Registered Attribute-Based Signature

Yijian Zhang[1], Jun Zhao[1], Ziqi Zhu[1], Junqing Gong[1,2,✉], and Jie Chen[1,✉]

[1] Shanghai Key Laboratory of Trustworthy Computing, School of Software Engineering, East China Normal University, Shanghai, China. jqgong@sei.ecnu.edu.cn, s080001@e.ntu.edu.sg

[2] Shanghai Qi Zhi Institute, Shanghai, China.

Abstract. This paper introduces the notion of *registered attribute-based signature* (registered ABS). Distinctly different from classical *attribute-based signature* (ABS), registered ABS allows any user to generate their own public/secret key pair and register it with the system. The *key curator* is critical to keep the system flowing, which is a fully transparent entity that does not retain secrets. Our results can be summarized as follows.

- This paper provides the first *definition* of registered ABS, which has never been defined.
- This paper presents the first generic *fully secure* registered ABS over the prime-order group from k-Lin assumption under the standard model, which supports various classes of predicate.
- This paper gives the first concrete registered ABS scheme for *arithmetic branching program* (ABP), which achieves *full security* in the standard model.

Technically, our registered ABS is inspired by the blueprint of Okamoto and Takashima[PKC'11]. We convert the prime-order *registered attribute-based encryption* (registered ABE) scheme of Zhu et al.[ASIACRYPT'23] via predicate encoding to registered ABS by employing the technique of *re-randomization with specialized delegation*, while we employ the different dual-system method considering the property of registration. Prior to our work, the work of solving the key-escrow issue was presented by Okamoto and Takashima[PKC'13] while their work considered the weak adversary in the random oracle model.

Keywords: Registered Attribute-Based Signature; Predicate Encoding; Dual System Encryption.

1 Introduction

Attribute-Based Signature. *Attribute-based signature* (ABS) [MPR08,OT11] provides the fine-grained control to authentication privileges while guaranteeing anonymous authentication of message, which extends the traditional digital signature [DH76]. In ABS for predicate $P : X \times Y \to \{0,1\}$, the signer employs signing key sk_y, where $y \in Y$ is his/her attribute set, to sign message under policy $x \in X$ only when $P(x, y) = 1$. Anyone can verify the signature by using solely public parameters. The basic security condition of ABS is *unforgeability*, i.e., an adversary holding a signing key with $P(x, y) = 0$ cannot generate a valid signature; furthermore, this should be ensured when the adversary has more than one key.

Decentralized Attribute-Based Signature. To circumvent the key escrow problem in ABS [SAH16a,DOT19a,DDM23], Okamoto and Takashima introduced the notion of *decentralized ABS* [OT13] which means that different authorities with attributes can join the system instead of having only one central authority. However, decentralized ABS just solves the problem that attributes from the single part, but the keys come from different central authorities, and if a sufficient number of authorities are compromised or corrupted, then the scheme will no longer ensure unforgeability.

This Work. Recently, the notion of *registration-based encryption* (RBE) [GHMR18] and *registered attribute-based encryption* (registered ABE) [HLWW23,ZZGQ23,FFM+23] has been studied, which allows users in the system to generate their own public/secret keys and then register their public keys together with the key curator. The key curator keeps nothing about secrets in contrast to the conventional attribute authority. However, the feasibility of this strategy in ABS is still unknown and a natural question that arises is

Can we construct a registered attribute-based signature scheme that even supports monotone span program?

More details, in the *registered attribute-based signature* (registered ABS) scheme, each user can generate his/her own key pair $(\mathsf{pk}, \mathsf{sk})$ locally and register (pk, y) for some $y \in Y$ into the system. Registration is performed by the key curator in a public and deterministic manner, and will generate a master public key mpk for anyone who wants to verify the signature as a traditional ABS. Besides, during the registration phase, each user can obtain his/her own helper key hk from the curator, which can be used to generate signature for policy $x \in X$ with sk when $P(x, y) = 1$. Finally, as the number of user in the system increases, the curator may trigger an update to all users' helper keys.

1.1 Results

In this work, we have addressed the above question. We propose the first generic registered attribute-based signature via predicate encoding [Wee14,CGW15]. Our scheme relies on the well-known k-Lin assumption for $k \geq 1$ over the prime-order bilinear group in the standard model. Our contribution is as follows.

- This paper introduces the first definition of registered ABS, formalizing the fact that any user can generate their own public/secret key pair and register it with the system. Furthermore, we formalize a security notion of registered ABS, i.e., the signature is unforgeable against the adversary with corrupted user information.
- This paper proposes the first generic approach for registered ABS supporting various classes of predicate over the prime-order group under the standard model, while the previous scheme of decentralized ABS [OT13] relies on the random oracle model.
- This paper gives the first concrete registered ABS scheme for the expressive predicate *arithmetic branching program*(ABP). It is fully secure under the k-Lin assumption in the standard model.

We present a concrete comparison in Table 1. Although this table only involves [OT11,OT13], we note that other ABS constructions [AHY15,SAH16b,DOT19b] published recently share similar properties and all of them suffer from key escrow issue.

Reference	Key-escrow	Standard	Assumption
ABS [OT11]	✗	✓	DLIN
Decentralized ABS [OT13]	†	✗	DLIN
Ours	✓	✓	k-Lin

Table 1. Comparison among prior works. Here, the column "Standard" denotes the standard model. "†" means that decentralized ABS is unable to completely eliminate key escrow issue since it still needs nontransparent authorities to store secret values.

1.2 Related Work

Since Maji et al. [MPR08] put forward the notion of ABS, there exists two research lines on ABS: The first line is to enhance the expression ability of ABS schemes. Herranz et al. [HLLR12] proposed an ABS scheme with constant-size signatures supporting threshold predicate. Okamoto et al. [OT11] proposed a fully secure ABS scheme for non-monotone span program in the standard model. Attrapadung et al. [AHY15] designed an ABS scheme with constant-size signatures that supports non-monotone span programs. Furthermore, Sakai et al. [SAH16b] built an ABS scheme supporting circuits via Groth-Sahai proofs over bilinear groups. Datta et al. [DOT19b] designed a fully secure ABS for ABP with unbounded multi-use of attributes. In these works, a central authority must be set to store master secret key. To tackle the key escrow problem, another line is to build decentralized ABS. Okamoto et al. [OT11] proposed the first decentralized multi-authority ABS scheme for non-monotone span programs while it is only provably secure under the random model. Our registered ABS can seen as an independent work of the second line.

Organization. We provide the technique overview in section 2. We give the definition of (slotted) registered ABS in section 3. The details of our slotted registered ABS are presented in section 4. Besides, a generic approach based on slotted registered ABS is presented in section 5. Finally, we derive a concrete slotted registered ABS scheme supporting expressive ABP in section 6.

2 Technique Overview

In this work, we construct a *registered attribute-based signature* (registered ABS) via predicate encoding [Wee14,CGW15], and the scheme is based on well-known k-Lin assumption in the standard model. The design core of registered ABS is similar to [HLWW23,ZZGQ23], we start from slotted registered ABS and then convert it to full-fledged registered ABS. Before going into the technical descriptions of the designing of primitives in registered setting, we first provide an overview of the notion of registered ABS.

2.1 Registered Attribute-Based Signature

Definition. We introduce the definition of registered ABS in the simplest setting, which is inspired by the idea of [HLWW23,ZZGQ23]. A registered ABS scheme for predicate $P : X \times Y \to \{0,1\}$ consists of the following six algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Reg}, \mathsf{Upd}, \mathsf{Sig}, \mathsf{Ver})$:

- Setup provides a common reference string crs for each user to register;
- Gen allows each user to generate their own public/secret key pair $(\mathsf{pk}, \mathsf{sk})$;
- Reg is a transparent and deterministic algorithm, which checks the validity of the registered user, and register user's pk and attribute $y \in Y$ into the master public key mpk; Upd returns helper key hk for the registered user;
- Sig with hk and user's secret key sk returns a signature on (x, m) when $P(x, y) = 1$, where $x \in X$ is the signature policy; Ver can check the validity of signature with just mpk.

Properties. Assuming $L \in \mathbb{N}$ denotes the user number in registered ABS system, it has some essential efficiency requirements: the size of crs to be $\mathsf{poly}(\lambda, L)$ where λ is the security parameter of the system, and the size of mpk, hk to be $\mathsf{poly}(\lambda, P, \log L)$ where P is the size of predicate.

Furthermore, the security of registered ABS means *unforgeability* compared with the IND-security of registered ABE. It ensures that no one is able to forge a signature passing the verification without the knowledge of the sign secret key. Similar to registered ABE, the security model needs to consider both honest and corrupted users.

Generic Approach. We state that the "power-of-two" approach in [HLWW23] can be improved to derive a generic approach to obtaining registered ABS. The approach needs a new primitive, namely *slotted registered attribute-based signature* (slotted registered ABS), as the underlying block. Slotted registered ABS has the similar syntax as registered ABS except that it does not consider the update of public parameters.

Based on the "power-of-two" approach, we replace encryption and decryption algorithms with $(\mathsf{Sig}, \mathsf{Ver})$, so it is necessary to demonstrate the reduction from the unforgeability of registered ABS to the unforgeability of underlying slotted registered ABS. The proof relies on the fact that the signature and verification text in registered ABS consist of multiple copies of underlying slotted registered ABS. More details are available in section 5. Next, we will construct a slotted registered ABS based the techniques of predicate encoding.

2.2 Slotted Registered ABS

Firstly, let us define some notations, which will be used. Our slotted registered ABS relies on an asymmetric bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ of prime-order p with pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. For $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_p$, we define $[a]_s = g_s^a$ as the implicit representation of a in $\mathbb{G}_s$. Then, we give preliminary which is an important part of our slotted registered ABS.

Preliminary. For a predicate $P : X \times Y \to \{0, 1\}$, define a (n, n_c, n_k)-predicate encoding: For all $x \in X$, $y \in Y$, one can efficiently and deterministically find $\mathbf{C}_x \in \mathbb{Z}_p^{n \times n_c}$, $\mathbf{K}_y \in \mathbb{Z}_p^{n \times n_k}$, $\mathbf{a}_y \in \mathbb{Z}_p^{1 \times n_k}$ and $\mathbf{d}_{x,y} \in \mathbb{Z}_p^{n_c + n_k}$ that forms $\mathbf{M}_{x,y} = \begin{pmatrix} \mathbf{a}_y & \mathbf{0}_{n_c} \\ \mathbf{K}_y & \mathbf{C}_x \end{pmatrix}$ such that

- when $P(x, y) = 1$, we have $\mathbf{M}_{x,y}\mathbf{d}_{x,y}^\top = (1, 0, \ldots, 0)^\top$;
- when $P(x, y) = 0$, we have $\{x, y, \alpha, (\alpha \| \mathbf{w})\mathbf{M}_{x,y}\} \approx_s \{x, y, \alpha, (0 \| \mathbf{w})\mathbf{M}_{x,y}\}$ where $\mathbf{w} \leftarrow \mathbb{Z}_p^n$.

Initial Idea. Our initial idea is to apply Naor's paradigm [BF01,BLS01], which has successfully transform some encryption schemes into signature's version [OT11,OT13,CLL+14], on existing slotted registered ABE. For such purpose, we choose Zhu et al.'s slotted registered ABE [ZZGQ23] as our start point, because their construction is based on *predicate encoding* supporting a large number of expressive predicates, even including arithmetic branching programs (ABP).

Start From Slotted Registered ABE. In slotted registered ABE, after initializing the common reference string crs, all users can generate their own key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)_{i \in [L]}$ and submit respective pk_i to the aggregator who subsequently outputs mpk and hk_i for user/slot i. The ciphertext ct_x is an encryption on (x, m) and can be decrypted correctly with $(\mathsf{sk}_i, \mathsf{hk}_i)$ if and only if $P(x, y_i) = 1$. We recap Zhu et al.'s slotted registered ABE construction based on predicate encoding as follows:

$$
\begin{aligned}
\mathsf{crs} &: [\alpha]_T, \{[v_j, \mathbf{w}_j]_1\}_{j \in [L]}, \{[r_i, r_i v_j, r_i \mathbf{w}_j, r_i v_i + \alpha]_2\}_{i \neq j} \\
\mathsf{pk}_i &: [u_i]_1, \{[u_i r_j]_2\}_{j \neq i} \\
\mathsf{sk}_i &: u_i \\
\mathsf{mpk} &: [\textstyle\sum_j ((v_j + u_j) a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}),\ \sum_j \mathbf{w}_j]_1, [\alpha]_T \\
\mathsf{hk}_i &: [r_i, r_i v_i + \alpha, r_i \textstyle\sum_{j \neq i} ((v_j + u_j) a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}),\ r_i \sum_{j \neq i} \mathbf{w}_j]_2 \\
\mathsf{ct}_x &: [s, s \textstyle\sum_j ((v_j + u_j) a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}),\ s \sum_j \mathbf{w}_j \mathbf{C}_x]_1, [s\alpha]_T \cdot \mathsf{m}
\end{aligned}
\tag{1}
$$

where for all $j \in [L]$, y_j is the attribute embedded in slot j; $\alpha, v_j, r_j, u_j \leftarrow \mathbb{Z}_p$ and $\mathbf{w}_j \in \mathbb{Z}_p^n$; $s \leftarrow \mathbb{Z}_p$ is the randomness in ct_x. As for decryption, it firstly computes the pairing result between hk_i and ct_x to cancel cross items from other slots $j \in [L] \setminus \{i\}$, then proceed the decryption of predicate encoding on slot i. If $P(x, y_i) = 1$, just use $\mathsf{sk}_i = u_i$

to recover $[\alpha s]_T$ and thus obtain the message m.

First Try. Now, we make an attempt to transform the slotted registered ABE in (1) into a slotted registered ABS. Our strategy is to treat $(\mathsf{hk}_i, \mathsf{sk}_i)$ as the sign secret key of user i, and ct_x as the verification text v, respectively. Here, sk_i should be privacy, while hk_i is publicly computed with crs and $(\mathsf{pk}_j)_{j\in[L]\setminus\{i\}}$. A signature $\sigma_{i,x,\mathsf{m}}$ is derived from $(\mathsf{sk}_i, \mathsf{hk}_i)$. Then it uses $(\mathsf{mpk}, x, \mathsf{m})$ to generate a verification text v_{i^*,x^*,m^*} to verify the validity of $\sigma_{i,x,\mathsf{m}}$. Intuitively, we have

$$\sigma_{i,x,\mathsf{m}} : [r_i, u_i r_i + r_i v_i + \alpha, r_i \sum_{j\neq i}((v_j + u_j)a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}),\ r_i \sum_{j\neq i} \mathbf{w}_j \mathbf{C}_x]_2$$
$$v_{i^*,x^*,\mathsf{m}^*} : [s, s\sum_j((v_j + u_j)a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}),\ s\sum_j \mathbf{w}_j \mathbf{C}_{x^*}]_1, [s\alpha]_T \cdot \mathsf{m}^*$$

where $\mathsf{crs}, \mathsf{mpk}, \mathsf{pk}_i, \mathsf{sk}_i$ and hk_i are identical to the equality (1). Observe that the verification process is identical to the decryption process in (1) except that it can directly recover m without u_i, then checkout if $\mathsf{m} = \mathsf{m}^*$.

Actually, the above scheme is insecure, since the information of u_i is leaked from $\sigma_{i,x,\mathsf{m}}$. Note that any signature could be forged readily if $\mathsf{sk}_i = u_i$ is leaked. Besides, $\sigma_{i,x,\mathsf{m}}$ does not involve m and the generation is completely deterministic.

Second Try. Inspired by the "re-randomization" technique of [OT11], we state that $\sigma_{i,x,\mathsf{m}}$ actually plays a role as a special decryption key. Concretely, it should remain the decryption ability of $(\mathsf{sk}_i, \mathsf{hk}_i)$, but still preserve the privacy of sk_i to avoid the forgery. However, such technique cannot be trivially applied to our scheme, since our secret key is the secret value chosen by user, rather than the well-constructed secret key generated by authority. Therefore, the problem is how to generate desired signature and ensure security proof in our slotted registered ABS.

Our technique path is as follows: Firstly, to protect the confidentiality of u_i, we generate extra entropy by appending a new equality into the signature, which ensures adversary cannot obtain secret information from honest users. Secondly, the privacy of predicate encoding can ensure that adversary cannot obtain secret information from corrupted users. Finally, we use $(\mathsf{sk}_i, \mathsf{hk}_i)$ to delegate a new signature $\sigma_{i,x,\mathsf{m}}$ as follows:

$$\begin{aligned}
\mathsf{pk}_i &: [u_i, c_i, d_i]_1, \{[u_i r_j]_2\}_{j\neq i};\\
\mathsf{sk}_i &: u_i, c_i, d_i\\
\sigma_{i,x,\mathsf{m}} &: [t, r_i, t(c_i + m\cdot d_i) + u_i r_i + r_i v_i + \alpha]_2,\\
&\quad [r_i \sum_{j\neq i}((v_j + u_j)a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}),\ r_i \sum_{j\neq i} \mathbf{w}_j \mathbf{C}_x]_2\\
v_{i^*,x^*,\mathsf{m}^*} &: [s, s(c_{i^*} + m^*\cdot d_{i^*}), s\sum_j((v_j + u_j)a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}), s\sum_j \mathbf{w}_j \mathbf{C}_{x^*}]_1, [s\alpha]_T
\end{aligned}$$

Here, we define a collusion-resistant hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_p$, $m \leftarrow \mathsf{H}(i, x, \mathsf{m})$ and $m^* \leftarrow \mathsf{H}(i^*, x^*, \mathsf{m}^*)$, where i is more like a pseudo-identity. t is the randomness newly sampled in each signature. Observe that u_i has been totally hidden in the signature as long as $m \neq m^*$ (in the similar sense of [CLL+14]). Thus, we can ensure that the adversary has no ability to forge a valid signature unless slot i is corrupted. Lastly, this construction is still unreasonable since the generation of v_{i^*,x^*,m^*} needs both mpk and pk_{i^*}, which contradicts the definition of verification algorithm, but we can fix it by aggregating pk_i from all users.

Our Slotted Registered ABS. Finally, putting the above together, we obtain a new slotted registered ABS as follows:

$$\mathsf{crs} : [\alpha]_T, \{[v_j, \mathbf{w}_j]_1\}_{j\in[L]}, \{[r_i, r_i v_j, r_i \mathbf{w}_j, r_i v_i + \alpha]_2\}_{i\neq j}, \mathsf{H}$$

$$\mathsf{pk}_i : [u_i, c_i, d_i]_1, [c_i, d_i]_2, \{[u_i r_j]_2\}_{j\neq i}$$

$$\mathsf{sk}_i : u_i, c_i, d_i$$

$$\mathsf{mpk} : [\sum_j ((v_j + u_j)a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}),\ \sum_j \mathbf{w}_j, \sum_j c_j, \sum_j d_j]_1, [\alpha]_T, \mathsf{H}$$

$$\mathsf{hk}_i : [r_i, r_i v_i + \alpha, r_i \sum_{j\neq i} ((v_j + u_j)a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}), r_i \sum_{j\neq i} \mathbf{w}_j,\ \sum_{j\neq i} c_j, \sum_{j\neq i} d_j]_2, \mathsf{H}$$

$$\sigma_{i,x,\mathsf{m}} : [t, r_i, t(c_i + m \cdot d_i) + u_i r_i + r_i v_i + \alpha, t \sum_{j\neq i} (c_j + m \cdot d_j), r_i \sum_{j\neq i} ((v_j + u_j)a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}),\ r_i \sum_{j\neq i} \mathbf{w}_j \mathbf{C}_x]_2$$

$$v_{i^*,x^*,\mathsf{m}^*} : [s, s \sum_j (c_j + m^* \cdot d_j), s \sum_j ((v_j + u_j)a_{y_j} + \mathbf{w}_j \mathbf{K}_{y_j}), s \sum_j \mathbf{w}_j \mathbf{C}_{x^*}]_1, [s\alpha]_T.$$

Then we apply the generic approach [CGW15] from composite-order group to prime-order group to above construction, and obtain a secure slotted registered ABS based on k-Lin assumption under the standard model. We still adopt the dual system encryption as proof strategy, while the proof detail is quite distinct from previous works [OT11,OT13] since there is no longer authority holding secret keys in the system.

2.3 Discussion and Open Problem

Here, we discuss the future work about registered ABS.

- Our registered ABS achieves various classes of predicate even including span programs, but the concrete scheme for more expressive predicate (e.g., finite state automata and circuits) is still unknown.
- The signer anonymity of ABS says that the generated signature reveals no information on the signer's attribute other than the fact that the signature is valid. However, just as mentioned in [SKAH18], ABS schemes derived from ABE generally do not provide anonymity property. This argument also works in our registered ABS. Intuitively, the public and deterministic registration of user attribute in registered ABS also hinders signer anonymity to a large extent. Thus, we list the realization of signer anonymity in the standard model as one of future works.
- Our work opens a new and promising path for pairing-based research on registered ABS. An open question, however, is whether we can propose registered ABS under the LWE assumption. Furthermore, the size of crs is $\mathsf{poly}(\lambda, L)$ where λ is the security parameter of the system and L is the number of users, it is still an open problem to reduce the size of crs to $\mathsf{poly}(\lambda)$ under standard assumption.

3 Preliminaries

For a finite set S, we write $s \leftarrow S$ to denote that s is picked uniformly from finite set S. Then, we use $|S|$ to denote the size of S. Let $\approx_s$ stand for two distributions being statistically indistinguishable, and $\approx_c$ denote two distributions being computationally indistinguishable. We use lower-case boldface to denote vectors (e.g., $\mathbf{a}$) and upper-case boldface to denote matrices (e.g. $\mathbf{M}$), and use "$\|$" to denote vector/matrix concatenation (e.g. $\mathbf{A}\|\mathbf{B}$).

3.1 Prime-Order Bilinear Groups

A generator $\mathcal{G}$ takes as input a security parameter 1^λ and outputs a description $\mathbb{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, where p is a prime, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate bilinear map. Group

operations in $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ and bilinear map e are computable in deterministic polynomial time in λ. Let $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $g_T = e(g_1, g_2) \in \mathbb{G}_T$ be the respective generators, we employ *implicit representation* of group elements: for a matrix $\mathbf{M}$ over $\mathbb{Z}_p$, we define $[\mathbf{M}]_s = g_s^{\mathbf{M}}, \forall s \in \{1, 2, T\}$, where exponentiation is carried out component-wise. Given $[\mathbf{A}]_1, [\mathbf{B}]_2$ where $\mathbf{A}$ and $\mathbf{B}$ have proper sizes, we let $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{AB}]_T$. We review *matrix Diffie-Hellman (MDDH) assumption*, which is implied by k-Lin [EHK+13].

Assumption 1 ((k, ℓ, d)-MDDH over $\mathbb{G}_s$, $s \in \{1, 2\}$) *Let $k, \ell, d \in \mathbb{N}$ with $k < \ell$. We say that the (k, ℓ, d)-MDDH assumption holds in $\mathbb{G}_s$ if for all PPT adversaries $\mathcal{A}$, the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{MDDH}_{\mathcal{A},s,k,\ell,d}(\lambda) = \left| \Pr[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_s, [\mathbf{SM}]_s) = 1] - \Pr[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_s, [\mathbf{U}]_s) = 1] \right|$$

where $\mathbb{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$, $\mathbf{M} \leftarrow \mathbb{Z}_p^{k\times\ell}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{d\times k}$ and $\mathbf{U} \leftarrow \mathbb{Z}_p^{d\times\ell}$.

3.2 Slotted Registered Attribute-Based Signature

Algorithm. A slotted registered attribute-based signature (slotted registered ABS) for predicate $P : X \times Y \to \{0, 1\}$ consists of the following six efficient algorithms:

- $\mathsf{Setup}(1^\lambda, P, 1^L) \to \mathsf{crs}$: It takes as input the security parameter 1^λ, description of predicate P and the upper bound 1^L of the number of slots, outputs a common reference string crs.
- $\mathsf{Gen}(\mathsf{crs}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$: It takes as input crs and slot number $i \in [L]$, outputs key pair $(\mathsf{pk}_i, \mathsf{sk}_i)$.
- $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i) \to 0/1$: It takes as input crs, i, pk_i and outputs a bit indicating whether pk_i is valid.
- $\mathsf{Agg}(\mathsf{crs}, \{\mathsf{pk}_i, y_i\}_{i\in L}) \to (\mathsf{mpk}, (\mathsf{hk}_j)_{j\in[L]})$: It takes as input crs and a series of pk_i with $y_i \in Y$ for all $i \in [L]$, outputs master public key mpk and a series of helper keys hk_j for all $j \in [L]$. This algorithm is deterministic.
- $\mathsf{Sig}(\mathsf{hk}, \mathsf{sk}, x, \mathsf{m}) \to \sigma$: It takes as input hk, sk, $x \in X$ and message m, outputs a signature σ.
- $\mathsf{Ver}(\mathsf{mpk}, \sigma, x, \mathsf{m}) \to 0/1$: It takes as input hk, σ, $x \in X$, m and outputs a bit indicating whether σ is valid.

Completeness. For all $\lambda, L \in \mathbb{N}$, all P, and all $i \in [L]$, we have

$$\Pr[\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i) = 1 | \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, P, 1^L); (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(\mathsf{crs}, i)] = 1.$$

Correctness. For all $\lambda, L \in \mathbb{N}$, all P, and all $i \in [L]$, all $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, P, 1^L)$, all $(\mathsf{pk}_{i^*}, \mathsf{sk}_{i^*}) \leftarrow \mathsf{Gen}(\mathsf{crs}, i^*)$, all $\{\mathsf{pk}_i\}_{i\in[L]\setminus\{i^*\}}$ such that $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i) = 1$, all $x \in X$ and $y_1, \ldots, y_L \in Y$ such that $P(x, y_{i^*}) = 1$, and all m, we have

$$\Pr\left[\mathsf{Ver}(\mathsf{mpk}, \sigma, x, \mathsf{m}) = 1 \;\middle|\; \begin{array}{l}(\mathsf{mpk}, (\mathsf{hk}_j)_{j\in[L]}) \leftarrow \mathsf{Agg}(\mathsf{crs}, (\mathsf{pk}_i, y_i)_{i\in[L]}); \\ \sigma \leftarrow \mathsf{Sig}(\mathsf{hk}_{i^*}, \mathsf{sk}_{i^*}, x, \mathsf{m})\end{array}\right] = 1.$$

Compactness. For all $\lambda, L \in \mathbb{N}$, all P, and all $i \in [L]$, it holds that $|\mathsf{mpk}| = \mathsf{poly}(\lambda, P, \log L)$ and $|\mathsf{hk}_i| = \mathsf{poly}(\lambda, P, \log L)$.

Unforgeability. For any group of colluding signers, it is impossible to generate a valid signature on any message under any signing policy. Concretely, for all $\lambda \in \mathbb{N}$ and all efficient adversaries $\mathcal{A}$, the advantage

$$\left| \Pr\left[\mathsf{Ver}(\mathsf{mpk}, \sigma^*, x^*, \mathsf{m}^*) = 1 \;\middle|\; \begin{array}{l} L \leftarrow \mathcal{A}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, P, 1^L) \\ \{\mathsf{pk}_i^*, y_i^*\}_{i\in[L]} \leftarrow \mathcal{A}^{\mathsf{OGen}(\cdot),\mathsf{OCor}(\cdot)}(\mathsf{crs}) \\ (\mathsf{mpk}, \{\mathsf{hk}_j\}_{j\in[L]}) \leftarrow \mathsf{Agg}(\mathsf{crs}, \{\mathsf{pk}_i^*, y_i^*\}_{i\in L}) \\ (i^*, x^*, \mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OSig}(\cdot)}(\mathsf{mpk}, \{\mathsf{hk}_j\}_{j\in[L]}) \end{array}\right] - \frac{1}{2} \right|$$

is negligible in λ, where oracles OGen, OCor and OSig work with initial setting $\{\mathcal{D}_i = \emptyset\}_{i\in[L]}, C = \emptyset$ and $\mathcal{S} = \emptyset$ as follows:

- $\mathsf{OGen}(i)$: run $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(\mathsf{crs}, i)$, set $\mathcal{D}_i[\mathsf{pk}_i] = \mathsf{sk}_i$ and return pk_i.

– $\mathsf{OCor}(i, \mathsf{pk})$: return $\mathcal{D}_i[\mathsf{pk}]$ and update $C = C \cup \{(i, \mathsf{pk})\}$.
– $\mathsf{OSig}(i, x, \mathsf{m})$: Return $\mathsf{Sig}(\mathsf{hk}_i, \mathcal{D}_i[\mathsf{pk}_i^*], x, \mathsf{m})$ and update $\mathcal{S} = \mathcal{S} \cup \{(i, x, \mathsf{m})\}$.

and for all $i \in [L]$, we require that $\mathcal{D}_i[\mathsf{pk}_i^*] \neq \perp$. For each query (i, x, m) to OSig, we have $P(x, y_i^*) = 1$. Besides, For the challenge $(i^*, x^*, \mathsf{m}^*, \sigma^*)$,

– it holds that $(i^*, x^*, \mathsf{m}^*) \notin \mathcal{S}$;
– if $(i^*, \mathsf{pk}_{i^*}^*) \in C$, it holds that $P(x^*, y_i^*) = 0$ for all $(i, \mathsf{pk}_i^*) \in C$.

Notice that, in the unforgeability model of slotted registered ABS, we consider both honest and corrupt case. On the other hand, the notion of our unforgeability is somewhat different from that in classical ABS, since the adversary is allowed to query signature for (x^*, m^*) on all slots except for challenge slot i^* in honest case. Also, if we consider the anonymity of registered ABS, i^* should be removed and then our unforgeability model would follow the unforgeability of classical ABS in similar sense.

3.3 Registered Attribute-Based Signature

Algorithms. A *registered attribute-based signature* for predicate $P : X \times Y \to \{0, 1\}$ consists of six algorithms:

– $\mathsf{Setup}(1^\lambda, P) \to \mathsf{crs}$: It takes as input the security parameter 1^λ, description of predicate P, outputs a common reference string crs.
– $\mathsf{Gen}(\mathsf{crs}, \mathsf{aux}) \to (\mathsf{pk}, \mathsf{sk})$: It takes as input crs and the public state aux, outputs key pair $(\mathsf{pk}, \mathsf{sk})$.
– $\mathsf{Reg}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}, y) \to (\mathsf{mpk}, \mathsf{aux}')$: It takes as input crs, aux, and pk along with $y \in Y$, outputs master public key mpk and updated state aux′.
– $\mathsf{Upd}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}) \to \mathsf{hk}$: It takes as input crs, aux, pk, outputs a helper key hk.
– $\mathsf{Sig}(\mathsf{mpk}, \mathsf{hk}, \mathsf{sk}, x, \mathsf{m}) \to \sigma/\mathtt{getupd}$: It takes as input mpk, hk, sk, $x \in X$ and message m, outputs a signature σ or a special symbol `getupd` to indicate that an updated helper key is need to generate the signature.
– $\mathsf{Ver}(\mathsf{mpk}, \sigma, x, \mathsf{m}) \to 0/1$: It takes as input mpk, σ, x, m and outputs 1 if σ is valid; otherwise, output 0.

Correctness. For all stateful adversary $\mathcal{A}$, the following advantage function is negligible in λ:

$$\Pr[b = 1 | \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, P); b = 0; \mathcal{A}^{\mathsf{ORegNT}(\cdot,\cdot),\mathsf{ORegT}(\cdot),\mathsf{OSig}(\cdot,\cdot),\mathsf{OVer}(\cdot,\cdot)}(\mathsf{crs})]$$

where the oracles work as follows with initial setting $\mathsf{aux} = \perp$, $\mathcal{S} = \emptyset$, $\mathcal{R} = \emptyset$ and $t = \perp$:

– $\mathsf{ORegNT}(\mathsf{pk}, y)$: run $(\mathsf{mpk}, \mathsf{aux}') \leftarrow \mathsf{Reg}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}, y)$, update $\mathsf{aux} = \mathsf{aux}'$, append $(\mathsf{mpk}, \mathsf{aux})$ to $\mathcal{R}$ and return $(|\mathcal{R}|, \mathsf{mpk}, \mathsf{aux})$;
– $\mathsf{ORegT}(y^*)$: run $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow \mathsf{Gen}(\mathsf{crs}, \mathsf{aux})$, $(\mathsf{mpk}, \mathsf{aux}') \leftarrow \mathsf{Reg}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}^*, y^*)$, update $\mathsf{aux} = \mathsf{aux}'$, compute $\mathsf{hk}^* \leftarrow \mathsf{Upd}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}^*)$, append $(\mathsf{mpk}, \mathsf{aux})$ to $\mathcal{R}$, return $(t = |\mathcal{R}|, \mathsf{mpk}, \mathsf{aux}, \mathsf{pk}^*, \mathsf{sk}^*, \mathsf{hk}^*)$;
– $\mathsf{OSig}(i, x, \mathsf{m})$: let $\mathcal{R}[i] = (\mathsf{mpk}_i, \cdot)$ and run $\sigma \leftarrow \mathsf{Sig}(\mathsf{mpk}_i, \mathsf{hk}^*, \mathsf{sk}^*, x, \mathsf{m})$; If $\sigma = \mathtt{getupd}$, run $\mathsf{hk}^* \leftarrow \mathsf{Upd}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}^*)$ and recompute $\sigma \leftarrow \mathsf{Sig}(\mathsf{mpk}_i, \mathsf{hk}^*, \mathsf{sk}^*, x, \mathsf{m})$. Then append (x, m, σ) to $\mathcal{S}$ and return $(|\mathcal{S}|, \sigma)$;
– $\mathsf{OVer}(i, j)$: let $\mathcal{R}[i] = (\mathsf{mpk}_i, \cdot)$ and $\mathcal{S}[j] = (x_j, \mathsf{m}_j, \sigma_j)$, compute $b_j \leftarrow \mathsf{Ver}(\mathsf{mpk}_i, \sigma_j, x_j, \mathsf{m}_j)$. If $b_j = 0$, set $b = 0$.

with the following restrictions:

– there exists one query to ORegT;
– for query $(i, x, \cdot)$ to OSig, it holds that $\mathcal{R}[i] \neq \perp$ and $P(x, y^*) = 1$;
– for query (i, j) to OVer, it holds that $t \leq i, \mathcal{R}[i] \neq \perp$ and $\mathcal{S}[j] \neq \perp$.

Compactness. Let $\mathcal{R}$ be defined as before. *Compactness* means that

$$|\mathsf{mpk}_i| = \mathrm{poly}(\lambda, P, \log i), \quad |\mathsf{hk}^*| = \mathrm{poly}(\lambda, P, \log |\mathcal{R}|);$$

where we let $\mathcal{R}[i] = (\mathsf{mpk}_i, \cdot)$ for all $i \in [|\mathcal{R}|]$.

Update Efficiency. It means that the number of invocations of Upd in OSig is at most $O(\log |\mathcal{R}|)$ and each invocation runs in $\mathrm{poly}(\log |\mathcal{R}|)$ time (in RAM model).

Unforgeability. For all stateful adversary $\mathcal{A}$, the advantage

$$\left| \Pr\left[\mathsf{Ver}(\mathsf{mpk}, \sigma^*, x^*, \mathsf{m}^*) = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, P); \\ (i^*, x^*, \mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}(\mathsf{crs}); \end{array} \right] - \frac{1}{2} \right|$$

is negligible in λ, where $\mathcal{A}$ has access to oracles $\mathsf{ORegHK}(\cdot)$, $\mathsf{OCorHK}(\cdot)$ and $\mathsf{OSig}(\cdot,\cdot,\cdot)$. These oracles work with initially setting $\mathsf{aux}, \mathsf{mpk} = \perp$, $\mathcal{R} = \emptyset$, $C = \emptyset$, $\mathcal{S} = \emptyset$ and a dictionary $\mathcal{K}$ with $\mathcal{K}[\mathsf{pk}] =$ for all possible pk:

- $\mathsf{ORegHK}(y)$: run $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{crs}, \mathsf{aux})$ and $(\mathsf{mpk}', \mathsf{aux}') \leftarrow \mathsf{Reg}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}, y)$, update $\mathsf{mpk} = \mathsf{mpk}', \mathsf{aux} = \mathsf{aux}'$, $\mathcal{K}[\mathsf{pk}] = \mathcal{K}[\mathsf{pk}] \cup \{y\}$, append $(\mathsf{pk}, \mathsf{sk})$ to $\mathcal{R}$ and return $(|\mathcal{R}|, \mathsf{mpk}, \mathsf{aux}, \mathsf{pk})$;
- $\mathsf{OCor}(i)$: let $\mathcal{R}[i] = (\mathsf{pk}, \mathsf{sk})$, append pk to C and return sk;
- $\mathsf{OSig}(i, x, \mathsf{m})$: let $\mathcal{R}[i] = (\mathsf{pk}, \mathsf{sk})$, compute $\mathsf{hk} \leftarrow \mathsf{Upd}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk})$ and run $\sigma \leftarrow \mathsf{Sig}(\mathsf{mpk}, \mathsf{hk}, \mathsf{sk}, x, \mathsf{m})$. Append (i, x, m) to $\mathcal{S}$ and return σ.

with the following restrictions:

- for query (i) to OCor or (i, x, m) to OSig, it holds that $\mathcal{R}[i] \neq \perp$. Besides, $\mathcal{R}[i^*] \neq \perp$;
- Let $\mathcal{R}[i] = (\mathsf{pk}, \mathsf{sk})$ and $\mathcal{K}[\mathsf{pk}] = y$, then it holds that $P(y, x) = 1$;
- Let $\mathcal{R}[i^*] = (\mathsf{pk}^*, \mathsf{sk}^*)$,
  - it holds that $(i^*, x^*, \mathsf{m}^*) \notin \mathcal{S}$;
  - if $\mathsf{pk}^* \in C$, it holds that $P(x^*, \mathcal{K}[\mathsf{pk}_i]) = 0$ for all $(\mathsf{pk}_i, \mathsf{sk}_i) \in \mathcal{R}$ such that $\mathsf{pk}_i \in C$;

3.4 Predicate Encodings

We review the notion of predicate encoding [Wee14,CGW15,ZZGQ23]; for simplicity, we use the formulation in [ABS17,ACGU20]. A predicate $P : X \times Y \to \{0, 1\}$ has a (n, n_c, n_k)-predicate encoding if: For all $x \in X$, $y \in Y$, there exist $\mathbf{C}_x \in \mathbb{Z}_p^{n \times n_c}$, $\mathbf{K}_y \in \mathbb{Z}_p^{n \times n_k}$, $\mathbf{a}_y \in \mathbb{Z}_p^{1 \times n_k}$, $\mathbf{d}_{x,y} \in \mathbb{Z}_p^{1 \times (n_k + n_c)}$ such that, letting

$$\mathbf{M}_{x,y} = \begin{pmatrix} \mathbf{a}_y & \mathbf{0}_{n_c} \\ \mathbf{K}_y & \mathbf{C}_x \end{pmatrix} \in \mathbb{Z}_p^{(1+n) \times (n_k + n_c)}$$

we have

- **correctness:** for $x \in X$ and $y \in Y$ such that $P(x, y) = 1$:

$$\mathbf{M}_{x,y} \mathbf{d}_{x,y}^\top = \mathbf{e}_1^\top;$$

- **security:** for $x \in X$ and $y \in Y$ such that $P(x, y) = 0$ and for all $\alpha \in \mathbb{Z}_p$:

$$\{x, y, \alpha, (\alpha \| \mathbf{w}) \mathbf{M}_{x,y}\} \approx_s \{x, y, \alpha, (0 \| \mathbf{w}) \mathbf{M}_{x,y}\}, \quad \mathbf{w} \leftarrow \mathbb{Z}_p^n.$$

Also, we require that (1) given P, one can efficiently determine n, n_c, n_k; (2) given x, one can efficiently compute $\mathbf{C}_x$; (3) given y, one can efficiently compute $\mathbf{K}_y$ and $\mathbf{a}_y$; (4) given both x and y, one can efficiently compute $\mathbf{d}_{x,y}$.

4 Slotted Registered ABS

In this section, we will propose a slotted registered ABS via predicate encoding under the matrix decisional Diffie-Hellman (MDDH) assumption.

4.1 Scheme

Our slotted registered ABS scheme from predicate encoding over prime-order bilinear group works as follows:

– $\mathsf{Setup}(1^\lambda, P, 1^L)$: Run $\mathbb{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$ and select a collusion-resistant hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_p$. Sample

$$\mathbf{A} \leftarrow \mathbb{Z}_p^{k\times(k+1)}, \mathbf{B} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{k} \leftarrow \mathbb{Z}_p^{1\times(k+1)}$$

and compute parameter (n, n_c, n_k) from P. For all $i \in [L]$, sample

$$\mathbf{D}_i \leftarrow \mathbb{Z}_p^{k\times k}, \mathbf{V}_i \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}, \mathbf{W}_i \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)n}, \mathbf{r}_i \leftarrow \mathbb{Z}_p^{1\times k}.$$

Set $\mathbf{B}_i = \mathbf{B}\mathbf{D}_i$ for each $i \in [L]$ and output

$$\mathsf{crs} = \begin{pmatrix} [\mathbf{A}]_1, \{[\mathbf{A}\mathbf{V}_i, \mathbf{A}\mathbf{W}_i]_1, [\mathbf{B}\mathbf{r}_i^\top, \mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \mathbf{k}^\top, \mathbf{B}_i]_2\}_{i\in[L]}, \\ \{[\mathbf{V}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{W}_i(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_j^\top)]_2\}_{j\in[L], i\in[L]\setminus\{j\}}, [\mathbf{A}\mathbf{k}^\top]_T, \mathsf{H} \end{pmatrix}.$$

– $\mathsf{Gen}(\mathsf{crs}, i)$: Sample $\mathbf{U}_i, \mathbf{Q}_i, \mathbf{T}_i \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$. Output

$$\mathsf{pk}_i = ([\mathbf{A}\mathbf{U}_i, \mathbf{A}\mathbf{Q}_i, \mathbf{A}\mathbf{T}_i]_1, \{[\mathbf{U}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{Q}_i\mathbf{B}_j, \mathbf{T}_i\mathbf{B}_j]_2\}_{j\in[L]\setminus\{i\}})$$

and $\mathsf{sk}_i = (\mathbf{U}_i, \mathbf{Q}_i, \mathbf{T}_i)$.

– $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$: Parse the public key pk_i as follows $([\mathbf{A}\mathbf{U}_i, \mathbf{A}\mathbf{Q}_i, \mathbf{A}\mathbf{T}_i]_1, \{[\mathbf{U}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{Q}_i\mathbf{B}_j, \mathbf{T}_i\mathbf{B}_j]_2\}_{j\in[L]\setminus\{i\}})$. For each $j \in [L] \setminus \{i\}$, check

$$e([\mathbf{A}]_1, [\mathbf{U}_i\mathbf{B}\mathbf{r}_j^\top]_2) \stackrel{?}{=} e([\mathbf{A}\mathbf{U}_i]_1, [\mathbf{B}\mathbf{r}_j^\top]_2),$$
$$e([\mathbf{A}]_1, [\mathbf{Q}_i\mathbf{B}_j]_2) \stackrel{?}{=} e([\mathbf{A}\mathbf{Q}_i]_1, [\mathbf{B}_j]_2),$$
$$e([\mathbf{A}]_1, [\mathbf{T}_i\mathbf{B}_j]_2) \stackrel{?}{=} e([\mathbf{A}\mathbf{T}_i]_1, [\mathbf{B}_j]_2).$$

If the above checks pass, output 1; otherwise, output 0.

– $\mathsf{Agg}(\mathsf{crs}, \{\mathsf{pk}_i, y_i\}_{i\in L})$: For all $i \in [L]$, parse $\mathsf{pk}_i = ([\mathbf{A}\mathbf{U}_i, \mathbf{A}\mathbf{Q}_i, \mathbf{A}\mathbf{T}_i]_1, \{[\mathbf{U}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{Q}_i\mathbf{B}_j, \mathbf{T}_i\mathbf{B}_j]_2\}_{j\in[L]\setminus\{i\}})$ and compute $\mathbf{K}_{y_i}$ from y_i. Output

$$\mathsf{mpk} = \begin{pmatrix} [\mathbf{A}]_1, [\mathbf{A}\mathbf{k}^\top]_T, \mathsf{H}, \left[\sum_{j\in[L]} \mathbf{A}\mathbf{Q}_j\right]_1, \left[\sum_{j\in[L]} \mathbf{A}\mathbf{T}_j\right]_1, \left[\sum_{j\in[L]} \mathbf{A}\mathbf{W}_j\right]_1, \\ \left[\sum_{j\in[L]} \left((\mathbf{A}\mathbf{V}_j + \mathbf{A}\mathbf{U}_j)(\mathbf{a}_{y_j} \otimes \mathbf{I}_{k+1}) + \mathbf{A}\mathbf{W}_j(\mathbf{K}_{y_j} \otimes \mathbf{I}_{k+1})\right)\right]_1 \end{pmatrix}$$

and for all $i \in [L]$, the hk_i is that

$$\begin{pmatrix} \mathsf{H}, [\mathbf{B}_i]_2, [\mathbf{B}\mathbf{r}_i^\top]_2, [\mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \mathbf{k}^\top]_2, \left[\sum_{j\in[L]\setminus\{i\}} \mathbf{Q}_j\mathbf{B}_i\right]_2, \left[\sum_{j\in[L]\setminus\{i\}} \mathbf{T}_j\mathbf{B}_i\right]_2, \\ \left[\sum_{j\in[L]\setminus\{i\}} \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_i^\top)\right]_2, \left[\sum_{j\in[L]\setminus\{i\}} (\mathbf{V}_j\mathbf{B}\mathbf{r}_i^\top + \mathbf{U}_j\mathbf{B}\mathbf{r}_i^\top)(\mathbf{I}_k \otimes \mathbf{a}_{y_j}) + \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_i^\top)(\mathbf{I}_k \otimes \mathbf{K}_{y_j})\right]_2 \end{pmatrix}.$$

– $\mathsf{Sig}(\mathsf{hk}_i, \mathsf{sk}_i, x, \mathsf{m})$: Sample $\mathbf{t} \leftarrow \mathbb{Z}_p^{1\times k}$, and compute $\mathbf{C}_x$. Compute $[\mathbf{k}_0^\top]_2 = [\mathbf{B}_i\mathbf{t}^\top]_2, [\mathbf{k}_1^\top]_2 = [\mathbf{B}\mathbf{r}_i^\top]_2$. Run $m \leftarrow \mathsf{H}(i, \mathsf{m}, x)$ and generate

$$\left(\begin{array}{l} \underbrace{[\mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \mathbf{k}^\top + \mathbf{U}_i\mathbf{B}\mathbf{r}_i^\top + (\mathbf{Q}_i\mathbf{B}_i\mathbf{t}^\top + m\cdot\mathbf{T}_i\mathbf{B}_i\mathbf{t}^\top)]_2}_{\mathbf{k}_2^\top}, \\ \underbrace{\left[\sum_{j\in[L]\setminus\{i\}} (\mathbf{V}_j\mathbf{B}\mathbf{r}_i^\top + \mathbf{U}_j\mathbf{B}\mathbf{r}_i^\top)\mathbf{a}_{y_j} + \mathbf{W}_j(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_i^\top)\mathbf{K}_{y_j}\right]_2}_{\mathbf{K}_3}, \\ \underbrace{\left[\sum_{j\in[L]\setminus\{i\}} \mathbf{W}_j(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_i^\top)\mathbf{C}_x\right]_2}_{\mathbf{K}_4}, \underbrace{\left[\sum_{j\in[L]\setminus\{i\}} (\mathbf{Q}_j\mathbf{B}_i\mathbf{t}^\top + m\cdot\mathbf{T}_j\mathbf{B}_i\mathbf{t}^\top)\right]}_{\mathbf{k}_5^\top} \end{array}\right)$$

Output signature $\sigma_{i,x,\mathsf{m}} = ([\mathbf{k}_0^\top]_2, [\mathbf{k}_1^\top]_2, [\mathbf{k}_2^\top]_2, [\mathbf{K}_3]_2, [\mathbf{K}_4]_2, [\mathbf{k}_5^\top]_2)$.

– $\mathsf{Ver}(\mathsf{mpk}, \sigma_{i^*,x,\mathsf{m}}, x, \mathsf{m})$: Parse mpk as

$$\left(\begin{array}{l} [\mathbf{A}]_1, [\mathbf{A}\mathbf{k}^\top]_T, \mathsf{H}, \left[\sum_{j\in[L]} \mathbf{A}\mathbf{Q}_j\right]_1, \left[\sum_{j\in[L]} \mathbf{A}\mathbf{T}_j\right]_1, \left[\sum_{j\in[L]} \mathbf{A}\mathbf{W}_j\right]_1, \\ \left[\sum_{j\in[L]} (\mathbf{A}\mathbf{V}_j + \mathbf{A}\mathbf{U}_j)(\mathbf{a}_{y_j}\otimes\mathbf{I}_{k+1}) + \mathbf{A}\mathbf{W}_j(\mathbf{K}_{y_j}\otimes\mathbf{I}_{k+1})\right]_1 \end{array}\right)$$

and signature $\sigma_{i^*,x,\mathsf{m}} = ([\mathbf{k}_0^\top]_2, [\mathbf{k}_1^\top]_2, [\mathbf{k}_2^\top]_2, [\mathbf{K}_3]_2, [\mathbf{K}_4]_2, [\mathbf{k}_5^\top]_2)$. Then compute $\mathbf{C}_x$ and $\mathbf{d}_{x,y_{i^*}}$ from x and y_{i^*}. Run $m \leftarrow \mathsf{H}(i^*, \mathsf{m}, x)$ and compute

$$\left(\begin{array}{l} [\underbrace{\mathbf{s}\mathbf{A}}_{\mathbf{v}_0}]_1, \underbrace{\left[\sum_{j\in[L]} (\mathbf{s}\mathbf{A}\mathbf{Q}_j + m\cdot\mathbf{s}\mathbf{A}\mathbf{T}_j)\right]_1}_{\mathbf{v}_1}, \underbrace{\left[\sum_{j\in[L]} \mathbf{s}\mathbf{A}\mathbf{W}_j(\mathbf{C}_x\otimes\mathbf{I}_{k+1})\right]_1}_{\mathbf{v}_2} \\ \underbrace{\left[\sum_{j\in[L]} (\mathbf{s}\mathbf{A}\mathbf{V}_j + \mathbf{s}\mathbf{A}\mathbf{U}_j)(\mathbf{a}_{y_j}\otimes\mathbf{I}_{k+1}) + \mathbf{s}\mathbf{A}\mathbf{W}_j(\mathbf{K}_{y_j}\otimes\mathbf{I}_{k+1})\right]_1}_{\mathbf{v}_3}, [\underbrace{\mathbf{s}\mathbf{A}\mathbf{k}^\top}_{v_4}]_T \end{array}\right)$$

where $\mathbf{s} \leftarrow \mathbb{Z}_p^{1\times k}$. Set the verification text $v_{i^*,x,\mathsf{m}} = ([\mathbf{v}_0]_1, [\mathbf{v}_1]_1, [\mathbf{v}_2]_1, [\mathbf{v}_3]_1, [v_4]_T)$. Recover

$$[\mathbf{z}]_T = e([\mathbf{v}_3\|\mathbf{v}_2]_1, [\mathbf{I}_{n_k+n_c}\otimes\mathbf{k}_1^\top]_2), [\mathbf{z}_2]_T = e([\mathbf{v}_0]_1, [\mathbf{K}_3\|\mathbf{K}_4]_2)$$
$$[z_3]_T = e([\mathbf{v}_0]_1, [\mathbf{k}_2^\top]_2), [z_4]_T = e([\mathbf{v}_1]_1, [\mathbf{k}_0^\top]_2), [z_5]_T = e([\mathbf{v}_0]_1, [\mathbf{k}_5^\top]_2),$$
$$[z_6]_T = [z_3 - z_4 + z_5]_T, [\mathbf{z}_7]_T = [(\mathbf{z}_1 - \mathbf{z}_2)\mathbf{d}_{x,y_{i^*}}^\top - z_6]_T$$

and check $[z_7]_T^{-1} \stackrel{?}{=} [v_4]_T$. If the above check passes, output 1; otherwise, output 0.

Correctness. For all $\lambda, L \in \mathbb{N}$, all P, all $i^* \in [L]$, all $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, P, 1^L)$, all $(\mathsf{pk}_{i^*}, \mathsf{sk}_{i^*}) \leftarrow \mathsf{Gen}(\mathsf{crs}, i^*)$, all $\{\mathsf{pk}_i\}_{i\in[L]\setminus\{i^*\}}$ such that $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i) = 1$, for all $y_1, \ldots, y_L \in Y$ and $x \in X$ with $P(x, y_{i^*}) = 1$ and all m, we have: $\sigma_{i^*,x,\mathsf{m}} = ([\mathbf{k}_0^\top]_2, [\mathbf{k}_1^\top]_2, [\mathbf{k}_2^\top]_2, [\mathbf{K}_3]_2, [\mathbf{K}_4]_2, [\mathbf{k}_5^\top]_2)$ and $v_{i^*,x,\mathsf{m}} = ([\mathbf{v}_0]_1, [\mathbf{v}_1]_1, [\mathbf{v}_2]_1, [\mathbf{v}_3]_1, [v_4]_1)$. We employ the predicate encoding as defined in section 3.4, namely

$$\mathbf{M}_{x,y_i} = \begin{pmatrix} \mathbf{a}_{y_i} & \mathbf{0}_{n_c} \\ \mathbf{K}_{y_i} & \mathbf{C}_x \end{pmatrix}, \quad \forall i \in [L].$$

We obtain

$$\begin{aligned}
\mathbf{z}_1 &= \sum_{i\in[L]} (\mathbf{sAV}_i + \mathbf{sAU}_i \| \mathbf{sAW}_i)(\mathbf{M}_{x,y_i} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{n_k+n_c} \otimes \mathbf{Br}_{i^*}^\top) \\
&= \sum_{i\in[L]} (\mathbf{sAV}_i + \mathbf{sAU}_i \| \mathbf{sAW}_i)(\mathbf{I}_{1+n} \otimes \mathbf{Br}_{i^*}^\top)\mathbf{M}_{x,y_i} \\
&= \sum_{i\in[L]} (\mathbf{sAV}_i\mathbf{Br}_{i^*}^\top + \mathbf{sAU}_i\mathbf{Br}_{i^*}^\top \| \mathbf{sAW}_i(\mathbf{I}_n \otimes \mathbf{Br}_{i^*}^\top))\mathbf{M}_{x,y_i} \\
\mathbf{z}_2 &= \sum_{i\in[L]\setminus\{i^*\}} (\mathbf{sAV}_i\mathbf{Br}_{i^*}^\top + \mathbf{sAU}_i\mathbf{Br}_{i^*}^\top \| \mathbf{sAW}_i(\mathbf{I}_n \otimes \mathbf{Br}_{i^*}^\top))\mathbf{M}_{x,y_i} \qquad (2) \\
z_3 &= \mathbf{sAV}_{i^*}\mathbf{Br}_{i^*}^\top + \mathbf{sAk}^\top + \mathbf{sAU}_{i^*}\mathbf{Br}_{i^*}^\top + (\mathbf{sAQ}_{i^*}\mathbf{B}_{i^*}\mathbf{t}^\top + m \cdot \mathbf{sAT}_{i^*}\mathbf{B}_{i^*}\mathbf{t}^\top) \\
z_4 &= \sum_{j\in[L]} (\mathbf{sAQ}_j\mathbf{B}_{i^*}\mathbf{t}^\top + m \cdot \mathbf{sAT}_j\mathbf{B}_{i^*}\mathbf{t}^\top) \\
z_5 &= \sum_{j\in[L]\setminus\{i^*\}} (\mathbf{sAQ}_j\mathbf{B}_{i^*}\mathbf{t}^\top + m \cdot \mathbf{sAT}_j\mathbf{B}_{i^*}\mathbf{t}^\top)
\end{aligned}$$

and then

$$\begin{aligned}
z_6 &= z_3 - (z_4 - z_5) = \mathbf{sAV}_{i^*}\mathbf{Br}_{i^*}^\top + \mathbf{sAk}^\top + \mathbf{sAU}_{i^*}\mathbf{Br}_{i^*}^\top \\
z_7 &= (\mathbf{z}_1 - \mathbf{z}_2)\mathbf{d}_{x,y_{i^*}}^\top - z_6 = -\mathbf{sAk}^\top
\end{aligned}$$

where

$$z_4 - z_5 = \mathbf{sAQ}_{i^*}\mathbf{B}_{i^*}\mathbf{t}^\top + m \cdot \mathbf{sAT}_{i^*}\mathbf{B}_{i^*}\mathbf{t}^\top.$$

Finally, we have $[z_7]_T^{-1} = [v_4]_T$. Notice that equality (2) follows from the property of tensor product: $(\mathbf{M}\otimes\mathbf{I})(\mathbf{I}\otimes\mathbf{a}^\top) = \mathbf{M}\otimes\mathbf{a}^\top = (\mathbf{I}\otimes\mathbf{a}^\top)\mathbf{M}$ for matrices of proper size; the computation of z_7 follows from the correctness of predicate encoding. This proves the correctness.

4.2 Security

Theorem 1. *The proposed slotted registered ABS scheme is unforgeable under MDDH assumption and collision-resistant hash functions.*

Game Sequence. We prove Theorem 1 via the following game sequences. Let L be the number of slots and i^* be the challenge slot, (x^*, m^*) be the challenge attribute and message pair; $(\mathsf{pk}_i^*, y_i^*)_{i\in[L]}$ be the challenge public keys and challenge "policy" to be registered. For all $i \in [L]$, $\mathcal{D}_i = \{\mathsf{pk}_i : \mathcal{D}_i[\mathsf{pk}_i] = \mathsf{sk}_i \neq \perp\}$ stores the response to $\mathsf{OGen}(i)$; $C_i = \{\mathsf{pk}_i : (i, \mathsf{pk}_i) \in C\}$ stores the response to $\mathsf{OCor}(i, \cdot)$. Define $Q = \sum_{i\in[L]} Q_i$ and $\sigma_{i,\kappa}$ as the κ-th($\kappa \in [Q_i]$) signature query's result in slot i, where Q_i denotes the number of signature queries in slot i.

- G_0 : Real Game. Recall that:
  - the common reference string is that

$$\mathsf{crs} = \begin{pmatrix} [\mathbf{A}]_1, \{[\mathbf{AV}_i, \mathbf{AW}_i]_1, [\mathbf{Br}_i^\top, \mathbf{V}_i\mathbf{Br}_i^\top + \mathbf{k}^\top, \mathbf{B}_i]_2\}_{i\in[L]}, \\ \{[\mathbf{V}_i\mathbf{Br}_j^\top, \mathbf{W}_i(\mathbf{I}_n \otimes \mathbf{Br}_j^\top)]_2\}_{j\in[L],i\in[L]\setminus\{j\}}, [\mathbf{Ak}^\top]_T, \mathsf{H} \end{pmatrix}.$$

  - For each $i \in [L]$, each public key $\mathsf{pk}_i \in \mathcal{D}_i$ is that

$$\mathsf{pk}_i = ([\mathbf{AU}_i, \mathbf{AQ}_i, \mathbf{AT}_i]_1, \{[\mathbf{U}_i\mathbf{Br}_j^\top, \mathbf{Q}_i\mathbf{B}_j, \mathbf{T}_i\mathbf{B}_j]_2\}_{j\in[L]\setminus\{i\}}).$$

    where the corresponding secret key is that $\mathsf{sk}_i = (\mathbf{U}_i, \mathbf{Q}_i, \mathbf{T}_i)$.

- For each $i \in [L]$ and each $\kappa \in [Q_i]$, the κ-th query to $\mathsf{OSig}(i, x, \mathsf{m})$ will output the result $\sigma_{i,\kappa}$ as

$$\left(\begin{array}{l} [\underbrace{\mathbf{B}_i\mathbf{t}^\top}_{\mathbf{k}_0^\top}]_2, [\underbrace{\mathbf{B}\mathbf{r}_i^\top}_{\mathbf{k}_1^\top}]_2, [\underbrace{\mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \mathbf{k}^\top + \mathbf{U}_i\mathbf{B}\mathbf{r}_i^\top + (\mathbf{Q}_i\mathbf{B}_i\mathbf{t}^\top + m \cdot \mathbf{T}_i\mathbf{B}_i\mathbf{t}^\top)}_{\mathbf{k}_2^\top}]_2 \\ \left[\underbrace{\sum_{j\in[L]\setminus\{i^*\}} (\mathbf{V}_j\mathbf{B}\mathbf{r}_i^\top + \mathbf{U}_j\mathbf{B}\mathbf{r}_i^\top)\mathbf{a}_{y_j} + \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_i^\top)\mathbf{K}_{y_j}}_{\mathbf{K}_3}\right]_2, \\ \left[\underbrace{\sum_{j\in[L]\setminus\{i\}} \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_i^\top)\mathbf{C}_x}_{\mathbf{K}_4}\right]_2, \left[\underbrace{\sum_{j\in[L]\setminus\{i\}} (\mathbf{Q}_j\mathbf{B}_i\mathbf{t}^\top + m \cdot \mathbf{T}_j\mathbf{B}_i\mathbf{t}^\top)}_{\mathbf{k}_5^\top}\right]_2 \end{array} \right)$$

  where $\mathbf{t} \leftarrow \mathbb{Z}_p^{1\times k}$ and $m_{i,\kappa} \leftarrow \mathsf{H}(i, \mathsf{m}, x)$.
- The challenge verification text is $v^*_{i^*,x^*,\mathsf{m}^*} = ([\mathbf{v}_0]_1, [\mathbf{v}_1]_1, [\mathbf{v}_2]_1, [\mathbf{v}_3]_1, [v_4]_T)$ where

$$\left(\begin{array}{l} [\underbrace{\mathbf{s}\mathbf{A}}_{\mathbf{v}_0}]_1, \left[\underbrace{\sum_{j\in[L]} (\mathbf{s}\mathbf{A}\mathbf{Q}_j + m \cdot \mathbf{s}\mathbf{A}\mathbf{T}_j)}_{\mathbf{v}_1}\right]_1, \left[\underbrace{\sum_{j\in[L]} \mathbf{s}\mathbf{A}\mathbf{W}_j(\mathbf{C}_x \otimes \mathbf{I}_{k+1})}_{\mathbf{v}_2}\right]_1 \\ \left[\underbrace{\sum_{j\in[L]} (\mathbf{s}\mathbf{A}\mathbf{V}_j + \mathbf{s}\mathbf{A}\mathbf{U}_j)(\mathbf{a}_{y_j} \otimes \mathbf{I}_{k+1}) + \mathbf{s}\mathbf{A}\mathbf{W}_j(\mathbf{K}_{y_j} \otimes \mathbf{I}_{k+1})}_{\mathbf{v}_3}\right]_1, [\underbrace{\mathbf{s}\mathbf{A}\mathbf{k}^\top}_{v_4}]_T \end{array} \right)$$

  notice that $\mathbf{s} \leftarrow \mathbb{Z}_p^{1\times k}$ and $m^* \leftarrow \mathsf{H}(i^*, \mathsf{m}^*, x^*)$.

– G_1 : Identical to G_0 except that we replace $\mathbf{s}\mathbf{A}$ in challenge verification text with $\mathbf{c} \leftarrow \mathbb{Z}_p^{1\times(k+1)}$. Then the challenge verification text is that

$$\left(\begin{array}{l} [\underbrace{\mathbf{c}}_{\mathbf{v}_0}]_1, \left[\underbrace{\sum_{j\in[L]} (\mathbf{c}\mathbf{Q}_j + m \cdot \mathbf{c}\mathbf{T}_j)}_{\mathbf{v}_1}\right]_1, \left[\underbrace{\sum_{j\in[L]} \mathbf{c}\mathbf{W}_j(\mathbf{C}_x \otimes \mathbf{I}_{k+1})}_{\mathbf{v}_2}\right]_1 \\ \left[\underbrace{\sum_{j\in[L]} (\mathbf{c}\mathbf{V}_j + \mathbf{c}\mathbf{U}_j)(\mathbf{a}_{y_j} \otimes \mathbf{I}_{k+1}) + \mathbf{c}\mathbf{W}_j(\mathbf{K}_{y_j} \otimes \mathbf{I}_{k+1})}_{\mathbf{v}_3}\right]_1, [\underbrace{\mathbf{c}\mathbf{k}^\top}_{v_4}]_T \end{array} \right).$$

  Observe that we have $\mathsf{G}_0 \approx_c \mathsf{G}_1$, which follows the MDDH assumption, ensuring that $([\mathbf{A}]_1, [\mathbf{s}\mathbf{A}]_1) \approx_c ([\mathbf{A}]_1, [\mathbf{c}]_1)$ where $\mathbf{A} \leftarrow \mathbb{Z}_p^{k\times(k+1)}$, $\mathbf{s} \leftarrow \mathbb{Z}_p^{1\times k}$ and $\mathbf{c} \leftarrow \mathbb{Z}_p^{1\times(k+1)}$. See Lemma 1 for more details.
– $\mathsf{G}_{2,\ell}(\ell \in [L])$: Identical to $\mathsf{G}_{2,\ell-1}$ except that we change crs into the following form:

$$\mathsf{crs} = \begin{pmatrix} [\mathbf{A}]_1, \{[\mathbf{A}\mathbf{V}_i, \mathbf{A}\mathbf{W}_i]_1, [\mathbf{B}\mathbf{r}_i^\top, \mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \alpha\mathbf{c}^\perp\mathbf{k}^\top, \mathbf{B}_i]_2\}_{i\in[L]}, \\ \{[\mathbf{V}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{W}_i(\mathbf{I}_n \otimes \mathbf{B}_j)]_2\}_{j\in[L], i\in[L]\setminus\{j\}}, [\mathbf{A}\mathbf{k}^\top]_T, \mathsf{H} \end{pmatrix}.$$

  where $\alpha_\ell \leftarrow \mathbb{Z}_p$ and $\mathbf{c}^\perp \leftarrow \mathbb{Z}_p^{2k+1}$ such that $\mathbf{A}\mathbf{c}^\perp = 0, \mathbf{c}\mathbf{c}^\perp = 1$. Note that $\mathsf{G}_{2,0}$ is identical to G_1; we have $\mathsf{G}_{2,\ell-1} \approx_c \mathsf{G}_{2,\ell}$, see section 4.3 for more details.

– G_3 : Identical to G_{2,Q_L} except that we replace the verification text into the following form:

$$\left(\begin{array}{l} [\underbrace{\mathbf{c}}_{\mathbf{v}_0}]_1, \left[\underbrace{\sum_{j\in[L]} (\mathbf{cQ}_j + m \cdot \mathbf{cT}_j)}_{\mathbf{v}_1} \right]_1, \left[\underbrace{\sum_{j\in[L]} \mathbf{cW}_j(\mathbf{C}_x \otimes \mathbf{I}_{k+1})}_{\mathbf{v}_2} \right]_1 \\ \left[\underbrace{\sum_{j\in[L]} (\mathbf{cV}_j + \mathbf{cU}_j)(\mathbf{a}_{y_j} \otimes \mathbf{I}_{k+1}) + \mathbf{cW}_j(\mathbf{K}_{y_j} \otimes \mathbf{I}_{k+1})}_{\mathbf{v}_3} \right]_1, \underbrace{g_T^*}_{v_4} \end{array} \right)$$

where g_T^* is sampled uniformly over G_T. We claim that $\mathsf{G}_{2,L} \approx_s \mathsf{G}_3$ which follows the following the statistical argument:

$$(\mathbf{Ak}^\top, \mathbf{k}^\top + \alpha \mathbf{c}^\perp, \mathbf{ck}^\top) \approx_s (\mathbf{Ak}^\top, \mathbf{k}^\top, \mathbf{ck}^\top - \alpha)$$

where $[\mathbf{ck}^\top - \alpha]_T$ is uniform, namely, g_T^*. See Lemma 4 for more details.

Observe that the advantage of $\mathcal{A}$ to forge a valid signature is negligible in G_3.

4.3 From $\mathsf{G}_{2,\ell-1}$ to $\mathsf{G}_{2,\ell}$

In this section, we prove $\mathsf{G}_{2,\ell-1} \approx_c \mathsf{G}_{2,\ell}$. Similar to [ZZGQ23], we consider the honest case and the corrupted case, respectively. For these two cases, we apply the following different strategies.

Honest Case. In this case, our proof must deal with both crs and signatures queried from $\mathcal{A}$. Here, only the challenger knows the secret key $\mathsf{sk}_\ell = (\mathbf{U}_\ell, \mathbf{Q}_\ell, \mathbf{T}_\ell)$ which is hidden from $\mathcal{A}$. Let Q_ℓ be the number of signatures queried by $\mathcal{A}$ on each slot $\ell \in \mathcal{D}_\ell \setminus C_\ell$, we use the following sub-sequence of games.

– $\mathsf{G}_{2,\ell-1,0}$: Identical to $\mathsf{G}_{2,\ell-1}$. Recall the crs is in the form

$$\mathsf{crs} = \left(\begin{array}{l} [\mathbf{Ak}^\top]_T, \mathsf{H}, [\mathbf{A}]_1, \{[\mathbf{AV}_i, \mathbf{AW}_i]_1, [\mathbf{B}_i]_2\}_{i\in[L]}, \\ \{[\mathbf{V}_i\mathbf{Br}_j^\top, \mathbf{W}_i(\mathbf{I}_n \otimes \mathbf{Br}_j^\top)]_2\}_{j\in[L], i\in[L]\setminus\{j\}}, \\ \{[\mathbf{Br}_i^\top, \mathbf{V}_i\mathbf{Br}_i^\top + \mathbf{k}^\top + \alpha\mathbf{c}^\perp]_2\}_{i<\ell}, [\mathbf{Br}_\ell^\top, \mathbf{V}_\ell\mathbf{Br}_\ell^\top + \mathbf{k}^\top]_2, \\ \{[\mathbf{Br}_i^\top, \mathbf{V}_i\mathbf{Br}_i^\top + \mathbf{k}^\top]_2\}_{i>\ell} \end{array} \right),$$

and the challenge verification text v^* is that

$$\left(\begin{array}{l} [\mathbf{c}]_1, \left[\sum_{j\in[L]} (\mathbf{cQ}_j + m^* \cdot \mathbf{cT}_j) \right]_1, \left[\sum_{j\in[L]} \mathbf{cW}_j(\mathbf{C}_x \otimes \mathbf{I}_{k+1}) \right]_1, \\ \left[\sum_{j\in[L]} (\mathbf{cV}_j + \mathbf{cU}_j)(\mathbf{a}_{y_j} \otimes \mathbf{I}_{k+1}) + \mathbf{cW}_j(\mathbf{K}_{y_j} \otimes \mathbf{I}_{k+1}) \right]_1, [\mathbf{ck}^\top]_T \end{array} \right).$$

Finally, for each $\kappa \in [Q_\ell]$, the corresponding signature $\sigma_{\ell,\kappa}$ is in the form:

$$\left(\begin{array}{l} [\mathbf{B}_\ell\mathbf{t}^\top]_2, [\mathbf{Br}_\ell^\top]_2, [\mathbf{V}_\ell\mathbf{Br}_\ell^\top + \mathbf{k}^\top + \mathbf{U}_\ell\mathbf{Br}_\ell^\top + (\mathbf{Q}_\ell\mathbf{B}_\ell\mathbf{t}^\top + m_{\ell,\kappa} \cdot \mathbf{T}_\ell\mathbf{B}_\ell\mathbf{t}^\top)]_2, \\ \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{V}_j\mathbf{Br}_\ell^\top + \mathbf{U}_j\mathbf{Br}_\ell^\top)\mathbf{a}_{y_j} + \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{Br}_\ell^\top)\mathbf{K}_{y_j} \right]_2, \\ \left[\sum_{j\in[L]\setminus\{\ell\}} \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{Br}_\ell^\top)\mathbf{C}_x \right]_2, \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{Q}_j\mathbf{B}_\ell\mathbf{t}^\top + m_{\ell,\kappa} \cdot \mathbf{T}_j\mathbf{B}_\ell\mathbf{t}^\top) \right] \end{array} \right)$$

where $\mathbf{t} \leftarrow \mathbb{Z}_p^{1\times k}$ and $m_{\ell,\kappa} \leftarrow \mathsf{H}(\ell, \mathsf{m}, x)$.

– $\mathsf{G}_{2,\ell-1,\kappa}(\kappa \in [Q_\ell])$: Identical to $\mathsf{G}_{2,\ell-1,\kappa-1}$ except that the signature $\sigma_{\ell,\kappa}$ is that

$$\left(\begin{array}{l}[\mathbf{B}_\ell \mathbf{t}^\top]_2, [\mathbf{B}\mathbf{r}_\ell^\top]_2, \\ [\mathbf{V}_\ell \mathbf{B}\mathbf{r}_\ell^\top + \mathbf{k}^\top + \mathbf{U}_\ell \mathbf{B}\mathbf{r}_\ell^\top + \boxed{\beta_{\ell,\kappa}\mathbf{c}^\perp} + (\mathbf{Q}_\ell \mathbf{B}_\ell \mathbf{t}^\top + m_{\ell,\kappa} \cdot \mathbf{T}_\ell \mathbf{B}_\ell \mathbf{t}^\top)]_2, \\ \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{V}_j \mathbf{B}\mathbf{r}_\ell^\top + \mathbf{U}_j \mathbf{B}\mathbf{r}_\ell^\top)\mathbf{a}_{y_j} + \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_\ell^\top)\mathbf{K}_{y_j}\right]_2, \\ \left[\sum_{j\in[L]\setminus\{\ell\}} \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_\ell^\top)\mathbf{C}_x\right]_2, \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{Q}_j \mathbf{B}_\ell \mathbf{t}^\top + m_{\ell,\kappa} \cdot \mathbf{T}_j \mathbf{B}_\ell \mathbf{t}^\top)\right]\end{array}\right)$$

where $\mathsf{G}_{2,\ell-1,\kappa-1} \approx_c \mathsf{G}_{2,\ell-1,\kappa}$ for all $\kappa \in [Q_\ell]$; see Lemma 2 for more details.

– $\mathsf{G}_{2,\ell-1,Q_\ell+1}$: Identical to $\mathsf{G}_{2,\ell-1,Q_\ell}$ except that the item marked with dashed box in crs is that

$$[\boxed{\mathbf{d}_\ell^\top}, \mathbf{V}_\ell \boxed{\mathbf{d}_\ell^\top} + \mathbf{k}^\top]_2$$

where $\mathbf{d}_\ell \leftarrow \mathbb{Z}_p^{1\times(k+1)}$; correspondingly, for all $\kappa \in [Q_\ell]$, the item marked with dashed box in $\sigma_{\ell,\kappa}$ is that

$$[\mathbf{V}_\ell \boxed{\mathbf{d}_\ell^\top} + \mathbf{k}^\top + \mathbf{U}_\ell \boxed{\mathbf{d}_\ell^\top} + \beta_{\ell,\kappa}\mathbf{c}^\perp]_2.$$

We have $\mathsf{G}_{2,\ell-1,Q_\ell+1} \approx_c \mathsf{G}_{2,\ell-1,Q_\ell}$, which follows the MDDH assumption:

$$([\mathbf{B}]_2, [\mathbf{B}\mathbf{r}_\ell^\top]_2) \approx_c ([\mathbf{B}]_2, [\mathbf{d}_\ell^\top]_2)$$

notice that $\mathbf{B} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{r}_\ell \leftarrow \mathbb{Z}_p^{1\times(k+1)}$ and $\mathbf{d}_\ell \leftarrow \mathbb{Z}_p^{1\times k}$. See Lemma 3 for more details.

– $\mathsf{G}_{2,\ell-1,Q_\ell+2}$: Identical to $\mathsf{G}_{2,\ell-1,Q_\ell+1}$ except that the item marked with dashed box in crs is that

$$[\mathbf{d}_\ell^\top, \mathbf{V}_\ell \mathbf{d}_\ell^\top + \mathbf{k}^\top + \boxed{\alpha \mathbf{c}^\perp}]_2.$$

We have $\mathsf{G}_{2,\ell-1,Q_\ell+1} \approx_s \mathsf{G}_{2,\ell-1,Q_\ell+2}$ which follows the following argument:

$$\begin{aligned}
&\begin{cases}\mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp, \mathbf{d}_\ell^\top, \mathbf{A}\mathbf{V}_\ell, \mathbf{V}_\ell \mathbf{B}, \mathbf{V}_\ell \mathbf{d}_\ell^\top + b\mathbf{c}^\perp \alpha; & \mathbf{A}\mathbf{U}_\ell \quad //\mathsf{crs};\ \mathsf{pk}_i; \\ \mathbf{c}, \mathbf{c}\mathbf{V}_\ell + \mathbf{c}\mathbf{U}_\ell; & \mathbf{d}_\ell^\top, \mathbf{U}_\ell \mathbf{d}_\ell^\top + \beta_{\ell,\kappa}\mathbf{c}^\perp \quad //v^*;\ \sigma_{i,\kappa}\end{cases} \\
\approx_s &\begin{cases}\mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp, \mathbf{d}_\ell^\top, \mathbf{A}\mathbf{V}_\ell, \mathbf{V}_\ell \mathbf{B}, \mathbf{V}_\ell \mathbf{d}_\ell^\top + \boxed{v_\ell \mathbf{c}^\perp} + b\mathbf{c}^\perp \alpha; & \mathbf{A}\mathbf{U}_\ell \\ \mathbf{c}, \mathbf{c}\mathbf{V}_\ell + \mathbf{c}\mathbf{U}_\ell + \boxed{v_\ell \mathbf{c}^\perp + u_\ell \mathbf{c}^\perp}; & \mathbf{d}_\ell^\top, \mathbf{U}_\ell \mathbf{d}_\ell^\top + \boxed{u_\ell \mathbf{c}^\perp} + \beta_{\ell,\kappa}\mathbf{c}^\perp\end{cases} \\
\approx_s &\begin{cases}\mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp, \mathbf{d}_\ell^\top, \mathbf{A}\mathbf{V}_\ell, \mathbf{V}_\ell \mathbf{B}, \mathbf{V}_\ell \mathbf{d}_\ell^\top + v_\ell \mathbf{c}^\perp + \cancel{b\mathbf{c}^\perp \alpha}; & \mathbf{A}\mathbf{U}_\ell \\ \mathbf{c}, \mathbf{c}\mathbf{V}_\ell + \mathbf{c}\mathbf{U}_\ell + v_\ell \mathbf{c}^\perp + u_\ell \mathbf{c}^\perp; & \mathbf{d}_\ell^\top, \mathbf{U}_\ell \mathbf{d}_\ell^\top + \cancel{u_\ell \mathbf{c}^\perp} + \beta_{\ell,\kappa}\mathbf{c}^\perp\end{cases}
\end{aligned}$$

where $b \in \{0,1\}$.

- The first $\approx_s$ follows that:

$$\mathbf{V}_\ell \mapsto \mathbf{V}_\ell + \mathbf{c}^\perp v_\ell \mathbf{d}^\perp \quad \text{and} \quad \mathbf{U}_\ell \mapsto \mathbf{U}_\ell + \mathbf{c}^\perp u_\ell \mathbf{d}^\perp$$

  where $\mathbf{c}^\perp \in \mathbb{Z}_p^{k+1}$ and $\mathbf{d}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{c}^\perp = 0, \mathbf{c}\mathbf{c}^\perp = 1, \mathbf{d}^\perp \mathbf{B} = 0, \mathbf{d}^\perp \mathbf{d}_\ell = 1$.
- The second $\approx_s$ holds since $\beta_{\ell,\kappa}$ is sampled randomly and hence preserve the privacy of u_ℓ. Then v_ℓ in crs also seems to be sampled randomly because u_ℓ hides v_ℓ in challenge verification text.

– $\mathsf{G}_{2,\ell-1,Q_\ell+3}$: Identical to $\mathsf{G}_{2,\ell-1,Q_\ell+2}$ except that the item marked with dashed box in crs is that

$$[\boxed{\mathbf{B}\mathbf{r}_\ell^\top}, \mathbf{V}_\ell \boxed{\mathbf{B}\mathbf{r}_\ell^\top} + \mathbf{k}^\top + \alpha \mathbf{c}^\perp]_2$$

where $\mathbf{B} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{r}_\ell \leftarrow \mathbb{Z}_p^{1\times(k+1)}$ and $\mathbf{d}_\ell \leftarrow \mathbb{Z}_p^{1\times k}$; correspondingly, for all $\kappa \in [Q_\ell]$, the item marked with dashed box in the $[\mathbf{k}_2^\top]_2$ of $\sigma_{\ell,\kappa}$ is that

$$[\mathbf{V}_\ell \boxed{\mathbf{B}\mathbf{r}_\ell^\top} + \mathbf{k}^\top + \mathbf{U}_\ell \boxed{\mathbf{B}\mathbf{r}_\ell^\top} + \beta_{\ell,\kappa}\mathbf{c}^\perp]_2.$$

We have $\mathsf{G}_{2,\ell-1,Q_\ell+2} \approx_c \mathsf{G}_{2,\ell-1,Q_\ell+3}$ which is symmetrical to $\mathsf{G}_{2,\ell-1,Q_\ell} \approx_c \mathsf{G}_{2,\ell-1,Q_\ell+1}$.

– $\mathsf{G}_{2,\ell-1,Q_\ell+4}$: Identical to $\mathsf{G}_{2,\ell-1,Q_\ell+3}$ except that the signature $\sigma_{\ell,\kappa}$ is that

$$\left(\begin{array}{l}[\mathbf{B}_\ell\mathbf{t}^\top]_2,\ [\mathbf{B}\mathbf{r}_\ell^\top]_2,\\ [\mathbf{V}_\ell\mathbf{B}\mathbf{r}_\ell^\top+\mathbf{k}^\top+\alpha\mathbf{c}^\perp+\mathbf{U}_\ell\mathbf{B}\mathbf{r}_\ell^\top+\cancel{\beta_{\ell,\kappa}\mathbf{c}^\perp}+(\mathbf{Q}_\ell\mathbf{B}_\ell\mathbf{t}^\top+m_{\ell,\kappa}\cdot\mathbf{T}_\ell\mathbf{B}_\ell\mathbf{t}^\top)]_2,\\ \left[\sum_{j\in[L]\setminus\{\ell\}}(\mathbf{V}_j\mathbf{B}\mathbf{r}_\ell^\top+\mathbf{U}_j\mathbf{B}\mathbf{r}_\ell^\top)\mathbf{a}_{y_j}+\mathbf{W}_j(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_\ell^\top)\mathbf{K}_{y_j}\right]_2,\\ \left[\sum_{j\in[L]\setminus\{\ell\}}\mathbf{W}_j(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_\ell^\top)\mathbf{C}_x\right]_2,\left[\sum_{j\in[L]\setminus\{\ell\}}(\mathbf{Q}_j\mathbf{B}_\ell\mathbf{t}^\top+m_{\ell,\kappa}\cdot\mathbf{T}_j\mathbf{B}_\ell\mathbf{t}^\top)\right]\end{array}\right).$$

We say $\mathsf{G}_{2,\ell-1,Q_\ell+3}\approx_c\mathsf{G}_{2,\ell-1,Q_\ell+4}$ for all $\kappa\in[Q_\ell]$. The proof is symmetrical to $\mathsf{G}_{2,\ell-1,0}\approx_c\mathsf{G}_{2,\ell-1,Q_\ell}$. Notice that $\mathsf{G}_{2,\ell-1,Q_\ell+4}\equiv\mathsf{G}_{2,\ell}$.

So far, we have finished dealing with the honest case.

Corrupted Case. In the corrupted case, $\mathcal{A}$ makes no query for signature oracle, so our proof just deals with crs in the similar sense to [ZZGQ23]. For each $\ell\in C_\ell$, the secret key $\mathsf{sk}_\ell=(\mathbf{U}_\ell,\mathbf{Q}_\ell,\mathbf{T}_\ell)$ been known to the adversary, but it is required that $P(x,y_\ell)=0$ for for the challenge (x^*,m^*). We start with the following sub-games:

– $\mathsf{G}'_{2,\ell-1,0}$: Identical to $\mathsf{G}_{2,\ell-1}$. Recall the crs is in the form

$$\mathsf{crs}=\left(\begin{array}{l}[\mathbf{A}\mathbf{k}^\top]_T,\mathsf{H},[\mathbf{A}]_1,\{[\mathbf{A}\mathbf{V}_i,\mathbf{A}\mathbf{W}_i]_1,[\mathbf{B}_i]_2\}_{i\in[L]},\\ \{[\mathbf{V}_i\mathbf{B}\mathbf{r}_j^\top,\mathbf{W}_i(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_j^\top)]_2\}_{j\in[L],i\in[L]\setminus\{j\}},\\ \{[\mathbf{B}\mathbf{r}_i^\top,\mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top+\mathbf{k}^\top+\alpha\mathbf{c}^\perp]_2\}_{i<\ell},\boxed{[\mathbf{B}\mathbf{r}_\ell^\top,\mathbf{V}_\ell\mathbf{B}\mathbf{r}_\ell^\top+\mathbf{k}^\top]_2},\\ \{[\mathbf{B}\mathbf{r}_i^\top,\mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top+\mathbf{k}^\top]_2\}_{i>\ell}\end{array}\right),$$

and the challenge verification text is that

$$v^*=\left(\begin{array}{l}[\mathbf{c}]_1,\left[\sum_{j\in[L]}(\mathbf{c}\mathbf{Q}_j+m^*\cdot\mathbf{c}\mathbf{T}_j)\right]_1,\left[\sum_{j\in[L]}\mathbf{c}\mathbf{W}_j(\mathbf{C}_x\otimes\mathbf{I}_{k+1})\right]_1,\\ \left[\sum_{j\in[L]}(\mathbf{c}\mathbf{V}_j+\mathbf{c}\mathbf{U}_j)(\mathbf{a}_{y_j}\otimes\mathbf{I}_{k+1})+\mathbf{c}\mathbf{W}_j(\mathbf{K}_{y_j}\otimes\mathbf{I}_{k+1})\right]_1,[\mathbf{c}\mathbf{k}^\top]_T\end{array}\right).$$

– $\mathsf{G}'_{2,\ell-1,1}$: Identical to $\mathsf{G}_{2,\ell-1,0}$ except that the item marked with $\boxed{\text{dashed box}}$ in crs is that

$$[\mathbf{d}_\ell^\top,\mathbf{V}_\ell\mathbf{d}_\ell^\top+\mathbf{k}^\top]_2.$$

We have $\mathsf{G}'_{2,\ell-1,0}\approx_c\mathsf{G}'_{2,\ell-1,1}$, which follows the MDDH assumption:

$$([\mathbf{B}]_2,[\mathbf{B}\mathbf{r}_\ell^\top]_2)\approx_c([\mathbf{B}]_2,[\mathbf{d}_\ell^\top]_2)$$

notice that $\mathbf{B}\leftarrow\mathbb{Z}_p^{(k+1)\times k},\mathbf{r}_\ell\leftarrow\mathbb{Z}_p^{1\times(k+1)}$ and $\mathbf{d}_\ell\leftarrow\mathbb{Z}_p^{1\times k}$. The proof is analogous to $\mathsf{G}_{2,\ell-1,\kappa}\approx_c\mathsf{G}_{2,\ell-1,Q_\ell+1}$ in the honest case and can be followed via the Lemma 2.

– $\mathsf{G}'_{2,\ell-1,2}$: Identical to $\mathsf{G}'_{2,\ell-1,1}$ except that the item marked with $\boxed{\text{dashed box}}$ in crs is that

$$[\mathbf{d}_\ell^\top,\mathbf{V}_\ell\mathbf{d}_\ell^\top+\mathbf{k}^\top+\alpha\mathbf{c}^\perp]_2.$$

We claim $\mathsf{G}'_{2,\ell-1,2} \approx_s \mathsf{G}'_{2,\ell-1,1}$ via the following argument ($b \in \{0,1\}$):

$$\begin{cases} \mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp, \mathbf{d}_\ell^\top, \mathbf{AV}_\ell, \mathbf{AW}_\ell, \mathbf{V}_\ell\mathbf{B}, \mathbf{W}_\ell(\mathbf{I}_n \otimes \mathbf{B}), \mathbf{V}_\ell\mathbf{d}_\ell^\top + b\mathbf{c}^\perp\alpha & //\mathsf{crs} \\ \mathbf{c}, \mathbf{cV}_\ell(\mathbf{a}_{y_\ell} \otimes \mathbf{I}_{k+1}) + \mathbf{cW}_\ell(\mathbf{K}_{y_\ell} \otimes \mathbf{I}_{k+1}) & \\ \mathbf{cW}_\ell(\mathbf{C}_{x^*} \otimes \mathbf{I}_{k+1}) & //\nu^* \end{cases}$$

$$\approx_s \begin{cases} \mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp, \mathbf{d}_\ell^\top, \mathbf{AV}_\ell, \mathbf{AW}_\ell, \mathbf{V}_\ell\mathbf{B}, \mathbf{W}_\ell(\mathbf{I}_n \otimes \mathbf{B}), \mathbf{V}_\ell\mathbf{d}_\ell^\top + \nu_\ell\mathbf{c}^\perp + b\mathbf{c}^\perp\alpha \\ \mathbf{c}, \mathbf{cV}_\ell(\mathbf{a}_{y_\ell} \otimes \mathbf{I}_{k+1}) + \mathbf{cW}_\ell(\mathbf{K}_{y_\ell} \otimes \mathbf{I}_{k+1}) + \nu_\ell\mathbf{a}_{y_\ell} \otimes \mathbf{d}^\perp + \mathbf{w}_\ell^\top\mathbf{K}_{y_\ell} \otimes \mathbf{d}^\perp, \\ \mathbf{cW}_\ell(\mathbf{C}_{x^*} \otimes \mathbf{I}_{k+1}) + \mathbf{w}_\ell^\top\mathbf{C}_{x^*} \otimes \mathbf{d}^\perp \end{cases}$$

$$\approx_s \begin{cases} \mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp, \mathbf{d}_\ell^\top, \mathbf{AV}_\ell, \mathbf{AW}_\ell, \mathbf{V}_\ell\mathbf{B}, \mathbf{W}_\ell(\mathbf{I}_n \otimes \mathbf{B}), \mathbf{V}_\ell\mathbf{d}_\ell^\top + \nu_\ell\mathbf{c}^\perp + b\mathbf{c}^\perp\alpha \\ \mathbf{c}, \mathbf{cV}_\ell(\mathbf{a}_{y_\ell} \otimes \mathbf{I}_{k+1}) + \mathbf{cW}_\ell(\mathbf{K}_{y_\ell} \otimes \mathbf{I}_{k+1}) + \cancel{\nu_\ell\mathbf{a}_{y_\ell} \otimes \mathbf{d}^\perp} + \mathbf{w}_\ell^\top\mathbf{K}_{y_\ell} \otimes \mathbf{d}^\perp, \\ \mathbf{cW}_\ell(\mathbf{C}_{x^*} \otimes \mathbf{I}_{k+1}) + \mathbf{w}_\ell^\top\mathbf{C}_{x^*} \otimes \mathbf{d}^\perp \end{cases}$$

$$\approx_s \begin{cases} \mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp, \mathbf{d}_\ell^\top, \mathbf{AV}_\ell, \mathbf{AW}_\ell, \mathbf{V}_\ell\mathbf{B}, \mathbf{W}_\ell(\mathbf{I}_n \otimes \mathbf{B}), \mathbf{V}_\ell\mathbf{d}_\ell^\top + \nu_\ell\mathbf{c}^\perp + \cancel{b\mathbf{c}^\perp\alpha} \\ \mathbf{c}, \mathbf{cV}_\ell(\mathbf{a}_{y_\ell} \otimes \mathbf{I}_{k+1}) + \mathbf{cW}_\ell(\mathbf{K}_{y_\ell} \otimes \mathbf{I}_{k+1}) + \mathbf{w}_\ell^\top\mathbf{K}_{y_\ell} \otimes \mathbf{d}^\perp, \\ \mathbf{cW}_\ell(\mathbf{C}_{x^*} \otimes \mathbf{I}_{k+1}) + \mathbf{w}_\ell^\top\mathbf{C}_{x^*} \otimes \mathbf{d}^\perp \end{cases}$$

Observe that:

- The first $\approx_s$ follows that:

$$\mathbf{V}_\ell \mapsto \mathbf{V}_\ell + \mathbf{c}^\perp\nu_\ell\mathbf{d}^\perp \quad \text{and} \quad \mathbf{W}_\ell \mapsto \mathbf{W}_\ell + \mathbf{c}^\perp(\mathbf{w}_\ell^\top \otimes \mathbf{d}^\perp)$$

  where $\nu_\ell \leftarrow \mathbb{Z}_p$ and $\mathbf{w}_\ell \leftarrow \mathbb{Z}_p^n$.
- The second $\approx_s$ follows the *α-privacy* of predicate encoding since $P(x^*, y_\ell) = 0$.
- The last $\approx_s$ holds since ν_ℓ is sampled randomly and only appears in crs.

– $\mathsf{G}'_{2,\ell-1,3}$: Identical to $\mathsf{G}'_{2,\ell-1,2}$ except that that the item marked with dashed box in crs is that

$$[\,\mathbf{Br}_\ell^\top, \mathbf{V}_\ell\mathbf{Br}_\ell^\top + \mathbf{k}^\top + \alpha\mathbf{c}^\perp]_2$$

where $\mathbf{B} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{r}_\ell \leftarrow \mathbb{Z}_p^{1\times(k+1)}$ and $\mathbf{d}_\ell \leftarrow \mathbb{Z}_p^{1\times k}$. We have $\mathsf{G}'_{2,\ell-1,2} \approx_c \mathsf{G}'_{2,\ell-1,3}$, which follows the MDDH assumption:

$$([\mathbf{B}]_2, [\mathbf{Br}_\ell^\top]_2) \approx_c ([\mathbf{B}]_2, [\mathbf{d}_\ell^\top]_2).$$

The proof is analogous to $\mathsf{G}_{2,\ell-1,Q_\ell+2} \approx_c \mathsf{G}_{2,\ell-1,Q_\ell+3}$ in the honest case and can be followed via the Lemma 2.

Notice that $\mathsf{G}'_{2,\ell-1,3} \equiv \mathsf{G}'_{2,\ell}$. So far, we have finished dealing with the corrupted case. Finally, we prove $\mathsf{G}_{2,\ell-1} \approx_c \mathsf{G}_{2,\ell}$ for each honest/corrupted slot $\ell \in \mathcal{D}_\ell$ by the above strategies.

4.4 Lemmata

In the following, we use $\mathsf{Adv}^i_{\mathcal{A}}(\lambda)$ to denote the advantage of $\mathcal{A}$ in G_i.

Lemma 1 **($\mathsf{G}_0 \approx_c \mathsf{G}_1$).** *For any adversary $\mathcal{A}$, there exists algorithm $\mathcal{B}_1$ with close running time to $\mathcal{A}$ such that*

$$|\mathsf{Adv}^0_{\mathcal{A}}(\lambda) - \mathsf{Adv}^1_{\mathcal{A}}(\lambda)| \le \mathsf{Adv}^{MDDH}_{\mathcal{B}_1}(\lambda) + \mathsf{negl}(\lambda).$$

Proof. Recall that the difference between the two games is that we replace $[\mathbf{sA}]_1$ in G_0 with $[\mathbf{c}]_1$, where $\mathbf{A} \leftarrow \mathbb{Z}_p^{k\times(k+1)}, \mathbf{s} \leftarrow \mathbb{Z}_p^{1\times k}$ and $\mathbf{c} \leftarrow \mathbb{Z}_p^{1\times(k+1)}$. This follows from $(k, k+1, 1)$-MDDH assumption, which ensures that:

$$([\mathbf{A}]_1, [\mathbf{sA}]_1) \approx_c ([\mathbf{A}]_1, [\mathbf{c}]_1).$$

On input $([\mathbf{A}]_1, [\hat{\mathbf{t}}]_1)$ where $\hat{\mathbf{t}} = \mathbf{sA}$ or $\hat{\mathbf{t}} = \mathbf{c}$, algorithm $\mathcal{B}_1$ works as follows:

Setup. Sample

$$\mathbf{B} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{k} \leftarrow \mathbb{Z}_p^{1\times(k+1)}, \{\mathbf{D}_i \leftarrow \mathbb{Z}_p^{k\times k}, \mathbf{V}_i \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}, \mathbf{W}_i \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)n}, \mathbf{r}_i \leftarrow \mathbb{Z}_p^{1\times k}\}_{i\in[L]}.$$

Set $\mathbf{B}_i = \mathbf{B}\mathbf{D}_i$ for each $i \in [L]$ and output

$$\mathsf{crs} = \begin{pmatrix} [\mathbf{A}]_1, \{[\mathbf{A}\mathbf{V}_i, \mathbf{A}\mathbf{W}_i]_1, [\mathbf{B}\mathbf{r}_i^\top, \mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \mathbf{k}^\top, \mathbf{B}_i]_2\}_{i\in[L]}, \\ \{[\mathbf{V}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{W}_i(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_j^\top)]_2\}_{j\in[L],i\in[L]\setminus\{j\}}, [\mathbf{A}\mathbf{k}^\top]_T, \mathsf{H} \end{pmatrix}.$$

Query. Here, we deal with the query from $\mathcal{A}$.

- For all $i \in [L]$ and each $(\mathsf{pk}_i, \mathsf{sk}_i) \in \mathcal{D}_i$ is generated honestly as :

$$\mathsf{pk}_i = ([\mathbf{A}\mathbf{U}_i, \mathbf{A}\mathbf{Q}_i, \mathbf{A}\mathbf{T}_i]_1, \{[\mathbf{U}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{Q}_i\mathbf{B}_j, \mathbf{T}_i\mathbf{B}_j]_2\}_{j\in[L]\setminus\{i\}})$$

  and $\mathsf{sk}_i = (\mathbf{U}_i, \mathbf{Q}_i, \mathbf{T}_i)$ where $\mathbf{U}_i, \mathbf{Q}_i, \mathbf{T}_i \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$.
- For $\mathsf{OSig}(i, x, \mathsf{m})$, sample $\mathbf{t} \leftarrow \mathbb{Z}_p^{1\times k}$ and compute $\mathbf{C}_x$, output signature $\sigma_{i,x,\mathsf{m}}$ as

$$\begin{pmatrix} [\underbrace{\mathbf{B}_i\mathbf{t}^\top}_{\mathbf{k}_0^\top}]_2, [\underbrace{\mathbf{B}\mathbf{r}_i^\top}_{\mathbf{k}_1^\top}]_2, [\underbrace{\mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \mathbf{k}^\top + \mathbf{U}_i\mathbf{B}\mathbf{r}_i^\top + (\mathbf{Q}_i\mathbf{B}_i\mathbf{t}^\top + m\cdot\mathbf{T}_i\mathbf{B}_i\mathbf{t}^\top)}_{\mathbf{k}_2^\top}]_2 \\ \left[\underbrace{\sum_{j\in[L]\setminus\{i^*\}} (\mathbf{V}_j\mathbf{B}\mathbf{r}_i^\top + \mathbf{U}_j\mathbf{B}\mathbf{r}_i^\top)\mathbf{a}_{y_j} + \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_i^\top)\mathbf{K}_{y_j}}_{\mathbf{K}_3}\right]_2, \\ \left[\underbrace{\sum_{j\in[L]\setminus\{i\}} \mathbf{W}_j(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_i^\top)\mathbf{C}_x}_{\mathbf{K}_4}\right]_2, \left[\underbrace{\sum_{j\in[L]\setminus\{i\}} (\mathbf{Q}_j\mathbf{B}_i\mathbf{t}^\top + m\cdot\mathbf{T}_j\mathbf{B}_i\mathbf{t}^\top)}_{\mathbf{k}_5^\top}\right]_2 \end{pmatrix}$$

Challenge. On input challenge (i^*, x^*, m^*), output $v_{i^*,x,\mathsf{m}}$ as

$$\begin{pmatrix} [\underbrace{\hat{\mathbf{t}}}_{\mathbf{v}_0}]_1, \left[\underbrace{\sum_{j\in[L]} (\hat{\mathbf{t}}\mathbf{Q}_j + m\cdot\hat{\mathbf{t}}\mathbf{T}_j)}_{\mathbf{v}_1}\right]_1, \left[\underbrace{\sum_{j\in[L]} \hat{\mathbf{t}}\mathbf{W}_j(\mathbf{C}_x \otimes \mathbf{I}_{k+1})}_{\mathbf{v}_2}\right]_1 \\ \left[\underbrace{\sum_{j\in[L]} (\hat{\mathbf{t}}\mathbf{V}_j + \hat{\mathbf{t}}\mathbf{U}_j)(\mathbf{a}_{y_j} \otimes \mathbf{I}_{k+1}) + \hat{\mathbf{t}}\mathbf{W}_j(\mathbf{K}_{y_j} \otimes \mathbf{I}_{k+1})}_{\mathbf{v}_3}\right]_1, [\underbrace{\hat{\mathbf{t}}\mathbf{k}^\top}_{v_4}]_T \end{pmatrix}.$$

Observe that when $\hat{\mathbf{t}} = \mathbf{s}\mathbf{A}$, the simulation is identical to G_0; when $\hat{\mathbf{t}} = \mathbf{c}$, the simulation is identical to G_1. This readily proves the lemma. □

Lemma 2 ($\mathsf{G}_{2,\ell-1,\kappa-1} \approx_c \mathsf{G}_{2,\ell-1,\kappa}$)**.** *For any adversary $\mathcal{A}$, there exists algorithm $\mathcal{B}_2$ with close running time to $\mathcal{A}$ such that*

$$|\mathsf{Adv}_{\mathcal{A}}^{2,\ell-1,\kappa-1}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{2,\ell-1,\kappa}(\lambda)| \le 2\cdot\mathsf{Adv}_{\mathcal{B}_2}^{MDDH}(\lambda) + \mathsf{negl}(\lambda).$$

Proof. The transition between $\mathsf{G}_{2,\ell-1,\kappa-1}$ and $\mathsf{G}_{2,\ell-1,\kappa}$ is similar to the secret key transition of IBE in [CGW15]. Recall that in $\mathsf{G}_{2,\ell-1,\kappa-1}$, we have

– For all $i < \kappa$, the signature $\sigma_{\ell,i}$ is that

$$\left(\begin{array}{l}[\mathbf{B}_\ell\mathbf{t}^\top]_2, [\mathbf{B}\mathbf{r}_\ell^\top]_2,\\ [\mathbf{V}_\ell\mathbf{B}\mathbf{r}_\ell^\top + \mathbf{k}^\top + \mathbf{U}_\ell\mathbf{B}\mathbf{r}_\ell^\top + \boxed{\beta_{\ell,i}\mathbf{c}^\perp} + (\mathbf{Q}_\ell\mathbf{B}_\ell\mathbf{t}^\top + m_{\ell,i}\cdot\mathbf{T}_\ell\mathbf{B}_\ell\mathbf{t}^\top)]_2,\\ \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{V}_j\mathbf{B}\mathbf{r}_\ell^\top + \mathbf{U}_j\mathbf{B}\mathbf{r}_\ell^\top)\mathbf{a}_{y_j} + \mathbf{W}_j(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_\ell^\top)\mathbf{K}_{y_j}\right]_2,\\ \left[\sum_{j\in[L]\setminus\{\ell\}} \mathbf{W}_j(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_\ell^\top)\mathbf{C}_x\right]_2, \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{Q}_j\mathbf{B}_\ell\mathbf{t}^\top + m_{\ell,\kappa}\cdot\mathbf{T}_j\mathbf{B}_\ell\mathbf{t}^\top)\right]\end{array}\right)$$

where $\beta_{\ell,i} \leftarrow \mathbb{Z}_p$ and $\mathbf{B}_\ell = \mathbf{B}\mathbf{D}_\ell$.

– For all $i \geq \kappa$, the signature $\sigma_{\ell,i}$ is that

$$\left(\begin{array}{l}[\mathbf{B}_\ell\mathbf{t}^\top]_2,\ [\mathbf{B}\mathbf{r}_\ell^\top]_2,\ [\mathbf{V}_\ell\mathbf{B}\mathbf{r}_\ell^\top + \mathbf{k}^\top + \mathbf{U}_\ell\mathbf{B}\mathbf{r}_\ell^\top + (\mathbf{Q}_\ell\mathbf{B}_\ell\mathbf{t}^\top + m_{\ell,i}\cdot\mathbf{T}_\ell\mathbf{B}_\ell\mathbf{t}^\top)]_2,\\ \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{V}_j\mathbf{B}\mathbf{r}_\ell^\top + \mathbf{U}_j\mathbf{B}\mathbf{r}_\ell^\top)\mathbf{a}_{y_j} + \mathbf{W}_j(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_\ell^\top)\mathbf{K}_{y_j}\right]_2,\\ \left[\sum_{j\in[L]\setminus\{\ell\}} \mathbf{W}_j(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_\ell^\top)\mathbf{C}_x\right]_2, \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{Q}_j\mathbf{B}_\ell\mathbf{t}^\top + m_{\ell,i}\cdot\mathbf{T}_j\mathbf{B}_\ell\mathbf{t}^\top)\right]\end{array}\right).$$

The verification text ν^* is that

$$\left(\begin{array}{l}[\mathbf{c}]_1, \left[\sum_{j\in[L]} (\mathbf{c}\mathbf{Q}_j + m^*\cdot\mathbf{c}\mathbf{T}_j)\right]_1, \left[\sum_{j\in[L]} \mathbf{c}\mathbf{W}_j(\mathbf{C}_x\otimes\mathbf{I}_{k+1})\right]_1,\\ \left[\sum_{j\in[L]} (\mathbf{c}\mathbf{V}_j + \mathbf{c}\mathbf{U}_j)(\mathbf{a}_{y_j}\otimes\mathbf{I}_{k+1}) + \mathbf{c}\mathbf{W}_j(\mathbf{K}_{y_j}\otimes\mathbf{I}_{k+1})\right]_1, [\mathbf{c}\mathbf{k}^\top]_T\end{array}\right).$$

The only difference between $\mathsf{G}_{2,\ell-1,\kappa-1}$ and $\mathsf{G}_{2,\ell-1,\kappa}$ is the item $\boxed{[\beta_{\ell,\kappa}]_2}$ in $\sigma_{\ell,\kappa}$. Since $\mathbf{B}_\ell = \mathbf{B}\mathbf{D}_\ell$, we argue that for all $i \in [L]$, there exist $\overline{\mathbf{d}}_\ell^\perp$ such that $\overline{\mathbf{d}}_\ell^\perp\mathbf{B}_i = \mathbf{0}$ and $\overline{\mathbf{d}}_\ell^\perp\overline{\mathbf{d}}_{\ell,\kappa}^\top = 1$, where $\overline{\mathbf{d}}_{\ell,\kappa} \leftarrow \mathbb{Z}_p^{1\times(k+1)}$. We prove $\mathsf{G}_{2,\ell-1,\kappa-1} \approx_c \mathsf{G}_{2,\ell-1,\kappa}$ by the following argument ($\beta \in \{0,1\}$):

$$\begin{aligned}
&\left\{\begin{array}{ll}\mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp; \quad \mathbf{A}\mathbf{Q}_\ell, \mathbf{A}\mathbf{T}_\ell, \mathbf{Q}_\ell\mathbf{B}, \mathbf{T}_\ell\mathbf{B}; & //\mathsf{crs};\ \mathsf{pk}_\ell;\\ \mathbf{c}, \mathbf{c}\mathbf{Q}_\ell + m^*\cdot\mathbf{c}\mathbf{T}_\ell; & //\nu^*\\ \mathbf{B}_\ell\mathbf{t}^\top, \mathbf{Q}_\ell\mathbf{B}_\ell\mathbf{t}^\top + m_{\ell,\kappa}\cdot\mathbf{T}_\ell\mathbf{B}_\ell\mathbf{t}^\top + b\mathbf{c}^\perp\beta; & //\sigma_{\ell,\kappa}\end{array}\right.\\
\approx_c &\left\{\begin{array}{l}\mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp; \quad \mathbf{A}\mathbf{Q}_\ell, \mathbf{A}\mathbf{T}_\ell, \mathbf{Q}_\ell\mathbf{B}, \mathbf{T}_\ell\mathbf{B}\\ \mathbf{c}, \mathbf{c}\mathbf{Q}_\ell + m^*\cdot\mathbf{c}\mathbf{T}_\ell;\\ \overline{\mathbf{d}}_{\ell,\kappa}^\top, \mathbf{Q}_\ell\overline{\mathbf{d}}_{\ell,\kappa}^\top + m_{\ell,\kappa}\cdot\mathbf{T}_\ell\overline{\mathbf{d}}_{\ell,\kappa}^\top + b\mathbf{c}^\perp\beta;\end{array}\right.\\
\approx_s &\left\{\begin{array}{l}\mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp; \quad \mathbf{A}\mathbf{Q}_\ell, \mathbf{A}\mathbf{T}_\ell, \mathbf{Q}_\ell\mathbf{B}, \mathbf{T}_\ell\mathbf{B}\\ \mathbf{c}, \mathbf{c}\mathbf{Q}_\ell + m^*\cdot\mathbf{c}\mathbf{T}_\ell + (q_{\ell,\kappa} + m^*\cdot t_{\ell,\kappa})\overline{\mathbf{d}}_\ell^\perp;\\ \mathbf{d}_{\ell,\kappa}^\top, \mathbf{Q}_\ell\overline{\mathbf{d}}_{\ell,\kappa}^\top + m_{\ell,\kappa}\cdot\mathbf{T}_\ell\overline{\mathbf{d}}_{\ell,\kappa}^\top + b\mathbf{c}^\perp\beta + (q_{\ell,\kappa} + m_{\ell,\kappa}\cdot t_{\ell,\kappa})\mathbf{c}^\perp;\end{array}\right.\\
\approx_s &\left\{\begin{array}{l}\mathbf{A}, \mathbf{B}_\ell, \mathbf{c}^\perp; \quad \mathbf{A}\mathbf{Q}_\ell, \mathbf{A}\mathbf{T}_\ell, \mathbf{Q}_\ell\mathbf{B}, \mathbf{T}_\ell\mathbf{B}\\ \mathbf{c}, \mathbf{c}\mathbf{Q}_\ell + m^*\cdot\mathbf{c}\mathbf{T}_\ell + (q_{\ell,\kappa} + m^*\cdot t_{\ell,\kappa})\mathbf{d}_\ell^\perp;\\ \overline{\mathbf{d}}_{\ell,\kappa}^\top, \mathbf{Q}_\ell\overline{\mathbf{d}}_{\ell,\kappa}^\top + m_{\ell,\kappa}\cdot\mathbf{T}_\ell\overline{\mathbf{d}}_{\ell,\kappa}^\top + \cancel{b\mathbf{c}^\perp\beta} + (q_{\ell,\kappa} + m_{\ell,\kappa}\cdot t_{\ell,\kappa})\mathbf{c}^\perp;\end{array}\right.
\end{aligned}$$

We justify each step as follows:

– The first $\approx_c$ follows from the MDDH assumption:

$$([\mathbf{B}_\ell]_2, [\mathbf{B}_\ell \mathbf{t}^\top]_2) \approx_c ([\mathbf{B}_\ell]_2, [\overline{\mathbf{d}}_{\ell,\kappa}^\top]_2)$$

where $\mathbf{B}_\ell \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{t} \leftarrow \mathbb{Z}_p^{1\times k}$ and $\overline{\mathbf{d}}_{\ell,\kappa} \leftarrow \mathbb{Z}_p^{1\times(k+1)}$.

– The second $\approx_s$ follows tha fact that:

$$\mathbf{Q}_\ell \mapsto \mathbf{Q}_\ell + \mathbf{c}^\perp q_{\ell,\kappa} \overline{\mathbf{d}}_{\ell,\kappa}^\perp \quad \text{and} \quad \mathbf{T}_\ell \mapsto \mathbf{T}_\ell + \mathbf{c}^\perp t_{\ell,\kappa} \overline{\mathbf{d}}_{\ell,\kappa}^\perp$$

where $\mathbf{c}^\perp \in \mathbb{Z}_p^{k+1}$ and $\overline{\mathbf{d}}_{\ell,\kappa}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{c}^\perp = 0, \mathbf{c}\mathbf{c}^\perp = 1, \overline{\mathbf{d}}_{\ell,\kappa}^\perp \mathbf{B} = 0, \overline{\mathbf{d}}_{\ell,\kappa}^\perp \mathbf{d}_\ell = 1$.

– The last $\approx_s$ follows the fact that item $(q_{\ell,\kappa} + m_{\ell,\kappa} \cdot t_{\ell,\kappa})\mathbf{c}^\perp$ hides the item $b\mathbf{c}^\perp\beta$ since $m^* \neq m_{\ell,\kappa}$.

This readily proves the lemma. □

Lemma 3 ($\mathsf{G}_{2,\ell-1,Q_\ell} \approx_c \mathsf{G}_{2,\ell-1,Q_\ell+1}$)**.** *For any adversary $\mathcal{A}$, there exists algorithm $\mathcal{B}_3$ with close running time to $\mathcal{A}$ such that*

$$|\mathsf{Adv}^{2,\ell-1,Q_\ell}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{2,\ell-1,Q_\ell+1}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{MDDH}_{\mathcal{B}_3}(\lambda) + \mathsf{negl}(\lambda).$$

Proof. This follows from the $(k, k+1, 1)$-MDDH assumption:

$$[\mathbf{B}]_2, [\mathbf{B}\mathbf{r}_\ell^\top]_2 \approx_c [\mathbf{B}]_2, [\mathbf{d}_\ell^\top]_2$$

where $\mathbf{B} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{r}_\ell \leftarrow \mathbb{Z}_p^{1\times k}$ and $\mathbf{d}_\ell \leftarrow \mathbb{Z}_p^{1\times(k+1)}$. On input $[\mathbf{B}]_2, [\hat{\mathbf{t}}^\top]_2$ where $\hat{\mathbf{t}}^\top = \mathbf{B}\mathbf{r}_\ell^\top$ or $\hat{\mathbf{t}}^\top = \mathbf{d}_\ell^\top$, the algorithm $\mathcal{B}_3$ works as follow:

Setup. Sample

$$\begin{aligned}&\mathbf{A} \leftarrow \mathbb{Z}_p^{k\times(k+1)},\ \mathbf{k} \leftarrow \mathbb{Z}_p^{1\times(k+1)},\ \mathbf{c} \leftarrow \mathbb{Z}_p^{1\times(k+1)},\ \alpha \leftarrow \mathbb{Z}_p\\ &\{\mathbf{D}_i \leftarrow \mathbb{Z}_p^{k\times k},\ \mathbf{V}_i \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)},\ \mathbf{W}_i \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)n},\ \mathbf{r}_i \leftarrow \mathbb{Z}_p^{1\times k}\}_{i\in[L]}\end{aligned}$$

Set $[\mathbf{B}_i]_2 = [\mathbf{B}\mathbf{D}_i]_2$ for each $i \in [L]$ and output

$$\mathsf{crs} = \begin{pmatrix} [\mathbf{A}\mathbf{k}^\top]_T, \mathsf{H}, [\mathbf{A}]_1, \{[\mathbf{A}\mathbf{V}_i, \mathbf{A}\mathbf{W}_i]_1, [\mathbf{B}_i]_2\}_{i\in[L]},\\ \{[\mathbf{V}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{W}_i(\mathbf{I}_n \otimes \mathbf{B}\mathbf{r}_j^\top)]_2\}_{j\in[L],i\in[L]\setminus\{j\}},\\ \{[\mathbf{B}\mathbf{r}_i^\top, \mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \mathbf{k}^\top + \alpha\mathbf{c}^\perp]_2\}_{i<\ell}, [\hat{\mathbf{t}}_\ell, \mathbf{V}_\ell\hat{\mathbf{t}}_\ell + \mathbf{k}^\top]_2,\\ \{[\mathbf{B}\mathbf{r}_i^\top, \mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \mathbf{k}^\top]_2\}_{i>\ell} \end{pmatrix},$$

Query. Here, we deal with the query from $\mathcal{A}$.

– For all $i \in [L]$ and each $(\mathsf{pk}_i, \mathsf{sk}_i) \in \mathcal{D}_i$ is generated honestly as:

- if $i \neq \ell$, the pk_i is that

$$([\mathbf{A}\mathbf{U}_i, \mathbf{A}\mathbf{Q}_i, \mathbf{A}\mathbf{T}_i]_1, \{[\mathbf{U}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{Q}_i\mathbf{B}_j, \mathbf{T}_i\mathbf{B}_j]_2\}_{j\in[L]\setminus\{i,\ell\}}, [\mathbf{U}_i\hat{\mathbf{t}}^\top]_2);$$

- if $i = \ell$, the pk_ℓ is that

$$([\mathbf{A}\mathbf{U}_\ell, \mathbf{A}\mathbf{Q}_\ell, \mathbf{A}\mathbf{T}_\ell]_1, \{[\mathbf{U}_\ell\mathbf{B}\mathbf{r}_j^\top, \mathbf{Q}_\ell\mathbf{B}_j, \mathbf{T}_\ell\mathbf{B}_j]_2\}_{j\in[L]\setminus\{\ell\}});$$

where $\mathbf{U}_i, \mathbf{Q}_i, \mathbf{T}_i \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$.

– For $\mathsf{OSig}(\ell, x, \mathsf{m})$, sample $\mathbf{t} \leftarrow \mathbb{Z}_p^{1\times k}$ and compute $\mathbf{C}_x$, output $\sigma_{\ell,x,\mathsf{m}}$ as

$$\left(\begin{array}{l}[\mathbf{B}_\ell\mathbf{t}^\top]_2, [\hat{\mathbf{t}}^\top]_2, [\mathbf{V}_\ell\hat{\mathbf{t}}^\top + \mathbf{k}^\top + \mathbf{U}_\ell\hat{\mathbf{t}}^\top + (\mathbf{Q}_\ell\mathbf{B}_\ell\mathbf{t}^\top + m\cdot\mathbf{T}_\ell\mathbf{B}_\ell\mathbf{t}^\top)]_2, \\ \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{V}_j\hat{\mathbf{t}}^\top + \mathbf{U}_j\hat{\mathbf{t}}^\top)\mathbf{a}_{y_j} + \mathbf{W}_j(\mathbf{I}_n\otimes\hat{\mathbf{t}}^\top)\mathbf{K}_{y_j}\right]_2, \\ \left[\sum_{j\in[L]\setminus\{\ell\}} \mathbf{W}_j(\mathbf{I}_n\otimes\hat{\mathbf{t}}^\top)\mathbf{C}_x\right]_2, \left[\sum_{j\in[L]\setminus\{\ell\}} (\mathbf{Q}_j\mathbf{B}_\ell\mathbf{t}^\top + m\cdot\mathbf{T}_j\mathbf{B}_\ell\mathbf{t}^\top)\right]\end{array}\right).$$

where $m \leftarrow \mathsf{H}(i, \mathsf{m}, x)$.

Challenge. On input challenge (i^*, x^*, m^*), output v_{i^*,x^*,m^*} as

$$\left(\begin{array}{l}[\mathbf{c}]_1, \left[\sum_{j\in[L]} (\mathbf{c}\mathbf{Q}_j + m\cdot\mathbf{c}\mathbf{T}_j)\right]_1, \left[\sum_{j\in[L]} \mathbf{c}\mathbf{W}_j(\mathbf{C}_x\otimes\mathbf{I}_{k+1})\right]_1 \\ \left[\sum_{j\in[L]} (\mathbf{c}\mathbf{V}_j + \mathbf{c}\mathbf{U}_j)(\mathbf{a}_{y_j}\otimes\mathbf{I}_{k+1}) + \mathbf{c}\mathbf{W}_j(\mathbf{K}_{y_j}\otimes\mathbf{I}_{k+1})\right]_1, [\mathbf{c}\mathbf{k}^\top]_T\end{array}\right).$$

Observe that when $\hat{\mathbf{t}}^\top = \mathbf{B}\mathbf{r}_\ell^\top$, the simulation is identical to $\mathsf{G}_{2,\ell-1,Q_\ell}$; when $\hat{\mathbf{t}}^\top = \mathbf{d}_\ell^\top$, the simulation is identical to $\mathsf{G}_{2,\ell-1,Q_\ell+1}$. □

Lemma 4 ($\mathsf{G}_{2,L} \approx_s \mathsf{G}_3$). *For any adversary $\mathcal{A}$, we have*

$$|\mathsf{Adv}^{2,L}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{3}_{\mathcal{A}}(\lambda)| = 0$$

Proof. First, in the process of simulating crs, we program $\mathbf{k}^\top$ in both $\mathsf{G}_{2,L}$ and G_3 as follow:

$$\mathbf{k}^\top \mapsto \mathbf{k}^\top - \mathbf{c}^\perp\alpha$$

where $\mathbf{k} \leftarrow \mathbb{Z}_p^{1\times(k+1)}, \alpha \leftarrow \mathbb{Z}_p$. Under the fact that $\mathbf{A}\mathbf{c}^\perp = \mathbf{0}$, crs works as follow:

$$\mathsf{crs} = \left(\begin{array}{l}[\mathbf{A}]_1, \{[\mathbf{A}\mathbf{V}_i, \mathbf{A}\mathbf{W}_i]_1, [\mathbf{B}\mathbf{r}_i^\top, \mathbf{V}_i\mathbf{B}\mathbf{r}_i^\top + \mathbf{k}^\top, \mathbf{B}_i]_2\}_{i\in[L]}, \\ \{[\mathbf{V}_i\mathbf{B}\mathbf{r}_j^\top, \mathbf{W}_i(\mathbf{I}_n\otimes\mathbf{B}\mathbf{r}_j^\top)]_2\}_{j\in[L],i\in[L]\setminus\{j\}}, [\mathbf{A}\mathbf{k}^\top]_T, \mathsf{H}\end{array}\right).$$

Then, α in the challenge verification text is that $[\mathbf{c}\mathbf{k}^\top - \alpha]_T$, where this follows from the fact that $\mathbf{c}\mathbf{c}^\perp = 1$. Furthermore, α only correlate to $[\mathbf{c}\mathbf{k}^\top - \alpha]_T$ in the challenge verification text. $[\alpha]_T$ is uniformly distributed over $\mathbb{G}_T$ which implies that the distribution of $[\mathbf{c}\mathbf{k}^\top - \alpha]_T$ is identical to a random coin in $\mathbb{G}_T$, just like in G_3. This readily proves the lemma. □

5 From Slotted Registered ABS to Full-fledged Registered ABS

We exploit the "power-of-two" approach from [HLWW23] to convert a slotted registered ABS to a full-fledged registered ABS.

Construction. Suppose a full-fledged registered ABS mostly supports $L = 2^\ell$ users, this approach needs $\ell+1$ copies of slotted registered ABS with $1, 2, 4, \cdots, 2^\ell$ slots. And the public state $\mathsf{aux} = (\mathcal{D}_1, \mathcal{D}_2, \mathsf{mpk})$ consists of the following terms:

– $\mathcal{D}_1[k, i] = (\mathsf{pk}, y)$: where $k \in [0, \ell]$ and $i \in [2^k]$. This dictionary assigns a user's (pk, y) to the slot i of the 2^k-slotted registered ABS scheme.

– $\mathcal{D}_2[k,n] = \mathsf{hk}$: where $k \in [0,\ell]$ and $n \in [L]$. This dictionary assigns a hk of slotted registered ABS to the 2^k-slotted registered ABS scheme and the user index n.
– $\mathsf{mpk} = (\mathsf{ctr}, \mathsf{mpk}_0, \cdots, \mathsf{mpk}_\ell)$ denotes the current master public key. Where $(\mathsf{mpk}_k)_{k\in[0,\ell]}$ denote the master public keys of $\ell+1$ copies of slotted registered ABS, and ctr denotes the number of currently registered users. When no registered user, we initially set $\mathsf{mpk} = (0, \perp, \cdots \perp)$.

When no registered user, we initially set $\mathsf{aux} = (\emptyset, \emptyset, \perp)$. Assuming a slotted registered ABS $\Pi_s = (s.\mathsf{Setup}, s.\mathsf{Gen}, s.\mathsf{IsValid}, s.\mathsf{Agg}, s.\mathsf{Sig}, s.\mathsf{Ver})$, a full-fledged registered ABS $\Pi = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Reg}, \mathsf{Upd}, \mathsf{Sig}, \mathsf{Ver})$ can be constructed as follows:

– $\mathsf{Setup}(1^\lambda, P, 1^L)$: Compute $\ell = \log L$. For all $k \in [0,\ell]$, run $\mathsf{crs}_k \leftarrow s.\mathsf{Setup}(1^\lambda, P, 1^{2^k})$. Output

$$\mathsf{crs} = (\mathsf{crs}_0, \cdots, \mathsf{crs}_\ell)$$

– $\mathsf{Gen}(\mathsf{crs}, \mathsf{aux})$: Fetch $\mathsf{crs} = (\mathsf{crs}_k)_{k\in[0,\ell]}$ and $\mathsf{aux} = (\mathcal{D}_1, \mathcal{D}_2, \mathsf{mpk})$, where $\mathsf{mpk} = \big(\mathsf{ctr}, (\mathsf{mpk}_k)_{k\in[0,\ell]}\big)$. For all $k \in [0,\ell]$, compute

$$i_k = (\mathsf{ctr} \bmod 2^k) + 1$$

and run $(\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow s.\mathsf{Gen}(\mathsf{crs}_k, i_k)$. Set $\mathsf{ctr}' = \mathsf{ctr}$ and output

$$\mathsf{pk} = (\mathsf{ctr}', \mathsf{pk}_0, \cdots, \mathsf{pk}_\ell) \quad \text{and} \quad \mathsf{sk} = (\mathsf{ctr}', \mathsf{sk}_0, \cdots, \mathsf{sk}_\ell)$$

– $\mathsf{Reg}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}, y)$: Fetch $\mathsf{crs} = (\mathsf{crs}_k)_{k\in[0,\ell]}$, $\mathsf{aux} = (\mathcal{D}_1, \mathcal{D}_2, \mathsf{mpk})$, and $\mathsf{pk} = \big(\mathsf{ctr}', (\mathsf{pk}_k)_{k\in[0.\ell]}\big)$, where $\mathsf{mpk} = (\mathsf{ctr}, (\mathsf{mpk}_k)_{k\in[0,\ell]})$. For all $k \in [0,\ell]$, do the following operates:
  - Compute $i_k = (\mathsf{ctr} \bmod 2^k) + 1$;
  - Check if $s.\mathsf{IsValid}(\mathsf{crs}_k, i_k, \mathsf{pk}_k) = 1$ and $\mathsf{ctr}' = \mathsf{ctr}$. If the check passes, set $\mathsf{ctr} = \mathsf{ctr} + 1$, if the check fails,the algorithm halts and output $(\mathsf{mpk}, \mathsf{aux})$;
  - Update $\mathcal{D}_1[k, i_k] = (\mathsf{pk}, y)$;
  - If $i_k = 2^k$: compute $(\mathsf{mpk}'_k, (\mathsf{hk}_{k,j})_{j\in[2^k]}) \leftarrow s.\mathsf{Agg}(\mathsf{crs}_k, (\mathcal{D}_1[k,i])_{i\in[2^k]})$. Update $\mathsf{mpk}_k = \mathsf{mpk}'_k$, and for all $j \in [2^k]$, update $\mathcal{D}_2[k, \mathsf{ctr} - 2^k + j] = \mathsf{hk}_{k,j}$.

  Update the master public key $\mathsf{mpk} = (\mathsf{ctr}, (\mathsf{mpk}_0, \cdots, \mathsf{mpk}_\ell))$ and $\mathsf{aux} = (\mathcal{D}_1, \mathcal{D}_2, \mathsf{mpk})$, output $(\mathsf{mpk}, \mathsf{aux})$.
– $\mathsf{Upd}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk})$: Fetch $\mathsf{crs} = (\mathsf{crs}_k)_{k\in[0,\ell]}$, $\mathsf{aux} = (\mathcal{D}_1, \mathcal{D}_2, \mathsf{mpk})$, and $\mathsf{pk} = \big(\mathsf{ctr}', (\mathsf{pk}_k)_{k\in[0.\ell]}\big)$, where $\mathsf{mpk} = (\mathsf{ctr}, (\mathsf{mpk}_k)_{k\in[0,\ell]})$. Output

$$\mathsf{hk} = \begin{cases} (\underbrace{\mathcal{D}_2[0, \mathsf{ctr}'+1]}_{\mathsf{hk}_0}, \cdots, \underbrace{\mathcal{D}_2[\ell, \mathsf{ctr}'+1]}_{\mathsf{hk}_\ell}) & \text{if } \mathsf{ctr}' < \mathsf{ctr} \\ \perp & \text{otherwise} \end{cases} \tag{3}$$

– $\mathsf{Sig}(\mathsf{mpk}, \mathsf{hk}, \mathsf{sk}, x, \mathsf{m})$: Fetch $\mathsf{mpk} = (\mathsf{ctr}, (\mathsf{mpk}_k)_{k\in[0,\ell]})$, $\mathsf{sk} = (\mathsf{ctr}', (\mathsf{sk}_k)_{k\in[0,\ell]})$ and $\mathsf{hk} = (\mathsf{hk}_k)_{k\in[0,\ell]}$. For all $k \in [0,\ell]$, if exists $d \in [0,\ell]$ such that $\mathsf{mpk}_d \neq \perp$ and $\mathsf{hk}_d = \perp$, output `getupd`; otherwise, compute:

$$\sigma_k = \begin{cases} s.\mathsf{Sig}(\mathsf{hk}_k, \mathsf{sk}_k, x, \mathsf{m}) & \text{if } \mathsf{mpk}_k \neq \perp \\ \perp & \text{otherwise} \end{cases} \tag{4}$$

Output $\sigma = (\mathsf{ctr}', \sigma_0, \cdots, \sigma_\ell)$.
– $\mathsf{Ver}(\mathsf{mpk}, \sigma, x, \mathsf{m})$: Fetch $\mathsf{mpk} = (\mathsf{ctr}, (\mathsf{mpk}_k)_{k\in[0,\ell]})$ and $\sigma = (\mathsf{ctr}', (\sigma_k)_{k\in[0,\ell]})$. Proceed as follows:
  - If $\mathsf{ctr}' \geq \mathsf{ctr}$: output $\perp$.
  - Otherwise, compute $\mathsf{ctr} = (a_\ell, \cdots, a_0)_2$ and $\mathsf{ctr}' = (b_\ell, \cdots, b_0)_2$. We denote k_d as the maximum $k \in [0,\ell]$ such that $a_k \neq b_k$.
  - Otherwise, output $s.\mathsf{Ver}(\mathsf{mpk}_{k_d}, \sigma_{k_d}, x, \mathsf{m})$.

Analysis. We would demonstrate the correctness, compactness, efficiency and unforgeability of the above construction via a series of theorems.

Theorem 2 (Correctness). *Suppose construction Π_s is complete and perfectly correct. Then construction Π is perfectly correct.*

Proof. After querying y^* to ORegT and obtaining the output $(t, \mathsf{mpk}, \mathsf{aux}, \mathsf{pk}^*, \mathsf{sk}^*, \mathsf{hk}^*)$, let $(i_\kappa, x_\kappa, \mathsf{m}_\kappa)$ be the κ-th query to OSig which returns $(t_\kappa, \sigma_\kappa)$. Then compute $b \leftarrow \mathsf{Ver}(\mathsf{mpk}_{i_\kappa}, \sigma_\kappa, x_\kappa, \mathsf{m}_\kappa)$:

- Parse $\mathsf{sk}^* = (\mathsf{ctr}^*, (\mathsf{sk}^*_k)_{k\in[0,\ell]})$, $\mathsf{pk}^* = (\mathsf{ctr}^*, (\mathsf{pk}^*_k)_{k\in[0,\ell]})$, $\mathsf{hk}^* = (\mathsf{hk}^*_k)_{k\in[0,\ell]}$, $\sigma_\kappa = (\mathsf{ctr}_\kappa, (\sigma_{\kappa,k})_{k\in[0,\ell]})$ and $\mathsf{aux} = (\mathsf{ctr}_{\mathsf{aux}}, \mathcal{D}_1, \mathcal{D}_2, \mathsf{mpk})$, and the master public key is that $\mathsf{mpk} = (\mathsf{ctr}_{\mathsf{aux}}, (\mathsf{mpk}_k)_{k\in[0,\ell]})$;
- Let k^* denote the max bit on which ctr^* and $\mathsf{ctr}_{\mathsf{aux}}$ differ.
- The challenger computes $b \leftarrow s.\mathsf{Ver}(\mathsf{mpk}_{k^*}, \sigma_{\kappa,k^*}, x_\kappa, \mathsf{m}_\kappa)$.

Similarly, we can show that $\mathcal{D}_2[\mathsf{ctr}^* + 1, k^*] = \mathsf{hk}_{k^*,\mathsf{ctr}^*}$ will never be updated after making a query to ORegT by following the lemma 6.3 in [HLWW23]. Thus, the signature σ_κ is well-formed and follows the correctness of Π_s, i.e., $b = 1$. □

Theorem 3 (Compactness). *Suppose construction Π_s is compact. Then construction Π is compact.*

Proof. Observe that $|\mathsf{mpk}| = |\mathsf{ctr}| + \sum_{i\in[0,\ell]} |\mathsf{mpk}_i|$ and $|\mathsf{hk}| = \sum_{i\in[0,\ell]} |\mathsf{hk}_i|$ in Π, where ctr is a ℓ-bit number. According to the compactness of Π_s, we have $|\mathsf{mpk}_i| = \mathsf{poly}(\lambda, P, \log L)$ and $|\mathsf{hk}_i| = \mathsf{poly}(\lambda, P, \log L)$ for all $i \in [0, \ell]$. Then it holds that $|\mathsf{mpk}| = \mathsf{poly}(\lambda, P, \log i)$ and $|\mathsf{hk}| = \mathsf{poly}(\lambda, P, \log |\mathcal{R}|)$. □

Theorem 4 (Update Efficiency). *Suppose construction Π_s is compact. Then construction Π meets update efficiency.*

Proof. Observe that the number of invocations of Upd is at most $\ell + 1 = O(\log |\mathcal{R}|)$ and Upd is only invoked when one of $(\mathsf{hk}_k)_{k\in[0,\ell]}$ is $\perp$. Thus, the number of invocations of Upd in OSig is at most $O(\log |\mathcal{R}|)$.

On the other hand, $|\mathsf{hk}_k| = \mathsf{poly}(\lambda, P, \log |\mathcal{R}|)$ for $k \in [0, \ell]$ according to the compactness of Π_s. Since aux maintains a dictionary $\mathcal{D}_2$ mapping each index slot index k to its set of helper decryption keys, each invocation of Upd runs in $\mathsf{poly}(\log |\mathcal{R}|)$ time (in RAM model). □

Theorem 5 (Unforgeability). *Suppose construction Π_s meets unforgeability. Then construction Π meets unforgeability.*

Proof. Analogous to [HLWW23], suppose that there exists an adversary $\mathcal{A}$ who breaks the unforgeability of Π with non-negligible advantage, then an algorithm $\mathcal{B}$ can be constructed to break the unforgeability of Π_s with non-negligible advantage. Concretely, $\mathcal{B}$ works as follows:

Setup. In the query phase, $\mathcal{B}$ makes as follows:
- guess a number $\delta \in [0, \ell]$ and send 1^{2^δ} to the challenger who returns a common reference string crs_δ;
- initialize $\mathsf{aux} = (\mathcal{D}_1, \mathcal{D}_2, \mathsf{mpk})$, where $\mathcal{D}_1 = \emptyset$, $\mathcal{D}_2 = \emptyset$ and $\mathsf{mpk} = (0, \perp, \ldots, \perp)$; Set $\mathcal{R} = \emptyset$, $C = \emptyset$, $\mathcal{S} = \emptyset$ and a dictionary $\mathcal{K}$ with $\mathcal{K}[\mathsf{pk}] =$ for all possible pk;
- it runs $\mathsf{crs}_k \leftarrow s.\mathsf{Setup}(1^\lambda, P, 1^{2^k})$ for each $k \in [0, \ell] \setminus \{\delta\}$;
- Finally, $\mathcal{B}$ sends $\mathsf{crs} = (\mathsf{crs}_0, \cdots, \mathsf{crs}_\ell)$ to $\mathcal{A}$.

Query. In the query phase, $\mathcal{B}$ simulates the queries $\mathcal{A}$ makes as follows:
- $\mathsf{ORegHK}(y)$: Fetch $\mathsf{mpk} = \big(\mathsf{ctr}, (\mathsf{mpk}_k)_{k\in[0,\ell]}\big)$. For all $k \in [0, \ell]$, compute $i_k = (\mathsf{ctr} \bmod 2^k) + 1$. Run $(\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow s.\mathsf{Gen}(\mathsf{crs}_k, i_k)$ if $k \neq \delta$; otherwise, query $\mathsf{pk}_k \leftarrow s.\mathsf{OGen}(i_k)$ and set $\mathsf{sk}_k = \perp$. Set $\mathsf{ctr}' = \mathsf{ctr}$, $\mathsf{pk} = (\mathsf{ctr}', \mathsf{pk}_0, \cdots, \mathsf{pk}_\ell)$ and $\mathsf{sk} = (\mathsf{ctr}', \mathsf{sk}_0, \cdots, \mathsf{sk}_\ell)$. Then run $(\mathsf{mpk}', \mathsf{aux}') \leftarrow \mathsf{Reg}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}, y)$, where if $i_k = 2^\delta$ in the last step, $\mathcal{B}$ submits $(\mathcal{D}_1[\delta, i])_{i\in[2^\delta]}$ to the challenger who returns $(\mathsf{mpk}_\delta, (\mathsf{hk}_{\delta,j})_{j\in[2^\delta]})$. Update $\mathsf{mpk} = \mathsf{mpk}'$, $\mathsf{aux} = \mathsf{aux}'$, $\mathcal{D}[\mathsf{pk}] = \mathcal{D}[\mathsf{pk}] \cup \{y\}$, append $(\mathsf{pk}, \mathsf{sk})$ to $\mathcal{R}$ and return $(|\mathcal{R}|, \mathsf{mpk}, \mathsf{aux}, \mathsf{pk})$;

– $\mathsf{OCor}(i)$: Let $\mathcal{R}[i] = (\mathsf{pk}, \mathsf{sk})$. Parse $\mathsf{pk} = (\mathsf{ctr}', \mathsf{pk}_0, \cdots, \mathsf{pk}_\ell)$ and secret key $\mathsf{sk} = (\mathsf{ctr}', \mathsf{sk}_0, \cdots, \mathsf{sk}_\ell)$, and query $\mathsf{sk}'_\delta \leftarrow s.\mathsf{OCor}(\mathsf{ctr}', \mathsf{pk}_\delta)$. Then update $\mathsf{sk}_\delta = \mathsf{sk}'_\delta$ along with the secret keys in $\mathcal{R}$ and sk. Append pk to C and return sk;

– $\mathsf{OSig}(i, x, \mathsf{m})$: let $\mathcal{R}[i] = (\mathsf{pk}, \mathsf{sk})$, parse $\mathsf{pk} = \left(\mathsf{ctr}', (\mathsf{pk}_k)_{k\in[0.\ell]}\right)$ and $\mathsf{mpk} = (\mathsf{ctr}, (\mathsf{mpk}_k)_{k\in[0,\ell]})$. Set $\mathsf{hk} = (\mathsf{hk}_k)_{k\in[0,\ell]}$ as in equality (3). For each $k \in [0, \ell]$, compute σ_k as in equality (4) if $k \neq \delta$; otherwise, query $\sigma_k \leftarrow s.\mathsf{OSig}(\mathsf{ctr}, x, \mathsf{m})$. Finally, append (i, x, m) to $\mathcal{S}$ and output $\sigma = (\mathsf{ctr}', \sigma_0, \cdots, \sigma_\ell)$.

Challenge. In the challenge phase, parse $\mathsf{mpk} = (\mathsf{ctr}, (\mathsf{mpk}_k)_{k\in[0,\ell]})$. After receiving the challenge $(i^*, x^*, \mathsf{m}^*, \sigma^*)$ where $\sigma^* = (\mathsf{ctr}^*, (\sigma^*_k)_{k\in[0,\ell]})$, the algorithm $\mathcal{B}$ proceeds as follows:

– Compute $\mathsf{ctr} = (a_\ell, \cdots, a_0)_2$ and $\mathsf{ctr}^* = (b_\ell, \cdots, b_0)_2$. We denote k_d as the maximum $k \in [0, \ell]$ such that $a_k \neq b_k$. If $\delta \neq k_d$, the experiment aborts;

– If $(i^*, x^*, \mathsf{m}^*) \notin \mathcal{S}$ and $\mathsf{pk}_\delta \notin C$, submit $(i^*, x^*, \mathsf{m}^*, \sigma^*_\delta)$ to the challenger.

Since δ is completely independent of $\mathcal{A}$, the above experiment aborts with $1/(\ell+1)$ probability, where $\ell = \log L$. Thus, if $\mathcal{A}$ can break the unforgeability of Π with advantage ϵ, then $\mathcal{B}$ can break the unforgeability of Π_s with advantage $\epsilon/(\ell+1)$. Since we have demonstrated that ϵ is negligible in section 4, the construction Π meets unforgeability. □

6 Concrete Slotted Registered ABS

In this section, we will present a concrete slotted registered ABS for ABP, which derives from the generic scheme in section 4. Note that other classes of predicate (e.g., inner-product, monotone span programs, and so on) can also be achieved in our slotted registered ABS using encodings in [CGW15]. Then we can employ the generic approach in section 5 to obtain the first registered ABS for ABP.

Preliminaries. An arithmetic span program [IW14], denoted by V, is defined by $(\mathbf{Y}, \mathbf{Z}) \in \mathbb{Z}_p^{m\times\ell} \times \mathbb{Z}_p^{m\times\ell}$ where

$$V(\mathbf{x}) = 1 \iff \mathbf{x} \in \mathbb{Z}_p^{1\times m} \text{ satisfies } V \iff \mathbf{e}_1 \in \mathsf{span}\langle \mathsf{diag}(\mathbf{x}) \cdot \mathbf{Y} + \mathbf{Z}\rangle.$$

Here we use notation: $\mathsf{diag}(\mathbf{x}) := \begin{pmatrix} x_1 & & \\ & \ddots & \\ & & x_m \end{pmatrix} \in \mathbb{Z}_p^{m\times m}$ for $\mathbf{x} = (x_1, \ldots, x_m)$ and note that $\mathsf{diag}(\mathbf{x}) = \mathsf{diag}(\mathbf{x})^\top$. And $\mathbf{e}_1 \in \mathsf{span}\langle \mathsf{diag}(\mathbf{x}) \cdot \mathbf{Y} + \mathbf{Z}\rangle$ means that there exists some $\boldsymbol{\omega} \in \mathbb{Z}_p^{1\times m}$ such that $\mathbf{e}_1 = \boldsymbol{\omega}(\mathsf{diag}(\mathbf{x}) \cdot \mathbf{Y} + \mathbf{Z})$

Recall the predicate encoding for ASP predicate (ciphertext-policy variant) in [CGW15]: let $n = 2m + \ell$, $n_c = 2m$ and $n_k = m + 1$, define

$$\mathbf{C}_{\mathbf{Y},\mathbf{Z}} = \begin{pmatrix} \mathbf{I}_m & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & \mathbf{I}_m \\ \mathbf{Y}^\top & \mathbf{Z}^\top \end{pmatrix}, \quad \mathbf{K}_{\mathbf{x}} = \begin{pmatrix} \mathbf{0}_m^\top & \mathsf{diag}(\mathbf{x}) \\ \mathbf{0}_m^\top & \mathbf{I}_m \\ \mathbf{e}_1^\top & \mathbf{0}_{\ell\times m} \end{pmatrix}, \quad \mathbf{a}_{\mathbf{x}} = (1 \| \mathbf{0}_m),$$
$$\mathbf{d}_{\mathbf{x},\mathbf{Y},\mathbf{Z}} = (1 \| \boldsymbol{\omega} \| -\boldsymbol{\omega} \cdot \mathsf{diag}(\mathbf{x}) \| -\boldsymbol{\omega}) \tag{5}$$

where $\mathbf{0}_m$ is a *row* zero vector of size m. Note that we work with *read-once* ASP as in [CGW15].

Scheme. Our concrete slotted registered ABS for read-once ASP from SXDH assumption works as follows:

– $\mathsf{Setup}(1^\lambda, P, 1^L)$: Run $\mathbb{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$ and select a collusion-resistant hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_p$. Sample

$$\mathbf{a} \leftarrow \mathbb{Z}_p^{1\times 2}, \ \mathbf{b}^\top \leftarrow \mathbb{Z}_p^2, \ \mathbf{k} \leftarrow \mathbb{Z}_p^{1\times 2}.$$

For all $i \in [L]$, sample

$$d_i, r_i \leftarrow \mathbb{Z}_p,\ \mathbf{V}_i \leftarrow \mathbb{Z}_p^{2\times 2},\ \mathbf{W}_i \leftarrow \mathbb{Z}_p^{2\times 2(2m+\ell)}.$$

For all $i \in [L]$, write $\mathbf{b}_i^\top = d_i \mathbf{b}^\top$ and output

$$\mathsf{crs} = \begin{pmatrix} [\mathbf{a}]_1, \left\{[\mathbf{a}\mathbf{V}_i, \mathbf{a}\mathbf{W}_i]_1, [\mathbf{b}^\top r_i, \mathbf{V}_i\mathbf{b}^\top r_i + \mathbf{k}^\top, \mathbf{b}_i^\top]_2\right\}_{i\in[L]}, \\ \left\{[\mathbf{V}_i\mathbf{b}^\top r_j, \mathbf{W}_i(\mathbf{I}_{2m+\ell}\otimes \mathbf{b}^\top r_j)]_2\right\}_{j\in[L], i\in[L]\setminus\{j\}}, [\mathbf{a}\mathbf{k}^\top]_T, \mathsf{H} \end{pmatrix}$$

– $\mathsf{Gen}(\mathsf{crs}, i)$: Sample $\mathbf{U}_i, \mathbf{Q}_i, \mathbf{T}_i \leftarrow \mathbb{Z}_p^{2\times 2}$. Fetch $\{\mathbf{b}_i^\top\}_{i\in[L]}$ and $\{[\mathbf{b}^\top r_j]_2\}_{j\in[L]\setminus\{i\}}$ from crs and output

$$\mathsf{pk}_i = ([\mathbf{a}\mathbf{U}_i, \mathbf{a}\mathbf{Q}_i, \mathbf{a}\mathbf{T}_i]_1, \{[\mathbf{U}_i\mathbf{b}^\top r_j, \mathbf{Q}_i\mathbf{b}_j^\top, \mathbf{T}_i\mathbf{b}_j^\top]_2\}_{j\in[L]\setminus\{i\}})$$

and $\mathsf{sk}_i = (\mathbf{U}_i, \mathbf{Q}_i, \mathbf{T}_i)$.

– $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$: Fetch $\{[\mathbf{b}^\top r_j, \mathbf{b}_j^\top]_2\}_{j\in[L]\setminus\{i\}}$ from crs and parse $\mathsf{pk}_i = ([\mathbf{a}\mathbf{U}_i, \mathbf{a}\mathbf{Q}_i, \mathbf{a}\mathbf{T}_i]_1, \{[\mathbf{U}_i\mathbf{b}^\top r_j, \mathbf{Q}_i\mathbf{b}_j^\top, \mathbf{T}_i\mathbf{b}_j^\top]_2\}_{j\in[L]\setminus\{i\}})$. For each $j \in [L]\setminus\{i\}$, check

$$e([\mathbf{a}]_1, [\mathbf{U}_i\mathbf{b}^\top r_j]_2) \stackrel{?}{=} e([\mathbf{a}\mathbf{U}_i]_1, [\mathbf{b}^\top r_j]_2),$$
$$e([\mathbf{a}]_1, [\mathbf{Q}_i\mathbf{b}_j^\top]_2) \stackrel{?}{=} e([\mathbf{a}\mathbf{Q}_i]_1, [\mathbf{b}_j^\top]_2),$$
$$e([\mathbf{a}]_1, [\mathbf{T}_i\mathbf{b}_j^\top]_2) \stackrel{?}{=} e([\mathbf{a}\mathbf{T}_i]_1, [\mathbf{b}_j^\top]_2).$$

If all these checks pass, output 1; otherwise, output 0.

– $\mathsf{Agg}(\mathsf{crs}, (\mathsf{pk}_i, \mathbf{x}_i)_{i\in[L]})$: For all $i \in [L]$, compute $\mathbf{K}_{x_i}$ as in equality (5) and output:

$$\mathsf{mpk} = \begin{pmatrix} [\mathbf{a}]_1, [\mathbf{a}\mathbf{k}^\top]_T, \mathsf{H}, \left[\sum_{j\in[L]} \mathbf{a}\mathbf{Q}_j\right]_1, \left[\sum_{j\in[L]} \mathbf{a}\mathbf{T}_j\right]_1, \left[\sum_{j\in[L]} \mathbf{a}\mathbf{W}_j\right]_1, \\ \left[\sum_{j\in[L]} (\mathbf{a}\mathbf{V}_j + \mathbf{a}\mathbf{U}_j)((1\|\mathbf{0}_m)\otimes \mathbf{I}_2) + \mathbf{a}\mathbf{W}_j\left(\mathbf{K}_{\mathbf{x}_j}\otimes \mathbf{I}_2\right)\right]_1 \end{pmatrix}$$

and for all $i \in [L]$, compute hk_i as

$$\begin{pmatrix} \mathsf{H}, [\mathbf{b}_i^\top]_2, [\mathbf{b}^\top r_i]_2, [\mathbf{V}_i\mathbf{b}^\top r_i + \mathbf{k}^\top]_2, \left[\sum_{j\in[L]\setminus\{i\}} \mathbf{Q}_j\mathbf{b}_i^\top\right]_2, \\ \left[\sum_{j\in[L]\setminus\{i\}} \mathbf{T}_j\mathbf{b}_i^\top\right]_2, \left[\sum_{j\in[L]\setminus\{i\}} \mathbf{W}_j(\mathbf{I}_{2m+\ell}\otimes \mathbf{b}^\top r_i)\right]_2, \\ \left[\sum_{j\in[L]\setminus\{i\}} (\mathbf{V}_j\mathbf{b}^\top r_i + \mathbf{U}_j\mathbf{b}^\top \mathbf{r}_i)(1\|\mathbf{0}_m) + \mathbf{W}_j(\mathbf{I}_{2m+\ell}\otimes \mathbf{b}^\top r_i)\mathbf{K}_{\mathbf{x}_j}\right]_2 \end{pmatrix}.$$

– $\mathsf{Sig}(\mathsf{hk}_i, \mathsf{sk}_i, (\mathbf{Y}, \mathbf{Z}), \mathsf{m})$: Sample $t \leftarrow \mathbb{Z}_p$, run $h \leftarrow \mathsf{H}(i, \mathsf{m}, (\mathbf{Y}, \mathbf{Z}))$ and compute $\mathbf{C}_{\mathbf{Y},\mathbf{Z}}$ as in equality (5). Parse $\mathsf{sk}_i = (\mathbf{U}_i, \mathbf{Q}_i, \mathbf{T}_i)$, then compute $[\mathbf{k}_0^\top]_2 = [\mathbf{b}_i^\top t]_2$, $[\mathbf{k}_1^\top]_2 = [\mathbf{b}^\top r_i]_2$ and

$$[\mathbf{k}_2^\top]_2 = [\mathbf{V}_i\mathbf{b}^\top r_i + \mathbf{k}^\top + \mathbf{U}_i\mathbf{b}^\top r_i + (\mathbf{Q}_i\mathbf{b}_i^\top t + h\cdot \mathbf{T}_i\mathbf{b}_i^\top t)]_2,$$
$$[\mathbf{K}_3]_2 = \left[\sum_{j\in[L]\setminus\{i\}} (\mathbf{V}_j\mathbf{b}^\top r_i + \mathbf{U}_j\mathbf{b}^\top \mathbf{r}_i)(1\|\mathbf{0}_m) + \mathbf{W}_j(\mathbf{I}_{2m+\ell}\otimes \mathbf{b}^\top r_i)\mathbf{K}_{\mathbf{x}_j}\right]_2,$$
$$[\mathbf{K}_4]_2 = \left[\sum_{j\in[L]\setminus\{i\}} \mathbf{W}_j(\mathbf{I}_{2m+\ell}\otimes \mathbf{b}^\top r_i)\mathbf{C}_{\mathbf{Y},\mathbf{Z}}\right]_2,$$
$$[\mathbf{k}_5^\top]_2 = \left[\sum_{j\in[L]\setminus\{i\}} (\mathbf{Q}_j\mathbf{b}_i^\top t + h\cdot \mathbf{T}_j\mathbf{b}_i^\top t)\right].$$

Output $\sigma_{i,(\mathbf{Y},\mathbf{Z}),\mathsf{m}} = ([\mathbf{k}_0^\top]_2, [\mathbf{k}_1^\top]_2, [\mathbf{k}_2^\top]_2, [\mathbf{K}_3]_2, [\mathbf{K}_4]_2, [\mathbf{k}_5^\top]_2)$.

– $\mathsf{Ver}(\mathsf{mpk}, \sigma_{i^*,(\mathbf{Y},\mathbf{Z}),\mathsf{m}}, (\mathbf{Y},\mathbf{Z}), \mathsf{m})$: Sample $s \leftarrow \mathbb{Z}_p$ and run $h^* \leftarrow \mathsf{H}(i^*, \mathsf{m}^*, (\mathbf{Y},\mathbf{Z}))$. Compute

$$[\mathbf{v}_0]_1 = [s\mathbf{a}]_1,\ [\mathbf{v}_1]_1 = \left[\sum_{j\in[L]} (s\mathbf{a}\mathbf{Q}_j + h^* \cdot s\mathbf{a}\mathbf{T}_j)\right]_1,$$
$$[\mathbf{v}_2]_1 = \left[\sum_{j\in[L]} s\mathbf{a}\mathbf{W}_j\left(\mathbf{C}_{\mathbf{Y},\mathbf{Z}} \otimes \mathbf{I}_2\right)\right]_1$$
$$[\mathbf{v}_3]_1 = \left[\sum_{j\in[L]} (s\mathbf{a}\mathbf{V}_j + s\mathbf{a}\mathbf{U}_j)((1\|\mathbf{0}_m) \otimes \mathbf{I}_2) + s\mathbf{a}\mathbf{W}_j(\mathbf{K}_{\mathbf{x}_j} \otimes \mathbf{I}_2)\right]_1,$$

and $[v_4]_T = [\mathbf{s}\mathbf{A}\mathbf{k}^\top]_T$. Parse $\sigma_{i^*,(\mathbf{Y},\mathbf{Z}),\mathsf{m}} = ([\mathbf{k}_0^\top]_2, [\mathbf{k}_1^\top]_2, [\mathbf{k}_2^\top]_2, [\mathbf{K}_3]_2, [\mathbf{K}_4]_2, [\mathbf{k}_5^\top]_2)$ and compute $\boldsymbol{\omega}$ such that $\mathbf{e}_1 = \boldsymbol{\omega}(\mathsf{diag}(\mathbf{x}_{i^*}) \cdot \mathbf{Y} + \mathbf{Z})$. Then recover

$$[\mathbf{z}]_T = e([\mathbf{v}_3\|\mathbf{v}_2]_1, [\mathbf{I}_{3m+1} \otimes \mathbf{k}_1^\top]_2),\ [\mathbf{z}_2]_T = e([\mathbf{v}_0]_1, [\mathbf{K}_3\|\mathbf{K}_4]_2)$$
$$[z_3]_T = e([\mathbf{v}_0]_1, [\mathbf{k}_2^\top]_2),\ [z_4]_T = e([\mathbf{v}_1]_1, [\mathbf{k}_0^\top]_2),\ [z_5]_T = e([\mathbf{v}_0]_1, [\mathbf{k}_5^\top]_2),$$
$$[z_6]_T = [z_3 - z_4 + z_5]_T,\ [z_7]_T = [(\mathbf{z}_1 - \mathbf{z}_2)(1\|\boldsymbol{\omega}\| - \boldsymbol{\omega} \cdot \mathsf{diag}(\mathbf{x})\| - \boldsymbol{\omega})^\top - z_6]_T$$

and check $[z_7]_T^{-1} \stackrel{?}{=} [v_4]_T$. If the above check passes, output 1; otherwise, output 0.

Acknowledgements. This work was supported in part by National Natural Science Foundation of China (61972156, 62372180,62002120,62372175), NSFC-ISF Joint Scientific Research Program (61961146004), Innovation Program of Shanghai Municipal Education Commission (2021-01-07-00-08-E00101) and the "Digital Silk Road" Shanghai International Joint Lab of Trustworthy Intelligent Software (22510750100).

References

ABS17. Miguel Ambrona, Gilles Barthe, and Benedikt Schmidt. Generic transformations of predicate encodings: Constructions and applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 36–66. Springer, Heidelberg, August 2017. 9

ACGU20. Michel Abdalla, Dario Catalano, Romain Gay, and Bogdan Ursu. Inner-product functional encryption with fine-grained access control. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 467–497. Springer, Heidelberg, December 2020. 9

AHY15. Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. Conversions among several classes of predicate encryption and applications to abe with various compactness tradeoffs. In *Advances in Cryptology–ASIACRYPT 2015: 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29–December 3, 2015, Proceedings, Part I 21*, pages 575–601. Springer, 2015. 2, 3

BF01. Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001. 4

BLS01. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *International conference on the theory and application of cryptology and information security*, pages 514–532. Springer, 2001. 4

CGW15. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, April 2015. 2, 3, 6, 9, 18, 24

CLL+14. Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter identity-based encryption via asymmetric pairings. *Designs, codes and cryptography*, 73:911–947, 2014. 4, 5

DDM23. Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Short attribute-based signatures for arbitrary turing machines from standard assumptions. *Des. Codes Cryptogr.*, 91(5):1845–1872, 2023. 1

DH76. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976. 1

DOT19a. Pratish Datta, Tatsuaki Okamoto, and Katsuyuki Takashima. Efficient attribute-based signatures for unbounded arithmetic branching programs. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 127–158. Springer, Heidelberg, April 2019. 1

DOT19b. Pratish Datta, Tatsuaki Okamoto, and Katsuyuki Takashima. Efficient attribute-based signatures for unbounded arithmetic branching programs. In *Public-Key Cryptography–PKC 2019: 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part I 22*, pages 127–158. Springer, 2019. 2, 3

EHK+13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013. 7

FFM+23. Danilo Francati, Daniele Friolo, Monosij Maitra, Giulio Malavolta, Ahmadreza Rahimi, and Daniele Venturi. Registered (inner-product) functional encryption. *Cryptology ePrint Archive*, 2023. 2

GHMR18. Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ahmadreza Rahimi. Registration-based encryption: Removing private-key generator from IBE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 689–718. Springer, Heidelberg, November 2018. 2

HLLR12. Javier Herranz, Fabien Laguillaumie, Benoît Libert, and Carla Rafols. Short attribute-based signatures for threshold predicates. In *Cryptographers' Track at the RSA Conference*, pages 51–67. Springer, 2012. 3

HLWW23. Susan Hohenberger, George Lu, Brent Waters, and David J. Wu. Registered attribute-based encryption. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 511–542. Springer, Heidelberg, April 2023. 2, 3, 4, 21, 23

IW14. Yuval Ishai and Hoeteck Wee. Partial garbling schemes and their applications. In *International Colloquium on Automata, Languages, and Programming*, pages 650–662. Springer, 2014. 24

MPR08. Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. *IACR Cryptol. ePrint Arch.*, page 328, 2008. 1, 3

OT11. Tatsuaki Okamoto and Katsuyuki Takashima. Efficient attribute-based signatures for non-monotone predicates in the standard model. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 35–52. Springer, Heidelberg, March 2011. 1, 2, 3, 4, 5, 6

OT13. Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized attribute-based signatures. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 125–142. Springer, Heidelberg, February / March 2013. 1, 2, 4, 6

SAH16a. Yusuke Sakai, Nuttapong Attrapadung, and Goichiro Hanaoka. Attribute-based signatures for circuits from bilinear map. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 283–300. Springer, Heidelberg, March 2016. 1

SAH16b. Yusuke Sakai, Nuttapong Attrapadung, and Goichiro Hanaoka. Attribute-based signatures for circuits from bilinear map. In *Public-Key Cryptography–PKC 2016: 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part I*, pages 283–300. Springer, 2016. 2, 3

SKAH18. Yusuke Sakai, Shuichi Katsumata, Nuttapong Attrapadung, and Goichiro Hanaoka. Attribute-based signatures for unbounded languages from standard assumptions. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 493–522. Springer, Heidelberg, December 2018. 6

Wee14. Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Heidelberg, February 2014. 2, 3, 9

ZZGQ23. Ziqi Zhu, Kai Zhang, Junqing Gong, and Haifeng Qian. Registered abe via predicate encodings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 66–97. Springer, 2023. 2, 3, 4, 9, 14, 16

Table of Contents