Key-Range Attribute-Based Signatures for Range of Inner Product and Its Applications

Masahito Ishizaka

KDDI Research, Inc., Saitama, Japan. xma-ishizaka@kddi.com

Abstract. In attribute-based signatures (ABS) for range of inner product (ARIP), recently proposed by Ishizaka and Fukushima at ICISC 2022, a secret-key labeled with an n-dimensional vector $\mathbf{x} \in \mathbb{Z}_p^n$ for a prime p can be used to sign a message under an n-dimensional vector $\mathbf{y} \in \mathbb{Z}_p^n$ and a range $[L, R] = \{L, L+1, \cdots, R-1, R\}$ with $L, R \in \mathbb{Z}_p$ iff their inner product is within the range, i.e., $\langle \mathbf{x}, \mathbf{y} \rangle \in [L, R] \pmod p$. We consider its key-range version, named key-range ARIP (KARIP), where the range $[L, R]$ is associated with a secret-key but not with a signature. We propose three generic KARIP constructions based on linearly homomorphic signatures and non-interactive witness-indistinguishable proof, which lead to concrete KARIP instantiations secure under standard assumptions with different features in terms of efficiency. We also show that KARIP has various applications, e.g., key-range ABS for range evaluation of polynomials/weighted averages/Hamming distance/Euclidean distance, key-range time-specific signatures, and key-range ABS for hyperellipsoid predicates.

Keywords: Key-Range attribute-based signatures for range of inner product, Adaptive unforgeablity, Signer-privacy, Key-delegatability.

1 Introduction

Attribute-Based Encryption (ABE) for Inner Products. In ABE for inner products [11], n-dimensional vector $\mathbf{x} \in \mathbb{Z}_p^n$ (resp. $\mathbf{y} \in \mathbb{Z}_p^n$) for a prime p is associated with secret-key (resp. ciphertext). The decryption succeeds iff $\langle \mathbf{x}, \mathbf{y} \rangle = 0 \pmod p$. It can be generically transformed into various ABE primitives, e.g., (anonymous) identity-based encryption (IBE), hidden-vector encryption (HVE) [6], the dual variant of HVE (= wildcarded IBE [1]), ABE for evaluation of polynomials/weighted averages, ABE for conjunctive/disjunctive normal form (CNF/DNF) formulas, and ABE for exact thresholds.

Attribute-Based Signatures for Range of Inner Product (ARIP) [9]. ARIP is a generalization of attribute-based signatures (ABS) for inner products which is the digital signature version of the above ABE for inner products. A secret-key associated with an n-dimensional vector $\mathbf{x} \in \mathbb{Z}_p^n$ is used to sign a message M under an n-dimensional vector $\mathbf{y} \in \mathbb{Z}_p^n$ and a range $[L, R]$ with $L, R \in \mathbb{Z}_p$.

The signing succeeds iff $\langle \mathbf{x}, \mathbf{y} \rangle \in [L, R] \pmod{p}$. Two security requirements are defined, unforgeability and signer-privacy. The latter means that any signature leaks no more information about $\mathbf{x}$ than the fact that its inner product with $\mathbf{y}$ is in the range $[L, R]$. ARIP has various applications. An ARIP scheme can be transformed into any of the following ABS primitives, ABS for range evaluation (RE) of polynomials (AREP), ABS for RE of weighted averages (AREWA), fuzzy identity-based signatures (FIBS), time-specific signatures (TSS) [13,10], ABS for RE of Hamming distance (AREHD), ABS for RE of Euclidean distance (AREED) and ABS for hyperellipsoid predicates (AHEP).

In this paper, we consider its key-range version, named key-range ARIP (KARIP). The range $[L, R]$ is associated with a secret-key but not with a signature. The ABS scheme by Sakai et al. [15] supporting any circuit as signer-predicate can be a KARIP scheme by properly configuring the circuit. Both a vector $\mathbf{x} \in \mathbb{Z}_p^n$ and a range $[L, R]$ are transformed into a binary attribute $x \in \{0,1\}^{(n+2)\cdot\lambda}$. In their ABS scheme, at signature generation, a signer generates a commitment of the non-interactive witness indistinguishable proof (NIWI) system by Groth and Sahai (GS) [8] for each bit $x[i] \in \{0,1\}$ of x. Thus, at least, its signature length linearly increases with $n\lambda$.

1.1 Contribution

In this work, we propose three generic constructions of KARIP, which lead to three concrete KARIP schemes with distinct features in terms of efficiency and key-delegatability. We show that KARIP has various applications.

1st Construction. It is generically constructed by NIWI, linearly homomorphic signatures (LHS)[1] [5] and append-only signatures (AOS)[2] [12]. In key-generation for $(\mathbf{x}, L, R)$, we choose an LHS tag τ, then define $n+2$ vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{n+2} \in \mathbb{Z}_p^{n+3}$ as $\boldsymbol{v}_i := (x_i, \underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{n-i}, 0, 0)$ for each $i \in [1, n]$, $\boldsymbol{v}_{n+1} := (0, \cdots, 0, 1, 0)$ and $\boldsymbol{v}_{n+2} := (0, \cdots, 0, 0, 1)$. We generate an LHS signature σ_i on each vector $\boldsymbol{v}_i$ under the common tag τ. In signing for $(\mathbf{y}, M)$, they are used to derive an LHS signature σ' with the same tag τ on $\boldsymbol{v}' := (\langle \mathbf{x}, \mathbf{y} \rangle, y_1, \cdots, y_n, M)$. In key-generation, we also generate AOS signatures. We consider a complete binary tree with p leaf nodes. C denotes the set of intermediate nodes covering all of the leaf nodes associated with from $[L]_2$ to $[R]_2$, where $[a]_2$ is the binary value of a. For each $c \in C$, parsed as $c[1] \,\|\, \cdots \,\|\, c[h_c]$ with $c[i] \in \{0,1\}$ and length $h_c \in [1, \lambda]$, we generate an AOS signature θ_c on $(\tau, c[1], \cdots, c[h_c]) \in (\{0,1\}^N)^{h_c+1}$. In signing, one of the AOS signatures is used to generate an AOS signature θ' on $(\tau, \langle \mathbf{x}, \mathbf{y} \rangle[1], \cdots, \langle \mathbf{x}, \mathbf{y} \rangle[\lambda]) \in (\{0,1\}^N)^{\lambda+1}$. If $\langle \mathbf{x}, \mathbf{y} \rangle \in [L, R]$, there exists a node $c \in C$ s.t. c is whether identical to or an ancestor of $\langle \mathbf{x}, \mathbf{y} \rangle$. An AOS signature θ_c for such a

[1] In LHS, l signatures $\{\sigma_i\}_{i=1}^l$ on vectors $\{\boldsymbol{v}_i\}_{i=1}^l$ associated with the common tag τ make us derive a signature on any linear summation $\sum_{i=1}^l \beta_i \cdot \boldsymbol{v}_i$ with same tag τ.

[2] In AOS, each message has a hierarchical structure. Any signature on a message M makes us derive a new signature on any descendant message M'.

node c derives θ'. Finally, we generate an NIWI proof that both of the LHS and AOS signatures σ', θ' are correct under the witness $(\langle \mathbf{x}, \mathbf{y} \rangle, \tau, \sigma', \theta')$. Clearly, our 1st construction is key-delegatable because of the message-appendability of the underlying AOS.

To instantiate it, we use the simplified ALP LHS scheme [9] and the GS proof [8]. As AOS, we search for a candidate satisfying both of the following conditions, namely (1) *Based on symmetric bilinear paring with prime order* and (2) *Its verification algorithm consists of only PPEs*. We refer to an hierarchical identity-based signatures scheme in by Chatterjee and Sarkar [7] to construct an original AOS scheme satisfying the conditions and rigorously prove its security, i.e., unforgeability, under the CDH assumption. To evaluate efficiency of the instantiated scheme, we rigorously calculate its secret-key and signature sizes. They are $N + \big(n + \lambda^2\big)|g|$ [bit] and $(27N + 27\lambda + 40)|g|$ [bit], where $|g|$ denotes bit length of an element in the bilinear group $\mathbb{G}$.

2nd Construction. It is generically constructed by LHS and NIWI. This construction is similar to the 1st ARIP scheme in [9]. In key-generation for $(\mathbf{x}, L, R)$, we define $n+2$ vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{n+2} \in \mathbb{Z}_p^{n+5}$ as $\boldsymbol{v}_i := (x_i, \underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{n-i}, 0, 0, 0, 0)$ for each $i \in [1, n]$, $\boldsymbol{v}_{n+1} := (0, \cdots, 0, L, R, 0, 1)$ and $\boldsymbol{v}_{n+2} := (0, \cdots, 0, 0, 0, 1, 0)$. For each vector $\boldsymbol{v}_i$, we generate an LHS signature σ_i with the common tag τ. In signing for $(\mathbf{y}, M)$, an LHS signature σ' with the tag τ on $\boldsymbol{v}' := (\langle \mathbf{x}, \mathbf{y} \rangle, y_1, \cdots, y_n, L, R, M, 1)$ is derived. Then, we generate an NIWI proof that σ' is a correct signature on $\boldsymbol{v}'$ and $\langle \mathbf{x}, \mathbf{y} \rangle \in [L, R]$.

We instantiate it by the simplified ALP LHS scheme [9] and the GS proof [8] to obtain a KARIP scheme secure under the DLIN, CDH and FlexCDH assumptions. To efficiently prove $\langle \mathbf{x}, \mathbf{y} \rangle \geq L$, we use the following fact. If $\langle \mathbf{x}, \mathbf{y} \rangle \geq L$, $\langle \mathbf{x}, \mathbf{y} \rangle = L$ or there exists a single index $t \in [1, \lambda]$ s.t. the leftmost $t-1$ bits of $\langle \mathbf{x}, \mathbf{y} \rangle$ and L are identical and the t-th bits of $\langle \mathbf{x}, \mathbf{y} \rangle$ and L are 1 and 0, respectively. More formally, $\exists t \in [1, \lambda + 1]$ s.t. $\bigwedge_{i=1}^{t-1} \langle \mathbf{x}, \mathbf{y} \rangle[i] = L[i] \bigwedge \langle \mathbf{x}, \mathbf{y} \rangle[t] = 1 \bigwedge L[t] = 0$. To prove $\langle \mathbf{x}, \mathbf{y} \rangle \leq R$, we also use the same fact. We rigorously prove that its secret-key and signature sizes are $N+4(n+2)|g|$ and $(18N+132\lambda+39)|g|$.

3rd Construction. It is generically constructed by LHS, NIWI and collision-resistant hash function (HF). It is similar to the 2nd ARIP scheme in [9]. In key-generation, we define only two vectors $\boldsymbol{v}_1 := (x_1, x_2, \cdots, x_n, L, R, 0, 1), \boldsymbol{v}_2 := (0, \cdots, 0, 1, 0) \in \mathbb{Z}_p^{n+4}$, then generate an LHS signature σ_i with the common tag τ on each vector $\boldsymbol{v}_i$. In signing, an LHS signature σ' on $\boldsymbol{v}' := (x_1, \cdots, x_n, L, R, h, 1)$, where h is the hash value of $(\mathbf{y}, M)$. Then, we generate an NIWI proof that σ' is a correct signature on $\boldsymbol{v}'$, $\langle \mathbf{x}, \mathbf{y} \rangle \in [L, R]$, and the inner product value is correctly calculated, i.e., $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i \cdot y_i \pmod{p}$.

To instantiate it, we use the same building blocks as our 2nd construction. We rigorously prove that its secret-key and signature sizes are $N + 8|g|$ and $(9n + 18N + 132\lambda + 42)|g|$.

Applications. As formally shown in [9], an ARIP scheme can be transformed into any of the following ABS primitives, namely AREP, AREWA, FIBS, TSS [13,10], AREHD, AREED and AHEP. The same transformation techniques also work for KARIP. A KARIP scheme can be transformed into their key-range versions. We emphasize that if the underlying KARIP scheme has key-delegatablity, the property is directly inherited after the transformation.

Paper Organization. In Sect. 2, we explain some notations, and define some computational assumptions, NIWI, LHS and AOS. In Sect. 3, we formally define KARIP. In Sect. 4 (resp. 5, 6), we propose our 1st (resp. 2nd, 3rd) generic KARIP construction, prove its security, and introduce its instantiation. In Sect. 7, we introduce the applications of KARIP.

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 $x_0 \in \mathbb{N}$ s.t. for every $x \geq x_0$, $f(x) \leq x^{-c}$. Given a binary string $x \in \{0,1\}^L$, for every $i \in [1, L]$, let $x[i] \in \{0,1\}$ denote its i-th bit. PPTA means probabilistic polynomial time algorithm. For a set A, $a \xleftarrow{\mathrm{U}} A$ means that an element a is chosen uniformly at random from A. For an integer $a \in \mathbb{N}$, $[a]_2$ denotes its binary value.

Symmetric Bilinear Pairing on Groups with Prime Order. $\mathcal{G}$ 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 bit 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 the following two conditions, (1) **Bilinearity:** For any $a, b \in \mathbb{Z}_p$, $e(g^a, g^b) = e(g, g)^{ab}$, (2) **Non-degeneracy:** $e(g, g) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ denotes the unit element of $\mathbb{G}_T$. In this work, $|g|$ denotes bit length of an element in the bilinear group $\mathbb{G}$.

Assumptions. We define the three computational hardness assumptions.

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

Definition 2. *The flexible CDH (FlexCDH) assumption [4] holds on the group $\mathbb{G}$ if for every PPT $\mathcal{A}$, $\mathsf{Adv}^{FlexCDH}_{\mathcal{A},\mathbb{G}}(\lambda) := \Pr[(g^\mu, g^{a\cdot\mu}, g^{ab\cdot\mu}) \leftarrow \mathcal{A}(g, g^a, g^b)]$ with $a, b \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and $\mu \neq 0$, is negligible.*

Definition 3. *The decisional linear (DLIN) assumption holds on the group $\mathbb{G}$ if for every PPT $\mathcal{A}$, $\mathsf{Adv}^{DLIN}_{\mathcal{A},\mathbb{G}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(g^a, g^b, g^{ab}, g^{bd}, g^{c+d})]| - \Pr[1 \leftarrow \mathcal{A}(g^a, g^b, g^{ab}, g^{bd}, g^z)]$ with $a, b, c, d, z \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, is negligible.*

2.1 Non-Interactive Witness Indistinguishable Proof (NIWI)

An NIWI system for the NP relation $R : \{0,1\}^* \times \{0,1\}^* \to 1/0$ consists of the following 3 polynomial-time algorithms. Note that $\mathtt{Ver}$ is deterministic and the others are probabilistic. Setup algorithm $\mathtt{Setup}$ takes a security parameter 1^λ for $\lambda \in \mathbb{N}$, then outputs a common reference string (CRS) crs. Proving algorithm $\mathtt{Pro}$ takes the CRS crs, a statement $x \in \{0,1\}^*$ and a witness $w \in \{0,1\}^*$, then outputs a proof π. Verification $\mathtt{Ver}$ takes the CRS crs, a statement $x \in \{0,1\}^*$ and a proof π, then outputs a verification result $1/0$. 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 $1 \leftarrow \mathtt{Ver}(crs, x, \pi)$.

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

Definition 4. *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 5. *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 $\mathcal{A}$, $\mathbf{Adv}^{\mathsf{WE}}_{\Sigma_{\mathrm{NIWI}},\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(crs) \mid crs \leftarrow \mathtt{Setup}(1^\lambda)] - \Pr[1 \leftarrow \mathcal{A}(crs) \mid (crs, ek) \leftarrow \mathtt{SimSetup}(1^\lambda)]|$ is negligible.*
2. *For every probabilistic algorithm $\mathcal{A}$,*

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

2.2 Linearly Homomorphic Signatures (LHS) [5,4]

An LHS scheme consists of the following 4 polynomial-time algorithms. Note that $\mathtt{Setup}$ and $\mathtt{Sig}$ are probabilistic, $\mathtt{Ver}$ is deterministic and $\mathtt{Derive}$ is (possibly) probabilistic.

Key-Generation $\mathtt{KGen}$: It takes a security parameter 1^λ for $\lambda \in \mathbb{N}$ and an integer $n \in \mathbb{N}$ that indicates the dimension of a vector to be signed, then outputs a key-pair (pk, sk). $(pk, sk) \leftarrow \mathtt{KGen}(1^\lambda, n)$

Signing $\mathtt{Sig}$: It takes the secret-key sk, a tag $\tau \in \{0,1\}^*$ and a vector $\boldsymbol{v} \in \mathbb{Z}_p^n$ to be signed, then outputs a signature σ. $\sigma \leftarrow \mathtt{Sig}(sk, \tau, \boldsymbol{v})$

Derivation $\mathtt{Derive}$: It takes the public-key pk, a tag $\tau \in \{0,1\}^*$ and l triples $\{\boldsymbol{v}_i, \sigma_i, \beta_i\}_{i=1}^l$, consisting of a vector $\boldsymbol{v}_i \in \mathbb{Z}_p^n$, a signature σ_i and a weight β_i, then outputs a signature $\overline{\sigma}$ on the weighted vector $\overline{\boldsymbol{v}} := \sum_{i=1}^l \beta_i \cdot \boldsymbol{v}_i \in \mathbb{Z}_p^n$. $\overline{\sigma} \leftarrow \mathtt{Derive}(pk, \tau, \{\boldsymbol{v}_i, \sigma_i, \beta_i\}_{i=1}^l)$

Verification $\mathtt{Ver}$**:** It takes the public-key pk, a tag $\tau \in \{0,1\}^*$, a vector $\boldsymbol{v} \in \mathbb{Z}_p^n$ and a signature σ, then outputs 1 or 0. $1/0 \leftarrow \mathtt{Ver}(pk, \tau, \boldsymbol{v}, \sigma)$

We require every LHS scheme to be correct. An LHS scheme is correct if for any $\lambda \in \mathbb{N}$, any $n \in \mathbb{N}$ and any $(pk, sk) \leftarrow \mathtt{KGen}(1^\lambda, n)$, the following two conditions hold, namely (1) $1 \leftarrow \mathtt{Ver}(pk, \tau, \boldsymbol{v}, \mathtt{Sig}(sk, \tau, \boldsymbol{v}))$ *for any tag* $\tau \in \{0,1\}^*$ *and any* $\boldsymbol{v} \in \mathbb{Z}_p^n$, and (2) $1 \leftarrow \mathtt{Ver}(pk, \tau, \sum_{i=1}^{l} \beta_i \boldsymbol{v}_i, \mathtt{Derive}(pk, \tau, \{\boldsymbol{v}_i, \sigma_i, \beta_i\}_{i=1}^{l}))$ *for any tag* $\tau \in \{0,1\}^*$*, any integer* $l \in \mathbb{N}$ *and any* l *triples* $\{\boldsymbol{v}_i \in \mathbb{Z}_p^n, \sigma_i, \beta_i \in \mathbb{Z}_p\}_{i=1}^{l}$ *s.t.* $1 \leftarrow \mathtt{Ver}(pk, \tau, \boldsymbol{v}_i, \sigma_i)$ *for each* $i \in [1, l]$.

As security notions for P-homomorphic signatures [2], a generalization of LHS and AOS, unforgeability and unlinkability-related strong context-hiding (SCH) and complete context-hiding (CCH) [3] have been defined. Since these notions are not needed for our KARIP constructions, we define only *weak* unforgeability weaker than the original notion of unforgeablity [2]. We consider the following experiment, where a PPT algorithm $\mathcal{A}$ adaptively accesses a signing oracle to get a signature on an arbitrarily chosen vector $\boldsymbol{v}$, then outputs a forged signature.

$\boldsymbol{Expt}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{A}}(1^\lambda, n)$:
1. $(pk, sk) \leftarrow \mathtt{Setup}(1^\lambda, n)$. $(\tau^* \in \{0,1\}^*, \boldsymbol{v}^* \in \mathbb{Z}_p^n, \sigma^*) \leftarrow \mathcal{A}^{\mathfrak{Sign}}(pk)$.

- $\mathfrak{Sign}(\tau \in \{0,1\}^*, \boldsymbol{v} \in \mathbb{Z}_p^n)$: $Q := Q \cup \{(\tau, \boldsymbol{v})\}$. **Rtrn** $\sigma \leftarrow \mathtt{Sig}(sk, \tau, \boldsymbol{v})$.

2. **Rtrn** 1 if (1) $1 \leftarrow \mathtt{Ver}(pk, \tau^*, \boldsymbol{v}^*, \sigma^*)$ and (2) one of the following conditions is satisfied.
   (a) $\tau^* \neq \tau_i$ for any entry $(\tau_i, \cdot) \in Q$ and $\boldsymbol{v}^* \neq \boldsymbol{0}$.
   (b) $\tau^* = \tau_i$ for $k > 0$ entries $(\tau_i, \boldsymbol{v}_i)$ in Q and $\boldsymbol{v}^* \notin \mathbf{span}\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k\}$.

Definition 6. *An LHS scheme* Σ_{LHS} *is* ***wUNF*** *if for every* $\lambda \in \mathbb{N}$*, every* $n \in \mathtt{poly}(\lambda)$ *and every PPT* $\mathcal{A}$*,* $\mathcal{A}$*'s advantage defined as* $\boldsymbol{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{A}}(\lambda) := \Pr[1 \leftarrow \boldsymbol{Expt}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{A}}(1^\lambda, n)]$ *is negligible.*

Unforgeablity, SCH and CCH of LHS are defined in Subsect. A.1.

2.3 Append-Only Signatures (AOS) [12]

An AOS scheme consists of the following 4 polynomial-time algorithms. Note that $\mathtt{Setup}$ and $\mathtt{Sig}$ are probabilistic, $\mathtt{Ver}$ is deterministic and $\mathtt{Derive}$ is (possibly) probabilistic.

Key-Generation $\mathtt{KGen}$**:** It takes a security parameter 1^λ, the maximum depth of message $H \in \mathbb{N}$ and bit length of a sub-message $L \in \mathbb{N}$, then outputs a key-pair (pk, sk). $(pk, sk) \leftarrow \mathtt{KGen}(1^\lambda, H, L)$

Signing $\mathtt{Sig}$**:** It takes the secret-key sk and a message $M \in (\{0,1\}^L)^{h \leq H}$, then outputs a signature σ. $\sigma \leftarrow \mathtt{Sig}(sk, M)$

Derivation $\mathtt{Derive}$**:** It takes the public-key pk, a message $M \in (\{0,1\}^L)^{h \leq H}$, a signature σ and a message $M' \in (\{0,1\}^L)^{h' \leq H}$, then outputs a signature σ'. $\sigma' \leftarrow \mathtt{Derive}(pk, M, \sigma, M')$

Verification $\mathtt{Ver}$**:** It takes the public-key pk, a message $M \in (\{0,1\}^L)^{h \leq H}$ and a signature σ, then outputs 1 or 0. $1/0 \leftarrow \mathtt{Ver}(pk, M, \sigma)$

We require every AOS scheme to be correct. An AOS scheme is correct if for any $\lambda \in \mathbb{N}$, any $H, L \in \mathbb{N}$ and any $(pk, sk) \leftarrow \mathtt{KGen}(1^\lambda, H, L)$, both of the following conditions hold, (1) $1 \leftarrow \mathtt{Ver}(pk, M, \mathtt{Sig}(sk, M))$ *for any* $M \in (\{0,1\}^L)^{h \le H}$, and (2) $1 \leftarrow \mathtt{Ver}(pk, M', \mathtt{Derive}(pk, M, \sigma, M'))$ *for any* $M \in (\{0,1\}^L)^{h \le H}$, $M' \in (\{0,1\}^L)^{h' \le H}$ *s.t.* $h \le h' \bigwedge_{i=1}^{h} M_i = M'_i$ *and any* σ *s.t.* $1 \leftarrow \mathtt{Ver}(pk, M, \sigma)$.

As LHS, we define only weak unforgeablity (wUNF) for AOS.

$\boldsymbol{Expt}^{\mathtt{UNF}}_{\Sigma_{\mathrm{AOS}},\mathcal{A}}(1^\lambda, H, L)$:
1. $(pk, sk) \leftarrow \mathtt{Setup}(1^\lambda, H, L)$. $(M^* \in (\{0,1\}^L)^{h^*}, \sigma^*) \leftarrow \mathcal{A}^{\mathfrak{Sign}}(pk)$.

- $\mathfrak{Sign}(M \in (\{0,1\}^L)^h)$: $Q := Q \cup \{M\}$. **Rtrn** $\sigma \leftarrow \mathtt{Sig}(sk, M)$.

2. **Rtrn** 1 if (1) $1 \leftarrow \mathtt{Ver}(pk, M^*, \sigma^*)$, and
(2) $h > h^* \vee \exists i \in [1,h]$ *s.t.* $m_i \neq m_i^*$ *for any* $M \in Q$, *where* $M \in (\{0,1\}^L)^h$ *for some* $h \le H$.
3. **Rtrn** 0.

Definition 7. *An AOS scheme* Σ_{AOS} *is* ***wUNF*** *if for every* $\lambda \in \mathbb{N}$, *every* $H, L \in \mathbb{N}$ *and every PPT* $\mathcal{A}$, $\boldsymbol{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{AOS}},\mathcal{A}}(\lambda) := \Pr[1 \leftarrow \boldsymbol{Expt}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{AOS}},\mathcal{A}}(1^\lambda, H, L)]$ *is negligible.*

Unforgeablity, SCH and CCH of AOS are defined in Subsect. A.2.

3 Key-Range ABS for Range of Inner-Product (KARIP)

A KARIP consists of the following four polynomial-time algorithms. Ver is deterministic and the others are probabilistic.

Setup Setup**:** It takes a security parameter 1^λ for $\lambda \in \mathbb{N}$ and a number of dimensions $n \in \mathbb{N}$, then outputs a public parameter pp and master-key mk. Assume that a prime p with bit length λ is chosen and included in pp. The other algorithms implicitly take pp as input. $(pp, mk) \leftarrow \mathtt{Setup}(1^\lambda, n)$

Key-Generation KGen**:** It takes mk and an n-dimensional vector $\mathbf{x} \in \mathbb{Z}_p^n$ and a range $[L, R] = \{L, L+1, \cdots, R-1, R\}$ with $L, R \in \mathbb{Z}_p$, then outputs a secret-key sk. $sk \leftarrow \mathtt{KGen}(mk, \mathbf{x}, L, R)$

Signing Sig**:** It takes a secret-key sk, a message $M \in \mathcal{M}$ and an n-dimensional vector $\mathbf{y} \in \mathbb{Z}_p^n$, then outputs a signature σ. $\sigma \leftarrow \mathtt{Sig}(sk, M, \mathbf{y})$

Verification Ver**:** It takes a signature σ, a message $M \in \mathcal{M}$ and an n-dimensional vector $\mathbf{y} \in \mathbb{Z}_p^n$, then outputs 1 or 0. $1/0 \leftarrow \mathtt{Ver}(\sigma, M, \mathbf{y})$

Every KARIP scheme must be correct. A KARIP scheme is correct if $\forall \lambda \in \mathbb{N}$, $\forall n \in \mathbb{N}$, $\forall (pp, mk) \leftarrow \mathtt{Setup}(1^\lambda, n)$, $\forall \mathbf{x} \in \mathbb{Z}_p^n$, $\forall L, R \in \mathbb{Z}_p$, $\forall sk \leftarrow \mathtt{KGen}(mk, \mathbf{x}, L, R)$, $\forall M \in \mathcal{M}$, $\forall \mathbf{y} \in \mathbb{Z}_p^n$ s.t. $\langle \mathbf{x}, \mathbf{y} \rangle \in [L, R] \pmod p$, $\forall \sigma \leftarrow \mathtt{Sig}(sk, M, \mathbf{y})$, $1 \leftarrow \mathtt{Ver}(\sigma, M, \mathbf{y})$ holds.

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

$\boldsymbol{Expt}^{\mathtt{UNF}}_{\Sigma_{\mathrm{KARIP}},\mathcal{A}}(1^\lambda)$:
1. $(pp, mk) \leftarrow \mathtt{Setup}(1^\lambda)$. $(\sigma^*, M^* \in \mathcal{M}, \mathbf{y}^* \in \mathbb{Z}_p^n) \leftarrow \mathcal{A}^{\mathfrak{Reveal},\mathfrak{Sign}}(pp)$.

- $\mathfrak{Reveal}(\mathbf{x} \in \mathbb{Z}_p^n, L, R \in \mathbb{Z}_p)$: $sk \leftarrow \mathtt{KGen}(mk, \mathbf{x})$. $Q := Q \cup \{(\mathbf{x}, L, R)\}$. **Rtrn** sk.
- $\mathfrak{Sign}(\mathbf{x} \in \mathbb{Z}_p^n, L, R \in \mathbb{Z}_p, M \in \mathcal{M}, \mathbf{y} \in \mathbb{Z}_p^n)$: $sk \leftarrow \mathtt{KGen}(mk, \mathbf{x}, L, R)$. $\sigma \leftarrow \mathtt{Sig}(sk, M, \mathbf{y})$.

$Q' := Q' \cup \{(M, \mathbf{y}, \sigma)\}$. **Rtrn** σ.

---

2. **Rtrn** 1 if (1) $1 \leftarrow \mathtt{Ver}(\sigma^*, M^*, \mathbf{y}^*)$, (2) $\forall(\mathbf{x}, L, R) \in Q$, $\langle \mathbf{x}, \mathbf{y}^* \rangle \notin [L, R]$ and (3) $(M^*, \mathbf{y}^*, \cdot) \notin Q'$.
3. **Rtrn** 0.

Definition 8. *A KARIP scheme Σ_{KARIP} is UNF if for every PPT $\mathcal{A}$, its advantage $\mathit{Adv}^{\mathit{UNF}}_{\Sigma_{\mathrm{KARIP}},\mathcal{A}}(\lambda) := \Pr[1 \leftarrow \boldsymbol{Expt}^{\mathit{UNF}}_{\Sigma_{\mathrm{KARIP}},\mathcal{A}}(1^\lambda, n)]$ is negligible.*

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

$\boldsymbol{Expt}^{\mathrm{PRV}}_{\Sigma_{\mathrm{KARIP}},\mathcal{A},0}(1^\lambda)$: //$\boldsymbol{Expt}^{\mathrm{PRV}}_{\Sigma_{\mathrm{KARIP}},\mathcal{A},1}$
$(pp, mk) \leftarrow \mathtt{Setup}(1^\lambda)$. $(pp, mk, \mu) \leftarrow \mathtt{SimSetup}(1^\lambda)$. **Rtrn** $b' \leftarrow \mathcal{A}^{\mathfrak{Reveal}, \mathfrak{Sign}}(pp, mk)$.

- $\mathfrak{Reveal}(\mathbf{x} \in \mathbb{Z}_p^n, L, R \in \mathbb{Z}_p)$:
  $sk \leftarrow \mathtt{KGen}(mk, \mathbf{x}, L, R)$. $sk \leftarrow \mathtt{SimKGen}(mk, \mu, \mathbf{x}, L, R)$. $Q := Q \cup \{(\mathbf{x}, L, R, sk)\}$. **Rtrn** sk.
- $\mathfrak{Sign}(\mathbf{x} \in \mathbb{Z}_p^n, L, R \in \mathbb{Z}_p, sk, M \in \mathcal{M}, \mathbf{y} \in \mathbb{Z}_p^n)$:
  Rtrn $\perp$ if $(\mathbf{x}, L, R, sk) \notin Q \vee \langle \mathbf{x}, \mathbf{y} \rangle \notin [L, R] \pmod p$.
  $\sigma \leftarrow \mathtt{Sig}(sk, M, \mathbf{y})$. $\sigma \leftarrow \mathtt{SimSig}(mk, \mu, M, \mathbf{y})$. **Rtrn** σ.

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

Definition 9. *A KARIP scheme Σ_{KARIP} is perfectly signer-private (PRV) if for every probabilistic algorithm $\mathcal{A}$, there exist polynomial-time algorithms $\{\mathtt{SimSetup}, \mathtt{SimKGen}, \mathtt{SimSig}\}$ such that $\mathcal{A}$'s advantage $\mathit{Adv}^{\mathit{PRV}}_{\Sigma_{\mathrm{KARIP}},\mathcal{A}}(\lambda) := |\sum_{b=0}^{1}(-1)^b \Pr[1 \leftarrow \boldsymbol{Expt}^{\mathit{PRV}}_{\Sigma_{\mathrm{KARIP}},\mathcal{A},b}(1^\lambda)]|$ becomes 0.*

Key-Delegatability. We say that a KARIP scheme is key-delegatable if for any vector $\mathbf{x} \in \mathbb{Z}_p^n$, any range $[L, R] \subseteq \mathbb{Z}_p$ and any subrange $[l, r] \subseteq \mathbb{Z}_p$ s.t. $L \le l \le r \le R$, any secret-key for $(\mathbf{x}, L, R)$ can generate a secret-key for $(\mathbf{x}, l, r)$.

4 Our 1st Generic Construction of KARIP

4.1 Construction

We use an algorithm Cover called covering. Consider a complete binary tree with 2^λ leaf nodes. The leftmost (resp. rightmost) leaf node is associated with 0^λ (resp. 1^λ). Since p is of bit length λ, for every integer $i \in \mathbb{Z}_p$, $[i]_2 \in \{0,1\}^\lambda$ is corresponded to a leaf node one-to-one. Cover takes a range $[L, R] \subseteq \mathbb{Z}_p$, then outputs a set C with the minimal cardinality, composed of intermediate nodes which *covers* all of the leaf nodes from $[L]_2$ to $[R]_2$. For every $i \in [L, R]$, there is a single $c \in C$ s.t. c is either identical to or an ancestor of $[i]_2$. Such a set can be efficiently and easily derived. Refer to Subsect. A.4 for the definition of Cover.

Our generic KARIP construction is built by an LHS scheme $\{\mathsf{L}.\mathtt{KGen}, \mathsf{L}.\mathtt{Sig}, \mathsf{L}.\mathtt{Derive}, \mathsf{L}.\mathtt{Ver}\}$, an AOS scheme $\{\mathsf{A}.\mathtt{KGen}, \mathsf{A}.\mathtt{Sig}, \mathsf{A}.\mathtt{Derive}, \mathsf{A}.\mathtt{Ver}\}$ and an NIWI proof system $\{\mathsf{N}.\mathtt{Setup}, \mathsf{N}.\mathtt{Pro}, \mathsf{N}.\mathtt{Ver}\}$.

$\mathtt{Setup}(1^\lambda, L)$**:** Generate $crs \leftarrow \mathsf{N}.\mathtt{Setup}(1^\lambda)$, $(pk_\mathsf{L}, sk_\mathsf{L}) \leftarrow \mathsf{L}.\mathtt{KGen}(1^\lambda, n+3)$ with tags whose bit length is $N \in \mathtt{poly}(\lambda)$ and $(pk_\mathsf{A}, sk_\mathsf{A}) \leftarrow \mathsf{A}.\mathtt{KGen}(1^\lambda, N+\lambda, 1)$. Output $pp := (crs, pk_\mathsf{L}, pk_\mathsf{A})$ and $mk := (sk_\mathsf{L}, sk_\mathsf{A})$.

$\mathtt{KGen}(mk, \mathbf{x}, L, R)$**:** Choose a tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Conduct the following two steps.

1. For each $i \in [1, n]$, let $\boldsymbol{v}_i := (x_i, \underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{n-i}, 0, 0) \in \mathbb{Z}_p^{n+3}$. Let $\boldsymbol{v}_{n+1} := (0, \cdots, 0, 1, 0) \in \mathbb{Z}_p^{n+3}$ and $\boldsymbol{v}_{n+2} := (0, \cdots, 0, 0, 1) \in \mathbb{Z}_p^{n+3}$. For each $\boldsymbol{v}_i$, generate an LHS signature with tag τ by $\sigma_i \leftarrow \mathsf{L}.\mathtt{Sig}(sk_{\mathsf{L}}, \tau, \boldsymbol{v}_i)$.
2. For each $c \in C$, generate an AOS signature on $(\tau, c[1], \cdots, c[h_c])$, i.e., $\theta_c \leftarrow \mathsf{A}.\mathtt{Sig}(sk_{\mathsf{A}}, (\tau, c[1], \cdots, c[h_c]))$, where c is parsed as $c[1] \,\|\, \cdots \,\|\, c[h_c]$ for some $h_c \in [1, \lambda]$. Note that this construction is key-delegatable. Consider a subrange $[l, r] \subseteq [L, R]$, and let $C' \leftarrow \mathtt{Cover}(l, r)$. For any $c' \in C'$, there must exist a single $c \in C$ s.t. c is either identical to or an ancestor of c'.

Output $sk := (\tau, \{\sigma_i\}_{i=1}^{n+2}, \{\theta_c\}_{c \in C})$.

$\mathtt{Sig}(sk, M, \mathbf{y})$**:** Parse sk as above. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Assume that $d \in [L, R]$. Conduct the following three steps.

1. Generate an LHS signature on $\boldsymbol{v}' := (d, y_1, \cdots, y_n, M, 1)$ by $\sigma' \leftarrow \mathsf{L}.\mathtt{Derive}(pk_{\mathsf{L}}, \tau, \{\boldsymbol{v}_i, \sigma_i, \beta_i\}_{i=1}^{n+2})$, where $\beta_{n+1} := M$, $\beta_{n+2} := 1$ and $\beta_i := y_i$ for each $i \in [1, n]$.
2. $d \in [L, R]$ implies that there is $c \in C$ s.t. c is whether identical to or an ancestor of $[d]_2$. Derive an AOS signature on $(\tau, d[1], \cdots, d[\lambda])$ from θ_c, i.e., $\theta' \leftarrow \mathsf{A}.\mathtt{Derive}(pk_{\mathsf{A}}, (\tau, c[1], \cdots, c[h_c]), \theta_c, (\tau, d[1], \cdots, d[\lambda]))$, where $[d]_2$ is parsed as $d[1] \,\|\, \cdots \,\|\, d[\lambda]$.
3. Define the NIWI relation $\mathcal{R}_{\mathsf{N}}$ as follows.
   - A statement $x = (\hat{\mathbf{y}}, \hat{M})$ consists of a vector $\hat{\mathbf{y}} = (\hat{y}_1, \cdots, \hat{y}_n) \in \mathbb{Z}_p^n$ and a message $\hat{M} \in \mathbb{Z}_p$. A witness $w = (\hat{d}, \hat{\tau}, \hat{\sigma}, \hat{\theta})$ consists of an inner product value $\hat{d} \in \mathbb{Z}_p$, an LHS tag $\hat{\tau} \in \{0,1\}^L$, an LHS signature $\hat{\sigma}$ and an AOS signature $\hat{\theta}$. $\mathcal{R}_{\mathsf{N}}$ takes a statement x and witness w then outputs 1 if both of the following conditions are satisfied.
     1. $1 \leftarrow \mathsf{L}.\mathtt{Ver}(pk_{\mathsf{L}}, \hat{\tau}, \hat{\boldsymbol{v}}, \hat{\sigma})$, where $\hat{\boldsymbol{v}} := (\hat{d}, \hat{y}_1, \cdots, \hat{y}_n, \hat{M}, 1)$.
     2. $1 \leftarrow \mathsf{A}.\mathtt{Ver}(pk_{\mathsf{A}}, (\hat{\tau}, \hat{d}[1], \cdots, \hat{d}[\lambda]), \hat{\theta})$.

   If we set $x := (\mathbf{y}, M)$ and $w := (d, \tau, \sigma, \theta)$, it obviously holds that $1 \leftarrow \mathcal{R}_{\mathsf{N}}(x, w)$. Output $\sigma \leftarrow \mathsf{N}.\mathtt{Pro}(crs, x, w)$.

$\mathtt{Ver}(\sigma, M, \mathbf{y})$**:** Set $x := (\mathbf{y}, M)$ and output $1/0 \leftarrow \mathsf{N}.\mathtt{Ver}(crs, x, \sigma)$.

As explained in the key-generation algorithm, this construction is key-delegatable. For its privacy and unforgeability, we give the following two theorems.

Theorem 1. *Our 1st KARIP scheme is* PRV *if the NIWI scheme is* WI*.*

Proof. The signer-privacy experiments w.r.t. our 1st KARIP scheme are simply denoted by $\boldsymbol{Expt}_0$ and $\boldsymbol{Expt}_1$. For the three simulation algorithms associated with $\boldsymbol{Expt}_1$, SimSetup and SimKGen are identical to the original ones[3]. SimSig is defined as follows.

[3] The auxiliary variable μ outputted by SimSetup is null.

$\mathtt{SimSig}(mk, M, \mathbf{y})$**:** Arbitrarily choose $\mathbf{x} \in \mathbb{Z}_p^n$ and $L, R \in \mathbb{Z}_p$ s.t. $d(:= \langle \mathbf{x}, \mathbf{y} \rangle) \in [L, R] \pmod p$. Choose $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Generate an LHS signature on $\boldsymbol{v}' := (d, y_1, \cdots, y_n, M, 1)$ by $\sigma' \leftarrow \mathsf{L.Sig}(sk_{\mathsf{L}}, \tau, \boldsymbol{v}')$. Generate an AOS signature on $(\tau, d[1], \cdots, d[\lambda])$ by $\theta'_c \leftarrow \mathsf{A.Sig}(sk_{\mathsf{A}}, (\tau, d[1], \cdots, d[\lambda]))$. Generate an NIWI proof $\pi \leftarrow \mathsf{N.Pro}(crs, x, w)$, where $x := (\mathbf{y}, M)$ and $w := (d, \tau, \sigma', \theta')$, then return π.

It holds that $1 \leftarrow \mathcal{R}_{\mathsf{N}}(x, w)$. Hence, if the NIWI scheme is `WI`, the simulated signature π distributes identically to the real one in $\boldsymbol{Expt}_0$. □

Theorem 2. *Our 1st KARIP scheme is `UNF` if the LHS scheme is `wUNF`, the AOS scheme is `wUNF`, and the NIWI system is `WI` and `WE`.*

Proof. We define six experiments as follows.

$\boldsymbol{Expt}_0$**:** The standard `UNF` experiment w.r.t. the KARIP scheme.

$\boldsymbol{Expt}_1$**:** The same as $\boldsymbol{Expt}_0$ except that it aborts when we choose a tag on the key-revelation or signing oracle, the tag matches a tag previously chosen.

$\boldsymbol{Expt}_2$**:** The same as $\boldsymbol{Expt}_1$ except for the signature generation on the signing oracle. In $\boldsymbol{Expt}_2$, we directly generate both of an LHS signature σ' on $\boldsymbol{v}' := (\langle \mathbf{x}, \mathbf{y} \rangle, y_1, \cdots, y_n, M, 1)$ and an AOS signature θ' on $(\tau, \langle \mathbf{x}, \mathbf{y} \rangle[1], \cdots, \langle \mathbf{x}, \mathbf{y} \rangle[\lambda])$ by using the LHS and AOS secret-keys, respectively.

$\boldsymbol{Expt}_3$**:** The same as $\boldsymbol{Expt}_2$ except for the CRS generation. In $\boldsymbol{Expt}_3$, the CRS crs is generated by $(crs, ek) \leftarrow \mathtt{SimSetup}(1^\lambda)$.

$\boldsymbol{Expt}_4$**:** Basically the same as $\boldsymbol{Expt}_3$. In $\boldsymbol{Expt}_4$, we extract the NIWI witness w^* for the NIWI proof σ^* by using the extraction key ek. Formally, extract $w^* \leftarrow \mathtt{Extract}(crs, ek, x^*, \sigma^*)$, where $x^* := (\mathbf{y}^*, M^*)$. The witness is parsed as $(d^* \in \mathbb{Z}_p, \tau^* \in \{0,1\}^N, \sigma^*, \theta^*)$. $\boldsymbol{Expt}_4$ aborts if w^* is not the correct witness for the statement x^*, i.e., $0 \leftarrow \mathcal{R}_{\mathsf{N}}(x^*, w^*)$.

$\boldsymbol{Expt}_5$**:** The same as $\boldsymbol{Expt}_4$ except that it aborts if one of the following three events occurs.

- **E1:** The extracted tag τ^* is identical to no tag previously chosen.
- **E2:** The tag τ^* has been already chosen on the signing oracle.
- **E3:** The tag τ^* has been already chosen on the key-revelation oracle and it holds that $d^* \neq \langle \hat{\mathbf{x}}, \mathbf{y}^* \rangle \pmod p$, where $d^* \in \mathbb{Z}_p$ is the extracted inner product value and $\hat{\mathbf{x}} \in \mathbb{Z}_p^n$ is the n-dimensional key-vector queried by $\mathcal{A}$.

For each $i \in [0, 5]$, let W_i denote the event that the experiment $\boldsymbol{Expt}_i$ outputs 1. We obtain $\mathtt{Adv}^{\mathtt{UNF}}_{\Sigma_{\mathrm{KARIP}}, \mathcal{A}, n}(\lambda) = \Pr[W_0] \leq \sum_{i=1}^{5} |\Pr[W_{i-1}] - \Pr[W_i]| + \Pr[W_5] \leq q(q-1)/2^{N+1} + \mathtt{Adv}^{\mathtt{WI}}_{\Sigma_{\mathrm{NIWI}}, \mathcal{B}_3}(\lambda) + \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}}, \mathcal{B}_5}(\lambda) + \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{AOS}}, \mathcal{B}_6}(\lambda)$ for some PPT adversary $\mathcal{B}_3, \mathcal{B}_5, \mathcal{B}_6$. The last inequality is obtained because of the following six lemmas. Lemma 1 is the same as Lemma 1 in [9] and can be proven in the same manner. Lemma 3 is true because the CRSs in $\boldsymbol{Expt}_2$ and $\boldsymbol{Expt}_3$ are indistinguishable if the NIWI system is `WE`. The other lemmas are proven below. For each $i \in \{1, 4, 5\}$, abort_i denotes the abort event firstly introduced in the experiment $\boldsymbol{Expt}_i$. □

Lemma 1. $\Pr[W_0] - \Pr[W_1] \leq q(q-1)/2^{N+1}$, *where* $q \in \mathtt{poly}(\lambda)$ *is the total number of times that* $\mathcal{A}$ *uses the key-revelation and signing oracles.*

Lemma 2. $|\Pr[W_1] - \Pr[W_2]| = 0$ *if the NIWI system is* `WI`.

Proof. In $\boldsymbol{Expt}_2$, on the signing oracle, we directly generate both of an LHS signature σ' and an AOS signature θ' by the LHS and AOS secret-keys, then generate a signature $\sigma(:=\pi)$ as $\pi \leftarrow \mathsf{N}.\mathtt{Pro}(crs, x, w)$, where $x := (\mathbf{y}, M)$ and $w := (d, \tau, \sigma', \theta')$. Since it holds that $1 \leftarrow \mathcal{R}_{\mathsf{N}}(x, w)$, the NIWI proof π distributes identically to the one in $\boldsymbol{Expt}_1$ if the NIWI system is `WI`. □

Lemma 3. $\Pr[W_2] - \Pr[W_3]$ *is negligible if the NIWI system is* `WE`. *Formally, there exists a PPT algorithm* $\mathcal{B}_3$ *s.t.* $\Pr[W_2] - \Pr[W_3] \leq \mathit{Adv}^{\mathit{WE}}_{\Sigma_{\mathrm{NIWI}},\mathcal{B}_3}(\lambda)$.

Lemma 4. $\Pr[W_3] - \Pr[W_4] = 0$ *if the NIWI system is* `WE`.

Proof. Obviously, $\Pr[W_4] = \Pr[W_3 \wedge \neg\mathsf{abort}_4]$. By a basic mathematical theorem, $\Pr[W_3] = \Pr[W_3 \wedge \mathsf{abort}_4] + \Pr[W_3 \wedge \neg\mathsf{abort}_4]$, which implies $\Pr[W_3] - \Pr[W_4] = \Pr[W_3 \wedge \mathsf{abort}_4]$. Assume that the case where $W_3 \wedge \mathsf{abort}_4$ occurs. Because of the event W_3, $1 \leftarrow \mathsf{N}.\mathtt{Ver}(crs, x^*, \sigma^*)$. Because of the event abort_4, $0 \leftarrow \mathcal{R}_{\mathsf{N}}(x^*, w^*)$. That contradicts to the `WE`. Hence, $\Pr[W_3] - \Pr[W_4] = 0$. □

Lemma 5. $\Pr[W_4] - \Pr[W_5]$ *is negligible if the LHS scheme is* `wUNF`. *Formally, there exists a PPT algorithm* $\mathcal{B}_5$ *s.t.* $\Pr[W_4] - \Pr[W_5] \leq \mathit{Adv}^{\mathit{wUNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{B}_5}(\lambda)$.

Proof. As the proof of Lemma 4, $\Pr[W_4] - \Pr[W_5] = \Pr[W_4 \wedge \mathsf{abort}_5]$ holds. Assume that $\mathcal{A}$ is a PPT algorithm which makes the event $W_4 \wedge \mathsf{abort}_5$ occur with a non-negligible probability. By using $\mathcal{A}$, a PPT simulator $\mathcal{B}_5$ attempts to win the `wUNF` experiment w.r.t. the LHS scheme.

$\mathcal{B}_5$ receives an honestly-generated public-key pk_{L}. $\mathcal{B}_5$ can access to the signing oracle $\mathfrak{Sign}_{\mathsf{L}}$. $\mathcal{B}_5$ honestly generates crs, ek, pk_{A} and sk_{A}. $\mathcal{B}_5$ sends $pp := (crs, pk_{\mathsf{L}}, pk_{\mathsf{A}})$ to $\mathcal{A}$ and run it. When $\mathcal{A}$ makes a query to the key-revelation or signing oracle, $\mathcal{B}_5$ behaves as follows.

$\mathfrak{Reveal}(\mathbf{x}, L, R)$**:** Choose a tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Honestly generate the $n+2$ vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{n+2} \in \mathbb{Z}_p^{n+3}$. For each vector $\boldsymbol{v}_i$, generate an LHS signature by $\sigma_i \leftarrow \mathfrak{Sign}_{\mathsf{L}}(\tau, \boldsymbol{v}_i)$. Let $C := \mathtt{Cover}(L, R)$. For each $c \in C$, generate $\theta_c \leftarrow \mathsf{A}.\mathtt{Sig}(sk_{\mathsf{A}}, (\tau, c[1], \cdots, c[h_c]))$. Return $sk := (\tau, \{\sigma_i\}_{i=1}^{n+2}, \{\theta_c\}_{c\in C})$.

$\mathfrak{Sign}(\mathbf{x}, L, R, \mathbf{y}, M)$**:** Choose $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Generate an LHS signature on a vector $\boldsymbol{v}' := (d, y_1, \cdots, y_n, M, 1)$ by $\sigma' \leftarrow \mathfrak{Sign}_{\mathsf{L}}(\tau, \boldsymbol{v}')$. Honestly generate an AOS signature on $(\tau, d[1], \cdots, d[\lambda])$, i.e., $\theta' \leftarrow \mathsf{A}.\mathtt{Sig}(sk_{\mathsf{A}}, (\tau, d[1], \cdots, d[\lambda]))$. Generate an NIWI proof $\pi \leftarrow \mathsf{N}.\mathtt{Pro}(crs, x, w)$, where $x := (\mathbf{y}, M)$ and $w := (d, \tau, \sigma', \theta')$, then return it.

Given a forged KARIP signature π^*, $\mathcal{B}_6$ extracts the witness behind the NIWI proof π^* by $w^* \leftarrow \mathtt{Extract}(crs, ek, x^*, \pi)$, where $x^* := (\mathbf{y}^*, M^*)$, and parse it as $(d^*, \tau^*, \sigma^*, \theta^*)$. $\mathcal{B}_5$ outputs a forged LHS signature σ^* with tag τ^* on vector $\boldsymbol{v}^* := (d^*, y_1^*, \cdots, y_n^*, M^*, 1)$.

The above is how $\mathcal{B}_5$ behaves. Because we have assumed that $\mathcal{A}$ makes the event $W_4 \wedge \mathsf{abort}_5$ occur, one of the three events **E1**, **E2** and **E3** must occur. Any of the events leads $\mathcal{B}_5$ to win the $\mathtt{wUNF}$ experiment.

E1: Every tag queried to $\mathfrak{Sign}_{\mathsf{L}}$ is not identical to τ^*. $W_4 \wedge \mathsf{abort}_5$ implies $\neg\mathsf{abort}_4$, which implies that σ^* is a valid LHS signature on the non-zero vector $\boldsymbol{v}^*$.

E2: W_4 implies $\neg\mathsf{abort}_1$, which implies that the extracted tag τ^* is identical to a single tag chosen on the signing oracle. Among multiple vectors whom $\mathcal{B}_5$ queried to $\mathfrak{Sign}_{\mathsf{L}}$, $\hat{\boldsymbol{v}} := (\langle \hat{\mathbf{x}}, \hat{\mathbf{y}} \rangle, \hat{y}_1, \cdots, \hat{y}_n, \hat{M}, 1)$ is the only vector tagged by τ^*, where $\hat{\mathbf{x}}, \hat{L}, \hat{R}, \hat{\mathbf{y}}$ and $\hat{M}$ denote variables queried to the signing oracle when the tag τ^* was chosen. W_4 implies that $(\mathbf{y}^*, M^*) \neq (\hat{\mathbf{y}}, \hat{M})$. Obviously, $\boldsymbol{v}^*$ is linearly independent of $\hat{\boldsymbol{v}}$.

E3: W_4 implies $\neg\mathsf{abort}_1$, which implies that the extracted tag τ^* is identical to a single tag chosen on the key-revelation oracle and it holds that $d^* \neq \langle \hat{\mathbf{x}}, \mathbf{y}^* \rangle \pmod p$. Among multiple vectors whom $\mathcal{B}_5$ queried to $\mathfrak{Sign}_{\mathsf{L}}$, there are $n+2$ vectors $\hat{\boldsymbol{v}}_1, \cdots, \hat{\boldsymbol{v}}_{n+2}$ tagged by τ^*. The vectors are expressed as follows. For each $i \in [1, n]$, $\hat{\boldsymbol{v}}_i = (\hat{x}_i, \underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{n-i}, 0, 0)$. The others are $\hat{\boldsymbol{v}}_{n+1} = (0, \cdots, 0, 1, 0)$ and $\hat{\boldsymbol{v}}_{n+2} = (0, \cdots, 0, 0, 1)$. Since $d^* \neq \langle \hat{\mathbf{x}}, \mathbf{y}^* \rangle \pmod p$, $\boldsymbol{v}^* = (d^*, y_1^*, \cdots, y_n^*, M^*, 1)$ is not in $\mathbf{span}(\{\hat{\boldsymbol{v}}_1, \cdots, \hat{\boldsymbol{v}}_{n+2}\})$.

Therefore, $\Pr[W_4] - \Pr[W_5] \leq \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}}, \mathcal{B}_5}(\lambda)$. □

Lemma 6. $\Pr[W_5]$ *is negligible if the AOS scheme is* $\mathtt{wUNF}$*. Formally, there exists a PPT algorithm* $\mathcal{B}_6$ *s.t.* $\Pr[W_5] \leq \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{AOS}}, \mathcal{B}_6}(\lambda)$.

Proof. Assume that $\mathcal{A}$ is a PPT algorithm which makes $\boldsymbol{Expt}_5$ outputs 1 with a non-negligible probability. By using $\mathcal{A}$, a PPT simulator $\mathcal{B}_6$ attempts to win the weak unforgeability experiment w.r.t. the underlying AOS scheme.

$\mathcal{B}_6$ receives a public-key pk_{A}, which has been honestly generated. $\mathcal{B}_6$ can access to the signing oracle $\mathfrak{Sign}_{\mathsf{A}}$. $\mathcal{B}_6$ honestly generates crs, ek, pk_{L} and sk_{L}. $\mathcal{B}_6$ sends $pp := (crs, pk_{\mathsf{L}}, pk_{\mathsf{A}})$ to $\mathcal{A}$ and run it.

$\mathfrak{Reveal}(\mathbf{x}, L, R)$**:** Choose a tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Honestly generate the $n+2$ vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{n+2} \in \mathbb{Z}_p^{n+3}$. For each vector $\boldsymbol{v}_i$, generate $\sigma_i \leftarrow \mathsf{L}.\mathtt{Sig}(sk_{\mathsf{L}}, \tau, \boldsymbol{v}_i)$. Let $C := \mathtt{Cover}(L, R)$. For each $c \in C$, generate $\theta_c \leftarrow \mathtt{Sig}_{\mathsf{A}}((\tau, c[1], \cdots, c[h_c]))$. Return $sk := (\mathbf{x}, L, R, \tau, \{\sigma_i\}_{i=1}^{n+2}, \{\theta_c\}_{c \in C})$.

$\mathfrak{Sign}(\mathbf{x}, L, R, \mathbf{y}, M)$**:** Choose $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Generate an LHS signature on the vector $\boldsymbol{v}' := (d, y_1, \cdots, y_n, M, 1)$, i.e., $\sigma' \leftarrow \mathsf{L}.\mathtt{Sig}(sk_{\mathsf{L}}, \tau, \boldsymbol{v}')$. Generate an AOS signature $\theta' \leftarrow \mathfrak{Sign}_{\mathsf{A}}((\tau, d[1], \cdots, d[\lambda]))$. Generate an NIWI proof $\pi \leftarrow \mathsf{N}.\mathtt{Pro}(crs, x, w)$, where $x := (\mathbf{y}, M)$ and $w := (d, \tau, \sigma', \theta')$, then return it.

$\mathcal{A}$ outputs a forged KARIP signature π^* on M^* under $\mathbf{y}^*$. We extract the witness for the NIWI proof π^* by $w^* \leftarrow \mathtt{Extract}(crs, ek, x^*, \pi)$, where $x^* := (\mathbf{y}^*, M^*)$, and parse it as $(d^*, \tau^*, \sigma^*, \theta^*)$. $\mathcal{B}_6$ outputs a forged AOS signature θ^* on $(\tau^*, d^*[1], \cdots, d^*[\lambda])$.

The above is the behavior of $\mathcal{B}_6$. We prove that $\mathcal{B}_6$ wins the experiment.

The assumption that W_5 occurs implies that neither abort_1 nor abort_5 occurs. Thus, the forged tag τ^* is identical to a single tag which was chosen on the key-revelation oracle and it holds that $d^* = \langle \hat{\mathbf{x}}, \mathbf{y}^* \rangle \pmod p$. When the tag τ^* was chosen, $\mathcal{B}_6$ makes the signing oracle reveal signatures on $(\tau^*, c[1], \cdots, c[\hat{h}_c])$ for all $c = c[1] \| \cdots \| c[\hat{h}_c] \in \mathtt{Cover}(\hat{L}, \hat{R})$. W_5 implies that $\langle \hat{\mathbf{x}}, \mathbf{y}^* \rangle \notin [\hat{L}, \hat{R}] \pmod p$, which implies that no $c \in \mathtt{Cover}(\hat{L}, \hat{R})$ is neither $[d^*]_2$ nor its ancestor. Thus, $\mathcal{B}_6$ wins. Hence, $\Pr[W_5] \leq \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{AOS}}, \mathcal{B}_6}(\lambda)$. □

4.2 Our AOS Scheme

We instantiate our generic construction in Subsect. 4.3. We use an NIWI proof by Groth and Sahai (GS) [8] secure under the decisional linear (DLIN) assumption. Its CRS consists of 3 vectors $\vec{f}_1, \vec{f}_2, \vec{f}_3 \in \mathbb{G}^3$, where $\vec{f}_1 = (f_1, 1, g)$, $\vec{f}_2 = (1, f_2, g)$ and $f_1, f_2 \in \mathbb{G}$. A commitment $\vec{C}$ to a group element $\mathcal{X} \in \mathbb{G}$ is given as $\vec{C} := (1, 1, \mathcal{X}) \cdot \vec{f}_1^{\,r} \cdot \vec{f}_2^{\,s} \cdot \vec{f}_3^{\,t}$, where $r, s, t \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. In the GS NIWI system, the prover can efficiently prove that committed variables satisfy a paring-product equation (PPE) in the form of $\prod_{i=1}^m e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^m \prod_{j=1}^m e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T$ for variables $\mathcal{X}_i \in \mathbb{G}$ and constants $\mathcal{A}_i \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$ and $t_T \in \mathbb{G}_T$.

Attrapadung, Libert and Peters (ALP) [4] proposed an LHS scheme unforgeable and CCH-secure under the flexible CDH (FlexCDH) assumption. Ishizaka et al. [9] simplified it to obtain another one weakly unforgeable under the same assumption. The LHS scheme is used for the instantiation. Its verification algorithm consists of only PPEs. Its full construction is in Subsect. C.1.

We searched for an AOS scheme used for the instantiation satisfying both of the two conditions, (1) *Based on symmetric, i.e., type-1, bilinear pairing with prime order* and (2) *Its verification algorithm consists of only PPEs.* We modified a hierarchical identity-based signatures (HIBS) scheme named HIBS-1 in [7] based on asymmetric type-3 bilinear paring with prime order, then obtained the following AOS scheme.

$\mathtt{KGen}(1^\lambda, H, L)$**:** $(\mathbb{G}, \mathbb{G}_T, e)$ denote the bilinear group description. g is a generator of $\mathbb{G}$. Choose $\beta \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and $g', U_1, \cdots, U_H, V_1, \cdots, V_L \xleftarrow{\mathrm{U}} \mathbb{G}$. Output (pk, sk), where $pk := (g, g^\beta, g', \{U_i\}_{i=1}^H, \{V_i\}_{i=1}^L)$ and $sk := \beta$.

$\mathtt{Sig}(sk, M)$**:** Parse $M \in (\{0,1\}^L)^{h \leq H}$ as $(m_1, \cdots, m_h)$. For each $i \in [1, h]$, choose $r_i \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and calculate $B_i := g^{r_i}$. Calculate $A := (g')^\beta \prod_{i=1}^h (U_i \prod_{j=1}^L V_j^{m_i[j]})^{r_i}$, where m_i is parsed as $m_i[1] \| \cdots \| m_i[L]$. Output $\sigma := (A, B_1, \cdots, B_h)$.

$\mathtt{Derive}(pk, M, \sigma, M')$**:** Parse $M' \in (\{0,1\}^L)^{h' \leq H}$ as $(m'_1, \cdots, m_h, \cdots, m'_{h'})$. For each $i \in [1, h']$, choose $r'_i \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and calculate $B'_i := B_i \cdot g^{r'_i}$ if $i \in [1, h]$ or $B'_i := g^{r'_i}$ otherwise. Then calculate $A' := A \cdot \prod_{i=1}^{h'} (U_i \prod_{j=1}^L V_j^{m'_i[j]})^{r'_i}$. Output $\sigma' := (A', B'_1, \cdots, B'_{h'})$.

$\mathtt{Ver}(pk, M, \sigma)$**:** Output 1 iff $e(A, g) = g(g^\beta, g') \prod_{i=1}^h e(B_i, U_i \prod_{j=1}^L V_j^{m_i[j]})$.

Theorem 3. *Our AOS scheme is* $\mathtt{wUNF}$ *under the CDH assumption w.r.t.* $\mathbb{G}$.

Proof. We assume that a PPT adversary $\mathcal{A}$ wins the $\mathtt{wUNF}$ experiment with a non-negligibility. A PPT simulator $\mathcal{B}$ solves the CDH problem by using $\mathcal{A}$. $\mathcal{B}$ receives a CDH problem instance (g, g^a, g^b), then behaves as follows.

Let $g^\beta := g^a$ and $g' := g^b$. Let $k := 2q$, where $q \in \mathtt{poly}(\lambda)$ denotes the maximal number of times that the signing oracle can be used. We assume that $k(L+1) < p$. For each $i \in [1, H]$, compute $U_i := (g^\beta)^{p-k\cdot s_i+x_i} \cdot g^{x'_i}$, where $s_i \xleftarrow{\mathrm{U}} [0, L]$, $x_i \xleftarrow{\mathrm{U}} \mathbb{Z}_k$ and $x'_i \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. For each $j \in [1, L]$, compute $V_j := (g^\beta)^{y_j} \cdot g^{y'_j}$, where $y_j \xleftarrow{\mathrm{U}} \mathbb{Z}_k$ and $y'_j \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. For an index $i \in [1, H]$ and a sub-message $m \in \{0,1\}^L$, define the following three functions.

$$J_i(m) := x'_i + \sum_{j=1}^{L} y'_j \cdot m[j], \quad L_i(m) := x_i + \sum_{j=1}^{L} y_j \cdot m[j]$$

$$F_i(m) := p - k \cdot s_i + x_i + \sum_{j=1}^{L} y_j \cdot m[j] \ \ (= p - k \cdot s_i + L_i(m))$$

Note that it holds that $U_i \prod_{j=1}^{L} V_j^{m[j]} = (g^\beta)^{F_i(m)} \cdot g^{J_i(m)}$. We often use the following theorem, which is proven in Subsect. B.3.

Theorem 4. *For any $i \in [1, H]$ and any $m \in \{0,1\}^L$, if $F_i(m) = 0 \pmod p$ then $L_i(m) = 0 \pmod k$.*

If $\mathcal{A}$ queries a message $M \in (\{0,1\}^L)^{h \le H}$ to the signing oracle, the simulator $\mathcal{B}$ generates a signature σ as follows. Consider the following two cases, **(S1)** $\nexists i \in [1, h]$ s.t. $L_i(m_i) \neq 0 \pmod k$ and **(S2)** Otherwise.

S1: Abort the simulation.

S2: It holds that $\exists i \in [1, h]$ s.t. $L_i(m_i) \neq 0 \pmod k$. Let t denote such an index i. Contraposition of Theorem 4 guarantees that $F_t(m_t) \neq 0 \pmod p$. For each $i \in [1, h]$, choose $r_i \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and compute $B_i := g^{r_i}$ if $i \neq t$ or $B_i := (g')^{-1/F_t(m_t)} \cdot g^{r_t}$ otherwise. Compute

$$\begin{aligned}
\Delta &:= (g')^{\frac{J_t(m_t)}{F_t(m_t)}} \cdot (g^\beta)^{r_t \cdot F_t(m_t)} \cdot g^{r_t \cdot J_t(m_t)} \\
&= (g')^\beta \cdot (g')^{-\frac{\beta \cdot F_t(m_t)}{F_t(m_t)}} \cdot (g')^{-\frac{J_t(m_t)}{F_t(m_t)}} \cdot (g^\beta)^{r_t \cdot F_t(m_t)} \cdot g^{r_t \cdot J_t(m_t)} \\
&= (g')^\beta \cdot g^{(r_t - \frac{b}{F_t(m_t)})(\beta \cdot F_t(m_t) + J_t(m_t))} = (g')^\beta \cdot (U_t \prod_{j=1}^{L} V_j^{m_t[j]})^{r_t - \frac{b}{F_t(m_t)}}
\end{aligned}$$

and $A := \Delta \cdot \prod_{i \in [1,h] \setminus \{t\}} U_i \prod_{j=1}^{L} V_j^{m_i[j]}$. Finalize $\sigma := (A, B_1, \cdots, B_h)$.

If $\mathcal{A}$ outputs a forged signature $\sigma^* = (A^*, B_1^*, \cdots, B_{h^*}^*)$ on a message $M^* = (m_1^*, \cdots, m_{h^*}^*)$ with depth $h^* \in [1, H]$, $\mathcal{B}$ considers the following two cases, **(F1)** $\exists i \in [1, h^*]$ s.t. $F_i(m_i^*) \neq 0 \pmod p$, and **(F2)** otherwise.

F1: Abort the simulation.

F2: It holds that $\forall i \in [1, h^*]$, $F_i(m_i^*) = 0 \pmod p$. We have assumed that $\mathcal{A}$ successfully forges a signature. There exist integers $r_1^*, \cdots, r_{h^*}^* \in \mathbb{Z}_p$ s.t. $A^* = (g')^\beta \prod_{i=1}^{h^*}(U_i \prod_{j=1}^{L} V_j^{m_i^*[j]})^{r_i^*}$ and $B_i^* = g^{r_i^*}$ for all $i \in [1, h^*]$. $\mathcal{B}$ outputs $A^* \cdot \{\prod_{i=1}^{h^*}(B_i^*)^{J_i(m_i^*)}\}^{-1} = (g')^\beta = g^{ab}$ as an answer to the CDH problem.

Let abort denote the event that $\mathcal{B}$ aborts the simulation. When abort does not occur, $\mathcal{B}$ perfectly simulates the weak unforgeability experiment to $\mathcal{A}$. Moreover, when abort does not occur and $\mathcal{A}$ wins, $\mathcal{B}$ solves the CDH problem. Thus, $\mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{AOS}},\mathcal{A}}(\lambda) \leq \frac{1}{\Pr[\neg\mathsf{abort}]} \cdot \mathtt{Adv}^{\mathtt{CDH}}_{\mathbb{G},\mathcal{B}}(\lambda)$. As proven in [7], $\frac{1}{\Pr[\neg\mathsf{abort}]}$ is upper bounded by $2 \cdot \{2q(L+1)\}^H$. Its rigorous proof is given in Subsect. B.4. □

4.3 Instantiation

For any $X \in \mathbb{G}$, $\iota_{\mathbb{G}}(X)$ denotes $(1_{\mathbb{G}}, 1_{\mathbb{G}}, X) \in \mathbb{G}^3$. For any $X \in \mathbb{G}_T$, $\iota_{\mathbb{G}_T}(X)$ denotes $(1_{\mathbb{G}_T}, 1_{\mathbb{G}_T}, X) \in \mathbb{G}_T^3$. Given $X \in \mathbb{G}_T$, $\mathit{\Gamma}_{\mathbb{G}_T}(X)$ denotes the 3×3 matrix which has X as the $(3,3)$-th element and $1_{\mathbb{G}_T}$ as any of the other elements. Given $h, g_1, g_2, g_3 \in \mathbb{G}$, $E(h, (g_1, g_2, g_3))$ denotes $(e(h, g_1), e(h, g_2), e(h, g_3)) \in \mathbb{G}_T^3$. For any $\vec{X} = (X_1, X_2, X_3) \in \mathbb{G}^3$ and $\vec{Y} = (Y_1, Y_2, Y_3) \in \mathbb{G}^3$, $F(\vec{X}, \vec{Y}) := \tilde{F}(\vec{X}, \vec{Y})^{1/2} \cdot \tilde{F}(\vec{Y}, \vec{X})^{1/2} \in \mathbb{G}_T^{3\times 3}$, where $\tilde{F}(\vec{X}, \vec{Y}) \in \mathbb{G}_T^{3\times 3}$ contains $e(X_i, Y_j)$ as the (i,j)-th element for all $i, j \in \{1,2,3\}$.

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

1. Generate a key-pair of the simplified ALP LHS scheme. Choose $\alpha \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Choose $g, h, g_1, \cdots, g_{n+3} \xleftarrow{\mathrm{U}} \mathbb{G}$. Choose $u', u_1, \cdots, u_N \xleftarrow{\mathrm{U}} \mathbb{G}$ for $N \in \mathbb{N}$. $H_{\mathbb{G}} : \{0,1\}^N \to \mathbb{Z}_p$ is a function which takes $\tau = \tau[1] \| \cdots \| \tau[N] \in \{0,1\}^N$ and outputs $u' \prod_{i=1}^{N} u_i^{\tau[i]} \in \mathbb{G}$.
2. Generate a key-pair of our AOS scheme in Subsect. 4.2. Choose $\beta \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and $H, U_1, \cdots, U_{\lambda+1}, V_1, \cdots, V_N \xleftarrow{\mathrm{U}} \mathbb{G}$. Note that $H \in \mathbb{G}$ was originally $g' \in \mathbb{G}$.
3. Generate a GS CRS $\boldsymbol{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$ as $\vec{f}_1 := (f_1, 1, g)$, $\vec{f}_2 := (1, f_2, g)$ and $\vec{f}_3 := \vec{f}_1^{\xi_1} \cdot \vec{f}_2^{\xi_2} \cdot (1,1,g)^{-1}$, where $f_1, f_2 \xleftarrow{\mathrm{U}} \mathbb{G}, \xi_1, \xi_2 \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.

Output (pp, mk), where $pp := (\mathbb{G}, \mathbb{G}_T, e, g, g^\alpha, h, \{g_i\}_{i=1}^{n+3}, u', \{u_i\}_{i=1}^{N}, g^\beta, H, \{U_i\}_{i=1}^{\lambda+1}, \{V_i\}_{i=1}^{N}, \boldsymbol{f})$ and $mk := (\alpha, \beta)$.

$\mathtt{KGen}(mk, \mathbf{x}, L, R)$**:** Choose an LHS tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Conduct the following two steps.

1. For each $i \in [1, n]$, let $\boldsymbol{v}_i := (x_i, \underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{n-i}, 0, 0) \in \mathbb{Z}_p^{n+3}$. Let $\boldsymbol{v}_{n+1} := (0, \cdots, 0, 1, 0) \in \mathbb{Z}_p^{n+3}$ and $\boldsymbol{v}_{n+2} := (0, \cdots, 0, 0, 1) \in \mathbb{Z}_p^{n+3}$. For $i \in [1, n+2]$, generate a signature of the ALP LHS scheme on $\boldsymbol{v}_i$ as $\sigma_i := (\sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3}, \sigma_{i,4}) := (\{(\prod_{j=1}^{n+3} g_i^{v_{ij}}) \cdot h^{s_i}\}^\alpha H_{\mathbb{G}}(\tau)^{r_i}, g^{r_i}, g^{s_i}, g^{\alpha \cdot s_i})$, where $r_i, s_i \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.

2. Calculate $C \leftarrow \texttt{Cover}(L, R)$. Each $c \in C$ is parsed as $c[1] \| \cdots \| c[h_c]$ with length $h_c \in [1, \lambda]$. For each $c \in C$, generate a signature of our AOS scheme on $(\tau, c[1], \cdots, c[h_c])$ as $\theta_c := (A_c, B_{c,1}, \cdots, B_{h_c+1}) := (H^\beta \cdot (U_1 \prod_{i=1}^{N} V_i^{\tau[i]})^{t_1} \cdot \prod_{i=1}^{h_c} (U_{i+1} \cdot V_1^{c[i]})^{t_{i+1}}, g^{t_1}, \cdots, g^{t_{h_c+1}})$, where $t_1, \cdots, t_{h_c+1} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.

Output $sk := (\tau, \{\sigma_i\}_{i=1}^{n+2}, \{\theta_c\}_{c \in C})$.

$\texttt{Sig}(sk, M, \mathbf{y})$**:** Parse sk as above. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Assume that $d \in [L, R]$. Conduct the following four steps.

1. Derive an LHS signature on $\boldsymbol{v}' := (d, y_1, \cdots, y_n, M, 1)$. Let $\beta_{n+1} := M$, $\beta_{n+2} := 1$ and $\beta_i := y_i$ for each $i \in [1, n]$. Choose $r' \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Compute $\sigma' := (\sigma'_1, \sigma'_2, \sigma'_3, \sigma'_4) := (\prod_{i=1}^{n+2} \sigma_{i,1}^{\beta_i} \cdot H_{\mathbb{G}}(\tau)^{r'}, \prod_{i=1}^{n+2} \sigma_{i,2}^{\beta_i} \cdot g^{r'}, \prod_{i=1}^{n+2} \sigma_{i,3}^{\beta_i}, \prod_{i=1}^{n+2} \sigma_{i,4}^{\beta_i})$.
2. $d \in [L, R]$ implies that there exists $c \in C$ s.t. c is either identical to or an ancestor of $[d]_2$. Parse c as $(c[1], \cdots, c[h_c])$. Parse θ_c as $(A, B_1, \cdots, B_{h_c+1})$. Compute $\theta' := (A', B'_1, \cdots, B'_{\lambda+1}) := (A \cdot (U_1 \prod_{i=1}^{N} V_i)^{t'_1} \prod_{i=1}^{\lambda} (U_{i+1} \cdot V_1^{d[i]})^{t'_{i+1}}, B_1 \cdot g^{t'_1}, \cdots, B_{h_c+1} \cdot g^{t'_{h_c+1}}, g^{t'_{h_c+2}}, \cdots, g^{t'_{\lambda+1}})$, where $t'_1, \cdots, t'_{\lambda+1} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.
3. Generate GS commitments for all of the following group elements.
   (a) $g^{\tau[i]}$, $g^{1-\tau[i]}$ and $V_i^{\tau[i]}$ (for all $i \in [1, N]$)
   (b) $H_{\mathbb{G}}(\tau)$
   (c) $g_1^{d[i]}$, $g_1^{1-d[i]}$ and $V_1^{d[i]}$ (for all $i \in [1, \lambda]$)
   (d) g_1^d
   (e) σ'_1, σ'_3 and σ'_4
   (f) A'

   They are denoted by $\vec{C}_{\tau[i]}$, $\vec{C}_{1-\tau[i]}$, $\vec{C}'_{\tau[i]}$, $\vec{C}_{H_{\mathbb{G}}(\tau)}$, $\vec{C}_{d[i]}$, $\vec{C}_{1-d[i]}$, $\vec{C}'_{d[i]}$, $\vec{C}_d$, $\vec{C}_{\sigma_1}$, $\vec{C}_{\sigma_3}$, $\vec{C}_{\sigma_4}$ and $\vec{C}_A$. A commitment $\vec{C}$ to an element $X \in \mathbb{G}$ is computed as $\iota_{\mathbb{G}}(X) \cdot \vec{f}_1^{r_X} \cdot \vec{f}_2^{s_X} \cdot \vec{f}_3^{t_X}$, where $r_X, s_X, t_X \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.
4. Generate GS proofs for all of the following PPEs.
   [a] $e(g^{\tau[i]}, g^{1-\tau[i]}) = 1_{\mathbb{G}_T}$, $e(g^{\tau[i]}, g) \cdot e(g^{1-\tau[i]}, g) = e(g, g)$ and $e(g^{\tau[i]}, V_i) = e(g, V_i^{\tau[i]})$ (for all $i \in [1, N]$)
   [b] $e(H_{\mathbb{G}}(\tau), g) = e(u', g) \prod_{i=1}^{N} e(u_i, g^{\tau[i]})$
   [c] $e(g_1^{d[i]}, g_1^{1-d[i]}) = 1_{\mathbb{G}_T}$, $e(g_1^{d[i]}, g_1) \cdot e(g_1^{1-d[i]}, g_1) = e(g_1, g_1)$ and $e(g^{d[i]}, V_1) = e(g, V_1^{d[i]})$ (for all $i \in [1, \lambda]$)
   [d] $e(g_1^d, g) = \prod_{i=1}^{\lambda} e(g_1^{d[i]}, g^{2^{i-1}})$
   [e] $e(\sigma'_1, g) = e(g_1^d, g^\alpha) \cdot e(\prod_{i=1}^{n} g_{i+1}^{y_i} \cdot g_{n+2}^{M} \cdot g_{n+3}^{1}, g^\alpha) \cdot e(h, \sigma'_4) \cdot e(H_{\mathbb{G}}(\tau), \sigma'_2)$
   [f] $e(\sigma'_3, g^\alpha) = e(g, \sigma'_4)$
   [g] $e(A', g) = e(g^\beta, H) \cdot e(U_1, B'_1) \prod_{i=1}^{\lambda} e(U_{i+1}, B'_{i+1}) \prod_{i=1}^{N} e(V_i^{\tau[i]}, B'_1) \prod_{i=1}^{\lambda} e(V_1^{d[i]}, B'_{i+1})$

   All PPEs surrounded by a grey rectangle are quadratic. The others are linear. The generated proofs are denoted by $\vec{\pi}_{\tau[i],mul}$, $\vec{\pi}_{\tau[i],sum}$, $\vec{\pi}_{\tau[i]}$, $\vec{\pi}_{H_{\mathbb{G}}(\tau)}$, $\vec{\pi}_{d[i],mul}$, $\vec{\pi}_{d[i],sum}$, $\vec{\pi}_{d[i]}$, $\vec{\pi}_d$, $\vec{\pi}_{\sigma_1}$, $\vec{\pi}_{\sigma_3}$ and $\vec{\pi}_A$. A GS proof $\vec{\pi}$ for a linear (resp. quadratic) PPE consists of 3 (resp. 9) group elements.

Output a signature σ which is set to

$$\begin{pmatrix} \{\vec{C}_{\tau[i]}, \vec{C}_{1-\tau[i]}, \vec{C}'_{\tau[i]}, \vec{\pi}_{\tau[i],mul}, \vec{\pi}_{\tau[i],sum}, \vec{\pi}_{\tau[i]}\}_{i=1}^{N}, \\ \{\vec{C}_{d[i]}, \vec{C}_{1-d[i]}, \vec{C}'_{d[i]}, \vec{\pi}_{d[i],mul}, \vec{\pi}_{d[i],sum}, \vec{\pi}_{d[i]}\}_{i=1}^{\lambda}, \\ \vec{C}_{H_{\mathbb{G}}(\tau)}, \vec{\pi}_{H_{\mathbb{G}}(\tau)}, \vec{C}_d, \vec{\pi}_d, \vec{C}_{\sigma_1}, \sigma'_2, \vec{C}_{\sigma_3}, \vec{C}_{\sigma_4}, \vec{\pi}_{\sigma_1}, \vec{\pi}_{\sigma_3}, \vec{C}_A, \{B'_i\}_{i=1}^{\lambda+1}, \vec{\pi}_A \end{pmatrix}. \tag{1}$$

$\mathtt{Ver}(\sigma, M, \mathbf{y})$: Each GS proof $\vec{\pi} \in \mathbb{G}^3$ (resp. $\vec{\pi} \in \mathbb{G}^9$) is parsed as (π_1, π_2, π_3) (resp. $(\vec{\pi}_1, \vec{\pi}_2, \vec{\pi}_3)$ with $\vec{\pi}_i \in \mathbb{G}^3$). Output 1 iff all of the 11 equations hold.

1. $F(\vec{C}_{\tau[i]}, \vec{C}_{1-\tau[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{\tau[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, N]$)
2. $E(g, \vec{C}_{\tau[i]}) \cdot E(g, \vec{C}_{1-\tau[i]}) = \iota_{\mathbb{G}_T}(e(g,g)) \prod_{k=1}^{3} E(\pi_{\tau[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, N]$)
3. $E(g, \vec{C}_{H_{\mathbb{G}}(\tau)}) = \iota_{\mathbb{G}_T}(e(u', g)) \prod_{i=1}^{N} E(u_i, \vec{C}_{\tau[i]}) \prod_{k=1}^{3} E(\pi_{\tau[i],k}, \vec{f}_k)$
4. $E(V_i, \vec{C}_{\tau[i]}) = E(g, \vec{C}'_{\tau[i]}) \prod_{k=1}^{3} E(\pi_{\tau[i],k}, \vec{f}_k)$ (for all $i \in [1, N]$)
5. $F(\vec{C}_{d[i]}, \vec{C}_{1-d[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{d[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
6. $E(g, \vec{C}_{d[i]}) \cdot E(g, \vec{C}_{1-d[i]}) = \iota_{\mathbb{G}_T}(e(g,g)) \prod_{k=1}^{3} E(\pi_{d[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
7. $E(g, \vec{C}_d) = \prod_{i=1}^{\lambda} E(g^{2^{i-1}}, \vec{C}_{d[i]}) \prod_{k=1}^{3} E(\pi_{d,k}, \vec{f}_k)$
8. $E(V_1, \vec{C}_{d[i]}) = E(g, \vec{C}'_{d[i]}) \prod_{k=1}^{3} E(\pi_{d[i],k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
9. $E(g, \vec{C}_{\sigma_1}) = E(g^{\alpha}, \vec{C}_d) \cdot \iota_{\mathbb{G}_T}(e(\prod_{i=1}^{n} g_{1+i}^{y_i} \cdot g_{n+2}^{M} \cdot g_{n+3}, g^{\alpha})) \cdot E(h, \vec{C}_{\sigma_4}) \cdot E(\sigma'_2, \vec{C}_{H_{\mathbb{G}}(\tau)}) \prod_{k=1}^{3} E(\pi_{\sigma_1,k}, \vec{f}_k)$
10. $E(g^{\alpha}, \vec{C}_{\sigma_3}) = E(g, \vec{C}_{\sigma_4}) \prod_{k=1}^{3} E(\pi_{\sigma_3,k}, \vec{f}_k)$
11. $E(g, \vec{C}_A) = \iota_{\mathbb{G}_T}(e(g^{\beta}, H) \cdot e(U_1, B'_1) \prod_{i=1}^{\lambda} e(U_{i+1}, B'_{i+1})) \prod_{i=1}^{N} E(B'_1, \vec{C}'_{\tau[i]}) \prod_{i=1}^{\lambda} E(B'_{i+1}, \vec{C}'_{d[i]}) \prod_{k=1}^{3} E(\pi_{A,k}, \vec{f}_k)$

Corollary 1. *Our 1st KARIP scheme is* `UNF` *if the DLIN, CDH and FlexCDH assumptions hold in the group* $\mathbb{G}$*. The scheme is* `PRV` *unconditionally.*

Efficiency Analysis. Every signature is expressed as (1). It consists of $(27N + 27\lambda + 40)$ elements in $\mathbb{G}$. Thus, $|\sigma| = (27N + 27\lambda + 40)|g|$ [bit]. Each secret-key consists of $(\tau, \{\sigma_i\}_{i=1}^{n+2}, \{\theta_c\}_{c \in C})$. τ and $\{\sigma_i\}_{i=1}^{n+2}$ are of N [bit] and $4(n+2)|g|$ [bit], respectively. Size of $\{\theta_c\}_{c \in C}$ is calculated as $|\{\theta_c\}_{c \in C}| = |\{(A_c, B_{c,1}, \cdots, B_{c,h_c+1})\}_{c \in C}| = \sum_{c \in C}(2 + h_c)|g| = (2|C| + \sum_{c \in C} h_c)|g| \leq (\lambda^2 + 5\lambda - 10)|g|$ [bit]. The last upper bound is because of the fact that both $|C|$ (= *the cardinality of the set* C) and $\sum_{c \in C} h_c$ are maximized when $[L, R] = [1, p-2]$ and their maximal values are $2\lambda - 2$ and $\lambda^2 + \lambda - 2$, respectively[4]. Thus, $|sk| = N + \mathcal{O}(n + \lambda^2)|g|$ [bit]. As explained in Subsect. 4.1, the KARIP scheme is key-delegatable. The analysis result is added as the first entry in Table 1.

5 Our 2nd Construction of KARIP

5.1 Construction

Our generic KARIP construction is built by an LHS scheme $\{\mathsf{L.KGen}, \mathsf{L.Sig}, \mathsf{L.Derive}, \mathsf{L.Ver}\}$ and an NIWI proof system $\{\mathsf{N.Setup}, \mathsf{N.Pro}, \mathsf{N.Ver}\}$.

[4] The latter value is obtained by $2 \times (2 + 3 + \cdots + \lambda) = \lambda^2 + \lambda - 2$.

Schemes	$\|sk\|$ [bit]	$\|\sigma\|$ [bit]	KD
Ours 1	$N + \mathcal{O}(n + \lambda^2)\|g\|$	$(27N + 27\lambda + 40)\|g\|$	✓
2	$N + 4(n + 2)\|g\|$	$(18N + 132\lambda + 39)\|g\|$	-
3	$N + 8\|g\|$	$(9n + 18N + 132\lambda + 42)\|g\|$	-

Table 1. Comparison of our KARIP schemes w.r.t. efficiency and key-delegatability.

$\mathtt{Setup}(1^\lambda, L)$**:** $crs \leftarrow \mathsf{N}.\mathtt{Setup}(1^\lambda)$ and $(pk_{\mathsf{L}}, sk_{\mathsf{L}}) \leftarrow \mathsf{L}.\mathtt{KGen}(1^\lambda, n+5)$ with tags whose bit length is $N \in \mathtt{poly}(\lambda)$. Output $pp := (crs, pk_{\mathsf{L}})$ and $mk := sk_{\mathsf{L}}$.

$\mathtt{KGen}(mk, \mathbf{x}, L, R)$**:** Choose a tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. For each $i \in [1, n]$, let $\boldsymbol{v}_i := (x_i, \underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{n-i}, \underbrace{0, 0, 0, 0}_{4}) \in \mathbb{Z}_p^{n+5}$. Let $\boldsymbol{v}_{n+1} := (0, \cdots, 0, L, R, 0, 1) \in \mathbb{Z}_p^{n+5}$ and $\boldsymbol{v}_{n+2} := (0, \cdots, 0, 0, 0, 1, 0) \in \mathbb{Z}_p^{n+5}$. For each $\boldsymbol{v}_i$, generate an LHS signature with tag τ by $\sigma_i \leftarrow \mathsf{L}.\mathtt{Sig}(sk_{\mathsf{L}}, \tau, \boldsymbol{v}_i)$. Output $sk := (\tau, \{\sigma_i\}_{i=1}^{n+2})$.

$\mathtt{Sig}(sk, M, \mathbf{y})$**:** Parse sk as above. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Assume that $d \in [L, R]$. Conduct the following two steps.

1. Generate an LHS signature on $\boldsymbol{v}' := (d, y_1, \cdots, y_n, L, R, M, 1)$ by $\sigma' \leftarrow \mathsf{L}.\mathtt{Derive}(pk_{\mathsf{L}}, \tau, \{\boldsymbol{v}_i, \sigma_i, \beta_i\}_{i=1}^{n+2})$, where $\beta_{n+1} := 1$, $\beta_{n+2} := M$ and $\beta_i := y_i$ for each $i \in [1, n]$.
2. Define the NIWI relation $\mathcal{R}_{\mathsf{N}}$ as follows.
   - A statement $x = (\hat{\mathbf{y}}, \hat{M})$ consists of a vector $\hat{\mathbf{y}} = (\hat{y}_1, \cdots, \hat{y}_n) \in \mathbb{Z}_p^n$ and a message $\hat{M} \in \mathbb{Z}_p$. A witness $w = (\hat{L}, \hat{R}, \hat{d}, \hat{\tau}, \hat{\sigma})$ consists of integers $\hat{L}, \hat{R} \in \mathbb{Z}_p$, an inner product value $\hat{d} \in \mathbb{Z}_p$, an LHS tag $\hat{\tau} \in \{0,1\}^L$ and an LHS signature $\hat{\sigma}$. $\mathcal{R}_{\mathsf{N}}$ takes a statement x and witness w then outputs 1 if both of the two conditions are satisfied.
     1. $1 \leftarrow \mathsf{L}.\mathtt{Ver}(pk_{\mathsf{L}}, \hat{\tau}, \hat{\boldsymbol{v}}, \hat{\sigma})$, where $\hat{\boldsymbol{v}} := (\hat{d}, \hat{y}_1, \cdots, \hat{y}_n, \hat{L}, \hat{R}, \hat{M}, 1)$.
     2. $\hat{d} \in [\hat{L}, \hat{R}] \pmod p$.

   If we set $x := (\mathbf{y}, M)$ and $w := (L, R, d, \tau, \sigma)$, it obviously holds that $1 \leftarrow \mathcal{R}_{\mathsf{N}}(x, w)$. Output $\sigma \leftarrow \mathsf{N}.\mathtt{Pro}(crs, x, w)$.

$\mathtt{Ver}(\sigma, M, \mathbf{y})$**:** Set $x := (\mathbf{y}, M)$ and output $1/0 \leftarrow \mathsf{N}.\mathtt{Ver}(crs, x, \sigma)$.

Because of the page restriction, we omit the proof of the following theorem. It is given in Subsect. B.1 and basically the same as the proofs of the security theorems of our 1st KARIP construction.

Theorem 5. *The construction is* `UNF` *if the LHS scheme is* `wUNF`*, and the NIWI system is* `WI` *and* `WE`*. It is* `PRV` *if the NIWI system is* `WI`*.*

5.2 Instantiation

We use the simplified ALP LHS scheme [9] and the GS NIWI proof [8].

$\mathtt{Setup}(1^\lambda, L)$**:** Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ whose order is a prime p. Conduct the following two steps.

1. Generate a key-pair of the simplified ALP LHS scheme [4]. Choose $\alpha \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Choose $g, h, g_1, \cdots, g_{n+5} \xleftarrow{\mathrm{U}} \mathbb{G}$. Choose $u', u_1, \cdots, u_N \xleftarrow{\mathrm{U}} \mathbb{G}$ for $N \in \mathbb{N}$. $H_{\mathbb{G}} : \{0,1\}^N \to \mathbb{Z}_p$ is a function which takes $\tau = \tau[1] \| \cdots \| \tau[N] \in \{0,1\}^N$ and outputs $u' \prod_{i=1}^{N} u_i^{\tau[i]} \in \mathbb{G}$.
2. Generate a GS CRS $\boldsymbol{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$.

Output $(pp, mk) := ((\mathbb{G}, \mathbb{G}_T, e, g, g^\alpha, h, \{g_i\}_{i=1}^{n+5}, u', \{u_i\}_{i=1}^{N}, \boldsymbol{f}), \alpha)$.

`KGen`$(mk, \mathbf{x}, L, R)$**:** Choose an LHS tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. For each $i \in [1, n]$, let $\boldsymbol{v}_i := (x_i, \underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{n-i}, \underbrace{0, 0, 0, 0}_{4}) \in \mathbb{Z}_p^{n+5}$. Let $\boldsymbol{v}_{n+1} := (0, \cdots, 0, L, R, 0, 1) \in \mathbb{Z}_p^{n+5}$ and $\boldsymbol{v}_{n+2} := (0, \cdots, 0, 0, 0, 1, 0) \in \mathbb{Z}_p^{n+5}$. For $i \in [1, n+2]$, generate a signature of the ALP LHS scheme on $\boldsymbol{v}_i$ as $\sigma_i := (\sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3}, \sigma_{i,4}) := (\{(\prod_{j=1}^{n+5} g_i^{v_{ij}}) \cdot h^{s_i}\}^\alpha H_{\mathbb{G}}(\tau)^{r_i}, g^{r_i}, g^{s_i}, g^{\alpha \cdot s_i})$, where $r_i, s_i \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.

Output $sk := (\tau, \{\sigma_i\}_{i=1}^{n+2})$.

`Sig`$(sk, M, \mathbf{y})$**:** Parse sk as above. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Assume that $d \in [L, R]$. Firstly, conduct the following three steps.

1. Derive an LHS signature on $\boldsymbol{v}' := (d, y_1, \cdots, y_n, L, R, M, 1)$. Let $\beta_{n+1} := 1$, $\beta_{n+2} := M$ and $\beta_i := y_i$ for any $i \in [1, n]$. Compute $\sigma' := (\prod_{i=1}^{n+2} \sigma_{i,1}^{\beta_i} \cdot H_{\mathbb{G}}(\tau)^{r'}, \prod_{i=1}^{n+2} \sigma_{i,2}^{\beta_i} \cdot g^{r'}, \prod_{i=1}^{n+2} \sigma_{i,3}^{\beta_i}, \prod_{i=1}^{n+2} \sigma_{i,4}^{\beta_i})$, where $r' \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.
2. Generate GS commitments for all of the following group elements.
   (a) $g^{\tau[i]}$ and $g^{1-\tau[i]}$ (for all $i \in [1, N]$)
   (b) $H_{\mathbb{G}}(\tau)$
   (c) $g_1^{d[i]}$, $g_1^{1-d[i]}$, $g_{n+2}^{L[i]}$, $g_{n+2}^{1-L[i]}$, $g_{n+3}^{R[i]}$ and $g_{n+3}^{1-R[i]}$ (for all $i \in [1, \lambda]$)
   (d) g_1^d, g_{n+2}^L and g_{n+3}^R
   (e) σ'_1, σ'_3 and σ'_4

   They are denoted by $\vec{C}_{\tau[i]}$, $\vec{C}_{1-\tau[i]}$, $\vec{C}_{H_{\mathbb{G}}(\tau)}$, $\vec{C}_{d[i]}$, $\vec{C}_{1-d[i]}$, $\vec{C}_{L[i]}$, $\vec{C}_{1-L[i]}$, $\vec{C}_{R[i]}$, $\vec{C}_{1-R[i]}$, $\vec{C}_d$, $\vec{C}_L$, $\vec{C}_R$, $\vec{C}_{\sigma_1}$, $\vec{C}_{\sigma_3}$ and $\vec{C}_{\sigma_4}$.
3. Generate GS proofs for all of the following PPEs.
   [a] $e(g^{\tau[i]}, g^{1-\tau[i]}) = 1_{\mathbb{G}_T}$ and $e(g^{\tau[i]}, g) \cdot e(g^{1-\tau[i]}, g) = e(g, g)$ (for all $i \in [1, N]$)
   [b] $e(H_{\mathbb{G}}(\tau), g) = e(u', g) \prod_{i=1}^{N} e(u_i, g^{\tau[i]})$
   [c] $e(g_1^{d[i]}, g_1^{1-d[i]}) = 1_{\mathbb{G}_T}$, $e(g_1^{d[i]}, g_1) \cdot e(g_1^{1-d[i]}, g_1) = e(g_1, g_1)$, $e(g_{n+2}^{L[i]}, g_{n+2}^{1-L[i]}) = 1_{\mathbb{G}_T}$, $e(g_{n+2}^{L[i]}, g) \cdot e(g_{n+2}^{1-L[i]}, g) = e(g_{n+2}, g)$, $e(g_{n+3}^{R[i]}, g_{n+3}^{1-R[i]}) = 1_{\mathbb{G}_T}$ and $e(g_{n+3}^{R[i]}, g) \cdot e(g_{n+3}^{1-R[i]}, g) = e(g_{n+3}, g)$ (for all $i \in [1, \lambda]$)
   [d] $e(g_1^d, g) = \prod_{i=1}^{\lambda} e(g_1^{d[i]}, g^{2^{i-1}})$, $e(g_{n+2}^L, g) = \prod_{i=1}^{\lambda} e(g_{n+2}^{L[i]}, g^{2^{i-1}})$ and $e(g_{n+3}^R, g) = \prod_{i=1}^{\lambda} e(g_{n+3}^{R[i]}, g^{2^{i-1}})$
   [e] $e(\sigma'_1, g) = e(g_1^d, g^\alpha) \cdot e(\prod_{i=1}^{n} g_{i+1}^{y_i} \cdot g_{n+4}^M \cdot g_{n+5}^1, g^\alpha) \cdot e(g_{n+2}^L, g^\alpha) \cdot e(g_{n+3}^R, g^\alpha) \cdot e(h, \sigma'_4) \cdot e(H_{\mathbb{G}}(\tau), \sigma'_2)$
   [f] $e(\sigma'_3, g^\alpha) = e(g, \sigma'_4)$

   They are denoted by $\vec{\pi}_{\tau[i],mul}$, $\vec{\pi}_{\tau[i],sum}$, $\vec{\pi}_{H_{\mathbb{G}}(\tau)}$, $\vec{\pi}_{d[i],mul}$, $\vec{\pi}_{d[i],sum}$, $\vec{\pi}_{L[i],mul}$, $\vec{\pi}_{L[i],sum}$, $\vec{\pi}_{R[i],mul}$, $\vec{\pi}_{R[i],sum}$, $\vec{\pi}_d$, $\vec{\pi}_L$, $\vec{\pi}_R$, $\vec{\pi}_{\sigma_1}$ and $\vec{\pi}_{\sigma_3}$.

What remains is proving $d \in [L, R] \pmod p$.
Firstly, we prove $d \geq L$. If $d \geq L$, there is only one index $i \in [1, \lambda+1]$ s.t.

$$d[i] = 1 \bigwedge L[i] = 0 \bigwedge_{j=1}^{i-1} d[i] = L[i]. \tag{2}$$

For each $i \in [1, \lambda+1]$, a Boolean variable $A_i \in \{0,1\}$ is defined to be 1 (resp. 0) if the condition (2) holds (resp. otherwise). It is obviously true that A_i is 1 iff $d \geq L$. Additionally, for each $i \in [1, \lambda]$, define three Boolean variables $B_i, C_i, D_i \in \{0,1\}$. B_i is 1 iff $\bigwedge_{j=1}^{i} d[j] = L[j]$. C_i is 1 iff $d[i] = 1 \bigwedge L[i] = 0$. D_i is 1 iff $d[i] = L[i]$.
Conduct the following two steps.

1. Generate GS commitments for all of the following group elements.
   (f) $g_1^{B_i}$, $g_1^{C_i}$ and $g_1^{D_i}$ (for all $i \in [1, \lambda]$)
   They are denoted by $\vec{C}_{B_i}$, $\vec{C}_{C_i}$ and $\vec{C}_{D_i}$.
2. Generate GS proofs for all of the following PPEs.
   [g] $e(g_1^{C_i}, g_{n+2}) = e(g_1^{d[i]}, g_{n+2}^{1-L[i]})$ (for all $i \in [1, \lambda]$)
   [h] $e(g_1^{D_i}, g_{n+2}) = e(g_1^{d[i]}, g_{n+2}^{L[i]}) \cdot e(g_1^{1-d[i]}, g_{n+2}^{1-L[i]})$ (for all $i \in [1, \lambda]$)
   [i] $e(g_1^{B_1}, g_1) = e(g_1, g_1^{D_1})$
   [j] $e(g_1^{B_i}, g_1) = e(g_1^{B_{i-1}}, g_1^{D_i})$ (for all $i \in [2, \lambda]$)
   [k] $e(g_1, g_1^{C_1}) \prod_{i=1}^{\lambda} e(g_1^{B_{i-1}}, g_1^{C_i}) \cdot e(g_1^{B_\lambda}, g_1) = e(g_1, g_1)$
   For the equation [k], the term $e(g_1, g_1^{C_1})$ (resp. $e(g_1^{B_{i-1}}, g_1^{C_i})$, $e(g_1^{B_\lambda}, g_1)$) is equivalent to $e(g_1, g_1)^{A_1}$ (resp. $e(g_1, g_1)^{A_i}$, $e(g_1, g_1)^{A_{\lambda+1}}$). Thus, the left side of the equation [k] is equivalent to $e(g_1, g_1)^{\sum_{i=1}^{\lambda+1} A_i}$. The generated proofs are denoted by $\vec{\pi}_{C_i}$, $\vec{\pi}_{D_i}$, $\vec{\pi}_{B_1}$, $\vec{\pi}_{B_i}$ and $\vec{\pi}_A$, respectively.

Next, we prove $d \leq R$. If $d \leq R$, there is only one index $i \in [1, \lambda+1]$ s.t.

$$d[i] = 0 \bigwedge R[i] = 1 \bigwedge_{j=1}^{i-1} d[i] = R[i]. \tag{3}$$

For each $i \in [1, \lambda+1]$, a Boolean variable $A'_i \in \{0,1\}$ is defined to be 1 (resp. 0) if the condition (3) holds (resp. otherwise). It is obviously true that A'_i is 1 iff $d \leq R$. Additionally, for each $i \in [1, \lambda]$, define three Boolean variables $E_i, F_i, G_i \in \{0,1\}$. E_i is 1 iff $\bigwedge_{j=1}^{i} d[j] = R[j]$. F_i is 1 iff $d[i] = 1 \bigwedge R[i] = 0$. G_i is 1 iff $d[i] = R[i]$.
Conduct the following two steps.

1. Generate GS commitments for all of the following group elements.
   (g) $g_1^{E_i}$, $g_1^{F_i}$ and $g_1^{G_i}$ (for all $i \in [1, \lambda]$)
   They are denoted by $\vec{C}_{E_i}$, $\vec{C}_{F_i}$ and $\vec{C}_{G_i}$.
2. Generate GS proofs for all of the following PPEs.
   [l] $e(g_1^{F_i}, g_{n+3}) = e(g_1^{1-d[i]}, g_{n+3}^{R[i]})$ (for all $i \in [1, \lambda]$)
   [m] $e(g_1^{G_i}, g_{n+3}) = e(g_1^{d[i]}, g_{n+3}^{R[i]}) \cdot e(g_1^{1-d[i]}, g_{n+3}^{1-R[i]})$ (for all $i \in [1, \lambda]$)
   [n] $e(g_1^{E_1}, g_1) = e(g_1, g_1^{G_1})$

[o] $e(g_1^{E_i}, g_1) = e(g_1^{E_{i-1}}, g_1^{G_i})$ (for all $i \in [2, \lambda]$)

[p] $e(g_1, g_1^{F_1}) \prod_{i=1}^{\lambda} e(g_1^{E_{i-1}}, g_1^{F_i}) \cdot e(g_1^{E_\lambda}, g_1) = e(g_1, g_1)$

They are denoted by $\vec{\pi}_{F_i}$, $\vec{\pi}_{G_i}$, $\vec{\pi}_{E_1}$, $\vec{\pi}_{E_i}$ and $\vec{\pi}_{A'}$.

Output a signature σ which is set to

$$\begin{pmatrix} \{\vec{C}_{\tau[i]}, \vec{C}_{1-\tau[i]}, \vec{\pi}_{\tau[i],mul}, \vec{\pi}_{\tau[i],sum}\}_{i=1}^{N}, \\ \{\{\vec{C}_{x[i]}, \vec{C}_{1-x[i]}, \vec{\pi}_{x[i],mul}, \vec{\pi}_{x[i],sum}\}_{i=1}^{\lambda}, \vec{C}_x, \vec{\pi}_x\}_{x\in\{d,L,R\}}, \\ \vec{C}_{H_{\mathbb{G}}(\tau)}, \vec{\pi}_{H_{\mathbb{G}}(\tau)}, \vec{C}_{\sigma_1}, \sigma_2', \vec{C}_{\sigma_3}, \vec{C}_{\sigma_4}, \vec{\pi}_{\sigma_1}, \vec{\pi}_{\sigma_3}, \vec{\pi}_A, \vec{\pi}_{A'}, \\ \{\vec{C}_{B_i}, \vec{C}_{C_i}, \vec{C}_{D_i}, \vec{\pi}_{B_i}, \vec{\pi}_{C_i}, \vec{\pi}_{D_i}, \vec{C}_{E_i}, \vec{C}_{F_i}, \vec{C}_{G_i}, \vec{\pi}_{E_i}, \vec{\pi}_{F_i}, \vec{\pi}_{G_i}\}_{i=1}^{\lambda} \end{pmatrix}. \tag{4}$$

$\mathtt{Ver}(\sigma, M, \mathbf{y})$**:** Each GS proof $\vec{\pi} \in \mathbb{G}^3$ (resp. $\vec{\pi} \in \mathbb{G}^9$), composed of 3 (resp. 9) elements in $\mathbb{G}$, is parsed as (π_1, π_2, π_3) (resp. $(\vec{\pi}_1, \vec{\pi}_2, \vec{\pi}_3)$ with $\vec{\pi}_i \in \mathbb{G}^3$). Output 1 iff all of the following equations hold.

1. $F(\vec{C}_{\tau[i]}, \vec{C}_{1-\tau[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{\tau[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, N]$)
2. $E(g, \vec{C}_{\tau[i]}) \cdot E(g, \vec{C}_{1-\tau[i]}) = \iota_{\mathbb{G}_T}(e(g,g)) \prod_{k=1}^{3} E(\pi_{\tau[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, N]$)
3. $E(g, \vec{C}_{H_{\mathbb{G}}(\tau)}) = \iota_{\mathbb{G}_T}(e(u', g)) \prod_{i=1}^{N} E(u_i, \vec{C}_{\tau[i]}) \prod_{k=1}^{3} E(\pi_{\tau[i],k}, \vec{f}_k)$
4. $F(\vec{C}_{d[i]}, \vec{C}_{1-d[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{d[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
5. $E(g, \vec{C}_{d[i]}) \cdot E(g, \vec{C}_{1-d[i]}) = \iota_{\mathbb{G}_T}(e(g,g)) \prod_{k=1}^{3} E(\pi_{d[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
6. $E(g, \vec{C}_d) = \prod_{i=1}^{\lambda} E(g^{2^{i-1}}, \vec{C}_{d[i]}) \prod_{k=1}^{3} E(\pi_{d,k}, \vec{f}_k)$
7. $E(g, \vec{C}_{\sigma_1}) = E(g^\alpha, \vec{C}_d) \cdot \iota_{\mathbb{G}_T}(e(\prod_{i=1}^{n} g_{1+i}^{y_i} \cdot g_{n+4}^{M} \cdot g_{n+5}, g^\alpha)) \cdot E(g^\alpha, \vec{C}_L) \cdot E(g^\alpha, \vec{C}_R) \cdot E(h, \vec{C}_{\sigma_4}) \cdot E(\sigma_2', \vec{C}_{H_{\mathbb{G}}(\tau)}) \prod_{k=1}^{3} E(\pi_{\sigma_1,k}, \vec{f}_k)$
8. $E(g^\alpha, \vec{C}_{\sigma_3}) = E(g, \vec{C}_{\sigma_4}) \prod_{k=1}^{3} E(\pi_{\sigma_3,k}, \vec{f}_k)$
9. $F(\vec{C}_{L[i]}, \vec{C}_{1-L[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{L[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
10. $E(g_{n+2}, \vec{C}_{L[i]}) \cdot E(g_{n+2}, \vec{C}_{1-L[i]}) = \iota_{\mathbb{G}_T}(e(g_{n+2}, g)) \prod_{k=1}^{3} E(\pi_{L[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
11. $E(g, \vec{C}_L) = \prod_{i=1}^{\lambda} E(g^{2^{i-1}}, \vec{C}_{L[i]}) \prod_{k=1}^{3} E(\pi_{L,k}, \vec{f}_k)$
12. $F(\vec{C}_{R[i]}, \vec{C}_{1-R[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{R[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
13. $E(g_{n+3}, \vec{C}_{R[i]}) \cdot E(g_{n+3}, \vec{C}_{1-R[i]}) = \iota_{\mathbb{G}_T}(e(g_{n+3}, g)) \prod_{k=1}^{3} E(\pi_{R[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
14. $E(g, \vec{C}_R) = \prod_{i=1}^{\lambda} E(g^{2^{i-1}}, \vec{C}_{R[i]}) \prod_{k=1}^{3} E(\pi_{R,k}, \vec{f}_k)$
15. $F(\iota_{\mathbb{G}}(g_{n+2}), \vec{C}_{C_i}) = F(\vec{C}_{d[i]}, \vec{C}_{1-L[i]}) \prod_{k=1}^{3} F(\vec{\pi}_{C_i,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
16. $F(\iota_{\mathbb{G}}(g_{n+2}), \vec{C}_{D_i}) = F(\vec{C}_{d[i]}, \vec{C}_{L[i]}) \cdot F(\vec{C}_{1-d[i]}, \vec{C}_{1-L[i]}) \prod_{k=1}^{3} F(\vec{\pi}_{D_i,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
17. $E(g_1, \vec{C}_{B_1}) = E(g_1, \vec{C}_{D_1}) \prod_{k=1}^{3} E(\pi_{B_1,k}, \vec{f}_k)$
18. $F(\iota_{\mathbb{G}}(g_1), \vec{C}_{B_i}) = F(\vec{C}_{B_{i-1}}, \vec{C}_{D_i}) \prod_{k=1}^{3} F(\vec{\pi}_{B_i,k}, \vec{f}_k)$ (for all $i \in [2, \lambda]$)
19. $F(\iota_{\mathbb{G}}(g_1), \vec{C}_{C_1}) \prod_{i=1}^{\lambda} \cdot F(\vec{C}_{B_{i-1}}, \vec{C}_{C_i}) \cdot F(\iota_{\mathbb{G}}(g_1), \vec{C}_{B_\lambda}) = \Gamma_{\mathbb{G}_T}(e(g_1, g_1)) \prod_{k=1}^{3} F(\vec{\pi}_{A,k}, \vec{f}_k)$
20. $F(\iota_{\mathbb{G}}(g_{n+3}), \vec{C}_{F_i}) = F(\vec{C}_{1-d[i]}, \vec{C}_{R[i]}) \prod_{k=1}^{3} F(\vec{\pi}_{F_i,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)

21. $F(\iota_{\mathbb{G}}(g_{n+3}), \vec{C}_{G_i}) = F(\vec{C}_{d[i]}, \vec{C}_{R[i]}) \cdot F(\vec{C}_{1-d[i]}, \vec{C}_{1-R[i]}) \prod_{k=1}^{3} F(\vec{\pi}_{G_i,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
22. $E(g_1, \vec{C}_{E_1}) = E(g_1, \vec{C}_{G_1}) \prod_{k=1}^{3} E(\pi_{E_1,k}, \vec{f}_k)$
23. $F(\iota_{\mathbb{G}}(g_1), \vec{C}_{E_i}) = F(\vec{C}_{E_{i-1}}, \vec{C}_{G_i}) \prod_{k=1}^{3} F(\vec{\pi}_{E_i,k}, \vec{f}_k)$ (for all $i \in [2, \lambda]$)
24. $F(\iota_{\mathbb{G}}(g_1), \vec{C}_{F_1}) \prod_{i=1}^{\lambda} \cdot F(\vec{C}_{E_{i-1}}, \vec{C}_{F_i}) \cdot F(\iota_{\mathbb{G}}(g_1), \vec{C}_{E_\lambda}) = \Gamma_{\mathbb{G}_T}(e(g_1, g_1)) \prod_{k=1}^{3} F(\vec{\pi}_{A',k}, \vec{f}_k)$

Corollary 2. *Our 2nd KARIP scheme is* `UNF` *if the DLIN, CDH and FlexCDH assumptions hold in the group* $\mathbb{G}$*. The scheme is* `PRV` *unconditionally.*

Efficiency Analysis. Every secret-key sk consists of a tag $\tau \in \{0,1\}^N$ and $4(n+2)$ group elements, i.e., $|sk| = N + 4(n+2)|g|$ [bit]. Every signature σ is expressed as (4). Its size is calculated by summing up all of the elements' size, i.e., $|\sigma| = (18N + 126\lambda + 58)|g|$ [bit]. Refer to Table 1.

6 Our 3rd Construction of KARIP

6.1 Construction

Hash Function. A hash function consists of the following two algorithms. Key-generation $\mathtt{KGen}$ is a probabilistic polynomial-time algorithm which takes a security parameter 1^λ with $\lambda \in \mathbb{N}$, then outputs a hash key hk. Evaluation $\mathtt{Eval}$ takes the hash key hk and a message M, then outputs a hash value $h \in \{0,1\}^l$ with $l \in \mathtt{poly}(\lambda)$. Its security is collision-resistance. A hash function is collision-resistant if for any $\lambda \in \mathbb{N}$ and any PPT algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ receives a hash key $hk \leftarrow \mathtt{KGen}(1^\lambda)$, then finds two messages M, M' s.t. $M \neq M' \wedge \mathtt{Eval}(hk, M) = \mathtt{Eval}(hk, M')$ is negligible.

Construction. Our generic KARIP construction is built by an LHS scheme $\{\mathsf{L}.\mathtt{KGen}, \mathsf{L}.\mathtt{Sig}, \mathsf{L}.\mathtt{Derive}, \mathsf{L}.\mathtt{Ver}\}$, an NIWI proof system $\{\mathsf{N}.\mathtt{Setup}, \mathsf{N}.\mathtt{Pro}, \mathsf{N}.\mathtt{Ver}\}$ and a collision-resistant hash function $\{\mathsf{H}.\mathtt{KGen}, \mathsf{H}.\mathtt{Eval}\}$.

$\mathtt{Setup}(1^\lambda, L)$**:** Generate $crs \leftarrow \mathsf{N}.\mathtt{Setup}(1^\lambda)$, $(pk_\mathsf{L}, sk_\mathsf{L}) \leftarrow \mathsf{L}.\mathtt{KGen}(1^\lambda, n+4)$ whose bit length of each tag is $N \in \mathtt{poly}(\lambda)$ and $hk \leftarrow \mathsf{H}.\mathtt{KGen}(1^\lambda)$. Output $pp := (crs, pk_\mathsf{L}, hk)$ and $mk := sk_\mathsf{L}$.

$\mathtt{KGen}(mk, \mathbf{x}, L, R)$**:** Choose a tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Let $\boldsymbol{v}_1 := (x_1, x_2, \cdots, x_n, L, R, 0, 1) \in \mathbb{Z}_p^{n+4}$ and $\boldsymbol{v}_2 := (0, \cdots, 0, 1, 0) \in \mathbb{Z}_p^{n+4}$. For each $i \in \{1,2\}$, generate an LHS signature with tag τ by $\sigma_i \leftarrow \mathsf{L}.\mathtt{Sig}(sk_\mathsf{L}, \tau, \boldsymbol{v}_i)$. Output $sk := (\tau, \{\sigma_i\}_{i=1}^2)$.

$\mathtt{Sig}(sk, M, \mathbf{y})$**:** Parse sk as above. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Assume that $d \in [L, R]$. Conduct the following three steps.

1. Let $h \leftarrow \mathsf{H}.\mathtt{Eval}(hk, (\mathbf{y}, M))$. Generate an LHS signature on $\boldsymbol{v}' := (x_1, \cdots, x_n, L, R, h, 1)$ by $\sigma' \leftarrow \mathsf{L}.\mathtt{Derive}(pk_\mathsf{L}, \tau, \{\boldsymbol{v}_i, \sigma_i, \beta_i\}_{i=1}^{n+4})$, where $\beta_1 := 1$ and $\beta_2 := h$.
2. Define the NIWI relation $\mathcal{R}_\mathsf{N}$ as follows.

– A statement $x = (\hat{\mathbf{y}}, \hat{M})$ consists of a signature-vector $\hat{\mathbf{y}} = (\hat{y}_1, \cdots, \hat{y}_n) \in \mathbb{Z}_p^n$ and a message $M \in \mathbb{Z}_p$. A witness $w = (\hat{\mathbf{x}}, \hat{L}, \hat{R}, \hat{d}, \hat{\tau}, \hat{\sigma})$ consists of a key-vector $\hat{\mathbf{x}} \in \mathbb{Z}_p^n$, integer $\hat{L}, \hat{R} \in \mathbb{Z}_p$, an inner product value $\hat{d} \in \mathbb{Z}_p$, an LHS tag $\hat{\tau} \in \{0,1\}^L$, and an LHS signature $\hat{\sigma}$. $\mathcal{R}_{\mathsf{N}}$ takes a statement x and witness w then outputs 1 if all the following three conditions are satisfied.
  1. $1 \leftarrow \mathsf{L.Ver}(pk_{\mathsf{L}}, \hat{\tau}, \hat{\boldsymbol{v}}, \hat{\sigma})$, where $\hat{\boldsymbol{v}} := (\hat{x}_1, \cdots, \hat{x}_n, \hat{L}, \hat{R}, \hat{h}, 1)$ and $\hat{h} = \mathsf{H.Eval}(hk, (\hat{\mathbf{y}}, \hat{M}))$.
  2. $\hat{d} = \langle \hat{\mathbf{x}}, \hat{\mathbf{y}} \rangle \pmod{p}$.
  3. $\hat{d} \in [\hat{L}, \hat{R}]$.

  If we set $x := (\mathbf{y}, M)$ and $w := (\mathbf{x}, L, R, d, \tau, \sigma)$, it obviously holds that $1 \leftarrow \mathcal{R}_{\mathsf{N}}(x, w)$. Output $\sigma \leftarrow \mathsf{N.Pro}(crs, x, w)$.

$\mathtt{Ver}(\sigma, M, \mathbf{y})$**:** Set $x := (\mathbf{y}, M)$ and output $1/0 \leftarrow \mathsf{N.Ver}(crs, x, \sigma)$.

Proof of the following theorem is given in Subsect. B.2.

Theorem 6. *Our 3rd KARIP scheme is* ***UNF*** *if the LHS scheme is* ***wUNF****, and the NIWI system is* ***WI*** *and* ***WE****, and the hash function is* ***CR****. The scheme is* ***PRV*** *if the NIWI system is* ***WI****.*

6.2 Instantiation

As our 1st and 2nd instantiations, we use the simplified ALP LHS scheme [9] and the GS NIWI proof [8]. We describe the full construction in Subsect. C.2 due to the page restriction. In key-generation, for the two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2$, we generate a signature σ_i of the simplified ALP LHS. Every secret-key sk consists of a tag τ and only 8 group elements, i.e., $|sk| = N + 8|g|$ [bit]. Signing algorithm is almost the same as the one of our 2nd instantiation in Subsect. 5.2. Since the vector $\boldsymbol{v}'$ of the LHS signature σ' has a form of $\boldsymbol{v}' = (x_1, \cdots, x_n, L, R, h, 1)$, the signer needs to additionally generate (1) GS commitments $\vec{C}_{x_i}, \vec{C}_{x_i} \in \mathbb{G}^3$ to $g^{x_i} \in \mathbb{G}$ and $g_i^{x_i} \in \mathbb{G}$ for each $i \in [1, n]$ and (2) GS proofs $\vec{\pi}_{x_i}, \vec{\pi}_{d,ip}$ for the PPEs $e(g_i^{x_i}, g) = e(g_i, g^{x_i})$ and $e(g^d, g) = \prod_{i=1}^n e(g^{x_i}, g^{y_i})$. Signature size is derived by simply adding bit length of newly generated GS commitments and proofs to signature size of our 2nd instantiated scheme, i.e., $|\sigma| = (6n + 18N + 126\lambda + 65)|g|$ [bit].

The 3rd instantiated scheme is the only one whose secret-key size is independent of n, and simultaneously the only one whose signature size is dependent on n. In comparison between the 1st and 2nd ones, the former has a disadvantage that its secret-key increases linearly with λ^2, but has an advantage that signature size is approximately one fifth of the size of the latter (if we ignore their constants and N-terms). Remind that only the 1st one is key-delegatable.

7 Applications of KARIP

[9] showed that an ARIP scheme is transformed into any of the following 7 ABS primitives, (1) ABS for range evaluation (RE) of polynomials (AREP), (2)

ABS for RE of weighted averages (AREWA), (3) fuzzy identity-based signatures (FIBS), (4) time-specific signatures (TSS) [13,10], (5) ABS for RE of Hamming distance (AREHD), (6) ABS for RE of Euclidean distance (AREED) and (7) ABS for hyperellipsoid predicates (AHEP). Their definitions are in Subsect. A.3. The same transformations work for KARIP. A KARIP scheme is transformed into any of *key-range* versions of the 7 ABS primitives. We emphasize that key-delegatability is inherited. If we use a key-delegatable KARIP scheme such as our 1st instantiated scheme, we obtain a key-delegatable key-range ABS scheme.

References

1. M. Abdalla, J. Birkett, D. Catalano, A.W Dent, J. Malone-Lee, G. Neven, J.C.N. Schuldt, and N.P. Smart. Wildcarded identity-based encryption. *Journal of Cryptology*, 24(1):42–82, 2011.
2. 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.
3. N. Attrapadung, B. Libert, and T. Peters. Computing on authenticated data: New privacy definitions and constructions. In *ASIACRYPT 2012*, pp. 367–385. Springer, 2012.
4. N. Attrapadung, B. Libert, and T. Peters. Efficient completely context-hiding quotable and linearly homomorphic signatures. In *PKC 2013*, pp. 386–404. Springer, 2013.
5. D. Boneh, D. Freeman, J. Katz, and B. Waters. Signing a linear subspace: Signature schemes for network coding. In *PKC 2009*, pp. 68–87. Springer, 2009.
6. D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC 2007*, pp. 535–554. Springer, 2007.
7. S. Chatterjee and P. Sarkar. Practical hybrid (hierarchical) identity-based encryption schemes based on the decisional bilinear diffie-hellman assumption. *International Journal of Applied Cryptography (IJACT)*, 3(1):47–83, 2013.
8. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT 2008*, pp. 415–432. Springer, 2008.
9. M. Ishizaka and K. Fukushima. Attribute-based signatures for range of inner product and its applications. In *ICISC 2022*, pp. 382–407. Springer, 2022.
10. M. Ishizaka and S. Kiyomoto. Time-specific signatures. In *ISC 2020*, pp. 20–38. Springer, 2020.
11. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, pp. 146–162. Springer, 2008.
12. E. Kiltz, A. Mityagin, S. Panjwani, and B. Raghavan. Append-only signatures. In *ICALP 2005*, pp. 434–445. Springer, 2005.
13. K. G. Paterson and E. A. Quaglia. Time-specific encryption. In *SCN 2010*, pp. 1–16. Springer, 2010.
14. A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT 2005*, pp. 457–473. Springer, 2005.
15. Y. Sakai, N. Attrapadung, and G. Hanaoka. Attribute-based signatures for circuits from bilinear map. In *PKC 2016*, pp. 283–300. Springer, 2016.

A Omitted Definitions

A.1 Unforgeability and Strong/Complete Context-Hiding of LHS

Unforgeability. For unforgeablity (UNF) of LHS, we consider the following experiment, where a PPT algorithm $\mathcal{A}$ receives adaptively accesses three type oracles, namely signing $\mathfrak{Sign}$, derivation $\mathfrak{Derive}$ and revelation $\mathfrak{Reveal}$, then outputs a forged signature σ^*. $\mathcal{H}$ denotes the space of handles used for the queue Q whose initial content is $\emptyset$ (empty).

$\boldsymbol{Expt}^{\mathtt{UNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{A}}(1^\lambda, n)$:

1. $(pk, sk) \leftarrow \mathtt{Setup}(1^\lambda, n)$. $(\tau^* \in \{0,1\}^*, \boldsymbol{v}^* \in \mathbb{Z}_p^n, \sigma^*) \leftarrow \mathcal{A}^{\mathfrak{Sign},\mathfrak{Derive},\mathfrak{Reveal}}(pk)$.

- $\mathfrak{Sign}(\tau \in \{0,1\}^*, \boldsymbol{v} \in \mathbb{Z}_p^n)$:
  Choose an unused handle $h \xleftarrow{\mathrm{U}} \mathcal{H}$. $\sigma \leftarrow \mathtt{Sig}(sk, \tau, \boldsymbol{v})$. $Q := Q \cup \{(h, \tau, \boldsymbol{v}, \sigma)\}$. **Rtrn** h.
- $\mathfrak{Derive}(\tau \in \{0,1\}^*, \{h_i \in \mathcal{H}, \boldsymbol{v}_i \in \mathbb{Z}_p^n, \beta_i \in \mathbb{Z}_p\}_{i=1}^l)$:
  Rtrn $\perp$ if $\exists i \in [1,l]$ s.t. $[\nexists(\boldsymbol{v}_i, \sigma_i)$ s.t. $(h_i, \tau, \boldsymbol{v}_i, \sigma_i) \notin Q]$.
  Choose an unused handle $h \xleftarrow{\mathrm{U}} \mathcal{H}$. $\overline{\sigma} \leftarrow \mathtt{Derive}(pk, \tau, \{\boldsymbol{v}_i, \sigma_i, \beta_i\}_{i=1}^l)$.
  $Q := Q \cup \{(h, \tau, \sum_{i=1}^l \beta_i \cdot \boldsymbol{v}_i, \overline{\sigma})\}$. **Rtrn** h.
- $\mathfrak{Reveal}(h \in \mathcal{H}, \tau \in \{0,1\}^*, \boldsymbol{v} \in \mathbb{Z}_p^n)$:
  Rtrn $\perp$ if $\nexists\sigma$ s.t. $(h, \tau, \boldsymbol{v}, \sigma) \in Q$. $Q' := Q' \cup \{(\tau, \boldsymbol{v})\}$. **Rtrn** σ.

---

2. **Rtrn** 1 if (1) $1 \leftarrow \mathtt{Ver}(pk, \tau^*, \boldsymbol{v}^*, \sigma^*)$ and (2) one of the two conditions is satisfied.
   (a) $\tau^* \neq \tau_i$ for any entry $(\tau_i, \cdot) \in Q'$ and $\boldsymbol{v}^* \neq \boldsymbol{0}$.
   (b) $\tau^* = \tau_i$ for $k > 0$ entries $(\tau_i, \boldsymbol{v}_i)$ in Q' and $\boldsymbol{v}^* \notin \mathbf{span}\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k\}$.

Definition 10. *An LHS scheme Σ_{LHS} is UNF if for every $\lambda \in \mathbb{N}$, every $n \in \mathtt{poly}(\lambda)$ and every PPT $\mathcal{A}$, $\boldsymbol{Adv}^{\mathtt{UNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{A}}(\lambda) := \Pr[1 \leftarrow \boldsymbol{Expt}^{\mathtt{UNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{A}}(1^\lambda, n)]$ is negligible.*

SCH and CCH. Both of them are security notions guaranteeing that no signature generated by the deriving algorithm Derive based on some original signatures can be linked to the original ones. In the former, the original signatures have been honestly generated by the signing algorithm Sig. In the latter, the only condition that the original signatures must satisfy is that they are correct ones, which means that they might have been dishonestly generated. Obviously, the latter notion is truly stronger than the former.

Definition 11. *An LHS scheme is SCH if for every $\lambda \in \mathbb{N}$, every $n \in \mathtt{poly}(\lambda)$, every $(pk, sk) \leftarrow \mathtt{KGen}(1^\lambda, n)$, every tag $\tau \in \{0,1\}^*$, every integer $l \in [1,n]$, all l linearly-independent vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_l \in \mathbb{Z}_p^n$ and all l weights $\beta_1, \cdots, \beta_l \in \mathbb{Z}_p$, the following two distributions are statistically close, namely*

- $\{sk, \{\sigma_i\}_{i=1}^l, \mathtt{Derive}(pk, \tau, \{\boldsymbol{v}_i, \sigma_i, \beta_i\}_{i=1}^l)\}$ *and*
- $\{sk, \{\sigma_i\}_{i=1}^l, \mathtt{Sig}(sk, \tau, \sum_{i=1}^l \beta_i \boldsymbol{v}_i)\}$,

where $\sigma_i \leftarrow \mathtt{Sig}(sk, \tau, \boldsymbol{v}_i)$ for each $i \in [1,l]$.

Definition 12. *An LHS scheme is CCH if for every $\lambda \in \mathbb{N}$, every $n \in \mathtt{poly}(\lambda)$, every $(pk, sk) \leftarrow \mathtt{KGen}(1^\lambda, n)$, every tag $\tau \in \{0,1\}^*$, every integer $l \in [1,n]$, all l linearly-independent vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_l \in \mathbb{Z}_p^n$, all l correct signatures $\sigma_1, \cdots, \sigma_l$ s.t. $1 \leftarrow \mathtt{Ver}(pk, \tau, \boldsymbol{v}_i)$, and all l weights $\beta_1, \cdots, \beta_l \in \mathbb{Z}_p$, the following two distributions are statistically close, namely*

- $\{sk, \{\sigma_i\}_{i=1}^l, \mathtt{Derive}(pk, \tau, \{\boldsymbol{v}_i, \sigma_i, \beta_i\}_{i=1}^l)\}$ *and*
- $\{sk, \{\sigma_i\}_{i=1}^l, \mathtt{Sig}(sk, \tau, \sum_{i=1}^l \beta_i \boldsymbol{v}_i)\}$.

A.2 Unforgeability and Strong/Complete Context-Hiding of AOS

Unforgeability. We consider the following experiment.

$\boldsymbol{Expt}^{\mathit{UNF}}_{\Sigma_{\mathrm{AOS}},\mathcal{A}}(1^\lambda, H, L)$:

1. $(pk, sk) \leftarrow \mathtt{Setup}(1^\lambda, H, L)$. $(M^* \in (\{0,1\}^L)^{h^*}, \sigma^*) \leftarrow \mathcal{A}^{\mathfrak{Sign},\mathfrak{Derive},\mathfrak{Reveal}}(pk)$.

- $\mathfrak{Sign}(M \in (\{0,1\}^L)^h)$:
  Choose an unused handle $h \xleftarrow{\mathrm{U}} \mathcal{H}$. $\sigma \leftarrow \mathtt{Sig}(sk, M)$. $Q := Q \cup \{(h, M, \sigma)\}$. **Rtrn** h.
- $\mathfrak{Derive}(h \in \mathcal{H}, M \in (\{0,1\}^L)^h, M' \in (\{0,1\}^L)^{h'})$:
  Rtrn $\perp$ if $\nexists \sigma$ s.t. $(h, M, \sigma) \in Q$.
  Choose an unused handle $h' \xleftarrow{\mathrm{U}} \mathcal{H}$. $\sigma' \leftarrow \mathtt{Derive}(pk, M, \sigma, M')$.
  $Q := Q \cup \{(h, M', \sigma')\}$. **Rtrn** h'.
- $\mathfrak{Reveal}(h \in \mathcal{H}, M \in (\{0,1\}^L)^h)$:
  Rtrn $\perp$ if $\nexists \sigma$ s.t. $(h, M, \sigma) \in Q$. $Q' := Q' \cup \{M\}$. **Rtrn** σ.

---

2. **Rtrn** 1 if (1) $1 \leftarrow \mathtt{Ver}(pk, M^*, \sigma^*)$, and
   (2) $h > h^* \vee \exists i \in [1, h]$ *s.t.* $m_i \neq m_i^*$ *for any* $M \in Q'$, *where* $M \in (\{0,1\}^L)^h$ *for some* $h \leq H$.
3. **Rtrn** 0.

Definition 13. *An AOS scheme* Σ_{AOS} *is* ***UNF*** *if for every* $\lambda \in \mathbb{N}$, *every* $H, L \in \mathbb{N}$ *and every PPT* $\mathcal{A}$, $\boldsymbol{Adv}^{\mathit{UNF}}_{\Sigma_{\mathrm{AOS}},\mathcal{A}}(\lambda) := \Pr[1 \leftarrow \boldsymbol{Expt}^{\mathit{UNF}}_{\Sigma_{\mathrm{AOS}},\mathcal{A}}(1^\lambda, H, L)]$ *is negligible.*

SCH and CCH. Defined as follows.

Definition 14. *An AOS scheme is* ***SCH*** *if for every* $\lambda \in \mathbb{N}$, *every* $H, L \in \mathbb{N}$, *every* $(pk, sk) \leftarrow \mathtt{KGen}(1^\lambda, H, L)$, *every* $M = (m_1, \cdots, m_h) \in (\{0,1\}^L)^h$, *every* $M' = (m'_1, \cdots, m'_{h'}) \in (\{0,1\}^L)^{h'}$ *s.t.* $h' > h$ *and* $m'_i = m_i$ *for all* $i \in [1, h]$, *the following two distributions are statistically close, (1)* $\{sk, \sigma, \mathtt{Derive}(pk, M, \sigma, M')\}$ *and (2)* $\{sk, \sigma, \mathtt{Sig}(sk, M')\}$, *where* $\sigma \leftarrow \mathtt{Sig}(sk, M)$.

Definition 15. *An AOS scheme is* ***CCH*** *if for every* $\lambda \in \mathbb{N}$, *every* $H, L \in \mathbb{N}$, *every* $(pk, sk) \leftarrow \mathtt{KGen}(1^\lambda, H, L)$, *every* $M = (m_1, \cdots, m_h) \in (\{0,1\}^L)^h$, *every* σ *s.t.* $1 \leftarrow \mathtt{Ver}(pk, M, \sigma)$, *every* $M' = (m'_1, \cdots, m'_{h'}) \in (\{0,1\}^L)^{h'}$ *s.t.* $h' > h$ *and* $m'_i = m_i$ *for all* $i \in [1, h]$, *the following two distributions are statistically close, (1)* $\{sk, \sigma, \mathtt{Derive}(pk, M, \sigma, M')\}$ *and (2)* $\{sk, \sigma, \mathtt{Sig}(sk, M')\}$.

A.3 Attribute-Based Signatures (ABS) for a General Predicate and Its Subclasses

Syntax. General ABS for predicate $f : \{0,1\}^* \to \{0,1\}$ in $\mathcal{F}$ consists of the following four polynomial-time algorithms. `Ver` is deterministic and the others are probabilistic.

Setup `Setup`**:** It takes a security parameter 1^λ for $\lambda \in \mathbb{N}$, then outputs a public parameter pp and master-key mk. Let $\mathcal{M}$ denote the message space. Note that the other algorithms implicitly take pp as input. $(pp, mk) \leftarrow \mathtt{Setup}(1^\lambda)$

Key-Generation `KGen`**:** It takes mk and an attribute $x \in \{0,1\}^*$, then outputs a secret-key sk. $sk \leftarrow \mathtt{KGen}(mk, x)$

Signing `Sig`**:** It takes a secret-key sk, a message $M \in \mathcal{M}$ and a predicate $f \in \mathcal{F}$, then outputs a signature σ. $\sigma \leftarrow \mathtt{Sig}(sk, M, f)$

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

Every ABS scheme must be correct. Informally the property means that every correctly generated signature is accepted. Formally the property is defined as follows. An ABS scheme is correct if $\forall \lambda \in \mathbb{N}$, $\forall (pp, mk) \leftarrow \mathtt{Setup}(1^\lambda)$, $\forall x \in \{0,1\}^*$, $\forall sk \leftarrow \mathtt{KGen}(mk, x)$, $\forall M \in \mathcal{M}$, $\forall f \in \mathcal{F}$ s.t. $1 \leftarrow f(x)$, $\forall \sigma \leftarrow \mathtt{Sig}(sk, M, f)$, $1 \leftarrow \mathtt{Ver}(\sigma, M, f)$ holds.

As security for ABS, we require unforgeability and signer-privacy. As a notion of unforgeability, we define unforgeablity against adaptively chosen predicate attack ($\mathtt{UNF}$). For a PPT algorithm $\mathcal{A}$, we consider the following experiment.

$\boldsymbol{Expt}^{\mathtt{UNF}}_{\Sigma_{\mathrm{ABS}},\mathcal{A}}(1^\lambda)$:
1. $(pp, mk) \leftarrow \mathtt{Setup}(1^\lambda)$. $(\sigma^*, M^* \in \mathcal{M}, f^* \in \mathcal{F}) \leftarrow \mathcal{A}^{\mathfrak{Reveal},\mathfrak{Sign}}(pp)$.

- $\mathfrak{Reveal}(x \in \{0,1\}^*)$: $sk \leftarrow \mathtt{KGen}(mk, x)$. $Q := Q \cup \{x\}$. **Rtrn** sk.
- $\mathfrak{Sign}(x \in \{0,1\}^*, M \in \mathcal{M}, f \in \mathcal{F})$: $sk \leftarrow \mathtt{KGen}(mk, x)$. $\sigma \leftarrow \mathtt{Sig}(sk, M, f)$. $Q' := Q' \cup \{(M, f, \sigma)\}$. **Rtrn** σ.

2. **Rtrn** 1 if (1) $1 \leftarrow \mathtt{Ver}(\sigma^*, M^*, y^*)$, (2) $\forall x \in Q$, $0 \leftarrow f^*(x)$ and (3) $(M^*, f^*, \cdot) \notin Q'$. **Rtrn** 0.

Definition 16. *An ABS scheme* Σ_{ABS} *is* $\mathtt{UNF}$ *if for every* $\lambda \in \mathbb{N}$ *and every PPT* $\mathcal{A}$*,* $\mathcal{A}$*'s advantage* $\boldsymbol{Adv}^{\mathtt{UNF}}_{\Sigma_{\mathrm{ABS}},\mathcal{A}}(\lambda) := \Pr[1 \leftarrow \boldsymbol{Expt}^{\mathtt{UNF}}_{\Sigma_{\mathrm{ABS}},\mathcal{A}}(1^\lambda)]$ *is negligible.*

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

$\boldsymbol{Expt}^{\mathtt{PRV}}_{\Sigma_{\mathrm{ABS}},\mathcal{A},0}(1^\lambda)$: // $\boldsymbol{Expt}^{\mathtt{PRV}}_{\Sigma_{\mathrm{ABS}},\mathcal{A},1}$
$(pp, mk) \leftarrow \mathtt{Setup}(1^\lambda)$. $(pp, mk, \mu) \leftarrow \mathtt{SimSetup}(1^\lambda)$. **Rtrn** $b' \leftarrow \mathcal{A}^{\mathfrak{Reveal},\mathfrak{Sign}}(pp, mk)$.

- $\mathfrak{Reveal}(x \in \{0,1\}^*)$: $sk \leftarrow \mathtt{KGen}(mk, x)$. $sk \leftarrow \mathtt{SimKGen}(mk, \mu, x)$. $Q := Q \cup \{(x, sk)\}$. **Rtrn** sk.
- $\mathfrak{Sign}(x \in \{0,1\}^*, sk, M \in \mathcal{M}, f \in \mathcal{F})$: **Rtrn** $\perp$ if $(x, sk) \notin Q \vee 0 \leftarrow f(x)$. $\sigma \leftarrow \mathtt{Sig}(sk, M, f)$. $\sigma \leftarrow \mathtt{SimSig}(mk, \mu, M, f)$. **Rtrn** σ.

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

Definition 17. *An ABS scheme* Σ_{ABS} *is perfectly signer-private (*$\mathtt{PRV}$*) if for every* $\lambda \in \mathbb{N}$ *and every probabilistic algorithm* $\mathcal{A}$*, there exist polynomial-time algorithms* $\{\mathtt{SimSetup}, \mathtt{SimKGen}, \mathtt{SimSig}\}$ *such that* $\mathcal{A}$*'s advantage* $\boldsymbol{Adv}^{\mathtt{PRV}}_{\Sigma_{\mathrm{ABS}},\mathcal{A}}(\lambda) := |\sum_{b=0}^{1}(-1)^b \Pr[1 \leftarrow \boldsymbol{Expt}^{\mathtt{PRV}}_{\Sigma_{\mathrm{ABS}},\mathcal{A},b}(1^\lambda)]|$ *becomes* 0*.*

ABS has various subclasses. Some examples are given below.

1. ABS for Range Evaluation of Polynomials (AREP) [9]: The attribute $x \in \{0,1\}^*$ in the general ABS is changed into a single variable $x \in \mathbb{Z}_p$ in AREP. The predicate f_{AREP}, associated with a d-dimensional *univariate* polynomial ϕ with coefficients $a_d, \cdots, a_0 \in \mathbb{Z}_p$ and a range $[L, R]$ with $L, R \in \mathbb{Z}_p$, is defined as

$$f_{\mathrm{AREP}}(x) := \begin{cases} 1 & (\text{If } \phi(x) := \sum_{i=0}^{d} a_i \cdot x^i \in [L, R] \pmod p) \\ 0 & (\text{Otherwise}). \end{cases}$$

2. **ABS for Range Evaluation of Weighted Average (AREWA) [9]:** The attribute x consists of t variables $x_1, \cdots, x_t \in \mathbb{Z}_p$. The predicate f_{AREWA}, associated with t coefficients $a_1, \cdots, a_t \in \mathbb{Z}_p$ and a range $[L, R]$ for $L, R \in \mathbb{Z}_p$, is defined as

$$f_{\text{AREWA}}(x_1, \cdots, x_t) := \begin{cases} 1 & (\text{If } \sum_{i=1}^{t} a_i \cdot x_i \in [L, R] \pmod p) \\ 0 & (\text{Otherwise}). \end{cases}$$

3. **Fuzzy IBS (FIBS):** This is a generalization of the ABS for exact thresholds. Let A be $\{1, \cdots, l\}$ for $l \in \mathbb{N}$. The attribute x is a set of attributes $S \subseteq A$. The predicate f_{FIBS}, associated with a set of attributes $S' \subseteq A$ and a range $[L, R]$ for $0 \leq L \leq R \leq l$, is defined as

$$f_{\text{FIBS}}(S) := \begin{cases} 1 & (\text{If } |S \cap S'| \in [L, R]) \\ 0 & (\text{Otherwise}). \end{cases}$$

This FIBS is a further generalization of the signature analogue of FIBE [14] since the upper bound R of the overlapped attributes can be set.

4. **Time-Specific Signatures (TSS) [13,10]:** TSS is a subclass of the ABS. The attribute $x \in \{0,1\}^*$ is a time-period $t \in [0, T-1]$ for an integer $T \in \mathbb{N}$. The predicate f_{TSS}, associated with a range $[L, R]$ with $L, R \in [0, T-1]$, is defined as

$$f_{\text{TSS}}(t) := \begin{cases} 1 & (\text{If } t \in [L, R] \\ 0 & (\text{Otherwise}). \end{cases}$$

5. **ABS for Range Evaluation of Hamming Distance (AREHD) [9]:** A signer with a (binary) string $x \in \{0,1\}^l$ can sign a message under a string $y \in \{0,1\}^l$ iff the Hamming distance between x and y is within a range $[L, R]$. The attribute x in the ABS is a string $x \in \{0,1\}^l$. The predicate f_{AREHD} is defined as

$$f_{\text{AREHD}}(x) := \begin{cases} 1 & (\text{If } \mathbf{HD}(x, y) \in [L, R]) \\ 0 & (\text{Otherwise}), \end{cases}$$

where the function $\mathbf{HD}(x, y)$ returns $\sum_{i=0}^{l-1} |x[i] - y[i]|$ which is the Hamming distance between x and y.

6. **ABS for Range Evaluation of Euclidean Distance (AREED) [9]:** A signer with a vector $\vec{X} \in \mathbb{Z}_p^n$ declares another vector $\vec{Y} \in \mathbb{Z}_p^n$ and a range $[L, R]$. If the Euclidean distance between the two vectors is within the range, the signing succeeds. The predicate f_{AREED} is defined as

$$f_{\text{AREED}}(\vec{X}) := \begin{cases} 1 & (\text{If } \mathbf{ED}(\vec{X}, \vec{Y}) \in [L, R]) \\ 0 & (\text{Otherwise}), \end{cases}$$

where the function $\mathbf{ED}(\vec{X}, \vec{Y})$ returns $\sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2} \in [L, R]$ which is the Euclidean distance between $\vec{X}$ and $\vec{Y}$.

7. ABS for Hyperellipsoid Predicates (AHEP) [9]: An n-dimensional hypersphere is a set of points (or vectors) whose Euclidean distance to the central point is constant. Let us consider a special type of ABS, where a secret-key is associated with a vector $\vec{X} \in \mathbb{Z}_p^n$, a signature is associated with a hypersphere with center $\vec{Y} \in \mathbb{Z}_p^n$ and radius $a \in \mathbb{Z}_p$ and the signing succeeds iff the vector $\vec{X}$ is inside of the hypersphere, named ABS for hypersphere predicates (AHSP). Obviously, AHSP is transformed from AREED defined above.

AHEP is a generalization of AHSP. Each hypersphere is generalized to a hyperellipsoid. The predicate f_{AHEP} is defined as

$$f_{\mathrm{AHEP}}(\vec{X}) := \begin{cases} 1 & (\text{If } \sum_{i=1}^{n}(X_i - Y_i)^2/a_i^2 \leq 1), \\ 0 & (\text{Otherwise}), \end{cases}$$

where $\vec{Y} \in \mathbb{Z}_p^n$ is the center and $a_i \in \mathbb{Z}_p$ is the radius in the i-th axis.

Their key-range versions are also a subclass of the general ABS. For instance, in key-range ABS for range evaluation of polynomials (KAREP), not only an attribute $x \in \mathbb{Z}_p$ but also a range $[L, R] \subseteq \mathbb{Z}_p$ is associated with a secret-key, and only a polynomial $\phi : \mathbb{Z}_p \to \mathbb{Z}_p$ is associated with a signature. The other key-range versions are denoted by KAREWA, KFIBS, KTSS, KAREHD, KAREED and KAHEP, respectively.

A.4 Formal Definition of the Covering Algorithm `Cover`

Assume that L and R are of bit length λ and $L \leq R$. An integer $a \in \mathbb{Z}_p$ with bit length λ is parsed as $a[1] \| \cdots \| a[\lambda]$ with $a[i] \in \{0, 1\}$. The algorithm `Cover` is defined as follows.

`Cover`(L, R): Let $l := L$. A set C is initially empty, i.e., $C := \emptyset$. While $l \leq R$, repeat the following steps.

- Derive the minimal integer $t \in [1, \lambda]$ satisfying both of the following conditions,
  1. $l[t] = \cdots = l[\lambda] = 0$
  2. $[\underbrace{l[1] \| \cdots \| l[t-1]}_{t-1} \| \underbrace{1 \cdots}_{\lambda+1-t}]_{10} \leq R$

  For a binary value a, $[a]_{10}$ means its decimal value. If such an integer t does not exist, $t := \lambda + 1$. Obviously, the node associated with $l[1] \| \cdots \| l[t-1] \in \{0,1\}^{t-1}$ covers all of the leaf nodes associated with from $[l]_2$ to $[l + 2^{\lambda+1-t} - 1]_2$. Let $C := C \cup \{l[1] \| \cdots \| l[t-1]\}$ and $l := l + 2^{\lambda+1-t}$.

Return C.

For instance, in a complete binary tree with 8 leaf nodes depicted in Fig. 1, `Cover`$(1, 6) = \{001, 01, 10, 110\}$, `Cover`$(0, 4) = \{0, 100\}$, `Cover`$(7, 7) = \{111\}$, and `Cover`$(0, 7) = \emptyset$.

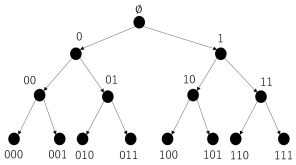

Fig. 1. A complete binary tree with depth 3

B Omitted Proofs

B.1 Proof of Theorem 5 (on the Security of Our 2nd Generic KARIP Construction)

We define five experiments as follows.

***Expt*$_0$:** The standard UNF experiment w.r.t. the KARIP scheme.
***Expt*$_1$:** It aborts when we choose a tag on the key-revelation or signing oracle, the tag matches a tag previously chosen.
***Expt*$_2$:** We directly generate an LHS signature σ' on $\boldsymbol{v}' := (\langle \mathbf{x}, \mathbf{y} \rangle, y_1, \cdots, y_n, L, R, M, 1)$ by using the LHS secret-key sk_{L}.
***Expt*$_3$:** The CRS crs is generated by $(crs, ek) \leftarrow \mathtt{SimSetup}(1^\lambda)$.
***Expt*$_4$:** We extract $w^* = (L^*, R^*, d^*, \tau^*, \sigma^*, \theta^*) \leftarrow \mathtt{Extract}(crs, ek, x^*, \sigma^*)$, where $x^* := (\mathbf{y}^*, M^*)$. It aborts if $0 \leftarrow \mathcal{R}_{\mathsf{N}}(x^*, w^*)$.

We obtain $\mathtt{Adv}^{\mathtt{UNF}}_{\Sigma_{\mathrm{KARIP}},\mathcal{A},n}(\lambda) = \Pr[W_0] \leq \sum_{i=1}^{4} |\Pr[W_{i-1}] - \Pr[W_i]| + \Pr[W_4] \leq q(q-1)/2^{N+1} + \mathtt{Adv}^{\mathtt{WI}}_{\Sigma_{\mathrm{NIWI}},\mathcal{B}_3}(\lambda) + \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{B}_5}(\lambda)$ for some PPT adversary $\mathcal{B}_3, \mathcal{B}_5$. The last inequality is obtained because of the following five lemmas. The first four lemmas can be proven in the same manner as the counterpart lemmas for our 1st construction, i.e., Lemmas 1-4. □

Lemma 7. $\Pr[W_0] - \Pr[W_1] \leq q(q-1)/2^{N+1}$, *where* $q \in \mathtt{poly}(\lambda)$ *is the total number of times that* $\mathcal{A}$ *uses the key-revelation and signing oracles.*

Lemma 8. $|\Pr[W_1] - \Pr[W_2]| = 0$ *if the NIWI system is* WI.

Lemma 9. $\Pr[W_2] - \Pr[W_3]$ *is negligible if the NIWI system is* WE. *Formally, there exists a PPT algorithm* $\mathcal{B}_3$ *s.t.* $\Pr[W_2] - \Pr[W_3] \leq \mathit{Adv}^{\mathit{WE}}_{\Sigma_{\mathrm{NIWI}},\mathcal{B}_3}(\lambda)$.

Lemma 10. $\Pr[W_3] - \Pr[W_4] = 0$ *if the NIWI system is* WE.

Lemma 11. $\Pr[W_4]$ *is negligible if the LHS scheme is* wUNF. *Formally, there exists a PPT algorithm* $\mathcal{B}_5$ *s.t.* $\Pr[W_4] \leq \mathit{Adv}^{\mathit{wUNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{B}_5}(\lambda)$.

Proof. Assume that $\mathcal{A}$ is a PPT algorithm which makes the event W_4 occur with a non-negligible probability. By using $\mathcal{A}$, a PPT simulator $\mathcal{B}_5$ attempts to win the wUNF experiment w.r.t. the LHS scheme.

$\mathcal{B}_5$ receives an LHS public-key pk_{L}. $\mathcal{B}_5$ can access to the signing oracle $\mathfrak{Sign}_{\mathsf{L}}$. $\mathcal{B}_5$ honestly generates crs and ek. $\mathcal{B}_5$ sends $pp := (crs, pk_{\mathsf{L}})$ to $\mathcal{A}$ and run it.

$\mathfrak{Reveal}(\mathbf{x}, L, R)$**:** Choose a tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Honestly generate the $n+2$ vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{n+2} \in \mathbb{Z}_p^{n+3}$. For each vector $\boldsymbol{v}_i$, generate an LHS signature by $\sigma_i \leftarrow \mathfrak{Sign}_{\mathsf{L}}(\tau, \boldsymbol{v}_i)$. Return $sk := (\tau, \{\sigma_i\}_{i=1}^{n+2})$.

$\mathfrak{Sign}(\mathbf{x}, L, R, \mathbf{y}, M)$**:** Choose $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Generate an LHS signature on a vector $\boldsymbol{v}' := (d, y_1, \cdots, y_n, L, R, M, 1)$ by $\sigma' \leftarrow \mathfrak{Sign}_{\mathsf{L}}(\tau, \boldsymbol{v}')$. Generate an NIWI proof $\pi \leftarrow \mathsf{N.Pro}(crs, x, w)$, where $x := (\mathbf{y}, M)$ and $w := (L, R, d, \tau, \sigma')$, then return it.

Given a forged KARIP signature π^*, $\mathcal{B}_5$ extracts the witness behind the NIWI proof π^* by $w^* \leftarrow \mathtt{Extract}(crs, ek, x^*, \pi)$, where $x^* := (\mathbf{y}^*, M^*)$, and parse it as $(L^*, R^*, d^*, \tau^*, \sigma^*)$. $\mathcal{B}_5$ outputs a forged LHS signature σ^* with tag τ^* on vector $\boldsymbol{v}^* := (d^*, y_1^*, \cdots, y_n^*, L^*, R^*, M^*, 1)$.

Because of the event W_4, one of the following three events must occur.

E1: τ^* has not been previously chosen.
E2: τ^* has been already chosen on the signing oracle.
E3: τ^* has been already chosen on the key-revelation oracle.

Any of the events leads $\mathcal{B}_5$ to win the $\mathtt{wUNF}$ experiment.

E1: Every tag queried to $\mathfrak{Sign}_{\mathsf{L}}$ is not identical to τ^*. W_4 implies $\neg\mathsf{abort}_4$, which implies that σ^* is a valid LHS signature on the non-zero vector $\boldsymbol{v}^*$.
E2: W_4 implies $\neg\mathsf{abort}_1$, which implies that τ^* is identical to a single tag chosen on the signing oracle. Among multiple vectors whom $\mathcal{B}_5$ queried to $\mathfrak{Sign}_{\mathsf{L}}$, $\hat{\boldsymbol{v}} := (\langle \hat{\mathbf{x}}, \hat{\mathbf{y}} \rangle, \hat{y}_1, \cdots, \hat{y}_n, \hat{L}, \hat{R}, \hat{M}, 1)$ is the only vector tagged by τ^*, where $\hat{\mathbf{x}}$, $\hat{L}, \hat{R}, \hat{\mathbf{y}}$ and $\hat{M}$ denote variables queried to the signing oracle when the tag τ^* was chosen. W_4 implies that $(\mathbf{y}^*, M^*) \neq (\hat{\mathbf{y}}, \hat{M})$. Obviously, $\boldsymbol{v}^*$ is linearly independent of $\hat{\boldsymbol{v}}$.
E3: W_4 implies $\neg\mathsf{abort}_1$, which implies that the extracted tag τ^* is identical to a single tag chosen on the key-revelation oracle. W_4 implies $1 \leftarrow \mathcal{R}_{\mathsf{N}}(x^*, w^*)$ implying $d^* \in [L^*, R^*]$. W_4 implies $\langle \hat{\mathbf{x}}, \mathbf{y}^* \rangle \notin [L^*, R^*]$. Hence, $d^* \neq \langle \hat{\mathbf{x}}, \mathbf{y}^* \rangle$. Among multiple vectors whom $\mathcal{B}_5$ queried to $\mathfrak{Sign}_{\mathsf{L}}$, there are $n+2$ vectors $\hat{\boldsymbol{v}}_1, \cdots, \hat{\boldsymbol{v}}_{n+2}$ tagged by τ^*. The vectors are expressed as follows. For each $i \in [1, n]$, $\hat{\boldsymbol{v}}_i = (\hat{x}_i, \underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{n-i}, \underbrace{0, 0, 0, 0}_{4})$. The others are $\hat{\boldsymbol{v}}_{n+1} = (0, \cdots, 0, \hat{L}, \hat{R}, 0, 1)$ and $\hat{\boldsymbol{v}}_{n+2} = (0, \cdots, 0, 0, 0, 1, 0)$. Since $d^* \neq \langle \hat{\mathbf{x}}, \mathbf{y}^* \rangle \pmod p$, $\boldsymbol{v}^*$ cannot be a linear combination of $\hat{\boldsymbol{v}}_1$ and $\hat{\boldsymbol{v}}_2$.

Therefore, $\Pr[W_4] \leq \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}}, \mathcal{B}_5}(\lambda)$. □

B.2 Proof of Theorem 6 (on the Security of Our 3rd Generic KARIP Construction)

We define six experiments as follows.

Expt$_0$**:** The standard $\mathtt{UNF}$ experiment w.r.t. the KARIP scheme.

***Expt*$_1$:** It aborts when we choose a tag on the key-revelation or signing oracle, the tag matches a tag previously chosen.

***Expt*$_2$:** We directly generate an LHS signature σ' on $\boldsymbol{v}' := (x_1, \cdots, x_n, L, R, h, 1)$ by using the LHS secret-key sk_{L}.

***Expt*$_3$:** The CRS crs is generated by $(crs, ek) \leftarrow \mathtt{SimSetup}(1^\lambda)$.

***Expt*$_4$:** We extract the witness $w^* = (L^*, R^*, d^*, \tau^*, \sigma^*, \theta^*) \leftarrow \mathtt{Extract}(crs, ek, x^*, \sigma^*)$, where $x^* := (\mathbf{y}^*, M^*)$. It aborts if $0 \leftarrow \mathcal{R}_{\mathsf{N}}(x^*, w^*)$.

***Expt*$_5$:** It aborts if there exists a query of $(\hat{\mathbf{y}}, \hat{M})$ to the signing oracle satisfying $\mathsf{H}.\mathtt{Eval}(hk, (\hat{\mathbf{y}}, \hat{M})) = \mathsf{H}.\mathtt{Eval}(hk, (\mathbf{y}^*, M^*))$.

We obtain $\mathtt{Adv}^{\mathtt{UNF}}_{\Sigma_{\mathrm{KARIP}},\mathcal{A},n}(\lambda) = \Pr[W_0] \leq \sum_{i=1}^{5} |\Pr[W_{i-1}] - \Pr[W_i]| + \Pr[W_5] \leq q(q-1)/2^{N+1} + \mathtt{Adv}^{\mathtt{WI}}_{\Sigma_{\mathrm{NIWI}},\mathcal{B}_3}(\lambda) + \mathtt{Adv}^{\mathtt{CR}}_{\Sigma_{\mathrm{HF}},\mathcal{B}_5}(\lambda) + \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{B}_6}(\lambda)$ for some PPT adversary $\mathcal{B}_3, \mathcal{B}_5, \mathcal{B}_6$. The last inequality is obtained because of the following six lemmas. The first four lemmas can be proven in the same manner as the counterpart lemmas for our 1st construction, i.e., Lemmas 1-4. □

Lemma 12. $\Pr[W_0] - \Pr[W_1] \leq q(q-1)/2^{N+1}$, *where* $q \in \mathtt{poly}(\lambda)$ *is the total number of times that* $\mathcal{A}$ *uses the key-revelation and signing oracles.*

Lemma 13. $|\Pr[W_1] - \Pr[W_2]| = 0$ *if the NIWI system is* ***WI****.*

Lemma 14. $\Pr[W_2] - \Pr[W_3]$ *is negligible if the NIWI system is* ***WE****. Formally, there exists a PPT algorithm* $\mathcal{B}_3$ *s.t.* $\Pr[W_2] - \Pr[W_3] \leq \mathtt{Adv}^{\mathtt{WE}}_{\Sigma_{\mathrm{NIWI}},\mathcal{B}_3}(\lambda)$.

Lemma 15. $\Pr[W_3] - \Pr[W_4] = 0$ *if the NIWI system is* ***WE****.*

Lemma 16. $\Pr[W_4] - \Pr[W_5]$ *is negligible if the hash function is* ***CR****. Formally, there exists a PPT algorithm* $\mathcal{B}_5$ *s.t.* $\Pr[W_4] - \Pr[W_5] \leq \mathtt{Adv}^{\mathtt{CR}}_{\Sigma_{\mathrm{HF}},\mathcal{B}_5}(\lambda)$.

Proof. As the proof of Lemma 4, $\Pr[W_4] - \Pr[W_5] = \Pr[W_4 \wedge \mathsf{abort}_5]$ holds. W_4 implies $(\hat{\mathbf{y}}, \hat{M}) \neq (\mathbf{y}^*, M^*)$. abort_5 implies $\mathsf{H}.\mathtt{Eval}(hk, (\hat{\mathbf{y}}, \hat{M})) = \mathsf{H}.\mathtt{Eval}(hk, (\mathbf{y}^*, M^*))$. We can easily construct a PPT algorithm $\mathcal{B}_5$ s.t. $\Pr[W_4 \wedge \mathsf{abort}_5] \leq \mathtt{Adv}^{\mathtt{CR}}_{\Sigma_{\mathrm{HF}},\mathcal{B}_5}(\lambda)$. □

Lemma 17. $\Pr[W_5]$ *is negligible if the LHS scheme is* ***wUNF****. Formally, there exists a PPT algorithm* $\mathcal{B}_5$ *s.t.* $\Pr[W_5] \leq \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}},\mathcal{B}_6}(\lambda)$.

Proof. Assume that $\mathcal{A}$ is a PPT algorithm which makes the event W_5 occur with a non-negligible probability. By using $\mathcal{A}$, a PPT simulator $\mathcal{B}_6$ attempts to win the $\mathtt{wUNF}$ experiment w.r.t. the LHS scheme.

$\mathcal{B}_6$ receives an LHS public-key pk_{L}. $\mathcal{B}_6$ can access to the signing oracle $\mathfrak{Sign}_{\mathsf{L}}$. $\mathcal{B}_6$ honestly generates crs, ek and hk. $\mathcal{B}_6$ sends $pp := (crs, pk_{\mathsf{L}}, hk)$ to $\mathcal{A}$ and run it.

$\mathfrak{Reveal}(\mathbf{x}, L, R)$**:** Choose a tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Honestly generate the two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{Z}_p^{n+4}$. For each vector $\boldsymbol{v}_i$, generate an LHS signature by $\sigma_i \leftarrow \mathfrak{Sign}_{\mathsf{L}}(\tau, \boldsymbol{v}_i)$. Return $sk := (\tau, \sigma_1, \sigma_2)$.

$\mathfrak{Sign}(\mathbf{x}, L, R, \mathbf{y}, M)$**:** Choose $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Let $h \leftarrow \mathsf{H}.\mathtt{Eval}(hk, (\mathbf{y}, M))$. Generate an LHS signature on a vector $\boldsymbol{v}' := (x_1, \cdots, x_n, L, R, h, 1)$ by $\sigma' \leftarrow \mathfrak{Sign}_{\mathsf{L}}(\tau, \boldsymbol{v}')$. Generate an NIWI proof $\pi \leftarrow \mathsf{N}.\mathtt{Pro}(crs, x, w)$, where $x := (\mathbf{y}, M)$ and $w := (\mathbf{x}, L, R, d, \tau, \sigma')$, then return it.

Given a forged KARIP signature π^*, $\mathcal{B}_6$ extracts the witness behind the NIWI proof π^* by $w^* = (\mathbf{x}^*, L^*, R^*, d^*, \tau^*, \sigma^*) \leftarrow \mathtt{Extract}(crs, ek, x^*, \pi)$, where $x^* := (\mathbf{y}^*, M^*)$. Let $h^* \leftarrow \mathsf{H}.\mathtt{Eval}(hk, (\mathbf{y}^*, M^*))$. $\mathcal{B}_6$ outputs a forged LHS signature σ^* with tag τ^* on vector $\boldsymbol{v}^* := (x_1^*, \cdots, x_n^*, L^*, R^*, h^*, 1)$.

Because of the event W_5, one of the three events **E1**, **E2** and **E3** (defined in the proof of Lemma 11 in Subsect. B.1) must occur. Any of the events leads $\mathcal{B}_6$ to win the $\mathtt{wUNF}$ experiment.

E1: Every tag queried to $\mathfrak{Sign}_{\mathsf{L}}$ is not identical to τ^*. W_5 implies $\neg\mathsf{abort}_4$, which implies that σ^* is a valid LHS signature on the non-zero vector $\boldsymbol{v}^*$.

E2: W_5 implies $\neg\mathsf{abort}_1$, which implies that τ^* is identical to a single tag chosen on the signing oracle. Among multiple vectors whom $\mathcal{B}_6$ queried to $\mathfrak{Sign}_{\mathsf{L}}$, $\hat{\boldsymbol{v}} := (\hat{x}_1, \cdots, \hat{x}_n, \hat{L}, \hat{R}, \hat{h}, 1)$ is the only vector tagged by τ^*, where $\hat{\mathbf{x}}, \hat{L}, \hat{R}, \hat{\mathbf{y}}$ and $\hat{M}$ denote variables queried to the signing oracle when the tag τ^* was chosen and $\hat{h} \leftarrow \mathsf{H}.\mathtt{Eval}(hk, (\hat{\mathbf{y}}, \hat{M}))$. W_5 implies that $h^* \neq \hat{h}$. Hence, $\boldsymbol{v}^*$ is linearly independent of $\hat{\boldsymbol{v}}$.

E3: W_5 implies $\neg\mathsf{abort}_1$, which implies that τ^* is identical to a single tag chosen on the key-revelation oracle. W_5 implies $1 \leftarrow \mathcal{R}_{\mathsf{N}}(x^*, w^*)$ implying $\langle \mathbf{x}^*, \mathbf{y}^* \rangle \in [L^*, R^*]$. W_5 implies $\langle \hat{\mathbf{x}}, \mathbf{y}^* \rangle \notin [L^*, R^*]$. Hence, $\mathbf{x}^* \neq \hat{\mathbf{x}}$. Among multiple vectors whom $\mathcal{B}_6$ queried to $\mathfrak{Sign}_{\mathsf{L}}$, there are only 2 vectors $\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2$ tagged by τ^*. They are $\hat{\boldsymbol{v}}_1 = (\hat{x}_1, \cdots, \hat{x}_n, \hat{L}, \hat{R}, 0, 1)$ and $\hat{\boldsymbol{v}}_{n+2} = (0, \cdots, 0, 0, 0, 1, 0)$. Since $\mathbf{x}^* \neq \hat{\mathbf{x}}$, $\boldsymbol{v}^*$ is not in $\mathbf{span}(\{\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2\})$.

Therefore, $\Pr[W_5] \leq \mathtt{Adv}^{\mathtt{wUNF}}_{\Sigma_{\mathrm{LHS}}, \mathcal{B}_6}(\lambda)$. □

B.3 Proof of Theorem 4

The following proof is cited from [7].

$F_i(m)$ is maximized when $s_i = L$, $x_i = 0$ and $y_j = 0$ for all $j \in [1, L]$ s.t. $m[j] = 1$. The maximal value is $p - k \cdot L > 0$ because $k(L+1) < p$.

$F_i(m)$ is minimized when $s_i = 0$, $x_i = k - 1$ and $y_j = k - 1$ for all $j \in [1, L]$ s.t. $m[j] = 1$. The minimal value is $p + (k-1) + \sum_{j \in [1,L]} (k-1) \cdot m[j] \leq p + (k-1)(L+1) < p + k(L+1) < 2p$.

We obtain

$$F_i(m) = 0 \pmod p \iff F_i(m) = p \iff k \cdot s_i = x_i + \sum_{j=1}^{L} y_j \cdot m[j].$$

Hence, we obtain

$$F_i(m) = 0 \pmod p \implies L_i(m) = k \cdot s_i \implies L_i(m) = 0 \pmod k.$$

□

B.4 Evaluation of the Probability $\Pr[\neg\mathsf{abort}]$ in the Proof of Theorem 3

S denotes the event that $\mathcal{B}$ aborts the simulation on the signing oracle. F denotes the event that $\mathcal{B}$ aborts the simulation in the forgery phase. Because of the definitions of the events abort, S and F,

$$\Pr[\neg\mathsf{abort}] = \Pr[\neg F] \cdot \Pr[\neg S \mid \neg F]. \tag{5}$$

We derive the lower bound of the second term in (5). We obtain

$$\Pr[\neg S \mid \neg F] = 1 - \Pr[S \mid \neg F] = 1 - \Pr\left[\bigvee_{i=1}^{q} S_i \mid \neg F\right] \geq 1 - \sum_{i=1}^{q} \Pr\left[S_i \mid \neg F\right],$$

where S_i denotes the event that $\mathcal{B}$ aborts the simulation on the i-th signing oracle query. Let $M_i = (m_{i,1}, \cdots, m_{i,h_i}) \in (\{0,1\}^L)^{h_i}$ denote the message queried as the i-th signing oracle query. We analyze the probability $\Pr[S_i \mid \neg F]$ as

$$\begin{aligned}
\Pr[S_i \mid \neg F] &= \Pr\left[\bigwedge_{j=1}^{h_i} L_j(m_{i,j}) = 0 \pmod k \,\middle|\, \bigwedge_{j=1}^{h^*} F_j(m_j^*) = 0 \pmod p\right] \\
&\leq \Pr\left[L_{\hat{j}}(m_{i,\hat{j}}) = 0 \pmod k \,\middle|\, \bigwedge_{j=1}^{h^*} F_j(m_j^*) = 0 \pmod p\right] = \frac{1}{k},
\end{aligned}$$

where $\hat{j}$ denotes the smallest integer $j \in [1, h_i]$ satisfying $m_{i,j} \neq m_j^*$. Thus, $\Pr[\neg S \mid \neg F] \geq 1 - q/k$.

Next, we derive the lower bound of the first term in (5) as follows.

$$\begin{aligned}
\Pr[\neg F] &= \Pr\left[\bigwedge_{i=1}^{h^*} F_i(m_i^*) = 0 \pmod p\right] \\
&= \Pr\left[\bigwedge_{i=1}^{h^*} x_i + \sum_{j=1}^{L} y_j \cdot m_i^*[j] = k \cdot s_i\right] \\
&= \Pr\left[\bigwedge_{i=1}^{h^*} \bigvee_{s_i' \in [0,L]} \left\{x_i + \sum_{j=1}^{L} y_j \cdot m_i^*[j] = k \cdot s_i' \wedge s_i = s_i'\right\}\right] \\
&= \Pr\left[\bigwedge_{i=1}^{h^*} \bigvee_{s_i' \in [0,L]} \left\{X_{i,s_i'} \wedge \tilde{X}_{i,s_i'}\right\}\right] \\
&= \Pr\left[\bigvee_{s_1', \cdots, s_{h^*} \in [0,L]} \bigwedge_{i=1}^{h^*} \left\{X_{i,s_i'} \wedge \tilde{X}_{i,s_i'}\right\}\right] \\
&= \sum_{s_1', \cdots, s_{h^*} \in [0,L]} \Pr\left[\bigwedge_{i=1}^{h^*} \left\{X_{i,s_i'} \wedge \tilde{X}_{i,s_i'}\right\}\right]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{s'_1,\cdots,s_{h^*}\in[0,L]} \Pr\left[\bigwedge_{i=1}^{h^*} X_{i,s'_i}\right] \cdot \Pr\left[\bigwedge_{i=1}^{h^*} \tilde{X}_{i,s'_i}\right] \\
&= \frac{1}{(L+1)^{h^*}} \sum_{s'_1,\cdots,s_{h^*}\in[0,L]} \Pr\left[\bigwedge_{i=1}^{h^*} X_{i,s'_i}\right] \\
&= \frac{1}{(L+1)^{h^*}} \Pr\left[\bigvee_{s'_1,\cdots,s_{h^*}\in[0,L]} \bigwedge_{i=1}^{h^*} X_{i,s'_i}\right] \\
&= \frac{1}{(L+1)^{h^*}} \Pr\left[\bigwedge_{i=1}^{h^*} \bigvee_{s'_i\in[0,L]} X_{i,s'_i}\right] \\
&= \frac{1}{(L+1)^{h^*}} \Pr\left[\bigwedge_{i=1}^{h^*} L_i(m_i^*) = 0 \pmod{k}\right] = \frac{1}{\{k(L+1)\}^{h^*}},
\end{aligned}
$$

where X_{i,s'_i} (resp. $\tilde{X}_{i,s'_i}$) denote the event that it holds $x_i + \sum_{j=1}^{L} y_j \cdot m_i^*[j] = k \cdot s'_i$ (resp. $s_i = s'_i$).

Therefore, we obtain

$$
\Pr[\neg\mathsf{abort}] \geq \left(1 - \frac{q}{k}\right) \frac{1}{\{k(L+1)\}^H} = \frac{1}{2\{2q(L+1)\}^H}
$$

because we have assumed that $k = 2q$.

C Omitted Schemes

C.1 A Simplified Variant [9] of the ALP LHS Scheme [4]

$\mathtt{KGen}(1^\lambda, n)$**:** Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ whose order is a prime p. Choose $\alpha \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Let $g, h, g_1, \cdots, g_n \xleftarrow{\mathrm{U}} \mathbb{G}$. Let $u', u_1, \cdots, u_N \xleftarrow{\mathrm{U}} \mathbb{G}$ for an integer $N \in \mathbb{N}$. Let $H_{\mathbb{G}}$ be a function which takes $\tau \in \{0,1\}^N$ as input, then outputs $u' \prod_{i=1}^{N} u_i^{\tau[i]} \in \mathbb{G}$. Output (pk, sk), where $pk := (\mathbb{G}, \mathbb{G}_T, g, g^\alpha, h, \{g_i\}_{i=1}^n, u', \{u_i\}_{i=1}^N)$ and $sk := \alpha$.

$\mathtt{Sig}(sk, \tau \in \{0,1\}^N, \boldsymbol{v} \in \mathbb{Z}_p^n)$**:** Parse $\boldsymbol{v}$ as $(v_1, \cdots, v_n)$. Choose $r, s \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Compute

$$
(\sigma_1, \sigma_2, \sigma_3, \sigma_4) := \left((\prod_{j=1}^{n} g_i^{v_i} h^s)^\alpha H_{\mathbb{G}}(\tau)^r, g^r, g^s, g^{\alpha \cdot s} \right).
$$

Output $\sigma := (\boldsymbol{v}, \tau, \sigma_1, \sigma_2, \sigma_3, \sigma_4)$.

$\mathtt{Derive}(pk, \tau \in \{0,1\}^N, \{\boldsymbol{v}_i \in \mathbb{Z}_p^n, \sigma_i, \beta_i \in \mathbb{Z}_p\})$**:** Parse σ_i as $(\boldsymbol{v}, \tau, \sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3}, \sigma_{i,4})$. Choose $\overline{r} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Compute

$$
(\overline{\sigma}_1, \overline{\sigma}_2, \overline{\sigma}_3, \overline{\sigma}_4) := \left(\prod_{i=1}^{l} \sigma_{i,1}^{\beta_i} \cdot H_{\mathbb{G}}(\tau)^{\overline{r}}, \prod_{i=1}^{l} \sigma_{i,2}^{\beta_i} \cdot g^{\overline{r}}, \prod_{i=1}^{l} \sigma_{i,3}^{\beta_i}, \prod_{i=1}^{l} \sigma_{i,4}^{\beta_i} \right).
$$

Output $\overline{\sigma} := (\sum_{i=1}^{l} \beta_i \cdot \boldsymbol{v}_i, \tau, \overline{\sigma}_1, \overline{\sigma}_2, \overline{\sigma}_3, \overline{\sigma}_4)$.

$\mathtt{Ver}(pk, \tau \in \{0,1\}^N, \boldsymbol{v} \in \mathbb{Z}_p^n, \sigma)$**:** Parse $\boldsymbol{v} \in \mathbb{Z}_p^n$ as $(v_1, \cdots, v_n)$. Parse σ as $(\boldsymbol{v}, \tau, \sigma_1, \sigma_2, \sigma_3, \sigma_4)$. Output 1 if both of the following two conditions hold.

$$e(g, \sigma_1) = e(\prod_{i=1}^{n} g_i^{v_i}, g^{\alpha}) \cdot e(h, \sigma_4) \cdot e(H_{\mathbb{G}}(\tau), \sigma_2)$$
$$e(g^{\alpha}, \sigma_2) = (g, \sigma_4)$$

Theorem 7. *The simplified variant of the ALP LHS scheme is unforgeable (under Definition 10) if the CDH and FlexCDH assumptions hold in the group* $\mathbb{G}$.

C.2 Instantiation of Our 3rd Generic KARIP Construction

$\mathtt{Setup}(1^{\lambda}, L)$**:** Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ whose order is a prime p. Conduct the following three steps.

1. Generate a key-pair of the simplified ALP LHS scheme [4] in basically the same manner as our 2nd generic construction in Subsect. 5.1. Number of group elements $n+5$ is reduced to $n+4$.
2. Generate a hash-key of the hash function, i.e., $hk \leftarrow \mathsf{H}.\mathtt{KGen}(1^{\lambda})$.
3. Generate a GS CRS $\boldsymbol{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$.

Output (pp, mk), where $pp := (\mathbb{G}, \mathbb{G}_T, e, g, g^{\alpha}, h, \{g_i\}_{i=1}^{n+4}, u', \{u_i\}_{i=1}^{N}, hk, \boldsymbol{f})$ and $mk := \alpha$.

$\mathtt{KGen}(mk, \mathbf{x}, L, R)$**:** Choose an LHS tag $\tau \xleftarrow{\mathrm{U}} \{0,1\}^N$. Let $\boldsymbol{v}_1 := (x_1, x_2, \cdots, x_n, L, R, 0, 1) \in \mathbb{Z}_p^{n+4}$ and $\boldsymbol{v}_2 := (0, \cdots, 0, 1, 0) \in \mathbb{Z}_p^{n+4}$. For $i \in \{1,2\}$, generate a signature σ_i of the ALP LHS scheme on $\boldsymbol{v}_i$. Output $sk := (\tau, \{\sigma_i\}_{i=1}^{2})$.

$\mathtt{Sig}(sk, M, \mathbf{y})$**:** Parse sk as above. Let $d := \langle \mathbf{x}, \mathbf{y} \rangle \pmod p$. Assume that $d \in [L, R]$. Firstly, conduct the following three steps.

1. Compute $h \leftarrow \mathsf{H}.\mathtt{Eval}(hk, (\mathbf{y}, M))$. Derive an LHS signature on $\boldsymbol{v}' := (x_1, \cdots, x_n, L, R, h, 1)$. Let $\beta_1 := 1$ and $\beta_2 := h$. Choose $r' \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Compute $\sigma' := (\sigma'_1, \sigma'_2, \sigma'_3, \sigma'_4) := (\prod_{i=1}^{2} \sigma_{i,1}^{\beta_i} \cdot H_{\mathbb{G}}(\tau)^{r'}, \prod_{i=1}^{2} \sigma_{i,2}^{\beta_i} \cdot g^{r'}, \prod_{i=1}^{2} \sigma_{i,3}^{\beta_i}, \prod_{i=1}^{2} \sigma_{i,4}^{\beta_i})$.
2. Generate GS commitments for all of the following group elements.
   (a) $g^{\tau[i]}$ and $g^{1-\tau[i]}$ (for all $i \in [1, N]$)
   (b) $H_{\mathbb{G}}(\tau)$
   (c) $g^{d[i]}$, $g^{1-d[i]}$, $g_{n+1}^{L[i]}$, $g_{n+1}^{1-L[i]}$, $g_{n+2}^{R[i]}$ and $g_{n+2}^{1-R[i]}$ (for all $i \in [1, \lambda]$)
   (d) g^d, g_{n+1}^{L} and g_{n+2}^{R}
   (e) σ'_1, σ'_3 and σ'_4
   (f) g^{x_i} and $g_i^{x_i}$ (for all $i \in [1, n]$)

   They are denoted by $\vec{C}_{\tau[i]}$, $\vec{C}_{1-\tau[i]}$, $\vec{C}_{H_{\mathbb{G}}(\tau)}$, $\vec{C}_{d[i]}$, $\vec{C}_{1-d[i]}$, $\vec{C}_{L[i]}$, $\vec{C}_{1-L[i]}$, $\vec{C}_{R[i]}$, $\vec{C}_{1-R[i]}$, $\vec{C}_{d}$, $\vec{C}_{L}$, $\vec{C}_{R}$, $\vec{C}_{\sigma_1}$, $\vec{C}_{\sigma_3}$, $\vec{C}_{\sigma_4}$, $\vec{C}_{x_i}$ and $\vec{C}'_{x_i}$. Note that the group elements (a)-(e) are (basically) unchanged from our 2nd KARIP scheme.
3. Generate GS proofs for all of the following PPEs.
   [a] $e(g^{\tau[i]}, g^{1-\tau[i]}) = 1_{\mathbb{G}_T}$ and $e(g^{\tau[i]}, g) \cdot e(g^{1-\tau[i]}, g) = e(g, g)$ (for all $i \in [1, N]$)

[b] $e(H_{\mathbb{G}}(\tau), g) = e(u', g) \prod_{i=1}^{N} e(u_i, g^{\tau[i]})$

[c] $e(g^{d[i]}, g^{1-d[i]}) = 1_{\mathbb{G}_T}$, $e(g^{d[i]}, g) \cdot e(g^{1-d[i]}, g) = e(g, g)$, $e(g_{n+1}^{L[i]}, g_{n+1}^{1-L[i]}) = 1_{\mathbb{G}_T}$, $e(g_{n+1}^{L[i]}, g) \cdot e(g_{n+1}^{1-L[i]}, g) = e(g_{n+1}, g)$,

[d] $e(g_{n+2}^{R[i]}, g_{n+2}^{1-R[i]}) = 1_{\mathbb{G}_T}$ and $e(g_{n+2}^{R[i]}, g) \cdot e(g_{n+2}^{1-R[i]}, g) = e(g_{n+2}, g)$ (for all $i \in [1, \lambda]$)

[e] $e(g^d, g) = \prod_{i=1}^{\lambda} e(g^{d[i]}, g^{2^{i-1}})$, $e(g_{n+1}^{L}, g) = \prod_{i=1}^{\lambda} e(g_{n+1}^{L[i]}, g^{2^{i-1}})$ and $e(g_{n+2}^{R}, g) = \prod_{i=1}^{\lambda} e(g_{n+2}^{R[i]}, g^{2^{i-1}})$

[f] $e(\sigma_1', g) = \prod_{i=1}^{n} e(g_i^{x_i}, g^{\alpha}) \cdot e(g_{n+1}^{L}, g^{\alpha}) \cdot e(g_{n+2}^{R}, g^{\alpha}) \cdot e(g_{n+3}^{h} \cdot g_{n+4}, g^{\alpha}) \cdot e(h, \sigma_4') \cdot e(H_{\mathbb{G}}(\tau), \sigma_2')$

[g] $e(\sigma_3', g^{\alpha}) = e(g, \sigma_4')$

[h] $e(g_i^{x_i}, g) = e(g_i, g^{x_i})$ (for all $i \in [1, n]$)

[i] $e(g^d, g) = \prod_{i=1}^{n} e(g^{x_i}, g^{y_i})$

They are denoted by $\vec{\pi}_{\tau[i],mul}$, $\vec{\pi}_{\tau[i],sum}$, $\vec{\pi}_{H_{\mathbb{G}}(\tau)}$, $\vec{\pi}_{d[i],mul}$, $\vec{\pi}_{d[i],sum}$, $\vec{\pi}_{L[i],mul}$, $\vec{\pi}_{L[i],sum}$, $\vec{\pi}_{R[i],mul}$, $\vec{\pi}_{R[i],sum}$, $\vec{\pi}_{d}$, $\vec{\pi}_{L}$, $\vec{\pi}_{R}$, $\vec{\pi}_{\sigma_1}$, $\vec{\pi}_{\sigma_3}$, $\vec{\pi}_{x_i}$ and $\vec{\pi}_{d,ip}$. Note that the PPEs [a]-[g] are (basically) unchanged from our 2nd KARIP scheme.

What remains is proving $d \in [L, R] \pmod p$. In the same manner as our second instantiated scheme in Subsect. 5.2, generate the following GS commitments and proofs, namely GS commitments $\{\vec{C}_{B_i}, \vec{C}_{C_i}, \vec{C}_{D_i}, \vec{C}_{E_i}, \vec{C}_{F_i}, \vec{C}_{G_i}\}_{i=1}^{\lambda}$, and GS proofs $\{\vec{\pi}_{C_i}, \vec{\pi}_{D_i}, \vec{\pi}_{B_i}, \vec{\pi}_{F_i}, \vec{\pi}_{G_i}, \vec{\pi}_{E_i}\}_{i=1}^{\lambda}$, $\vec{\pi}_A$ and $\vec{\pi}_{A'}$. Finally, output a signature σ which is set to

$$\left(\begin{array}{c} \{\vec{C}_{\tau[i]}, \vec{C}_{1-\tau[i]}, \vec{\pi}_{\tau[i],mul}, \vec{\pi}_{\tau[i],sum}\}_{i=1}^{N}, \\ \{\{\vec{C}_{x[i]}, \vec{C}_{1-x[i]}, \vec{\pi}_{x[i],mul}, \vec{\pi}_{x[i],sum}\}_{i=1}^{\lambda}, \vec{C}_{x}, \vec{\pi}_{x}\}_{x \in \{d,L,R\}}, \\ \vec{C}_{H_{\mathbb{G}}(\tau)}, \vec{\pi}_{H_{\mathbb{G}}(\tau)}, \vec{C}_{\sigma_1}, \sigma_2', \vec{C}_{\sigma_3}, \vec{C}_{\sigma_4}, \vec{\pi}_{\sigma_1}, \vec{\pi}_{\sigma_3}, \vec{\pi}_{A}, \vec{\pi}_{A'}, \\ \{\vec{C}_{B_i}, \vec{C}_{C_i}, \vec{C}_{D_i}, \vec{\pi}_{B_i}, \vec{\pi}_{C_i}, \vec{\pi}_{D_i}, \vec{C}_{E_i}, \vec{C}_{F_i}, \vec{C}_{G_i}, \vec{\pi}_{E_i}, \vec{\pi}_{F_i}, \vec{\pi}_{G_i}\}_{i=1}^{\lambda}, \\ \boxed{\{\vec{C}_{x_i}, \vec{C}'_{x_i}, \vec{\pi}_{x_i}\}_{i=1}^{n}, \vec{\pi}_{d,ip}} \end{array} \right). \tag{6}$$

The only difference between (6) and (4) is w.r.t. the elements in a rectangle.

$\mathtt{Ver}(\sigma, M, \mathbf{y})$: Each GS proof $\vec{\pi} \in \mathbb{G}^3$ (resp. $\vec{\pi} \in \mathbb{G}^9$), composed of 3 (resp. 9) elements in $\mathbb{G}$, is parsed as (π_1, π_2, π_3) (resp. $(\vec{\pi}_1, \vec{\pi}_2, \vec{\pi}_3)$ with $\vec{\pi}_i \in \mathbb{G}^3$). Output 1 iff all of the following equations hold.

1. $F(\vec{C}_{\tau[i]}, \vec{C}_{1-\tau[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{\tau[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, N]$)
2. $E(g, \vec{C}_{\tau[i]}) \cdot E(g, \vec{C}_{1-\tau[i]}) = \iota_{\mathbb{G}_T}(e(g, g)) \prod_{k=1}^{3} E(\pi_{\tau[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, N]$)
3. $E(g, \vec{C}_{H_{\mathbb{G}}(\tau)}) = \iota_{\mathbb{G}_T}(e(u', g)) \prod_{i=1}^{N} E(u_i, \vec{C}_{\tau[i]}) \prod_{k=1}^{3} E(\pi_{\tau[i],k}, \vec{f}_k)$
4. $F(\vec{C}_{d[i]}, \vec{C}_{1-d[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{d[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
5. $E(g, \vec{C}_{d[i]}) \cdot E(g, \vec{C}_{1-d[i]}) = \iota_{\mathbb{G}_T}(e(g, g)) \prod_{k=1}^{3} E(\pi_{d[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
6. $E(g, \vec{C}_{d}) = \prod_{i=1}^{\lambda} E(g^{2^{i-1}}, \vec{C}_{d[i]}) \prod_{k=1}^{3} E(\pi_{d,k}, \vec{f}_k)$
7. $E(g, \vec{C}_{\sigma_1}) = \prod_{i=1}^{n} E(g^{\alpha}, \vec{C}_{x_i}) \cdot E(g^{\alpha}, \vec{C}_{L}) \cdot E(g^{\alpha}, \vec{C}_{R}) \iota_{\mathbb{G}_T}(e(g_{n+3}^{h} \cdot g_{n+4}, g^{\alpha})) \cdot E(h, \vec{C}_{\sigma_4}) \cdot E(\sigma_2', \vec{C}_{H_{\mathbb{G}}(\tau)}) \prod_{k=1}^{3} E(\pi_{\sigma_1,k}, \vec{f}_k)$

8. $E(g^{\alpha}, \vec{C}_{\sigma_3}) = E(g, \vec{C}_{\sigma_4}) \prod_{k=1}^{3} E(\pi_{\sigma_3,k}, \vec{f}_k)$
9. $F(\vec{C}_{L[i]}, \vec{C}_{1-L[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{L[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
10. $E(g, \vec{C}_{L[i]}) \cdot E(g, \vec{C}_{1-L[i]}) = \iota_{\mathbb{G}_T}(e(g_{n+1}, g)) \prod_{k=1}^{3} E(\pi_{L[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
11. $E(g, \vec{C}_L) = \prod_{i=1}^{\lambda} E(g^{2^{i-1}}, \vec{C}_{L[i]}) \prod_{k=1}^{3} E(\pi_{L,k}, \vec{f}_k)$
12. $F(\vec{C}_{R[i]}, \vec{C}_{1-R[i]}) = \prod_{k=1}^{3} F(\vec{\pi}_{R[i],mul,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
13. $E(g, \vec{C}_{R[i]}) \cdot E(g, \vec{C}_{1-R[i]}) = \iota_{\mathbb{G}_T}(e(g_{n+2}, g)) \prod_{k=1}^{3} E(\pi_{R[i],sum,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
14. $E(g, \vec{C}_R) = \prod_{i=1}^{\lambda} E(g^{2^{i-1}}, \vec{C}_{R[i]}) \prod_{k=1}^{3} E(\pi_{R,k}, \vec{f}_k)$
15. $F(\iota_{\mathbb{G}}(g_{n+1}), \vec{C}_{C_i}) = F(\vec{C}_{d[i]}, \vec{C}_{1-L[i]}) \prod_{k=1}^{3} F(\vec{\pi}_{C_i,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
16. $F(\iota_{\mathbb{G}}(g_{n+1}), \vec{C}_{D_i}) = F(\vec{C}_{d[i]}, \vec{C}_{L[i]}) \cdot F(\vec{C}_{1-d[i]}, \vec{C}_{1-L[i]}) \prod_{k=1}^{3} F(\vec{\pi}_{D_i,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
17. $E(g, \vec{C}_{B_1}) = E(g, \vec{C}_{D_1}) \prod_{k=1}^{3} E(\pi_{B_1,k}, \vec{f}_k)$
18. $F(\iota_{\mathbb{G}}(g), \vec{C}_{B_i}) = F(\vec{C}_{B_{i-1}}, \vec{C}_{D_i}) \prod_{k=1}^{3} F(\vec{\pi}_{B_i,k}, \vec{f}_k)$ (for all $i \in [2, \lambda]$)
19. $F(\iota_{\mathbb{G}}(g), \vec{C}_{C_1}) \prod_{i=1}^{\lambda} \cdot F(\vec{C}_{B_{i-1}}, \vec{C}_{C_i}) \cdot F(\iota_{\mathbb{G}}(g), \vec{C}_{B_\lambda}) = \Gamma_{\mathbb{G}_T}(e(g, g)) \prod_{k=1}^{3} F(\vec{\pi}_{A,k}, \vec{f}_k)$
20. $F(\iota_{\mathbb{G}}(g_{n+2}), \vec{C}_{F_i}) = F(\vec{C}_{1-d[i]}, \vec{C}_{R[i]}) \prod_{k=1}^{3} F(\vec{\pi}_{F_i,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
21. $F(\iota_{\mathbb{G}}(g_{n+2}), \vec{C}_{G_i}) = F(\vec{C}_{d[i]}, \vec{C}_{R[i]}) \cdot F(\vec{C}_{1-d[i]}, \vec{C}_{1-R[i]}) \prod_{k=1}^{3} F(\vec{\pi}_{G_i,k}, \vec{f}_k)$ (for all $i \in [1, \lambda]$)
22. $E(g, \vec{C}_{E_1}) = E(g, \vec{C}_{G_1}) \prod_{k=1}^{3} E(\pi_{E_1,k}, \vec{f}_k)$
23. $F(\iota_{\mathbb{G}}(g), \vec{C}_{E_i}) = F(\vec{C}_{E_{i-1}}, \vec{C}_{G_i}) \prod_{k=1}^{3} F(\vec{\pi}_{E_i,k}, \vec{f}_k)$ (for all $i \in [2, \lambda]$)
24. $F(\iota_{\mathbb{G}}(g), \vec{C}_{F_1}) \prod_{i=1}^{\lambda} \cdot F(\vec{C}_{E_{i-1}}, \vec{C}_{F_i}) \cdot F(\iota_{\mathbb{G}}(g), \vec{C}_{E_\lambda}) = \Gamma_{\mathbb{G}_T}(e(g, g)) \prod_{k=1}^{3} F(\vec{\pi}_{A',k}, \vec{f}_k)$
25. $E(g, \vec{C}_{x_i}) = E(g_i, \vec{C}'_{x_i}) \prod_{k=1}^{3} E(\pi_{x_i,k}, \vec{f}_k)$
26. $E(g, \vec{C}_d) = \prod_{i=1}^{n} E(g^{y_i}, \vec{C}_{x_i}) \prod_{k=1}^{3} E(\pi_{d,ip,k}, \vec{f}_k)$

Note that the first 24 relations are (basically) unchanged from our 2nd KARIP scheme.

Corollary 3. *Our 3rd KARIP scheme is* `UNF` *if the DLIN, CDH and FlexCDH assumptions hold in the group* $\mathbb{G}$ *and the hash function is collision-resistant. The scheme is* `PRV` *unconditionally.*