Strongly Unforgeable Signatures Resilient to Polynomially Hard-to-Invert Leakage under Standard Assumptions[*]

Masahito Ishizaka and Kanta Matsuura

Institute of Industrial Science, The University of Tokyo, Tokyo, Japan.
{ishimasa, kanta}@iis.u-tokyo.ac.jp

Abstract. A signature scheme is said to be weakly unforgeable, if it is hard to forge a signature on a message not signed before. A signature scheme is said to be strongly unforgeable, if it is hard to forge a signature on any message. In some applications, the weak unforgeability is not enough and the strong unforgeability is required, e.g., the Canetti, Halevi and Katz transformation.
Leakage-resilience is a property which guarantees that even if secret information such as the secret-key is partially leaked, the security is maintained. Some security models with leakage-resilience have been proposed. The hard-to-invert leakage model, a.k.a. auxiliary (input) leakage model, proposed by Dodis et al. at STOC'09 is especially meaningful one, since the leakage caused by a function which information-theoretically reveals the secret-key, e.g., one-way permutation, is considered.
In this work, we propose a generic construction of digital signature strongly unforgeable and resilient to polynomially hard-to-invert leakage which can be instantiated under standard assumptions such as the decisional linear assumption. We emphasize that our instantiated signature is not only the first one resilient to polynomially hard-to-invert leakage under standard assumptions, but also the first one which is strongly unforgeable and has hard-to-invert leakage-resilience.

Keywords: Digital signature, Strong existential unforgeability, Leakage-resilience, Hard-to-invert leakage, Auxiliary(-input) leakage.

1 Introduction

Strongly Unforgeable Signature. We say that a signature scheme is weakly (existentially) unforgeable if it is hard to forge a signature on a message not signed before. We say that a signature scheme is strongly (existentially) unforgeable if it is hard to forge a signature on any message which can be a message signed before. Since most of the signature schemes generate a signature randomly, there is a gap between the weak unforgeability and strong unforgeability. Moreover, in some applications, strongly unforgeable signature scheme is required, e.g., Canetti-Halevi-Katz transformation [13].

[*]This is the full version of a short paper appeared in the proceedings for the 21st Information Security Conference (ISC2018).

Leakage-Resilient Cryptography. Leakage-resilience is a property which guarantees that even if secret information such as the secret key is partially leaked, the security is maintained. Any scheme whose security has been proven only in a security model without leakage-resilience is not guaranteed to be secure when some information about the secret information such as the secret-key are leaked. There exist some side-channel attacks which are real threats to us, e.g., cold-boot attack [21], so leakage-resilient cryptographic schemes are practically desirable.

In the security model considering leakage-resilience, a side-channel attack caused by an adversary is modeled as a polynomial time computable function $f : \{0,1\}^{|Secret|} \to \{0,1\}^{*}$ [1]. The adversary is allowed to choose an arbitrary leakage function f, query it to the leakage oracle, then learn $f(Secret)$. If we allow the adversary to choose the identity map as f, the adversary acquires the secret key entirely and is able to break the security model with no failures. Hence, we have to impose a restriction on f. Several security models in which different restrictions are imposed on f have been proposed.

In bounded leakage (BL) model [1], the output bit-length of f is restricted. More concretely, only f satisfying $f : \{0,1\}^{|Secret|} \to \{0,1\}^{l(k)}$ such that $l(k) < k$ can be chosen[2]. To make the output bit-length of f unbounded, noisy leakage (NL) model [28] was invented. In NL model, only $f : \{0,1\}^{|Secret|} \to \{0,1\}^{*}$ such that, when we observe $f(Secret)$, the minimum entropy of the secret key sk drops by at most $l(k) < k$ can be chosen. Any function which information-theoretically reveals the secret key sk is excluded in each one of the two models. Thus, for instance, one-way permutation cannot be chosen in each one of the models. To remove such a restriction, hard-to-invert leakage (HL) model, a.k.a. auxiliary (input) leakage (AL) model, was invented [16]. In HL model, the function f must be a hard-to-invert function. More concretely, only f such that, given $f(Secret)$, no PPT algorithm can compute sk with a probability larger than $\mu(k)$ can be chosen, where $\mu(\cdot)$ is a negligible function such that $\mu(k) > 2^{-k}$. The larger $\mu(k)$ is, the larger the function class of f is. HL model is a generalization of BL and NL model, thus has a larger function class. Moreover, HL model is useful in the context of the composition. There may be the case when we want to use the same pair of public key and secret key of a hard-to-invert leakage-resilient cryptographic scheme for multiple schemes. Their composition remains secure as long as each one of the other schemes has been proven to be secure in the standard (non-leakage-resilient) security model [32, 14].

Large number of cryptographic schemes with leakage-resilience have been proposed. For instance, public-key encryption [1, 15, 5, 14], identity-based encrypiton [12, 26, 24, 32], attribute-based encrypiton [26, 35, 34, 36], identification [2, 15], and authenticated key agreement [2, 15], have been proposed.

Related Works. Katz and Vaikuntanathan [25] defined that fully leakage-resilient (FLR) signature is a signature resilient to not only the direct leakage from the secret-key, but also the leakage from the randomness used to generate the secret-key and signa-

[1] $Secret$ denotes the secret information. $|Secret|$ denotes the bit-length of $Secret$.

[2] k denotes the minimum entropy of the secret key sk. If the secret-key is generated uniformly at random, k is equivalent to the bit-length of the secret-key $|sk|$.

tures generated by the signing oracle. FLR or non-FLR signature schemes secure in the bounded-leakage model have been proposed in [2, 25, 27, 10].

The concept of the hard-to-invert leakage-resilience was presented by Dodis et al. [16]. They proposed symmetric-key encryption schemes which is IND-CPA or IND-CCA secure and resilient to exponentially hard-to-invert leakage. In a subsequent work, Dodis et al. [14] started a research on public-key encryption with hard-to-invert leakage-resilience. They defined two leakage-function classes. The class $\mathcal{H}_{\mathsf{ow}}(\xi(\lambda))$ (resp. $\mathcal{H}_{\mathsf{pkow}}(\xi(\lambda))$) consists of every polynomial-time computable function $f : \{0,1\}^{|pk|+|sk|} \rightarrow \{0,1\}^*$ such that any PPT algorithm $\mathcal{A}$ which is given $f(pk, sk)$ (resp. $(pk, f(pk, sk))$) as input is able to guess sk correctly only with a probability smaller than $\xi(\lambda)$, where $\xi(\lambda) > 2^{-k}$ is a negligible function and (pk, sk) is a randomly generated key-pair. They proved that the BHHO encryption scheme [7] and a slightly modified version of the GPV encryption scheme [18] are IND-CPA secure in HL model w.r.t. the function class $\mathcal{H}_{\mathsf{ow}}(1/\mu_1(\lambda))$, where $\mu_1(\lambda)$ is a sub-exponential function. They also mentioned in Subsect. 1.2 of [14] that a PKE scheme which is IND-CPA secure in HL model w.r.t. $\mathcal{H}_{\mathsf{pkow}}(1/\mu_2(\lambda))$, where $\mu_2(\lambda)$ is a polynomial function, is given in its full paper.

Faust et al. presented the first research result on digital signature with hard-to-invert leakage-resilience [17]. To construct a signature scheme secure in HL model, there is an obstacle whom we have to overcome. It is how to prevent from an adversary choosing an algorithm generating a valid signature on a message as a leakage-function, then output the pair of signature and message. Faust et al. proposed a signature scheme which is wEUF-CMA (weakly existentially unforgeability under adaptively chosen messages attack) secure in HL model w.r.t. the function class $\mathcal{H}_{\mathsf{pkow}}(1/\mu_3(\lambda))$, where $\mu_3(\lambda)$ is an exponential function. Their solution to overcome the obstacle explained earlier is to include a ciphertext of the secret-key sk in a signature. Specifically, their signature scheme adopts the labeled public-key encryption (LPKE) as a building block, and includes a ciphertext of the secret-key in a signature. Moreover, for their signature scheme, the hardness parameter $1/\mu_3(\lambda)$ in the leakage function class is set as $1/\mu_3(\lambda) << 2^{-l_{dk}}$, where $l_{dk} \in \mathbb{N}$ denotes the bit-length of the decryption-key dk of the LPKE scheme. This solution effectively works. The reason is as follows. Since any PPT algorithm is able to guess the decryption-key dk correctly with probability $2^{-l_{dk}}$, any PPT inverter in the definition of the function class $\mathcal{H}_{\mathsf{pkow}}(1/\mu_3(\lambda))$ which is given a signature including a ciphertext C of the secret-key sk is able to guess sk correctly with probability $2^{-l_{dk}} >> 1/\mu_3(\lambda)$ by guessing the decryption-key dk, then decrypting the ciphertext C with the guessed dk. Hence, the signing algorithm is excluded from the class $\mathcal{H}_{\mathsf{pkow}}(1/\mu_3(\lambda))$. By the way, they showed that their signature scheme can be instantiated under standard assumptions such as the DLIN assumption [4].

Independently of Faust et al. [17], Yuen et al. [33] also presented a research result on signature secure in HL model. To overcome the obstacle to construct a signature with hard-to-invert leakage resilience, Yuen et al. proposed an original security model, which is named *selective auxiliary input model*. In the security model, the adversary is allowed to choose as the leakage-functions only functions which are independent of the public-key. They proposed a signature scheme secure in the security model. Their signature scheme is FLR and resilient to polynomially hard-to-invert leakage. Here, their definition of leakage function f being resilient to polynomially hard-to-invert leakage is as follows: any PPT algorithm which is given $(pk, S, f(state))$ is able to guess *sk* correctly only with a negligible probability, where (pk, sk) is a randomly generated key-pair, S is a set of randomly generated signatures on the messages queried to the signing oracle, and *state* is a set of randomnesses used to generate *sk* and the signatures S. Their definition of leakage-function is undesirable since it depends on the signatures generated on the signing oracle.

Subsequently, Wang et al. [30] proposed a signature scheme secure in the selective auxiliary input model. Their signature scheme is FLR and resilient to polynomially hard-to-invert leakage. Their definition for a function to be resilient to polynomially hard-to-invert leakage is not the same as the one by Yuen et al. [33]. It is improved as follows: any PPT algorithm which is given $f(sk)$ is able to identify *sk* only with a negligible probability. However, their scheme needs differing input obfuscator (diO), indistinguishable obfuscator (iO), and point-function obfuscator with auxiliary input (AIPO), each one of which has been constructed only under strong assumptions.

Note that each one of the signature schemes with auxiliary leakage resilience by Faust et al., Yuen et al., and Wang et al., is not strongly existentially unforgeable, but weakly existentially unforgeable.

Boneh et al. [9] proposed a method to transform a weakly unforgeable signature scheme into a strongly unforgeable one. However, their transformation can be applied to *partitioned* signatures only. In a subsequent work, Steinfeld et al. [29] proposed a method to transform *any* weakly unforgeable signature into a strongly unforgeable one. Note that each transformation by Boneh et al. and Steinfeld et al. has a common property such that each one of the public-key, secret-key and signature of the strongly unforgeable signature scheme becomes each one of the public-key, secret-key and signature of the weakly unforgeable signature whom some new elements are added to. Huang et al. [22] proposed a transformation which no new elements are added to the public-key, secret-key and signature.

Wang et al. [31] modified the transformation by Steinfeld et al. [29] to get a transformation from a signature weakly existentially unforgeable and FLR in the bounded leakage model to a strongly unforgeable one. The transformation by Steinfeld et al. utilizes two chameleon hash functions (with no leakage-resilience). In the transformation by Wang et al., one of the chameleon hash functions is assumed to satisfy a property such that any PPT algorithm cannot find a strong collision even if the algorithm is given a length-bounded information about the secret-key.

The transformation by Wang et al. needs to add some new elements to both the key-pair and signature. Huang et al. [20] modifies the transformation by Huang et al. [22] to get a method to transform a signature weakly existentially unforgeable and FLR in the BL model into a strongly unforgeable one which no elements are added to the signature[3].

Our Results. We propose a generic construction of signature scheme strongly which is unforgeable and resilient to polynomially hard-to-invert leakage and can be instantiated under standard assumptions. Specifically, we give an example of its instantiation under

[3]By the transformation in [20], some new elements are added to the public-key and secret-key.

the decisional linear (DLIN) assumption [4]. Our security model is not the selective auxiliary leakage model [33], so the leakage-function can be dependent on the public-key.

Our result is a desirable one because of the following two independent points. Firstly, our signature instantiation is the first one which is resilient to polynomially hard-to-invert leakage under standard assumptions. Secondly, our signature instantiation is the first one which is strongly unforgeable and has hard-to-invert leakage-resilience.

Our Approach. Our result is obtained by modifying the one by Faust et al. [17]. Before explaining how the modification is done, we explain the result by Faust et al. in detail.

Faust et al. proposed a generic construction of a signature scheme secure in the wEUF-CMA security model w.r.t. the function class $\mathcal{H}_{\mathtt{pkow}}(\xi(\lambda))$. It consists of three building blocks. They are second pre-image resistant hash function (SPRHF), labeled PKE (LPKE) whose decryption-key dk has bit-size $l_{dk} \in \mathbb{N}$, and non-interactive zero-knowledge proof (NIZK) whose trapdoor td has bit-size $l_{td} \in \mathbb{N}$. The hardness parameter of the leakage function class is set as $\xi(\lambda) = 2^{-(\lambda+l_{dk}+l_{td})}$. A signature σ on a message m consists of an LPKE ciphertext c and an NIZK proof π. Concretely, the ciphertext c is an LPKE ciphertext encrypting the secret-key sk under the label m, and the proof π is an NIZK proof which proves that there exists a secret-key sk' such that the ciphertext c is a ciphertext of sk' on the label m and the hashed value of sk' is equivalent to the hashed value of the real secret-key sk which is included in the public-key pk.

Intuitively speaking, the security proof for the signature by Faust et al. is done as follows. By modifying the initial security game several times, we get the final game $\mathtt{Game}_{final}$. In $\mathtt{Game}_{final}$, for a signature $\sigma = (c, \pi)$ on the signing oracle, the ciphertext c is generated by encrypting $0^{|sk|}$ instead of sk, and the proof π is generated by using the trapdoor td instead of sk. In addition, the adversary is considered to win the game, if he successfully outputs a signature $\sigma^* = (c^*, \pi^*)$ and a message m^* such that c^* is a valid ciphertext of sk^* on label m^*, and π^* is a valid proof. We prove that every PPT $\mathcal{A}$ wins the game only with a negligible probability by a reduction to the hard-to-invert property of the leakage-function $f \in \mathcal{H}_{\mathtt{pkow}}(2^{-(\lambda+l_{dk}+l_{td})})$. In the reduction, a simulator $\mathcal{S}$ needs both td and dk to simulate $\mathtt{Game}_{final}$ and decrypt the ciphertext c^*. However, by the definition of the leakage function class $\mathcal{H}_{\mathtt{pkow}}(\cdot)$, $\mathcal{S}$ is given neither td nor dk, so $\mathcal{S}$ has to guess them, and the guess succeeds with probability $2^{-(l_{dk}+l_{td})}$. By the above reason, the hardness parameter for Faust et al.'s signature scheme becomes $2^{-(\lambda+l_{dk}+l_{td})}$.

The above is the result by Faust et al. We modify the result with three steps.

In the first step, we generalize the second pre-image resistance (SPR) property of the SPRHF, which is one of the building blocks. Intuitively, the SPR property is a property such that no PPTA given a key-pair (pk, sk) is able to find a secret-key sk^* which is not sk, but has a hashed value equivalent to the hashed value of sk with a non-negligible probability. We generalize it to a property such that no PPTA given (pk, sk) is able to find a secret-key sk^* such that a relation holds between sk^* and sk and another relation also holds between sk^* and pk with a non-negligible probability.

The second step is to modify the definition of the leakage-function class $\mathcal{H}_{\mathtt{pkow}}(\cdot)$. In the modified definition of the function class, the PPT algorithm (or inverter) $\mathcal{A}$ is given not only the public-key of the key-pair (pk, sk), but also some variables which are generated during generation of the key-pair and are not directly included in either pk or *sk*. Specifically, for our signature scheme, the variables are the decryption-key *dk* and the trapdoor *td*. If the definition of the leakage-function class is modified to such one, the simulator in the proof for $\texttt{Game}_{final}$ is not forced to guess *dk* and *td* with probability $2^{-(l_{dk}+l_{td})}$, so the polynomially hard-to-invert leakage-resilience security is achieved. Instantiating the generic construction of the signature scheme, we can concretely generate the first signature scheme (weakly unforgeable and) resilient to polynomially hard-to-invert leakage under standard assumptions such as the DLIN assumption.

In the third step, we apply a methodology which is invented by modifying the one by Wang et al. [31] to the weakly unforgeable signature scheme in the second step, then get a strongly unforgeable one. Note that unlike Wang et al., we do not propose a generic transformation from a weakly unforgeable and resilient to hard-to-invert leakage to a strongly unforgeable one. In the transformation by Wang et al., a chameleon hash function with strong collision-resistance in the bounded leakage model (BLR-CHF) was used. We use a CHF with strong collision-resistance in HL model (HLR-CHF). Moreover, the secret-key of the strongly unforgeable signature scheme obtained by the transformation by Wang et al. includes not only the *original* secret-key, i.e., the secret-key of the weakly unforgeable signature, but also the secret-key of the BLR-CHF. However, the secret-key of our strongly unforgeable signature includes only the secret-key of the HLR-CHF. By instantiating the signature scheme, we obtain the first concrete construction of digital signature strongly unforgeable and resilient to polynomially hard-to-invert leakage under standard assumptions such as the DLIN assumption.

Paper Organization. This paper is organized as follows. In Sect. 2, we give basic notations and the syntax and the definition of security or property of labeled public-key encryption, non-interactive zero-knowledge proof and chameleon hash function. In Sect. 3, we give the syntax and the definition of strong unforgeability in HL model of digital signature. In Sect. 4, our generic construction of signature and its security proof are given. In Sect. 5, we show that the generic construction of signature in Sect. 4 can be instantiated under the DLIN assumption.

2 Preliminaries

Notation. For $a, b \in \mathbb{N}$, $[a, b]$ denotes $\{x \in \mathbb{N} \mid a \leq x \leq b\}$. For $\lambda \in \mathbb{N}$, 1^λ denotes a security parameter. We say that a function $h : \mathbb{N} \to \mathbb{R}$ is negligible if for every $c \in \mathbb{N}$, there exists $x_0 \in \mathbb{N}$ such that $h(x) \leq x^{-c}$ for every $x \geq x_0$. $\mathcal{G}$ is a function which takes 1^λ as input, and randomly outputs $(p, \mathbb{G}, g)$, where p is a prime number whose bit-size is λ, $\mathbb{G}$ is a multiplicative cyclic group whose order is p, and g is a generator of $\mathbb{G}$. PPTA means probabilistic polynomial time algorithm.

2.1 Hardness Assumptions

Discrete Logarithm (DL) Assumption. For $\lambda \in \mathbb{N}$, let $(p, \mathbb{G}, g) \leftarrow \mathcal{G}(1^\lambda)$. DL assumption holds, if for every PPTA $\mathcal{A}$, the probability $\Pr[x \leftarrow \mathcal{A}(p, \mathbb{G}, g, g^x) \mid x \overset{\text{U}}{\leftarrow} \mathbb{Z}_p]$ is negligible in λ.

Decisional Linear (DLIN) Assumption [4]. For $\lambda \in \mathbb{N}$, let $(p, \mathbb{G}, g) \leftarrow \mathcal{G}(1^\lambda)$. DLIN assumption holds, if for every PPTA $\mathcal{A}$,

$$\Big|\Pr[1 \leftarrow \mathcal{A}(p, \mathbb{G}, g_1, g_2, g_3, g_1^{r_1}, g_2^{r_2}, g_3^{r_1+r_2}) \mid g_1, g_2, g_3 \overset{\mathrm{U}}{\leftarrow} \mathbb{G}, r_1, r_2 \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p]$$
$$- \Pr[1 \leftarrow \mathcal{A}(p, \mathbb{G}, g_1, g_2, g_3, g_1^{r_1}, g_2^{r_2}, g_3^{u}) \mid g_1, g_2, g_3 \overset{\mathrm{U}}{\leftarrow} \mathbb{G}, r_1, r_2, u \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p]\Big|$$

is negligible in λ.

2.2 Labeled Public Key Encryption

Syntax. Labeled public key encryption (LPKE) consists of three polynomial time algorithms {Gen, Enc, Dec}. Gen and Enc are probabilistic. Dec is deterministic.

$\mathsf{Gen}(1^\lambda) \to (ek, dk)$. The key generation algorithm takes 1^λ as input, and outputs an encryption key ek, and a decryption key dk. Plaintext space $\mathcal{M}$, ciphetext space $\mathcal{C}$, and label space $\mathcal{L}$ are uniquely determined by ek.

$\mathsf{Enc}(ek, m, L) \to C$. The encryption algorithm takes the encryption key ek, a plaintext $M \in \mathcal{M}$, and a label $L \in \mathcal{L}$ as inputs, and outpus a ciphertext C.

$\mathsf{Dec}(dk, C, L) \to M\ /\ \bot$. The decryption algorithm[4] takes the decryption key dk, a ciphetext $C \in \mathcal{C}$, and a label $L \in \mathcal{L}$ as inputs, and outputs a plaintext M or $\bot$.

A LPKE scheme must be correct. LPKE scheme Σ_{LPKE} = {Gen, Enc, Dec} is correct, if for every $\lambda \in \mathbb{N}$, every $(ek, dk) \leftarrow \mathsf{Gen}(1^\lambda)$, every $M \in \mathcal{M}$, every $L \in \mathcal{L}$, and every $C \leftarrow \mathsf{Enc}(ek, M, L)$, it holds that $M \leftarrow \mathsf{Dec}(dk, C, L)$.

2.2.1 Ciphertext Indistinguishability

To define weak ciphertext indistinguishability against adaptively chosen label and ciphertexts attacks (IND-wLCCA) for a LPKE scheme Σ_{LPKE} = {Gen, Enc, Dec}, we use the following game which is played between an adversary $\mathcal{A}$ and challenger $\mathcal{CH}$.

Key-Generation. $\mathcal{CH}$ runs $(ek, dk) \leftarrow \mathsf{Gen}(1^\lambda)$, and sends ek to $\mathcal{A}$.

Query. $\mathcal{A}$ is allowed to use the decryption oracle **Dec** adaptively.

- **Dec(C, L):** $\mathcal{A}$ queries a ciphertext $C \in \mathcal{C}$ and a label $L \in \mathcal{L}$. $\mathcal{CH}$ returns $M\ /\ \bot \leftarrow \mathsf{Dec}(dk, C, L)$.

Challenge(M_0, M_1, L^*). $\mathcal{A}$ sends two plaintexts $M_0, M_1 \in \mathcal{M}$, and a label $L^* \in \mathcal{L}$. $\mathcal{CH}$ sets $b \overset{\mathrm{U}}{\leftarrow} \{0, 1\}$, then returns $C^* \leftarrow \mathsf{Enc}(ek, M_b, L^*)$.

Query 2. $\mathcal{A}$ is allowed to use the decryption oracle **Dec** adaptively.

- **Dec(C, L):** $\mathcal{A}$ queries a ciphertext $C \in \mathcal{C}$ and a label $L \in \mathcal{L}$ such that $L \neq L^*$. $\mathcal{CH}$ returns $M\ /\ \bot \leftarrow \mathsf{Dec}(dk, C, L)$.

Guess(b'). $\mathcal{A}$ sends $b' \in \{0, 1\}$ to $\mathcal{CH}$.

Definition 1. *LPKE scheme Σ_{LPKE} is IND-wLCCA secure if for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{IND\text{-}wLCCA}}_{\mathcal{A},\Sigma_{LPKE}}(\lambda) = |2 \cdot \Pr[b' = b] - 1|$ is negligible.*

[4]Although Dec needs the encryption-key ek as an input since ek includes information such as the prime p , the group $\mathbb{G}$, and etc., we often omit ek as the input.

2.3 Non-Interactive Zero-Knowledge Proof

Syntax. Non-interactive zero-knowledge proof (NIZK) Π_{NIZK} for a language L consists of three polynomial time algorithms $\{\mathsf{Gen}, \mathsf{Pro}, \mathsf{Ver}\}$. Each one of Gen and Pro is probabilistic. Ver is deterministic. R_L denotes the witness relation.

$\mathsf{Gen}(1^\lambda) \rightarrow crs$**.** The key-generation algorithm takes 1^λ as an input, and outputs a common reference string (CRS) crs.

$\mathsf{Pro}(crs, x, w) \rightarrow \pi$**.** The proof-generation algorithm takes the CRS crs, a statement x, and a witness w as inputs, and outputs a proof π.

$\mathsf{Ver}(crs, x, \pi) \rightarrow 1 \,/\, 0$**.** The proof-verification algorithm takes the CRS crs, a statement x, and a proof π as inputs, and outputs 1 or 0.

A NIZK scheme must be correct. A NIZK scheme $\Sigma_{\text{NIZK}} = \{\mathsf{Gen}, \mathsf{Pro}, \mathsf{Ver}\}$ is correct if for every $\lambda \in \mathbb{N}$, every $crs \leftarrow \mathsf{Gen}(1^\lambda)$, every (x, w) such that $(x, w) \in R_L$, and every $\pi \leftarrow \mathsf{Pro}(crs, x, w)$, it holds that $1 \leftarrow \mathsf{Ver}(crs, x, \pi)$.

We give the definitions of soundness and zero-knowledge for a NIZK scheme.

Definition 2. *A NIZK scheme $\Sigma_{\text{NIZK}} = \{\mathsf{Gen}, \mathsf{Pro}, \mathsf{Ver}\}$ is sound if for every $\lambda \in \mathbb{N}$, every crs $\leftarrow \mathsf{Gen}(1^\lambda)$, and every PPT $\mathcal{A}$,*

$$\Pr\left[\mathcal{A}(crs) \rightarrow (x, \pi) \ s.t.\ [\mathsf{Ver}(crs, x, \pi) \rightarrow 1] \wedge [x \notin L]\right]$$

is negligible.

Definition 3. *A NIZK scheme $\Sigma_{\text{NIZK}} = \{\mathsf{Gen}, \mathsf{Pro}, \mathsf{Ver}\}$ is zero-knowledge if for every $\lambda \in \mathbb{N}$ and every PPT $\mathcal{A}$, there exists a PPT $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that*

$$\Big|\Pr\left[\mathcal{A}^{O_0^{crs}(x,w)}(crs) \rightarrow 1 \mid \mathsf{Gen}(1^\lambda) \rightarrow crs\right] - \\ \Pr\left[\mathcal{A}^{O_1^{crs,td}(x,w)}(crs) \rightarrow 1 \mid S_1(1^\lambda) \rightarrow (crs, td)\right]\Big|$$

is negligible, where $O_0^{crs}(x, w)$ returns $\mathsf{Pro}(crs, x, w)$ (resp. $\perp$), if $(x, w) \in R_L$ (resp. $(x, w) \notin R_L$), and $O_1^{crs,td}(x, w)$ returns $\mathcal{S}_2(crs, x, td)$ (resp. $\perp$), if $(x, w) \in R_L$ (resp. $(x, w) \notin R_L$).

2.4 Chameleon Hash Function

Syntax. A chameleon hash function (CHF) scheme consists of the polynomial time algorithms $\{\mathsf{Gen}, \mathsf{Eval}, \mathsf{TC}, \mathsf{SKVer}, \mathsf{SKVer2}\}$. Gen and TC are probabilistic, and Eval, SKVer and $\mathsf{SKVer2}$ are deterministic.

$\mathsf{Gen}(1^\lambda) \rightarrow (pk, sk)$**.** The key-generation algorithm takes a security parameter 1^λ, where $\lambda \in \mathbb{N}$, as an input, and outputs a public-key pk and a secret-key (or trapdoor) sk. The message space $\mathcal{M}$, randomness space $\mathcal{R}$ and hashed value space $\mathcal{H}$ are uniquely determined by pk.

$\mathsf{Eval}(pk, m; r) \rightarrow h$**.** The evaluation algorithm takes the public-key pk and a message $m \in \mathcal{M}$ as inputs, and outputs the hashed value $h \in \mathcal{H}$ which was calculated under a randomness $r \in \mathcal{R}$.

$\mathsf{TC}(pk, sk, (m_1, r_1), m_2) \to r_2$. The trapdoor collision finder algorithm takes the public-key pk, the secret-key sk, a pair of a message and randomness $(m_1, r_1) \in \mathcal{M} \times \mathcal{R}$, and a message $m_2 \in \mathcal{M}$ as inputs, and outputs a randomness $r_2 \in \mathcal{R}$.

$\mathsf{SKVer}(pk, sk') \to 1 \,/\, 0$. The first secret-key-verification algorithm takes the public-key pk and a secret-key sk' as inputs, and outputs 1 or 0.

$\mathsf{SKVer2}(pk, sk', sk^\dagger) \to 1 \,/\, 0$. The second secret-key-verification algorithm takes the public-key pk, a secret-key sk', and a secret-key $sk^\dagger$ as inputs, and outputs 1 or 0. Even if the two secret-keys are inputted in the reversed order, the output is required to be equivalent. Thus, for any $\lambda \in \mathbb{N}$, any $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, and any two valid secret-keys sk' and $sk^\dagger$, it holds that $\mathsf{SKVer2}(pk, sk', sk^\dagger) = \mathsf{SKVer2}(pk, sk^\dagger, sk')$.

A CHF scheme must be correct. A CHF scheme $\Sigma_{\mathrm{CHF}} = \{\mathsf{Gen}, \mathsf{Eval}, \mathsf{TC}, \mathsf{SKVer}, \mathsf{SKVer2}\}$ is correct, if for every $\lambda \in \mathbb{N}$, every $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, every $m \in \mathcal{M}$, every $m' \in \mathcal{M}$, every $r \in \mathcal{R}$, and every $r' := \mathsf{TC}(pk, sk, (m, r), m')$, it holds that $[\mathsf{Eval}(pk, m; r) = \mathsf{Eval}(pk, m'; r')] \wedge [1 \leftarrow \mathsf{SKVer}(pk, sk)] \wedge [1 \leftarrow \mathsf{SKVer2}(pk, sk, sk)]$.

We give the definitions of two standard properties for the CHF scheme. They are strong collision-resistance and random trapdoor collision.

Definition 4. *A CHF scheme $\Sigma_{\mathrm{CHF}} = \{\mathsf{Gen}, \mathsf{Eval}, \mathsf{TC}, \mathsf{SKVer}, \mathsf{SKVer2}\}$ is strongly collision-resistant, if for every $\lambda \in \mathbb{N}$ and every PPT $\mathcal{A}$, it holds that*

$$\begin{aligned}\Pr[\mathcal{A}(pk) \to ((m_1, r_1), (m_2, r_2)) \ s.t.\ &[(m_1, r_1) \neq (m_2, r_2)] \\ &\wedge[\mathsf{Eval}(pk, m_1; r_1) = \mathsf{Eval}(pk, m_2; r_2)]]\end{aligned}$$

is negligible, where $(pk, sk) \overset{\mathrm{R}}{\leftarrow} \mathsf{Gen}(1^\lambda)$.

Definition 5. *A CHF scheme $\Sigma_{\mathrm{CHF}} = \{\mathsf{Gen}, \mathsf{Eval}, \mathsf{TC}, \mathsf{SKVer}, \mathsf{SKVer2}\}$ is said to have the property of random trapdoor collision, if for any $\lambda \in \mathbb{N}$, any $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and any two messages $m_1, m_2 \in \mathcal{M}$, a randomness r_1 chosen uniformly at random from $\mathcal{R}$ distributes identically with $r_2 := \mathsf{TC}(pk, (m_1, r_1), m_2)$.*

We give the definition of an original property for a CHF scheme. The property is hard-to-compute-secret-key (HtC-SK).

Definition 6. *$\Sigma_{\mathrm{CHF}} = \{\mathsf{Gen}, \mathsf{Eval}, \mathsf{TC}, \mathsf{SKVer}, \mathsf{SKVer2}\}$ is said to have the property of HtC-SK, if for every $\lambda \in \mathbb{N}$, and every PPT $\mathcal{A}$, it holds that*

$$\Pr\left[\mathcal{A}(pk, sk) \to sk^* \ s.t.\ [1 \leftarrow \mathsf{SKVer}(pk, sk^*)] \wedge [0 \leftarrow \mathsf{SKVer2}(pk, sk^*, sk)]\right]$$

is negligible, where $(pk, sk) \overset{\mathrm{R}}{\leftarrow} \mathsf{Gen}(1^\lambda)$.

Remark. The property is related to the second pre-image resistance (SPR) [15, 17]. The algorithm SKVer given pk and sk as inputs is an algorithm which verifies whether or not a relation holds between pk and sk. The algorithm $\mathsf{SKVer2}$ given two secret-keys sk and $sk^\dagger$ can be defined as the algorithm outputting 1 iff the two keys are equivalent. Thus, the HtC-SK property can be the SPR property. We can say that the HtC-SK property is a generalization of the SPR property.

3 Digital Signature

Syntax. Digital signature consists of the polynomial time algorithms $\{\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver}\}$. Gen and Sig are probabilistic, and Ver is deterministic.

$\mathsf{Gen}(1^\lambda) \rightarrow (pk, sk)$**.** The key-generation algorithm takes 1^λ, where $\lambda \in \mathbb{N}$, as an input, and outputs a public-key pk and a secret-key sk. The message space $\mathcal{M}$ is uniquely determined by pk.

$\mathsf{Sig}(pk, m, sk) \rightarrow \sigma$**.** The signing algorithm takes the public-key pk, a message $m \in \mathcal{M}$, and the secret-key sk as inputs, and outputs a signature σ.

$\mathsf{Ver}(pk, m, \sigma) \rightarrow 1 \,/\, 0$**.** The signature-verification algorithm takes the public-key pk, a message $m \in \mathcal{M}$, and a signature σ as inputs, and outputs 1 or 0.

A signature scheme must be correct. A signature scheme $\Sigma_{\text{SIG}} = \{\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver}, \mathsf{SKVer}, \mathsf{SKVer2}\}$ is correct if for every $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, every $m \in \mathcal{M}$, and every $\sigma \leftarrow \mathsf{Sig}(pk, m, sk)$, it holds that $[1 \leftarrow \mathsf{Ver}(pk, m, \sigma)]$.

3.1 Strong Existential Unforgeability in HL Model

We define the strong existential unfrogeability in HL model for a signature scheme. Specifically, we define the strong existential unforgeability against adaptively chosen messages attacks in hard-to-invert leakage model (HL-sEUF-CMA) for a signature scheme $\Sigma_{\text{SIG}} = \{\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver}\}$.

At first, we define a game which is played between an adversary $\mathcal{A}$ and challenger $\mathcal{CH}$ as follows. Note that a leakage function $f : \{0,1\}^{|pk|+|sk|} \rightarrow \{0,1\}^*$ whose randomness space is denoted by $\mathcal{R}$ is included in a class $\mathcal{F}_{\Sigma_{\text{SIG}}}(\lambda)$, i.e., $f \in \mathcal{F}_{\Sigma_{\text{SIG}}}(\lambda)$ [5].

Key-Generation. $\mathcal{CH}$ runs $(pk, sk) \leftarrow \text{SIG}.\mathsf{Gen}(1^\lambda)$. $\mathcal{CH}$ chooses $r \overset{\text{R}}{\leftarrow} \mathcal{R}$, then computes $f(pk, sk; r)$. $\mathcal{CH}$ sends $(pk, f(pk, sk; r))$ to $\mathcal{A}$. $\mathcal{CH}$ initializes the list $\mathcal{L}_S$ as an empty set $\emptyset$.

Query. $\mathcal{A}$ is allowed to use the signing oracle **Sign**, adaptively.

Sign$(m \in \mathcal{M})$**:** $\mathcal{CH}$ generates $\sigma \leftarrow \text{SIG}.\mathsf{Sig}(pk, m, sk)$, then sends σ to $\mathcal{A}$. After that, $\mathcal{CH}$ sets $\mathcal{L}_S := \mathcal{L}_S \cup \{(m, \sigma)\}$.

Forgery(m^*, σ^*)**.** $\mathcal{CH}$ receives (m^*, σ^*) sent by $\mathcal{A}$.

In the above game, $\mathcal{A}$ is said to win the game if $[1 \leftarrow \text{SIG}.\mathsf{Ver}(pk, m^*, \sigma^*)] \wedge [(m^*, \sigma^*) \notin \mathcal{L}_S]$. The advantage $\mathsf{Adv}^{\mathcal{F}(\lambda)\text{-}HL\text{-}sEUF\text{-}CMA}_{\Sigma_{\text{SIG}},\mathcal{A}}(\lambda)$ is defined as the probability $\Pr[\mathcal{A}\ \textit{wins}.]$.

Definition 7. *Σ_{SIG} is HL-sEUF-CMA-secure with respect to the leakage-function class $\mathcal{F}_{\Sigma_{\text{SIG}}}(\lambda)$, if for every PPT $\mathcal{A}$ and every function $f \in \mathcal{F}_{\Sigma_{\text{SIG}}}(\lambda)$, $\mathsf{Adv}^{\mathcal{F}(\lambda)\text{-}HL\text{-}sEUF\text{-}CMA}_{\Sigma_{\text{SIG}},\mathcal{A}}(\lambda)$ is negligible.*

Remark. Weak existential unforgeability is defined in the same manner as the strong existential unforgeability except for the winning condition by the adversary $\mathcal{A}$ in the security game. The adversary is said to win the game if the signature σ^* is a valid signature on the message m^*, i.e., $[1 \leftarrow \text{SIG}.\mathsf{Ver}(pk, m^*, \sigma^*)]$, and the message m^* has not been queried to the signing oracle **Sign**.

[5] In this paper, the function class $\mathcal{F}_{\Sigma_{\text{SIG}}}(\lambda)$ can be simply written as $\mathcal{F}(\lambda)$, if it is obvious that the function class is for the signature scheme Σ_{SIG}.

4 Signature Strongly Existentially Unforgeable and Resilient to Polynomially Hard-to-Invert Leakage

In Subsect. 4.1, the generic construction of our signature scheme is given. In Subsect. 4.2, the signature scheme is proven to be strongly existentially unforgeable and resilient to polynomially hard-to-invert leakage. In the next section, i.e., Section 5, we show that the signature scheme can be instantiated under the DLIN assumption.

4.1 Construction

Our generic construction of signature scheme $\Sigma_{\text{SIG}} = \{\mathsf{SIG.Gen}, \mathsf{SIG.Sig}, \mathsf{SIG.Ver}\}$ has the following 4 building blocks: An LPKE scheme $\Sigma_{\text{LPKE}} = \{\mathsf{LPKE.Gen}, \mathsf{LPKE.Enc}, \mathsf{LPKE.Dec}\}$, an NIZK scheme $\Sigma_{\text{NIZK}} = \{\mathsf{NIZK.Gen}, \mathsf{NIZK.Pro}, \mathsf{NIZK.Ver}\}$, a CHF scheme $\Sigma_{\text{CHF}} = \{\mathsf{CHF.Gen}, \mathsf{CHF.Eval}, \mathsf{CHF.TC}, \mathsf{CHF.SKVer}, \mathsf{CHF.SKVer2}\}$ and a CHF scheme $\Sigma_{\text{CHF2}} = \{\mathsf{CHF2.Gen}, \mathsf{CHF2.Eval}, \mathsf{CHF2.TC}\}$.

The signature scheme Σ_{SIG} is generically constructed as follows.

$\mathsf{SIG.Gen}(1^\lambda)$**:** Run $(ek, dk) \leftarrow \mathsf{LPKE.Gen}(1^\lambda)$, $(pk_1, sk_1) \leftarrow \mathsf{CHF.Gen}(1^\lambda)$ and $(pk_2, sk_2) \leftarrow \mathsf{CHF2.Gen}(1^\lambda)$. Run $(crs, td) \leftarrow \mathcal{S}_1(1^\lambda)$, where $\mathcal{S}_1$ is the first simulator in the definition of zero-knowledge for Σ_{NIZK}.
$\mathcal{R}_E$ and $\mathcal{R}_{E2}$ denote the randomness space of $\mathsf{CHF.Eval}$ and $\mathsf{CHF2.Eval}$, respectively. $\tilde{\mathcal{M}}$, $\bar{\mathcal{M}}$, $\mathcal{C}$, $\mathcal{P}$ and $\mathcal{K}_1$ denote the message space of Σ_{CHF}, the label space of Σ_{LPKE} (or the hashed value space of Σ_{CHF2}), the ciphertext space of Σ_{LPKE}, the proof space of Σ_{NIZK}, and the secret-key space of Σ_{CHF}, respectively. $\mathcal{M}$ is a space satisfying $\tilde{\mathcal{M}} = \mathcal{M} \| \mathcal{C} \| \mathcal{P}$.
Verification-key and signing-key are set as $pk := (pk_1, pk_2, ek, crs)$ and $sk := sk_1$, respectively. `Return` (pk, sk). We define language L as

$$L := \Big\{(c, \bar{m}) \in \mathcal{C} \times \bar{\mathcal{M}} \mid {}^{\exists} sk_1 \in \mathcal{K}_1 \ s.t.\ [c \leftarrow \mathsf{LPKE.Enc}(ek, sk_1, \bar{m})] \wedge [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1)]\Big\}. \tag{1}$$

$\mathsf{SIG.Sig}(pk, m \in \mathcal{M}, sk)$**:** pk is parsed as (pk_1, pk_2, ek, crs). sk is written as sk_1. Do as follows in order.

- $r'_E \overset{\text{U}}{\leftarrow} \mathcal{R}_E$, $r_{E2} \overset{\text{U}}{\leftarrow} \mathcal{R}_{E2}$, $m' \overset{\text{U}}{\leftarrow} \mathcal{M}$, $c' \overset{\text{U}}{\leftarrow} \mathcal{C}$, $\pi' \overset{\text{U}}{\leftarrow} \mathcal{P}$, $\sigma' := (c', \pi')$.
- $h := \mathsf{CHF.Eval}(pk_1, m' \| \sigma'; r'_E)$, $\bar{m} := \mathsf{CHF2.Eval}(pk_2, h; r_{E2})$.
- $c := \mathsf{LPKE.Enc}(ek, sk_1, \bar{m})$, $x := (c, \bar{m})$, $w := sk_1$, $\pi := \mathsf{NIZK.Pro}(crs, x, w)$.
- $\sigma := (c, \pi)$, $r_E := \mathsf{CHF.TC}(pk_1, sk_1, (m' \| \sigma', r'_E), m \| \sigma)$.

`Return` $\sigma^\dagger := (\sigma, r_E, r_{E2}) = (c, \pi, r_E, r_{E2})$.

$\mathsf{SIG.Ver}(pk, m \in \mathcal{M}, \sigma^\dagger)$**:** pk is parsed as (pk_1, pk_2, ek, crs). $\sigma^\dagger$ is parsed as (c, π, r_E, r_{E2}). $h := \mathsf{CHF.Eval}(pk_1, m \| \sigma; r_E)$. $\bar{m} := \mathsf{CHF2.Eval}(pk_2, h; r_{E2})$. $x := (c, \bar{m})$. `Return` $\mathsf{NIZK.Ver}(crs, x, \pi)$.

4.2 Proof of Strong Unforgeability in Polynomially Hard-to-Invert Leakage Model

Before giving the theorem for the strong unforgeability in the hard-to-invert leakage model of the signature scheme Σ_{SIG}, we give the definitions of the leakage-function class $\mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$ for the signature scheme and the strong collision-resistance in HL model w.r.t. the function class $\mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$ for the chameleon hash function Σ_{CHF}.

Definition 8. *Function class $\mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$ consists of every polynomial-time computable probabilistic (or deterministic) function $f : \{0,1\}^{|pk_1|+|pk_2|+|ek|+|crs|+|sk_1|} \to \{0,1\}^*$ which has a randomness space $\mathcal{R}$ and satisfies the following condition: for every PPT $\mathcal{B}$,*

$$\begin{aligned}&\Pr[\mathcal{B}(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, pk_2, ek, crs, sk_1; r)) \to sk_1^* \\ &\text{s.t. } [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \mathsf{CHF.SKVer2}(pk_1, sk_1^*, sk_1)]] \qquad (2)\end{aligned}$$

is negligible, where $(pk_1, sk_1) \overset{R}{\leftarrow} \mathsf{CHF.Gen}(1^\lambda)$, $(pk_2, sk_2) \overset{R}{\leftarrow} \mathsf{CHF2.Gen}(1^\lambda)$, $(ek, dk) \overset{R}{\leftarrow} \mathsf{LPKE.Gen}(1^\lambda)$, $(crs, td) \overset{R}{\leftarrow} \mathcal{S}_1(1^\lambda)$ and $r \overset{R}{\leftarrow} \mathcal{R}$.

Remark. If the chameleon hash function Σ_{CHF} is a CHF with the second pre-image resistance [15, 17], the algorithm CHF.SKVer2 is defined as the equality-checking algorithm, and the secret-key sk_1^* which satisfies $[1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \mathsf{CHF.SKVer2}(pk_1, sk_1^*, sk_1)]$ is the original secret-key sk_1 only. So, the probability (2) is simply written as $\Pr[\mathcal{B}(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, pk_2, ek, crs, sk_1; r)) \to sk_1]$.

Definition 9. *CHF scheme $\Sigma_{\mathrm{CHF}} = \{\mathsf{CHF.Gen}, \mathsf{CHF.Eval}, \mathsf{CHF.TC}, \mathsf{CHF.SKVer}, \mathsf{CHF.SKVer2}\}$ is said to be strongly collision-resistant in HL model with respect to the function class $\mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$, if for every PPT $\mathcal{A}$ and every function $f : \{0,1\}^{|pk_1|+|pk_2|+|ek|+|crs|+|sk_1|} \to \{0,1\}^*$ which is included in the function class $\mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$ and has a randomness space $\mathcal{R}$,*

$$\begin{aligned}&\Pr[\mathcal{A}(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, pk_2, ek, crs, sk_1; r)) \to ((m_1, r_1), (m_2, r_2)) \\ &\text{s.t. } [(m_1, r_1) \neq (m_2, r_2)] \wedge [\mathsf{CHF.Eval}(pk_1, m_1; r_1) = \mathsf{CHF.Eval}(pk_1, m_2; r_2)]]\end{aligned}$$

is negligible, where $(pk_1, sk_1) \overset{R}{\leftarrow} \mathsf{CHF.Gen}(1^\lambda)$, $(pk_2, sk_2) \overset{R}{\leftarrow} \mathsf{CHF2.Gen}(1^\lambda)$, $(ek, dk) \overset{R}{\leftarrow} \mathsf{LPKE.Gen}(1^\lambda)$, $(crs, td) \overset{R}{\leftarrow} \mathcal{S}_1(1^\lambda)$ and $r \overset{R}{\leftarrow} \mathcal{R}$.

The strong unforgeability in HL model of the signature scheme Σ_{SIG} is guaranteed by the following theorem.

Theorem 1. *Σ_{SIG} is HL-sEUF-CMA w.r.t. the function class $\mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$, if*

- *Σ_{LPKE} is IND-wLCCA,*
- *Σ_{NIZK} is sound and zero-knowledge,*
- *Σ_{CHF} is strongly collision-resistant in HL model w.r.t. the function class $\mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$, random trapdoor collision, and HtC-SK, and*
- *Σ_{CHF2} is strongly collision-resistant and random trapdoor collision.*

Proof of Theorem 1. Hereafter, $q_s \in \mathbb{N}$ denotes the number of times that PPT adversary $\mathcal{A}$ uses the signing oracle **Sign**. To prove Theorem 1, we use multiple games $\mathtt{Game}_i$, where $i \in \{0, 1, 2, 3, 4, 5, 6, 7, 7|1, \cdots, 7|q_s\}$.

The first game $\mathtt{Game}_0$ is the normal AL-sEUF-CMA game w.r.t. the signature scheme Σ_{SIG} and the function class $\mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$. Specifically, $\mathtt{Game}_0$ is the following game.

Key-Generation. $\mathcal{CH}$ runs $(pk_1, sk_1) \leftarrow \mathrm{CHF.Gen}(1^\lambda)$, $(pk_2, sk_2) \leftarrow \mathrm{CHF2.Gen}(1^\lambda)$, $(ek, dk) \leftarrow \mathrm{LPKE.Gen}(1^\lambda)$, and $(crs, td) \leftarrow \mathcal{S}_1(1^\lambda)$. pk and sk are set as $pk := (pk_1, pk_2, ek, crs)$ and $sk := sk_1$, respectively. For a function $f \in \mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$, $\mathcal{CH}$ chooses $r \overset{\mathrm{R}}{\leftarrow} \mathcal{R}$, then computes $f(pk, sk; r)$. $\mathcal{CH}$ sends $(pk, f(pk, sk; r))$ to $\mathcal{A}$. $\mathcal{L}_S$ is set to $\emptyset$.

Query. When $\mathcal{A}$ issues a message $m \in \mathcal{M}$ as a query to the signing oracle **Sign**, $\mathcal{CH}$ generates a signature (c, π, r_E, r_{E2}) on the message as follows.
- $r'_E \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_E$, $r_{E2} \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_{E2}$, $m' \overset{\mathrm{U}}{\leftarrow} \mathcal{M}$, $c' \overset{\mathrm{U}}{\leftarrow} \mathcal{C}$, $\pi' \overset{\mathrm{U}}{\leftarrow} \mathcal{P}$, $\sigma' := (c', \pi')$.
- $h := \mathrm{CHF.Eval}(pk_1, m' \| \sigma'; r'_E)$, $\bar{m} := \mathrm{CHF2.Eval}(pk_2, h; r_{E2})$.
- $c := \mathrm{LPKE.Enc}(ek, sk_1, \bar{m})$, $x := (c, \bar{m})$, $w := sk_1$, $\pi := \mathrm{NIZK.Pro}(crs, x, w)$.
- $\sigma := (c, \pi)$, $r_E := \mathrm{CHF.TC}(pk_1, sk_1, (m' \| \sigma', r'_E), m \| \sigma)$.

$\mathcal{CH}$ returns a signature (c, π, r_E, r_{E2}) to $\mathcal{A}$. $\mathcal{CH}$ sets $\mathcal{L}_S := \mathcal{L}_S \cup \{(m, c, \pi, r_E, r_{E2})\}$.

Forgery$(m^*, (c^*, \pi^*, r^*_E, r^*_{E2}))$**.** $\mathcal{CH}$ is given a message $m^* \in \mathcal{M}$ and a signature $(c^*, \pi^*, r^*_E, r^*_{E2})$.

$\mathcal{A}$ wins the game, if $[1 \leftarrow \mathrm{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r^*_E, r^*_{E2}) \notin \mathcal{L}_S]$, where $h^* := \mathrm{CHF.Eval}(pk_1, m^* \| (c^*, \pi^*); r^*_E)$, $\bar{m}^* := \mathrm{CHF2.Eval}(pk_2, h^*; r^*_{E2})$ and $x^* := (c^*, \bar{m}^*)$.

We define the games $\mathtt{Game}_i$, where $i \in \{1, 2, 3, 4, 5, 6, 7, 7|1, \cdots, 7|q_s\}$, as follows.

Game$_1$**.** $\mathtt{Game}_1$ is the same as $\mathtt{Game}_0$ except that $\mathcal{CH}$ generates a common reference string crs by running $crs \leftarrow \mathrm{NIZK.Gen}(1^\lambda)$ in **Key-Generation**.

Game$_2$**.** $\mathtt{Game}_2$ is the same as $\mathtt{Game}_1$ except that $\mathcal{A}$'s winning condition is changed to the following one, where $sk^*_1 := \mathrm{LPKE.Dec}(dk, c^*, \bar{m}^*)$: $[1 \leftarrow \mathrm{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r^*_E, r^*_{E2}) \notin \mathcal{L}_S] \wedge [1 \leftarrow \mathrm{CHF.SKVer}(pk_1, sk^*_1)]$.

Game$_3$**.** $\mathtt{Game}_3$ is the same as $\mathtt{Game}_2$ except that $\mathcal{A}$'s winning condition is changed to the following one: $[1 \leftarrow \mathrm{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r^*_E, r^*_{E2}) \notin \mathcal{L}_S] \wedge [1 \leftarrow \mathrm{CHF.SKVer}(pk_1, sk^*_1)] \wedge [1 \leftarrow \mathrm{CHF.SKVer2}(pk_1, sk^*_1, sk_1)]$.

Game$_4$**:** $\mathtt{Game}_4$ is the same as $\mathtt{Game}_3$ except that $\mathcal{A}$'s winning condition is changed to the following one: $[1 \leftarrow \mathrm{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r^*_E, r^*_{E2}) \notin \mathcal{L}_S] \wedge [1 \leftarrow \mathrm{CHF.SKVer}(pk_1, sk^*_1)] \wedge [1 \leftarrow \mathrm{CHF.SKVer2}(pk_1, sk^*_1, sk_1)] \wedge [[\bar{m}^* \notin \{\bar{m}_1, \cdots, \bar{m}_{q_s}\}] \vee [{}^\exists i \in [1, q_s]\ s.t.\ [\bar{m}^* = \bar{m}_i] \wedge [(h^*, r^*_{E2}) = (h_i, r_{E2,i})] \wedge [(m^*, c^*, \pi^*, r^*_E) \neq (m_i, c_i, \pi_i, r_{E,i})]]]$, where, for $i \in [1, q_s]$, each one of $\bar{m}_i, h_i, r_{E2,i}, c_i, \pi_i$ and $r_{E,i}$ is the element which was generated when computing the reply to the i-th signing oracle query.

Game$_5$ $\mathtt{Game}_5$ is the same as $\mathtt{Game}_4$ except that when $\mathcal{A}$ issues a message $m \in \mathcal{M}$ as a query to the signing oracle **Sign**, $\mathcal{CH}$ generates a signature (c, π, r_E, r_{E2}) on the message as follows.
- $r_E, r'_E \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_E$, $r'_{E2} \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_{E2}$, $m' \overset{\mathrm{U}}{\leftarrow} \mathcal{M}$, $c' \overset{\mathrm{U}}{\leftarrow} \mathcal{C}$, $\pi' \overset{\mathrm{U}}{\leftarrow} \mathcal{P}$, $\sigma' := (c', \pi')$.
- $h' := \mathrm{CHF.Eval}(pk_1, m' \| \sigma'; r'_E)$, $\bar{m} := \mathrm{CHF2.Eval}(pk_2, h'; r'_{E2})$.
- $c := \mathrm{LPKE.Enc}(ek, sk_1, \bar{m})$, $x := (c, \bar{m})$, $w := sk_1$, $\pi := \mathrm{NIZK.Pro}(crs, x, w)$.
- $\sigma := (c, \pi)$, $h := \mathrm{CHF.Eval}(pk_1, m \| \sigma; r_E)$.
- $r_{E2} := \mathrm{CHF2.TC}(pk_2, sk_2, (h', r'_{E2}), h)$.

Game$_6$**.** $\mathtt{Game}_6$ is the same as $\mathtt{Game}_5$ except that the following two points. Firstly, $\mathcal{CH}$ generates a common reference string crs by running $(crs, td) \leftarrow \mathcal{S}_1(1^\lambda)$ in **Key-Generation**. Secondly, when replying to a query to **Sign** in **Query**, $\mathcal{CH}$ generates a proof π by using $\mathcal{S}_2$, instead of NIZK.Pro, where $\mathcal{S}_2$ denotes the second simulator in the definition of zero-knowledge for Σ_{NIZK}.

$\texttt{Game}_7 (= \texttt{Game}_{7|0})$. $\texttt{Game}_7$ is the same as $\texttt{Game}_6$ except that $\mathcal{A}$'s winning condition is changed to the following one: $[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \wedge [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \mathsf{CHF.SKVer2}(pk_1, sk_1^*, sk_1)] \wedge [\bar{m}^* \notin \{\bar{m}_1, \cdots, \bar{m}_{q_s}\}]$.

$\texttt{Game}_{7|1}, \cdots, \texttt{Game}_{7|q_s}$. $\texttt{Game}_{7|i}$, where $i \in [1, q_s]$, is the same as $\texttt{Game}_{7|0}$ except that when replying to the j-th signing oracle query, where $j \leq i$, $\mathcal{CH}$ generates the ciphertext c_j by running $c_j \leftarrow \mathsf{LPKE.Enc}(ek, 0^{|sk_1|}, \bar{m}_j)$, where $0^{|sk_1|}$ denotes the bitstring of $|sk_1|$ number of 0.

Hereafter, W_i, where $i \in \{0, 1, 2, 3, 4, 5, 6, 7, 7|1, \cdots, 7|q_s\}$, denotes the event that $\mathcal{A}$ wins the game $\texttt{Game}_i$. It holds obviously that

$$\begin{aligned}\mathsf{Adv}_{\Sigma_{\mathrm{SIG}},\mathcal{A}}^{\mathcal{F}_{\Sigma_{\mathrm{SIG}}}^{HtI}(\lambda)\text{-}HL\text{-}sEUF\text{-}CMA}(\lambda) &= \Pr[W_0] \\ &\leq \sum_{i=1}^{7} |\Pr[W_{i-1}] - \Pr[W_i]| + \sum_{i=1}^{q_s} \left|\Pr[W_{7|i-1}] - \Pr[W_{7|i}]\right| + \Pr\left[W_{7|q_s}\right].\end{aligned}$$

Theorem 1 is proven by the above inequality and the following all lemmas.

Lemma 1. $|\Pr[W_0] - \Pr[W_1]|$ *is negligible if* Σ_{NIZK} *is zero-knowledge.*

Lemma 2. $|\Pr[W_1] - \Pr[W_2]|$ *is negligible if* Σ_{NIZK} *is sound.*

Lemma 3. $|\Pr[W_2] - \Pr[W_3]|$ *is negligible if* Σ_{CHF} *is HtC-SK.*

Lemma 4. $|\Pr[W_3] - \Pr[W_4]|$ *is negligible if* Σ_{CHF2} *is strongly collision-resistant.*

Lemma 5. $|\Pr[W_4] - \Pr[W_5]|$ *is negligible if each one of* Σ_{CHF} *and* Σ_{CHF2} *is random trapdoor collision.*

Lemma 6. $|\Pr[W_5] - \Pr[W_6]|$ *is negligible if* Σ_{NIZK} *is zero-knowledge.*

Lemma 7. $|\Pr[W_6] - \Pr[W_{7|0}]|$ *is negligible if* Σ_{CHF} *is strongly collision-resistant in HL model w.r.t. the function class* $\mathcal{F}_{\Sigma_{\mathrm{SIG}}}^{HtI}(\lambda)$.

Lemma 8. *For any* $i \in [1, q_s]$, $|\Pr[W_{7|i-1}] - \Pr[W_{7|i}]|$ *is negligible if* Σ_{LPKE} *is IND-wLCCA.*

Lemma 9. $\Pr[W_{7|q_s}]$ *is negligible.*

Proof of each lemma is given in Sect. A. □

5 Instantiation under the DLIN assumption

As a concrete construction for the chameleon hash function Σ_{CHF}, we adopt the chameleon hash function $\Pi_{\mathrm{CHF},n}$ given in Fig. 1. For $\Pi_{\mathrm{CHF},n}$, we obtain Theorem 2, Theorem 3, and Theorem 4, whose proofs are given in A.10, A.11 and A.12, respectively.

Theorem 2. *For any* $n \in \mathbb{N}$, $\Pi_{\mathrm{CHF},n}$ *is HtC-SK under the DL assumption.*

```
CHF.Gen($1^\lambda, 1^n$) :
    $(p, \mathbb{G}, g) \leftarrow \mathcal{G}(1^\lambda)$. $x_1, \cdots, x_n, a_1, \cdots, a_n \overset{\text{U}}{\leftarrow} \mathbb{Z}_p$. $g_1 := g^{a_1}, \cdots, g_n := g^{a_n}$.$y := \prod_{i=1}^{n} g_i^{x_i}$.
    Return $pk := (p, \mathbb{G}, g_1, \cdots, g_n, y)$ and $sk := (x_1, \cdots, x_n)$.
CHF.Eval($pk, m; \mathbf{r}$) :
    $\mathbf{r} \overset{\text{U}}{\leftarrow} \mathbb{Z}_p^n$, where $\mathbf{r}$ is parsed as $(r_1, \cdots, r_n)$. Return $\left(y \cdot \Pi_{i=1}^{n} g_i^{r_i}\right)^{J(m)}$.
CHF.TC($pk, sk, (m, \mathbf{r}), m'$) :
    $sk \in \mathbb{Z}_p^n$ is parsed as $(x_1, \cdots, x_n)$. $\mathbf{r} \in \mathbb{Z}_p^n$ is parsed as $(r_1, \cdots, r_n)$.
    For $i \in [1, n]$, $r'_i := J(m)(x_i - r_i)/J(m') - x_i$. Return $\mathbf{r}' := (r'_1, \cdots, r'_n)$.
CHF.SKVer($pk, sk^*$) :
    $sk^* \in \mathbb{Z}_p^n$ is parsed as $(x_1^*, \cdots, x_n^*)$. Return 1, if $\left[y = \prod_{i=1}^{n} g_i^{x_i^*}\right]$. Return 0, otherwise;
CHF.SKVer2($pk, sk^*, sk'$) :
    $sk^* \in \mathbb{Z}_p^n$ is parsed as $(x_1^*, \cdots, x_n^*)$. $sk' \in \mathbb{Z}_p^n$ is parsed as $(x'_1, \cdots, x'_n)$.
    Return 1, if $\left[\bigwedge_{i=1}^{n} \left[x_i^* = x'_i\right]\right]$. Return 0, otherwise;
```

Fig. 1. Construction of CHF Scheme $\Pi_{\text{CHF},n}$, where $J : \{0,1\}^* \rightarrow \mathbb{Z}_p \setminus \{0\}$ is a collision-resistant hash function.

Theorem 3. *For any $n \in \mathbb{N}$, $\Pi_{\text{CHF},n}$ is random trapdoor collision.*

Theorem 4. *For any chameleon hash function Π_{CHF2}, any LPKE scheme Π_{LPKE}, any NIZK scheme Π_{NIZK}, and any integer $n \in \mathbb{N}$, $\Pi_{\text{CHF},n}$ is strongly collision-resistant in HL model w.r.t. the function class $\mathcal{F}_{\Pi_{\text{SIG}}}^{HtI}(\lambda)$ under the collision-resistance of the hash function $J : \{0,1\}^* \rightarrow \mathbb{Z}_p \setminus \{0\}$ and the DL assumption, where Π_{SIG} denotes the instantiation of the signature scheme Σ_{SIG} by Π_{CHF}, Π_{CHF2}, Π_{LPKE} and Π_{NIZK}.*

As a concrete construction for the chameleon hash function Σ_{CHF2}, we adopt the chameleon hash function $\Pi_{\text{CHF},1}$ which is $\Pi_{\text{CHF},n}$ in Fig. 1 with $n = 1$. The following corollary is obtained by Theorem 4, obviously.

Corollary 1. *For any $n \in \mathbb{N}$, $\Pi_{\text{CHF},n}$ is strongly collision-resistant under the collision-resistance of the hash function $J : \{0,1\}^* \rightarrow \mathbb{Z}_p \setminus \{0\}$ and the DL assumption.*

Thus, the random trapdoor collision and strong collision-resistance of the CHF scheme $\Pi_{\text{CHF},1}$ are guaranteed by Theorem 3 and Corollary 1, respectively.

As a concrete construction for the LPKE scheme Σ_{LPKE}, we adopt $\Pi_{\text{LPKE},l}$ given in Fig. 2. The LPKE scheme is a modification of the LPKE scheme by Camenisch et al. [11] which is IND-LCCA secure[6] under the DLIN assumption and the collision-resistance of hash function. Faust et al. [17] modifies the scheme by Camenisch et al. to get the LPKE scheme $\Pi_{\text{LPKE},l}$ which achieves a weaker security, i.e., IND-wLCCA, but encrypts a plaintext of arbitrary length. Thus,

Theorem 5. *For any $l \in \mathbb{N}$, $\Pi_{\text{LPKE},l}$ is IND-wLCCA under the collision-resistance of the hash function $H_{CL} : \{0,1\}^* \rightarrow \mathbb{Z}_p$ and the DLIN assumption.*

[6]IND-LCCA is stronger security notion than IND-wLCCA. For the details, refer to [17].

```
LPKE.Gen($1^\lambda, 1^l$) :
  $(p, \mathbb{G}, g) \leftarrow \mathcal{G}(1^\lambda)$. $a_1, \cdots, a_l, b_1, b_2 \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p$. $\hat{g}_0 := g, \hat{g}_1 := g^{b_1}, \hat{g}_2 := g^{b_2}, g_1 := g^{a_1}, \cdots, g_l := g^{a_l}$.
  $u_1, u_2, u_3, v_1, v_2, v_3, w_1, w_2, w_3 \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p$. $d_1 := \hat{g}_0^{u_1} \cdot \hat{g}_1^{u_2}, d_2 := \hat{g}_0^{u_1} \cdot \hat{g}_2^{u_3}$,
  $e_1 := \hat{g}_0^{v_1} \cdot \hat{g}_1^{v_2}, e_2 := \hat{g}_0^{v_1} \cdot \hat{g}_2^{v_3}, h_1 := \hat{g}_0^{w_1} \cdot \hat{g}_1^{w_2}, h_2 := \hat{g}_0^{w_1} \cdot \hat{g}_2^{w_3}$.
  $ek := (p, \mathbb{G}, \hat{g}_0, \hat{g}_1, \hat{g}_2, g_1, \cdots, g_l, d_1, d_2, e_1, e_2, h_1, h_2)$. $dk := (u_1, u_2, u_3, v_1, v_2, v_3, w_1, w_2, w_3)$.
  Return($ek, dk$).
LPKE.Enc($ek, x \in \{0,1\}^l, L \in \{0,1\}^*$) :
  For $i \in [1, l]$, the $i$-th bit of $x \in \{0,1\}^l$ is denoted by $x_i \in \{0,1\}$.
  For every $i \in [1, l]$, do:
    $r_i, s_i \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p$. $\mathbf{y}_i := (y_{i,1}, y_{i,2}, y_{i,3}) := (\hat{g}_0^{r_i+s_i}, \hat{g}_1^{r_i}, \hat{g}_2^{s_i})$. $z_i := h_1^{r_i} \cdot h_2^{s_i} \cdot g_i^{x_i}$.
    $c_i := (d_1 \cdot e_1^{t_i})^{r_i} \cdot (d_2 \cdot e_2^{t_i})^{s_i}$, where $t_i := H_{CL}(\mathbf{y}_i, z_i, L)$. $\mathbf{c}_i := (\mathbf{y}_i, z_i, c_i)$.
  Return $\mathbf{C} := \{\mathbf{c}_i\}_{i \in [1,l]}$.
LPKE.Dec($ek, dk, \mathbf{C}, L$) :
  For every $i \in [1, l]$, do:
    $\tilde{c}_i := y_{i,1}^{u_1+t_i v_1} \cdot y_{i,2}^{u_2+t_i v_2} \cdot y_{i,3}^{u_3+t_i v_3}$, where $t_i := H_{CL}(\mathbf{y}_i, z_i, L)$.
    If $\tilde{c}_i \neq c_i$, then return $\perp$.
    Else if $z_i/(y_{i,1}^{w_1} \cdot y_{i,2}^{w_2} \cdot y_{i,3}^{w_3}) = g_i$, then $x'_i := 1$. Else, then $x'_i := 0$.
    The $i$-th bit of $x'$ is set as $x'_i$.
  Return $x' \in \{0,1\}^l$.
```

Fig. 2. Construction of LPKE Scheme $\Pi_{\mathrm{LPKE},l}$, where $H_{CL} : \{0,1\}^* \to \mathbb{Z}_p$ is a collision-resistant hash function.

As a concrete construction for the non-interactive zero-knowledge proof Σ_{NIZK}, we adopt the Groth-Sahai proof Π_{NIZK} in [19] whose soundness and zero-knowledge are guaranteed under the DLIN assumption.

By the schemes $\Pi_{\mathrm{CHF},n}, \Pi_{\mathrm{CHF2}}, \Pi_{\mathrm{NIZK}}$ and $\Pi_{\mathrm{LPKE},n\lambda}$, where λ denotes the integer in the security parameter 1^λ of $\Pi_{\mathrm{CHF},n}$, our concrete signature scheme Π_{SIG} is constructed. Hereafter, for $i \in [1, n]$ and $j \in [1, \lambda]$, the j-th bit of $x_i \in \mathbb{Z}_p$ in $\Pi_{\mathrm{CHF},n}$ is denoted by $x_{ij} \in \{0,1\}$, and the prime and group in $\Pi_{\mathrm{LPKE},n\lambda}$ are written as $\hat{p}$ and $\hat{\mathbb{G}}$, respectively. By the signing algorithm of Π_{SIG}, a ciphertext $\mathbf{C}$ and a proof π are generated as follows.

The ciphertext $\mathbf{C}$ is generated by running $\mathbf{C} \leftarrow \mathsf{LPKE.Enc}(ek, (x_1, \cdots, x_n), \bar{m})$, where LPKE.Enc is the encryption algorithm of $\Pi_{\mathrm{LPKE},n\lambda}$. $\mathbf{C}$ is parsed as $\{\mathbf{c}_{ij}\}_{i \in [1,n], j \in [1,\lambda]}$. $\mathbf{c}_{ij}$ is parsed as $(\mathbf{y}_{ij}, z_{ij} \in \hat{\mathbb{G}}, c_{ij} \in \hat{\mathbb{G}})$. $\mathbf{y}_{ij}$ is parsed as $(y_{ij,1}, y_{ij,2}, y_{ij,3}) \in \hat{\mathbb{G}}^3$.

By using the proof-generation algorithm of the NIZK scheme Π_{NIZK}, we generate the proof π. Actually, the proof π is a proof which proves that

$$\exists \{r_{ij} \in \mathbb{Z}_{\hat{p}}, s_{ij} \in \mathbb{Z}_{\hat{p}}, x_{ij} \in \{0,1\}\}_{i \in [1,n], j \in [1,\lambda]} \ \textit{such that}$$
$$\left[\prod_{i=1}^{n} \prod_{j=1}^{\lambda} g_i^{2^{j-1} \cdot x_{ij}} = y\right] \bigwedge_{i \in [1,n], j \in [1,\lambda]} \left[\left[\hat{g}_0^{r_{ij}+s_{ij}} = y_{ij,1}\right] \wedge \left[\hat{g}_1^{r_{ij}} = y_{ij,2}\right] \wedge \left[\hat{g}_2^{s_{ij}} = y_{ij,3}\right]\right.$$
$$\left. \wedge \left[h_1^{r_{ij}} \cdot h_2^{s_{ij}} \cdot g_{ij}^{x_{ij}} = z_{ij}\right] \wedge \left[(d_1 \cdot e_1^{t_{ij}})^{r_{ij}} \cdot (d_2 \cdot e_2^{t_{ij}})^{s_{ij}} = c_{ij}\right] \wedge \left[x_{ij}(1 - x_{ij}) = 0\right]\right],$$

where $y = \prod_{i=1}^{n} g_i^{x_i} \in \mathbb{G}$.

Acknowledgements

This work was supported by JSPS KAKENHI (Grant Number JP17KT0081). This work was also supported by JSPS and DST under the Japan-India Science Cooperative Program.

References

1. Akavia, A., Goldwasser, S., Vaikuntanathan, V. :Simultaneous hardcore bits and cryptography against memory attacks. In: TCC 2009, LNCS 5444, pp. 474-495, 2009.
2. Alwen, J., Dodis, Y., Wichs, D. :Leakage-resilient public-key cryptography in the bounded-retrieval model. In: CRYPTO 2009, LNCS 5677, pp. 36-54, 2009.
3. Brands, S.A. :An efficient off-line electronic cash system based on the representation problem. Technical report, 1993.
4. Boneh, D., Boyen, X., Shacham, H. :Short group signatures. In: CRYPTO 2004, LNCS 3152, pp.41-55, 2004.
5. Brakerski, Z., Goldwasser, S. :Circular and leakage resilient public-key encryption under subgroup indistinguishability (or: quadratic residuosity strikes back). In: CRYPTO 2010, LNCS 6223, pp. 1-10, 2010.
6. Bellare, M., Goldreich, O., Goldwasser, S. :Incremental cryptography: the case of hashing and signing. In:CRYPTO 1994, LNCS 839, pp. 216-233, 1994.
7. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R. :Circular-secure encryption from decision diffie-hellman. In: CRYPTO 2008, LNCS 5157, pp. 108-125, 2008.
8. Brakerski, Z., Kalai, Y.T., Katz, J., Vaikuntanathan, V. :Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In: FOCS 2010, pp. 501-510, 2010.
9. Boneh, D., Shen, E., Waters, B. :Strongly unforgeable signatures based on computational diffie-hellman. In: PKC 2006, LNCS 3958, pp. 229-240, 2006.
10. Boyle, E., Segev, G., Wichs, D. :Fully leakage-resilient signatures. In: EUROCRYPT 2011, LNCS 6632, pp. 89-108, 2011.
11. Camenisch, J., Chandran, N., Shoup, V. :A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In: EUROCRYPT 2009, LNCS 5479, pp. 351-368, 2009.
12. Chow, S.S.M, Dodis, Y., Rouselakis, Y., Waters, B. :Practical leakage-resilient identity-based encryption from simple assumptions. In: ACMCCS 2010, pp. 152-161, 2010.
13. Canetti, R., Halevi, S., Katz, J. :Chosen-ciphertext security from identity-based encryption. In: EUROCRYPT 2004, LNCS 3027, pp. 207-222, 2004.
14. Dodis, Y., Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V. :Public-key encryption with auxiliary inputs. In: TCC 2010, LNCS 5978, pp. 361-381, 2010.
15. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D. :Efficient public-key cryptography in the presence of key leakage. In: ASIACRYPT 2010, LNCS 6477, pp. 613-631, 2010.
16. Dodis, Y., Kalai, Y.T., Lovett, S. :On Cryptography with auxiliary input. In: STOC 2009, pp. 621-630, 2009.
17. Faust, S., Haway, C., Nielsen, J.B., Nordholt, P.S., Zottarel, A. :Signature schemes secure against hard-to-invert leakage. In: ASIACRYPT 2012, LNCS 7658, pp.98-115, 2012.
18. Gentry, C., Peikert, C., Vaikuntanathan, V. :Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008, pp. 197-206, 2008.
19. Groth, J., Sahai, A. :Efficient non-interactive proof systems for bilinear groups. In: EUROCRYPT 2008, LNCS 4965, pp. 415-432, 2008.

20. Huang, J., Huang, Q., Pan, C. :A black-box construction of strongly unforgeable signature schemes in the bounded leakage model. In: ProvSec 2016, LNCS 10005, pp. 320-339, 2016.
21. Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W. :Lest we remember: cold boot attacks on encryption keys. In: USENIX Security Symposium, pp. 45-60, 2008.
22. Huang, Q., Wong, D.S., Zhao, Y. :Generic transformation to strongly unforgeable signatures. In: ACNS 2007, LNCS 4521, pp. 1-17, 2007.
23. Ishizaka, M., Matsuura, K. :Strongly unforgeable signature resilient to polynomially hard-to-invert leakage under standard assumptions. In: ISC 2018, LNCS ????, pp. ???-???, 2018.
24. Kurosawa, K., Phong, L.T. :Leakage resilient IBE and IPE under the DLIN assumption. In: ACNS 2013, LNCS 7954, pp. 487-501, 2013.
25. Katz, J., Vaikuntanathan, V. :Signature schemes with bounded leakage resilience. In: ASIACRYPT 2009, LNCS 5912, pp. 703-720, 2009.
26. Lewko, A., Rouselakis, Y., Waters, B. :Achieving leakage resilience through dual system encryption. In: TCC 2011, LNCS 6597, pp. 70-88, 2011.
27. Malkin, T., Teranishi, I., Vahlis, Y., Yung, M. :Signatures resilient to continual leakage on memory and computation. In: TCC 2011, LNCS 6597, pp. 89-108, 2011.
28. Naor, M., Segev, G. :Public-key cryptosystems resilient to key leakage. In: CRYPTO 2009, LNCS 5677, pp. 18-35, 2009.
29. Steinfeld, R., Pieprzyk, J., Wang, H. :How to strengthen any weakly unforgeable signature into a strongly unforgeable signature. In: CT-RSA 2007, LNCS 4377, pp. 357-371, 2007.
30. Wang, Y., Matsuda, T., Hanaoka, G., Tanaka, K. :Signatures resilient to uninvertible leakage. In: SCN 2016, LNCS 9841, pp. 372-390, 2016.
31. Wang, Y., Tanaka, K. :Generic transformation to strongly existentially unforgeable signature schemes with leakage resiliency. In: ProvSec 2014, LNCS 8782, pp. 117-129, 2014.
32. Yuen, T.H., Chow, S.S.M, Zhang, Y., Yiu, S.M. :Identity-based encryption resilient to continual auxiliary leakage. In: EUROCRYPT 2012, LNCS 7232, pp. 117-134, 2012.
33. Yuen, T.H., Yiu, S.M., Hui, L.C.K. :Fully leakage-resilient signatures with auxiliary inputs. In: ACISP 2012, LNCS 7372, pp. 294-307, 2012.
34. Zhang, M. :New model and construction of ABE: achieving key resilient-leakage and attribute direct-revocation. In: ACISP 2014, LNCS 8544, pp. 192-208, 2014.
35. Zhang, M., Shi, W., Wang, C., Chen, Z., Mu, Y. :Leakage-resilient attribute-based encryption with fast decryption: models, analysis and constructions. In: ISPEC 2013, LNCS 7863, pp. 75-90, 2013.
36. Zhang, M., Wang, C., Takagi, T., Mu, Y.: Functional encryption resilient to hard-to-invert leakage. In: The Computer Journal, doi:10.1093/comjnl/bxt105, 2013.

A Proofs of Some Theorems and Lemmas

A.1 Proof of Lemma 1

We prove that if we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_0] - \Pr[W_1]|$ non-negligible, then we are able to construct a PPT algorithm which breaks the zero-knowledge property for Σ_{NIZK}.

We consider a PPT simulator $\mathcal{S}$. On one hand, the simulator $\mathcal{S}$ is a PPT algorithm attempting to break the zero-knowledge for Σ_{NIZK}. On the other hand, $\mathcal{S}$ is the challenger in $\mathtt{Game}_0$ or $\mathtt{Game}_1$. $\mathcal{S}$ is given a common reference string crs. If crs was generated by $(crs, td) \leftarrow \mathcal{S}_1(1^\lambda)$ (resp. $crs \leftarrow \text{NIZK}.\mathsf{Gen}(1^\lambda)$), then $\mathcal{S}$ simulates $\mathtt{Game}_0$ (resp. $\mathtt{Game}_1$) against the PPT adversary $\mathcal{A}$ properly. The concrete behaviour by $\mathcal{S}$ is the following.

Key-Generation. $\mathcal{S}$ runs $(pk_1, sk_1) \leftarrow \mathsf{CHF.Gen}(1^\lambda)$, $(pk_2, sk_2) \leftarrow \mathsf{CHF2.Gen}(1^\lambda)$ and $(ek, dk) \leftarrow \mathsf{LPKE.Gen}(1^\lambda)$. $\mathcal{S}$ is given a common reference string crs of Σ_{NIZK}. $\mathcal{S}$ sets pk and sk to $pk := (pk_1, pk_2, ek, crs)$ and $sk := sk_1$, respectively. $\mathcal{S}$ computes $f(pk, sk; r)$ where $r \xleftarrow{\mathrm{R}} \mathcal{R}$. $\mathcal{S}$ sends $(pk, f(pk, sk; r))$ to $\mathcal{A}$. $\mathcal{S}$ sets $\mathcal{L}_S$ to $\emptyset$.

Query. When $\mathcal{A}$ issues a message $m \in \mathcal{M}$ as a query to the signing oracle **Sign**, $\mathcal{S}$ generates a signature (c, π, r_E, r_{E2}) on the message in the normal manner. $\mathcal{S}$ sets $\mathcal{L}_S$ to $\mathcal{L}_S \cup \{(m, c, \pi, r_E, r_{E2})\}$.

Forgery$(m^*, c^*, \pi^*, r_E^*, r_{E2}^*)$**.** $\mathcal{S}$ outputs 1, if $[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S]$, where $h^* := \mathsf{CHF.Eval}(pk_1, m^* \| (c^*, \pi^*); r_E^*)$, $\bar{m}^* := \mathsf{CHF2.Eval}(pk_2, h^*; r_{E2}^*)$ and $x^* := (c^*, \bar{m}^*)$. $\mathcal{S}$ outputs 0, otherwise.

It is obvious that if the common reference string is generated by $(crs, td) \leftarrow \mathcal{S}_1(1^\lambda)$ (resp. $crs \leftarrow \mathsf{NIZK.Gen}(1^\lambda)$), then $\mathcal{S}$ simulates the game $\mathtt{Game}_0$ (resp. $\mathtt{Game}_1$) against $\mathcal{A}$ perfectly, and if and only if the event W_0 (resp. W_1) occurs, $\mathcal{S}$ outputs 1. Hence, we obtain $\Pr[W_0] = \Pr[1 \leftarrow \mathcal{S}(crs) \mid (crs, td) \leftarrow \mathcal{S}_1(1^\lambda)]$ and $\Pr[W_1] = \Pr[1 \leftarrow \mathcal{S}(crs) \mid crs \leftarrow \mathsf{NIZK.Gen}(1^\lambda)]$. Hence, $|\Pr[W_0] - \Pr[W_1]| = |\Pr[1 \leftarrow \mathcal{S}(crs) \mid crs \leftarrow \mathsf{NIZK.Gen}(1^\lambda)] - \Pr[1 \leftarrow \mathcal{S}(crs) \mid (crs, td) \leftarrow \mathcal{S}_1(1^\lambda)]|$. □

A.2 Proof of Lemma 2

We prove that if we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_1] - \Pr[W_2]|$ non-negligible, then we are able to construct a PPT algorithm which breaks the soundness property for Σ_{NIZK}.

We consider a PPT simulator $\mathcal{S}$ attempting to break the soundness of the NIZK scheme Σ_{NIZK}. Specifically, $\mathcal{S}$ behaves as follows.

Key-Generation. $\mathcal{S}$ runs $(pk_1, sk_1) \leftarrow \mathsf{CHF.Gen}(1^\lambda)$, $(pk_2, sk_2) \leftarrow \mathsf{CHF2.Gen}(1^\lambda)$ and $(ek, dk) \leftarrow \mathsf{LPKE.Gen}(1^\lambda)$. $\mathcal{S}$ is given a common reference string crs of Σ_{NIZK}. $\mathcal{S}$ sets pk and sk to $pk := (pk_1, pk_2, ek, crs)$ and $sk := sk_1$, respectively. $\mathcal{S}$ computes $f(pk, sk; r)$ where $r \xleftarrow{\mathrm{R}} \mathcal{R}$, then sends $(pk, f(pk, sk; r))$ to $\mathcal{A}$. $\mathcal{S}$ sets $\mathcal{L}_S$ to $\emptyset$.

Query. When $\mathcal{A}$ issues a message $m \in \mathcal{M}$ as a query to the signing oracle **Sign**, $\mathcal{S}$ generates a signature (c, π, r_E, r_{E2}) on the message in the normal manner. $\mathcal{S}$ sets $\mathcal{L}_S$ to $\mathcal{L}_S \cup \{(m, c, \pi, r_E, r_{E2})\}$.

Forgery$(m^*, c^*, \pi^*, r_E^*, r_{E2}^*)$**.** $\mathcal{S}$ checks whether the following condition is satisfied or not: $[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \wedge [0 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)]$, where $h^* := \mathsf{CHF.Eval}(pk_1, m^* \| (c^*, \pi^*); r_E^*)$, $\bar{m}^* := \mathsf{CHF2.Eval}(pk_2, h^*; r_{E2}^*)$, $x^* := (c^*, \bar{m}^*)$ and $sk_1^* := \mathsf{LPKE.Dec}(dk, c^*, \bar{m}^*)$.
If the condition is satisfied, then $\mathcal{S}$ outputs $(x^*, \pi^*) = (c^*, \bar{m}^*, \pi^*)$.

It is obvious that $\mathcal{S}$ simulates $\mathtt{Game}_1$ or $\mathtt{Game}_2$, perfectly.

By the way, the definitions of W_1 and W_2 gives us the following equations.

$$\Pr[W_1] = \Pr[[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S]] \quad (3)$$

$$\begin{aligned}\Pr[W_2] = \Pr[&[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\ &\wedge [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)]] \quad (4)\end{aligned}$$

Hence, we obtain

$$
\begin{aligned}
&|\Pr[W_1] - \Pr[W_2]| \\
&= \Pr\left[[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \right.\\
&\quad \left.\wedge [0 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)]\right] \\
&= \Pr\left[\mathcal{S}(crs) \to (x^*, \pi^*) \ s.t.\ [1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \right.\\
&\quad \left.\wedge [0 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)]\right]. \qquad (5)
\end{aligned}
$$

By the definition of the language L given in (1), the following statement is true: for any $(c, \bar{m}) \in L$, there exists sk_1 such that $[c \leftarrow \mathsf{LPKE.Enc}(ek, sk_1, \bar{m})] \wedge [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1)]$.

By the above statement, the correctness of Π_{LPKE}, and the correctness of Π_{CHF}, the following statement is true: for any $(c, \bar{m}) \in L$, it holds that $[1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1)]$, where $sk_1 := \mathsf{LPKE.Dec}(dk, c, \bar{m})$.

The contraposition of the above statement is the following statement: for any $c \in C$ and any $\bar{m} \in \bar{\mathcal{M}}$, if $[0 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1)]$, where $sk_1 := \mathsf{LPKE.Dec}(dk, c, \bar{m})$, then $(c, \bar{m}) \notin L$.

By the above statement and the equation (5), we obtain

$$|\Pr[W_1] - \Pr[W_2]| = \Pr\left[\mathcal{S}(crs) \to (x^*, \pi^*) \ s.t.\ [1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [x^* \notin L]\right].$$

□

A.3 Proof of Lemma 3

We prove that if we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_2] - \Pr[W_3]|$ non-negligible, then we are able to construct a PPT algorithm which breaks the HtC-SK property for Σ_{CHF}.

We consider a PPT simulator $\mathcal{S}$ who behaves as a PPT adversary trying to break the property of HtC-SK for Σ_{CHF}. The concrete behaviour by $\mathcal{S}$ is the following.

Key-Generation. $\mathcal{S}$ is given (pk_1, sk_1) of the keys of Σ_{CHF}. $\mathcal{S}$ runs $(pk_2, sk_2) \leftarrow \mathsf{CHF2.Gen}(1^\lambda)$, $(ek, dk) \leftarrow \mathsf{LPKE.Gen}(1^\lambda)$ and $crs \leftarrow \mathsf{NIZK.Gen}(1^\lambda)$. $\mathcal{S}$ sets pk and sk to $pk := (pk_1, pk_2, ek, crs)$ and $sk := sk_1$, respectively. $\mathcal{S}$ computes $f(pk, sk; r)$, where $r \overset{\mathrm{R}}{\leftarrow} \mathcal{R}$, then sends $(pk, f(pk, sk; r))$ to $\mathcal{A}$. $\mathcal{S}$ sets $\mathcal{L}_S$ to $\emptyset$.

Query. When $\mathcal{A}$ issues a message $m \in \mathcal{M}$ as a query to the signing oracle **Sign**, $\mathcal{S}$ generates a signature (c, π, r_E, r_{E2}) on the message in the normal manner. $\mathcal{S}$ sets $\mathcal{L}_S$ to $\mathcal{L}_S \cup \{(m, c, \pi, r_E, r_{E2})\}$.

Forgery$(m^*, c^*, \pi^*, r_E^*, r_{E2}^*)$**.** $\mathcal{S}$ checks whether the following condition is satisfied or not: $[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \wedge [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)] \wedge [0 \leftarrow \mathsf{CHF.SKVer2}(pk_1, sk_1^*, sk_1)]$, where $h^* := \mathsf{CHF.Eval}(pk_1, m^* \| (c^*, \pi^*); r_E^*)$, $\bar{m}^* := \mathsf{CHF2.Eval}(pk_2, h^*; r_{E2}^*)$, $x^* := (c^*, \bar{m}^*)$ and $sk_1^* := \mathsf{LPKE.Dec}(dk, c^*, \bar{m}^*)$.

If the condition is satisfied, $\mathcal{S}$ outputs sk_1^*.

It is obvious that $\mathcal{S}$ simulates $\mathtt{Game}_2$ or $\mathtt{Game}_3$ against $\mathcal{A}$ perfectly.

By the definitions of W_2 and W_3, we obtain

$$\begin{aligned}\Pr[W_2] = \Pr[&[1 \leftarrow \text{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\ &\wedge [1 \leftarrow \text{CHF.SKVer}(pk_1, sk_1^*)]] \\ \Pr[W_3] = \Pr[&[1 \leftarrow \text{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\ &\wedge [1 \leftarrow \text{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \text{CHF.SKVer2}(pk_1, sk_1^*, sk_1)]]\end{aligned}$$

Hence, we obtain

$$\begin{aligned}|\Pr[W_2] - \Pr[W_3]| &= \Pr[[1 \leftarrow \text{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\ \wedge [1 \leftarrow \text{CHF.SKVer}(pk_1, sk_1^*)] &\wedge [0 \leftarrow \text{CHF.SKVer2}(pk_1, sk_1^*, sk_1)]]. \quad (6)\end{aligned}$$

The above probability is equal to the probability with whom $\mathcal{S}$ wins the HtC-SK property game for Π_{CHF}. □

A.4 Proof of Lemma 4

We prove that if we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_3] - \Pr[W_4]|$ non-negligible, then we are able to construct a PPT algorithm which breaks the strong collision-resistance property for Σ_{CHF2}.

We consider a PPT simulator $\mathcal{S}$ who behaves as a PPT adversary trying to break the property of strong collision-resistance for Σ_{CHF2}. The concrete behaviour by $\mathcal{S}$ is the following.

Key-Generation. $\mathcal{S}$ is given a public key pk_2 of Σ_{CHF2}. $\mathcal{S}$ runs $(pk_1, sk_1) \leftarrow \text{CHF.Gen}(1^\lambda)$, $(ek, dk) \leftarrow \text{LPKE.Gen}(1^\lambda)$ and $crs \leftarrow \text{NIZK.Gen}(1^\lambda)$. $\mathcal{S}$ sets pk and sk to $pk := (pk_1, pk_2, ek, crs)$ and $sk := sk_1$, respectively. $\mathcal{S}$ computes $f(pk, sk; r)$, where $r \overset{\text{R}}{\leftarrow} \mathcal{R}$, then sends $(pk, f(pk, sk; r))$ to $\mathcal{A}$. $\mathcal{S}$ sets $\mathcal{L}_S$ and $\mathcal{L}_{\bar{m}}$ to $\emptyset$.

Query. When $\mathcal{A}$ issues a message $m_i \in \mathcal{M}$ as the i-th query to the signing oracle **Sign**, $\mathcal{S}$ generates a signature $(c_i, \pi_i, r_{E,i}, r_{E2,i})$ on the message as follows.

- $r'_{E,i} \overset{\text{U}}{\leftarrow} \mathcal{R}_E$, $r_{E2,i} \overset{\text{U}}{\leftarrow} \mathcal{R}_{E2}$, $m'_i \overset{\text{U}}{\leftarrow} \mathcal{M}$, $c'_i \overset{\text{U}}{\leftarrow} \mathcal{C}$, $\pi'_i \overset{\text{U}}{\leftarrow} \mathcal{P}$, $\sigma'_i := (c'_i, \pi'_i)$.
- $h_i := \text{CHF.Eval}(pk_1, m'_i \| \sigma'_i; r'_{E,i})$, $\bar{m}_i := \text{CHF2.Eval}(pk_2, h_i; r_{E2,i})$.
- $c_i := \text{LPKE.Enc}(ek, sk_1, \bar{m}_i)$, $x_i := (c_i, \bar{m}_i)$, $w := sk_1$, $\pi_i := \text{NIZK.Pro}(crs, x_i, w)$.
- $\sigma_i := (c_i, \pi_i)$, $r_{E,i} := \text{CHF.TC}(pk_1, sk_1, (m'_i \| \sigma'_i, r'_{E,i}), m_i \| \sigma_i)$.

$\mathcal{S}$ returns $(c_i, \pi_i, r_{E2,i}, r_{E,i})$ to $\mathcal{A}$. $\mathcal{L}_S$ is set to $\mathcal{L}_S \cup \{(m_i, c_i, \pi_i, r_{E2,i}, r_{E,i})\}$. $\mathcal{L}_{\bar{m}}$ is set to $\mathcal{L}_{\bar{m}} \cup \{\bar{m}_i\}$.

Forgery$(m^*, c^*, \pi^*, r_E^*, r_{E2}^*)$. $\mathcal{S}$ sets $h^* := \text{CHF.Eval}(pk_1, m^* \| (c^*, \pi^*); r_E^*)$, $\bar{m}^* := \text{CHF2.Eval}(pk_2, h^*; r_{E2}^*)$, $x^* := (c^*, \bar{m}^*)$ and $sk_1^* := \text{LPKE.Dec}(dk, c^*, \bar{m}^*)$. If $[1 \leftarrow \text{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \wedge [1 \leftarrow \text{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \text{CHF.SKVer2}(pk_1, sk_1^*, sk_1)] \wedge [{}^\exists i \in [1, q_s] \ s.t.\ [\bar{m}^* = \bar{m}_i] \wedge [(h^*, r_{E2}^*) \neq (h_i, r_{E2,i})]]$, then $\mathcal{S}$ outputs $((h^*, r_{E2}^*), (h_i, r_{E2,i}))$.

It is obvious that $\mathcal{S}$ simulates $\mathtt{Game}_3$ or $\mathtt{Game}_4$ against $\mathcal{A}$ perfectly. By the definitions of W_3 and W_4, we obtain

$$\begin{aligned}
&\Pr[W_3] = \Pr[[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\
&\wedge [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \mathsf{CHF.SKVer2}(pk_1, sk_1^*, sk_1)]] \qquad (7) \\
&\Pr[W_4] = \Pr[[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\
&\wedge [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \mathsf{CHF.SKVer2}(pk_1, sk_1^*, sk_1)] \\
&\wedge \Big[[\bar{m}^* \notin \mathcal{L}_{\bar{m}}] \vee \Big[{}^{\exists} i \in [1, q_s] \ s.t.\ [\bar{m}^* = \bar{m}_i] \wedge [(h^*, r_{E2}^*) = (h_i, r_{E2,i})] \\
&\wedge [(m^*, c^*, \pi^*, r_E^*) \neq (m_i, c_i, \pi_i, r_{E,i})]]]]] \qquad (8)
\end{aligned}$$

By (7) and (8), we obtain

$$\begin{aligned}
&|\Pr[W_3] - \Pr[W_4]| \\
&= \Pr[[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\
&\quad \wedge [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \mathsf{CHF.SKVer2}(pk_1, sk_1^*, sk_1)] \\
&\quad \wedge \Big[{}^{\exists} i \in [1, q_s] \ s.t.\ [\bar{m}^* = \bar{m}_i] \wedge [(h^*, r_{E2}^*) \neq (h_i, r_{E2,i})]\Big]\Big]
\end{aligned}$$

The above probability is equal to the probability with whom $\mathcal{S}$ wins the strong collision-resistance game for Π_{CHF2}. □

A.5 Proof of Lemma 5

For a message $\hat{m} \in \mathcal{M}$ queried to the signing oracle in $\mathtt{Game}_4$ (resp. $\mathtt{Game}_5$), $P_4(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}})$ (resp. $P_5(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}})$) denotes the probability that the signature $(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}}) \in \mathcal{C} \times \mathcal{P} \times \mathcal{R}_E \times \mathcal{R}_{E2}$ is generated.

For the probability $P_4(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}})$, we obtain

$$\begin{aligned}
&P_4(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}}) = \Pr[[\hat{c} = \mathsf{LPKE.Enc}(ek, sk_1, \bar{m})] \wedge [\hat{\pi} = \mathsf{NIZK.Pro}(crs, x, sk_1)] \\
&\wedge [\hat{r_E} = \mathsf{CHF.TC}(pk_1, sk_1, (m' \| (c', \pi'), r_E'), \hat{m} \| (\hat{c}, \hat{\pi}))] \wedge [\hat{r_{E2}} = r_{E2}] \\
&\mid r_E' \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_E, r_{E2} \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_{E2}, m' \overset{\mathrm{U}}{\leftarrow} \mathcal{M}, c' \overset{\mathrm{U}}{\leftarrow} \mathcal{C}, \pi' \overset{\mathrm{U}}{\leftarrow} \mathcal{P}\Big] \qquad (9) \\
&= \Pr[[\hat{c} = \mathsf{LPKE.Enc}(ek, sk_1, \bar{m})] \wedge [\hat{\pi} = \mathsf{NIZK.Pro}(crs, x, sk_1)] \\
&\mid r_E' \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_E, r_{E2} \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_{E2}, m' \overset{\mathrm{U}}{\leftarrow} \mathcal{M}, c' \overset{\mathrm{U}}{\leftarrow} \mathcal{C}, \pi' \overset{\mathrm{U}}{\leftarrow} \mathcal{P}\Big] \\
&\cdot \Pr[\hat{r_E} = \mathsf{CHF.TC}(pk_1, sk_1, (m' \| (c', \pi'), r_E'), \hat{m} \| (\hat{c}, \hat{\pi})) \\
&\mid r_E' \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_E, m' \overset{\mathrm{U}}{\leftarrow} \mathcal{M}, c' \overset{\mathrm{U}}{\leftarrow} \mathcal{C}, \pi' \overset{\mathrm{U}}{\leftarrow} \mathcal{P}\Big] \cdot \Pr\Big[\hat{r_{E2}} = r_{E2} \mid r_{E2} \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_{E2}\Big] \qquad (10) \\
&= \Pr[[\hat{c} = \mathsf{LPKE.Enc}(ek, sk_1, \bar{m})] \wedge [\hat{\pi} = \mathsf{NIZK.Pro}(crs, x, sk_1)] \\
&\mid r_E' \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_E, r_{E2} \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_{E2}, m' \overset{\mathrm{U}}{\leftarrow} \mathcal{M}, c' \overset{\mathrm{U}}{\leftarrow} \mathcal{C}, \pi' \overset{\mathrm{U}}{\leftarrow} \mathcal{P}\Big] \cdot \frac{1}{|\mathcal{R}_E|} \cdot \frac{1}{|\mathcal{R}_{E2}|}, \qquad (11)
\end{aligned}$$

where $h' := \mathsf{CHF.Eval}(pk_1, m' \| (c', \pi'); r_E')$, $\bar{m} := \mathsf{CHF2.Eval}(pk_2, h'; r_{E2})$ and $x := (\hat{c}, \bar{m})$ (9) is the definition. The transformation from (9) to (10) is correct since each event is independent. The transformation from (10) to (11) is correct since the CHF scheme Π_{CHF} is random trapdoor collision.

On the other hand, for the probability $P_5(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}})$, we obtain

$$\begin{aligned}
&P_5(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}}) = \Pr[[\hat{c} = \mathsf{LPKE.Enc}(ek, sk_1, \bar{m})] \wedge [\hat{\pi} = \mathsf{NIZK.Pro}(crs, x, sk_1)] \\
&\quad \wedge [\hat{r_E} = r_E] \wedge \left[\hat{r_{E2}} = \mathsf{CHF2.TC}(pk_2, sk_2, (h', r'_{E2}), \hat{h})\right] \\
&\quad \left| r_E, r'_E \xleftarrow{U} \mathcal{R}_E, r'_{E2} \xleftarrow{U} \mathcal{R}_{E2}, m' \xleftarrow{U} \mathcal{M}, c' \xleftarrow{U} C, \pi' \xleftarrow{U} \mathcal{P}\right] \qquad (12) \\
&= \Pr[[\hat{c} = \mathsf{LPKE.Enc}(ek, sk_1, \bar{m})] \wedge [\hat{\pi} = \mathsf{NIZK.Pro}(crs, x, sk_1)] \\
&\quad \left| r'_E \xleftarrow{U} \mathcal{R}_E, r'_{E2} \xleftarrow{U} \mathcal{R}_{E2}, m' \xleftarrow{U} \mathcal{M}, c' \xleftarrow{U} C, \pi' \xleftarrow{U} \mathcal{P}\right] \\
&\quad \cdot \Pr\left[\hat{r_E} = r_E \mid r_E \xleftarrow{U} \mathcal{R}_E\right] \cdot \Pr\left[\hat{r_{E2}} = \mathsf{CHF2.TC}(pk_2, sk_2, (h', r'_{E2}), \hat{h})\right. \\
&\quad \left| r_E, r'_E \xleftarrow{U} \mathcal{R}_E, r'_{E2} \xleftarrow{U} \mathcal{R}_{E2}, m' \xleftarrow{U} \mathcal{M}, c' \xleftarrow{U} C, \pi' \xleftarrow{U} \mathcal{P}\right] \qquad (13) \\
&= \Pr[[\hat{c} = \mathsf{LPKE.Enc}(ek, sk_1, \bar{m})] \wedge [\hat{\pi} = \mathsf{NIZK.Pro}(crs, x, sk_1)] \\
&\quad \left| r'_E \xleftarrow{U} \mathcal{R}_E, r'_{E2} \xleftarrow{U} \mathcal{R}_{E2}, m' \xleftarrow{U} \mathcal{M}, c' \xleftarrow{U} C, \pi' \xleftarrow{U} \mathcal{P}\right] \cdot \frac{1}{|\mathcal{R}_E|} \cdot \frac{1}{|\mathcal{R}_{E2}|}, \qquad (14)
\end{aligned}$$

where $h' := \mathsf{CHF.Eval}(pk_1, m' \| (c', \pi'); r'_E)$, $\bar{m} := \mathsf{CHF2.Eval}(pk_2, h'; r'_{E2})$, $x := (\hat{c}, \bar{m})$, and $\hat{h} := \mathsf{CHF.Eval}(pk_1, \hat{m} \| (\hat{c}, \hat{\pi}); r_E)$. (12) is the definition. The transformation from (12) to (13) is correct since each event is independent. The transformation from (13) to (14) is correct since the CHF scheme Π_{CHF2} is random trapdoor collision.

By (11) and (14), $P_4(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}}) = P_5(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}})$. Thus, for any $\hat{m} \in \mathcal{M}$ and any signature $(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}}) \in C \times \mathcal{P} \times \mathcal{R}_E \times \mathcal{R}_{E2}$, the probability in $\mathtt{Game}_4$ that the signature $(\hat{c}, \hat{\pi}, \hat{r_E}, \hat{r_{E2}})$ on the message $\hat{m}$ is generated is equal to the one in $\mathtt{Game}_5$. □

A.6 Proof of Lemma 6

We prove that if we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_5] - \Pr[W_6]|$ non-negligible, then we are able to construct a PPT algorithm which breaks the zero-knowledge property for Σ_{NIZK}.

We consider a PPT simulator $\mathcal{S}$ attempting to break the zero-knowledge property for the NIZK scheme Σ_{NIZK}. Specifically, $\mathcal{S}$ behaves as follows.

Key-Generation. $\mathcal{S}$ is given a common reference string crs of Σ_{NIZK}. $\mathcal{S}$ runs $(pk_1, sk_1) \leftarrow \mathsf{CHF.Gen}(1^\lambda)$, $(pk_2, sk_2) \leftarrow \mathsf{CHF2.Gen}(1^\lambda)$ and $(ek, dk) \leftarrow \mathsf{LPKE.Gen}(1^\lambda)$. $\mathcal{S}$ sets pk and sk to $pk := (pk_1, pk_2, ek, crs)$ and $sk := sk_1$, respectively. $\mathcal{S}$ computes $f(pk, sk; r)$, where $r \xleftarrow{R} \mathcal{R}$, then sends $(pk, f(pk, sk; r))$ to $\mathcal{A}$. $\mathcal{S}$ sets each list of $\mathcal{L}_S$ and $\mathcal{L}_{\bar{m}}$ to $\emptyset$.

Query. When $\mathcal{A}$ issues a message $m_i \in \mathcal{M}$ as the i-th query to the signing oracle **Sign**, $\mathcal{S}$ generates a signature $(c_i, \pi_i, r_{E,i}, r_{E2,i})$ on the message as follows.

- $r_{E,i}, r'_{E,i} \xleftarrow{U} \mathcal{R}_E$, $r'_{E2,i} \xleftarrow{U} \mathcal{R}_{E2}$, $m'_i \xleftarrow{U} \mathcal{M}$, $c'_i \xleftarrow{U} C$, $\pi'_i \xleftarrow{U} \mathcal{P}$, $\sigma'_i := (c'_i, \pi'_i)$.
- $h'_i := \mathsf{CHF.Eval}(pk_1, m'_i \| \sigma'_i; r'_{E,i})$, $\bar{m}_i := \mathsf{CHF2.Eval}(pk_2, h'_i; r'_{E2,i})$.
- $c_i := \mathsf{LPKE.Enc}(ek, sk_1, \bar{m}_i)$, $x_i := (c_i, \bar{m}_i)$, $w := sk_1$.
- Issues (x_i, w) as an query to O_{zk}, then receives a proof π_i.
- $\sigma_i := (c_i, \pi_i)$, $h_i := \mathsf{CHF.Eval}(pk_1, m_i \| \sigma_i; r_{E,i})$.

– $r_{E2,i} := \mathsf{CHF2.TC}(pk_2, sk_2, (h'_i, r'_{E2,i}), h_i)$.
$\mathcal{S}$ returns $(c_i, \pi_i, r_{E2,i}, r_{E,i})$ to $\mathcal{A}$. $\mathcal{L}_S$ is set to $\mathcal{L}_S \cup \{(m_i, c_i, \pi_i, r_{E2,i}, r_{E,i})\}$. $\mathcal{L}_{\bar{m}}$ is set to $\mathcal{L}_{\bar{m}} \cup \{\bar{m}_i\}$.

Forgery$(m^*, c^*, \pi^*, r^*_E, r^*_{E2})$. $\mathcal{S}$ checks whether the following condition is satisfied or not: $[1 \leftarrow \mathsf{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r^*_E, r^*_{E2}) \notin \mathcal{L}_S] \wedge [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk^*_1)] \wedge [1 \leftarrow \mathsf{CHF.SKVer2}(pk_1, sk^*_1, sk_1)] \wedge [[\bar{m}^* \notin \{\bar{m}_1, \cdots, \bar{m}_{q_s}\}] \vee [{}^{\exists}i \in [1, q_s]\ s.t.\ [\bar{m}^* = \bar{m}_i] \wedge [(h^*, r^*_{E2}) = (h_i, r_{E2,i})] \wedge [(m^*, c^*, \pi^*, r^*_E) \neq (m_i, c_i, \pi_i, r_{E,i})]]]$, where $h^* := \mathsf{CHF.Eval}(pk_1, m^* \| (c^*, \pi^*); r^*_E)$, $\bar{m}^* := \mathsf{CHF2.Eval}(pk_2, h^*; r^*_{E2})$, $x^* := (c^*, \bar{m}^*)$ and $sk^*_1 := \mathsf{LPKE.Dec}(dk, c^*, \bar{m}^*)$.
If the condition is satisfied, $\mathcal{S}$ outputs 1. Else, then $\mathcal{S}$ outputs 0.

It is obvious that if the common reference string crs is generated by $(crs, td) \leftarrow \mathcal{S}_1(1^\lambda)$ (resp. $crs \leftarrow \mathsf{NIZK.Gen}(1^\lambda)$) and the proof-generation oracle $\mathcal{O}_{zk}$ is $\mathcal{O}_1^{crs,td}$ (resp. $\mathcal{O}_0^{crs}$), then $\mathcal{S}$ simulates $\mathtt{Game}_6$ (resp. $\mathtt{Game}_5$) against $\mathcal{A}$ perfectly, and if and only if W_6 (resp. W_5) occurs, $\mathcal{S}$ outputs 1. Hence, we obtain

$$\begin{aligned}|\Pr[W_5] - \Pr[W_6]| = \Big|&\Pr\Big[1 \leftarrow \mathcal{S}^{\mathcal{O}_0^{crs}(x,w)}(crs) \mid crs \leftarrow \mathsf{NIZK.Gen}(1^\lambda)\Big] \\ &- \Pr\Big[1 \leftarrow \mathcal{S}^{\mathcal{O}_1^{crs,td}(x,w)}(crs) \mid (crs, td) \leftarrow \mathcal{S}_1(1^\lambda)\Big]\Big|.\end{aligned}$$

□

A.7 Proof of Lemma 7

We prove that if we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_6] - \Pr[W_7]|$ non-negligible, then we are able to construct a PPT algorithm which breaks the property of strong collision- resistance in HL model w.r.t. $\mathcal{F}^{Htl}_{\Sigma_{\mathrm{SIG}}}(\lambda)$ for Σ_{CHF}.

We consider a PPT simulator $\mathcal{S}$ who behaves as a PPT adversary trying to break the property of strong collision resistance in HL model w.r.t. $\mathcal{F}^{Htl}_{\Sigma_{\mathrm{SIG}}}(\lambda)$ for Σ_{CHF}. The concrete behaviour by $\mathcal{S}$ is the following.

Key-Generation. $\mathcal{S}$ is given $(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, pk_2, ek, crs, sk_1; r))$, where $(pk_1, sk_1) \xleftarrow{\mathrm{R}} \mathsf{CHF.Gen}(1^\lambda)$, $(pk_2, sk_2) \xleftarrow{\mathrm{R}} \mathsf{CHF2.Gen}(1^\lambda)$, $(ek, dk) \xleftarrow{\mathrm{R}} \mathsf{LPKE.Gen}(1^\lambda)$, $(crs, td) \xleftarrow{\mathrm{R}} \mathcal{S}_1(1^\lambda)$ and $r \xleftarrow{\mathrm{R}} \mathcal{R}$. $\mathcal{S}$ sets pk to (pk_1, pk_2, ek, crs). $\mathcal{S}$ sends the public key pk and the leakage $f(pk_1, pk_2, ek, crs, sk_1; r)$ to $\mathcal{A}$. $\mathcal{S}$ sets $\mathcal{L}_S$ and $\mathcal{L}_{\bar{m}}$ to $\emptyset$.

Query. When $\mathcal{A}$ issues a message $m_i \in \mathcal{M}$ as the i-th query to the signing oracle **Sign**, $\mathcal{S}$ generates a signature $(c_i, \pi_i, r_{E,i}, r_{E2,i})$ on the message as follows.
– $r_{E,i}, r'_{E,i} \xleftarrow{\mathrm{U}} \mathcal{R}_E$, $r'_{E2,i} \xleftarrow{\mathrm{U}} \mathcal{R}_{E2}$, $m'_i \xleftarrow{\mathrm{U}} \mathcal{M}$, $c'_i \xleftarrow{\mathrm{U}} \mathcal{C}$, $\pi'_i \xleftarrow{\mathrm{U}} \mathcal{P}$, $\sigma'_i := (c'_i, \pi'_i)$.
– $h'_i := \mathsf{CHF.Eval}(pk_1, m'_i \| \sigma'_i; r'_{E,i})$, $\bar{m}_i := \mathsf{CHF2.Eval}(pk_2, h'_i; r'_{E2,i})$.
– $c_i := \mathsf{LPKE.Enc}(ek, sk_1, \bar{m}_i)$, $x_i := (c_i, \bar{m}_i)$, $\pi_i := \mathcal{S}_2(crs, x_i, td)$.
– $\sigma_i := (c_i, \pi_i)$, $h_i := \mathsf{CHF.Eval}(pk_1, m_i \| \sigma_i; r_{E,i})$.
– $r_{E2,i} := \mathsf{CHF2.TC}(pk_2, sk_2, (h'_i, r'_{E2,i}), h_i)$.
$\mathcal{S}$ returns $(c_i, \pi_i, r_{E2,i}, r_{E,i})$ to $\mathcal{A}$. $\mathcal{L}_S$ is set to $\mathcal{L}_S \cup \{(m_i, c_i, \pi_i, r_{E2,i}, r_{E,i})\}$. $\mathcal{L}_{\bar{m}}$ is set to $\mathcal{L}_{\bar{m}} \cup \{\bar{m}_i\}$.

Forgery$(m^*, c^*, \pi^*, r_E^*, r_{E2}^*)$. $\mathcal{S}$ checks whether the following condition is satisfied or not: $[1 \leftarrow \text{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \wedge [1 \leftarrow \text{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \text{CHF.SKVer2}(pk_1, sk_1^*, sk_1)] \wedge [{}^{\exists}i \in [1, q_s]\ s.t.\ [\bar{m}^* = \bar{m}_i] \wedge [(h^*, r_{E2}^*) = (h_i, r_{E2,i})] \wedge [(m^*, c^*, \pi^*, r_E^*) \neq (m_i, c_i, \pi_i, r_{E,i})]]$, where $h^* := \text{CHF.Eval}(pk_1, m^* \| (c^*, \pi^*); r_E^*)$, $\bar{m}^* := \text{CHF2.Eval}(pk_2, h^*; r_{E2}^*)$, $x^* := (c^*, \bar{m}^*)$ and $sk_1^* := \text{LPKE.Dec}(dk, c^*, \bar{m}^*)$.
If the condition is satisfied, $\mathcal{S}$ outputs $((m^* \| (c^*, \pi^*), r_E^*), (m_i \| (c_i, \pi_i), r_{E,i}))$.

It is obvious that $\mathcal{S}$ simulates $\texttt{Game}_6$ or $\texttt{Game}_7$, perfectly. The definitions of W_6 and W_7 gives us the following equations.

$$\begin{aligned}
\Pr[W_6] = \Pr\,&[[1 \leftarrow \text{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\
&\wedge [1 \leftarrow \text{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \text{CHF.SKVer2}(pk_1, sk_1^*, sk_1)] \\
&\wedge \Big[[\bar{m}^* \notin \mathcal{L}_{\bar{m}}] \vee \Big[{}^{\exists}i \in [1, q_s]\ s.t.\ [\bar{m}^* = \bar{m}_i] \wedge [(h^*, r_{E2}^*) = (h_i, r_{E2,i})] \\
&\wedge [(m^*, c^*, \pi^*, r_E^*) \neq (m_i, c_i, \pi_i, r_{E,i})]]]]] \qquad (15)\\
\Pr[W_7] = \Pr\,&[[1 \leftarrow \text{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\
&\wedge [1 \leftarrow \text{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \text{CHF.SKVer2}(pk_1, sk_1^*, sk_1)] \\
&\wedge [\bar{m}^* \notin \mathcal{L}_{\bar{m}}]] \qquad (16)
\end{aligned}$$

By (15) and (16), we obtain

$$\begin{aligned}
&|\Pr[W_6] - \Pr[W_7]| \\
\leq\ &\Pr[[1 \leftarrow \text{NIZK.Ver}(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r_E^*, r_{E2}^*) \notin \mathcal{L}_S] \\
&\wedge [1 \leftarrow \text{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \text{CHF.SKVer2}(pk_1, sk_1^*, sk_1)] \\
&\wedge \Big[{}^{\exists}i \in [1, q_s]\ s.t.\ [\bar{m}^* = \bar{m}_i] \wedge [(h^*, r_{E2}^*) = (h_i, r_{E2,i})] \\
&\wedge [(m^*, c^*, \pi^*, r_E^*) \neq (m_i, c_i, \pi_i, r_{E,i})]]]
\end{aligned}$$

The last probability is equal to the probability with whom $\mathcal{S}$ wins the game for the strong collision-resistance in HL model w.r.t. the function class $\mathcal{F}_{\Sigma_{\text{SIG}}}^{HtI}(\lambda)$ for Π_{CHF}. $\square$

A.8 Proof of Lemma 8

We prove that for any $k \in [1, q_s]$ if we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_{7|k-1}] - \Pr[W_{7|k}]|$ non-negligible, then we are able to construct a PPT algorithm which breaks the IND-wLCCA security for Σ_{LPKE}.

We consider a PPT simulator $\mathcal{S}$ attempting to break the IND-wLCCA security for the LPKE scheme Σ_{LPKE}. $\mathcal{CH}$ denotes the challenger in the IND-wLCCA security game. Specifically, $\mathcal{S}$ behaves as follows.

Key-Generation. $\mathcal{S}$ is given an encryption-key ek of Σ_{LPKE}. $\mathcal{S}$ runs $(pk_1, sk_1) \leftarrow \text{CHF.Gen}(1^\lambda)$, $(pk_2, sk_2) \leftarrow \text{CHF2.Gen}(1^\lambda)$ and $(crs, td) \leftarrow \mathcal{S}_1(1^\lambda)$. $\mathcal{S}$ sets pk and sk to $pk := (pk_1, pk_2, ek, crs)$ and $sk := sk_1$, respectively. $\mathcal{S}$ computes $f(pk, sk; r)$, where $r \overset{\text{R}}{\leftarrow} \mathcal{R}$, then sends $(pk, f(pk, sk; r))$ to $\mathcal{A}$. $\mathcal{S}$ sets each list of $\mathcal{L}_S$ and $\mathcal{L}_{\bar{m}}$ to $\emptyset$.

Query. We consider the case when $\mathcal{A}$ issues a message $m_i \in \mathcal{M}$ as the i-th query to **Sign**. If $i \geq k+1$ (resp. $i \leq k-1$), then $\mathcal{S}$ generates a signature $(c_i, \pi_i, r_{E,i}, r_{E2,i})$ on m_i as follows.

- $r_{E,i}, r'_{E,i} \overset{U}{\leftarrow} \mathcal{R}_E$, $r'_{E2,i} \overset{U}{\leftarrow} \mathcal{R}_{E2}$, $m'_i \overset{U}{\leftarrow} \mathcal{M}$, $c'_i \overset{U}{\leftarrow} \mathcal{C}$, $\pi'_i \overset{U}{\leftarrow} \mathcal{P}$, $\sigma'_i := (c'_i, \pi'_i)$.
- $h'_i :=$ CHF.Eval$(pk_1, m'_i \| \sigma'_i; r'_{E,i})$, $\bar{m}_i :=$ CHF2.Eval$(pk_2, h'_i; r'_{E2,i})$.
- $c_i :=$ LPKE.Enc$(ek, sk_1, \bar{m}_i)$ (resp. $c_i :=$ LPKE.Enc$(ek, 0^{|sk_1|}, \bar{m}_i)$).
- $x_i := (c_i, \bar{m}_i)$, $\pi_i := \mathcal{S}_2(crs, x_i, td)$.
- $\sigma_i := (c_i, \pi_i)$, $h_i :=$ CHF.Eval$(pk_1, m_i \| \sigma_i; r_{E,i})$.
- $r_{E2,i} :=$ CHF2.TC$(pk_2, sk_2, (h'_i, r'_{E2,i}), h_i)$.

Else if $i = k$, then $\mathcal{S}$ generates a signature $(c_k, \pi_k, r_{E,k}, r_{E2,k})$ on m_k in the same manner as the case of $i \geq k+1$ or $i \leq k-1$ except that how the ciphertext $c_i = c_k$ is generated. In the case of $i = k$, $\mathcal{S}$ sends $(sk_1, 0^{|sk_1|}, \bar{m}_k)$ to $\mathcal{CH}$ in **Challenge** in IND-wLCCA game for Σ_{LPKE}, then gets a ciphertext c_k.

$\mathcal{S}$ returns the generated signature $(c_i, \pi_i, r_{E,i}, r_{E2,i})$ to $\mathcal{A}$. $\mathcal{S}$ sets $\mathcal{L}_S$ to $\mathcal{L}_S \cup \{(m_i, c_i, \pi_i, r_{E,i}, r_{E2,i})\}$. $\mathcal{S}$ sets $\mathcal{L}_{\bar{m}}$ to $\mathcal{L}_{\bar{m}} \cup \{\bar{m}_i\}$.

Forgery$(m^*, c^*, \pi^*, r^*_E, r^*_{E2})$. $\mathcal{S}$ computes $h^* :=$ CHF.Eval$(pk_1, m^* \| (c^*, \pi^*); r^*_E)$ and $\bar{m}^* :=$ CHF2.Eval$(pk_2, h^*; r^*_{E2})$. $\mathcal{S}$ issues $(c^*, \bar{m}^*)$ as a query to **Dec** in **Query 2** in IND-wLCCA game, then receives sk^*_1. $\mathcal{S}$ outputs $\beta' := 1$ in **Guess** in IND-wLCCA game, when the following condition is satisfied: $[1 \leftarrow$ NIZK.Ver$(crs, x^*, \pi^*)] \wedge [(m^*, c^*, \pi^*, r^*_E, r^*_{E2}) \notin \mathcal{L}_S] \wedge [1 \leftarrow$ CHF.SKVer$(pk_1, sk^*_1)] \wedge [1 \leftarrow$ CHF.SKVer2$(pk_1, sk^*_1, sk_1)] \wedge [\bar{m}^* \notin \mathcal{L}_{\bar{m}}]$, where $x^* := (c^*, \bar{m}^*)$.

Let $\beta \in \{0,1\}$ be the challenge-bit in the IND-wLCCA security game for Π_{LPKE}. It is obvious that $\mathcal{S}$ simulates $\mathtt{Game}_{7|k-1}$ (resp. $\mathtt{Game}_{7|k}$) when $\beta = 0$ (resp. $\beta = 1$), and if and only if $W_{7|k-1}$ (resp. $W_{7|k}$) happens, $\mathcal{S}$ outputs $\beta' = 1$. It is also obvious that when $W_{7|k-1}$ or $W_{7|k}$ occurs, the label $\bar{m}^*$ in the query $(c^*, \bar{m}^*)$ to the oracle **Dec** in **Query 2** issued by $\mathcal{S}$ satisfies $\bar{m}^* \neq \bar{m}_k$, so the query $(c^*, \bar{m}^*)$ is not a forbidden query. Hence, we obtain $\Pr[W_{7|k-1}] = \Pr[\beta' = 1 \mid \beta = 0]$ and $\Pr[W_{7|k}] = \Pr[\beta' = 1|\beta = 1]$.

It is obvious that $\Pr[\beta' = \beta] = \Pr[\beta' = 0 \wedge \beta = 0] + \Pr[\beta' = 1 \wedge \beta = 1] = \frac{1}{2}(\Pr[\beta' = 0|\beta = 0] + \Pr[\beta' = 1|\beta = 1]) = \frac{1}{2}(\Pr[\beta' = 1|\beta = 1] - \Pr[\beta' = 1|\beta = 0] + 1)$.

Hence, we obtain $\mathsf{Adv}^{IND\text{-}wLCCA}_{\Sigma_{\mathrm{LPKE}},\mathcal{S}} = |2 \cdot \Pr[\beta' = \beta] - 1| = |\Pr[\beta' = 1|\beta = 1] - \Pr[\beta' = 1|\beta = 0]| = |\Pr[W_{7|k-1}] - \Pr[W_{7|k}]|$. □

A.9 Proof of Lemma 9

We prove that if we assume that there is a PPT adversary $\mathcal{A}$ which makes $\Pr[W_{7|q_s}]$ non-negligible, then we are able to construct a PPT algorithm which breaks the property of hardness of inversion for the leakage-function $f \in \mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$.

We consider a PPT algorithm $\mathcal{S}$. $\mathcal{S}$ behaves as a PPT algorithm in the definition of the leakage-function class $\mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$. $\mathcal{S}$ is given $(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, pk_2, ek, crs, sk_1; r))$ as inputs, and simulates $\mathtt{Game}_{7|q_s}$ against $\mathcal{A}$. The concrete behaviour by $\mathcal{S}$ is the following.

Key-Generation. $\mathcal{S}$ is given $(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, pk_2, ek, crs, sk_1; r))$, where $(pk_1, sk_1) \overset{R}{\leftarrow}$ CHF.Gen(1^λ), $(pk_2, sk_2) \overset{R}{\leftarrow}$ CHF2.Gen(1^λ), $(ek, dk) \overset{R}{\leftarrow}$ LPKE.Gen(1^λ), $(crs, td) \overset{R}{\leftarrow} \mathcal{S}_1(1^\lambda)$ and $r \overset{R}{\leftarrow} \mathcal{R}$. $\mathcal{S}$ sets pk to (pk_1, pk_2, ek, crs). $\mathcal{S}$ sends $(pk, f(pk_1, pk_2, ek, crs, sk_1; r))$ to $\mathcal{A}$. $\mathcal{S}$ sets each list of $\mathcal{L}_S$ and $\mathcal{L}_{\bar{m}}$ to $\emptyset$.

Query. When $\mathcal{A}$ issues a message $m_i \in \mathcal{M}$ as the i-th query to the signing oracle **Sign**, $\mathcal{S}$ generates a signature $(c_i, \pi_i, r_{E,i}, r_{E2,i})$ on the message as follows.

- $r_{E,i}, r'_{E,i} \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_E$, $r'_{E2,i} \overset{\mathrm{U}}{\leftarrow} \mathcal{R}_{E2}$, $m'_i \overset{\mathrm{U}}{\leftarrow} \mathcal{M}$, $c'_i \overset{\mathrm{U}}{\leftarrow} C$, $\pi'_i \overset{\mathrm{U}}{\leftarrow} \mathcal{P}$, $\sigma'_i := (c'_i, \pi'_i)$.
- $h'_i :=$ CHF.Eval$(pk_1, m'_i \| \sigma'_i; r'_{E,i})$, $\bar{m}_i :=$ CHF2.Eval$(pk_2, h'_i; r'_{E2,i})$.
- $c_i :=$ LPKE.Enc$(ek, sk_1, \bar{m}_i)$, $x_i := (c_i, \bar{m}_i)$, $\pi_i := \mathcal{S}_2(crs, x_i, td)$.
- $\sigma_i := (c_i, \pi_i)$, $h_i :=$ CHF.Eval$(pk_1, m_i \| \sigma_i; r_{E,i})$.
- $r_{E2,i} :=$ CHF2.TC$(pk_2, sk_2, (h'_i, r'_{E2,i}), h_i)$.

$\mathcal{S}$ returns $(c_i, \pi_i, r_{E2,i}, r_{E,i})$ to $\mathcal{A}$. $\mathcal{L}_S$ is set to $\mathcal{L}_S \cup \{(m_i, c_i, \pi_i, r_{E2,i}, r_{E,i})\}$. $\mathcal{L}_{\bar{m}}$ is set to $\mathcal{L}_{\bar{m}} \cup \{\bar{m}_i\}$.

Forgery$(m^*, c^*, \pi^*, r^*_E, r^*_{E2})$. $\mathcal{S}$ sets $h^* :=$ CHF.Eval$(pk_1, m^* \| (c^*, \pi^*); r^*_E)$, $\bar{m}^* :=$ CHF2.Eval$(pk_2, h^*; r^*_{E2})$ and $sk^*_1 :=$ LPKE.Dec$(dk, c^*, \bar{m}^*)$. $\mathcal{S}$ outputs sk^*_1.

It is obvious that $\mathcal{S}$ simulates $\mathtt{Game}_{7|q_s}$ against $\mathcal{A}$ perfectly. If $\mathcal{A}$ wins the game $\mathtt{Game}_{7|q_s}$, then $\mathcal{S}$ is able to acquire a secret-key sk^*_1 such that $[1 \leftarrow \text{CHF.SKVer}(pk_1, sk^*_1)] \wedge [1 \leftarrow \text{CHF.SKVer2}(pk_1, sk^*_1, sk_1)]$. Hence, we obtain

$$\begin{aligned}&\Pr\left[\mathcal{S}\left(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, pk_2, ek, crs, sk_1; r)\right) \to sk^*_1\right.\\ &\quad\left.\text{s.t. } [1 \leftarrow \text{CHF.SKVer}\left(pk_1, sk^*_1\right)] \wedge [1 \leftarrow \text{CHF.SKVer2}\left(pk_1, sk^*_1, sk_1\right)]\right] = \Pr\left[W_{7|q_s}\right],\end{aligned}$$

where $(pk_1, sk_1) \overset{\mathrm{R}}{\leftarrow}$ CHF.Gen(1^λ), $(pk_2, sk_2) \overset{\mathrm{R}}{\leftarrow}$ CHF2.Gen(1^λ), $(ek, dk) \overset{\mathrm{R}}{\leftarrow}$ LPKE.Gen(1^λ), $(crs, td) \overset{\mathrm{R}}{\leftarrow} \mathcal{S}_1(1^\lambda)$ and $r \overset{\mathrm{R}}{\leftarrow} \mathcal{R}$.

If we assume that there exists a polynomial function $poly(\lambda)$ such that $\Pr\left[W_{7|q_s}\right] \geq 1/poly(\lambda)$, then we obtain

$$\begin{aligned}&\Pr\left[\mathcal{S}\left(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, pk_2, ek, crs, sk_1; r)\right) \to sk^*_1\right.\\ &\quad\left.\text{s.t. } [1 \leftarrow \text{CHF.SKVer}\left(pk_1, sk^*_1\right)] \wedge [1 \leftarrow \text{CHF.SKVer2}\left(pk_1, sk^*_1, sk_1\right)]\right] \geq 1/poly(\lambda),\end{aligned}$$

where $(pk_1, sk_1) \overset{\mathrm{R}}{\leftarrow}$ CHF.Gen(1^λ), $(pk_2, sk_2) \overset{\mathrm{R}}{\leftarrow}$ CHF2.Gen(1^λ), $(ek, dk) \overset{\mathrm{R}}{\leftarrow}$ LPKE.Gen(1^λ), $(crs, td) \overset{\mathrm{R}}{\leftarrow} \mathcal{S}_1(1^\lambda)$ and $r \overset{\mathrm{R}}{\leftarrow} \mathcal{R}$.

This contradicts to the hardness of inversion for the leakage-function $f \in \mathcal{F}^{HtI}_{\Sigma_{\mathrm{SIG}}}(\lambda)$. □

A.10 Proof of Theorem 2

We prove the theorem by proving that if we assume that there is a PPT adversary $\mathcal{A}$ which breaks the HtC-SK property for $\Pi_{\mathrm{CHF},n}$, then we are able to construct a PPT algorithm $\mathcal{S}$ which breaks the DL assumption. We prove the theorem for the case that $n \geq 2$. The theorem for the case that $n = 1$ can be proven in the same manner.

$\mathcal{S}$ is given $(p, \mathbb{G}, g, g^a)$ as an instance of the discrete logarithm problem. Then, $\mathcal{S}$ behaves as follows: Chooses $j \overset{\mathrm{U}}{\leftarrow} [1, n]$ and sets $g_j := g^a$. For every $i \in [1, n]$, chooses $x_i \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p$, then sets $\mathbf{x} := (x_1, \cdots, x_n)$. For every $i \in [1, n] \setminus \{j\}$, chooses $a_i \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p$, then sets $g_i := g^{a_i}$. Sets $y := \prod_{i \in [1,n]} g_i^{x_i}$.

$\mathcal{S}$ gives $pk := (p, \mathbb{G}, g, g_1, \cdots, g_n, y)$ to $\mathcal{A}$. After that, $\mathcal{S}$ receives $\mathbf{x}^* := (x^*_1, \cdots, x^*_n) \in \mathbb{Z}_p^n$ sent by $\mathcal{A}$. Since we are considering a PPT adversary $\mathcal{A}$ breaking the HtC-SK property for $\Pi_{\mathrm{CHF},n}$, $\mathbf{x}^*$ satisfies the relation $[\mathbf{x}^* \neq \mathbf{x}] \wedge [\prod_{i \in [1,n]} g_i^{x^*_i} = \prod_{i \in [1,n]} g_i^{x_i}]$.

Hence, we obtain $g_j^{x_j^*-x_j} = \prod_{i\in[1,n]\setminus\{j\}} g_i^{x_i-x_i^*}$ and $g_j = g^{(\sum_{i\in[1,n]\setminus\{j\}} a_i(x_i-x_i^*))/(x_j^*-x_j)} = g^a$.

$\mathcal{S}$ outputs $a = (\sum_{i\in[1,n]\setminus\{j\}} a_i(x_i - x_i^*))/(x_j^* - x_j)$. Note that $\Pr[x_j^* \neq x_j] \geq 2/n$, where $n \geq 2$. □

A.11 Proof of Theorem 3

We consider a key-pair (pk, sk), a message $m \in \mathcal{M}$, and a message $m' \in \mathcal{M}$. pk and sk are parsed as $pk = (p, \mathbb{G}, g_1, \cdots, g_n, y)$ and $sk = (x_1, \cdots, x_n)$, respectively.

$P(\hat{\mathbf{r}})$ denotes the probability that the randomness $\hat{\mathbf{r}}$ is chosen as the randomness used to compute the hash value for the message m. Hence, $P(\hat{\mathbf{r}}) = \Pr[\hat{\mathbf{r}} = \mathbf{r} \mid \mathbf{r} \overset{U}{\leftarrow} \mathbb{Z}_p^n] = \Pr[\bigwedge_{i\in[1,n]}[\hat{r}_i = r_i] \mid \mathbf{r} \overset{U}{\leftarrow} \mathbb{Z}_p^n] = \prod_{i\in[1,n]} \Pr[\hat{r}_i = r_i \mid r_i \overset{U}{\leftarrow} \mathbb{Z}_p] = 1/p^n$.

$P(\hat{\mathbf{r}'})$ denotes the probability that CHF.TC$(pk, sk, (m, \mathbf{r}), m')$ outputs $\hat{\mathbf{r}'} \in \mathbb{Z}_p^n$, where the randomness $\mathbf{r} \in \mathbb{Z}_p^n$ is chosen uniformly at random. Hence,

$P(\hat{\mathbf{r}'}) = \Pr[\hat{\mathbf{r}'} = \text{CHF.TC}(pk_1, sk_1, (m, \mathbf{r}), m') \mid \mathbf{r} \overset{U}{\leftarrow} \mathbb{Z}_p^n] = \Pr[\bigwedge_{i\in[1,n]}[\hat{r}'_i = J(m)(x_i - r_i)/J(m') - x_i] \mid \mathbf{r} \overset{U}{\leftarrow} \mathbb{Z}_p^n] = \prod_{i\in[1,n]} \Pr[\hat{r}'_i = J(m)(x_i - r_i)/J(m') - x_i \mid r_i \overset{U}{\leftarrow} \mathbb{Z}_p] = 1/p^n$.

Hence, $P(\hat{\mathbf{r}}) = P(\hat{\mathbf{r}'}) = 1/p^n$. Therefore, for any (pk, sk) and any $m, m' \in \mathcal{M}$, $\mathbf{r} \overset{U}{\leftarrow} \mathcal{R}$ and $\mathbf{r}' :=$ CHF.Eval$(pk, sk, (m, \mathbf{r}), m')$ distribute identically. □

A.12 Proof of Theorem 4

We prove the theorem by the argument of game-transformation. We use four games $\mathtt{Game}_0$, $\mathtt{Game}_1$, $\mathtt{Game}_2$ and $\mathtt{Game}_3$. Each game is defined as follows.

$\mathtt{Game}_0$. $\mathtt{Game}_0$ is the game of strong collision-resistance for the CHF scheme $\Pi_{\mathrm{CHF},n}$ in the auxiliary leakage model w.r.t. the function class $\mathcal{F}_{\Pi_{\mathrm{SIG}}}^{HtI}(\lambda)$. Concretely, the game is the following.

$\mathcal{CH}$ runs $(pk_1, sk_1) \leftarrow$ CHF.Gen(1^λ), $(pk_2, sk_2) \leftarrow$ CHF2.Gen(1^λ), $(ek, dk) \leftarrow$ LPKE.Gen(1^λ) and $(crs, td) \leftarrow \mathcal{S}_1(1^\lambda)$. pk_1 is parsed as $(p, \mathbb{G}, g_1, \cdots, g_n, y)$. sk_1 is parsed as $(x_1, \cdots, x_n)$. pk is set as (pk_1, pk_2, ek, crs). sk is set as sk_1. $\mathcal{CH}$ sets r as $r \overset{R}{\leftarrow} \mathcal{R}$, where $\mathcal{R}$ is the randomness space of a leakage function $f \in \mathcal{F}_{\Pi_{\mathrm{SIG}}}^{HtI}(\lambda)$, then computes $f(pk, sk; r)$. $\mathcal{CH}$ sends $(pk, sk_2, dk, td, f(pk, sk; r))$ to $\mathcal{A}$. Then, $(m, \mathbf{r})$ and $(m', \mathbf{r}')$, where $m, m' \in \{0, 1\}^*$ and $\mathbf{r}, \mathbf{r}' \in \mathbb{Z}_p^n$, are sent to $\mathcal{CH}$ by $\mathcal{A}$. $\mathbf{r}$ and $\mathbf{r}'$ are parsed as $(r_1, \cdots, r_n)$ and $(r'_1, \cdots, r'_n)$, respectively. $\mathcal{A}$ is said to win the game, if the following condition is satisfied:

$$[[m \neq m'] \vee [[m = m'] \wedge [\mathbf{r} \neq \mathbf{r}']]] \wedge \left[\left(y \cdot \prod_{i=1}^{n} g_i^{r_i}\right)^{J(m)} = \left(y \cdot \prod_{i=1}^{n} g_i^{r'_i}\right)^{J(m')}\right].$$

$\mathtt{Game}_1$. $\mathtt{Game}_1$ is the same as $\mathtt{Game}_0$ except that the winning condition by $\mathcal{A}$ is changed to the following one: $[[[m \neq m'] \wedge [J(m) \neq J(m')] \wedge [[\mathbf{x} \neq \mathbf{x}^*] \vee [\mathbf{x} = \mathbf{x}^*]]] \vee [[m = m'] \wedge [\mathbf{r} \neq \mathbf{r}']]] \wedge [(y \cdot \prod_{i=1}^{n} g_i^{r_i})^{J(m)} = (y \cdot \prod_{i=1}^{n} g_i^{r'_i})^{J(m')}]$, where, for $i \in [1, n]$, $x_i^* := (J(m)r_i - J(m')r'_i)/(J(m') - J(m))$, $\mathbf{x}^* := (x_1^*, \cdots, x_n^*)$ and $\mathbf{x} := (x_1, \cdots, x_n)$.

Game$_2$. Game$_2$ is the same as Game$_1$ except that the winning condition by $\mathcal{A}$ is changed to the following one: $[[[m \neq m'] \wedge [J(m) \neq J(m')] \wedge [\mathbf{x} = \mathbf{x}^*]] \vee [[m = m'] \wedge [\mathbf{r} \neq \mathbf{r}']]] \wedge [(y \cdot \prod_{i=1}^{n} g_i^{r_i})^{J(m)} = (y \cdot \prod_{i=1}^{n} g_i^{r'_i})^{J(m')}]$.

Game$_3$. Game$_3$ is the same as Game$_2$ except that the winning condition by $\mathcal{A}$ is changed to the following one: $[m = m'] \wedge [\mathbf{r} \neq \mathbf{r}'] \wedge [(y \cdot \prod_{i=1}^{n} g_i^{r_i})^{J(m)} = (y \cdot \prod_{i=1}^{n} g_i^{r'_i})^{J(m')}]$.

W_i, where $i \in \{0, 1, 2, 3\}$, denotes the event that $\mathcal{A}$ wins the game Game$_i$. It holds that

$$\Pr[W_0] \leq |\Pr[W_0] - \Pr[W_1]| + |\Pr[W_1] - \Pr[W_2]| + |\Pr[W_2] - \Pr[W_3]| .$$

By the above inequality and the following lemmas, Theorem 4 is proven.

Lemma 10. $|\Pr[W_0] - \Pr[W_1]|$ *is negligible, if* $J : \{0,1\}^* \to \mathbb{Z}_p \setminus \{0\}$ *is a collision-resistant hash function.*

Lemma 11. $|\Pr[W_1] - \Pr[W_2]|$ *is negligible, if the discrete logarithm assumption holds.*

Lemma 12. $|\Pr[W_2] - \Pr[W_3]|$ *is negligible under the hard-to-invert property of the function* $f \in \mathcal{F}_{\Pi_{\mathrm{SIG}}}^{HtI}(\lambda)$.

Lemma 13. $\Pr[W_3]$ *is negligible, if the discrete logarithm assumption holds.*

□

Proof of Lemma 10. We prove that if there is a PPT $\mathcal{A}$ which makes $|\Pr[W_0] - \Pr[W_1]|$ non-negligible, we can construct a PPT $\mathcal{S}$ which breaks the collision-resistance of the hash function $J : \{0,1\}^* \to \mathbb{Z}_p \setminus \{0\}$. Let us consider a PPT $\mathcal{S}$ which behaves as follows.

$\mathcal{S}$ randomly generates (pk_1, sk_1), (pk_2, sk_2), (ek, dk) and (crs, td). pk_1 and sk_1 are parsed as $(p, \mathbb{G}, g_1, \cdots, g_n, y)$ and $(x_1, \cdots, x_n)$, respectively. pk and sk are set as (pk_1, pk_2, ek, crs) and sk_1, respectively. $\mathcal{S}$ sets r as $r \overset{\mathrm{R}}{\leftarrow} \mathcal{R}$, where $\mathcal{R}$ is the randomness space of a leakage function $f \in \mathcal{F}_{\Pi_{\mathrm{SIG}}}^{HtI}(\lambda)$, then computes $f(pk, sk; r)$. $\mathcal{S}$ sends $(pk, sk_2, dk, td, f(pk, sk; r))$ to $\mathcal{A}$. Then, $\mathcal{S}$ receives $(m, \mathbf{r})$ and $(m', \mathbf{r}')$ from $\mathcal{A}$. If $[m \neq m'] \wedge [J(m) = J(m')] \wedge [(y \cdot \prod_{i=1}^{n} g_i^{r_i})^{J(m)} = (y \cdot \prod_{i=1}^{n} g_i^{r'_i})^{J(m')}]$, then $\mathcal{S}$ outputs (m, m'). We obtain

$$\begin{aligned}|\Pr[W_0] - \Pr[W_1]| &\leq \Pr[[m \neq m'] \wedge [J(m) = J(m')] \wedge [(y \cdot \prod_{i=1}^{n} g_i^{r_i})^{J(m)} = (y \cdot \prod_{i=1}^{n} g_i^{r'_i})^{J(m')}]] \\ &= \Pr[\mathcal{S}(\cdot) \to (m, m') \ s.t.\ [m \neq m'] \wedge [J(m) = J(m')]].\end{aligned}$$

If we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_0] - \Pr[W_1]|$ non-negligible, $\mathcal{S}$ is able to break the collision-resistance property of the hash function J.

□

Proof of Lemma 11. We prove that if there is a PPT $\mathcal{A}$ which makes $|\Pr[W_1] - \Pr[W_2]|$ non-negligible, a PPT $\mathcal{S}$ which breaks the HtC-SK property for $\Pi_{\mathrm{CHF},n}$ can be constructed. Let us consider a PPT $\mathcal{S}$ which behaves as follows.

$\mathcal{S}$ is given the keys (pk_1, sk_1) of $\Pi_{\mathrm{CHF},n}$. pk_1 and sk_1 are parsed as $(p, \mathbb{G}, g_1, \cdots, g_n, y)$ and $(x_1, \cdots, x_n)$, respectively. $\mathcal{S}$ randomly generates (pk_2, sk_2), (ek, dk) and (crs, td). pk

and sk are set as (pk_1, pk_2, ek, crs) and sk_1, respectively. $\mathcal{S}$ sets r as $r \overset{\mathrm{R}}{\leftarrow} \mathcal{R}$, then computes $f(pk, sk; r)$. $\mathcal{S}$ sends $(pk, sk_2, dk, td, f(pk, sk; r))$ to $\mathcal{A}$. Then, $\mathcal{S}$ receives $(m, \mathbf{r})$ and $(m', \mathbf{r}')$ from $\mathcal{A}$. $\mathcal{S}$ computes $x_i^* := (J(m) \cdot r_i - J(m') \cdot r'_i)/(J(m') - J(m))$ for $i \in [1, n]$ and sets $\mathbf{x}^* := (x_1^*, \cdots, x_n^*)$ and $\mathbf{x} := (x_1, \cdots, x_n)$. If $[m \neq m'] \wedge [J(m) \neq J(m')] \wedge [\mathbf{x} \neq \mathbf{x}^*] \wedge [(y \cdot \prod_{i=1}^{n} g_i^{r_i})^{J(m)} = (y \cdot \prod_{i=1}^{n} g_i^{r'_i})^{J(m')}]$, then $\mathcal{S}$ outputs $\mathbf{x}^*$.

We obtain $\prod_{i\in[1,n]} g_i^{J(m)\cdot(x_i+r_i)} = \prod_{i\in[1,n]} g_i^{J(m')\cdot(x_i+r'_i)}$, $g^{\sum_{i\in[1,n]} a_i \cdot J(m)\cdot(x_i+r_i)} = g^{\sum_{i\in[1,n]} a_i \cdot J(m')\cdot(x_i+r'_i)}$, and $g^{\sum_{i\in[1,n]} a_i \cdot x_i} = g^{\sum_{i\in[1,n]} \frac{J(m')\cdot r'_i - J(m)\cdot r_i}{J(m)-J(m')}}$. In the transition from the first equation to the second one, we used the fact that for every $g_i \in \mathbb{G}$ where $i \in [1, n]$, there exists an integer $a_i \in \mathbb{Z}_p$ such that $g_i = g^{a_i}$.

Likewise, we obtain $\prod_{i\in[1,n]} g_i^{J(m)\cdot(x_i^*+r_i)} = \prod_{i\in[1,n]} g_i^{J(m')\cdot(x_i^*+r'_i)}$, $g^{\sum_{i\in[1,n]} a_i \cdot J(m)\cdot(x_i^*+r_i)} = g^{\sum_{i\in[1,n]} a_i \cdot J(m')\cdot(x_i^*+r'_i)}$, and $g^{\sum_{i\in[1,n]} a_i \cdot x_i^*} = g^{\sum_{i\in[1,n]} \frac{J(m')\cdot r'_i - J(m)\cdot r_i}{J(m)-J(m')}}$.

Hence, we obtain $g^{\sum_{i\in[1,n]} a_i \cdot x_i} = g^{\sum_{i\in[1,n]} a_i \cdot x_i^*}$, and $y = \prod_{i\in[1,n]} g_i^{x_i} = \prod_{i\in[1,n]} g_i^{x_i^*}$.

As a result, we obtain

$$
\begin{aligned}
&|\Pr[W_1] - \Pr[W_2]| \\
&\leq \Pr\left[[m \neq m'] \wedge [J(m) \neq J(m')] \wedge [\mathbf{x} \neq \mathbf{x}^*] \wedge \left[(y \cdot \prod_{i\in[1,n]} g_i^{r_i})^{J(m)} = (y \cdot \prod_{i\in[1,n]} g_i^{r'_i})^{J(m')}\right]\right] \\
&= \Pr\left[\mathcal{S}(p, \mathbb{G}, g_1, \cdots, g_n, y, \mathbf{x}) \rightarrow \mathbf{x}^* \ s.t. \ \left[y = \prod_{i\in[1,n]} g_i^{x_i^*}\right] \wedge [\mathbf{x} \neq \mathbf{x}^*]\right].
\end{aligned}
$$

If we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_1] - \Pr[W_2]|$ non-negligible, $\mathcal{S}$ is able to break the property of HtC-SK for $\Pi_{\mathrm{CHF},n}$. We have already proven Theorem 2 which says that the property of HtC-SK for $\Pi_{\mathrm{CHF},n}$ can be proven under the discrete logarithm assumption. Hence, $|\Pr[W_1] - \Pr[W_2]|$ is negligible under the DL assumption. □

Proof of Lemma 12. Let f be a leakage function $f \in \mathcal{F}_{\Pi_{\mathrm{SIG}}}^{HtI}(\lambda)$. We prove that if there is a PPT $\mathcal{A}$ which makes $|\Pr[W_2] - \Pr[W_3]|$ non-negligible, we can construct a PPT $\mathcal{S}$ which breaks the hardness of inversion for the function. Let us consider a PPT $\mathcal{S}$ which behaves as follows.

$\mathcal{S}$ is given $(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, pk_2, ek, crs, sk_1; r))$, where (pk_1, sk_1), (pk_2, sk_2), (ek, dk) and (crs, td) are randomly generated and $r \in \mathcal{R}$ is randomly chosen. pk_1 and sk_1 are parsed as $(p, \mathbb{G}, g_1, \cdots, g_n, y)$ and $(x_1, \cdots, x_n)$, respectively. $\mathcal{S}$ sends $(pk_1, pk_2, ek, crs, f(pk_1, pk_2, ek, crs, sk_1; r))$ to $\mathcal{A}$. Then, $\mathcal{S}$ receives $(m, \mathbf{r})$ and $(m', \mathbf{r}')$ from $\mathcal{A}$. $\mathcal{S}$ computes $x_i^* := (J(m)r_i - J(m')r'_i)/(J(m') - J(m))$ for $i \in [1, n]$ and sets $sk_1^* := (x_1^*, \cdots, x_n^*)$. $\mathcal{S}$ outputs sk_1^*. We obtain

$$
\begin{aligned}
&|\Pr[W_2] - \Pr[W_3]| \\
&\leq \Pr[[m \neq m'] \wedge [J(m) \neq J(m')] \wedge [sk_1 = sk_1^*] \wedge [(y \cdot \prod_{i\in[1,n]} g_i^{r_i})^{J(m)} = (y \cdot \prod_{i\in[1,n]} g_i^{r'_i})^{J(m')}]] \\
&= \Pr[\mathcal{S}(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, sk_1; r)) \rightarrow sk_1^* \ s.t. \ [sk_1^* = sk_1]] \\
&= \Pr[\mathcal{S}(pk_1, pk_2, ek, crs, sk_2, dk, td, f(pk_1, sk_1; r)) \rightarrow sk_1^* \\
&\quad s.t. \ [1 \leftarrow \mathsf{CHF.SKVer}(pk_1, sk_1^*)] \wedge [1 \leftarrow \mathsf{CHF.SKVer2}(pk_1, sk_1^*, sk_1)].
\end{aligned}
$$

If we assume that there is a PPT adversary $\mathcal{A}$ which makes $|\Pr[W_2] - \Pr[W_3]|$ non-negligible, $\mathcal{S}$ breaks the hardness of inversion of the function $f \in \mathcal{F}^{HtI}_{\Pi_{\mathrm{SIG}}}(\lambda)$. $\square$

Proof of Lemma 13. We prove that if there is a PPT $\mathcal{A}$ which makes $\Pr[W_4]$ non-negligible, we can construct a PPT $\mathcal{S}$ which breaks the n-representation assumption [3, 6]. We give the definition of the assumption below.

Definition 10. *We say that n-representation assumption holds if for every PPT* $\mathcal{A}$, $\Pr[\mathcal{A}(p, \mathbb{G}, g_1, \cdots, g_n) \to ((x_1, \cdots, x_n), (x'_1, \cdots, x'_n))$ *s.t.* $[(x_1, \cdots, x_n) \neq (x'_1, \cdots, x'_n)] \wedge [\prod_{i\in[1,n]} g_i^{x_i} = \prod_{i\in[1,n]} g_i^{x'_i}]]$ *is negligible, where* $(p, \mathbb{G}) \overset{\mathrm{R}}{\leftarrow} \mathcal{G}(1^\lambda)$, $g_1, \cdots, g_n \overset{\mathrm{U}}{\leftarrow} \mathbb{G}$ *and* $x_1, \cdots, x_n \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p$.

Validity of the assumption is guaranteed by the following theorem [3, 6].

Theorem 6. *n-representation assumption holds under the DL assumption.*

Let us consider a PPT $\mathcal{S}$ which behaves as follows. $\mathcal{S}$ is given $(p, \mathbb{G}, g_1, \cdots, g_n)$ as an instance of the n-representation problem. $\mathcal{S}$ chooses $x_1, \cdots, x_n \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p$, and sets $y := \prod_{i\in[1,n]} g_i^{x_i}$. Then, $\mathcal{S}$ sets pk_1 and sk_1 as $(p, \mathbb{G}, g_1, \cdots, g_n, y)$ and $(x_1, \cdots, x_n)$, respectively. $\mathcal{S}$ randomly generates (pk_2, sk_2), (ek, dk) and (crs, td). pk and sk are set as (pk_1, pk_2, ek, crs) and sk_1, respectively. $\mathcal{S}$ sets r as $r \overset{\mathrm{R}}{\leftarrow} \mathcal{R}$, then computes $f(pk, sk; r)$. $\mathcal{S}$ sends $(pk, sk_2, dk, td, f(pk, sk; r))$ to $\mathcal{A}$. Then, $\mathcal{S}$ receives $(m, \mathbf{r})$ and $(m', \mathbf{r}')$ from $\mathcal{A}$. If $[m = m'] \wedge [\mathbf{r} \neq \mathbf{r}'] \wedge [(y \cdot \prod_{i=1}^n g_i^{r_i})^{J(m)} = (y \cdot \prod_{i=1}^n g_i^{r'_i})^{J(m')}]$, then $\mathcal{S}$ outputs $(\mathbf{r}, \mathbf{r}')$. We obtain $\left(y \cdot \prod_{i\in[1,n]} g_i^{r_i}\right)^{J(m)} = \left(y \cdot \prod_{i\in[1,n]} g_i^{r'_i}\right)^{J(m')} = \left(y \cdot \prod_{i\in[1,n]} g_i^{r'_i}\right)^{J(m)}$. Hence, $\prod_{i\in[1,n]} g_i^{r_i} = \prod_{i\in[1,n]} g_i^{r'_i}$. Theorefore, we obtain

$$\begin{aligned}\Pr[W_4] &= \Pr\left[[m = m'] \wedge [\mathbf{r} \neq \mathbf{r}'] \wedge \left[\left(y \cdot \prod_{i\in[1,n]} g_i^{r_i}\right)^{J(m)} = \left(y \cdot \prod_{i\in[1,n]} g_i^{r'_i}\right)^{J(m')}\right]\right] \\ &= \Pr\left[\mathcal{S}(p, \mathbb{G}, g_1, \cdots, g_n) \to (\mathbf{r}, \mathbf{r}') \text{ s.t. } [\mathbf{r} \neq \mathbf{r}'] \wedge \left[\prod_{i\in[1,n]} g_i^{r_i} = \prod_{i\in[1,n]} g_i^{r'_i}\right]\right].\end{aligned}$$

If we assume that there is a PPT adversary $\mathcal{A}$ which makes $\Pr[W_4]$ non-negligible, $\mathcal{S}$ is able to break the n-representation assumption. $\square$