Homomorphic Signatures for Subset and Superset Mixed Predicates and Its Applications

Masahito Ishizaka⋆ and Kazuhide Fukushima

KDDI Research, Inc., Saitama, Japan.
{xma-ishizaka, ka-fukushima}@kddi.com

Abstract. In homomorphic signatures for subset predicates (HSSB), each message (to be signed) is a set. Any signature on a set M allows us to derive a signature on any subset $M' \subseteq M$. Its superset version, which should be called homomorphic signatures for superset predicates (HSSP), allows us to derive a signature on any superset $M' \supseteq M$. In this paper, we propose homomorphic signatures for subset and superset mixed predicates (HSSM) as a simple combination of HSSB and HSSP. In HSSM, any signature on a message of a set-pair (M, W) allows us to derive a signature on any (M', W') such that $M' \subseteq M$ and $W' \supseteq W$. We propose an original HSSM scheme which is unforgeable under the decisional linear assumption and completely context-hiding. We show that HSSM has various applications, which include disclosure-controllable HSSB, disclosure-controllable redactable signatures, (key-delegatable) superset/subset predicate signatures, and wildcarded identity-based signatures.

Keywords: Homomorphic signatures for subset and superset mixed predicates, Unforgeablity, Complete context-hiding, Decisional linear assumption.

1 Introduction

$\mathcal{P}$-Homomorphic Signatures ($\mathcal{P}$-HS) [5]. In ordinary digital signatures, if a signed massage is partially altered, its signature immediately turns invalid. In $\mathcal{P}$-HS for a predicate $\mathcal{P} : \mathcal{M} \times \mathcal{M} \rightarrow 1/0$, any signature on any message M allows any user to derive a signature on any message M' satisfying the predicate, i.e., $1 \leftarrow \mathcal{P}(M, M')$. Strong context-hiding (SCH) [5] is a strong privacy (or unlinkability) related security notion, which guarantees that any signature derived from any signature which has been honestly generated distributes as a fresh signature directly generated by the secret-key. Complete context-hiding (CCH) [6] is a stronger notion, which guarantees that any signature derived from any *valid* signature (which might have been *dishonestly* generated) distributes as a signature generated by the secret-key. Redactable signatures (RS) [23] is a subclass of $\mathcal{P}$-HS. We can partially redact, i.e., black-out, a signed message while retaining validity of the signature. In append-only signatures (AOS) [18],

⋆ Corresponding Author

we can repeatedly add any message to the tail of a signed message. In quotable signatures [5,7], we can extract any substring from a signed message.

HS for Subset Predicates (HSSB) [5,6]. HSSB is also a subclass of $\mathcal{P}$-HS. Each message is a set. Any signature on a set M derives a signature on any subset $M' \subseteq M$. Ahn et al. [5] proposed a generic SCH-secure transformation into HSSB from attribute-based encryption (ABE) [8] satisfying a property that any secret-key for an attribute-set A allows us to derive a perfectly re-randomized secret-key for any subset $A' \subseteq A$. Attrapadung et al. [6] proposed an HSSB scheme which is CCH-secure and unforgeable under the decisional linear (DLIN) and q-simultaneous flexible paring (SFP) assumptions w.r.t. a symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, built by the Groth-Sahai non-interactive witness indistinguishable (GS NIWI) proof [15], Waters signatures [24] and Abe-Haralambiev-Ohkubo structure-preserving signatures (AHO SPS) [4,3]. In key-generation, a signer generates a long-term AHO key-pair. When signing a message M, the signer generates a one-time Waters key-pair (g^x, x), where x is chosen uniformly at random from $\mathbb{Z}_p$, i.e., $x \overset{\text{U}}{\longleftarrow} \mathbb{Z}_p$, and g is a generator of $\mathbb{G}$, then generates an AHO signature on the Waters public-key g^x. The signer generates a Waters signature on every $m \in M$, then generates GS proofs that the AHO signature is valid and every Waters signature on $m \in M$ is valid.

HS for Superset Predicates (HSSP) [21]. HSSP is the superset counterpart of HSSB. It is called history-hiding append-only signatures (H2AOS) in [21]. Any signature on a set M derives a signature on any superset $M' \supseteq M$. Libert et al. [21] proposed a CCH-secure HSSP scheme based on the similar technique to the Attrapadung et al.'s HSSB scheme. It is built by an arbitrary SPS scheme satisfying unforgeability against extended random messages attacks [2], GS NIWI proof [15], and Boneh-Lynn-Shacham (BLS) signatures [11] instantiated by the Waters programmable hash function $H_{\mathbb{G}}$ [24]. In key-generation, a signer generates a long-term key-pair of the SPS scheme. When the signer signs a message W, she generates a fresh one-time BLS key-pair (Y, y), where $Y := g^y$ with $y \overset{\text{U}}{\longleftarrow} \mathbb{Z}_p$. The signer divides the BLS secret-key y into $|W|$ number of shares by using $|W|$-out-of-$|W|$ secret sharing. Specifically, she randomly chooses $\{\gamma_w \in \mathbb{Z}_p\}_{w \in W}$ s.t. $\sum_{w \in W} \gamma_w = y \pmod p$. For each $w \in W$, she computes $(\sigma_{w,1}, \sigma_{w,2}) := (H_{\mathbb{G}}(w)^{\gamma_w}, g^{\gamma_w}) \in \mathbb{G}^2$, where $\sigma_{w,1}$ is a BLS signature on w under the *pseudo* BLS public-key $\sigma_{w,2}$. The signer also generates a SPS signature $(\theta_1, \cdots, \theta_{l_{sps}}) \in \mathbb{G}^{l_{sps}}$ on $Y \in \mathbb{G}$. The signer finally generates a GS proof that (1) *the SPS signature on Y is valid*, (2) *the BLS signature $\sigma_{w,1}$ on w under the public-key $\sigma_{w,2}$ is valid for each $w \in W$* and (3) $\prod_{w \in W} \sigma_{w,2} = Y$.

1.1 Our Contributions

HS for Subset and Superset Mixed Predicates (HSSM). We formally define HSSM as a simple combination of HSSB and HSSP. Any signature on a message of a set-pair (M, W) allows us to derive a signature on any (M', W') such that $M' \subseteq M$

and $W' \supseteq W$. As $\mathcal{P}$-HS [5,6], we define unforgeability and strong/complete context-hiding (SCH/CCH). Unforgeability is a security notion required for any $\mathcal{P}$-HS. We emphasize the importance of SCH and CCH by using the following example similar to an example in [12] used to emphasize the importance of unlinkability for sanitizable signatures. Given an honestly-generated signature σ on a set-pair (M, W), we honestly derive signatures σ_1 and σ_2 on (M_1, W_1) and (M_2, W_2), respectively, then open only the derived signatures to the public. If the HSSM scheme is SCH, both σ_1 and σ_2 are independent of σ and the link between σ_1 and σ_2 is unseen. Otherwise, it is possible that the link is seen. In this case, the following two types of private information are leaked, (1) *M contains at least $M_1 \cup M_2$ as a subset*, and (2) *every element in $W_1 \cap W_2$ is older data than any element in $W_1 \setminus (W_1 \cap W_2)$ and in $W_2 \setminus (W_1 \cap W_2)$*. If the HSSM scheme is CCH, even when the signatures σ, σ_1 and σ_2 have been dishonestly generated, the same security is guaranteed. Thus, CCH is more desirable. If the number of the derived signatures is $n > 2$, a non-SCH HSSB scheme can cause a more serious problem, e.g., we can guess that M contains at least $\bigcup_{i=1}^{n} M_i$.

Our HSSM Scheme. Our HSSM scheme is a simple combination of the Attrapadung et al.'s HSSB scheme [6] and the Libert et al.'s HSSP schemes [21]. As the underlying SPS scheme, we use the AHO SPS scheme [4,3] with message space $\mathbb{G}^2$. On signature generation, a fresh one-time Waters key-pair (X, x) and BLS key-pair (Y, y) are generated, where $X := g^x$ and $Y := g^y$ with $x, y \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p$. An AHO signature $(\theta_1, \cdots, \theta_7)$ on $(X, Y) \in \mathbb{G}^2$ is generated. As the HSSB scheme [6], for each $m \in M$, a Waters signature $(\sigma_{m,1}, \sigma_{m,2})$ on m is generated. As the HSSP scheme [21], the original BLS secret-key $y \in \mathbb{Z}_p$ is divided into $|W|$ number of shares $\{\gamma_w \in \mathbb{Z}_p\}_{w \in W}$ by $|W|$-out-of-$|W|$ secret sharing, then for each $w \in W$, a BLS signature $\sigma_{w,1}$ on w under the pseudo BLS public-key $g^{\gamma_w} (=: \sigma_{w,2})$ is generated. Finally, the GS proof is properly generated.

We show that HSSM has following five applications.

Application 1: Disclosure-Controllablwe (DC) HSSB. In the ordinary HSSB [5,6], any sub-message $m \in M$ can be deleted anytime. For some realistic applications of HSSB, there might be a case where we would like to make some sub-messages undeletable. For instance, HSSB can be used to prove one's credentials. A person named Alice is given an HSSM signature on a message M including her identifiable information (e.g., name, her birth date) and all of her credentials. When she proves that she has all of the required credentials to an organization, she might want to hide all of the unrequired credentials. In this application, her identifiable information is usually not deleted in any situation. Even in the ordinary HSSB, when and only when the original signature is generated, some sub-messages can be designated as undeletable. For instance, for a message $M = \{A, B, C\}$, if the signer wants to make B undeletable, she produces a special sub-message D stating that B is undeletable and signs the updated set $M' := M \cup \{D\}$. This simple approach has a problem that the sub-message D itself can be deleted. To resolve it, we prepend bit 1 to D and bit 0 to the others, and we make an agreement in advance that every message must have one sub-message starting with bit 1. In our DCHSSB, any deletable sub-message can be changed to undeletable anytime. The change is one-way, which means we cannot make any undeletable sub-message deletable. If every sub-message goes undeletable at some point, the message is finalized. We show that DCHSSB can be transformed from HSSM. More specifically, DCHSSB is a subclass of HSSM. From our HSSM scheme, we obtain a DCHSSB scheme secure under the DLIN assumption. To the best of our knowledge, our DCHSSB scheme is the first one which is adaptively unforgeable and strongly context-hiding under the standard assumptions.

Application 2: Disclosure-Controllable RS. In redactable signatures (RS) with maximal number of sub-messages $N \in \mathbb{N}$, each message M is an ordered list in the form of $(m_1, \cdots, m_n)$ for some $n \in [1, N]$, where $m_i \in \{0,1\}^L \cup \{*\}$. Each (non-redacted) sub-message $m_i (\neq *)$ can be changed to $*$, which means it has been redacted, i.e., blacked out. In the ordinary RS, any (non-redacted) sub-message can be redacted anytime. In disclosure-controllable RS (DCRS), any sub-message which is *non-redacted* and *redactable* can be *unredactable*. Specifically, each sub-message has one of the following three states, namely **S1**: not redacted yet and redactable, **S2**: already redacted, and **S3**: not redacted yet and unredactable. Any state only transitions from **S1** to **S2** or from **S1** to **S3**. If every sub-message goes in **S2** or **S3**, the message is finalized. We show that DCRS can be transformed from DCHSSB. From our DCHSSB scheme, we obtain a DCRS scheme secure under the DLIN assumption. As our DCHSSB scheme explained earlier, our DCRS scheme is the first one which is adaptively unforgeable and strongly context-hiding under the standard assumptions.

Application 3: (Key-Delegatable) Subset Predicate Signatures (SBPS). Subset predicate signatures is the digital signature analogue of subset predicate encryption [17]. A secret-key associated with a set $X \in 2^{\{0,1\}^L}$ succeeds in generating a signature on a message associated with any superset $Y \supseteq X$. Our SBPS has key-delegatability, which means that a secret-key for X generates a new secret-key for any superset $X' \supseteq X$. As attribute-based signatures [22,9], we define unforgeability and signer-privacy. The latter security guarantees that any signature with a set Y has no more information about the signer's set X than the fact that $X \subseteq Y$. Identity-based ring signatures (IBRS) [25] is a subclass of SBPS because IBRS is identical to SBPS with the following two restrictions, (1) key-delegation is not allowed and (2) cardinality of the set X is fixed to 1. HSSP can be transformed into IBRS as shown in [21]. In fact, it can be transformed into the stronger primitive SBPS.

Application 4: (Key-Delegatable) Superset Predicate Signatures (SPPS). SPPS is the dual primitive of SBPS. A secret-key associated with a set $X \in 2^{\{0,1\}^L}$ generates a signature associated with any subset $Y \subseteq X$ and generates another secret-key associated with any subset $X' \subseteq X$. We show that SPPS can be transformed from HSSM in a simple and efficient manner. Actually, SPPS can be transformed from a weaker primitive HSSB. However, its transformation itself is somewhat complicated. Moreover, if we instantiate it by any one of the existing

SCH-secure HSSB scheme, an inefficient SPPS scheme whose secret-key and signature lengths increase linearly with the message length is obtained.

Application 5: Wildcarded Identity-Based Signatures (WIBS). WIBS is the digital signatures analogue of wildcarded identity-based encryption [1]. Each secret-key is associated with an identity $X = (x_1, \cdots, x_n) \in (\{0,1\}^L)^n$ for some $L, n \in \mathbb{N}$, and it succeeds in generating a signature associated with any wildcarded identity $Y = (y_1, \cdots, y_n) \in (\{0,1\}^L \cup \{*\})^n$ s.t. $y_i \neq * \implies x_i = y_i$ for all $i \in [1, n]$. We show that WIBS can be transformed from HSSM efficiently.

Paper Organization. In Sect. 2, we explain notions used in this paper, and define symmetric bilinear paring and some computational assumptions. In Sect. 3, we define general $\mathcal{P}$-HS, then show that HSSM is a subclass of $\mathcal{P}$-HS. In Sect. 4, we propose our HSSM schemes. In Sect. 5, we present the applications of HSSM.

2 Preliminaries

Notations. For $\lambda \in \mathbb{N}$, 1^λ denotes a security parameter. A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if for every $c \in \mathbb{N}$, there exists $x_0 \in \mathbb{N}$ such that for every $x \geq x_0$, $f(x) \leq x^{-c}$. We parse a binary string $x \in \{0,1\}^L$ as $x[L-1]||\cdots||x[0]$, where $x[i] \in \{0,1\}$ for all $i \in [0, L-1]$. PPT is the abbreviation of probabilistic polynomial-time. For a set A, $a \overset{\text{U}}{\leftarrow} A$ means that an element a is chosen uniformly at random from A. For a set A, $|A|$ denotes its cardinality.

$\mathcal{G}$ takes a security parameter 1^λ with $\lambda \in \mathbb{N}$ and outputs a group description $(p, \mathbb{G}, \mathbb{G}_T, e, g)$. p is a prime with length λ. $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicative groups with order p. g is a generator of $\mathbb{G}$. $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently-computable function which satisfies both of the following conditions, namely (1) bilinearity: for any $a, b \in \mathbb{Z}_p$, $e(g^a, g^b) = e(g,g)^{ab}$, and (2) non-degeneracy: $e(g,g) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ denotes the unit element of $\mathbb{G}_T$.

We define three computational hardness assumptions.

Definition 1. *The computational Diffie-Hellman (CDH) assumption holds on the group $\mathbb{G}$ if for every PPT $\mathcal{A}$, its advantage $\mathit{Adv}^{\mathit{CDH}}_{\mathcal{A},\mathbb{G}}(\lambda) := \Pr[g^{ab} \leftarrow \mathcal{A}(g, g^a, g^b)]$ (with $a, b \overset{\text{U}}{\leftarrow} \mathbb{Z}_p$) is negligible.*

Definition 2. *The decisional linear (DLIN) assumption holds on the group $\mathbb{G}$ if for every PPT $\mathcal{A}$, its advantage $\mathit{Adv}^{\mathit{DLIN}}_{\mathcal{A},\mathbb{G}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(g^a, g^b, g^{ab}, g^{bd}, g^{c+d})]| - \Pr[1 \leftarrow \mathcal{A}(g^a, g^b, g^{ab}, g^{bd}, g^z)]$ (with $a, b, c, d, z \overset{\text{U}}{\leftarrow} \mathbb{Z}_p$) is negligible.*

Definition 3. *The q-simultaneous flexible pairing (q-SFP) problem [4] relative to the group $\mathbb{G}$ is given $g_z, h_z, g_r, h_r, a, \tilde{a}, b, \tilde{b} \in \mathbb{G}$ and q tuples $\{(z_j, r_j, s_j, t_j, u_j, v_j, w_j) \in \mathbb{G}^7\}_{j=1}^q$ such that for each $j \in [1, q]$,*

$$e(a, \tilde{a}) = e(g_z, z_j) \cdot e(g_r, r_j) \cdot e(s_j, t_j), \quad e(b, \tilde{b}) = e(h_z, z_j) \cdot e(h_r, u_j) \cdot e(v_j, w_j), \quad (1)$$

to find a new tuple $(z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star) \in \mathbb{G}^7$ satisfying the conditions (1) and the condition $z^\star \notin \{1_{\mathbb{G}}, z_1, \cdots, z_q\}$. The q-SFP assumption holds if for any PPT $\mathcal{A}$, its advantage $\mathit{Adv}^{q\text{-}\mathit{SFP}}_{\mathcal{A},\mathbb{G}}(\lambda)$ to solve the above problem is negligible.

3 HS for Subset and Superset Mixed Predicates (HSSM)

We firstly define Homomorphic signatures for a general predicate $\mathcal{P} : \mathcal{M} \times \mathcal{M} \times 1/0$ (abbreviated as $\mathcal{P}$-HS) [5], then define HS for subset and superset mixed predicates (HSSM) as a subclass of $\mathcal{P}$-HS.

Syntax. $\mathcal{P}$-HS consists of the following four polynomial-time algorithms. `Ver` is deterministic and the others are probabilistic.

Key-Generation `KGen`: This algorithm takes 1^λ, then outputs a public-key pk and a secret-key sk. $(pk, sk) \leftarrow \texttt{KGen}(1^\lambda)$

Signing `Sig`: Take a secret-key sk and a message $M \in \mathcal{M}$, then output a signature σ. $\sigma \leftarrow \texttt{Sig}(sk, M)$

Derivation `Derive`: Take a public-key pk, a message M, a signature σ and a message $M' \in \mathcal{M}$, then output a signature σ'. $\sigma' \leftarrow \texttt{Derive}(pk, M, \sigma, M')$

Verification `Ver`: Take a public-key pk, a message $M \in \mathcal{M}$ and a signature σ, then output 1 (meaning *accept*) or 0 (*reject*). $1/0 \leftarrow \texttt{Ver}(pk, M, \sigma)$

Every $\mathcal{P}$-HS scheme must be correct. A $\mathcal{P}$-HS scheme is correct if for any $\lambda \in \mathbb{N}$, any $(pk, sk) \leftarrow \texttt{KGen}(1^\lambda)$, any $M \in \mathcal{M}$ and any $M' \in \mathcal{M}$ s.t. $1 \leftarrow \mathcal{P}(M, M')$, both of the following two conditions are satisfied, namely (1) $1 \leftarrow \texttt{Ver}(pk, M, \texttt{Sig}(sk, M))$ and (2) $1 \leftarrow \texttt{Ver}(pk, M, \texttt{Derive}(pk, M, \texttt{Sig}(sk, M), M'))$.

Security. As security, we define unforgeability and context-hiding. As unforgeability notions, we define unforgeability (`UNF`) and weak unforgeability (`wUNF`) with the following experiments.

```
$\boldsymbol{Expt}^{\texttt{UNF}}_{\Sigma_{\text{P-HS}},\mathcal{A}}(1^\lambda)$:   // $\boldsymbol{Expt}^{\texttt{wUNF}}_{\Sigma_{\text{P-HS}},\mathcal{A}}$
  1. Generate $(pk, sk) \leftarrow \texttt{KGen}(1^\lambda)$. Initialize two tables $Q, Q' := \emptyset$
  2. $(\sigma^*, M^* \in \mathcal{M}) \leftarrow \mathcal{A}^{\mathfrak{Sign},\mathfrak{Derive},\mathfrak{Reveal}}(pk)$
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  - $\mathfrak{Sign}(M \in \mathcal{M})$:
      $\sigma \leftarrow \texttt{Sig}(sk, M)$. Choose an unused handle $h \in \mathcal{H}$.
      $Q := Q \cup \{(h, M, \sigma)\}$. Rtrn $h$
  - $\mathfrak{Derive}(h \in \mathcal{H}, M' \in \mathcal{M})$:
      Rtrn $\bot$. Rtrn $\bot$ if $\nexists(M, \sigma)$ s.t. $(h, M, \sigma) \in Q \wedge 1 \leftarrow \mathcal{P}(M, M')$.
      $\sigma' \leftarrow \texttt{Derive}(pk, M, \sigma, M')$. Choose an unused handle $h' \in \mathcal{H}$.
      $Q := Q \cup \{(h', M', \sigma')\}$. Rtrn $h'$
  - $\mathfrak{Reveal}(h \in \mathcal{H})$:
      Rtrn $\bot$ if $\nexists(M, \sigma)$ s.t. $(h, M, \sigma) \in Q$. Update $Q' := Q' \cup \{M\}$. Rtrn $\sigma$
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  3. Rtrn 1 if (1) $1 \leftarrow \texttt{Ver}(pk, M^*, \sigma^*)$ and (2) $0 \leftarrow \mathcal{P}(M, M^*)$ for all $M \in Q'$.
  4. Rtrn 0
```

In the experiment for `UNF`, the PPT adversary $\mathcal{A}$ receives an honestly generated public-key pk, adaptively accesses three oracles, namely signing, derivation and (signature-)revelation, then outputs a forged signature σ^* on M^*. The grey command **Rtrn** $\bot$. in the derivation oracle is considered only in the experiment for `wUNF`, which means that the derivation oracle is useless because it always returns $\bot$. `UNF` is defined as follows and `wUNF` is analogously defined.

Definition 4. *A scheme $\Sigma_{\text{P-HS}}$ is* `UNF` *if for any $\lambda \in \mathbb{N}$ and any PPT algorithm $\mathcal{A}$, its advantage $\texttt{Adv}^{\texttt{UNF}}_{\Sigma_{\text{P-HS}},\mathcal{A}}(\lambda) := \Pr[1 \leftarrow \boldsymbol{Expt}^{\texttt{UNF}}_{\Sigma_{\text{P-HS}},\mathcal{A}}(1^\lambda)]$ is negligible.*

As privacy (or unlinkability) related security notions, we define strong context-hiding (`SCH`) [5] and complete context-hiding (`CCH`) [6].

Definition 5. *A $\mathcal{P}$-HS scheme is* `CCH` *(resp.* `SCH`*) if for any $\lambda \in \mathbb{N}$, any $(pk, sk) \leftarrow \texttt{KGen}(1^\lambda)$, any message $M \in \mathcal{M}$, any valid signature σ s.t. $1 \leftarrow \texttt{Ver}(pk, M, \sigma)$ (resp. any honestly-generated signature $\sigma \leftarrow \texttt{Sig}(sk, M)$), any message $M' \in \mathcal{M}$ s.t. $1 \leftarrow \mathcal{P}(M, M')$, the following two distributions are identical, namely (1) $\{sk, \sigma, \texttt{Derive}(pk, M, \sigma, M')\}$ and (2) $\{sk, \sigma, \texttt{Sig}(sk, M')\}$.*

It is obvious that, for any $\mathcal{P}$-HS scheme, `CCH` implies `SCH`, and the conjunction of `SCH` and `wUNF` implies `UNF`.

HSSM as a Subclass of $\mathcal{P}$-HS. In HSSM, each message is a pair of sets $(M, W) \in 2^{\{0,1\}^L} \times 2^{\{0,1\}^L}$ for an integer $L \in \mathbb{N}$. Predicate $\mathcal{P}_{\texttt{mixed}}$ for HSSM takes two messages $(M, W), (M', W') \in 2^{\{0,1\}^L} \times 2^{\{0,1\}^L}$, then outputs 1 if (1) M' *is a subset of* M, *i.e.,* $M' \subseteq M$, and (2) W' *is a superset of* W, *i.e.,* $W' \supseteq W$. HSSM is a simple combination of HS for subset predicates (HSSB) [5,6] and HS for superset predicates (HSSP) originally called history-hiding append-only signatures in [21].

4 Our HSSM Schemes

Groth-Sahai Non-Interactive Witness-Indistinguishable (GS NIWI) Proof [15]. In the GS NIWI proof system, based on a symmetric bilinear pairing $e : \mathbb{G}^2 \rightarrow \mathbb{G}_T$ with groups $\mathbb{G}, \mathbb{G}_T$ whose order is a prime p, a CRS consists of three vectors $\vec{f}_1, \vec{f}_2, \vec{f}_3 \in \mathbb{G}^3$, where $\vec{f}_1 = (f_1, 1, g)$ and $\vec{f}_2 = (1, f_2, g)$ with $f_1, f_2 \in \mathbb{G}$. A commitment $\vec{C}$ to an element $\mathcal{X} \in \mathbb{G}$ is given as $(1, 1, \mathcal{X}) \cdot \vec{f}_1^{\,r} \cdot \vec{f}_2^{\,s} \cdot \vec{f}_3^{\,t}$ with $r, s, t \xleftarrow{\text{U}} \mathbb{Z}_p$. The CRS is in one of the two settings. In the perfect soundness setting, where the vector $\vec{f}_3$ is chosen outside the span of the other vectors $\vec{f}_1$ and $\vec{f}_2$, any commitment is perfectly hiding. In the perfect witness-indistinguishability(WI) setting, the CRS satisfies $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \cdot \vec{f}_2^{\,\xi_2}$ with $\xi_1, \xi_2 \in \mathbb{Z}_p$, and from any commitment $\vec{C} = (f_1^{r+\xi_1 t}, f_2^{s+\xi_2 t}, \mathcal{X} \cdot g^{r+s+t(\xi_1+\xi_2)})$ distributing identically to a ciphertext of the Boneh-Boyen-Shacham (BBS) encryption [10], the committed variable $\mathcal{X}$ is extracted by using $\beta_1 = \log_g(f_1)$ and $\beta_2 = \log_g(f_2)$. The two settings of CRS are hard to distinguish under the decisional linear (DLIN) assumption.

Given n variables $\mathcal{X}_1, \cdots, \mathcal{X}_n \in \mathbb{G}$, a prover computes a commitment $\vec{C}_{\mathcal{X}_i} \in \mathbb{G}^3$ to each variable $\mathcal{X}_i$ with randomness $r_i, s_i, t_i \xleftarrow{\text{U}} \mathbb{Z}_p$, then computes a proof for a pairing-product equation (PPE) in a general form of $\prod_{i=1}^n e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^n \prod_{j=1}^n e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T$ with constants $\mathcal{A}_i \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$ and $t_T \in \mathbb{G}_T$. In this paper, we use only a simpler form of $\prod_{i=1}^n e(\mathcal{A}_i, \mathcal{X}_i) = t_T$. In this case, the proof $\vec{\pi} \in \mathbb{G}^3$ is simply $(\prod_{i=1}^n \mathcal{A}_i^{r_i}, \prod_{i=1}^n \mathcal{A}_i^{s_i}, \prod_{i=1}^n \mathcal{A}_i^{t_i})$. Each commitment $\vec{C}_{\mathcal{X}_i}$ is publicly and perfectly re-randomized by choosing $r'_i, s'_i, t'_i \xleftarrow{\text{U}} \mathbb{Z}_p$ then computing $\vec{C'}_{\mathcal{X}_i} := \vec{C}_{\mathcal{X}_i} \cdot \vec{f}_1^{r'_i} \cdot \vec{f}_2^{s'_i} \cdot \vec{f}_3^{t'_i}$. The proof $\vec{\pi}$ must be naturally updated to $\vec{\pi} \cdot (\prod_{i=1}^n \mathcal{A}_i^{r'_i}, \prod_{i=1}^n \mathcal{A}_i^{s'_i}, \prod_{i=1}^n \mathcal{A}_i^{t'_i})$.

Attrapadung et al.'s HSSB Scheme [6]. Their HSSB scheme is based on the GS proof [15], Waters signatures [24] (described in Appendix 3) and Abe-Haralambiev-Ohkubo (AHO) structure-preserving signatures (SPS) [4,3] (described in Appendix 4). In key-generation, a signer generates a long-term AHO key-pair. When signing a message $M \in 2^{\{0,1\}^L}$, the signer freshly generates a one-time Waters key-pair (X, x), where $X := g^x$ with $x \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p$. For each sub-message $m \in M$, the signer generates a Waters signature $(\sigma_{m,1}, \sigma_{m,2}) := (h^x \cdot H_{\mathbb{G}}(m)^{\chi_m}, g^{\chi_m}) \in \mathbb{G}^2$ under a public group element $h \in \mathbb{G}$ and a randomness $\chi_m \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p$. The signer also generates an AHO signature $(\theta_1, \cdots, \theta_7) \in \mathbb{G}^7$ on the Waters public-key $X \in \mathbb{G}$. Finally, the signer computes GS commitments to $X, \{\sigma_{m,1}\}_{m \in M}, \theta_1, \theta_2, \theta_5 \in \mathbb{G}$, then computes GS proofs that (1) *the AHO signature* $(\theta_1, \cdots, \theta_7)$ *on* X *is valid* and (2) *the Waters signature* $(\sigma_{m,1}, \sigma_{m,2})$ *on* m *under the public-key* X *is valid for each* $m \in M$. When a public party derives a signature on a subset $M' \subseteq M$, for each deleted sub-message $m \in M \setminus M'$, the commitment to $\sigma_{m,1}$, (non-committed) $\sigma_{m,2}$ and the GS proof related to this Waters signature are deleted from the original HSSB signature, then all of the remaining variables are perfectly re-randomized.

Libert et al.'s HSSP Scheme [21]. Libert et al.'s HSSP scheme is similar to the Attrapadung et al.'s HSSB scheme. It is based on an arbitrary SPS scheme satisfying unforgeability against extended random messages attacks [2], GS NIWI proof [15], and Boneh-Lynn-Shacham (BLS) signatures [11] instantiated by the Waters programmable hash function [24], i.e., $H_{\mathbb{G}}$. In key-generation, a signer generates a long-term key-pair of the SPS scheme. When signing a message $W \in 2^{\{0,1\}^L}$, the signer generates a one-time BLS key-pair (Y, y), where $Y := g^y$ with $y \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p$. The signer divides the BLS secret-key y into $|W|$ number of shares by using $|W|$-out-of-$|W|$ secret sharing[1]. Specifically, she uniform-randomly chooses $\{\gamma_w \in \mathbb{Z}_p\}_{w \in W}$ satisfying that $\sum_{w \in W} \gamma_w = y \pmod{p}$. For each $w \in W$, she computes $(\sigma_{w,1}, \sigma_{w,2}) := (H_{\mathbb{G}}(w)^{\gamma_w}, g^{\gamma_w}) \in \mathbb{G}^2$, where the first element $\sigma_{w,1}$ is a BLS signature on w under the *pseudo* BLS public-key $\sigma_{w,2}$. The signer also generates a SPS signature $(\theta_1, \cdots, \theta_{l_{sps}}) \in \mathbb{G}^{l_{sps}}$ on $Y \in \mathbb{G}$. Finally, the signer computes GS commitments to $Y, \{\sigma_{w,1}, \sigma_{w,2}\}_{w \in W}, \theta_1, \cdots, \theta_{l_{sps}} \in \mathbb{G}$, then computes GS proofs that (1) *the SPS signature* $(\theta_1, \cdots, \theta_{l_{sps}})$ *on* Y *is valid*, (2) *the BLS signature* $\sigma_{w,1}$ *on* w *under the pseudo public-key* $\sigma_{w,2}$ *is valid for each* $w \in W$ and (3) $\prod_{w \in W} \sigma_{w,2} = Y$. When a public user derives a signature on a superset $W' \supseteq W$, $\{\gamma_w \in \mathbb{Z}_p\}_{w \in W'}$ s.t. $\sum_{w \in W'} \gamma_w = 0 \pmod{p}$ are randomly chosen. For each *added* element $w \in W' \backslash W$, we compute $(\sigma_{w,1}, \sigma_{w,2}) := (H_{\mathbb{G}}(w)^{\gamma'_w}, g^{\gamma'_w})$ then compute GS commitments to them. For each *inherited* element $w \in W' \cap W$, we update the committed variables $\sigma_{w,1}, \sigma_{w,2} \in \mathbb{G}$ to $\sigma_{w,1} \cdot H_{\mathbb{G}}(w)^{\gamma'_w}$ and $\sigma_{w,2} \cdot g^{\gamma'_w}$,

[1] $|W|$ denotes cardinality of the set W.

respectively. The GS proof for the relation $\prod_{w \in W} \sigma_{w,2} = Y$ is properly updated and all of the GS commitments and proofs are perfectly re-randomized.

4.1 Construction

Our HSSM scheme is a simple combination of the Attrapadung et al.'s HSSB and Libert et al.'s HSSP schemes. As the underlying SPS scheme, we also use the AHO SPS scheme [4,3] with message space $\mathbb{G}^2$. On signature generation, a fresh one-time Waters and BLS key-pairs (X, x), (Y, y) are generated, where $X := g^x$ and $Y := g^y$ with $x, y \xleftarrow{\text{U}} \mathbb{Z}_p$. An AHO signature $(\theta_1, \cdots, \theta_7)$ on $(X, Y) \in \mathbb{G}^2$ is generated. For each $m \in M$, a Waters signature $(\sigma_{m,1}, \sigma_{m,2})$ on m is generated. The original BLS secret-key $y \in \mathbb{Z}_p$ is divided into $|W|$ number of shares $\{\gamma_w \in \mathbb{Z}_p\}_{w \in W}$ by $|W|$-out-of-$|W|$ secret sharing, then for each $w \in W$, a BLS signature $\sigma_{w,1}$ on w under the pseudo BLS public-key $g^{\gamma_w} (=: \sigma_{w,2})$ is generated. Finally, the GS commitments and proofs are properly generated.

For any element $Z \in \mathbb{G}$, $\iota(Z)$ denotes $(1, 1, Z) \in \mathbb{G}^3$. For any $h, g_1, g_2, g_3 \in \mathbb{G}$, $E(h, (g_1, g_2, g_3))$ denotes $(e(h, g_1), e(h, g_2), e(h, g_3)) \in \mathbb{G}_T^3$. Our HSSM scheme is formally described as follows.

$\mathtt{KGen}(1^\lambda, L)$**:** Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ whose order is a prime p of bit length λ. g denotes a generator of $\mathbb{G}$. Conduct the following three steps.

1. Generate the public parameter of the Waters signatures [24] by choosing $h, u', u_0, \cdots, u_{L-1} \xleftarrow{\text{U}} \mathbb{G}$. The Waters programmable hash function $H_{\mathbb{G}} : \{0,1\}^L \to \mathbb{G}$ takes a sub-message $m \in \{0,1\}^L$, being parsed as $m[L-1] \parallel \cdots \parallel m[0]$, then outputs $u' \prod_{i=0}^{L-1} u_i^{m[i]} \in \mathbb{G}$.
2. Generate a key-pair $(pk_{\mathsf{s}}, sk_{\mathsf{s}})$ of the AHO SPS [4,3]. Parse it as $((g_r, h_r, g_z, h_z, g_1, h_1, g_2, h_2, A, B), (\alpha_a, \alpha_b, \gamma_z, \delta_z, \gamma_1, \delta_1, \gamma_2, \delta_2))$.
3. Generate a CRS $\boldsymbol{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$ for the perfect WI setting of the GS NIWI proof [15]. Concretely, $\vec{f}_1 := (f_1, 1, g)$, $\vec{f}_2 := (1, f_2, g)$ and $\vec{f}_3 := \vec{f}_1^{\xi_1} \cdot \vec{f}_2^{\xi_2} \cdot (1, 1, g)^{-1}$, where $f_1, f_2 \xleftarrow{\text{U}} \mathbb{G}$ and $\xi_1, \xi_2 \xleftarrow{\text{U}} \mathbb{Z}_p$.

Output (pk, sk), where $pk := (\mathbb{G}, \mathbb{G}_T, g, h, u', \{u_i\}_{i=0}^{L-1}, pk_{\mathsf{s}}, \boldsymbol{f})$ and $sk := sk_{\mathsf{s}}$.

$\mathtt{Sig}(sk, M \in 2^{\{0,1\}^L}, W \in 2^{\{0,1\}^L})$**:** Firstly, conduct the following six steps.

1. Choose $x, y \xleftarrow{\text{U}} \mathbb{Z}_p$. Let $X := g^x$ and $Y := g^y$.
2. Generate an AHO signature $\sigma_{\mathsf{s}} = (\theta_1, \cdots, \theta_7)$ on a message $(X, Y) \in \mathbb{G}^2$.
3. For each $m \in M$, generate a Waters signature on m, i.e., $(\sigma_{m,1}, \sigma_{m,2}) := (h^x \cdot H_{\mathbb{G}}(m)^{\chi_m}, g^{\chi_m})$ with $\chi_m \xleftarrow{\text{U}} \mathbb{Z}_p$.
4. Choose $\{\gamma_w \in \mathbb{Z}_p\}_{w \in W}$ satisfying $\sum_{w \in W} \gamma_w = y \pmod p$ uniformly at random. Then for each $w \in W$, compute $(\sigma_{w,1}, \sigma_{w,1}) := (H_{\mathbb{G}}(w)^{\gamma_w}, g^{\gamma_w})$.
5. Compute GS commitments for all of the group elements X, Y, θ_1, θ_2, θ_5, $\{\sigma_{m,1}\}_{m \in M}$ and $\{\sigma_{w,1}, \sigma_{w,2}\}_{w \in W}$. The commitments are denoted by $\vec{C}_X, \vec{C}_Y, \vec{C}_{\theta_1}, \vec{C}_{\theta_2}, \vec{C}_{\theta_5}, \vec{C}_{\sigma_{m,1}}, \vec{C}_{\sigma_{w,1}}, \vec{C}_{\sigma_{w,2}}$, respectively. Specifically, for each committed element $Z \in \mathbb{G}$, the commitment $\vec{C}_Z$ is computed as $\iota_{\mathbb{G}}(Z) \cdot \vec{f}_1^{r_Z} \cdot \vec{f}_2^{s_Z} \cdot \vec{f}_3^{t_Z}$ with $r_Z, s_Z, t_Z \xleftarrow{\text{U}} \mathbb{Z}_p$.

6. Compute GS proofs for all of the following PPEs.

$$
\begin{aligned}
A \cdot e(\theta_3, \theta_4)^{-1} &= e(g_z, \theta_1) \cdot e(g_r, \theta_2) \cdot e(g_1, X) \cdot e(g_2, Y) && (2)\\
B \cdot e(\theta_6, \theta_7)^{-1} &= e(h_z, \theta_1) \cdot e(h_r, \theta_5) \cdot e(h_1, X) \cdot e(h_2, Y) && (3)\\
e(\sigma_{m,1}, g) &= e(X, h) \cdot e(\sigma_{m,2}, H_{\mathbb{G}}(m)) \quad \textbf{(For each } m \in M\textbf{)} && (4)\\
e(\sigma_{w,1}, g) &= e(\sigma_{w,2}, H_{\mathbb{G}}(w)) \qquad\qquad \textbf{(For each } w \in W\textbf{)} && (5)\\
e(Y, g) &= \prod_{w \in W} e(\sigma_{w,2}, g) && (6)
\end{aligned}
$$

The proofs are denoted by $\vec{\pi}_A$, $\vec{\pi}_B$, $\vec{\pi}_m$, $\vec{\pi}_w$ and $\vec{\pi}_{sum}$, respectively, and computed as follows.

$$
\begin{aligned}
\vec{\pi}_A &:= (g_z^{r_{\theta_1}} g_r^{r_{\theta_2}} g_1^{r_X} g_2^{r_Y}, g_z^{s_{\theta_1}} g_r^{s_{\theta_2}} g_1^{s_X} g_2^{s_Y}, g_z^{t_{\theta_1}} g_r^{t_{\theta_2}} g_1^{t_X} g_2^{t_Y})\\
\vec{\pi}_B &:= (h_z^{r_{\theta_1}} h_r^{r_{\theta_2}} h_1^{r_X} h_2^{r_Y}, h_z^{s_{\theta_1}} h_r^{s_{\theta_2}} h_1^{s_X} h_2^{s_Y}, h_z^{t_{\theta_1}} h_r^{t_{\theta_2}} h_1^{t_X} h_2^{t_Y})\\
\vec{\pi}_m &:= (g^{r_{\sigma_{m,1}}} \cdot h^{-r_X}, g^{s_{\sigma_{m,1}}} \cdot h^{-s_X}, g^{t_{\sigma_{m,1}}} \cdot h^{-t_X})\\
\vec{\pi}_w &:= (g^{r_{\sigma_{w,1}}} \cdot H_{\mathbb{G}}(w)^{-r_{\sigma_{w,2}}}, g^{s_{\sigma_{w,1}}} \cdot H_{\mathbb{G}}(w)^{-s_{\sigma_{w,2}}}, g^{t_{\sigma_{w,1}}} \cdot H_{\mathbb{G}}(w)^{-t_{\sigma_{w,2}}})\\
\vec{\pi}_{sum} &:= (g^{r_Y - \sum_{w \in W} r_{\sigma_{w,2}}}, g^{s_Y - \sum_{w \in W} s_{\sigma_{w,2}}}, g^{t_Y - \sum_{w \in W} t_{\sigma_{w,2}}})
\end{aligned}
$$

Finally, output a signature σ which is

$$
\begin{pmatrix} \vec{C}_X, \vec{C}_Y, \{\vec{C}_{\theta_i}\}_{i \in \{1,2,5\}}, \{\theta_i\}_{i \in \{3,4,6,7\}}, \vec{\pi}_A, \vec{\pi}_B, \\ \{\vec{C}_{\sigma_{m,1}}, \sigma_{m,2}, \vec{\pi}_m\}_{m \in M}, \{\vec{C}_{\sigma_{w,1}}, \vec{C}_{\sigma_{w,2}}, \vec{\pi}_w\}_{w \in W}, \vec{\pi}_{sum} \end{pmatrix} \in \mathbb{G}^{28+7|M|+9|W|} \tag{7}
$$

`Derive`$(pk, (M, W), \sigma, (M', W'))$**:** Parse σ as (7). Assume that $1 \leftarrow \mathcal{P}_{\texttt{mixed}}((M, W), (M', W'))$. Conduct the following six steps.

1. Re-randomize the GS commitments $\vec{C}_X, \vec{C}_Y \in \mathbb{G}^3$. For each variable $Z \in \{X, Y\}$, $\vec{C}'_Z := \vec{C}_Z \cdot \vec{f}_1^{r'_Z} \cdot \vec{f}_2^{s'_Z} \cdot \vec{f}_3^{t'_Z}$, where $r'_Z, s'_Z, t'_Z \overset{\text{U}}{\longleftarrow} \mathbb{Z}_p$.
2. Randomize the AHO signature $(\theta_1, \cdots, \theta_7) \in \mathbb{G}^7$ in the same manner as [4,3,6] explained in Appendix 4. Choose $\eta_2, \eta_5, \mu, \nu \overset{\text{U}}{\longleftarrow} \mathbb{Z}_p$, then

$$
\begin{aligned}
&\vec{C}'_{\theta_1} := \vec{C}_{\theta_1} \cdot \vec{f}_1^{r'_{\theta_1}} \cdot \vec{f}_2^{s'_{\theta_1}} \cdot \vec{f}_3^{t'_{\theta_1}},\\
&\vec{C}'_{\theta_2} := \vec{C}_{\theta_2} \cdot \iota_{\mathbb{G}}(\theta_4^{\eta_2}) \cdot \vec{f}_1^{r'_{\theta_2}} \cdot \vec{f}_2^{s'_{\theta_2}} \cdot \vec{f}_3^{t'_{\theta_2}}, \quad \theta'_3 := (\theta_3 \cdot g_r^{-\eta_2})^{1/\mu}, \quad \theta'_4 := \theta_4^{\mu},\\
&\vec{C}'_{\theta_5} := \vec{C}_{\theta_5} \cdot \iota_{\mathbb{G}}(\theta_7^{\eta_5}) \cdot \vec{f}_1^{r'_{\theta_5}} \cdot \vec{f}_2^{s'_{\theta_5}} \cdot \vec{f}_3^{t'_{\theta_5}}, \quad \theta'_6 := (\theta_6 \cdot h_r^{-\eta_5})^{1/\nu}, \quad \theta'_7 := \theta_7^{\nu},
\end{aligned}
$$

   where $r'_{\theta_i}, r'_{\theta_i}, t'_{\theta_i} \overset{\text{U}}{\longleftarrow} \mathbb{Z}_p$ for each $i \in \{1, 2, 5\}$. The GS proofs $\vec{\pi}_A, \vec{\pi}_B \in \mathbb{G}^3$ are naturally updated as follows.

$$
\begin{aligned}
\vec{\pi}'_A &:= \vec{\pi}_A \cdot (g_z^{r'_{\theta_1}} g_r^{r'_{\theta_2}} g_1^{r'_X} g_2^{r'_Y}, g_z^{s'_{\theta_1}} g_r^{s'_{\theta_2}} g_1^{s'_X} g_2^{s'_Y}, g_z^{t'_{\theta_1}} g_r^{t'_{\theta_2}} g_1^{t'_X} g_2^{t'_Y})\\
\vec{\pi}'_B &:= \vec{\pi}_B \cdot (h_z^{r'_{\theta_1}} h_r^{r'_{\theta_2}} h_1^{r'_X} h_2^{r'_Y}, h_z^{s'_{\theta_1}} h_r^{s'_{\theta_2}} h_1^{s'_X} h_2^{s'_Y}, h_z^{t'_{\theta_1}} h_r^{t'_{\theta_2}} h_1^{t'_X} h_2^{t'_Y})
\end{aligned}
$$

3. For each $m \in M'$, re-randomize the Waters signature $(\sigma_{m,1}, \sigma_{m,2}) \in \mathbb{G}$. Choose $\chi'_m \overset{\text{U}}{\longleftarrow} \mathbb{Z}_p$, then compute $\sigma'_{m,2} := \sigma_{m,2} \cdot g^{\chi'_m}$ and $\vec{C}'_{\sigma_{m,1}} := \vec{C}_{\sigma_{m,1}} \cdot$

$\iota_{\mathbb{G}}(H_{\mathbb{G}}(m)^{\chi'_m}) \cdot \vec{f}_1^{\,r'_{\sigma_{m,1}}} \cdot \vec{f}_2^{\,s'_{\sigma_{m,1}}} \cdot \vec{f}_3^{\,t'_{\sigma_{m,1}}}$, where $r'_{\sigma_{m,1}}, s'_{\sigma_{m,1}}, t'_{\sigma_{m,1}} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. The GS proof $\vec{\pi}_m$ is naturally updated to $\vec{\pi}'_m := \vec{\pi}_m \cdot (g^{r'_{\sigma_{m,1}}} \cdot h^{-r'_X}, g^{s'_{\sigma_{m,1}}} \cdot h^{-s'_X}, g^{t'_{\sigma_{m,1}}} \cdot h^{-t'_X})$.

4. Choose $\{\gamma'_w \in \mathbb{Z}_p\}_{w \in W'}$ s.t. $\sum_{w \in W'} \gamma'_w = 0 \pmod{p}$ uniformly at random from $\mathbb{Z}_p$.
5. For each $w \in W$, re-randomize $(\sigma_{w,1}, \sigma_{w,2}) \in \mathbb{G}^2$ and their commitments $\vec{C}_{\sigma_{w,1}}, \vec{C}_{\sigma_{w,2}} \in \mathbb{G}^3$. Compute $\vec{C}'_{\sigma_{w,1}} := \vec{C}_{\sigma_{w,1}} \cdot \iota_{\mathbb{G}}(H_{\mathbb{G}}(w)^{\gamma'_w}) \cdot \vec{f}_1^{\,r'_{\sigma_{w,1}}} \cdot \vec{f}_2^{\,s'_{\sigma_{w,1}}} \cdot \vec{f}_3^{\,t'_{\sigma_{w,1}}}$ and $\vec{C}'_{\sigma_{w,2}} := \vec{C}_{\sigma_{w,2}} \cdot \iota_{\mathbb{G}}(g^{\gamma'_w}) \cdot \vec{f}_1^{\,r'_{\sigma_{w,2}}} \cdot \vec{f}_2^{\,s'_{\sigma_{w,2}}} \cdot \vec{f}_3^{\,t'_{\sigma_{w,2}}}$, where $r'_{\sigma_{w,1}}, s'_{\sigma_{w,1}}, t'_{\sigma_{w,1}} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. The GS proof $\vec{\pi}_w$ is naturally updated to $\vec{\pi}'_w := \vec{\pi}_w \cdot (g^{r'_{\sigma_{w,1}}} \cdot H_{\mathbb{G}}(w)^{-r'_{\sigma_{w,2}}}, g^{s'_{\sigma_{w,1}}} \cdot H_{\mathbb{G}}(w)^{-s'_{\sigma_{w,2}}}, g^{t'_{\sigma_{w,1}}} \cdot H_{\mathbb{G}}(w)^{-t'_{\sigma_{w,2}}})$.
6. For each $w \in W' \setminus W$, newly generate $(\sigma_{w,1}, \sigma_{w,2}) := (H_{\mathbb{G}}(w)^{\gamma'_w}, g^{\gamma'_w})$. Then generate a GS commitments $\vec{C}'_{\sigma_{w,1}} := \iota_{\mathbb{G}}(H_{\mathbb{G}}(w)^{\gamma'_w}) \cdot \vec{f}_1^{\,r'_{\sigma_{w,1}}} \cdot \vec{f}_2^{\,s'_{\sigma_{w,1}}} \cdot \vec{f}_3^{\,t'_{\sigma_{w,1}}}$ and $\vec{C}'_{\sigma_{w,2}} := \iota_{\mathbb{G}}(g^{\gamma'_w}) \cdot \vec{f}_1^{\,r'_{\sigma_{w,2}}} \cdot \vec{f}_2^{\,s'_{\sigma_{w,2}}} \cdot \vec{f}_3^{\,t'_{\sigma_{w,2}}}$, where $r'_{\sigma_{w,1}}, s'_{\sigma_{w,1}}, t'_{\sigma_{w,1}}, r'_{\sigma_{w,2}}, s'_{\sigma_{w,2}}, t'_{\sigma_{w,2}} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Then generate a GS proof $\vec{\pi}'_w := (g^{r'_{\sigma_{w,1}}} \cdot H_{\mathbb{G}}(w)^{-r'_{\sigma_{w,2}}}, g^{s'_{\sigma_{w,1}}} \cdot H_{\mathbb{G}}(w)^{-s'_{\sigma_{w,2}}}, g^{t'_{\sigma_{w,1}}} \cdot H_{\mathbb{G}}(w)^{-t'_{\sigma_{w,2}}})$.

Finally, output a signature σ' composed of all of the updated variables.

$\mathtt{Ver}(pk, (M, W), \sigma)$: Parse σ as (7). Each GS proof $\pi \in \mathbb{G}^3$ is parsed as (π_1, π_2, π_3). Output 1 if and only if all of the following five equations hold.

1. $\iota_{\mathbb{G}_T}(A) \cdot e(\theta_3, \iota_{\mathbb{G}}(\theta_4))^{-1} = E(g_z, \vec{C}_{\theta_1}) \cdot E(g_r, \vec{C}_{\theta_2}) \cdot E(g_1, \vec{C}_X) \cdot E(g_2, \vec{C}_Y) \cdot \prod_{k=1}^{3} E(\pi_{A,k}, \vec{f}_k)$
2. $\iota_{\mathbb{G}_T}(B) \cdot e(\theta_6, \iota_{\mathbb{G}}(\theta_7))^{-1} = E(h_z, \vec{C}_{\theta_1}) \cdot E(h_r, \vec{C}_{\theta_5}) \cdot E(h_1, \vec{C}_X) \cdot E(h_2, \vec{C}_Y) \cdot \prod_{k=1}^{3} E(\pi_{B,k}, \vec{f}_k)$
3. $E(g, \vec{C}_{\sigma_{m,1}}) = E(h, \vec{C}_X) \cdot E(H_{\mathbb{G}}(m), \iota_{\mathbb{G}}(\sigma_{m,2})) \cdot \prod_{k=1}^{3} E(\pi_{m,k}, \vec{f}_k)$ (for each $m \in M$)
4. $E(g, \vec{C}_{\sigma_{w,1}}) = E(H_{\mathbb{G}}(w), \vec{C}_{\sigma_{w,2}}) \cdot \prod_{k=1}^{3} E(\pi_{w,k}, \vec{f}_k)$ (for each $w \in W$)
5. $E(g, \vec{C}_Y) = \prod_{w \in W} E(g, \vec{C}_{\sigma_{w,2}}) \cdot \prod_{k=1}^{3} E(\pi_{sum,k}, \vec{f}_k)$

Theorem 1. *Our HSSM scheme is* `CCH` *unconditionally, and* `wUNF` *under the DLIN assumption w.r.t the group* $\mathbb{G}$ *and the q-SFP assumption, where* $q \in \mathtt{poly}(\lambda)$ *is the maximal number of times that the signing oracle can be accessed.*

Proof. Among various variables included in a derived signature, GS commitments and proofs distribute identically to fresh ones because of the following two facts, (1) GS NIWI proof system is perfectly WI, and (2) they are perfectly re-randomized in the derivation algorithm. The other variables, i.e., $\{\theta_i\}_{i \in \{3,4,6,7\}}$ and $\{\sigma_{m,2}\}_{m \in M'}$, also distribute identically to fresh ones because they are perfectly re-randomized in the derivation algorithm. Hence, our HSSM scheme is `CCH`. To prove its `wUNF`, we define the following 4 experiments.

Expt$_0$**:** This is the standard `wUNF` experiment for the HSSM scheme. The forged signature σ^* is parsed as $(\overrightarrow{C}^*_X, \overrightarrow{C}^*_Y, \{\overrightarrow{C}^*_{\theta_i}\}_{i\in\{1,2,5\}}, \{\theta^*_i\}_{i\in\{3,4,6,7\}}, \overrightarrow{\pi}^*_A, \overrightarrow{\pi}^*_B, \{\overrightarrow{C}^*_{\sigma_{m,1}}, \sigma^*_{m,2}, \overrightarrow{\pi}^*_m\}_{m\in M^*}, \{\overrightarrow{C}^*_{\sigma_{w,1}}, \overrightarrow{C}^*_{\sigma_{w,2}}, \overrightarrow{\pi}^*_w\}_{w\in W^*}, \overrightarrow{\pi}^*_{sum})$.

Expt$_1$**:** The same as ***Expt***$_0$ except that the GS CRS $\boldsymbol{f} = (\overrightarrow{f}_1, \overrightarrow{f}_2, \overrightarrow{f}_3)$ is generated as a perfectly sound one. Specifically, $\overrightarrow{f}_1 := (f_1, 1, g)$, $\overrightarrow{f}_2 := (1, f_2, g)$ and $\overrightarrow{f}_3 := \overrightarrow{f}_1^{\xi_1} \cdot \overrightarrow{f}_2^{\xi_2}$, where $f_1 := g^{\phi_1}$ and $f_2 := g^{\phi_2}$ with $\phi_1, \phi_2, \xi_1, \xi_2 \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p$. Note that in this and later experiments, all GS commitments are perfectly binding. From the forged GS commitments, we can extract all of the hidden variables $X^*, Y^*, \{\theta^*_i\}_{i\in\{1,2,5\}}, \{\sigma^*_{m,1}\}_{m\in M^*}$ and $\{\sigma^*_{w,1}, \sigma^*_{w,2}\}_{w\in W^*}$ by using the BBS decryption keys (ϕ_1, ϕ_2). Since the forged GS proofs are perfectly sound, the extracted variables satisfy all of the five equations (2)-(6).

Expt$_2$**:** For $\kappa \in [1, q]$, $(X_\kappa, Y_\kappa) \in \mathbb{G}^2$ denote the group elements (X, Y) randomly chosen on the κ-th query to the signing oracle. ***Expt***$_2$ is the same as ***Expt***$_1$ except that it aborts if $\nexists \kappa \in [1, q]$ s.t. $(X_\kappa, Y_\kappa) = (X^*, Y^*)$.

Expt$_3$**:** Identical to ***Expt***$_2$ except that it aborts if $\exists \kappa \in [1, q]$ s.t. $(X_\kappa, Y_\kappa) = (X^*, Y^*) \wedge M^* \nsubseteq M_\kappa$, where M_κ is the κ-th query of M to the signing oracle.

S_i denotes the event where ***Expt***$_i$ outputs 1. We obtain $\mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{HSSM}},\mathcal{A},n}(\lambda) = \Pr[S_0] \le \sum_{i=1}^{3} |\Pr[S_{i-1}] - \Pr[S_i]| + \Pr[S_3]$. Because of the following four lemmas, the rightmost formula is negligible under the DLIN and q-SFP assumptions[2]. Lemma 1 is true as shown in [6,21]. □

Lemma 1. $|\Pr[S_0] - \Pr[S_1]|$ *is negligible under the DLIN assumption w.r.t.* $\mathbb{G}$.

Lemma 2. $|\Pr[S_1] - \Pr[S_2]|$ *is negligible under the q-SFP assumption.*

Proof. Let Abort denote the aborting event added in ***Expt***$_2$. Since ***Expt***$_2$ is the same as ***Expt***$_1$ except for the case that the aborting event occurs, $\Pr[S_2] = \Pr[S_1 \wedge \neg\mathsf{Abort}]$. By a basic mathematical theorem, $\Pr[S_1] - \Pr[S_1 \wedge \neg\mathsf{Abort}] = \Pr[S_1] - \Pr[S_2] = \Pr[S_1 \wedge \mathsf{Abort}]$. Let $\mathcal{A}$ denote a PPT adversary which makes the event $S_1 \wedge \mathsf{Abort}$ occur with a non-negligible probability. Let $\mathcal{B}_2$ denote a PPT adversary which, by using $\mathcal{A}$ as black-box, attempts to win the existential unforgeability against adaptively chosen messages attacks (`EUF-CMA`) experiment (defined in Appendix 1) w.r.t. the AHO SPS scheme. $\mathcal{B}_2$ behaves as follows.

$\mathcal{B}_2$ receives an honestly-generated public-key pk_{s} of the AHO scheme. $\mathcal{B}_2$ honestly generates a GS CRS $\boldsymbol{f}$ in the perfect soundness setting and public parameters $h, u', u_0, \cdots, u_{L-1}$ of the Waters signatures. If $\mathcal{A}$ issues (M, W) as a query to the signing oracle, $\mathcal{B}_2$ generates $(X, Y) := (g^x, g^y)$ (where $x, y \overset{\mathrm{U}}{\leftarrow} \mathbb{Z}_p$), then sends the message (X, Y) as a query to the signing oracle to get an AHO signature σ_{s}. $\mathcal{B}_2$ honestly generates the other elements of the HSSM signature σ on (M, W). Since we have assumed that the event $S_1 \wedge \mathsf{Abort}$ occurs, (1) σ^*_{s} *is a valid AHO signature on the message* (X^*, Y^*), and (2) *the message has not been queried to the signing oracle by* $\mathcal{B}_2$. Thus, $\mathcal{B}_2$ wins. It holds that

[2] The CDH assumption is implied by the DLIN and q-SFP assumptions

$\Pr[S_1 \wedge \mathsf{Abort}] \leq \mathtt{Adv}^{\mathtt{EUF\text{-}CMA}}_{\Sigma_{\mathrm{AHO}},\mathcal{B}_2}(\lambda)$, where the right side denotes $\mathcal{B}_2$'s advantage in the $\mathtt{EUF\text{-}CMA}$ experiment which is negligible under the q-SFP assumption. □

Lemma 3. $|\Pr[S_2] - \Pr[S_3]|$ *is negligible under the CDH assumption w.r.t.* $\mathbb{G}$.

Proof. Let Abort denote the aborting event added in $\boldsymbol{Expt}_3$. As the proof of Lemma 2, it holds that $\Pr[S_2] - \Pr[S_3] = \Pr[S_2 \wedge \mathsf{Abort}]$. Let $\mathcal{A}$ denote a PPT adversary which makes the event $S_2 \wedge \mathsf{Abort}$ occur with a non-negligible probability. Let $\mathcal{B}_3$ denote a PPT adversary which, by using $\mathcal{A}$ as black-box, attempts to win the $\mathtt{EUF\text{-}CMA}$ experiment w.r.t. the Waters scheme. $\mathcal{B}_3$ behaves as follows.

$\mathcal{B}_3$ receives honestly-generated public parameters $h, u', u_0, \cdots, u_{L-1}$ and a public-key $X'(:= g^{x'})$ (where $x' \xleftarrow{\mathrm{U}} \mathbb{Z}_p$) of the Waters signature. $\mathcal{B}_3$ honestly generates a GS CRS $\boldsymbol{f}$ in the perfect soundness setting and an AHO key-pair $(pk_{\mathsf{s}}, sk_{\mathsf{s}})$. Since we have assumed that the event $S_2 \wedge \mathsf{Abort}$ occurs, it will hold that $\exists \kappa \in [1, q]$ s.t. $(X_\kappa, Y_\kappa) = (X^*, Y^*) \wedge M_\kappa \not\supseteq M^*$. $\mathcal{B}_3$ guesses such an index κ and chooses $\kappa_{guess} \xleftarrow{\mathrm{U}} [1, q]$. The guess is correct at least with probability $1/q$. $\mathcal{B}_3$ proceeds under an assumption that the guess is correct. If $\mathcal{A}$ issues (M, W) as the κ-th query to the signing oracle, $\mathcal{B}_3$ considers the following two cases.

1. $\kappa \neq \kappa_{guess}$**:** $\mathcal{B}_3$ honestly generates the whole HSSM signature σ oneself.

2. $\kappa = \kappa_{guess}$**:** $\mathcal{B}_3$ uses the given $X' \in \mathbb{G}$ as the Waters public-key. For each $m \in M$, $\mathcal{B}_3$ queries $m \in \{0, 1\}^L$ to the signing oracle to get a Waters signature $(\sigma_{m,1}, \sigma_{m,2})$. $\mathcal{B}_3$ honestly generates the HSSM signature σ oneself.

Since we have assumed that $S_2 \wedge \mathsf{Abort}$ occurs, (1) *there exists* $m^* \in M^*$ *s.t.* $m^* \notin M_{\kappa_{guess}}$, (2) *the Waters signature* $(\sigma^*_{m^*,1}, \sigma^*_{m^*,2})$ *on* m^* *is valid*, and (3) *the message* m^* *has not been queried to the signing oracle by* $\mathcal{B}_3$. Thus, $\mathcal{B}_3$ wins (under the assumption that $\mathcal{B}_3$'s guess κ_{guess} is correct). It holds that $\Pr[S_2 \wedge \mathsf{Abort}] \leq q \cdot \mathtt{Adv}^{\mathtt{EUF\text{-}CMA}}_{\Sigma_{\mathrm{Waters}},\mathcal{B}_3}(\lambda)$, where $\mathtt{Adv}^{\mathtt{EUF\text{-}CMA}}_{\Sigma_{\mathrm{Waters}},\mathcal{B}_3}(\lambda)$ is $\mathcal{B}_3$'s advantage in the $\mathtt{EUF\text{-}CMA}$ experiment w.r.t. the Waters scheme which is negligible under the CDH assumption. In Appendix 6, we rigorously prove that there exists a PPT adversary $\mathcal{B}'_3$ s.t. $\Pr[S_2 \wedge \mathsf{Abort}] \leq 4q \cdot d_M \cdot (L+1) \cdot \mathtt{Adv}^{\mathtt{CDH}}_{\mathcal{B}'_3,\mathbb{G}}(\lambda)$, where $d_M \in \mathtt{poly}(\lambda)$ is the maximal cardinality of the set M. □

Lemma 4. $\Pr[S_3]$ *is negligible under the CDH assumption w.r.t.* $\mathbb{G}$.

Proof. We adopt the same proof approach as [21]. We prove that there exists a PPT adversary $\mathcal{B}_4$ s.t. $\Pr[S_3] \leq 4q \cdot d_W \cdot (L+1) \cdot \mathtt{Adv}^{\mathtt{CDH}}_{\mathcal{B}_4,\mathbb{G}}(\lambda)$, where $d_W \in \mathtt{poly}(\lambda)$ is the maximal cardinality of the set W. Let $\mathcal{A}$ denote a PPT adversary which makes the event S_3 occur with a non-negligible probability. Let $\mathcal{B}_4$ denote a PPT adversary which uses $\mathcal{A}$ to solve the CDH problem. $\mathcal{B}_4$ behaves as follows.

Receive (g, g^a, g^b) as an instance of the CDH problem. Honestly generate a key-pair $(pk_{\mathsf{s}}, sk_{\mathsf{s}})$ of the AHO scheme and a GS CRS $\boldsymbol{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$ in the perfect soundness setting. Then conduct the following two steps.

1. Set $l := 2d_W$. Choose uniformly at random an integer k satisfying $0 \leq k \leq L$. Assume that $l(L+1) \leq p$.

2. Let $h \xleftarrow{\mathrm{U}} \mathbb{G}$. Choose $x', x_0, \cdots, x_{L-1} \xleftarrow{\mathrm{U}} \mathbb{Z}_l$ and $y', y_0, \cdots, y_{L-1} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. For an element $w \in \{0,1\}^L$, define two functions $J, K : \{0,1\}^L \to \mathbb{Z}_p$ as $J(w) := x' + \sum_{i=0}^{L-1} x_i \cdot w[i] - lk$ and $K(w) := y' + \sum_{i=0}^{L-1} y_i \cdot w[i]$. Set $u' := (g^a)^{-lk+x'} \cdot g^{y'}$ and $u_i := (g^a)^{x_i} \cdot g^{y_i}$ for $i \in [0, L-1]$. It holds that $u' \prod_{i=0}^{L-1} u_i^{w[i]} = (g^a)^{-lk+x'+\sum_{i=0}^{L-1} x_i \cdot w[i]} \cdot g^{y'+\sum_{i=0}^{L-1} y_i \cdot w[i]} = (g^b)^{J(w)} \cdot g^{K(w)}$.

Set $pk := (\mathbb{G}, \mathbb{G}_T, g, h, u', \{u_i\}_{i=0}^{L-1}, pk_{\mathsf{s}}, \boldsymbol{f})$ and send it to $\mathcal{A}$. Since we have assumed that the event S_3 occurs, it must hold that $\exists \kappa \in [1, q]$ s.t. $(X_\kappa, Y_\kappa) = (X^*, Y^*) \wedge W_\kappa \not\subseteq W^*$. $\mathcal{B}_4$ guesses such an index κ by $\kappa_{guess} \xleftarrow{\mathrm{U}} [1, q]$. The guess is correct at least with probability $1/q$. $\mathcal{B}_4$ proceeds under an assumption that the guess is correct. When $\mathcal{A}$ queries to the signing oracle, $\mathcal{B}_4$ behaves as follows.

$\mathfrak{Sign}(M, W)$: Assume that this query is the κ-th query to the oracle. $\mathcal{B}_4$ considers the following two cases.

1. **$\kappa \neq \kappa_{guess}$:** $\mathcal{B}_4$ honestly generates the whole HSSM signature σ oneself, then returns it to $\mathcal{A}$.

2. **$\kappa = \kappa_{guess}$:** If $\nexists w \in W$ s.t. $J(w) = 0 \pmod p$, $\mathcal{B}_4$ aborts the simulation. Otherwise, $\exists w \in W$ s.t. $J(w) = 0 \pmod p$. Hereafter, such an element w is denoted by w'. $\mathcal{B}_4$ sets $Y := g^b$. For each $w \in W \setminus \{w'\}$, $\mathcal{B}_4$ chooses $\gamma_w \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ then computes $(\sigma_{w,1}, \sigma_{w,1}) := (H_{\mathbb{G}}(w)^{\gamma_w}, g^{\gamma_w})$. $\mathcal{B}_4$ computes $(\sigma_{w',1}, \sigma_{w',2}) := ((g^b \cdot g^{-\sum_{w \in W \setminus \{w'\}} \gamma_w})^{K(w')}, g^b \cdot g^{-\sum_{w \in W \setminus \{w'\}} \gamma_w})$. Since $J(w') = 0 \pmod p$, they distribute as $(H_{\mathbb{G}}(w')^{\gamma_{w'}}, g^{\gamma_{w'}})$, where $\gamma_{w'} := b - \sum_{w \in W \setminus \{w'\}} \gamma_w$. $\mathcal{B}_4$ honestly generates the other elements of the HSSM signature σ oneself, then returns σ to $\mathcal{A}$.

$\mathcal{B}_4$ receives a forged signature σ^* sent by $\mathcal{A}$. Since we have assumed that the event S_3 occurs, all of the following three conditions hold, namely (a) $Y^* = Y_{\kappa_{guess}}$, (b) $W^* \not\supseteq W_{\kappa_{guess}}$, and (c) *there exist* $\{\gamma_w \in \mathbb{Z}_p\}_{w \in W^*}$ *s.t.* $(\sigma_{w,1}^*, \sigma_{w,2}^*) = (H_{\mathbb{G}}(w)^{\gamma_w}, g^{\gamma_w})$ *for each* $w \in W^*$ *and* $\sum_{w \in W^*} \gamma_w = b$. $\mathcal{B}_4$ aborts the simulation if $\exists w \in W^*$ s.t. $J(w) = 0 \pmod l$. Otherwise, for each $w \in W^*$, $J(w) \neq 0 \pmod l$, which implies $J(w) \neq 0 \pmod p$. $\mathcal{B}_4$ computes $\prod_{w \in W^*} \left\{ \frac{\sigma_{w,1}^*}{(\sigma_{w,2}^*)^{K(w)}} \right\}^{\frac{1}{J(w)}} = \prod_{w \in W^*} \left\{ \frac{(g^a)^{J(w) \cdot \gamma_w} \cdot g^{K(w) \cdot \gamma_w}}{g^{\gamma_w \cdot K(w)}} \right\}^{\frac{1}{J(w)}} = \prod_{w \in W^*} (g^a)^{\gamma_w} = (g^a)^{\sum_{w \in W^*} \gamma_w} = g^{ab}$ then outputs it as the answer to the CDH problem.

Let $\mathsf{SimAbort}$ denote the event that $\mathcal{B}_4$ aborts the simulation. At least when S_3 has occurred and $\mathcal{B}_4$ has not aborted the simulation, $\mathcal{B}_4$ finds the correct answer to the CDH problem. Thus, it holds $\mathtt{Adv}_{\mathcal{B}_4, \mathbb{G}}^{\mathtt{CDH}}(\lambda) \geq \Pr[S_3 \wedge \neg \mathsf{SimAbort}] = \Pr[\neg \mathsf{SimAbort} \mid S_3] \cdot \Pr[S_3]$, implying $\Pr[S_3] \leq \frac{1}{\Pr[\neg \mathsf{SimAbort} \mid S_3]} \cdot \mathtt{Adv}_{\mathcal{B}_4, \mathbb{G}}^{\mathtt{CDH}}(\lambda)$.

We analyze the probability $\Pr[\neg \mathsf{SimAbort} \mid S_3]$. We define three events.

E_1**:** $(X^*, Y^*) = (X_{\kappa_{guess}}, Y_{\kappa_{guess}})$
E_2**:** $\exists w' \in W_{\kappa_{guess}}$ s.t. $w' \notin W^* \wedge J(w') = 0 \pmod p$
E_3**:** $\forall w \in W^*$, $J(w) \neq 0 \pmod l$

We obtain

$$\begin{aligned}
&\Pr[\neg\mathsf{SimAbort} \mid S_3] \\
&= \Pr[\mathsf{E}_1 \wedge \mathsf{E}_2 \wedge \mathsf{E}_3 \mid S_3] \\
&= \Pr[\mathsf{E}_1 \mid S_3] \cdot \Pr[\mathsf{E}_2 \wedge \mathsf{E}_3 \mid S_3 \wedge \mathsf{E}_1] \\
&= \Pr[\mathsf{E}_1 \mid S_3] \cdot \Pr[\mathsf{E}_2 \mid S_3 \wedge \mathsf{E}_1] \cdot \Pr[\mathsf{E}_3 \mid S_3 \wedge \mathsf{E}_1 \wedge \mathsf{E}_2].
\end{aligned}$$

We analyze each term. Obviously, $\Pr[\mathsf{E}_1 \mid S_3] \geq 1/q$. The second term is analyzed as follows.

$$\begin{aligned}
&\Pr[\mathsf{E}_2 \mid S_3 \wedge \mathsf{E}_1] \\
&= \Pr[J(w') = 0 \pmod p \mid S_3 \wedge \mathsf{E}_1] \\
&= \Pr[J(w') = 0 \pmod l \mid S_3 \wedge \mathsf{E}_1] \\
&\quad \cdot \Pr[J(w') = 0 \pmod p \mid S_3 \wedge \mathsf{E}_1 \wedge J(w') = 0 \pmod l] \\
&= \frac{1}{l}\frac{1}{L+1}
\end{aligned}$$

The third term is as follows.

$$\begin{aligned}
&\Pr[\mathsf{E}_3 \mid S_3 \wedge \mathsf{E}_1 \wedge \mathsf{E}_2] \\
&= \Pr\left[\bigwedge_{w \in W^*} J(w) \neq 0 \pmod l \,\middle|\, S_3 \wedge \mathsf{E}_1 \wedge \mathsf{E}_2\right] \\
&\geq 1 - \sum_{w \in W^*} \Pr[J(w) = 0 \pmod l \mid S_3 \wedge \mathsf{E}_1 \wedge \mathsf{E}_2] \\
&= 1 - \sum_{w \in W^*} \frac{1}{l} \geq 1 - \frac{d_W}{l}
\end{aligned}$$

As a result, we obtain

$$\Pr[\neg\mathsf{SimAbort} \mid S_3] \geq \frac{1}{q} \cdot \frac{1}{l} \cdot \frac{1}{L+1} \cdot \left(1 - \frac{d_W}{l}\right) = \frac{1}{4q \cdot d_W \cdot (L+1)}$$

Therefore, $\Pr[S_3] \leq 4q \cdot d_W \cdot (L+1) \cdot \mathtt{Adv}^{\mathtt{CDH}}_{\mathcal{B}_4,\mathbb{G}}(\lambda)$. □

4.2 Another Construction from the DLIN Assumption Only

By replacing the AHO SPS scheme [4,3] in the above HSSM scheme with Abe et al.'s SPS scheme [2] satisfying unforgeability against extended random messages attacks (`UF-XRMA`) (defined in Subsect. Appendix 1) under the DLIN assumption, we obtain an HSSM scheme secure under the DLIN assumption only. In the modified HSSM scheme, the signer chooses $x, y \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, then generates an Abe et al.'s SPS signature $(\theta_0, \cdots, \theta_7) \in \mathbb{G}^8$ on a message composed of six group elements $(M_1, \cdots, M_6) := (C^x, C^y, F^x, F^y, U_1^x, U_2^y) \in \mathbb{G}^6$, where $U, F, U_1, U_2 \in \mathbb{G}$ are group generators. For each element in $\{M_1, \cdots, M_6, \theta_0, \cdots, \theta_7\}$, the signer computes a GS commitment. The verification algorithm of the Abe et al.'s SPS scheme consists of seven PPEs. For each PPE, the signer computes a GS proof.

HSSB/RS Schemes	CH	DC	Assumption
ABC+12 [5] w. [19]	Strong	-	Subgroup Decision [20]
ALP12 [6]	Complete	-	DLIN
Ours	Complete	✓	DLIN

Table 1. Comparison among existing HSSB/RS schemes. CH and DC mean context-hiding and disclosure-controllability, respectively.

5 Applications

5.1 Disclosure-Controllable (DC) HSSB

In HSSB [5,6], any signature on a set M generates a signature on any subset $M' \subseteq M$. In the ordinary HSSB, any sub-message $m \in M$ can be deleted anytime. We define DCHSSB that any deletable sub-message $m \in M$ can be *undeletable* anytime. The change of deletability is one-way, which means that any undeletable sub-message cannot be made deletable again. If every sub-message is undeletable, the message is finalized.

DCHSSB is defined as follows. Given a set M, $D_M (\subseteq M)$ denotes the set of its undeletable sub-messages. From a signature on M with D_M, we can derive a signature on M' with a set $D'_{M'}$ of its undeletable sub-messages, if $M \subseteq M'$ and $D'_{M'} \supseteq D_M$. Obviously, DCHSSB is a subclass of HSSM. Specifically, DCHSSB is identical to HSSM with a restriction that *any message* (M, W) *satisfies* $M \supseteq W$.

From our HSSM scheme in Subsect 4.2, a DCHSSB scheme secure under the DLIN assumption is derived. To the best of our knowledge, there has been two HSSB schemes which satisfy all of the three conditions, namely **C1:** adaptively unforgeable[3], **C2:** strongly context-hiding, and **C3:** secure under standard assumptions. They are the scheme by Attrapadung et al. [6] secure under the DLIN assumption, and the scheme by Ahn et al. [5] instantiated with the ciphertext-policy attribute-based encryption (CP-ABE) scheme [19]. Since their disclosure-controllability have not been proven, our scheme is the first disclosure-controllable one satisfying all of the above three conditions. See Table 1.

5.2 Disclosure-Controllable Redactable Signatures (DCRS)

In redactable signatures (RS) with maximal number of sub-messages $N \in \mathbb{N}$, each message M is an ordered list in the form of $(m_1, \cdots, m_n)$ for some $n \in [1, N]$, where $m_i \in \{0,1\}^L \cup \{*\}$. Each (non-redacted) sub-message $m_i (\neq *)$ can be changed to $*$, which means it has been redacted, i.e., blacked out. RS is a subclass of $\mathcal{P}$-HS defined in Sect. 3. The predicate $\mathcal{P}_{\texttt{redact}}$ takes M and M', then outputs 1 iff $n = n' \bigwedge_{i=1}^{n} m_i \neq m'_i \implies m'_i = *$, where M' is parsed as $(m'_1, \cdots, m'_{n'})$ for some $n' \in [1, N]$. A RS scheme (parameterized by L and

[3] Our unforgeability with Def. 4 is adaptive. In selective unforgeability [5], the adversary $\mathcal{A}$ must choose the target message M^* before receiving the public-key pk.

N) can be transformed from an HSSB scheme with sub-message length $L' := L+1+\log(N+1)$ and maximal cardinality of message $K := N+1$. A RS message $M = (m_1, \cdots, m_n)$ is changed to an HSSB message $\overline{M} = \bigcup_{i \in [1,n] \text{ s.t. } m_i \neq *} \{i \parallel 0 \parallel m_i\} \bigcup \{n+1 \parallel 1^{L+1}\}$. Redacting m_i in M is deleting the element $i \parallel 0 \parallel m_i$ from $\overline{M}$[4].

In the ordinary RS, any (non-redacted) sub-message can be redacted anytime. In DCRS, any sub-message which is *non-redacted* and *redactable* can be *unredactable*. Specifically, each sub-message has one of the following three states, namely **S1**: not redacted yet and redactable, **S2**: already redacted, and **S3**: not redacted yet and unredactable. Any state only transitions from **S1** to **S2** or from **S1** to **S3**. If every sub-message is in **S2** or **S3**, the message is finalized.

DCRS is defined as follows. Each message $M = (m_1, \cdots, m_n)$ is paired with $R_M \subseteq [1, n]$ which is a set of indices for unredactable sub-messages in M. The predicate $\mathcal{P}_{\texttt{dc-redact}}$ takes (M, R_M) and $(M', R'_{M'})$, then outputs 1 iff $n = n' \bigwedge R'_{M'} \supseteq R_M \bigwedge_{i=1}^{n} m_i \neq m'_i \implies m'_i = * \wedge i \notin R_M$. Any DCHSSB scheme can be transformed into a DCRS scheme basically in the same manner as the above transformation from HSSB to RS. A RS message $M = (m_1, \cdots, m_n)$ is changed to the same HSSB message $\overline{M}$ as above. For any $i \in R_M$, the element $i \parallel 0 \parallel m_i \in \overline{M}$ is designated as an undeletable sub-message.

From our DCHSSB scheme, a DCRS scheme secure under the DLIN assumption is derived. By applying the Attrapadung et al.'s HSSB scheme [5] and the Ahn et al.'s HSSB scheme [5] instantiated with [19] to the above HSSB-to-RS transformation, we obtain secure RS schemes. Because they are not DC, ours is the first DCRS satisfying the conditions **C1**, **C2** and **C3**. See Table 1.

5.3 Efficient Superset Predicate Signatures (SPPS)

SBPS is the digital signature analogue of subset predicate encryption [17]. SPPS is the *superset* analogue of SBPS. We consider a stronger primitive in a sense that it has *key-delegatability*. SPPS is a subclass of the following key-delegatable predicate signatures (KDPS) which is formally defined in Subsect. Appendix 5.

In KDPS, setup algorithm $\texttt{Setup}$, given a security parameter 1^λ, generates a public-parameter pp and master-key mk. Key-generation $\texttt{KGen}$ generates a secret-key for a key index $X \in \mathbf{X}$. Key-delegation $\texttt{KDel}$, given a secret-key for $X \in \mathbf{X}$, generates a secret-key for a key-index $X' \in \mathbf{X}$ s.t. a key predicate $\mathcal{P}_{\mathsf{X}} : \mathbf{X}^2 \to \{0,1\}$ holds between X and X', i.e., $1 \leftarrow \mathcal{P}_{\mathsf{X}}(X, X')$. Signing algorithm $\texttt{Sig}$, given a secret-key for $X \in \mathbf{X}$, generates a signature on a message $M \in \mathcal{M}$ associated with a signature index $Y \in \mathbf{Y}$ s.t. a signature predicate $\mathcal{P}_{\mathsf{Y}} : \mathbf{X} \times \mathbf{Y} \to \{0,1\}$ holds, i.e., $1 \leftarrow \mathcal{P}_{\mathsf{Y}}(X, Y)$. Verification $\texttt{Ver}$ verifies a signature. As security for KDPS, we require unforgeability and signer-privacy.

[4] As shown in [16], RS with a fixed number of sub-messages $N \in \mathbb{N}$ can be obtained in a simpler way. A RS message $M = (m_1, \cdots, m_N)$ is changed to an HSSB message $\overline{M} = \bigcup_{i \in [1,n] \text{ s.t. } m_i \neq *} \{i \parallel m_i\}$. Redacting the sub-message m_i in M is just deleting the element $i \parallel m_i$ from $\overline{M}$.

As a concrete notion of unforgeability, we define (weak) existential unforgeability against adaptively-chosen messages and predicate attacks, abbreviated as `EUF-CMA`. We define an experiment, where a PPT adversary $\mathcal{A}$ receives an honestly-generated public-parameter pp then adaptively accesses two oracles, namely (key-)revelation and signing oracles. The former, given a key index $X \in \mathbf{X}$, returns a secret-key for X. The latter, given message $M \in \mathcal{M}$ and signature index $Y \in \mathbf{Y}$, returns a signature on M associated with Y. $\mathcal{A}$ wins the experiment if $\mathcal{A}$ succeeds in forging a correct signature σ^* on a message $M^* \in \mathcal{M}$ with $Y^* \in \mathbf{Y}$ satisfying both of the two conditions: (1) *Every* $X \in \mathbf{X}$ *queried to the (key-)revelation oracle satisfies* $0 \leftarrow \mathcal{P}_{\mathsf{Y}}(X, Y^*)$ and (2) *Every (*$M \in \mathcal{M}$*,* $Y \in \mathbf{Y}$*) queried to the signing oracle satisfies* $(M, Y) \neq (M^*, Y^*)$.

Signer-privacy guarantees that any signature reveals no information about the signer's key index X except for the fact that it satisfies the signature index Y. As a notion of signer-privacy, we define perfect signer-privacy. We define two experiments. In the real experiment, a probabilistic algorithm $\mathcal{A}$ queries an honestly-generated secret-key sk for a key index X, a message M and a signature index Y s.t. $1 \leftarrow \mathcal{P}_{\mathsf{Y}}(X, Y)$ to a signing oracle, then receives an honestly-generated signature σ. In the simulated experiment, $\mathcal{A}$ receives a signature σ which has been generated with no information about X or sk. If both experiments are hard to distinguish, the signer-privacy is satisfied.

SPPS is a subclass of KDPS. The message space is $\mathcal{M} := \{0,1\}^N$ for some $N \in \mathtt{poly}(\lambda)$. The key index space and signature index space are $\mathbf{X} := \mathbf{Y} := 2^{\{0,1\}^L}$ for some $L \in \mathtt{poly}(\lambda)$. The key predicate $\mathcal{P}_{\mathsf{X}}$, given $X, X' \in \mathbf{X}$, outputs 1 if $X' \subseteq X$ or 0 otherwise. The signature predicate $\mathcal{P}_{\mathsf{Y}}$, given $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$, outputs 1 if $Y \subseteq X$ or 0 otherwise. A SPPS scheme is obtained from an HSSM scheme with sub-message length $\max\{L, N\} + 1$ as follows.

$\mathtt{Setup}(1^\lambda, L, N)$: It generates a key-pair of the HSSM scheme, i.e., $(pp, mk) \leftarrow \text{HSSM}.\mathtt{KGen}(1^\lambda, \max\{L, N\} + 1)$, then outputs it.

$\mathtt{KGen}(mk, X)$: It generates an HSSM *signature sk* on a message

$$(M_X, W_X) := \left(\bigcup_{x \in X} \{0 \parallel x\}, \emptyset \right), \tag{8}$$

i.e., $sk \leftarrow \text{HSSM}.\mathtt{Sig}(mk, (M_X, W_X))$, then outputs it.

$\mathtt{KDel}(sk, X')$: sk is assumed to be a secret-key for $X \in \mathbf{X}$. Given an HSSM signature sk on (M_X, W_X) in (8), it derives an HSSM signature sk' on $(M_{X'}, W_{X'}) := (\cup_{x \in X'} \{0 \parallel x'\}, \emptyset)$, i.e., $sk' \leftarrow \text{HSSM}.\mathtt{Derive}(pp, (M_X, W_X), sk, (M_{X'}, W_{X'}))$, then outputs it.

$\mathtt{Sig}(sk, Y, m)$: sk is assumed to be a secret-key for $X \in \mathbf{X}$. Given an HSSM signature sk on (M_X, W_X) in (8), it derives an HSSM signature σ on

$$(M_Y, W_Y) := \left(\bigcup_{y \in Y} \{0 \parallel y\}, \bigcup_{y \in Y} \{0 \parallel y\} \bigcup \{1 \parallel m\} \right), \tag{9}$$

i.e., $\sigma \leftarrow \text{HSSM}.\mathtt{Derive}(pp, (M_X, W_X), sk, (M_Y, W_Y))$, then outputs it.

$\mathtt{Ver}(\sigma, Y, m)$**:** It verifies the HSSM signature σ on (M_Y, W_Y) in (9). It outputs $1/0 \leftarrow \text{HSSM}.\mathtt{Ver}(pp, (M_Y, W_Y), \sigma)$.

Theorem 2. *The SPPS scheme is* `PRV` *(resp.* `EUF-CMA`*) if the HSSM scheme is* `PRV` *(resp.* `SCH` *and* `wUNF`*).*

Proof. We firstly prove `PRV` then `EUF-CMA`.

The proof of `PRV` is simple. The signing oracle takes a SPPS secret-key sk for X which is an honestly-generated HSSM signature on (M_X, W_X) in (8), then honestly generates a SPPS signature σ which is an HSSM signature on (M_Y, W_Y) in (9). If the HSSM scheme is `SCH`, σ distributes like a fresh signature directly generated by the HSSM secret-key and has no information about X.

For the proof of `EUF-CMA`, we define two experiments. $\boldsymbol{Expt}_0$ is the standard `EUF-CMA` experiment w.r.t. the SPPS scheme. $\boldsymbol{Expt}_1$ is the same as $\boldsymbol{Expt}_0$ except that on the signing oracle the signature σ is directly generated by the HSSM secret-key mk. Specifically, the signing oracle, given $Y \in \mathbf{Y}$ and $m \in \mathcal{M}$, returns $\sigma \leftarrow \text{HSSM}.\mathtt{Sig}(mk, (M_Y, W_Y))$ with (M_Y, W_Y) in (9). If the HSSM scheme is `SCH`, $\boldsymbol{Expt}_1$ distributes identically to $\boldsymbol{Expt}_0$. If the HSSM scheme is `wUNF`, any PPT adversary $\mathcal{A}$ wins the experiment $\boldsymbol{Expt}_1$ only with a negligible probability.

A reduction algorithm $\mathcal{B}$ receives a public-key pp of the HSSM scheme, then gives it to $\mathcal{A}$. When $\mathcal{A}$ queries $(Y \in \mathbf{Y}, m \in \mathcal{M})$ to the signing oracle, $\mathcal{B}$ makes the signing oracle generate an HSSM signature σ on (M_Y, W_Y) in (9), then makes the signature-revelation oracle reveal it. When $\mathcal{A}$ queries $X \in \mathbf{X}$ to the key-revelation oracle, $\mathcal{B}$ makes the signing oracle generate an HSSM signature sk on (M_X, W_X) in (8), then makes the signature-revelation oracle reveal it. Consider a situation where $\mathcal{A}$ wins. For the forged SPPS signature σ^* on m^* associated with $Y^* \in \mathbf{Y}$, following three statements are true. Firstly, σ^* is a correct HSSM signature on $(M_{Y^*}, W_{Y^*}) := (\bigcup_{y \in Y^*}\{0 \parallel y\}, \bigcup_{y \in Y^*}\{0 \parallel y\} \bigcup \{1 \parallel m^*\})$. Secondly, for every X queried to the key-revelation oracle, it holds that $Y^* \not\subseteq X$, which implies that $0 \leftarrow \mathcal{P}_{\mathtt{mixed}}((M_X, W_X), (M_{Y^*}, W_{Y^*}))$. Thirdly, for every (Y, m) queried to the signing oracle, it holds that $(Y, m) \neq (Y^*, m^*)$, which implies that $0 \leftarrow \mathcal{P}_{\mathtt{mixed}}((M_Y, W_Y), (M_{Y^*}, W_{Y^*}))$. Thus, if $\mathcal{A}$ wins, then $\mathcal{B}$ also wins. □

Let us instantiate it by our HSSM scheme secure under the q-SFP and DLIN assumptions in Subsect. 4.1. Since its secret-key for X is an HSSM signature on (M_X, W_X) in (8) with $|M_X| = |X|$ and $|W_X| = 0$, it consists of $28 + 7|M_X| + 9|W_X| = 28 + 7|X|$ number of elements in $\mathbb{G}$. Since its signature for Y is an HSSM signature on (M_Y, W_Y) in (9) with $|M_Y| = |Y|$ and $|W_Y| = |Y| + 1$, it consists of $28 + 7|Y| + 9(|Y| + 1) = 37 + 16|Y|$ group elements.

In fact, SPPS can be obtained from HSSB in an inefficient and somewhat complicated manner. K denotes the maximal cardinality of X or Y. An HSSB with sub-message length $2 + \max\{L, 1 + \log N, \log K\}$ is needed. A SPPS secret-key for X is an HSSB signature on a set $M_X := \bigcup_{x \in X}\{00 \parallel x\} \bigcup_{i=0}^{|X|-1}\{01 \parallel i\} \bigcup_{j=0}^{N-1}\{10 \parallel j \parallel 0\} \bigcup_{j=0}^{N-1}\{10 \parallel j \parallel 1\}$. A SPPS signature on a message $m \in \{0,1\}^N$ associated with Y is an HSSB signature on $M_Y := \bigcup_{y \in Y}\{00 \parallel y\} \bigcup \{01 \parallel |Y| - 1\} \bigcup_{j \in [0, N-1]}\{10 \parallel j \parallel m[j]\}$. We instantiate it by our HSSM scheme in Subsect. 4.1. Its secret-key consists of $28 + 7(2|X| + 2N) + 9 \cdot 0 = 28 + 14(|X| + N)$ group elements. Its signature consists of $28 + 7(|Y| + 1 + N) + 9 \cdot 0 = 35 + 7(|Y| + N)$ group elements. Thus, its secret-key and signature lengths increase linearly with N. To the best of our knowledge, since any existing SCH-secure HSSB scheme has a property that its length of a signature on a set increases linearly with the cardinality of the set, any one of them leads to a SPPS scheme with a property that its secret-key and signature lengths increase linearly with N.

5.4 Efficient Wildcarded Identity-Based Signatures (WIBS)

WIBS is the digital signatures analogue of wildcarded identity-based encryption [1]. WIBS is a subclass of the ordinary (i.e., non key-delegatable) PS. The message space is $\mathcal{M} := \{0,1\}^N$ for some $N \in \texttt{poly}(\lambda)$. The key index space is $\mathbf{X} := (\{0,1\}^L)^n$ for some $L, n \in \texttt{poly}(\lambda)$. The signature index space is $\mathbf{Y} := (\{0,1\}^L \cup \{*\})^n$. The signature predicate $\mathcal{P}_{\mathsf{Y}}$, given $X = (x_1, \cdots, x_n) \in \mathbf{X}$ and $Y = (y_1, \cdots, y_n) \in \mathbf{Y}$, outputs 1 if $y_i \neq * \implies x_i = y_i$ for all $i \in [1, n]$, or 0 otherwise.

An HSSM scheme can be transformed into a WIBS scheme as follows.

$\texttt{Setup}(1^\lambda, L, N)$: It generates a key-pair of the HSSM scheme with sub-message length $\max\{L+\log L, N\}+1$, i.e., $(pp, mk) \leftarrow \text{HSSM}.\texttt{KGen}(1^\lambda, \max\{L+\log L, N\} + 1)$, then outputs it.

$\texttt{KGen}(mk, X)$: It generates an HSSM signature sk on a message $(M_X, W_X) := (\bigcup_{i=1}^{n} \{0 \parallel i \parallel x_i\}, \emptyset)$, i.e., $sk \leftarrow \text{HSSM}.\texttt{Sig}(mk, (M_X, W_X))$, then outputs it.

$\texttt{Sig}(sk, Y, m)$: Assume that sk is a secret-key for $X \in \mathbf{X}$. Given an HSSM signature sk on (M_X, W_X), it derives an HSSM signature σ on $(M_Y, W_Y) := (\bigcup_{i\in[1,n] \text{ s.t. } y_i \neq *} \{0 \parallel i \parallel y_i\}, \bigcup_{i\in[1,n] \text{ s.t. } y_i \neq *} \{0 \parallel i \parallel y_i\} \bigcup \{1 \parallel m\})$, i.e., $\sigma \leftarrow \text{HSSM}.\texttt{Derive}(pp, (M_X, W_X), sk, (M_Y, W_Y))$, then outputs it.

$\texttt{Ver}(\sigma, Y, m)$: It verifies the HSSM signature σ on (M_Y, W_Y). It outputs $1/0 \leftarrow \text{HSSM}.\texttt{Ver}(pp, (M_Y, W_Y), \sigma)$.

Its security is guaranteed by a theorem similar to Theorem 2. As SPPS, WIBS can also be transformed from HSSB inefficiently.

5.5 Subset Predicate Signatures (SBPS)

SBPS is a subclass of KDPS. The spaces $\mathcal{M}$, $\mathbf{X}$ and $\mathbf{Y}$ are defined as SPPS. $\mathcal{P}_{\mathsf{X}}$ takes $X, X' \in \mathbf{X}$ then outputs 1 iff $X \subseteq X'$. $\mathcal{P}_{\mathsf{Y}}$ takes $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ then outputs 1 iff $X \subseteq Y$.

Identity-based ring signatures (IBRS) [25] is a subclass of SBPS. Specifically, IBRS is identical to SBPS with the following restrictions, namely (1) there is no key-delegation and (2) cardinality of the set $X \in \mathbf{X}$ is fixed to 1. As applications of HSSP, Libert et al. [21] have mentioned only IBRS and append-only signatures [18]. In fact, SBPS can also be considered as an application of HSSP.

SBPS is transformed from HSSP as follows. By $K \in \texttt{poly}(\lambda)$, we denote the maximal cardinality of the set $X \in \mathbf{X}$ or $Y \in \mathbf{Y}$.

$\texttt{Setup}(1^\lambda, L, N)$: Generate a key-pair of the HSSP scheme with sub-message length $\max\{L, N + \log K\} + 1$, i.e., $(pp, mk) \leftarrow \text{HSSP}.\texttt{KGen}(1^\lambda, \max\{L, N + \log K\} + 1)$, then output it.

$\texttt{KGen}(mk, X)$: Output $sk \leftarrow \text{HSSP}.\texttt{Sig}(mk, W_X)$, where $W_X := \bigcup_{x \in X}\{0 \parallel x\}$.

$\texttt{KDel}(sk, X')$: Assume that sk is a for $X \in \mathbf{X}$. Output $sk' \leftarrow \text{HSSP}.\texttt{Derive}(pp, W_X, sk, W_{X'})$, where $W_{X'} := \cup_{x \in X'}\{0 \parallel x'\}$.

$\texttt{Sig}(sk, Y, m)$: Assume that sk is for $X \in \mathbf{X}$. Cardinality of the set Y is denoted by $|Y| \in [1, K]$. Output $\sigma \leftarrow \text{HSSP}.\texttt{Derive}(pp, W_X, sk, W_Y)$, where $W_Y := \bigcup_{y \in Y}\{0 \parallel y\} \bigcup \{1 \parallel m \parallel |Y|\}$.

$\texttt{Ver}(\sigma, Y, m)$: Output $1/0 \leftarrow \text{HSSP}.\texttt{Ver}(pp, W_Y, \sigma)$.

Appendix 1 Digital Signatures

Syntax. A digital signatures scheme consists of the following three polynomial time algorithms. Note that $\texttt{Ver}$ is deterministic and the others are probabilistic.

Key-Generation $\texttt{KGen}$: It takes a security parameter 1^λ for $\lambda \in \mathbb{N}$, then outputs a public-parameter pp, public-key pk and secret-key sk. $\mathcal{M}$ denotes space of messages. Assume that pp is implicitly inputted to the signing and verification algorithms. $(pp, pk, sk) \leftarrow \texttt{KGen}(1^\lambda)$

Siging $\texttt{Sig}$: It takes a secret-key sk and a message $M \in \mathcal{M}$, then outputs a signature σ. $\sigma \leftarrow \texttt{Sig}(sk, M)$

Verification $\texttt{Ver}$: It takes a public-key pk, a message $M \in \mathcal{M}$ and a signature σ, then outputs 1 or 0. $1/0 \leftarrow \texttt{Ver}(pk, M, \sigma)$

We require every digital signatures scheme to be correct. A scheme is correct if for every $\lambda \in \mathbb{N}$, every $(pp, pk, sk) \leftarrow \texttt{KGen}(1^\lambda)$, every $M \in \mathcal{M}$, every $\sigma \leftarrow \texttt{Sig}(sk, M)$, it holds that $1 \leftarrow \texttt{Ver}(pk, M, \sigma)$.

Security. The most standard unforgeability notion for digital signatures is (weak) existential unforgeability against chosen messages attacks ($\texttt{EUF-CMA}$) [14]. We consider an experiment for a PPT adversary $\mathcal{A}$ defined as follows.

```
$\boldsymbol{Expt}^{\texttt{EUF-CMA}}_{\Sigma_{\rm DS},\mathcal{A}}(1^\lambda)$:   // $\boldsymbol{Expt}^{\texttt{EUF-CMA}}_{\Sigma_{\rm DS},\mathcal{A}}$
  $(pp, pk, sk) \leftarrow \texttt{KGen}(1^\lambda)$. $Q := \emptyset$. $(M^* \in \mathcal{M}, \sigma^*) \leftarrow \mathcal{A}^{\mathfrak{Sign}}(pp, pk)$
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  - $\mathfrak{Sign}(M \in \mathcal{M})$:   $\sigma \leftarrow \texttt{Sig}(sk, M)$. $Q := Q \cup \{M\}$. $\mathbf{Rtrn}$ $\sigma$
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  $\mathbf{Rtrn}$ 1 if $1 \leftarrow \texttt{Ver}(pk, M^*, \sigma^*) \wedge M^* \notin Q$.
```

Definition 6. *A digital signatures scheme $\Sigma_{\rm DS}$ is $\texttt{EUF-CMA}$ if for any $\lambda \in \mathbb{N}$ and any PPT algorithm $\mathcal{A}$, its advantage $\boldsymbol{Adv}^{\texttt{EUF-CMA}}_{\Sigma_{\rm DS},\mathcal{A}}(\lambda) := \Pr[1 \leftarrow \boldsymbol{Expt}^{\texttt{EUF-CMA}}_{\Sigma_{\rm DS},\mathcal{A}}(1^\lambda)]$ is negligible.*

We define the other two unforgeability notions, namely unforgeability against random messages attacks ($\texttt{UF-RMA}$) [13] and unforgeability against extended random messages attacks ($\texttt{UF-XRMA}$) [2]. For $\texttt{UF-RMA}$ or $\texttt{UF-XRMA}$, we consider an experiment which is the same as the one for $\mathtt{EUF\text{-}CMA}$ except for the signing oracle. In the experiment for $\mathtt{UF\text{-}RMA}$, the signing oracle takes no input and returns a signature σ on a randomly-chosen message $M (\overset{\mathrm{U}}{\leftarrow} \mathcal{M})$. In the experiment for $\mathtt{UF\text{-}XRMA}$, the signing oracle returns not only a signature σ on randomly-chosen message $M (\overset{\mathrm{U}}{\leftarrow} \mathcal{M})$ but also some information aux about the random coins used to select M. The two notions are defined analogously to $\mathtt{EUF\text{-}CMA}$, cf. Def. 6.

Appendix 2 Non-Interactive Witness-Indistinguishable (NIWI) Proof

Syntax. An NIWI system for the NP relation $R : \{0,1\}^* \times \{0,1\}^* \rightarrow 1/0$ consists of the following 3 polynomial-time algorithms. Note that $\mathtt{Ver}$ is deterministic and the others are probabilistic.

Setup $\mathtt{Setup}$: It takes a security parameter 1^λ for $\lambda \in \mathbb{N}$, then outputs a common reference string (CRS) crs. $crs \leftarrow \mathtt{Setup}(1^\lambda)$

Proving $\mathtt{Pro}$: It takes the CRS crs, a statement $x \in \{0,1\}^*$ and a witness $w \in \{0,1\}^*$, then outputs a proof π. $\pi \leftarrow \mathtt{Pro}(crs, x, w)$

Verification $\mathtt{Ver}$: It takes the CRS crs, a statement $x \in \{0,1\}^*$ and a proof π, then outputs a verification result, which is 1 (accept) or 0 (reject). $1/0 \leftarrow \mathtt{Ver}(crs, x, \pi)$

We require every NIWI system to be correct. An NIWI system is correct if for every $\lambda \in \mathbb{N}$, every $crs \leftarrow \mathtt{Setup}(1^\lambda)$, every $x \in \{0,1\}^*$, every $w \in \{0,1\}^*$ s.t. $1 \leftarrow R(x, w)$, and every $\pi \leftarrow \mathtt{Pro}(crs, x, w)$, it holds that $1 \leftarrow \mathtt{Ver}(crs, x, \pi)$.

Security. We define two security requirements, namely perfect witness-indistinguishability ($\mathtt{WI}$) and perfect witness-extractability ($\mathtt{WE}$).

Definition 7. *An NIWI system is perfectly witness-indistinguishable ($\mathtt{WI}$), if for every $\lambda \in \mathbb{N}$, every $crs \leftarrow \mathtt{Setup}(1^\lambda)$, every $x \in \{0,1\}^*$, and every $w_0, w_1 \in \{0,1\}^*$ s.t. $1 \leftarrow R(x, w_b)$ for each $b \in \{0,1\}$, $\mathtt{Pro}(crs, x, w_0)$ distributes identically to $\mathtt{Pro}(crs, x, w_1)$.*

Definition 8. *An NIWI system is perfectly witness-extractable ($\mathtt{WE}$), if for every $\lambda \in \mathbb{N}$, there exist two algorithms $\mathtt{SimSetup}$ and $\mathtt{Extract}$ that satisfy both of the following two conditions.*

1. *For every PPT algorithm $\mathcal{A}$, $\mathtt{Adv}^{\mathtt{WE}}_{\Sigma_{\mathrm{NIWI}},\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(crs) \mid crs \leftarrow \mathtt{Setup}(1^\lambda)] - \Pr[1 \leftarrow \mathcal{A}(crs) \mid (crs, ek) \leftarrow \mathtt{SimSetup}(1^\lambda)]|$ is negligible.*
2. *For every PPT algorithm $\mathcal{A}$,*

$$\Pr\left[\begin{array}{c} (crs, ek) \leftarrow \mathtt{SimSetup}(1^\lambda); (x, \pi) \leftarrow \mathcal{A}(crs); \\ w \leftarrow \mathtt{Extract}(crs, ek, x, \pi) : 1 \leftarrow \mathtt{Ver}(crs, x, \pi) \wedge 0 \leftarrow R(x, w) \end{array}\right] = 0.$$

Appendix 3 Waters Signatures [24]

Their scheme is based on a symmetric bilinear paring $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with prime order p and generator $g \in \mathbb{G}$.

$\mathtt{KGen}(1^\lambda, L)$**:** $L \in \mathbb{N}$ denotes bit length of a message. Generate $pp := (h, u', u_0, \cdots, u_{L-1})$, where $h, u', u_0, \cdots, u_{L-1} \xleftarrow{\mathrm{U}} \mathbb{G}$. For each message $m \in \{0,1\}^L$ being parsed as $m[L-1] \parallel \cdots \parallel m[0]$, the programmable hash function $H_{\mathbb{G}} : \{0,1\}^L \to \mathbb{G}$ takes M then outputs $u' \cdot \prod_{i=0}^{L-1} u_i^{m[i]} \in \mathbb{G}$. Generate $(pk, sk) := (X, x)$, where $X := g^x$ with $x \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Output (pp, pk, sk).

$\mathtt{Sig}(sk, M)$**:** Choose $r \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Output $\sigma := (h^x \cdot H_{\mathbb{G}}(m)^r, g^r)$.

$\mathtt{Ver}(pk, M, \sigma)$**:** Parse σ as (A, B). Output 1 if $e(A, g) = e(X, h) \cdot e(H_{\mathbb{G}}(m), B)$. Output 0 otherwise.

Theorem 3. *Waters signatures scheme is* `EUF-CMA` *under the* `CDH` *assumption w.r.t. the group* $\mathbb{G}$.

Appendix 4 Abe-Haralambiev-Ohkubo (AHO) SPS [4,3]

Their scheme is based on a symmetric bilinear paring $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with prime order p and generator $g \in \mathbb{G}$.

$\mathtt{KGen}(1^\lambda, n)$**:** $n \in \mathbb{N}$ denotes the maximal number of group elements to be signed. Choose generators $g_r, h_r \xleftarrow{\mathrm{U}} \mathbb{G}$. Let $pp := (g_r, h_r)$.
Choose elements $\gamma_z, \delta_z, \alpha_a, \alpha_b \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and $\gamma_i, \delta_i \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ for $i \in [1, n]$. Compute $g_z := g_r^{\gamma_z}$, $h_z := h_r^{\delta_z}$, $A := e(g_r, g^{\alpha_a})$, $B := e(h_r, g^{\alpha_b})$, and $g_i := g_r^{\delta_i}$ and $h_i := h_r^{\delta_i}$ for $i \in [1, n]$. Output (pp, pk, sk), where $pk := (g_z, h_z, \{g_i, h_i\}_{i=1}^n, A, B)$ and $sk := (\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n)$.

$\mathtt{Sig}(sk, (M_1, \cdots, M_n))$**:** Choose $\eta, \rho_a, \rho_b, \omega_a, \omega_b \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Compute $\theta_1 := g^\eta$ and

$$\theta_2 := g^{\rho_a - \gamma_z \eta} \cdot \prod_{i=1}^n M_i^{\gamma_i}, \quad \theta_3 := g_r^{\omega_a}, \quad \theta_4 := g^{(\alpha_a - \rho_a)/\omega_a},$$

$$\theta_5 := g^{\rho_b - \delta_z \eta} \cdot \prod_{i=1}^n M_i^{\delta_i}, \quad \theta_6 := h_r^{\omega_b}, \quad \theta_7 := g^{(\alpha_b - \rho_b)/\omega_b}.$$

Output a signature $\sigma := (\theta_1, \cdots, \theta_7)$.

$\mathtt{Ver}(pk, (M_1, \cdots, M_n), \sigma)$**:** Output 1 if both of the following two equations hold, namely $A = e(g_z, \theta_1) \cdot e(g_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot \prod_{i=1}^n e(g_i, M_i)$ and $B = e(h_z, \theta_1) \cdot e(h_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot \prod_{i=1}^n e(h_i, M_i)$.

As shown in [4,3], any signature $\sigma = (\theta_1, \cdots, \theta_n)$ can be publicly randomized as follows. The first element is unchanged, i.e., $\theta_1' := \theta_1$. We choose $\eta_2, \eta_5, \mu, \nu \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ then compute

$$\theta_2' := \theta_2 \cdot \theta_4^{\eta_2}, \quad \theta_3' := (\theta_3 \cdot g_r^{-\eta_2})^{1/\mu}, \quad \theta_4' := \theta_4^{\mu},$$

$$\theta_5' := \theta_5 \cdot \theta_7^{\eta_5}, \quad \theta_6' := (\theta_6 \cdot h_r^{-\eta_5})^{1/\nu}, \quad \theta_7' := \theta_7^{\nu}.$$

According to [4,3], $(\theta_2', \theta_3', \theta_4') \in \mathbb{G}^3$ uniformly distribute under a restriction that $e(g_r, \theta_2') \cdot e(\theta_3', \theta_4') = e(g_r, \theta_2) \cdot e(\theta_3, \theta_4)$, and $(\theta_5', \theta_6', \theta_7') \in \mathbb{G}^3$ uniformly distribute under a restriction that $e(h_r, \theta_5') \cdot e(\theta_6', \theta_7') = e(h_r, \theta_5) \cdot e(\theta_6, \theta_7)$.

Theorem 4. *Let $q \in \mathtt{poly}(\lambda)$ denote the maximal number of signing queries. The signature scheme is* EUF-CMA *under the q-SFP assumption.*

Appendix 5 Key-Delegatable Predicate Signatures

Key-delegatable predicate signatures (KDPS) consists of the five polynomial-time algorithms. Ver is deterministic and the others are probabilistic.

Setup Setup: It takes 1^λ, then outputs a public parameter *pp* and master-key *mk*. $\mathbf{X}$, $\mathbf{Y}$ and $\mathcal{M}$ denote the space of key index, signature index and message, respectively. Note that the other algorithms implicitly take *pp* as input. $(pp, mk) \leftarrow \mathtt{Setup}(1^\lambda)$

Key-Generation KGen: It takes *mk* and a key index $X \in \mathbf{X}$, then outputs a secret-key *sk*. $sk \leftarrow \mathtt{KGen}(mk, X)$

Key-Delegation KDel: It takes a secret-key *sk* for a key index $X \in \mathbf{X}$ and a key index $X' \in \mathbf{X}$ s.t. $1 \leftarrow \mathcal{P}_{\mathsf{X}}(X, X')$, then outputs a secret-key sk'. $sk' \leftarrow \mathtt{KDel}(sk, X')$

Signing Sig: It takes a secret-key *sk* for a key index $X \in \mathbf{X}$, a signature index $Y \in \mathbf{Y}$ s.t. $1 \leftarrow \mathcal{P}_{\mathsf{Y}}(X, Y')$ and a message $M \in \mathcal{M}$, then outputs a signature σ. $\sigma \leftarrow \mathtt{Sig}(sk, Y, M)$

Verification Ver: It takes a signature σ, a signature index $Y \in \mathbf{Y}$ and a message $M \in \mathcal{M}$, then outputs 1 or 0. $1/0 \leftarrow \mathtt{Ver}(\sigma, Y, M)$

Every KDPS scheme must be correct. Informally the property means that every correctly generated signature is accepted. Formally the property is defined as follows. A KDPS scheme is correct if $\forall \lambda \in \mathbb{N}$, $\forall (pp, mk) \leftarrow \mathtt{Setup}(1^\lambda)$, $\forall X \in \mathbf{X}$, $\forall sk \leftarrow \mathtt{KGen}(mk, X)$, $\forall X' \in \mathbf{X}$ s.t. $1 \leftarrow \mathcal{P}_{\mathsf{X}}(X, X')$, $\forall sk' \leftarrow \mathtt{KDel}(sk, X')$, $\forall Y \in \mathbf{Y}$ s.t. $1 \leftarrow \mathcal{P}_{\mathsf{Y}}(X', Y)$, $\forall M \in \mathcal{M}$, $\forall \sigma \leftarrow \mathtt{Sig}(sk', Y, M)$, $1 \leftarrow \mathtt{Ver}(\sigma, Y, M)$ holds.

As security for KDPS, we require unforgeability and signer-privacy. As a notion of unforgeability, we define *(weak) existential unforgeability against adaptively-chosen messages and predicate attack* (EUF-CMA). For a PPT algorithm $\mathcal{A}$, we consider the following experiment.

```
$\boldsymbol{Expt}_{\Sigma_{\mathrm{KDPS}},\mathcal{A}}^{\mathtt{EUF\text{-}CMA}}(1^\lambda)$:
 1. $(pp, mk) \leftarrow \mathtt{Setup}(1^\lambda)$. $(\sigma^*, Y^* \in \mathbf{Y}, M^* \in \mathcal{M}) \leftarrow \mathcal{A}^{\mathfrak{Reveal},\mathfrak{Sign}}(pp)$.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - $\mathfrak{Reveal}(X \in \mathbf{X})$:  $sk \leftarrow \mathtt{KGen}(mk, X)$. $Q := Q \cup \{X\}$. $\mathbf{Rtrn}$ $sk$.
    - $\mathfrak{Sign}(X \in \mathbf{X}, Y \in \mathbf{Y}, M \in \mathcal{M})$:  $sk \leftarrow \mathtt{KGen}(mk, X)$. $\sigma \leftarrow \mathtt{Sig}(sk, M, Y)$.
      $Q' := Q' \cup \{(Y, M, \sigma)\}$. $\mathbf{Rtrn}$ $\sigma$.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 2. $\mathbf{Rtrn}$ 1 if (1) $1 \leftarrow \mathtt{Ver}(\sigma^*, Y^*, M^*)$, (2) $\forall X \in Q$, $0 \leftarrow \mathcal{P}_{\mathsf{Y}}(X, Y^*)$
      and (3) $(Y^*, M^*, \cdot) \notin Q'$.
 3. $\mathbf{Rtrn}$ 0.
```

Definition 9. *A KDPS scheme Σ_{KDPS} is* ***EUF-CMA*** *if for every $\lambda \in \mathbb{N}$ and every PPT $\mathcal{A}$, $\mathcal{A}$'s advantage $\boldsymbol{Adv}^{\textit{EUF-CMA}}_{\Sigma_{\mathrm{KDPS}},\mathcal{A}}(\lambda) := \Pr[1 \leftarrow \boldsymbol{Expt}^{\textit{EUF-CMA}}_{\Sigma_{\mathrm{KDPS}},\mathcal{A}}(1^\lambda)]$ is negligible.*

As a notion of signer-privacy, we define perfect signer-privacy ($\texttt{PRV}$). For a probabilistic algorithm $\mathcal{A}$, we consider the following two experiments.

```
$\boldsymbol{Expt}^{\texttt{PRV}}_{\Sigma_{\mathrm{KDPS}},\mathcal{A},0}(1^\lambda)$:  $//\boldsymbol{Expt}^{\texttt{PRV}}_{\Sigma_{\mathrm{KDPS}},\mathcal{A},1}$
  $(pp, mk) \leftarrow \texttt{Setup}(1^\lambda)$. $(pp, mk, \mu) \leftarrow \texttt{SimSetup}(1^\lambda)$.
  **Rtrn** $b' \leftarrow \mathcal{A}^{\mathfrak{Reveal},\mathfrak{Sign}}(pp, mk)$.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - $\mathfrak{Reveal}(X \in \mathbf{X})$:
      $sk \leftarrow \texttt{KGen}(mk, X)$. $sk \leftarrow \texttt{SimKGen}(mk, \mu, X)$. $Q := Q \cup \{(X, sk)\}$. **Rtrn** $sk$.
    - $\mathfrak{Sign}(X \in \mathbf{X}, sk, Y \in \mathbf{Y}, M \in \mathcal{M})$:
      **Rtrn** $\perp$ if $(X, sk) \notin Q \vee 0 \leftarrow \mathcal{P}_{\mathsf{Y}}(X, Y)$.
      $\sigma \leftarrow \texttt{Sig}(sk, Y, M)$. $\sigma \leftarrow \texttt{SimSig}(mk, \mu, Y, M)$. **Rtrn** $\sigma$.
```

The latter is associated with 3 polynomial-time algorithms $\{\texttt{SimSetup}, \texttt{SimKGen}, \texttt{SimSig}\}$. The grey parts are considered in the latter, but ignored in the former.

Definition 10. *A KDPS scheme Σ_{KDPS} is perfectly signer-private (PRV) if for every $\lambda \in \mathbb{N}$ and every probabilistic algorithm $\mathcal{A}$, there exist polynomial-time algorithms $\{\texttt{SimSetup}, \texttt{SimKGen}, \texttt{SimSig}\}$ such that $\mathcal{A}$'s advantage $\boldsymbol{Adv}^{\textit{PRV}}_{\Sigma_{\mathrm{KDPS}},\mathcal{A}}(\lambda) := |\sum_{b=0}^{1}(-1)^b \Pr[1 \leftarrow \boldsymbol{Expt}^{\textit{PRV}}_{\Sigma_{\mathrm{KDPS}},\mathcal{A},b}(1^\lambda)]|$ is 0.*

Appendix 6 An Omitted Part in the Proof of Lemma 3

We prove that there exists a PPT adversary $\mathcal{B}_3'$ s.t. $\Pr[S_2 \wedge \mathsf{Abort}] \leq 4q \cdot d_M \cdot (L+1) \cdot \texttt{Adv}^{\texttt{CDH}}_{\mathcal{B}_3',\mathbb{G}}(\lambda)$, where $d_M \in \texttt{poly}(\lambda)$ denotes the maximal cardinality of the set M.

Let $\mathcal{A}$ denote a PPT adversary which makes the event $S_2 \wedge \mathsf{Abort}$ occur with a non-negligible probability. Let $\mathcal{B}_3'$ denote a PPT adversary which, by using $\mathcal{A}$ as black-box, attempts to solve the CDH problem relative to the group $\mathbb{G}$. $\mathcal{B}_3'$ behaves as follows.

Receive (g, g^a, g^b) as an instance of the CDH problem. Honestly generate a key-pair $(pk_{\mathsf{s}}, sk_{\mathsf{s}})$ of the AHO scheme and a GS CRS $\boldsymbol{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$ in the perfect soundness setting. Then conduct the following two steps.

1. Set $l := 2d_M$. Choose uniformly at random an integer k satisfying $0 \leq k \leq L$. Assume that $l(L+1) \leq p$.
2. Let $h := g^a$. Choose $x', x_0, \cdots, x_{L-1} \xleftarrow{\mathrm{U}} \mathbb{Z}_l$ and $y', y_0, \cdots, y_{L-1} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. For an element $m \in \{0,1\}^L$, define two functions $J, K : \{0,1\}^L \to \mathbb{Z}_p$ as $J(m) := x' + \sum_{i=0}^{L-1} x_i \cdot m[i] - lk$ and $K(m) := y' + \sum_{i=0}^{L-1} y_i \cdot m[i]$. Set $u' := (g^b)^{-lk+x'} \cdot g^{y'}$ and $u_i := (g^b)^{x_i} \cdot g^{y_i}$ for $i \in [0, L-1]$. It holds that $u' \prod_{i=0}^{L-1} u_i^{m[i]} = (g^b)^{-lk+x'+\sum_{i=0}^{L-1} x_i \cdot m[i]} \cdot g^{y'+\sum_{i=0}^{L-1} y_i \cdot m[i]} = (g^b)^{J(m)} \cdot g^{K(m)}$.

Set $pk := (\mathbb{G}, \mathbb{G}_T, g, h, u', \{u_i\}_{i=0}^{L-1}, pk_{\mathsf{s}}, \boldsymbol{f})$ and send it to $\mathcal{A}$. Since we have assumed that the event $S_2 \wedge \mathsf{Abort}$ occurs, it will hold that $\exists \kappa \in [1, q]$ s.t.

$(X_\kappa, Y_\kappa) = (X^*, Y^*) \wedge M_\kappa \not\supseteq M^*$. $\mathcal{B}'_3$ guesses such an index κ and chooses $\kappa_{guess} \overset{\text{U}}{\leftarrow} [1, q]$. The guess is correct at least with probability $1/q$. $\mathcal{B}'_3$ proceeds under an assumption that the guess is correct. When $\mathcal{A}$ issues a query to the signing oracle, $\mathcal{B}'_3$ behaves as follows.

$\mathfrak{Sign}(M, W)$: Assume that this query is the κ-th query to the oracle. $\mathcal{B}'_3$ considers the following two cases.

1. **$\kappa \neq \kappa_{guess}$:** $\mathcal{B}'_3$ honestly generates the whole HSSM signature σ oneself, then returns it to $\mathcal{A}$.

2. **$\kappa = \kappa_{guess}$:** If $\exists m \in M$ s.t. $J(m) = 0 \pmod{l}$, $\mathcal{B}'_3$ aborts the simulation. Otherwise, every $m \in M$ satisfies $J(m) \neq 0 \pmod{l}$, which implies $J(m) \neq 0 \pmod{p}$. $\mathcal{B}'_3$ sets $X := g^b$. Then, for each $m \in M$, $\mathcal{B}'_3$ behaves as follows. We have assumed that it holds that $J(m) \neq 0 \pmod{p}$. $\mathcal{B}'_3$ chooses $\chi_m \overset{\text{U}}{\leftarrow} \mathbb{Z}_p$, then generates $(\sigma_{m,1}, \sigma_{m,2}) := ((g^a)^{\frac{K(m)}{J(m)}} (u' \prod_{i=0}^{L-1} u_i^{m[i]})^{\chi_m}, (g^a)^{-\frac{1}{J(m)}} g^{\chi_m})$. Let $\chi'_m := \chi_m - \frac{a}{J(m)}$. Obviously, $\sigma_{m,2} = g^{\chi'_m}$. It holds that $\sigma_{m,1} = g^{ab} \cdot \left\{(g^b)^{J(m)} \cdot g^{K(m)}\right\}^{-\frac{a}{J(m)}} \cdot \left\{(g^b)^{J(m)} \cdot g^{K(m)}\right\}^{\chi_m} = g^{ab} \cdot \left\{(g^b)^{J(m)} \cdot g^{K(m)}\right\}^{\chi'_m} = h^b \cdot H_{\mathbb{G}}(m)^{\chi'_m}$. Thus, the Waters signature $(\sigma_{m,1}, \sigma_{m,2})$ correctly distributes. $\mathcal{B}'_3$ honestly generates the other elements of the HSSM signature σ, then returns σ to $\mathcal{A}$.

$\mathcal{B}'_3$ receives a forged signature σ^* sent by $\mathcal{A}$. Since we have assumed that the event $S_2 \wedge \mathsf{Abort}$ occurs, all of the following three conditions hold, namely (a) $X^* = X_{\kappa_{guess}}$, (b) $M^* \not\subseteq M_{\kappa_{guess}}$, and (c) *for each* $m \in M^*$*,* $(\sigma^*_{m,1}, \sigma^*_{m,2}) = (h^b \cdot H_{\mathbb{G}}(m)^{\chi_m}, g^{\chi_m})$ *with some* $\chi_m \in \mathbb{Z}_p$.

The second condition (b) implies that $\exists m^* \in M^*$ s.t. $m^* \notin M_{\kappa_{guess}}$. $\mathcal{B}'_3$ arbitrarily chooses a single element m^* satisfying the above condition, then aborts the simulation if $J(m^*) \neq 0 \pmod{p}$. $\mathcal{B}'_3$ computes $\sigma^*_{m^*,1}/(\sigma^*_{m^*,2})^{K(m^*)} = h^b \cdot H_{\mathbb{G}}(m^*)^{\chi_{m^*}}/(g^{K(m^*)})^{\chi_{m^*}} = h^b = g^{ab} \in \mathbb{G}$ then outputs it as the answer to the CDH problem.

Let $\mathsf{SimAbort}$ denote the event that $\mathcal{B}'_3$ aborts the simulation. At least when the event $S_2 \wedge \mathsf{Abort}$ has occurred and $\mathcal{B}'_3$ has not aborted the simulation, $\mathcal{B}'_3$ finds the correct answer to the CDH problem. Thus, it holds that

$$\begin{aligned}\mathtt{Adv}^{\mathtt{CDH}}_{\mathcal{B}'_3, \mathbb{G}}(\lambda) &\geq \Pr[S_2 \wedge \mathsf{Abort} \wedge \neg\mathsf{SimAbort}] \\ &= \Pr[\neg\mathsf{SimAbort} \mid S_2 \wedge \mathsf{Abort}] \cdot \Pr[S_2 \wedge \mathsf{Abort}],\end{aligned}$$

which implies that

$$\Pr[S_2 \wedge \mathsf{Abort}] \leq \frac{1}{\Pr[\neg\mathsf{SimAbort} \mid S_2 \wedge \mathsf{Abort}]} \cdot \mathtt{Adv}^{\mathtt{CDH}}_{\mathcal{B}'_3, \mathbb{G}}(\lambda).$$

We analyze the probability $\Pr[\neg\mathsf{SimAbort} \mid S_2 \wedge \mathsf{Abort}]$. We define three events.

E_1: $(X^*, Y^*) = (X_{\kappa_{guess}}, Y_{\kappa_{guess}})$
E_2: $\forall m \in M_{\kappa_{guess}}$, $J(m) \neq 0 \pmod{l}$

E_3**:** $\exists m^* \in M^*$ s.t. $m^* \notin M_{\kappa_{guess}} \wedge J(m^*) = 0 \pmod p$

We obtain

$$\begin{aligned}
&\Pr[\neg\mathsf{SimAbort} \mid S_2 \wedge \mathsf{Abort}] \\
&= \Pr[\mathsf{E}_1 \wedge \mathsf{E}_2 \wedge \mathsf{E}_3 \mid S_2 \wedge \mathsf{Abort}] \\
&= \Pr[\mathsf{E}_1 \mid S_2 \wedge \mathsf{Abort}] \cdot \Pr[\mathsf{E}_2 \wedge \mathsf{E}_3 \mid S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1] \\
&= \Pr[\mathsf{E}_1 \mid S_2 \wedge \mathsf{Abort}] \cdot \Pr[\mathsf{E}_3 \mid S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1] \cdot \Pr[\mathsf{E}_2 \mid S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1 \wedge \mathsf{E}_3].
\end{aligned}$$

We analyze each term. Obviously, $\Pr[\mathsf{E}_1 \mid S_2 \wedge \mathsf{Abort}] \geq 1/q$. The second term is analyzed as follows.

$$\begin{aligned}
&\Pr[\mathsf{E}_3 \mid S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1] \\
&= \Pr[J(m^*) = 0 \pmod p \mid S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1] \\
&= \Pr[J(m^*) = 0 \pmod l \mid S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1] \\
&\quad \cdot \Pr[J(m^*) = 0 \pmod p \mid S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1 \wedge J(m^*) = 0 \pmod l] \\
&= \frac{1}{l}\frac{1}{L+1}
\end{aligned}$$

The third term is as follows.

$$\begin{aligned}
&\Pr[\mathsf{E}_2 \mid S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1 \wedge \mathsf{E}_3] \\
&= \Pr\left[\bigwedge_{m \in M_{\kappa_{guess}}} J(m) \neq 0 \pmod l \,\middle|\, S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1 \wedge \mathsf{E}_3\right] \\
&\geq 1 - \sum_{m \in M_{\kappa_{guess}}} \Pr[J(m) = 0 \pmod l \mid S_2 \wedge \mathsf{Abort} \wedge \mathsf{E}_1 \wedge \mathsf{E}_3] \\
&= 1 - \sum_{m \in M_{\kappa_{guess}}} \frac{1}{l} \geq 1 - \frac{d_M}{l}
\end{aligned}$$

As a result, we obtain

$$\Pr[\neg\mathsf{SimAbort} \mid S_2 \wedge \mathsf{Abort}] \geq \frac{1}{q} \cdot \frac{1}{l} \cdot \frac{1}{L+1} \cdot \left(1 - \frac{d_M}{l}\right) = \frac{1}{4q \cdot d_M \cdot (L+1)}$$

Therefore, $\Pr[S_2 \wedge \mathsf{Abort}] \leq 4q \cdot d_M \cdot (L+1) \cdot \mathtt{Adv}^{\mathtt{CDH}}_{\mathcal{B}'_3,\mathbb{G}}(\lambda)$. □

References

1. M. Abdalla, D. Catalano, A.W. Dent, J. Malone-Lee, G. Neven, and N.P. Smart. Identity-based encryption gone wild. In *ICALP 2006*, pp. 300–311. Springer, 2006.
2. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In *ASIACRYPT 2012*, pp. 4–24. Springer, 2012.

3. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *CRYPTO 2010*, pp. 209–236. Springer, 2010.
4. M. Abe, K. Haralambiev, and M. Ohkubo. Signing on elements in bilinear groups for modular protocol design. *Cryptology ePrint Archive*, 2010.
5. J.H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat, and B. Waters. Computing on authenticated data. In *TCC 2012*, pp. 1–20. Springer, 2012.
6. N. Attrapadung, B. Libert, and T. Peters. Computing on authenticated data: New privacy definitions and constructions. In *ASIACRYPT 2012*, pp. 367–385. Springer, 2012.
7. N. Attrapadung, B. Libert, and T. Peters. Efficient completely context-hiding quotable and linearly homomorphic signatures. In *PKC 2013*, pp. 386–404. Springer, 2013.
8. J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE SP 2007*, pp. 321–334. IEEE, 2007.
9. J. Blömer, F. Eidens, and J. Juhnke. Enhanced security of attribute-based signatures. In *CANS 2018*, pp. 235–255. Springer, 2018.
10. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO 2004*, pp. 41–55. Springer, 2004.
11. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *ASIACRYPT 2002*, pp. 514–532. Springer, 2001.
12. X. Bultel, P. Lafourcade, R. Lai, G. Malavolta, D. Schröder, S. Aravinda, and K. Thyagarajan. Efficient invisible and unlinkable sanitizable signatures. In *PKC 2019*, pp. 159–189. Springer, 2019.
13. S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9:35–67, 1996.
14. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
15. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT 2008*, pp. 415–432. Springer, 2008.
16. M. Ishizaka, K. Fukushima, and S. Kiyomoto. Trapdoor sanitizable and redactable signatures with unlinkability, invisibility and strong context-hiding. In *ICISC 2022*, pp. 337–362. Springer, 2022.
17. J. Katz, M. Maffei, G. Malavolta, and D. Schröder. Subset predicate encryption and its applications. In *CANS 2017*, pp. 115–134. Springer, 2017.
18. E. Kiltz, A. Mityagin, S. Panjwani, and B. Raghavan. Append-only signatures. In *ICALP 2005*, pp. 434–445. Springer, 2005.
19. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT 2010*, pp. 62–91. Springer, 2010.
20. A. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC 2010*, volume 5978 of LNCS, pp. 455–479. Springer, 2010.
21. B. Libert, M. Joye, M. Yung, and T. Peters. Secure efficient history-hiding append-only signatures in the standard model. In *PKC 2015*, pp. 450–473. Springer, 2015.
22. H.K. Maji, M. Prabhakaran, and M. Rosulek. Attribute-based signatures. In *CT-RSA 2011*, pp. 376–392. Springer, 2011.
23. R. Steinfeld, L. Bull, and Y. Zheng. Content extraction signatures. In *ICISC 2001*, pp. 285–304. Springer, 2001.

24. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT 2005*, pp. 114–127. Springer, 2005.
25. F. Zhang and K. Kim. Id-based blind signature and ring signature from pairings. In *ASIACRYPT 2002*, pp. 533–547. Springer, 2002.