked $\frac{1-o(1)}{2}\log p$ bits per each signing process. This completes the proof of Theorem 3. □

C Proof of Theorem 5

Let $\mathcal{A}^*$ be an adversary can break the security of the scheme Σ^*_{W}. Without loss of generality, we assume that $\mathcal{A}^*$ can make totally at most q queries, which contains q_g group oracles $(\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_{\hat{e}})$ queries and q_s signing oracle $(\mathcal{O}_{\mathsf{Sign}})$ queries, i.e., $q_g + q_s \leq q$. In the count q_g, the group oracle queries by leakage functions f_i and h_i specified by $\mathcal{A}^*$ are also included. We bound the advantage of $\mathcal{A}^*$ against Σ^*_{W} in the following game $\mathcal{G}^*$. $\mathcal{A}^*$ plays the game $\mathcal{G}^*$ with a simulator $\mathcal{S}$ as follows.
`Game` $\mathcal{G}^*$: Let $X_1, X_2, \{U_i, i \in [0,n]\}, \{L_i, i \in [q_s]\}, \{R_i, i \in [q_s]\}, \{Y_i, i \in [q_{g_1}], q_{g_1} \in [0, 2q_g + 2]\}$, and $\{Z_i, i \in [q_{g_2}], q_{g_2} \in [0, 2q_g]\}$ be indeterminates. They are correspond to randomly chosen group elements from the scheme Σ^*_{W}, or more precisely their discrete logarithms. That is to say, X_1 corresponds to x_1. X_2 corresponds to x_2. U_i corresponds to u_i. L_i corresponds to the randomness l_i that used to update the secret key. R_i corresponds to the randomness r_i that used in the signature invocation. Besides that, $\mathcal{A}^*$ may query the group oracles with some bit strings that not previously obtained from the group oracles. In order to record thus values we introduce indeterminates Y_i and Z_i which correspond to the discrete logarithm of the elements of G_1 and G_2, respectively. For simplicity sake, we denote them by $\{U\}, \{R\}, \{Y\}$, and $\{Z\}$, respectively. Note that the secret key $X = X_2^{x_1} = g^{x_1 x_2}$, which depends on X_1 and X_2, hence we cannot define X as an independent indeterminate. Let $\{m_i : i \in [q_s]\}$ be the messages that chosen by $\mathcal{A}^*$ used to query to the signing oracle. We define the following events:

- $\mathbf{E}$: $\mathcal{A}^*$ computes or guesses the secret key X by any of the leakage functions f_i or $h_i (i \in [q_s])$.
- $\overline{\mathbf{E}}$: the complement of the event $\mathbf{E}$.
- $\mathbf{F}$: $\mathcal{A}^*$ outputs a forgery on a new message.

Hence, we have

$$\mathbf{Adv}^{euf-cmla}_{\Sigma^*_{\mathsf{W}},\mathcal{A}^*} = \mathbf{Pr}[\mathbf{F}|\mathbf{E}]\mathbf{Pr}[\mathbf{E}] + \mathbf{Pr}[\mathbf{F}|\overline{\mathbf{E}}]\mathbf{Pr}[\overline{\mathbf{E}}] \leq \mathbf{Pr}[\mathbf{E}] + \mathbf{Pr}[\mathbf{F}|\overline{\mathbf{E}}]. \tag{14}$$

$\mathcal{S}$ maintains the following two lists of pair to answer and record $\mathcal{A}^*$'s queries

$$\mathcal{L}_1 = \{(F_{1,i}, \xi_{1,i}) : i \in [\tau_1]\}, \tag{15}$$

$$\mathcal{L}_2 = \{(F_{2,i}, \xi_{2,i}) : i \in [\tau_2]\}, \tag{16}$$

where $F_{1,i} \in \mathbb{Z}_p[X_1, X_2, \{U\}, \{R\}, \{Y\}], F_{2,i} \in \mathbb{Z}_p[X_1, X_2, \{U\}, \{R\}, \{Y\}, \{Z\}]$ and $\xi_{1,i}, \xi_{2,i}$ are bit strings from the encoding sets Ξ_1 (of group G_1) and Ξ_2 (of group G_2), respectively.

Initially, i.e., at step $\tau = 0$ and $\tau_1 = 2q_s + n + q_{g_1} + 1, \tau_2 = q_{g_2} + 1$), $\mathcal{S}$ creates the following lists

$$\begin{aligned}
\mathcal{L}_1 = \Big\{ & (1, \xi_{1,1}), \{(U_i, \xi_{1,i+1}) : i \in [n]\}, \{(Y_i, \xi_{1,i+n+1}) : i \in [q_{g_1}]\}, \\
& \{(X_1 X_2 + (U_0 + \textstyle\sum_{j \in \mathcal{M}_i} U_j) R_i, \xi_{1,2i+n+q_{g_1}}), (R_i, \xi_{1,2i+n+q_{g_1}+1}) : i \in [q_s]\} \Big\}, \\
\mathcal{L}_2 = \Big\{ & (X_1 X_2, \xi_{2,1}), \{(Z_i, \xi_{2,i+1}) : i \in [q_{g_2}]\} \Big\},
\end{aligned}$$

where $\mathcal{M}_i$ (corresponds to m_i) be the set of all j such that $m_{i,j} = 1$, $\xi_{1,i}, \xi_{2,i}$ are chosen randomly and distinctly from Ξ_1 and Ξ_2, respectively. At step $\tau \in [0, q_g]$ of the game,

$$\tau_1 + \tau_2 = \tau + 2q_s + n + q_{g_1} + q_{g_2} + 2. \tag{17}$$

For the initial entries of the two lists, they are correspond to the group elements of the public parameters and the signatures on the corresponding message which chosen by $\mathcal{A}^*$. $\{Y\}, \{Z\}$ correspond to the group elements that $\mathcal{A}^*$ will guess in the actual interaction. In the game, $\mathcal{A}^*$ can query the group oracles with at most two new (guessed) elements and it also will output two new elements from G_1 as the forgery, so $q_{g_1} + q_{g_2} \leq 2q_g + 2$. Therefore, from the equation (17) we have (w.l.o.g., assuming $q_s \geq n + 4$)

$$\tau_1 + \tau_2 \leq q_g + 2q_s + n + 2q_g + 2 + 2 \leq 3(q_g + q_s) \leq 3q. \tag{18}$$

The `Group Oracles` $(\mathcal{O}_1, \mathcal{O}_2)$ and `Pairing Oracle` $(\mathcal{O}_{\hat{e}})$ are run by the adversary $\mathcal{A}^*$ and simulator $\mathcal{S}$ which is similar to that in the proof of the Theorem 2.

`Leakage`: For every signing process, $\mathcal{A}^*$ specifies two leakage functions f_i and h_i correspond to the signing phase 1 and 2, respectively, on the internal state which have involved in the computation, i.e., $f_i(S_{i-1}, (l_i, r_i))$ and $h_i(S'_{i-1}, (l_i, \sigma'_{i,1}, \sigma'_{i,2}))$ (for $|f_i| = |h_i| = \lambda$), respectively. In the mean time, $\mathcal{A}^*$ maintains two lists $\mathcal{L}_1^{f_i}$ and $\mathcal{L}_1^{h_i}$ which contain polynomial-string pairs, where the polynomials are from $\mathbb{Z}_p[X_1, X_2, \{U\}, \{R\}, \{Y\}, \{L\}]$ and the strings are from the encoding set ξ_1 of G_1. Intuitively, the polynomials in the lists $\mathcal{L}_1^{f_i}$ and $\mathcal{L}_1^{h_i}$ correspond to the elements of group G_1 that can be computed by f_i and h_i, respectively. The polynomials in $\mathcal{L}_1^{f_i}$ are of the form

$$c_{1,i} \sum_{j=0}^{i-1} L_j + c_{2,i} L_i + c_{3,i} V_i, \tag{19}$$

where $c_{1,i}, c_{2,i}, c_{3,i} \in Z_p$ are chosen by $\mathcal{A}^*$ and $V_i \in \mathcal{L}_1$. The polynomials in $\mathcal{L}_1^{h_i}$ are of the form

$$d_{1,i}(X_2(X_1 - \sum_{j=0}^{i-1} L_j)) + d_{2,i} L_i + d_{3,i}(\sum_{j=0}^{i} L_j + (U_0 + \sum_{j \in \mathcal{M}_i} U_j) R_i) + d_{4,i} W_i, \tag{20}$$

where $d_{1,i}, d_{2,i}, d_{3,i}, d_{4,i} \in \mathbb{Z}_p$ are also chosen by $\mathcal{A}^*$ and $W_i \in \mathcal{L}_1$.

$\mathcal{A}^*$ should also maintain lists $\mathcal{L}_2^{f_i}$ and $\mathcal{L}_2^{h_i} (0 \leq i \leq q_s)$ that correspond to the elements of the group G_2 that can be computed by f_i and h_i, which only up the probability $\mathbf{Pr}[\mathbf{E}]$ a constant factor. `Output`: After finishing the queries, $\mathcal{A}^*$ outputs $(m^*, (\xi_{1,\sigma_1}, \xi_{1,\sigma_2})) \in \mathbb{Z}_p \times \mathcal{L}_1 \times \mathcal{L}_1 (\sigma_1, \sigma_2 \in [\tau_1])$. It corresponds to the forgery outputted by $\mathcal{A}^*$ in the actual interaction. Let F_{1,σ_1} and F_{1,σ_2} be the polynomials which correspond to ξ_{1,σ_1} and ξ_{1,σ_2} in $\mathcal{L}_1$, respectively. $\mathcal{S}$ computes the polynomial:

$$F_{1,\sigma} = X_1 X_2 + F_{1,\sigma_2}(U_0 + \sum_{j \in \mathcal{M}^*} U_j) - F_{1,\sigma_1}. \tag{21}$$

The degree of $F_{1,\sigma}$ is at most *three*. Then $\mathcal{S}$ chooses random and independent values $x_1, x_2, \{u\}, \{r\}, \{y\}$, and $\{z\}$ from $\mathbb{Z}_p$ and evaluates the polynomials in the lists $\mathcal{L}_1$ and $\mathcal{L}_2$. We say that $\mathcal{A}^*$ wins the game $\mathcal{G}^*$ if one of the following cases occurs:

- *Case 1*: There exists collision among group G_1, i.e., $F_{1,i}(x_1, x_2, \{u\}, \{r\}, \{y\}) = F_{1,j}(x_1, x_2, \{u\}, \{r\}, \{y\})$ in $\mathbb{Z}_p$, for some two polynomials $F_{1,i} \neq F_{1,j}$ in list $\mathcal{L}_1$.
- *Case 2*: There exists collision among group G_2 in list $\mathcal{L}_2$.
- *Case 3*: There exists collision among group G_1 in list $\mathcal{L}_1^{f_i}$.
- *Case 4*: There exists collision among group G_1 in list $\mathcal{L}_1^{h_i}$.
- *Case 5*: There exists collision among group G_2 in list $\mathcal{L}_2^{f_i}$.
- *Case 6*: There exists collision among group G_2 in list $\mathcal{L}_2^{h_i}$.
- *Case 7*: $F - X = 0$ in $\mathbb{Z}_p$.
- *Case 8*: $F_{1,\sigma}(x_1, x_2, \{u\}, \{r\}, \{y\}) = 0$ in $\mathbb{Z}_p$.

`Game` $\mathcal{G}^*$ *`vs.`* `actual EUF-CMLA game`: The success probability of $\mathcal{A}^*$ in the actual EUF-CMLA game is bounded by its success probability in the above game $\mathcal{G}^*$ with a negligible probability gap. The reasons are as follows:

- Case 1 means that $\mathcal{A}^*$ can provoke collisions among group elements of G_1, i.e., $F_{1,i} \neq F_{1,j}$ but $g^{F_{1,i}(x_1,x_2,\{u\},\{r\},\{y\})} = g^{F_{1,j}(x_1,x_2,\{u\},\{r\},\{y\})}$ in the actual EUF-CMLA game with a fixed generator g. Cases 2-6 have the similar meanings. As long as these six cases do not occur, then the view of $\mathcal{A}^*$ is identical that in the game $\mathcal{G}^*$ and in the actual interaction. Therefore, if $\mathcal{A}^*$ cannot provoke collisions in these lists, then its adaptive strategies are no more powerful than non-adaptive ones.
- Case 7 means that $\mathcal{A}^*$ computes or guesses the secret key X from the leakage function f_i or h_i.
- Case 8 means that $(\xi_{1,\sigma_1}, \xi_{1,\sigma_2})$ is a valid forgery on a new message m^*.

`Advantage`: We now analyze the advantage of $\mathcal{A}^*$ in the game $\mathcal{G}^*$ one by one correspond to the above cases.

- $\mathbf{Pr}[Cases 1-6]$: i.e., the probability of $\mathcal{A}^*$ can provoke collisions in the lists $L_1, L_2, L_1^{f_i}, L_1^{h_i}, L_2^{f_i}$ or $L_2^{h_i}$. We first analyze the case 1, since $F_{1,i} \neq F_{1,j} \Leftrightarrow F' = F_{1,i} - F_{1,j} \neq 0$, and the degree of the polynomials in the list $\mathcal{L}_1$ at most *two*. Note that in the presence of leakage, the polynomials are evaluated with values chosen from independent distributions with min-entropy $\log p - 2\lambda$[1].

[1] Since $\mathcal{A}^*$ can obtain at most 2λ bits of leakage about $l_i (i \in [q_s])$ and at most λ bits of $r_i (i \in [q_s])$. According to the result of the Lemma 1, l_i and r_i have min-entropy at least $\log p - 2\lambda$ in the view of $\mathcal{A}^*$.

According to the Schwartz-Zippel Lemma (with $\Delta = 2\lambda$), $\mathbf{Pr}[F'(x_1, x_2, \{u\}, \{r\}, \{y\}) = 0] \leq \frac{2}{p}2^{2\lambda}$. The list $\mathcal{L}_1$ has τ_1 entries, so there exists at most $\binom{\tau_1}{2}$ distinct pairs $(F_{1,i}, F_{1,j})$, the probability of the case 1 occurs is at most $\binom{\tau_1}{2} \cdot \frac{2}{p}2^{2\lambda}$. Similarly, the probability of the case 2 occurs is at most $\binom{\tau_2}{2} \cdot \frac{4}{p}2^{2\lambda}$. To the cases 3 and 4, the total length of the lists $\mathcal{L}_1^{f_i}, \mathcal{L}_1^{h_i}$ is at most $O(q_o + q_s) = O(q)$, and thus there can be at most $O(q^2)$ pairs of distinct polynomials of degree at most *two* evaluated to the same value. According to the Schwartz-Zippel Lemma, $\mathbf{Pr}[Cases3, 4] \leq O(\frac{q^2}{p}2^{2\lambda})$. Similarly, $\mathbf{Pr}[Cases5, 6] \leq O(\frac{q^2}{p}2^{2\lambda})$ too.

- $\mathbf{Pr}[Case7]$: i.e., $\mathbf{Pr}[\mathbf{E}]$. We first show that $F - X$ is a non-zero polynomial. If $c_{1,i} = c_{2,i} = 0$ in equation (19), then the lists $L_1^{f_i}$ do not contain the polynomial X. If $c_{1,i} \neq 0$ or $c_{2,i} \neq 0$, then the polynomials in (19) will contain polynomial L_i or L_{i-1}, or both. Hence the polynomial X cannot appear in any of the lists $\mathcal{L}_1^{f_i}$. Similarly, the lists $\mathcal{L}_1^{h_i}$ also cannot contain X. Therefore, $F - X$ is a non-zero polynomial of degree at most *two*. Hence $\mathbf{Pr}[Case7] \leq \frac{2}{p}2^{2\lambda}$.
- $\mathbf{Pr}[Case8]$: i.e., $\mathbf{Pr}[\mathbf{F}|\overline{\mathbf{E}}]$. If the event $\mathbf{E}$ has not occurred which means that $\mathcal{A}^*$ has not compute or guess the secret key X, the only meaningful leakage for $\mathcal{A}^*$ is that of $r_i (i = 1, \ldots, q_s)$. Since at most λ bits of r_i will be leaked by f_i, from the view point of $\mathcal{A}^*$ the values r_i have min-entropy at least $\log p - \lambda$. According to the Schwartz-Zippel lemma (if $F_{1,\sigma}$ is non-zero which proved in Lemma 8 below), $\mathbf{Pr}[\mathbf{F}|\overline{\mathbf{E}}] \leq \frac{3}{p}2^{\lambda}$.

Therefore, the advantage of $\mathcal{A}^*$ is $\mathbf{Adv}^{euf-cmla}_{\Sigma^*_{\mathrm{W}},\mathcal{A}^*} \leq O(\frac{q^2}{p}2^{2\lambda})$. From the point of the length of the leakage, it can be leaked $\frac{1-o(1)}{2}\log p$ bits per each signing invocation.

Lemma 8. *$F_{1,\sigma}$ is a non-zero polynomial in $\mathbb{Z}_p[X_1, X_2, \{U\}, \{R\}, \{Y\}]$.*

Proof. From the design of the group oracles and the initial elements of the list $\mathcal{L}_1$, we can see that any polynomial in $\mathcal{L}_1$ is computed by either adding or subtracting two polynomials previously existing in the list. Therefore, w.l.o.g., we can write the forgery $(F_{1,\sigma_1}, F_{1,\sigma_2})$ as follows.

$$F_{1,\sigma_1} = c_1 + c_2X_1 + c_3X_2 + \sum_{i=1}^{n} c_{4,i}U_i + \sum_{i=1}^{q_s} c_{5,i}R_i + \sum_{i=1}^{q_{g_1}} c_{6,i}Y_i + \sum_{i=1}^{q_s} c_{7,i}(X_1X_2 + (U_0 + \sum_{j\in\mathcal{M}_i} U_j)R_i) \quad (22)$$

$$F_{1,\sigma_2} = d_1 + d_2X_1 + d_3X_2 + \sum_{i=1}^{n} d_{4,i}U_i + \sum_{i=1}^{q_s} d_{5,i}R_i + \sum_{i=1}^{q_{g_1}} d_{6,i}Y_i + \sum_{i=1}^{q_s} d_{7,i}(X_1X_2 + (U_0 + \sum_{j\in\mathcal{M}_i} U_j)R_i) \quad (23)$$

where $c_j, d_j (j = 1, 2, 3), c_{j,i}, d_{j,i} (j = 4, 5, 6, 7) \in \mathbb{Z}_p$ are chosen by $\mathcal{A}^*$. We divide them into two cases:

- *Case 1:* $c_{7,i} = d_{7,i} = 0$, for $\forall i \in [q_s]$.
  In this case, both F_{1,σ_1} and F_{1,σ_2} do not contain the term X_1X_2. Hence the polynomial $F_{1,\sigma_2}(U_0 + \sum_{j\in\mathcal{M}^*} U_j) - F_{1,\sigma_1}$ in (21) also does not contain X_1X_2. Therefore, in the polynomial $F_{1,\sigma}$, the coefficient of the term X_1X_2 is non-zero, and thus $F_{1,\sigma}$ is non-zero.
- *Case 2:* $c_{7,k} \neq 0$ or $d_{7,k} \neq 0$, for $\exists k \in [q_s]$.
  - If $d_{7,k} \neq 0$, then the coefficient of the term $U_0^2R_k$ is non-zero, and thus $F_{1,\sigma}$ is non-zero.
  - If $c_{7,k} \neq 0$, since $m^* \neq m_k$, and thus it exists at least one bit is unequal, w.l.o.g., we assume that $0 = m_j^* \neq m_{k,j} = 1$. Hence the coefficient of the term U_jR_k is non-zero, and thus $F_{1,\sigma}$ is non-zero.

Therefore, the polynomial $F_{1,\sigma}$ is non-zero. This completes the proof of Theorem 5. □