Registered Functional Encryption for Quadratic Functions from MDDH

Qiaohan Chu[1], Li Lin[2], Chen Qian[3], and Jie Chen[1(✉)]

[1] Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai, China
s080001@e.ntu.edu.sg
[2] Ant Group, China
[3] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China

Abstract. We present a Registered Functional Encryption (RFE) scheme for inner product and a RFE scheme for quadratic functions based on pairings and relying on the Matrix Decision Diffie-Hellman (MDDH) assumption and bilateral MDDH assumption. Previously, RFE is only known to be constructed from indistinguishability obfuscation (iO) in Francati-Friolo-Maitra-Malavolta-Rahimi-Venturi [Asiacrypt '23].

Keywords: Registered Encryption · Functional Encryption · Quadratic Functions.

1 Introduction

Functional Encryption (FE) [10,33] is an encryption that allows evaluation on ciphertexts and meanwhile gets rid of "all-or-nothing". Concretely, in an FE scheme, given a ciphertext associated with a message x, the decryption can only recover $f(x)$ but nothing else, with a secret key associated with a function f. There have been a plenty of works focusing on FE constructions in recent years, which can be roughly categoried into "general" style [20,11,34,8,6,7,30,27], and "practical" style [3,5,35,9,4,36,23,25].

In FE, there is a central entity holding a master secret key, which is crucial in preserving the privacy of messages and issuing the secret keys. An important problem of FE is that if the central entity is compromised by the adversary, then the adversary can totally destroy the security of the system. Such a problem is called *key-escrow* problem.

To tackle the key-escrow problem, recently, a line of research has made efforts in constructing Registered Identity-Based Encryption (RIBE) [21,22,26,16,24], Registered Attribute-Based Encryption (RABE) [31,29,18,19], and Registered Functional Encryption (RFE) [18]. The above encryptions can be subsumed into Registered Encryption. In Registered Encryption, each user generates his own public/secret key pair, and the central entity is replaced with a semi-honest key curator that does not hold any secret information and is in charge of aggregating public information (e.g., public keys from registered users).

For Registered Encryption, there are some efficiency requirements that for L users totally, each user only needs to update their decryption key at most $O(\log L)$ times over the lifetime of the system, and the size of master public key and each update should be bounded by $\mathsf{poly}(\lambda, \log L)$, where λ is the security parameter and poly is a universal polynomial.

Recently, Francati, Friolo, Maitra, Malavolta, Rahimi, and Venturi [18] presented a feasible construction of RFE supporting all circuits from indistinguishability obfuscation (iO). This construction lies more in "general" style, which means that it seems far from being practical. Thus, from practical side, we seek for RFE constructions lying more in "practical" style. This motivation is analogous to the motivation of practical FE [3,5].

1.1 Contributions

We present two practical RFE schemes:

- A Registered Inner Product Functional Encryption (RIPFE) scheme relying on the Matrix Decision Diffie-Hellman (MDDH) assumption. Our RIPFE scheme achieves weakly selective-IND security (c.f. Definition 4) and weakly selective-SIM security (c.f. Definition 5). The efficiency properties are as below:
  - the size of common reference string is $L \cdot n \cdot \mathsf{poly}(\lambda, |\mathcal{F}|, \log L)$;
  - the running time of key generation and registration is $L \cdot n \cdot \mathsf{poly}(\lambda, |\mathcal{F}|, \log L)$ and $L \cdot n \cdot \mathsf{poly}(\lambda, |\mathcal{F}|, \log L) + L^2 \cdot n \cdot \mathsf{poly}(\lambda)$ respectively;
  - the size of master public key and helper decryption key is bounded by $\log L \cdot n \cdot \mathsf{poly}(\lambda)$ and $\log L \cdot n \cdot \mathsf{poly}(\lambda) + \log L \cdot O(\log L)$ respectively;
  - the number of updates in the system is at most $O(\log L)$;
  - the update operation can be implemented in $\log L \cdot n \cdot \mathsf{poly}(\lambda) + \log L \cdot O(\log L)$ time in the RAM model of computation,

  where L is the a-priori fixed number of users, n is the length of the message, λ is the security parameter, $|\mathcal{F}|$ is the size of the function space, and poly is a universal polynomial.

  Our RIPFE supports a bounded number of functions in the system. We stress that this is something inherent to the inner product functionality itself. Similar limitation also exists in Inner Product Functional Encryption (IPFE).
- A Registered Quadratic Functional Encryption (RQFE) scheme relying on the MDDH assumption and the bilateral MDDH (bi-MDDH) assumption. Our RQFE scheme achieves weakly selective-IND security and weakly selective-SIM security. The efficiency properties are as below:
  - the size of common reference string is $L \cdot (n+1) \cdot \mathsf{poly}(\lambda, |\mathcal{F}|, \log L)$;
  - the running time of key generation and registration is $L \cdot (n+1) \cdot \mathsf{poly}(\lambda, |\mathcal{F}|, \log L)$ and $L \cdot (n+1) \cdot \mathsf{poly}(\lambda, |\mathcal{F}|, \log L) + L^2 \cdot (n+1) \cdot \mathsf{poly}(\lambda)$ respectively;
  - the size of master public key and helper decryption key is bounded by $\log L \cdot (n+1) \cdot \mathsf{poly}(\lambda)$ and $\log L \cdot (n+1) \cdot \mathsf{poly}(\lambda) + \log L \cdot n^2 \cdot \mathsf{poly}(\lambda) + \log L \cdot O(\log L)$ respectively;

- the number of updates in the system is at most $O(\log L)$;
- the update operation can be implemented in $\log L \cdot (n+1) \cdot \mathsf{poly}(\lambda) + \log L \cdot n^2 \cdot \mathsf{poly}(\lambda) + \log L \cdot O(\log L)$ time in the RAM model of computation,

where n^2 is the length of the message.

Both of our schemes are imposed an a-priori fixed number of users, as the previous pairing-based Registered Encryption schemes. And for our constructions, when the functions of the users that are corrupted by the adversary cover the entire function space, weakly selective-SIM security is equivalent to selective-SIM security.

1.2 Related Works

Previous works that tried to resolve the key-escrow problem of FE mainly concentrated on constructing decentralized FE, where the central trust is distributed to many entities. It should be noted that decentralized FE is in the private key setting, while FE can be in the public key setting. Chotard, Sans, Gay, Phan, and Pointcheval [15] introduced the notion of decentralized Multi-Client Functional Encryption (DMCFE). In DMCFE, secret keys are issued by multiple entities instead of a central entity. Till now, a sequence of DMCFE schemes have been presented [15,32,2,1]. Chotard, Sans, Gay, Phan, and Pointcheval [14] introduced another notion of dynamic decentralized Functional Encryption (DDFE), and presented constructions for various functionalities. The major difference between DDFE and DMCFE is that DDFE allows multiple users to join the system dynamically, while DMCFE has no easy way to do this.

Decentralized FE mitigates the reliance on the central trusted entity, but the secret keys are still issued by the entities rather than the users. If a sufficient number of entities are compromised, the system will be insecure.

2 Technical Overview

Starting Point. We start from the observation that the pairing-based slotted RABE scheme in [29] is roughly a combination of a centralized Attribute-Based Encryption (ABE) and a traditional public key encryption. Thus, to construct slotted RFE schemes, we try the way of combining a centralized FE and a traditional public key encryption. Since as stated in [18], an RFE can be transformed from a slotted RFE following the generic compiler in [29], our task can be mainly set to constructing the slotted RFE schemes. Commonly, we start with the inner product setting.

Slotted Registered Inner Product Functional Encryption. We use the IPFE scheme in [35] as the underlying centralized FE scheme. We provide a brief description of this IPFE scheme:

$$\begin{aligned}
\mathsf{crs} =&([\mathbf{A}]_1, [\mathbf{AV}]_1);\\
\mathsf{ct} =&(C_1 = [\mathbf{sA}]_1, C_2 = [\mathbf{sAV} + \mathbf{x}]_1);\\
\mathsf{sk} =&([\mathbf{Vf}]_2);\\
\mathsf{Dec} :\ & e(C_2, [\mathbf{f}]_2) \cdot e(C_1, \mathsf{sk})^{-1} = [\mathbf{sAVf} + \mathbf{xf} - \mathbf{sAVf}]_T = [\mathbf{xf}]_T.
\end{aligned}$$

As [29], we first put the sk in the crs, thus eliminating the central entity for issuing the sk. Then, from mathematical side, we observe that the key point to recover $[\mathbf{xf}]_T$ is to obtain $[\mathbf{sAVf}]_T$. Thus, to embed the public key encryption, our idea is to add an extra term regarding the public key encryption to $[\mathbf{sAVf}]_T$, so that obtaining $[\mathbf{sAVf}]_T$ requires obtaining the term regarding the public key encryption first.

The term regarding the public key encryption in our scheme follows a similar algebraic structure in [29], i.e., the aggregation of public keys attaching to a user slot. Concretely, it is of the form as $[\mathbf{sA}\sum_t \mathbf{U}_t\mathbf{W}_i\mathbf{f}_i]_T$ (as $[\mathbf{sAVf}]_T$, this is reflected in the decryption). And we require the public key of each user only leaks information about $[\mathbf{U}_i\mathbf{W}_t\mathbf{f}_j]_2$ for any $t \neq i$, so that only the user i himself can subtract $[\mathbf{sAU}_i\mathbf{W}_i\mathbf{f}_i]_T$ and thus subtract $[\mathbf{sA}\sum_t \mathbf{U}_t\mathbf{W}_i\mathbf{f}_i]_T$, with his secret key $\mathbf{U}_i$. At this time, our draft is as below:

$$\begin{aligned}
\mathsf{crs} =&([\mathbf{A}]_1, [\mathbf{AV}]_1, [\mathbf{AV}_1]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i,j}, \{[\mathbf{V}_1\mathbf{W}_i\mathbf{f} + \mathbf{Vf}_j]_2\}_{i,j});\\
\mathsf{pk}_i =&([P^2_{i,t,j}]]_1 = [\mathbf{AU}_i]_1, \{[P^2_{i,t,j}]_2\} = \{[\mathbf{U}_i\mathbf{W}_t\mathbf{f}_j]_2\}_{i\neq t,j});\\
\mathsf{sk}_i =&\mathbf{U}_i;\\
\mathsf{mpk} =&([\mathbf{A}\sum_i \mathbf{U}_i]_1, [\mathbf{A}]_1, [\mathbf{AV}]_1, [\mathbf{AV}_1]_1);\\
\mathsf{hsk}_i =&(i, \mathbf{f}_i, [\sum_{t\neq i} \mathbf{U}_t\mathbf{W}_i\mathbf{f}_i]_2, [\mathbf{V}_1\mathbf{W}_i\mathbf{f}_i + \mathbf{Vf}_i]_2, [\mathbf{W}_i\mathbf{f}_i]_2);\\
\mathsf{ct} =&(C_1 = [\mathbf{sA}]_1, C_2 = [\mathbf{sAV} + \mathbf{x}]_1, C_3 = [\mathbf{sAV}_1 + \mathbf{sA}\sum_t \mathbf{U}_t]_1).
\end{aligned}$$

In the above draft, $\mathbf{V}_1$ is used to link $\mathbf{V}$ and $\sum_t \mathbf{U}_t$.

It seems that we have almost accomplished our task, except validating the pk_i. However, note that in light of the syntax of slotted RFE (c.f. Section 3.6), the registered $\{\mathbf{f}_i\}_i$ should be embedded in the mpk and thus in the ct. To do this, since the aggregation of mpk takes only public information, thus all we can utilize to embed the registered $\{\mathbf{f}_i\}_i$ are $\{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i,j}$ and $\{[\mathbf{V}_1\mathbf{W}_i\mathbf{f} + \mathbf{Vf}_j]_2\}_{i,j}$ from the crs. Observe that these are in G_2, thus we cannot carry over the above method of "sticking an extra term to $[\mathbf{sAVf}]_T$" by adding them to C_2 or C_3 in the ct, since C_2 and C_3 are in G_1. Instead, we use an independent term $[\mathbf{sAV}_2\sum_t \mathbf{W}_t\mathbf{f}_t]_T$ to embed the registered $\{\mathbf{f}_i\}_i$. To link $[\mathbf{sAV}_2\sum_t \mathbf{W}_t\mathbf{f}_t]_T$ with the above draft, we replace C_2 in the ct with $[\mathbf{sAV} + \mathbf{x} + \eta \cdot (1, 0, ..., 0)]_1$, and add $C_4 = [\mathbf{sAV}_2]_1, C_5 = [\mathbf{sAV}_2\sum_t \mathbf{W}_t\mathbf{f}_t - \eta]_T$ to the ct. C_5 and the replaced C_2 are used to ensure that $[\mathbf{sAV}_2\sum_t \mathbf{W}_t\mathbf{f}_t]_T$ works, C_4 is used to cancel $[\mathbf{sAV}_2\sum_t \mathbf{W}_t\mathbf{f}_t]_T$.

Note that there is a minor flaw that $\eta \cdot (1, 0, ..., 0)\mathbf{f}_i$ might not equal η (if the first component of $\mathbf{f}_i$ is not equal to 1), so that $[-\eta]_T$ in C_5 cannot cancel out $[\eta \cdot (1, 0, ..., 0)\mathbf{f}_i]_T$. To tackle this, in the course of encryption and decryption, we change $\mathbf{f}_i$ into $\mathbf{f}'_i$, where $\mathbf{f}_i = \alpha_i \mathbf{f}'_i$ and the first component of $\mathbf{f}'_i$ is 1. Since $\mathbf{f}_i$ is public, this can be achieved without sacrificing the privacy. As for the correctness, we can simply multiply α_i to the decryption output in the exponent. Then our draft becomes:

$$\begin{aligned}
\mathsf{crs} =&([\mathbf{A}]_1, [\mathbf{AV}]_1, [\mathbf{AV}_1]_1, [\mathbf{AV}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i,j}, \{[\mathbf{V}_1\mathbf{W}_i\mathbf{f} + +\mathbf{V}_2\mathbf{W}_j\mathbf{f}_j + \mathbf{V}\mathbf{f}_j]_2\}_{i,j});\\
\mathsf{pk}_i =&([P_i^1]_1 = [\mathbf{AU}_i]_1, \{[P_{i,t,j}^2]_2\} = \{[\mathbf{U}_i\mathbf{W}_t\mathbf{f}_j]_2\}_{i \neq t,j});\\
\mathsf{sk}_i =&\mathbf{U}_i;\\
\mathsf{mpk} =&([\mathbf{A}\sum_i \mathbf{U}_i]_1, [\sum_i \mathbf{W}_i\mathbf{f}'_i]_2, [\mathbf{A}]_1, [\mathbf{AV}]_1, [\mathbf{AV}_1]_1, [\mathbf{AV}_2]_1);\\
\mathsf{hsk}_i =&(i, \mathbf{f}_i, [\sum_{t \neq i} \mathbf{U}_t\mathbf{W}_i\mathbf{f}'_i]_2, [\mathbf{V}_1\mathbf{W}_i\mathbf{f}'_i + \mathbf{V}_2\mathbf{W}_i\mathbf{f}'_i + \mathbf{V}\mathbf{f}'_i]_2, [\mathbf{W}_i\mathbf{f}'_i]_2);\\
\mathsf{ct} =&(C_1 = [\mathbf{sA}]_1, C_2 = [\mathbf{sAV} + \mathbf{x} + \eta \cdot (1, 0, ..., 0)]_1, C_3 = [\mathbf{sAV}_1 + \mathbf{sA}\sum_t \mathbf{U}_t]_1,\\
&C_4 = [\mathbf{sAV}_2]_1, C_5 = [\mathbf{sAV}_2 \sum_t \mathbf{W}_t\mathbf{f}'_t - \eta]_T).
\end{aligned}$$

As for the validity of the pk_i, note that for the correctness, it is sufficed to ensure that $e(P_i^1, [\mathbf{W}_t\mathbf{f}_j]_2) = e([\mathbf{A}]_1, [P_{i,t,j}^2]_2)$ for each $t \neq i$ and each j. As long as the equations hold, the decryption will proceed successfully. And for the security, we make the secret key $\mathbf{U}_i$ enabled to be extracted by bounding its range, so that the validity of the pk_i in the security can be easily checked. We leave the details of extracting the secret key in the following section for security.

Then we obtain our slotted RIPFE scheme as in Section 4.1. Overall, we first combine the centralized FE scheme [35] and the traditional public key encryption, next we embed the registered $\{\mathbf{f}_i\}_i$ into the mpk, then we add a validation mechanism for the public keys, eventually we construct a slotted RIPFE scheme satisfying the syntax of slotted RIPFE.

Slotted Registered Quadratic Functional Encryption. For the slotted RQFE, we use the Quadratic Functional Encryption (QFE) scheme in [36] as the underlying centralized FE scheme. This QFE scheme is based on the IPFE scheme in [35], which is used by us for constructing our slotted RIPFE scheme. Thus, a preliminary blueprint of our slotted RQFE seems to follow the framework of QFE in [36], i.e., using our slotted RIPFE as a building block in the black-box manner. However, a halfway issue is that "the cancel of η" doesn't work any more, if applying our slotted RIPFE to our slotted RQFE in a straight-forward manner. Inspired by [18], we tackle this by modifying the encrypted message $(\mathbf{x}_1, \mathbf{x}_2)$ into $(\mathbf{x}_1 \| \eta_1, \mathbf{x}_2 \| \eta_2)$, and modifying the function $\mathbf{f}_i$ into $\hat{\mathbf{f}}_i$ (c.f. equation (2) to see $\hat{\mathbf{f}}_i$), so that $((\mathbf{x}_1 \| \eta_1) \otimes (\mathbf{x}_2 \| \eta_2))\hat{\mathbf{f}}_i = (\mathbf{x}_1 \otimes \mathbf{x}_2)\mathbf{f}_i + \eta_1 \cdot \eta_2$. Then we can cancel out $[\eta_1 \cdot \eta_2]_T$ by replacing η in "C_5" (in our slotted RIPFE) with $\eta_1 \cdot \eta_2$.

Security. Different from the existing pairing-based Registered Encryption schemes, which all achieve adaptive-IND security, our RFE schemes achieve security in weaker flavors, i.e., weakly selective-IND security, or weakly selective-SIM security. Roughly speaking, weakly selective-IND security is almost the same as selective-IND security, except that we require that for any function $\mathbf{f}$ in the function space, $\mathbf{f}(\mathbf{x}_0^*) = \mathbf{f}(\mathbf{x}_1^*)$, where $\mathbf{x}_0^*, \mathbf{x}_1^*$ are the challenge messages, while selective-IND security only requires the function $\mathbf{f}$ involved in the secret key queries to satisfy $\mathbf{f}(\mathbf{x}_0^*) = \mathbf{f}(\mathbf{x}_1^*)$; weakly selective-SIM security allows the adversary to obtain more function values in the final simulation game but without revealing the encrypted message. Weakly selective-SIM security seems to be acceptable, since in centralized FE, under the precondition "not revealing the message", the adversary can query secret keys as many as possible, which implies that the adversary can obtain function values as many as possible. Further, for our constructions, when the functions of the users that are corrupted by the adversary cover the entire function space, weakly selective-SIM security is equivalent to selective-SIM security.

Our security analysis roughly follows the proof ideas of IPFE in [35], QFE in [36], and ABE in [12,13]. That is, we first utilize the computational indistinguishability from the MDDH assumption to transit the real game into a game convenient for statistical indistinguishability, which relies on the "orthogonality", then we switch the challenge message in a sequence of steps.

However, a "rough blueprint" usually encounters some "unnoticeable" and tough challenges. In our security analysis, a notable challenge of our slotted RIPFE is that when programming the MDDH assumption, the challenger needs to know the secret key $\mathbf{U}_i$ determined by the adversary (which is not known directly by the challenger), so that the challenger can simulate the challenge ct. To tackle this, we require that each secret key $\mathbf{U}_i$ is sampled over a bounded range (i.e., the range of brute-force discrete log for the evaluated function value). Since the challenger knows the master secret key (the master secret key only appears in the security analysis, not in the real construction), thus the challenger can obtain $[\mathbf{U}_i]_T$ with the master secret key, and then extract $\mathbf{U}_i$ by brute-force discrete log. If $\mathbf{U}_i$ cannot be computed or $[\mathbf{U}_i]_T$ is not unique (with negligible probability), we require the challenger to abort.

A similar challenge also appears in the security analysis of our slotted RQFE. However, for our slotted RQFE, when programming the bi-MDDH assumption and the MDDH assumption, carrying out the extraction method of our slotted RIPFE in a straight-forward manner would be failed. This is because at this time, the challenger lacks the matrices regarding the assumptions (the reduction is in the black-box manner), and thus fails in computing $[\mathbf{U}_i]_T$. To tackle this, we require each pk_i to extra include some auxiliary information, which can be utilized by the challenger to compute $[\mathbf{U}_i]_T$ and further to compute $\mathbf{U}_i$.

3 Preliminaries

3.1 Notations

We use negl to denote a negligible function in the security parameter $\lambda \in \mathbb{N}$, use $\leftarrow_R$ to denote random sampling, use $\mathbf{0}$ to denote a zero matrix of proper size, use $\mathbf{I}_n$ to denote an identity matrix of size $n \times n$, where $n \in \mathbb{N}$, and use $\|$ to denote concatenation of matrices. For an integer N, we use $[N]$ to denote the set $\{1, ..., N\}$.

3.2 Prime-Order Bilinear Groups

A prime-order group generator $\mathcal{G}$ takes as input the security parameter λ in unary notation and outputs a description $\mathbb{G} = (p, G_1, G_2, G_T, e)$, where p is a prime, G_1, G_2, G_T are cyclic groups of order p, and $e : G_1 \times G_2 \to G_T$ is an asymmetric non-degenerated bilinear mapping. Let $[1]_1 = g_1 \in G_1, [1]_2 = g_2 \in G_2$ and $[1]_T = g_T = e(g_1, g_2) \in G_T$ be the respective generators. For any $a, b \in \mathbb{Z}_p$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab} = g_T^{ab} = [ab]_T$. We define $[\mathbf{M}]_1 = g_1^{\mathbf{M}}, [\mathbf{M}]_2 = g_2^{\mathbf{M}}$ and $[\mathbf{M}]_T = g_T^{\mathbf{M}}$, where $\mathbf{M}$ is a matrix over $\mathbb{Z}_p$, and exponentiation is carried out component-wise. We also define $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{AB}]_T$, where $\mathbf{A}, \mathbf{B}$ are matrices over $\mathbb{Z}_p$.

3.3 Matrix Diffie-Hellman Assumption

Let $k, l, d \in \mathbb{N}$. The Matrix Decision Diffie-Hellman (MDDH) assumption [17] says that for all $\mathsf{p.p.t}$ adversaries $\mathcal{A}$, the following advantage function $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{MDDH}_{k,l}^d}(\lambda)$ is negligible in λ:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{MDDH}_{k,l}^d}(\lambda) := |\mathsf{Pr}[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_1, \boxed{[\mathbf{MS}]_1}) = 1] - \mathsf{Pr}[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_1, \colorbox{lightgray}{$[\mathbf{U}]_1$}) = 1]|,$$

where $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda), \mathbf{M} \leftarrow \mathbb{Z}_p^{l \times k}, \mathbf{S} \leftarrow \mathbb{Z}_p^{k \times d}$, and $\mathbf{U} \leftarrow \mathbb{Z}_p^{l \times d}$.

MDDH assumption also holds similarly in G_2.

3.4 Bilateral Matrix Diffie-Hellman Assumption

Let $k, l, d \in \mathbb{N}$. The bilateral Matrix Decision Diffie-Hellman (bi-MDDH) assumption says that for all $\mathsf{p.p.t}$ adversaries $\mathcal{A}$, the following advantage function $\mathsf{Adv}_{\mathcal{A}}^{\text{bi-MDDH}_{k,l}^d}(\lambda)$ is negligible in λ:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\text{bi-MDDH}_{k,l}^d}(\lambda) := |&\mathsf{Pr}[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_1, [\mathbf{M}]_2, \boxed{[\mathbf{MS}]_1}, \boxed{[\mathbf{MS}]_2}) = 1] - \\ &\mathsf{Pr}[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_1, [\mathbf{M}]_2, \colorbox{lightgray}{$[\mathbf{U}]_1$}, \colorbox{lightgray}{$[\mathbf{U}]_2$}) = 1]|,\end{aligned}$$

where $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda), \mathbf{M} \leftarrow \mathbb{Z}_p^{l \times k}, \mathbf{S} \leftarrow \mathbb{Z}_p^{k \times d}$, and $\mathbf{U} \leftarrow \mathbb{Z}_p^{l \times d}$.

3.5 Registered Functional Encryption

Syntax. Suppose a message space $\mathcal{M}$, and a function space $\mathcal{F} = \{f_u : \mathcal{M} \to \mathcal{Y}\}$. A Registered Functional Encryption scheme consists of the following six algorithms, where Setup, KeyGen, Enc are randomized algorithms, and RegPK, Update, Dec are deterministic algorithms:

- $\mathsf{Setup}(1^\lambda, 1^{|\mathcal{F}|}) \to \mathsf{crs}$: This algorithm takes as input the security parameter λ in unary notation, and the size of the function space $|\mathcal{F}|$. Output a common reference string crs.
- $\mathsf{KeyGen}(\mathsf{crs}, \mathsf{aux}) \to (\mathsf{pk}, \mathsf{sk})$: This algorithm takes as input the crs, and a (possibly empty) state aux. Output a public key pk, and a secret key sk.
- $\mathsf{RegPK}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}, f) \to (\mathsf{mpk}, \mathsf{aux}')$: This algorithm takes as input the crs, an aux, a pk, and a function $f \in \mathcal{F}$. Output a master public key mpk, and a new state aux'.
- $\mathsf{Enc}(\mathsf{mpk}, m) \to \mathsf{ct}$: This algorithm takes as input the mpk, and a message $m \in \mathcal{M}$. Output a ciphertext ct.
- $\mathsf{Update}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}) \to \mathsf{hsk}$: This algorithm takes as input the crs, an aux, and a pk. Output a helper decryption key hsk.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{hsk}, \mathsf{ct}) \to f(m) \in \mathcal{Y} / \perp / \mathsf{GetUpdate}$: This algorithm takes as input a sk, a hsk, and a ct. Output a function value $f(m) \in \mathcal{Y}$, or $\perp$, or $\mathsf{GetUpdate}$.

Correctness, Compactness, and Update Efficiency. Let $\Pi_{RFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{RegPK}, \mathsf{Enc}, \mathsf{Update}, \mathsf{Dec})$ be a RFE scheme with message space $\mathcal{M}$ and function space $\mathcal{F}$. For a security parameter λ and an adversary $\mathcal{A}$, we define the following $\mathsf{Game}^{Corr}_{\Pi_{RFE},\mathcal{A}}(\lambda)$ between $\mathcal{A}$ and the challenger:

- Setup phase: The challenger runs $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^{|\mathcal{F}|})$, and initializes the state $\mathsf{aux} = \perp$ and the initial master public key $\mathsf{mpk}_0 = \perp$. Also, the challenger initializes three counters $\mathsf{ctr}_{reg} = 0, \mathsf{ctr}_{enc} = 0, \mathsf{ctr}^*_{reg} = \perp$ to keep track of the number of registration queries, the number of encryption queries, and the index of the target key, respectively. Also, it sets $\mathsf{out} = 0$ (this variable defines the output of the expriment). Finally, the challenger sends crs to $\mathcal{A}$.
- Query phase: The adversary $\mathcal{A}$ can make the following queries:
  - Register non-target key query: $\mathcal{A}$ sends a public key pk and a function $f \in \mathcal{F}$ to the challenger, then the challenger proceeds as follows:
    1. The challenger increments $\mathsf{ctr}_{reg} = \mathsf{ctr}_{reg} + 1$ and runs $(\mathsf{mpk}_{\mathsf{ctr}_{reg}}, \mathsf{aux}') \leftarrow \mathsf{RegPK}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}, f)$.
    2. The challenger updates $\mathsf{aux} = \mathsf{aux}'$ and sends $(\mathsf{ctr}_{reg}, \mathsf{mpk}_{\mathsf{ctr}_{reg}}, \mathsf{aux})$ to $\mathcal{A}$.
  - Register target key query: $\mathcal{A}$ sends a target function $f^* \in \mathcal{F}$ to the challenger. If $\mathsf{ctr}^*_{reg} \neq \perp$ (i.e., $\mathcal{A}$ has already make a register target key query), the challenger returns $\perp$. Otherwise, the challenger proceeds as follows:
    1. The challenger increments $\mathsf{ctr}_{reg} = \mathsf{ctr}_{reg} + 1$, then runs $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, \mathsf{aux})$ and $(\mathsf{mpk}_{\mathsf{ctr}_{reg}}, \mathsf{aux}')\mathsf{RegPK}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}^*, f^*)$.
    2. The challenger updates $\mathsf{aux} = \mathsf{aux}'$ and stores the index of the target identity $\mathsf{ctr}^*_{reg} \leftarrow \mathsf{ctr}_{reg}$. Then run $\mathsf{hsk}^* \leftarrow \mathsf{Update}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}^*)$.

3. The challenger sends $(\mathsf{ctr}_{reg}, \mathsf{mpk}_{\mathsf{ctr}_{reg}}, \mathsf{aux}, \mathsf{pk}^*, \mathsf{hsk}^*, \mathsf{sk}^*)$ to $\mathcal{A}$.

- Encryption query: $\mathcal{A}$ chooses an index $\mathsf{ctr}[reg]^* \leq i \leq \mathsf{ctr}_{reg}$ of a public key, and a message $m \in \mathcal{M}$. If $\mathsf{ctr}^*_{reg} = \perp$, the challenger returns $\perp$. Otherwise, the challenger sets $\mathsf{ctr}_{enc} = \mathsf{ctr}_{enc} + 1, m_{\mathsf{ctr}_{enc}} = m$, and runs $\mathsf{ct}_{\mathsf{ctr}_{enc}} \leftarrow \mathsf{Enc}(\mathsf{mpk}_i, m_{\mathsf{ctr}_{enc}})$. Finally, the challenger returns $(\mathsf{ctr}_{enc}, \mathsf{ct}_{\mathsf{ctr}_{enc}})$ to $\mathcal{A}$.
- Decryption query: $\mathcal{A}$ chooses an index $1 \leq j \leq \mathsf{ctr}_{enc}$. The challenger runs $y_j \leftarrow \mathsf{Dec}(\mathsf{sk}^*, \mathsf{hsk}^*, \mathsf{ct}_j)$. If $y_j = \mathsf{GetUpdate}$, the challenger updates the helper decryption key $\mathsf{hsk}^* \leftarrow \mathsf{Update}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}^*)$ and recomputes $y_j \leftarrow \mathsf{Dec}(\mathsf{sk}^*, \mathsf{hsk}^*, \mathsf{ct}_j)$. If $y_j \neq f^*(m_j)$, the experiment halts with output 1.

- End phase: When the adversary $\mathcal{A}$ has finished making queries and the experiment has not halted (as a result of a decryption query), then the experiment outputs 0.

Definition 1 (Correctness). *We say an RFE scheme $\mathit{\Pi}_{RFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{RegPK}, \mathsf{Enc}, \mathsf{Update}, \mathsf{Dec})$ with messge space $\mathcal{M}$ and function space $\mathcal{F}$ is correct (resp. perfectly correct) if for all (possibly unbounded) adversaries $\mathcal{A}$ making at most a polynomial number of queries, we have*

$$\Pr[\mathsf{Game}^{Corr}_{\mathit{\Pi}_{RFE},\mathcal{A}}(\lambda) = 1] = \mathsf{negl} \ \textit{(resp.}\ \Pr[\mathsf{Game}^{Corr}_{\mathit{\Pi}_{RFE},\mathcal{A}}(\lambda) = 1] = 0\textit{)}.$$

Definition 2 (Compactness). *Let N be the number of registration queries the adversary $\mathcal{A}$ makes in $\mathsf{Game}^{Corr}_{\mathit{\Pi}_{RFE},\mathcal{A}}(\lambda)$, and n be the length of the encrypted message. There exists a universal polynomial $\mathsf{poly}(\cdot,\cdot,\cdot)$ such that for all $i \in \mathsf{ctr}_{reg}$, $|\mathsf{mpk}_i| = \mathsf{poly}(\lambda, n, \log N)$. We also require that the size of the helper decryption key hsk^* satisfies $|\mathsf{hsk}^*| = \mathsf{poly}(\lambda, n, \log N)$ (at all points in $\mathsf{Game}^{Corr}_{\mathit{\Pi}_{RFE},\mathcal{A}}(\lambda)$).*

Definition 3 (Update Efficiency). *Let N be the number of registration queries the adversary $\mathcal{A}$ makes in $\mathsf{Game}^{Corr}_{\mathit{\Pi}_{RFE},\mathcal{A}}(\lambda)$. Then, in the course of $\mathsf{Game}^{Corr}_{\mathit{\Pi}_{RFE},\mathcal{A}}(\lambda)$, the challenger invokes the update algorithm Update at most $O(\log N)$ times, where each invocation runs in $\mathsf{poly}(\log N)$ time in the RAM model of computation. Specifically, we model Update as a RAM program that has random access to its input. Thus, the running time of Update in the RAM model can be smaller than the input length.*

Security Definition. The security definition of RFE is analogous to the standard FE security definition. Namely, each user of a slot should only gain the function value $f_i(m)$ and nothing else about the message $m \in \mathcal{M}$, given the ciphertext ct of m, the slot secret key sk_i, and the helper decryption key hsk_i, where $f_i \in \mathcal{F}$ is the registered function of slot i. As the standard FE, we define two kinds of security, one is weakly selective-indistinguishability security (weakly selective-IND security), and the other is selective-simulation security (selective-SIM security). We provide the formal definitions as below:

Weakly Selective-IND Security. We define the following $\mathsf{Game}^{Sel-IND}_{\mathit{\Pi}_{RFE},\mathcal{A}}(\lambda)$ between $\mathcal{A}$ and the challenger:

- $\mathsf{Setup\ phase}$: The adversary $\mathcal{A}$ chooses two messages (m_0^*, m_1^*) over $\mathcal{M}$, and sends them to the challenger. The challenger runs $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^{|\mathcal{F}|}, (m_0^*, m_1^*))$, and initializes the state $\mathsf{aux} = \perp$ and the master public key $\mathsf{mpk} = \perp$. Also, the challenger initializes a counter $\mathsf{ctr} = 0$ (for the number of honest registration queries made by $\mathcal{A}$), a set of corrupted public keys $\mathcal{C} = \emptyset$, and a dictionary $\mathsf{D} = \emptyset$ (storing the mapping between registered public keys and their corresponding functions). Finally, the challenger sends crs to $\mathcal{A}$.
- $\mathsf{Query\ phase}$: The adversary $\mathcal{A}$ can make the following queries:
  - $\mathsf{Register\ corrupted\ key\ query}$: $\mathcal{A}$ sends a public key pk and a function $f \in \mathcal{F}$ to the challenger. Then the challenger proceeds as follows:
    1. The challenger computes $(\mathsf{mpk}', \mathsf{aux}') \leftarrow \mathsf{RegPK}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}, f)$.
    2. The challenger updates $\mathsf{aux} = \mathsf{aux}', \mathsf{mpk} = \mathsf{mpk}', \mathcal{C} = \mathcal{C} \cup \{\mathsf{pk}\}$, and $\mathsf{D}[\mathsf{pk}] = \mathsf{D}[\mathsf{pk}] \cup \{f\}$.
    3. The challenger sends $(\mathsf{aux}, \mathsf{mpk})$ to $\mathcal{A}$.
  - $\mathsf{Register\ honest\ key\ query}$: $\mathcal{A}$ sends a target function $f \in \mathcal{F}$. Then the challenger proceeds as follows:
    1. The challenger sets $\mathsf{ctr} = \mathsf{ctr}{+}1$ and runs $(\mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}}) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, \mathsf{aux})$.
    2. The challenger registers $(\mathsf{pk}_{\mathsf{ctr}}, f)$ by running $(\mathsf{mpk}', \mathsf{aux}') \leftarrow \mathsf{RegPK}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}_{\mathsf{ctr}}, f)$.
    3. The challenger updates $\mathsf{aux} = \mathsf{aux}', \mathsf{mpk} = \mathsf{mpk}'$, and $\mathsf{D}[\mathsf{pk}_{\mathsf{ctr}}] = \mathsf{D}[\mathsf{pk}_{\mathsf{ctr}}] \cup \{f\}$.
    4. The challenger sends $(\mathsf{ctr}, \mathsf{aux}, \mathsf{mpk}, \mathsf{pk}_{\mathsf{ctr}})$ to $\mathcal{A}$.
  - $\mathsf{Corrupt\ honest\ key}$: $\mathcal{A}$ chooses an index $i \in [\mathsf{ctr}]$. The challenger updates $\mathcal{C} = \mathcal{C} \cup \{\mathsf{pk}_i\}$ and sends sk_i to $\mathcal{A}$, where $(\mathsf{pk}_i, \mathsf{sk}_i)$ is the i-th public and secret key generated during the i-th honest registration query.
- $\mathsf{Challenge\ phase}$: The challenger runs $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, m_b^*)$, where $b \leftarrow \{0, 1\}$, and sends ct^* to $\mathcal{A}$.
- $\mathsf{Output\ phase}$: $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

An adversary $\mathcal{A}$ is said valid if $f(m_0^*) = f(m_1^*)$ for each $f \in \mathcal{F}$. And as shown in [28,18], security without post-challenge queries (corrupt honest key queries) implies security with post-challenge queries. In other words, selective corruptions imply adaptive corruptions. Note that compared with the definition in [18], in which an adversary $\mathcal{A}$ is said valid if $f(m_0^*) = f(m_1^*)$ for every $f \in \{f \in \mathsf{D}[\mathsf{pk}] | \mathsf{pk} \in \mathcal{C}\}$, our definition is weaker. This is also why our definition is called weakly selective-IND security.

Definition 4 (Weakly Selective-IND Security). *We say an RFE scheme $\Pi_{RFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{RegPK}, \mathsf{Enc}, \mathsf{Update}, \mathsf{Dec})$ with messge space $\mathcal{M}$ and function space $\mathcal{F}$ is weakly selective-IND secure, if for all* p.p.t *valid adversaries $\mathcal{A}$, we have*

$$|\mathsf{Pr}[\mathsf{Game}_{\Pi_{RFE},\mathcal{A}}^{Sel-IND}(\lambda, 1) = 1] - \mathsf{Pr}[\mathsf{Game}_{\Pi_{RFE},\mathcal{A}}^{Sel-IND}(\lambda, 0) = 1]| = \mathsf{negl}.$$

Selective-SIM Security. We define the following real ensembles:

- $\mathsf{Setup}(1^\lambda, 1^L, 1^{|\mathcal{F}|}, m^*) \rightarrow \mathsf{crs}$: This algorithm takes as input the security parameter λ in unary notation, the number of slots L, the size of the function space $|\mathcal{F}|$, and a challenge message m^* from $\mathcal{A}$. Output a common reference string crs.
- $\mathsf{KeyGen}(\mathsf{crs}, \mathsf{aux}) \rightarrow (\mathsf{pk}, \mathsf{sk})$: This algorithm takes as input the crs, and a (possibly empty) state aux. Output a public key pk, and a secret key sk.
- $\mathsf{RegPK}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}, f) \rightarrow (\mathsf{mpk}, \mathsf{aux}')$: This algorithm takes as input the crs, an aux, a pk, and a function $f \in \mathcal{F}$ by $\mathcal{A}$. Output a master public key mpk, and a new state aux'.
- $\mathsf{Enc}(\mathsf{mpk}, m^*) \rightarrow \mathsf{ct}$: This algorithm takes as input the mpk, and a challenge message $m^* in \mathcal{M}$ from $\mathcal{A}$. Output a challenge ciphertext ct.

We define the following ideal ensembles:

- $\mathsf{Setup}^*(1^\lambda, 1^L, 1^{|\mathcal{F}|}, m^*) \rightarrow (\mathsf{crs}^*, \mathsf{msk}^*)$: This algorithm takes as input the security parameter λ in unary notation, the number of slots L, the size of the function space $|\mathcal{F}|$, and a challenge message m^* from $\mathcal{A}$. Output a simulated common reference string crs^*, and a master secret key msk^*.
- $\mathsf{KeyGen}^*(\mathsf{crs}^*, \mathsf{aux}^*) \rightarrow (\mathsf{pk}^*, \mathsf{sk}^*)$: This algorithm takes as input the crs^*, and a (possibly empty) state aux^*. Output a public key pk^*, and a secret key sk^*.
- $\mathsf{RegPK}^*(\mathsf{crs}^*, \mathsf{aux}^*, \mathsf{pk}^*, f) \rightarrow (\mathsf{mpk}^*, \mathsf{aux}'^*)$: This algorithm takes as input the crs^*, an aux^*, a pk^*, and a function $f \in \mathcal{F}$. Output a master public key mpk^*, and a new state aux'^*.
- $\mathsf{Enc}^*(\mathsf{msk}^*, \sigma) \rightarrow \mathsf{ct}^*$: This algorithm takes as input the msk^*, and auxiliary information σ. Output a challenge ciphertext ct^*.

Definition 5 (Selective-SIM Security). *We say an RFE scheme Π_{RFE} =* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{RegPK}, \mathsf{Enc}, \mathsf{Update}, \mathsf{Dec})$ *with messge space $\mathcal{M}$ and function space $\mathcal{F}$ is selective-SIM secure, if for all* p.p.t *adversaries $\mathcal{A}$, the above two ensembles in $\mathcal{A}$'s view, i.e.,* $(\mathsf{crs}, \mathsf{pk}, \mathsf{mpk}, \mathsf{aux}', \mathsf{ct})$ *and* $(\mathsf{crs}^*, \mathsf{pk}^*, \mathsf{mpk}^*, \mathsf{aux}'^*, \mathsf{ct}^*)$*, are computationally indistinguishable.*

*We further define **weakly seletive-SIM security**, which demonstrates the adversary obtain more function values than they should have obtained, but without revealing the message. This seems to be acceptable, since in centralized FE, under the precondition "not revealing the message", the adversary can query secret keys as many as possible.*

In our constructions, when the functions of the users that are corrupted by the adversary cover the entire function space, weakly selective-SIM security is equivalent to selective-SIM security.

3.6 Slotted Registered Functional Encryption

Syntax. Suppose a message space $\mathcal{M}$, and a function space $\mathcal{F} = \{f_u : \mathcal{M} \rightarrow \mathcal{Y}\}$. A slotted Registered Functional Encryption scheme consists of the following six algorithms, where $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}$ are randomized algorithms, and $\mathsf{IsValid}, \mathsf{Aggr}, \mathsf{Dec}$ are deterministic algorithms:

- $\mathsf{Setup}(1^\lambda, 1^L, 1^{|\mathcal{F}|}) \to \mathsf{crs}$: This algorithm takes as input the security parameter λ in unary notation, the number of slots L, and the size of the function space $|\mathcal{F}|$. Output a common reference string crs.
- $\mathsf{KeyGen}(\mathsf{crs}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$: This algorithm takes as input the crs, and a slot index $i \in [L]$. Output a public key pk_i, and a secret key sk_i.
- $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i) \to 0/1$: This algorithm takes as input the crs, a slot index $i \in [L]$, and a public key pk_i. Output a bit 0, or 1.
- $\mathsf{Aggr}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i \in \mathcal{F})\}_{i\in[L]}) \to (\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$: This algorithm takes as input the crs, and L pairs $(\mathsf{pk}_1, f_1 \in \mathcal{F}), ..., (\mathsf{pk}_L, f_L \in \mathcal{F})$. Output a master public key mpk, and L helper decryption keys $\{\mathsf{hsk}_i\}_{i\in[L]}$.
- $\mathsf{Enc}(\mathsf{mpk}, m) \to \mathsf{ct}$: This algorithm takes as input the mpk, and a message $m \in \mathcal{M}$. Output a ciphertext ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{hsk}, \mathsf{ct}) \to f(m) \in \mathcal{Y}/\perp$: This algorithm takes as input a sk, a hsk, and a ct. Output a function value $f(m) \in \mathcal{Y}$, or $\perp$.

Definition 6 (Completeness). *We say a slotted RFE scheme $\Pi_{sRFE} =$ (Setup, $\mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Aggr}, \mathsf{Enc}, \mathsf{Dec}$) with messge space $\mathcal{M}$ and function space $\mathcal{F}$ is completeness, if for any $\lambda \in \mathbb{N}$, any $L \in \mathbb{N}$, and any $i \in [L]$, we have*

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

Definition 7 (Perfect Correctness). *We say a slotted RFE scheme $\Pi_{sRFE} =$ ($\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Aggr}, \mathsf{Enc}, \mathsf{Dec}$) with messge space $\mathcal{M}$ and function space $\mathcal{F}$ is perfectly correct, if for any $\lambda \in \mathbb{N}$, any $L \in \mathbb{N}$, any $i \in [L]$, any $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^L, 1^{|\mathcal{F}|})$, any $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i)$, for all collection of public key $\{\mathsf{pk}_j\}_{j\in[L]\setminus\{i\}}$ such that $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, j, \mathsf{pk}_j)$, for any $m \in \mathcal{M}$, and any $f_1, ..., f_L \in \mathcal{F}$, we have*

$$\mathsf{Pr}\left[f_i(m) \leftarrow \mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \middle| \begin{array}{l} (\mathsf{mpk}, \{\mathsf{hsk}_t\}_{t\in[L]}) \leftarrow \mathsf{Aggr}(\mathsf{crs}, \{(\mathsf{pk}_t, f_t)\}_{t\in[L]}); \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, m); \end{array} \right] = 1.$$

Definition 8 (Compactness). *Let n be the length of the encrypted message. There exists a universal polynomial $\mathsf{poly}(\cdot, \cdot)$ such that the size of the master public key and individual helper secret key output by Aggr is $\mathsf{poly}(\lambda, n)$.*

Security Definition. The security definition of slotted RFE is analogous to the security definition of RFE. As RFE, we define weakly selective-indistinguishability security (weakly selective-IND security), and selective-simulation security (selective-SIM security), for slotted RFE. We provide the formal definitions as below:

Weakly Selective-IND Security. We define the following $\mathsf{Game}^{Sel-IND}_{\Pi_{sRFE},\mathcal{A}}(\lambda)$ between $\mathcal{A}$ and the challenger:

- $\mathsf{Setup\ phase}$: The adversary $\mathcal{A}$ chooses two messages (m_0^*, m_1^*) over $\mathcal{M}$, and sends (m_0^*, m_1^*) as well as the number of slots L to the challenger. The challenger runs $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^L, 1^{|\mathcal{F}|}, (m_0^*, m_1^*))$, and initializes a counter $\mathsf{ctr} = 0$, a dictionary $\mathsf{D} = \emptyset$, and a set of corrupted slot indexes $\mathcal{C} = \emptyset$. Finally, the challenger sends crs to $\mathcal{A}$.

- Query phase: The adversary $\mathcal{A}$ can make the following queries:
  - Honest key generation query: $\mathcal{A}$ sends $i \in [L]$ to the challenger. The challenger sets $\mathsf{ctr} = \mathsf{ctr} + 1$, runs $(\mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}}) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i)$, and sets $\mathsf{D}[\mathsf{ctr}] = (i, \mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}})$. Finally, the challenger sends $(\mathsf{ctr}, \mathsf{pk}_{\mathsf{ctr}})$ to $\mathcal{A}$.
  - Corruption query: $\mathcal{A}$ sends $j \in [\mathsf{ctr}]$ to the challenger. The challenger sends sk', where $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[j]$, to $\mathcal{A}$. Let $\mathcal{Q}_{Corr}$ be the set of corruption queries made by $\mathcal{A}$.
- Challenge phase: $\mathcal{A}$ sends the challenge $(\{c_i^*, f_i^*, \mathsf{pk}_i^*\}_{i\in[L]})$, where $c_i^* \in [\mathsf{ctr}] \cup \{\perp\}$. Then, for each $i \in [L]$, the challenger proceeds as follows:
  - If $c_i^* \in [\mathsf{ctr}]$, the challenger retrieves $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[c_i^*]$. If $i' = i$, the challenger sets $\mathsf{pk}_i = \mathsf{pk}'$. In addition, if $c_i^* \in \mathcal{Q}_{Corr}$, the challenger updates $\mathcal{C} = \mathcal{C} \cup \{i\}$. Otherwise, if $i' \neq i$, the challenger aborts.
  - If $c_i^* = \perp$, the challenger checks the validity of pk_i^*. If $0 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$, the challenger aborts. Otherwise, the challenger sets $\mathsf{pk}_i = \mathsf{pk}_i^*$ and updates $\mathcal{C} = \mathcal{C} \cup \{i\}$.

  Finally, the challenger sends $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, m_b^*)$, where $b \leftarrow \{0,1\}$ and $\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]} \leftarrow \mathsf{Aggr}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i^*)\})$, to $\mathcal{A}$.
- Output phase: $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

An adversary $\mathcal{A}$ is said valid if $f(m_0^*) = f(m_1^*)$ for each $f \in \mathcal{F}$. And as shown in [28,18], security without post-challenge queries (corruption queries) implies security with post-challenge queries. Note that compared with the definition in [18], in which an adversary $\mathcal{A}$ is said valid if $f_i^*(m_0^*) = f_i^*(m_1^*)$ for every $i \in \mathcal{C}$, our definition is weaker. This is also why our definition is called weakly selective-IND security.

Definition 9 (Weakly Selective-IND Security). *We say a slotted RFE scheme* $\Pi_{sRFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Aggr}, \mathsf{Enc}, \mathsf{Dec})$ *with messge space* $\mathcal{M}$ *and function space* $\mathcal{F}$ *is weakly selective-IND secure, if for all* p.p.t *valid adversaries* $\mathcal{A}$*, we have*

$$|\Pr[\mathsf{Game}^{Sel-IND}_{\Pi_{sRFE},\mathcal{A}}(\lambda, 1) = 1] - \Pr[\mathsf{Game}^{Sel-IND}_{\Pi_{sRFE},\mathcal{A}}(\lambda, 0) = 1]| = \mathsf{negl}.$$

Selective-SIM Security. We define the following real ensembles:

- $\mathsf{Setup}(1^\lambda, 1^L, 1^{|\mathcal{F}|}, m^*) \rightarrow \mathsf{crs}$: This algorithm takes as input the security parameter λ in unary notation, the number of slots L, the size of the function space $|\mathcal{F}|$, and a challenge message m^* from $\mathcal{A}$. Output a common reference string crs.
- $\mathsf{KeyGen}(\mathsf{crs}, i) \rightarrow (\mathsf{pk}_i, \mathsf{sk}_i)$: This algorithm takes as input the crs, and a slot index $i \in [L]$. Output a public key pk_i, and a secret key sk_i.
- $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i) \rightarrow 0/1$: This algorithm takes as input the crs, a slot index $i \in [L]$, and a public key pk_i. Output a bit 0, or 1.
- $\mathsf{Aggr}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i \in \mathcal{F})\}_{i\in[L]}) \rightarrow (\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$: This algorithm takes as input the crs, and L pairs $(\mathsf{pk}_1, f_1 \in \mathcal{F}), ..., (\mathsf{pk}_L, f_L \in \mathcal{F})$. Output a master public key mpk, and L helper decryption keys $\{\mathsf{hsk}_i\}_{i\in[L]}$.
- $\mathsf{Enc}(\mathsf{mpk}, m^*) \rightarrow \mathsf{ct}$: This algorithm takes as input the mpk, and a challenge message $m \in \mathcal{M}$. Output a ciphertext ct.

We define the following ideal ensembles:

- $\mathsf{Setup}^*(1^\lambda, 1^L, 1^{|\mathcal{F}|}, m^*) \to (\mathsf{crs}^*, \mathsf{msk}^*)$: This algorithm takes as input the security parameter λ in unary notation, the number of slots L, the size of the function space $|\mathcal{F}|$, and a challenge message m^* from $\mathcal{A}$. Output a simulated common reference string crs^*, and a master secret key msk^*.
- $\mathsf{KeyGen}^*(\mathsf{crs}^*, i) \to (\mathsf{pk}_i^*, \mathsf{sk}_i^*)$: This algorithm takes as input the crs^*, and a slot index $i \in [L]$. Output a public key pk_i^*, and a secret key sk_i^*.
- $\mathsf{IsValid}^*(\mathsf{crs}^*, i, \mathsf{pk}_i^*) \to 0/1$: This algorithm takes as input the crs^*, a slot index $i \in [L]$, and a public key pk_i^*. Output a bit 0, or 1.
- $\mathsf{Aggr}^*(\mathsf{crs}^*, \{(\mathsf{pk}_i^*, f_i \in \mathcal{F})\}_{i\in[L]}) \to (\mathsf{mpk}^*, \{\mathsf{hsk}_i^*\}_{i\in[L]})$: This algorithm takes as input the crs^*, and L pairs $(\mathsf{pk}_1^*, f_1), ..., (\mathsf{pk}_L^*, f_L)$. Output a master public key mpk^*, and L helper decryption keys $\{\mathsf{hsk}_i^*\}_{i\in[L]}$.
- $\mathsf{Enc}^*(\mathsf{msk}^*, \sigma) \to \mathsf{ct}^*$: This algorithm takes as input the msk^*, and auxiliary information σ. Output a ciphertext ct^*.

Definition 10 (Selective-SIM Security). *We say a slotted RFE scheme $\Pi_{sRFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Aggr}, \mathsf{Enc}, \mathsf{Dec})$ with messge space $\mathcal{M}$ and function space $\mathcal{F}$ is selecive-SIM secure, if for all* p.p.t *adversaries $\mathcal{A}$, the above two ensembles in $\mathcal{A}$'s view, i.e., $(\mathsf{crs}, \{\mathsf{pk}_i\}_i, \mathsf{mpk}, \{\mathsf{hsk}_i\}_i, \mathsf{ct})$ and $(\mathsf{crs}^*, \{\mathsf{pk}_i^*\}_i, \mathsf{mpk}^*, \{\mathsf{hsk}_i^*\}_i, \mathsf{ct}^*)$, are computationally indistinguishable.*

*We further define **weakly seletive-SIM security**, which demonstrates the adversary obtain more function values than they should have obtained, but without revealing the message. This is seems to be acceptable, since in centralized FE, under the precondition "not revealing the message", the adversary can query secret keys as many as possible.*

In our constructions, when the functions of the users that are corrupted by the adversary cover the entire function space, weakly selective-SIM security is equivalent to selective-SIM security.

4 Slotted Registered Inner-Product Functional Encryption

4.1 Construction Π_{sRIPFE}

We construct a slotted Registered Inner-Product Functional Encryption (slotted RIPFE) scheme $\Pi_{sRIPFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Aggr}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M} = \mathbb{Z}_p^{1\times n}$, function space $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n\times 1}\}_{j<n}$, where $n \in \mathbb{N}$, a function value upperbound B, and an a-priori fixed number of slots $L = L(\lambda)$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|})$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. Output
$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{AV}]_1, [\mathbf{AV}_1]_1, [\mathbf{AV}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + \mathbf{V}\mathbf{f}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}).\end{aligned}$$

- $\mathsf{KeyGen}(\mathsf{crs}, i)$: Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, which satisfies each component of $\mathbf{U}_i$ is bounded by the brute-force discrete log bound B (used in the security proofs). Store $\mathsf{sk}_i = \mathbf{U}_i$. Output
$$\mathsf{pk}_i = ([P_i^1]_1 = [\mathbf{A}\mathbf{U}_i]_1, \{[P_{i,t,j}^2]_2\} = \{[\mathbf{U}_i\mathbf{W}_t\mathbf{f}_j]_2\}_{t\neq i, j\in[|\mathcal{F}|]}).$$
- $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$: Check
  1. whether $[P_i^1]_1$ in pk_i are elements in G_1?
  2. whether $\{[P_{i,t,j}^2]_2\}$ in pk_i are elements in G_2?
  3. whether $e([P_i^1]_1, [\mathbf{W}_t\mathbf{f}_j]_2) = e([\mathbf{A}]_1, [P_{i,t,j}^2]_2)$, for each $t \neq i, j \in [|\mathcal{F}|]$?

  If any fails, output 0. Otherwise, output 1.
- $\mathsf{Aggr}(\mathsf{crs}, (\mathsf{pk}_1, \mathbf{f}^1), ..., (\mathsf{pk}_L, \mathbf{f}^L))$: For the target functions $\{\mathbf{f}^i\}_{i\in[L]}$, set each $\mathbf{f}^i = \alpha_i \mathbf{f}^{'i}$, where the first component of $\mathbf{f}^{'i}$ equals 1. Output
$$\mathsf{mpk} = ([\mathbf{A}\sum_{i\in[L]}\mathbf{U}_i]_1, [\sum_{i\in[L]}\mathbf{W}_i\mathbf{f}^{'i}]_2, [\mathbf{A}]_1, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1);$$
$$\{\mathsf{hsk}_i = (i, \mathbf{f}^i, [\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{f}^{'i}]_2, [\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{f}^{'t}]_2), [\mathbf{V}_1\mathbf{W}_i\mathbf{f}^{'i} + \mathbf{V}_2\mathbf{W}_i\mathbf{f}^{'i} + \mathbf{V}\mathbf{f}^{'i}]_2,$$
$$[\mathbf{W}_i\mathbf{f}^{'i}]_2\}_{i\in[L]}.$$
- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x} \in \mathcal{M})$: Sample $\mathbf{s} \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta \leftarrow_R \mathbb{Z}_p$. Output
$$\mathsf{ct} = (C_1 = [\mathbf{s}\mathbf{A}]_1, C_2 = [\mathbf{s}\mathbf{A}\mathbf{V} + \mathbf{x} + \eta \cdot (\underbrace{1, 0, ..., 0}_{n's})]_1,$$
$$C_3 = [\mathbf{s}\mathbf{A}\mathbf{V}_1 + \mathbf{s}\mathbf{A}\sum_{t\in[L]}\mathbf{U}_t]_1, C_4 = [\mathbf{s}\mathbf{A}\mathbf{V}_2]_1,$$
$$C_5 = [\mathbf{s}\mathbf{A}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{f}^{'t} - \eta]_T).$$
- $\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct})$: For the target $\mathbf{f}^i$, set $\mathbf{f}^i = \alpha_i\mathbf{f}^{'i}$, where the first component of $\mathbf{f}^{'i}$ equals 1. Since α_i can be gained from $\mathbf{f}^i$, $[\mathbf{W}_i\mathbf{f}^{'i}]_2$ can be gained by computing $[\alpha_i^{-1}\mathbf{W}_i\mathbf{f}^i]_2$. Similar for $[\mathbf{V}_1\mathbf{W}_i\mathbf{f}^{'i} + \mathbf{V}_2\mathbf{W}_i\mathbf{f}^{'i} + \mathbf{V}\mathbf{f}^{'i}]_2$. Then compute $[\alpha_i^{-1}\gamma]_T$ as follows:
$$e(C_2, \underbrace{[\mathbf{f}^{'i}]_2}_{\mathsf{hsk}_i}) \cdot e(C_1, \underbrace{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}^{'i} + \mathbf{V}_2\mathbf{W}_i\mathbf{f}^{'i} + \mathbf{V}\mathbf{f}^{'i}]_2}_{\mathsf{hsk}_i})^{-1} \cdot e(C_3, \underbrace{[\mathbf{W}_i\mathbf{f}^{'i}]_2}_{\mathsf{hsk}_i}) \cdot$$
$$e(C_1, \underbrace{[\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{f}^{'i}]_2}_{\mathsf{hsk}_i})^{-1} \cdot e(C_1, [\underbrace{\mathbf{U}_i}_{\mathsf{sk}_i}\underbrace{\mathbf{W}_i\mathbf{f}^{'i}]_2}_{\mathsf{hsk}_i})^{-1} \cdot e(C_4, \underbrace{[\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{f}^{'t}]_2}_{\mathsf{hsk}_i})^{-1} \cdot C_5.$$

  Output γ by brute-force discrete log.

Theorem 1 (Completeness of Construction Π_{sRIPFE}). *The slotted RIPFE construction Π_{sRIPFE} with message space $\mathcal{M} = \mathbb{Z}_p^{1\times n}$, function space $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n\times 1}\}_{j<n}$, a function value upperbound B, and an a-priori fixed number of slots $L = L(\lambda)$ is complete.*

Proof. We proceed the checks as follows:

1. It is easy to see that $[P_i^1]_1$ in pk_i are elements in G_1.
2. It is easy to see that $\{[P^2_{i,t,j}]_2\}$ in pk_i are elements in G_2.
3. For each $t \neq i, j \in [|\mathcal{F}|]$, we have $e([P_i^1]_1, [\mathbf{W}_t\mathbf{f}_j]_2) = [\mathbf{A}\mathbf{U}_i\mathbf{W}_t\mathbf{f}_j]_T = e([\mathbf{A}]_1, [P^2_{i,t,j}]_2)$.

All the checks pass, thus, $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$ outputs 1 and completeness holds.

Theorem 2 (Compactness of Construction Π_{sRIPFE}). *The slotted RIPFE construction Π_{sRIPFE} with message space $\mathcal{M} = \mathbb{Z}_p^{1\times n}$, function space $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n\times 1}\}_{j<n}$, a function value upperbound B, and an a-priori fixed number of slots $L = L(\lambda)$ satisfies the following properties:*

- $|\mathsf{mpk}| = n \cdot \mathsf{poly}(\lambda)$.
- *For each* $i \in [L]$, $|\mathsf{hsk}_i| = n \cdot \mathsf{poly}(\lambda) + O(\log L)$.

Proof. We can see in mpk, $[\mathbf{A}\sum_{i\in[L]}\mathbf{U}_i]_1$ is of kn group elements, $[\sum_{i\in[L]}\mathbf{W}_i\mathbf{f}^i]_2$ is of n group elements, $[\mathbf{A}]_1$ is of $k(k+1)$ group elements, $[\mathbf{A}\mathbf{V}]_1$ is of kn group elements, $[\mathbf{A}\mathbf{V}_1]_1$ is of kn group elements, and $[\mathbf{A}\mathbf{V}_2]_1$ is of kn group elements. Each group element is of $\mathsf{poly}(\lambda)$ size. Thus, when we omit k, which is the parameter of the MDDH assumption, the size of mpk is $n \cdot \mathsf{poly}(\lambda)$.

Similarly, in hsk_i, i is of $O(\log L)$ size, $\mathbf{f}^i$ is of n ring elements, $[\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{f}^i]_2$ is of $(k+1)$ group elements, $[\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{f}^t]_2$ is of n group elements, $[\mathbf{V}_1\mathbf{W}_i\mathbf{f}^{'i} + \mathbf{V}_2\mathbf{W}_i\mathbf{f}^{'i} + \mathbf{V}\mathbf{f}^{'i}]_2$ is of $(k+1)$ group elements, and $[\mathbf{W}_i\mathbf{f}^{'i}]_2$ is of n group elements. Each ring element or group element is of $\mathsf{poly}(\lambda)$ size. Thus, when we omit k, which is the parameter of the MDDH assumption, the size of hsk_i is $n \cdot \mathsf{poly}(\lambda) + O(\log L)$.

Theorem 3 (Correctness of Construction Π_{sRIPFE}). *The slotted RIPFE construction Π_{sRIPFE} with message space $\mathcal{M} = \mathbb{Z}_p^{1\times n}$, function space $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n\times 1}\}_{j<n}$, a function value upperbound B, and an a-priori fixed number of slots $L = L(\lambda)$ is perfectly correct.*

Proof. We proceed the proof in two steps.

1. **We first demonstrate the correctness in the case that each $\mathsf{pk}_i, i \in [L]$ is of the right form.**
   Fix some λ, message size $n = n(\lambda)$, the number of slots $L = L(\lambda)$, the size of function space $|\mathcal{F}|$, and an index $i \in [L]$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|})$ and $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i)$ be defined as in Construction Π_{sRIPFE}. Take any set of public keys $\{\mathsf{pk}_j\}_{j\in[L]\setminus\{i\}}$, where $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, j, \mathsf{pk}_j)$. For each $j \in [L]$, let $\mathbf{f}^j$ be the target function associated with pk_j. Let $(\mathsf{mpk}, \{\mathsf{hsk}_i\}) \leftarrow \mathsf{Aggr}(\mathsf{crs}, \{(\mathsf{pk}_j, \mathbf{f}^j)\}_{j\in[L]})$. For a message $\mathbf{x} \in \mathcal{M}$, let $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathbf{x})$. Then

the correctness follows:

$$\begin{aligned}
&e(C_2,[\mathbf{f}^{'i}]_2)\cdot e(C_1,[\mathbf{V}_1\mathbf{W}_i\mathbf{f}^{'i}+\mathbf{V}_2\mathbf{W}_i\mathbf{f}^{'i}+\mathbf{V}\mathbf{f}^{'i}]_2)^{-1}\cdot e(C_3,[\mathbf{W}_i\mathbf{f}^{'i}]_2)\cdot\\
&e(C_1,[\sum_{t\neq i,t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{f}^{'i}]_2)^{-1}\cdot e(C_1,[\mathbf{U}_i\mathbf{W}_i\mathbf{f}^{'i}]_2)^{-1}\cdot e(C_4,[\sum_{t\neq i,t\in[L]}\mathbf{W}_t\mathbf{f}^{'t}]_2)^{-1}\cdot C_5\\
=&[\mathbf{sAV}\mathbf{f}^{'i}+\mathbf{x}\mathbf{f}^{'i}+\eta\cdot(1,0,...,0)\mathbf{f}^{'i}-\mathbf{sAV}_1\mathbf{W}_i\mathbf{f}^{'i}-\mathbf{sAV}_2\mathbf{W}_i\mathbf{f}^{'i}-\mathbf{sAV}\mathbf{f}^{'i}+\\
&\mathbf{sAV}_1\mathbf{W}_i\mathbf{f}^{'i}+\mathbf{sA}\sum_{t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{f}^{'i}-\mathbf{sA}\sum_{t\neq i,t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{f}^{'i}-\mathbf{sAU}_i\mathbf{W}_i\mathbf{f}^{'i}-\\
&\mathbf{sAV}_2\sum_{t\neq i,t\in[L]}\mathbf{W}_t\mathbf{f}^{'t}+\mathbf{sAV}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{f}^{'t}-\eta]_T\\
=&[\mathbf{x}\mathbf{f}^{'i}]_T.
\end{aligned}$$

Then by multiplying α_i in the exponent, which can be obtained from $\mathbf{f}^i$, we can finally obtain $[\mathbf{x}\mathbf{f}^i]_T$.

2. **We then demonstrate that correctness holds as long as $1\leftarrow\mathsf{IsValid}(\mathsf{crs},j,\mathsf{pk}_j)$.** We replace $[\mathbf{AU}_i]_1$ and $[\mathbf{U}_i\mathbf{W}_t\mathbf{f}_j]_2$ with $[P_i^1]_1$ and $[P_{i,t,j}^2]_2$, respectively. Then, we have

$$\begin{aligned}
&e(C_2,[\mathbf{f}^{'i}]_2)\cdot e(C_1,[\mathbf{V}_1\mathbf{W}_i\mathbf{f}^{'i}+\mathbf{V}_2\mathbf{W}_i\mathbf{f}^{'i}+\mathbf{V}\mathbf{f}^{'i}]_2)^{-1}\cdot e(C_3,[\mathbf{W}_i\mathbf{f}^{'i}]_2)\cdot\\
&e(C_1,[\sum_{t\neq i,t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{f}^{'i}]_2)^{-1}\cdot e(C_1,[\mathbf{U}_i\mathbf{W}_i\mathbf{f}^{'i}]_2)^{-1}\cdot e(C_4,[\sum_{t\neq i,t\in[L]}\mathbf{W}_t\mathbf{f}^{'t}]_2)^{-1}\cdot C_5\\
=&[\mathbf{sAV}\mathbf{f}^{'i}+\mathbf{x}\mathbf{f}^{'i}+\eta\cdot(1,0,...,0)\mathbf{f}^{'i}-\mathbf{sAV}_1\mathbf{W}_i\mathbf{f}^{'i}-\mathbf{sAV}_2\mathbf{W}_i\mathbf{f}^{'i}-\mathbf{sAV}\mathbf{f}^{'i}+\\
&\mathbf{sAV}_1\mathbf{W}_i\mathbf{f}^{'i}+\mathbf{s}\sum_{t\in[L]}P_t^1\mathbf{W}_i\mathbf{f}^{'i}-\mathbf{sA}\sum_{t\neq i,t\in[L]}P_{t,i,i}^{'2}-\mathbf{sAU}_i\mathbf{W}_i\mathbf{f}^{'i}-\\
&\mathbf{sAV}_2\sum_{t\neq i,t\in[L]}\mathbf{W}_t\mathbf{f}^{'t}+\mathbf{sAV}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{f}^{'t}-\eta]_T, \qquad (1)
\end{aligned}$$

where $P_{t,i,i}^{'2}=\alpha_i^{-1}\cdot(P_{t,i,i}^2)$.
By $1\leftarrow\mathsf{IsValid}(\mathsf{crs},j,\mathsf{pk}_j)$, we have

$$[P_t^1\mathbf{W}_i\mathbf{f}^{'i}]_T=[\mathbf{A}P_{t,i,i}^{'2}]_T,$$

for each $t\neq i$. Thus we have

$$[\mathbf{s}\sum_{t\neq i,t\in[L]}P_t^1\mathbf{W}_i\mathbf{f}^{'i}]_T=[\mathbf{sA}\sum_{t\neq i,t\in[L]}P_{t,i,i}^{'2}]_T.$$

Since the rest $[\mathbf{s}P_i^1\mathbf{W}_i\mathbf{f}^{'i}]_T\overset{?}{=}[\mathbf{sAW}_i\mathbf{f}^{'i}]_T$ depends on whether the adversary desires to proceed a successful decryption, we don't need to consider it. Thus the formula (1) equals $[\mathbf{x}\mathbf{f}^{'i}]_T$.

Theorem 4 (Weakly Selective-IND Security of Construction Π_{sRIPFE}). *The slotted RIPFE construction Π_{sRIPFE} with message space $\mathcal{M}=\mathbb{Z}_p^{1\times n}$, function space $\mathcal{F}=\{\mathbf{f}_j\in\mathbb{Z}_p^{n\times 1}\}_{j<n}$, a function value upperbound B, and an a-priori*

fixed number of slots $L = L(\lambda)$ is weakly selective-IND secure relying on the MDDH assumption.

Proof.
Games. We define the following games:

- Game_0: This is the game with the real construction and choosing $\mathbf{x}_1^*$ as the encrypted message.
- Game_1: This is the same as Game_0, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_0^*, \mathbf{x}_1^*)$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|})$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Output
$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + \mathbf{V}\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]});\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$
  - $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_1^*, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, and $\eta \leftarrow_R \mathbb{Z}_p$. Output
$$\begin{aligned}\mathsf{ct} = (&C_1 = [\mathbf{c}]_1, C_2 = [\mathbf{c}\mathbf{V} + \mathbf{x}_1^* + \eta \cdot \underbrace{(1, 0, ..., 0)}_{n's}]_1,\\ &C_3 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]} \mathbf{U}_t]_1, C_4 = [\mathbf{c}\mathbf{V}_2]_1,\\ &C_5 = [\mathbf{c}\mathbf{V}_2 \sum_{t\in[L]} \mathbf{W}_t\mathbf{f}'^{t} - \eta]_T).\end{aligned}$$
- Game_2: This is the same as Game_1, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_0^*, \mathbf{x}_1^*)$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_0^*, \mathbf{x}_1^*))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Output
$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + (\widetilde{\mathbf{V}} + \mathbf{a}^{\perp\top}(\mathbf{x}_0^* - \mathbf{x}_1^*))\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]});\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_0^*, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, and $\eta \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\mathbf{c}]_1, C_2 &= [\mathbf{c}\widetilde{\mathbf{V}} + \mathbf{x}_0^* + \eta\cdot(\underbrace{1, 0, ..., 0}_{n's})]_1,\\ C_3 &= [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1, C_4 = [\mathbf{c}\mathbf{V}_2]_1,\\ C_5 &= [\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{f}'^{t} - \eta]_T).\end{aligned}$$

– Game_3: This is the same as Game_2, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_0^*, \mathbf{x}_1^*)$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|})$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. Output

$$\begin{aligned}\mathsf{crs} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ \{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + \widetilde{\mathbf{V}}\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]});\\ \mathsf{msk} = (\mathbf{a}^\perp, \mathbf{A}, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

  - $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_0^*)$: Sample $\mathbf{s} \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\mathbf{s}\mathbf{A}]_1, C_2 &= [\mathbf{s}\mathbf{A}\widetilde{\mathbf{V}} + \mathbf{x}_0^* + \eta\cdot(\underbrace{1, 0, ..., 0}_{n's})]_1,\\ C_3 &= [\mathbf{s}\mathbf{A}\mathbf{V}_1 + \mathbf{s}\mathbf{A}\sum_{t\in[L]}\mathbf{U}_t]_1, C_4 = [\mathbf{s}\mathbf{A}\mathbf{V}_2]_1,\\ C_5 &= [\mathbf{s}\mathbf{A}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{f}'^{t} - \eta]_T).\end{aligned}$$

Lemma 1. *We have* $|\mathsf{Adv}_{\mathsf{Game}_0,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_1,\mathcal{A}}(\lambda)| \le \mathsf{Adv}_{\mathcal{B}_1,G_1}^{MDDH}(\lambda)$*, where* $\mathcal{B}_1$ *is the adversary for the MDDH assumption in* G_1.

Proof. Suppose a challenger $\mathcal{B}_1$. The adversary $\mathcal{A}$ sends $(\mathbf{x}_0^*, \mathbf{x}_1^*)$ and the number of slots L to $\mathcal{B}_1$. $\mathcal{B}_1$ runs the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, and samples $\mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. $\mathcal{B}_1$ receives $([\mathbf{A}]_1, [T]_1)$ from the underlying MDDH assumption. For $\mathbf{A}$, there exist $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Then $\mathcal{B}_1$ generates

$$\begin{aligned}\mathsf{crs} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ \{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + \mathbf{V}\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]});\\ \mathsf{msk} = (\mathbf{V}, \mathbf{V}_1, \mathbf{V}_2, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

$\mathcal{B}_1$ initializes a counter $\mathsf{ctr} = 0$, a dictionary $\mathsf{D} = \emptyset$, and a set of corrupted slot indexes $\mathcal{C} = \emptyset$. $\mathcal{B}_1$ sends crs to $\mathcal{A}$.

Upon receiving an honest key generation query $i \in [L]$ from $\mathcal{A}$, $\mathcal{B}_1$ sets $\mathsf{ctr} = \mathsf{ctr} + 1$, and generates $(\mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}})$ as follows:

Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, which satisfies each component of $\mathbf{U}_i$ is bounded by B. Set

$$\mathsf{pk}_{\mathsf{ctr}} = ([\mathbf{A}\mathbf{U}_i]_1, \{[\mathbf{U}_i \mathbf{W}_t \mathbf{f}_j]_2\}_{t \neq i, j \in [|\mathcal{F}|]});$$
$$\mathsf{sk}_{\mathsf{ctr}} = \mathbf{U}_i.$$

Upon receiving a corruption query $j \in [\mathsf{ctr}]$ from $\mathcal{A}$, $\mathcal{B}_1$ sends sk' to $\mathcal{A}$, where $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[j]$. Let $\mathcal{Q}_{Corr}$ be the set of corruption queries made by $\mathcal{A}$.

Upon receiving the challenge $(\{c_i^*, \mathbf{f}^{i*}, \mathsf{pk}_i^*\}_{i\in[L]})$ from $\mathcal{A}$, where $c_i^* \in [\mathsf{ctr}] \cup \{\perp\}$. Then for each $i \in [L]$, $\mathcal{B}_1$ proceeds as follows:

- If $c_i^* \in [\mathsf{ctr}]$, $\mathcal{B}_1$ retrieves $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[c_i^*]$. If $i' = i$, $\mathcal{B}_1$ sets $\mathsf{pk}_i = \mathsf{pk}'$. In addition, if $c_i^* \in \mathcal{Q}$, $\mathcal{B}_1$ updates $\mathcal{C} = \mathcal{C} \cup \{i\}$. Otherwise, if $i' \neq i$, $\mathcal{B}_1$ aborts.
- If $c_i^* = \perp$, $\mathcal{B}_1$ checks the validity of pk_i^*. If $0 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_1$ aborts. Otherwise, $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_1$ computes $\mathsf{sk}_i = \mathbf{U}_i$ with msk and brute-force discrete log (If $\mathbf{U}_i$ cannot be computed, $\mathcal{B}_1$ aborts. And note that only with negligible probability, $[\mathbf{U}_i]_T$ is not unique, in this case, $\mathcal{B}_1$ also aborts.). Then $\mathcal{B}_1$ sets $\mathsf{pk}_i = \mathsf{pk}_i^*$ and updates $\mathcal{C} = \mathcal{C} \cup \{i\}$. $\mathcal{B}_1$ sets $\mathbf{f}^{i*} = \alpha_i \mathbf{f}^{'i*}$ for each $i \in [L]$.

$\mathcal{B}_1$ samples $\eta \leftarrow_R \mathbb{Z}_p$ and generates ct^* as follows:

$$C_1 = [\boxed{T}]_1, C_2 = [\boxed{T}\mathbf{V} + \mathbf{x}_1^* + \eta \cdot (\underbrace{1, 0, ..., 0}_{n's})]_1,$$
$$C_3 = [\boxed{T}\mathbf{V}_1 + \boxed{T}\sum_{t\in[L]} \mathbf{U}_t]_1, C_4 = [\boxed{T}\mathbf{V}_2]_1,$$
$$C_5 = [\boxed{T}\mathbf{V}_2 \sum_{t\in[L]} \mathbf{W}_t \mathbf{f}^{'t*} - \eta]_T).$$

$\mathcal{B}_1$ generates $\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]}$ as follows:

$$\mathsf{mpk} = ([\mathbf{A}\sum_{i\in[L]} \mathbf{U}_i]_1, [\sum_{i\in[L]} \mathbf{W}_i \mathbf{f}^{'i*}]_2, [\mathbf{A}]_1, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1);$$
$$\{\mathsf{hsk}_i = (i, \mathbf{f}^{i*}, [\sum_{t\neq i, t\in[L]} \mathbf{U}_t \mathbf{W}_i \mathbf{f}^{'i*}]_2, [\sum_{t\neq i, t\in[L]} \mathbf{W}_t \mathbf{f}^{'t*}]_2), [\mathbf{V}_1 \mathbf{W}_i \mathbf{f}^{'i*} + \mathbf{V}_2 \mathbf{W}_i \mathbf{f}^{'i*} + \mathbf{V}\mathbf{f}^{'i*}]_2,$$
$$[\mathbf{W}_i \mathbf{f}^{'i*}]_2\}_{i\in[L]}.$$

$\mathcal{B}_1$ sends $(\mathsf{ct}^*, \mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$ to $\mathcal{A}$.

Observe when $T = \mathbf{s}\mathbf{A}$, where $\mathbf{s} \leftarrow_R \mathbb{Z}_p^{1\times k}$, the distributions are as Game_0; when $T = \mathbf{c}$, where $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$ satisfying $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, the distributions are as Game_1. Then if $\mathcal{A}$ can distinguish Game_0 and Game_1, $\mathcal{B}_1$ can utilize $\mathcal{A}$ to break the MDDH assumption in G_1. Thus lead to contradiction.

Lemma 2. *We have* $|\mathsf{Adv}_{\mathsf{Game}_1,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_2,\mathcal{A}}(\lambda)| = 0$.

Proof. Sample $\widetilde{\mathbf{V}} \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$. Then change the variables by embedding the challenge $(\mathbf{x}_0^*, \mathbf{x}_1^*)$ into $\mathbf{V}$ as follows:

$$\mathbf{V} = \widetilde{\mathbf{V}} + \mathbf{a}^{\perp\top}(\mathbf{x}_0^* - \mathbf{x}_1^*),$$

where the distributions are taken over the random choices of $\mathbf{V}$ and $\widetilde{\mathbf{V}}$.

We have $\mathbf{AV} = \mathbf{A}(\widetilde{\mathbf{V}} + \mathbf{a}^{\perp\top}(\mathbf{x}_0^* - \mathbf{x}_1^*)) = \mathbf{A}\widetilde{\mathbf{V}}$. Then crs is changed into

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\boxed{\mathbf{A}\widetilde{\mathbf{V}}}]_1, [\mathbf{AV}_1]_1, [\mathbf{AV}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + \boxed{(\widetilde{\mathbf{V}} + \mathbf{a}^{\perp\top}(\mathbf{x}_0^* - \mathbf{x}_1^*))}\,\mathbf{f}_j]_2\}_{i\in[L],j\in|[\mathcal{F}]|}).\end{aligned}$$

ct^* is changed into

$$\begin{aligned}\mathsf{ct}^* = (C_1 = [\mathbf{c}]_1, C_2 = [&\underbrace{\boxed{\mathbf{c}\widetilde{\mathbf{V}} + \mathbf{x}_0^*}}_{\mathbf{c}(\widetilde{\mathbf{V}}+\mathbf{a}^{\perp\top}(\mathbf{x}_0^*-\mathbf{x}_1^*))+\mathbf{x}_1^*} + \eta\cdot(\underbrace{1,0,...,0}_{n's})]_1,\\ C_3 &= [\mathbf{cV}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1, C_4 = [\mathbf{cV}_2]_1,\\ C_5 &= [\mathbf{cV}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{f}'^{t*} - \eta]_T).\end{aligned}$$

Therefore, Game_1 and Game_2 are identically distributed. Notably, it is required that for each $j \in |[\mathcal{F}]|$, we have $\mathbf{f}_j(\mathbf{x}_0^*) = \mathbf{f}_j(\mathbf{x}_1^*)$, thus $(\widetilde{\mathbf{V}} + \mathbf{a}^{\perp\top}(\mathbf{x}_0^* - \mathbf{x}_1^*))\mathbf{f}_j = \widetilde{\mathbf{V}}\mathbf{f}_j$.

Lemma 3. *We have* $|\mathsf{Adv}_{\mathsf{Game}_2,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_3,\mathcal{A}}(\lambda)| \le \mathsf{Adv}^{MDDH}_{\mathcal{B}_2,G_1}(\lambda)$*, where* $\mathcal{B}_2$ *is the adversary for the MDDH assumption in* G_1.

Proof. This proof is similar to the proof of Lemma 1, we omit it here.

Theorem 5 (Weakly Selective-SIM Security of Construction Π_{sRIPFE}). *The slotted RIPFE construction* Π_{sRIPFE} *with message space* $\mathcal{M} = \mathbb{Z}_p^{1\times n}$*, function space* $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n\times 1}\}_{j<n}$*, a function value upperbound* B*, and an a-priori fixed number of slots* $L = L(\lambda)$ *is weakly selective-SIM secure relying on the MDDH assumption.*

Remark 1. When the functions of the users that are corrupted by the adversary cover the entire function space, for Π_{sRIPFE}*, weakly selective-SIM security is equivalent to selective-SIM security.*

Proof.

Games. We define the following games, where Game_2 is the output of the simulator:

- Game_0: This is as the real scheme.
- Game_1: This is the same as Game_0, except that we change Setup and Enc with the challenge message $\mathbf{x}^*$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|})$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{AV}]_1, [\mathbf{AV}_1]_1, [\mathbf{AV}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + \mathbf{V}\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]});\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

  - $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}^*, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, and $\eta \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\mathbf{c}]_1, C_2 &= [\mathbf{cV} + \mathbf{x}^* + \eta \cdot \underbrace{(1, 0, ..., 0)}_{n's}]_1,\\ C_3 &= [\mathbf{cV}_1 + \mathbf{c}\sum_{t\in[L]} \mathbf{U}_t]_1, C_4 = [\mathbf{cV}_2]_1,\\ C_5 &= [\mathbf{cV}_2 \sum_{t\in[L]} \mathbf{W}_t\mathbf{f}'^{t} - \eta]_T).\end{aligned}$$

- Game_2: This is the same as Game_1, except that we change Setup and Enc with the challenge message $\mathbf{x}^*$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, \mathbf{x}^*)$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{AV}_1]_1, [\mathbf{AV}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + (\widetilde{\mathbf{V}} - \mathbf{a}^{\perp\top}\mathbf{x}^*)\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]});\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

  - $\mathsf{Enc}(\mathsf{mpk}, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, and $\eta \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\mathbf{c}]_1, C_2 &= [\mathbf{c}\widetilde{\mathbf{V}} + \eta \cdot \underbrace{(1, 0, ..., 0)}_{n's}]_1,\\ C_3 &= [\mathbf{cV}_1 + \mathbf{c}\sum_{t\in[L]} \mathbf{U}_t]_1, C_4 = [\mathbf{cV}_2]_1,\\ C_5 &= [\mathbf{cV}_2 \sum_{t\in[L]} \mathbf{W}_t\mathbf{f}'^{t} - \eta]_T).\end{aligned}$$

Lemma 4. *We have* $|\Pr[\mathsf{Game}_{0,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Game}_{1,\mathcal{A}}(\lambda) = 1]| \leq \mathsf{Adv}^{MDDH}_{\mathcal{B}_1, G_1}(\lambda)$, *where* $\mathcal{B}_1$ *is the adversary for the MDDH assumption in* G_1.

Proof. Suppose a challenger $\mathcal{B}_1$. The adversary $\mathcal{A}$ sends $\mathbf{x}^*$ and the number of slots L to $\mathcal{B}_1$. $\mathcal{B}_1$ receives $([\mathbf{A}]_1, [T]_1)$ from the underlying MDDH assumption. For $\mathbf{A}$, there exist $\mathbf{a}^{\perp} \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Then $\mathcal{B}_1$ interacts with $\mathcal{A}$ as follows:

$\mathcal{B}_1$ generates the output of Setup:

Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, and sample $\mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + \mathbf{V}\mathbf{f}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]});\\ \mathsf{msk} = (&\mathbf{V}, \mathbf{V}_1, \mathbf{V}_2, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

$\mathcal{B}_1$ sends crs to $\mathcal{A}$.

$\mathcal{B}_1$ generates the output of KeyGen for honest users:

Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, which satisfies each component of $\mathbf{U}_i$ is bounded by B. Output

$$\begin{aligned}\mathsf{pk}_{\mathsf{ctr}} &= ([\mathbf{A}\mathbf{U}_i]_1, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{f}_j]_2\}_{t\neq i, j\in[|\mathcal{F}|]});\\ \mathsf{sk}_{\mathsf{ctr}} &= \mathbf{U}_i.\end{aligned}$$

Upon receiving $(\{\mathbf{f}^{i*}, \mathsf{pk}_i^*\}_{i\in[L]})$ from $\mathcal{A}$, for each $i \in [L]$, $\mathcal{B}_1$ generates the output of $\mathsf{IsValid}$:

Check the validity of pk_i^*. If $0 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_1$ aborts. Otherwise, $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_1$ computes $\mathsf{sk}_i = \mathbf{U}_i$ with msk and brute-force discrete log (If $\mathbf{U}_i$ cannot be computed, $\mathcal{B}_1$ aborts. And note that only with negligible probability, $[\mathbf{U}_i]_T$ is not unique, in this case, $\mathcal{B}_1$ also aborts.).

$\mathcal{B}_1$ generates the output of Aggr:

Output

$$\begin{aligned}&\mathsf{mpk} = ([\mathbf{A}\sum_{i\in[L]}\mathbf{U}_i]_1, [\sum_{i\in[L]}\mathbf{W}_i\mathbf{f}^{'i*}]_2, [\mathbf{A}]_1, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1);\\ &\{\mathsf{hsk}_i = (i, \mathbf{f}^{i*}, [\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{f}^{'i*}]_2, [\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{f}^{'t*}]_2, [\mathbf{V}_1\mathbf{W}_i\mathbf{f}^{'i*} + \mathbf{V}_2\mathbf{W}_i\mathbf{f}^{'i*} + \mathbf{V}\mathbf{f}^{'i*}]_2,\\ &\qquad [\mathbf{W}_i\mathbf{f}^{'i*}]_2)\}_{i\in[L]}.\end{aligned}$$

$\mathcal{B}_1$ sends $(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$ to $\mathcal{A}$.

$\mathcal{B}_1$ generates the output of Enc:

Sample $\eta \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{ct}^* = (C_1 = [\boxed{T}]_1, C_2 = [\boxed{T}\mathbf{V} + \mathbf{x}^* + \eta \cdot \underbrace{(1, 0, ..., 0)}_{n's}]_1,\\ C_3 = [\boxed{T}\mathbf{V}_1 + \boxed{T}\sum_{t\in[L]}\mathbf{U}_t]_1, C_4 = [\boxed{T}\mathbf{V}_2]_1,\\ C_5 = [\boxed{T}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{f}'^{t*} - \eta]_T).\end{aligned}$$

$\mathcal{B}_1$ sends ct^* to $\mathcal{A}$.

Observe when $T = \mathbf{sA}$, where $\mathbf{s} \leftarrow_R \mathbb{Z}_p^{1\times k}$, the distributions are as Game_0; when $T = \mathbf{c}$, where $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$ satisfying $\mathbf{ca}^{\perp\top} = 1$, the distributions are as Game_1. Then if $\mathcal{A}$ can distinguish Game_0 and Game_1, $\mathcal{B}_1$ can utilize $\mathcal{A}$ to break the MDDH assumption in G_1. Thus lead to contradiction.

Lemma 5. *We have* $|\Pr[\mathsf{Game}_{1,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Game}_{2,\mathcal{A}}(\lambda) = 1| = 0$.

Proof. Sample $\widetilde{\mathbf{V}} \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$. Then change the variables by embedding the challenge $\mathbf{x}^*$ into $\mathbf{V}$ as follows:

$$\mathbf{V} = \widetilde{\mathbf{V}} - \mathbf{a}^{\perp\top}\mathbf{x}^*,$$

where the distributions are taken over the random choices of $\mathbf{V}$ and $\widetilde{\mathbf{V}}$.

We have $\mathbf{AV} = \mathbf{A}(\widetilde{\mathbf{V}} - \mathbf{a}^{\perp\top}\mathbf{x}^*) = \mathbf{A}\widetilde{\mathbf{V}}$. Then crs is changed into

$$\begin{aligned}\mathsf{crs} = (\mathbb{G}, [\mathbf{A}]_1, [\boxed{\mathbf{A}\widetilde{\mathbf{V}}}]_1, [\mathbf{AV}_1]_1, [\mathbf{AV}_2]_1, \{[\mathbf{W}_i\mathbf{f}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ \{[\mathbf{V}_1\mathbf{W}_i\mathbf{f}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{f}_j + \boxed{(\widetilde{\mathbf{V}} - \mathbf{a}^{\perp\top}\mathbf{x}^*)}\mathbf{f}_j]_2\}_{i\in[L],j\in[\mathcal{F}|]}).\end{aligned}$$

ct^* is changed into

$$\begin{aligned}\mathsf{ct}^* = (C_1 = [\mathbf{c}]_1, C_2 = [\underbrace{\boxed{\mathbf{c}\widetilde{\mathbf{V}}}}_{\mathbf{c}(\widetilde{\mathbf{V}}-\mathbf{a}^{\perp\top}\mathbf{x}^*)+\mathbf{x}^*} + \eta \cdot \underbrace{(1, 0, ..., 0)}_{n's}]_1,\\ C_3 = [\mathbf{cV}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1, C_4 = [\mathbf{cV}_2]_1,\\ C_5 = [\mathbf{cV}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{f}'^{t*} - \eta]_T).\end{aligned}$$

Therefore, Game_1 and Game_2 are identically distributed. Note that $(\widetilde{\mathbf{V}} - \mathbf{a}^{\perp\top}\mathbf{x}^*)\mathbf{f}_j = \widetilde{\mathbf{V}}\mathbf{f}_j - \mathbf{a}^{\perp\top}\mathbf{x}^*\mathbf{f}_j$, where $\mathbf{x}^*\mathbf{f}_j$ is the function value.

5 Slotted Registered Quadratic Functional Encryption

5.1 Construction $\mathit{\Pi}_{sRQFE}$

We construct a slotted Registered Quadratic Functional Encryption (slotted RQFE) scheme $\mathit{\Pi}_{sRQFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Aggr}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M} = \mathbb{Z}_p^{1\times n} \times \mathbb{Z}_p^{1\times n}$, function space $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n^2\times 1}\}$, where $n \in \mathbb{N}$, a function value upperbound B, and an a-priori fixed number of slots $L = L(\lambda)$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|})$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}, \mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}, \mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. Set

  $$\mathbf{H} = \begin{pmatrix} \mathbf{A}_1 \otimes \mathbf{I}_{n+1} \\ \mathbf{I}_{n+1} \otimes \mathbf{A}_2 \end{pmatrix} \in \mathbb{Z}_p^{(k+k')(n+1)\times(n+1)^2}.$$

  For each $\mathbf{f} = (\mathbf{f}_{11}, ..., \mathbf{f}_{1n}, \mathbf{f}_{21}, ..., \mathbf{f}_{2n}, ..., \mathbf{f}_{n1}, ..., \mathbf{f}_{nn}) \in \mathcal{F}$, set

  $$\begin{aligned} \hat{\mathbf{f}} = (&\hat{\mathbf{f}}_{11}, ..., \hat{\mathbf{f}}_{1n}, \hat{\mathbf{f}}_{1(n+1)}, \hat{\mathbf{f}}_{21}, ..., \hat{\mathbf{f}}_{2n}, \hat{\mathbf{f}}_{2(n+1)}, ..., \\ &\hat{\mathbf{f}}_{n1}, ..., \hat{\mathbf{f}}_{nn}, \hat{\mathbf{f}}_{n(n+1)}, \hat{\mathbf{f}}_{(n+1)1}, ..., \hat{\mathbf{f}}_{(n+1)n}, \hat{\mathbf{f}}_{(n+1)(n+1)}) \\ = (&\mathbf{f}_{11}, ..., \mathbf{f}_{1n}, 0, \mathbf{f}_{21}, ..., \mathbf{f}_{2n}, 0, ..., \mathbf{f}_{n1}, ..., \mathbf{f}_{nn}, 0, 0, ..., 0, 1). \end{aligned} \tag{2}$$

  Output

  $$\begin{aligned} \mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{AV}]_1, [\mathbf{AV}_1]_1, [\mathbf{AV}_2]_1, \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \\ &\{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{VH}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}). \end{aligned}$$

- $\mathsf{KeyGen}(\mathsf{crs}, i)$: Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, which satisfies each component of $\mathbf{U}_i$ is bounded by the brute-force discrete log bound B (used in the security proofs). Store $\mathsf{sk}_i = \mathbf{U}_i$. Output

  $$\mathsf{pk}_i = ([P_i^1]_1 = [\mathbf{AU}_i]_1, \{[P_{i,t,j}^2]_2\} = \{[\mathbf{U}_i\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{t\neq i,j\in[|\mathcal{F}|]}, \{[P_{i,t,j}^3]_2\} = \{[\mathbf{U}_i\mathbf{W}_t\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{t\neq i,j\in[|\mathcal{F}|]}).$$

- $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$: Check
  1. whether $[P_i^1]_1$ in pk_i are elements in G_1?
  2. whether $\{[P_{i,t,j}^2]_2\}$ and $\{[P_{i,t,j}^3]_2\}_{t\neq i,j\in|[\mathcal{F}]|}$ in pk_i are elements in G_2?
  3. whether $e([P_i^1]_1, [\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}_j]_2) = e([\mathbf{A}]_1, [P_{i,t,j}^2]_2)$, and $e([P_i^1]_1, [\mathbf{W}_t\mathbf{B}\hat{\mathbf{f}}_j]_2) = e([\mathbf{A}]_1, [P_{i,t,j}^3]_2)$ for each $t \neq i, j \in [|\mathcal{F}|]$?

  If any fails, output 0. Otherwise, output 1.

- $\mathsf{Aggr}(\mathsf{crs}, (\mathsf{pk}_1, \mathbf{f}^1), ..., (\mathsf{pk}_L, \mathbf{f}^L))$: For the target functions $\{\mathbf{f}^i\}_{i\in[L]}$, set $\hat{\mathbf{f}}^i$ as equation (2). Output

$$\begin{aligned}\mathsf{mpk} = (&[\mathbf{A}\sum_{i\in[L]}\mathbf{U}_i]_1, [\sum_{i\in[L]}\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{AV}]_1, [\mathbf{AV}_1]_1,\\ &[\mathbf{AV}_2]_1);\\ \{\mathsf{hsk}_i = (&i, \mathbf{f}^i, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2, [\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^i]_2,\\ &[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{VH}\hat{\mathbf{f}}^i]_2, [\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2)\}_{i\in[L]}.\end{aligned}$$

- $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{M})$: Sample $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Set

$$\hat{\mathbf{x}}_1 = (\mathbf{x}_1\|\eta_1), \hat{\mathbf{x}}_2 = (\mathbf{x}_2\|\eta_2).$$

Sample $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}, \mathbf{s}_2, \mathbf{s} \leftarrow_R \mathbb{Z}_p^{1\times k}$. Outout

$$\begin{aligned}\mathsf{ct} = (&C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{sA}]_1,\\ &C_4 = [\underbrace{\mathbf{sAV} + (\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2\|\mathbf{y}_1 \otimes \mathbf{s}_2)}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{sAV}_1 + \mathbf{sA}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ &C_6 = [\mathbf{sAV}_2]_1, C_7 = [-\mathbf{sAV}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

- $\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct})$: For the target $\mathbf{f}^i$, set $\hat{\mathbf{f}}^i$ as equation (2). Then compute $[\gamma]_T$ as follows:

$$\begin{aligned}&[(\mathbf{y}_1\otimes\mathbf{y}_2)\underbrace{\hat{\mathbf{f}}^i}_{\mathsf{hsk}_i}]_T \cdot e(C_4, \underbrace{[\mathbf{H}\hat{\mathbf{f}}^i]_2}_{\mathsf{hsk}_i})^{-1} \cdot e(C_3, \underbrace{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{VH}\hat{\mathbf{f}}^i]_2}_{\mathsf{hsk}_i})\cdot\\ &e(C_5, \underbrace{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2}_{\mathsf{hsk}_i})^{-1} \cdot e(C_3, \underbrace{[\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2}_{\mathsf{hsk}_i}) \cdot e(C_3, [\underbrace{\mathbf{U}_i}_{\mathsf{sk}_i}\underbrace{\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i}_{\mathsf{hsk}_i}]_2)\cdot\\ &e(C_6, \underbrace{[\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t]_2}_{\mathsf{hsk}_i}) \cdot C_7.\end{aligned}$$

Output γ by brute-force discrete log.

Theorem 6 (Completeness of Construction Π_{sRQFE}). *The slotted RQFE construction Π_{sRQFE} with message space $\mathcal{M} = \mathbb{Z}_p^{1\times n} \times \mathbb{Z}_p^{1\times n}$, function space $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n^2\times 1}\}$, where $n \in \mathbb{N}$, a function value upperbound B, and an a-priori fixed number of slots $L = L(\lambda)$ is complete.*

Proof. We proceed the checks as follows:

1. It is easy to see that $[P_i^1]_1$ in pk_i are elements in G_1.
2. It is easy to see that $\{[P_{i,t,j}^2]_2\}$ and $\{[P_{i,t,j}^3]_2\}$ in pk_i are elements in G_2.
3. For each $t \neq i, j \in [|\mathcal{F}|]$, we have

$$e([P_i^1]_1, [\mathbf{W}_t\mathbf{H}\mathbf{f}_j]_2) = [\mathbf{A}\mathbf{U}_i\mathbf{W}_t\mathbf{H}\mathbf{f}_j]_T = e([\mathbf{A}]_1, [P_{i,t,j}^2]_2)$$
$$e([P_i^1]_1, [\mathbf{W}_t\mathbf{B}\mathbf{f}_j]_2) = [\mathbf{A}\mathbf{U}_i\mathbf{W}_t\mathbf{B}\mathbf{f}_j]_T = e([\mathbf{A}]_1, [P_{i,t,j}^3]_2).$$

All the checks pass, thus, $\mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i)$ outputs 1 and completeness holds.

Theorem 7 (Compactness of Construction $\mathit{\Pi}_{sRQFE}$). *The slotted RQFE construction $\mathit{\Pi}_{sRQFE}$ with message space $\mathcal{M} = \mathbb{Z}_p^{1\times n} \times \mathbb{Z}_p^{1\times n}$, function space $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n^2\times 1}\}$, where $n \in \mathbb{N}$, a function value upperbound B, and an a-priori fixed number of slots $L = L(\lambda)$ satisfies the following properties:*

- $|\mathsf{mpk}| = (n+1)\cdot \mathsf{poly}(\lambda)$.
- *For each $i \in [L]$,* $|\mathsf{hsk}_i| = (n+1)\cdot \mathsf{poly}(\lambda) + n^2 \cdot \mathsf{poly}(\lambda) + O(\log L)$.

Proof. We can see in mpk, $[\mathbf{A}\sum_i \mathbf{U}_i]_1$ is of $k(k+k')(n+1)$ group elements, $[\sum_i \mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2$ is of $(k+k')(n+1)$ group elements, $[\mathbf{A}]_1$ is of $k(k+1)$ group elements, $[\mathbf{A}_1]_1$ is of $k'(n+1)$ group elements, $[\mathbf{A}_1]_2$ is of $k'(n+1)$ group elements, $[\mathbf{A}_2]_2$ is of $k(n+1)$ group elements, $[\mathbf{A}\mathbf{V}]_1$ is of $k(k+k')(n+1)$ group elements, $[\mathbf{A}\mathbf{V}_1]_1$ is of $k(k+k')(n+1)$ group elements, and $[\mathbf{A}\mathbf{V}_2]_1$ is of $k(k+k')(n+1)$ group elements. Each group element is of $\mathsf{poly}(\lambda)$ size. Thus, when we omit k and k', which are the parameters of the MDDH assumption and the bi-MDDH assumption respectively, the size of mpk is $(n+1)\cdot\mathsf{poly}(\lambda)$.

Similarly, in hsk_i, i is of $O(\log L)$ size, $\mathbf{f}^i$ is of n^2 ring elements, $[\mathbf{A}_1]_2$ is of $k'(n+1)$ group elements, $[\mathbf{A}_2]_2$ is of $k(n+1)$ group elements, $[\sum_{t\neq i, t\in[L]} \mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2$ is of $(k+1)$ group elements, $[\sum_{t\neq i, t\in[L]} \mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t]_2$ is of $(k+k')(n+1)$ group elements, $[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}^i]_2$ is of $(k+1)$ group elements, and $[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2$ is of $(k+k')(n+1)$ group elements. Each ring element or group element is of $\mathsf{poly}(\lambda)$ size. Thus, when we omit k and k', which are the parameters of the MDDH assumption and the bi-MDDH assumption respectively, the size of hsk_i is $(n+1)\cdot\mathsf{poly}(\lambda) + n^2\cdot\mathsf{poly}(\lambda) + O(\log L)$.

Theorem 8 (Correctness of Construction $\mathit{\Pi}_{sRQFE}$). *The slotted RQFE construction $\mathit{\Pi}_{sRQFE}$ with message space $\mathcal{M} = \mathbb{Z}_p^{1\times n} \times \mathbb{Z}_p^{1\times n}$, function space $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n^2\times 1}\}$, where $n \in \mathbb{N}$, a function value upperbound B, and an a-priori fixed number of slots $L = L(\lambda)$ is perfectly correct.*

Proof. We proceed the proof in two steps.

1. **We first demonstrate the correctness in the case that each $\mathsf{pk}_i, i \in [L]$ is of the right form.**
Fix some λ, message size $n = n(\lambda)$, the number of slots $L = L(\lambda)$, the size of function space $|\mathcal{F}|$, and an index $i \in [L]$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|})$ and $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, i)$ be defined as in Construction $\mathit{\Pi}_{sRQFE}$. Take

any set of public keys $\{\mathsf{pk}_j\}_{j\in[L]\setminus\{i\}}$, where $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, j, \mathsf{pk}_j)$. For each $j \in [L]$, let $\mathbf{f}^j$ be the target function associated with pk_j. Let $(\mathsf{mpk}, \{\mathsf{hsk}_i\}) \leftarrow \mathsf{Aggr}(\mathsf{crs}, \{(\mathsf{pk}_j, \mathbf{f}^j)\}_{j\in[L]})$. For a message $\mathbf{x} \in \mathcal{M}$, let $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathbf{x})$. Then the correctness follows:

$$\begin{aligned}
&[(\mathbf{y}_1 \otimes \mathbf{y}_2)\hat{\mathbf{f}}^i]_T \cdot e(C_4, [\mathbf{H}\hat{\mathbf{f}}^i]_2)^{-1} \cdot e(C_3, [\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}^i]_2)\cdot \\
&e(C_5, [\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2)^{-1} \cdot e(C_3, [\sum_{t\neq i, t\in[L]} \mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2) \cdot e(C_3, [\mathbf{U}_i\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2)\cdot \\
&e(C_6, [\sum_{t\neq i, t\in[L]} \mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t]_2) \cdot C_7 \\
=&[(\mathbf{y}_1 \otimes \mathbf{y}_2)\hat{\mathbf{f}}^i - \mathbf{sAVH}\hat{\mathbf{f}}^i - (\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2 \| \mathbf{y}_1 \otimes \mathbf{s}_2)\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{sAV}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{sAV}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \\
&\mathbf{sAVH}\hat{\mathbf{f}}^i - \mathbf{sAV}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i - \mathbf{sA}\sum_{t\in[L]} \mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{sA}\sum_{t\neq i, t\in[L]} \mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{sAU}_i\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \\
&\mathbf{sAV}_2\sum_{t\neq i, t\in[L]} \mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \mathbf{sAV}_2\sum_{t\in[L]} \mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1 \cdot \eta_2]_T \\
=&[(\mathbf{y}_1 \otimes \mathbf{y}_2)\hat{\mathbf{f}}^i - \eta_1 \cdot \eta_2]_T \\
=&[(\mathbf{x}_1 \otimes \mathbf{x}_2)\mathbf{f}^i + \eta_1 \cdot \eta_2 - \eta_1 \cdot \eta_2]_T \\
=&[(\mathbf{x}_1 \otimes \mathbf{x}_2)\mathbf{f}^i]_T.
\end{aligned}$$

2. **We then demonstrate that correctness holds as long as $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, j, \mathsf{pk}_j)$.** We replace $[\mathbf{AU}_i]_1, [\mathbf{U}_i\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}_j]_2$ and $[\mathbf{U}_i\mathbf{W}_t\mathbf{B}_t\hat{\mathbf{f}}_j]_2$ with $[P_i^1]_1, [P_{i,t,j}^2]_2$ and $[P_{i,t,j}^3]_2$, respectively. Then, we have

$$\begin{aligned}
&[(\mathbf{y}_1 \otimes \mathbf{y}_2)\hat{\mathbf{f}}^i]_T \cdot e(C_4, [\mathbf{H}\hat{\mathbf{f}}^i]_2)^{-1} \cdot e(C_3, [\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}^i]_2)\cdot \\
&e(C_5, [\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2)^{-1} \cdot e(C_3, [\sum_{t\neq i, t\in[L]} \mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2) \cdot e(C_3, [\mathbf{U}_i\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_2)\cdot \\
&e(C_6, [\sum_{t\neq i, t\in[L]} \mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t]_2) \cdot C_7 \\
=&[(\mathbf{y}_1 \otimes \mathbf{y}_2)\hat{\mathbf{f}}^i - \mathbf{sAVH}\hat{\mathbf{f}}^i - (\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2 \| \mathbf{y}_1 \otimes \mathbf{s}_2)\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{sAV}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{sAV}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \\
&\mathbf{sAVH}\hat{\mathbf{f}}^i - \mathbf{sAV}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i - \mathbf{s}\sum_{t\in[L]} P_t^1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \mathbf{sA}\sum_{t\neq i, t\in[L]} P_{t,i,i}^2 + \mathbf{sAU}_i\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i + \\
&\mathbf{sAV}_2\sum_{t\neq i, t\in[L]} \mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \mathbf{sAV}_2\sum_{t\in[L]} \mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1 \cdot \eta_2]_T. \qquad (3)
\end{aligned}$$

By $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, j, \mathsf{pk}_j)$, we have

$$[P_t^1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_T = [\mathbf{A}P_{t,i,i}^2]_T,$$

for each $t \neq i$. Thus we have

$$[\mathbf{s}\sum_{t\neq i} P_t^1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_T = [\mathbf{sA}\sum_{t\neq i} P_{t,i,i}^2]_T.$$

Since the rest $[\mathbf{s}P_i^1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_T \overset{?}{=} [\mathbf{s}\mathbf{A}\mathbf{U}_i\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^i]_T$ depends on whether the adversary desires to proceed a successful decryption, we don't need to consider it. Thus the formula (3) equals $[(\mathbf{x}_1 \otimes \mathbf{x}_2)\mathbf{f}^i]_T$.

Theorem 9 (Weakly Selective-IND Security of Construction $\mathit{\Pi}_{sRQFE}$). *The slotted RQFE construction $\mathit{\Pi}_{sRQFE}$ with message space $\mathcal{M} = \mathbb{Z}_p^{1\times n} \times \mathbb{Z}_p^{1\times n}$, function space $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n^2\times 1}\}$, a function value upperbound B, and an a-priori fixed number of slots $L = L(\lambda)$ is weakly selective-IND secure relying on the MDDH assumption and bi-MDDH assumption.*

Proof.
Games. We define the following games:

- Game_0: This is the game with the real construction and choosing $(\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as the encrypted message.
- Game_1: This is the same as Game_0, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|})$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}, \mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}, \mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Output
  $$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$
  - $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}, \mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output
  $$\begin{aligned}\mathsf{ct} = (C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(1)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(1)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\mathbf{V} + (\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(1)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$
- Game_2: This is the same as Game_1, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}, \mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}, \mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$

for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}, \mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(1)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

- $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$.

$$\begin{aligned}\mathsf{ct} = (C_1 = [&\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(1)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(1)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [&\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [&\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

– Game_3: This is the same as Game_2, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:

  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}, \mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}, \mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}, \mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^{(1)} \otimes \hat{\mathbf{x}}_2^{(1)})\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

  - $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [&\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(1)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(1)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [&\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [&\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

- Game_4: This is the same as Game_3, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$, and $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^{(1)} \otimes \hat{\mathbf{x}}_2^{(1)})\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

  - $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_2^{(1)}, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\mathbf{y}_1]_1, C_2 &= [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(1)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 &= [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 &= [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

- Game_5: This is the same as Game_4, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, $\mathbf{y}_2 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^{(1)} \otimes \hat{\mathbf{x}}_2^{(1)})\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\mathbf{y}_1]_1, C_2 = [\mathbf{y}_2]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

– Game_6: This is the same as Game_5, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, $\mathbf{y}_2 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^{(0)} \otimes \hat{\mathbf{x}}_2^{(0)})\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\mathbf{y}_1]_1, C_2 = [\mathbf{y}_2]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

– Game_7: This is the same as Game_6, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$.

Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^{(0)} \otimes \hat{\mathbf{x}}_2^{(0)})\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_2^{(0)}, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\mathbf{y}_1]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(0)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

– Game_8: This is the same as Game_7, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}$, $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^{(0)} \otimes \hat{\mathbf{x}}_2^{(0)})\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

- $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(0)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(0)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

– Game_9: This is the same as Game_8, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}$, $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(0)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\end{aligned}$$

$$\mathsf{msk} = (\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).$$

- $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct} = (&C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(0)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(0)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ &C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ &C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

– Game_{10}: This is the same as Game_9, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}', \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}']_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}'\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\end{aligned}$$

$$\mathsf{msk} = (\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}', \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).$$

- $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}$, $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{ct} = (&C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(0)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(0)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ &C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}' + (\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(0)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ &C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

- Game_{11}: This is the same as Game_{10}, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}', \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}']_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}'\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}', \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

  - $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}$, $\mathbf{s}_2, \mathbf{s} \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [&\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(0)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(0)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{s}\mathbf{A}]_1,\\ C_4 &= [\underbrace{\mathbf{s}\mathbf{A}\widetilde{\mathbf{V}}' + (\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(0)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{s}\mathbf{A}\mathbf{V}_1 + \mathbf{s}\mathbf{A}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 &= [\mathbf{s}\mathbf{A}\mathbf{V}_2]_1, C_7 = [-\mathbf{s}\mathbf{A}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

Lemma 6. *We have* $|\mathsf{Adv}_{\mathsf{Game}_0,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_1,\mathcal{A}}(\lambda)| \le \mathsf{Adv}^{MDDH}_{\mathcal{B}_1,G_1}(\lambda)$, *where* $\mathcal{B}_1$ *is the adversary for the MDDH assumption in* G_1.

Proof. Suppose a challenger $\mathcal{B}_1$. The adversary $\mathcal{A}$ sends $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ and the number of slots L to $\mathcal{B}_1$. $\mathcal{B}_1$ runs the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, and samples $\mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. $\mathcal{B}_1$ receives $([\mathbf{A}]_1, [T]_1)$ from the underlying MDDH assumption in G_1. For $\mathbf{A}$, there exist $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. For each $\mathbf{f} \in \mathcal{F}$, $\mathcal{B}_1$ sets $\hat{\mathbf{f}}$ as equation (2). Then $\mathcal{B}_1$ generates

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]});\\ \mathsf{msk} = (&\mathbf{A}_1, \mathbf{A}_2, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

$\mathcal{B}_1$ initializes a counter $\mathsf{ctr} = 0$, a dictionary $\mathsf{D} = \emptyset$, and a set of corrupted slot indexes $\mathcal{C} = \emptyset$. $\mathcal{B}_1$ sends crs to $\mathcal{A}$.

Upon receiving an honest key generation query $i \in [L]$ from $\mathcal{A}$, $\mathcal{B}_1$ sets $\mathsf{ctr} = \mathsf{ctr} + 1$, and generates $(\mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}})$ as follows:

Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, which satisfies each component of $\mathbf{U}_i$ is bounded by B. Set

$$\begin{aligned}&\mathsf{pk}_{\mathsf{ctr}} = ([\mathbf{A}\mathbf{U}_i]_1, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{t\neq i, j\in[|\mathcal{F}|]}, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{t\neq i, j\in[|\mathcal{F}|]});\\ &\mathsf{sk}_{\mathsf{ctr}} = \mathbf{U}_i.\end{aligned}$$

Upon receiving a corruption query $j \in [\mathsf{ctr}]$ from $\mathcal{A}$, $\mathcal{B}_1$ sends sk' to $\mathcal{A}$, where $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[j]$. Let $\mathcal{Q}_{Corr}$ be the set of corruption queries made by $\mathcal{A}$.

Upon receiving the challenge $(\{c_i^*, \mathbf{f}^{i*}, \mathsf{pk}_i^*\}_{i\in[L]})$ from $\mathcal{A}$, where $c_i^* \in [\mathsf{ctr}] \cup \{\perp\}$. Then for each $i \in [L]$, $\mathcal{B}_1$ proceeds as follows:

- If $c_i^* \in [\mathsf{ctr}]$, $\mathcal{B}_1$ retrieves $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[c_i^*]$. If $i' = i$, $\mathcal{B}_1$ sets $\mathsf{pk}_i = \mathsf{pk}'$. In addition, if $c_i^* \in \mathcal{Q}$, $\mathcal{B}_1$ updates $\mathcal{C} = \mathcal{C} \cup \{i\}$. Otherwise, if $i' \neq i$, $\mathcal{B}_1$ aborts.
- If $c_i^* = \perp$, $\mathcal{B}_1$ checks the validity of pk_i^*. If $0 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_1$ aborts. Otherwise, $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_1$ computes $\mathsf{sk}_i = \mathbf{U}_i$ with msk and brute-force discrete log (If $\mathbf{U}_i$ cannot be computed, $\mathcal{B}_1$ aborts. And note that only with negligible probability, $[\mathbf{U}_i]_T$ is not unique, in this case, $\mathcal{B}_1$ also aborts.). Then $\mathcal{B}_1$ sets $\mathsf{pk}_i = \mathsf{pk}_i^*$ and updates $\mathcal{C} = \mathcal{C} \cup \{i\}$.

$\mathcal{B}_1$ samples $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}, \mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}, \eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$ and generates ct^* as follows:

$$\begin{aligned}&C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(1)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(1)}}_{\mathbf{y}_2}]_2, C_3 = [\boxed{T}]_1,\\ &C_4 = [\underbrace{\boxed{T}\mathbf{V} + (\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(1)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)}_{\mathbf{y}_0}]_1, C_5 = [\boxed{T}\mathbf{V}_1 + \boxed{T}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ &C_6 = [\boxed{T}\mathbf{V}_2]_1, C_7 = [-\boxed{T}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{t*} - \eta_1\cdot\eta_2]_T).\end{aligned}$$

$\mathcal{B}_1$ generates $\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]}$ as follows:

$$\begin{aligned}\mathsf{mpk} = ([&\mathbf{A}\sum_{i\in[L]}\mathbf{U}_i]_1, [\sum_{i\in[L]}\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1,\\ &[\mathbf{A}\mathbf{V}_2]_1);\\ \{\mathsf{hsk}_i = (&i, \mathbf{f}^{i*}, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{i*}]_2,\\ &[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2)\}_{i\in[L]}.\end{aligned}$$

$\mathcal{B}_1$ sends $(\mathsf{ct}^*, \mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$ to $\mathcal{A}$.

Observe when $T = \mathbf{s}\mathbf{A}$, where $\mathbf{s} \leftarrow_R \mathbb{Z}_p^{1\times k}$, the distributions are as Game_0; when $T = \mathbf{c}$, where $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$ satisfying $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, the distributions are as Game_1. Then if $\mathcal{A}$ can distinguish Game_0 and Game_1, $\mathcal{B}_1$ can utilize $\mathcal{A}$ to break the MDDH assumption in G_1. Thus lead to contradiction.

Lemma 7. *We have* $|\mathsf{Adv}_{\mathsf{Game}_1,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_2,\mathcal{A}}(\lambda)| = 0$.

Proof. Sample $\widetilde{\mathbf{V}} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$. Then change the variables $\mathbf{V}$ as follows:

$$\mathbf{V} = \widetilde{\mathbf{V}} - \mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \mathbf{x}_2^{(1)} \| \mathbf{y}_1 \otimes \mathbf{s}_2),$$

where the distributions are taken over the random choices of $\mathbf{V}$ and $\widetilde{\mathbf{V}}$.

We have $\mathbf{A}\mathbf{V} = \mathbf{A}(\widetilde{\mathbf{V}} - \mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \mathbf{x}_2^{(1)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)) = \mathbf{A}\widetilde{\mathbf{V}}$. Then crs is changed into

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\boxed{\widetilde{\mathbf{V}}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \boxed{\widetilde{\mathbf{V}}}\mathbf{H}\hat{\mathbf{f}}_j \boxed{-\mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(1)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)}\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\end{aligned}$$

ct^* is changed into

$$\begin{aligned}\mathsf{ct}^* = (C_1 = [&\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(1)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(1)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [&\underbrace{\mathbf{c}\boxed{\widetilde{\mathbf{V}}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [&\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{t*} - \eta_1\cdot\eta_2]_T).\end{aligned}$$

Therefore, Game_1 and Game_2 are identically distributed.

Lemma 8. *We have* $|\mathsf{Adv}_{\mathsf{Game}_2,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_3,\mathcal{A}}(\lambda)| = 0$.

Proof. It is easy to see

$$\begin{aligned}&(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(1)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)\mathbf{H}\hat{\mathbf{f}}_j\\ =&(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(1)})(\mathbf{A}_1 \otimes \mathbf{I}_{n+1}) + (\mathbf{y}_1 \otimes \mathbf{s}_2)(\mathbf{I}_{n+1} \otimes \mathbf{A}_2)\\ =&(\mathbf{s}_1\mathbf{A}_1) \otimes \hat{\mathbf{x}}_2^{(1)} + (\mathbf{s}_1\mathbf{A}_1) \otimes (\mathbf{s}_2\mathbf{A}_2) + \hat{\mathbf{x}}_1^{(1)} \otimes (\mathbf{s}_2\mathbf{A}_2)\\ =&(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^{(1)} \otimes \hat{\mathbf{x}}_2^{(1)})\mathbf{f}_j.\end{aligned}$$

Therefore, Game_2 and Game_3 are identically distributed.

Lemma 9. *We have* $|\mathsf{Adv}_{\mathsf{Game}_3,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_4,\mathcal{A}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_2}^{bi\text{-}MDDH}(\lambda)$*, where* $\mathcal{B}_2$ *is the adversary for the bi-MDDH assumption.*

Proof. Suppose a challenger $\mathcal{B}_2$. The adversary $\mathcal{A}$ sends $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ and the number of slots L to $\mathcal{B}_2$. $\mathcal{B}_2$ runs the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, and samples $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$, and $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}, \eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. $\mathcal{B}_2$ receives $([\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [T]_1, [T]_2)$ from the underlying bi-MDDH assumption. For $\mathbf{A}$, there exist $\mathbf{a}^{\perp} \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. For each $\mathbf{f} \in \mathcal{F}$, $\mathcal{B}_2$ sets $\hat{\mathbf{f}}$ as equation (2). Then $\mathcal{B}_2$ generates

$$\begin{aligned}
\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\
&\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\
&\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}((\boxed{T} + \hat{\mathbf{x}}_1^{(1)}) \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^{(1)} \otimes \hat{\mathbf{x}}_2^{(1)})\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]});\\
\mathsf{msk} = (&\mathbf{A}, \mathbf{a}^{\perp}, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).
\end{aligned}$$

$\mathcal{B}_2$ initializes a counter $\mathsf{ctr} = 0$, a dictionary $\mathsf{D} = \emptyset$, and a set of corrupted slot indexes $\mathcal{C} = \emptyset$. $\mathcal{B}_2$ sends crs to $\mathcal{A}$.

Upon receiving an honest key generation query $i \in [L]$ from $\mathcal{A}$, $\mathcal{B}_2$ sets $\mathsf{ctr} = \mathsf{ctr} + 1$, and generates $(\mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}})$ as follows:

Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, which satisfies each component of $\mathbf{U}_i$ is bounded by B. Set

$$\begin{aligned}
\mathsf{pk}_{\mathsf{ctr}} &= ([\mathbf{A}\mathbf{U}_i]_1, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{t\neq i,j\in[|\mathcal{F}|]}, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{t\neq i,j\in[|\mathcal{F}|]});\\
\mathsf{sk}_{\mathsf{ctr}} &= \mathbf{U}_i.
\end{aligned}$$

Upon receiving a corruption query $j \in [\mathsf{ctr}]$ from $\mathcal{A}$, $\mathcal{B}_2$ sends sk' to $\mathcal{A}$, where $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[j]$. Let $\mathcal{Q}_{Corr}$ be the set of corruption queries made by $\mathcal{A}$.

Upon receiving the challenge $(\{c_i^*, \mathbf{f}^{i*}, \mathsf{pk}_i^*\}_{i\in[L]})$ from $\mathcal{A}$, where $c_i^* \in [\mathsf{ctr}] \cup \{\perp\}$. Then for each $i \in [L]$, $\mathcal{B}_2$ proceeds as follows:

- If $c_i^* \in [\mathsf{ctr}]$, $\mathcal{B}_2$ retrieves $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[c_i^*]$. If $i' = i$, $\mathcal{B}_2$ sets $\mathsf{pk}_i = \mathsf{pk}'$. In addition, if $c_i^* \in \mathcal{Q}$, $\mathcal{B}_2$ updates $\mathcal{C} = \mathcal{C} \cup \{i\}$. Otherwise, if $i' \neq i$, $\mathcal{B}_2$ aborts.
- If $c_i^* = \perp$, $\mathcal{B}_2$ checks the validity of pk_i^*. If $0 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_2$ aborts. Otherwise, $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_2$ computes $\mathsf{sk}_i = \mathbf{U}_i$ with msk and brute-force discrete log (If $\mathbf{U}_i$ cannot be computed, $\mathcal{B}_2$ aborts. And note that only with negligible probability, $[\mathbf{U}_i]_T$ is not unique, in this case, $\mathcal{B}_2$ also aborts.). Then $\mathcal{B}_2$ sets $\mathsf{pk}_i = \mathsf{pk}_i^*$ and updates $\mathcal{C} = \mathcal{C} \cup \{i\}$.

$\mathcal{B}_2$ samples $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Generate ct^* as follows:

$$\begin{aligned}
&C_1 = [\boxed{T} + \hat{\mathbf{x}}_1^{(1)}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(1)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\
&C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]} \mathbf{U}_t]_1,\\
&C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2 \sum_{t\in[L]} \mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{t*} - \eta_1 \cdot \eta_2]_T).
\end{aligned}$$

$\mathcal{B}_2$ generates $\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]}$ as follows:

$$\begin{aligned}\mathsf{mpk} = ([\mathbf{A}\sum_{i\in[L]}\mathbf{U}_i]_1, [\sum_{i\in[L]}\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1,\\ [\mathbf{A}\mathbf{V}_2]_1);\\ \{\mathsf{hsk}_i = (i, \mathbf{f}^{i*}, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{i*}]_2,\\ [\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2)\}_{i\in[L]}.\end{aligned}$$

$\mathcal{B}_2$ sends $(\mathsf{ct}^*, \mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$ to $\mathcal{A}$.

Observe when $T = \mathbf{s}_1\mathbf{A}_1$, where $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}$, the distributions are as Game_3; when $T = \mathbf{y}_1'$, where $\mathbf{y}_1' \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$ satisfying $\mathbf{y}_1' + \hat{\mathbf{x}}_1^{(1)} = \mathbf{y}_1$ and $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, the distributions are as Game_4. Then if $\mathcal{A}$ can distinguish Game_3 and Game_4, $\mathcal{B}_2$ can utilize $\mathcal{A}$ to break the bi-MDDH assumption. Thus lead to contradiction.

Lemma 10. *We have* $|\mathsf{Adv}_{\mathsf{Game}_4,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_5,\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{MDDH}_{\mathcal{B}_3,G_2}(\lambda)$, *where* $\mathcal{B}_3$ *is the adversary for the MDDH assumption in* G_2.

Proof. Suppose a challenger $\mathcal{B}_3$. The adversary $\mathcal{A}$ sends $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ and the number of slots L to $\mathcal{B}_3$. $\mathcal{B}_3$ runs the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, and samples $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$, and $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}, \eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. $\mathcal{B}_3$ receives $([\mathbf{A}_2]_2, [T]_2)$ from the underlying MDDH assumption in G_2. For $\mathbf{A}$, there exist $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. For each $\mathbf{f} \in \mathcal{F}$, $\mathcal{B}_3$ sets $\hat{\mathbf{f}}$ as equation (2). Then $\mathcal{B}_3$ generates

$$\begin{aligned}\mathsf{crs} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes (\boxed{T} + \hat{\mathbf{x}}_2^{(1)}) - \hat{\mathbf{x}}_1^{(1)} \otimes \hat{\mathbf{x}}_2^{(1)})\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]});\\ \mathsf{msk} = (\mathbf{A}, \mathbf{a}^\perp, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

$\mathcal{B}_3$ initializes a counter $\mathsf{ctr} = 0$, a dictionary $\mathsf{D} = \emptyset$, and a set of corrupted slot indexes $\mathcal{C} = \emptyset$. $\mathcal{B}_3$ sends crs to $\mathcal{A}$.

Upon receiving an honest key generation query $i \in [L]$ from $\mathcal{A}$, $\mathcal{B}_3$ sets $\mathsf{ctr} = \mathsf{ctr} + 1$, and generates $(\mathsf{pk}_{\mathsf{ctr}}, \mathsf{sk}_{\mathsf{ctr}})$ as follows:

Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, which satisfies each component of $\mathbf{U}_i$ is bounded by B. Set

$$\begin{aligned}\mathsf{pk}_{\mathsf{ctr}} = ([\mathbf{A}\mathbf{U}_i]_1, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{t\neq i, j\in[|\mathcal{F}|]}, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{t\neq i, j\in[|\mathcal{F}|]});\\ \mathsf{sk}_{\mathsf{ctr}} = \mathbf{U}_i.\end{aligned}$$

Upon receiving a corruption query $j \in [\mathsf{ctr}]$ from $\mathcal{A}$, $\mathcal{B}_3$ sends sk' to $\mathcal{A}$, where $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[j]$. Let $\mathcal{Q}_{Corr}$ be the set of corruption queries made by $\mathcal{A}$.

Upon receiving the challenge $(\{c_i^*, \mathbf{f}^{i*}, \mathsf{pk}_i^*\}_{i \in [L]})$ from $\mathcal{A}$, where $c_i^* \in [\mathsf{ctr}] \cup \{\perp\}$. Then for each $i \in [L]$, $\mathcal{B}_3$ proceeds as follows:

- If $c_i^* \in [\mathsf{ctr}]$, $\mathcal{B}_3$ retrieves $(i', \mathsf{pk}', \mathsf{sk}') = \mathsf{D}[c_i^*]$. If $i' = i$, $\mathcal{B}_3$ sets $\mathsf{pk}_i = \mathsf{pk}'$. In addition, if $c_i^* \in \mathcal{Q}$, $\mathcal{B}_3$ updates $\mathcal{C} = \mathcal{C} \cup \{i\}$. Otherwise, if $i' \neq i$, $\mathcal{B}_3$ aborts.
- If $c_i^* = \perp$, $\mathcal{B}_3$ checks the validity of pk_i^*. If $0 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_3$ aborts. Otherwise, $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_3$ computes $\mathsf{sk}_i = \mathbf{U}_i$ with msk and brute-force discrete log (If $\mathbf{U}_i$ cannot be computed, $\mathcal{B}_3$ aborts. And note that only with negligible probability, $[\mathbf{U}_i]_T$ is not unique, in this case, $\mathcal{B}_3$ also aborts.). Then $\mathcal{B}_3$ sets $\mathsf{pk}_i = \mathsf{pk}_i^*$ and updates $\mathcal{C} = \mathcal{C} \cup \{i\}$.

$\mathcal{B}_3$ samples $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1 \times (k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Generate ct^* as follows:

$$\begin{aligned}
&C_1 = [\mathbf{y}_1]_1, C_2 = [\boxed{T} + \hat{\mathbf{x}}_2^{(1)}]_2, C_3 = [\mathbf{c}]_1,\\
&C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t \in [L]} \mathbf{U}_t]_1,\\
&C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2 \sum_{t \in [L]} \mathbf{W}_t \mathbf{H}\hat{\mathbf{f}}^{t*} - \eta_1 \cdot \eta_2]_T).
\end{aligned}$$

$\mathcal{B}_3$ generates $\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i \in [L]}$ as follows:

$$\begin{aligned}
\mathsf{mpk} = (&[\mathbf{A}\sum_{i \in [L]} \mathbf{U}_i]_1, [\sum_{i \in [L]} \mathbf{W}_i \mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1,\\
&[\mathbf{A}\mathbf{V}_2]_1);\\
\{\mathsf{hsk}_i = (&i, \mathbf{f}^{i*}, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\sum_{t \neq i, t \in [L]} \mathbf{U}_t \mathbf{W}_i \mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\sum_{t \neq i, t \in [L]} \mathbf{W}_t \mathbf{H}\hat{\mathbf{f}}^{i*}]_2,\\
&[\mathbf{V}_1 \mathbf{W}_i \mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}_2 \mathbf{W}_i \mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{W}_i \mathbf{H}\hat{\mathbf{f}}^{i*}]_2)\}_{i \in [L]}.
\end{aligned}$$

$\mathcal{B}_3$ sends $(\mathsf{ct}^*, \mathsf{mpk}, \{\mathsf{hsk}_i\}_{i \in [L]})$ to $\mathcal{A}$.

Observe when $T = \mathbf{s}_2\mathbf{A}_2$, where $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1 \times k}$, the distributions are as Game_4; when $T = \mathbf{y}_2'$, where $\mathbf{y}_2' \leftarrow_R \mathbb{Z}_p^{1 \times (n+1)}$ satisfying $\mathbf{y}_2' + \hat{\mathbf{x}}_2^{(1)} = \mathbf{y}_2$ and $\mathbf{y}_2 \leftarrow_R \mathbb{Z}_p^{1 \times (n+1)}$, the distributions are as Game_5. Then if $\mathcal{A}$ can distinguish Game_4 and Game_5, $\mathcal{B}_3$ can utilize $\mathcal{A}$ to break the MDDH assumption G_2. Thus lead to contradiction.

Lemma 11. *We have* $|\mathsf{Adv}_{\mathsf{Game}_5, \mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_6, \mathcal{A}}(\lambda)| = 0$.

Proof. This lemma follows that for the challenge messages $(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}), (\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$, for each $i \in [|\mathcal{F}|]$, we have

$$(\mathbf{x}_1^{(0)} \otimes \mathbf{x}_2^{(0)})\mathbf{f}_j = (\mathbf{x}_1^{(1)} \otimes \mathbf{x}_2^{(1)})\mathbf{f}_j.$$

Thus we have
$(\hat{\mathbf{x}}_1^{(0)} \otimes \hat{\mathbf{x}}_2^{(0)})\hat{\mathbf{f}}_j = (\mathbf{x}_1^{(0)} \otimes \mathbf{x}_2^{(0)})\mathbf{f}_j + \eta_1 \cdot \eta_2 = (\mathbf{x}_1^{(1)} \otimes \mathbf{x}_2^{(1)})\mathbf{f}_j + \eta_1 \cdot \eta_2 = (\hat{\mathbf{x}}_1^{(1)} \otimes \hat{\mathbf{x}}_2^{(1)})\hat{\mathbf{f}}_j$.
Therefore, Game_5 and Game_6 are identically distributed.

Lemma 12. *We have* $|\mathsf{Adv}_{\mathsf{Game}_6,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_7,\mathcal{A}}(\lambda)| \le \mathsf{Adv}^{MDDH}_{\mathcal{B}_4,G_2}(\lambda)$*, where* $\mathcal{B}_4$ *is the adversary for the MDDH assumption in* G_2*.*

Proof. This proof is similar to the proof of Lemma 10, we omit it here.

Lemma 13. *We have* $|\mathsf{Adv}_{\mathsf{Game}_7,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_8,\mathcal{A}}(\lambda)| \le \mathsf{Adv}^{bi\text{-}MDDH}_{\mathcal{B}_5}(\lambda)$*, where* $\mathcal{B}_5$ *is the adversary for the bi-MDDH assumption.*

Proof. This proof is similar to the proof of Lemma 9, we omit it here.

Lemma 14. *We have* $|\mathsf{Adv}_{\mathsf{Game}_8,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_9,\mathcal{A}}(\lambda)| = 0$*.*

Proof. It is easy to see

$$\begin{aligned}
&(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(0)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)\mathbf{H}\hat{\mathbf{f}}_j \\
=&(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(0)})(\mathbf{A}_1 \otimes \mathbf{I}_{n+1}) + (\mathbf{y}_1 \otimes \mathbf{s}_2)(\mathbf{I}_{n+1} \otimes \mathbf{A}_2) \\
=&(\mathbf{s}_1\mathbf{A}_1) \otimes \hat{\mathbf{x}}_2^{(0)} + (\mathbf{s}_1\mathbf{A}_1) \otimes (\mathbf{s}_2\mathbf{A}_2) + \hat{\mathbf{x}}_1^{(0)} \otimes (\mathbf{s}_2\mathbf{A}_2) \\
=&(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^{(0)} \otimes \hat{\mathbf{x}}_2^{(0)})\mathbf{f}_j.
\end{aligned}$$

Therefore, Game_8 and Game_9 are identically distributed.

Lemma 15. *We have* $|\mathsf{Adv}_{\mathsf{Game}_9,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_{10},\mathcal{A}}(\lambda)| = 0$*.*

Proof. Sample $\widetilde{\mathbf{V}}' \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$. Then change the variables $\widetilde{\mathbf{V}}$ as follows:

$$\widetilde{\mathbf{V}} = \widetilde{\mathbf{V}}' + \mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \mathbf{x}_2^{(0)} \| \mathbf{y}_1 \otimes \mathbf{s}_2),$$

where the distributions are taken over the random choices of $\widetilde{\mathbf{V}}$ and $\widetilde{\mathbf{V}}'$.

We have $\mathbf{A}\widetilde{\mathbf{V}} = \mathbf{A}(\widetilde{\mathbf{V}}' + \mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \mathbf{x}_2^{(0)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)) = \mathbf{A}\widetilde{\mathbf{V}}'$. Then crs is changed into

$$\begin{aligned}
\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\boxed{\widetilde{\mathbf{V}}'}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \\
&\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \\
&\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \boxed{\widetilde{\mathbf{V}}'}\mathbf{H}\hat{\mathbf{f}}_j\boxed{\cancel{-\mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^{(0)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)\mathbf{H}\hat{\mathbf{f}}_j}}]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));
\end{aligned}$$

ct^* is changed into

$$\begin{aligned}
\mathsf{ct}^* = (&C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^{(0)}}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^{(0)}}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1, \\
&C_4 = [\underbrace{\mathbf{c}\boxed{\widetilde{\mathbf{V}}' + (\mathbf{s}_1 \otimes \mathbf{x}_2^{(0)} \| \mathbf{y}_1 \otimes \mathbf{s}_2)}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1, \\
&C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{t*} - \eta_1 \cdot \eta_2]_T).
\end{aligned}$$

Therefore, Game_9 and Game_{10} are identically distributed.

Lemma 16. *We have* $|\mathsf{Adv}_{\mathsf{Game}_{10},\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_{11},\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{MDDH}_{\mathcal{B}_6,G_1}(\lambda)$, *where* $\mathcal{B}_6$ *is the adversary for the MDDH assumption in* G_1.

Proof. This proof is similar to the proof of Lemma 6, we omit it here.

Theorem 10 (Weakly Selective-SIM Security of Construction Π_{sRQFE}). *The slotted RQFE construction* Π_{sRQFE} *with message space* $\mathcal{M} = \mathbb{Z}_p^{1\times n} \times \mathbb{Z}_p^{1\times n}$, *function space* $\mathcal{F} = \{\mathbf{f}_j \in \mathbb{Z}_p^{n^2\times 1}\}$, *a function value upperbound* B, *and an a-priori fixed number of slots* $L = L(\lambda)$ *is weakly selective-SIM secure relying on the MDDH assumption and bi-MDDH assumption.*

Remark 2. When the functions of the users that are corrupted by the adversary cover the entire function space, for Π_{sRQFE}, *weakly selective-SIM security is equivalent to selective-SIM security.*

Proof.

Games. We define the following games, where Game_5 is the output of the simulator:

- Game_0: This is as the real scheme.
- Game_1: This is the same as Game_0, except that we change Setup and Enc with the challenge message $(\mathbf{x}_1^*, \mathbf{x}_2^*)$ as follows:
  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|})$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}, \mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}, \mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Output
$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\end{aligned}$$
$$\mathsf{msk} = (\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).$$
  - $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^*, \mathbf{x}_2^*), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}, \mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output
$$\begin{aligned}\mathsf{ct} = (C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^*}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^*}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\mathbf{V} + (\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^* \| \mathbf{y}_1 \otimes \mathbf{s}_2)}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

- Game_2: This is the same as Game_1, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^*, \mathbf{x}_2^*)$ as follows:

  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^*, \mathbf{x}_2^*))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}, \mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}, \mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}, \mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^* \| \mathbf{y}_1 \otimes \mathbf{s}_2)\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]}));\end{aligned}$$

$$\mathsf{msk} = (\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).$$

  - $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^*, \mathbf{x}_2^*), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$.

$$\begin{aligned}\mathsf{ct} = (C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^*}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^*}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1 \cdot \eta_2]_T).\end{aligned}$$

- Game_3: This is the same as Game_2, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^*, \mathbf{x}_2^*)$ as follows:

  - $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^{(1)}, \mathbf{x}_2^*))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}, \mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}, \mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}, \mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^* \otimes \hat{\mathbf{x}}_2^*)\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]}));\end{aligned}$$

$$\mathsf{msk} = (\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).$$

- $\mathsf{Enc}(\mathsf{mpk}, (\mathbf{x}_1^*, \mathbf{x}_2^*), \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^*}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^*}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

– Game_4: This is the same as Game_3, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^*, \mathbf{x}_2^*)$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^*, \mathbf{x}_2^*))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$, and $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$. Output

$$\begin{aligned}\mathsf{crs} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^* \otimes \hat{\mathbf{x}}_2^*)\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\end{aligned}$$
$$\mathsf{msk} = (\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).$$

- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_2^*, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct} = (C_1 = [\mathbf{y}_1]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^*}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).\end{aligned}$$

– Game_5: This is the same as Game_4, except that we change Setup and Enc with the challenge messages $(\mathbf{x}_1^*, \mathbf{x}_2^*)$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^n, 1^L, 1^{|\mathcal{F}|}, (\mathbf{x}_1^*, \mathbf{x}_2^*))$: Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}$, $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{(k+k')(n+1)\times(k+k')(n+1)}$ for each $i \in [L]$. There exists an $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Sample $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, $\mathbf{y}_2 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$.

Output

$$
\begin{aligned}
\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\
&\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\
&\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes \mathbf{y}_2 - \hat{\mathbf{x}}_1^* \otimes \hat{\mathbf{x}}_2^*)\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\
\mathsf{msk} = (&\mathbf{a}^{\perp}, \mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).
\end{aligned}
$$

- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{msk}, \{\mathsf{sk}_i\}_{i\in[L]})$: Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$
\begin{aligned}
\mathsf{ct} = (&C_1 = [\mathbf{y}_1]_1, C_2 = [\mathbf{y}_2]_2, C_3 = [\mathbf{c}]_1,\\
&C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\
&C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^t - \eta_1\cdot\eta_2]_T).
\end{aligned}
$$

Lemma 17. *We have* $|\mathsf{Adv}_{\mathsf{Game}_0,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_1,\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{MDDH}_{\mathcal{B}_1,G_1}(\lambda)$, *where* $\mathcal{B}_1$ *is the adversary for the MDDH assumption in* G_1.

Proof. Suppose a challenger $\mathcal{B}_1$. The adversary $\mathcal{A}$ sends $(\mathbf{x}_1^*, \mathbf{x}_2^*)$ and the number of slots L to $\mathcal{B}_1$. $\mathcal{B}_1$ receives $([\mathbf{A}]_1, [T]_1)$ from the underlying MDDH assumption. For $\mathbf{A}$, there exist $\mathbf{a}^{\perp} \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. Then $\mathcal{B}_1$ interacts with $\mathcal{A}$ as follows:

$\mathcal{B}_1$ generates the output of Setup:

Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, and sample $\mathbf{V}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$, $\mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}$, $\mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. For each $\mathbf{f} \in \mathcal{F}$, set $\hat{\mathbf{f}}$ as equation (2). Output

$$
\begin{aligned}
\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1, \{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\
&\{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\
\mathsf{msk} = (&\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{V}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).
\end{aligned}
$$

$\mathcal{B}_1$ sends crs to $\mathcal{A}$.

$\mathcal{B}_1$ generates the output of KeyGen for honest users:

Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, which satisfies each component of $\mathbf{U}_i$ is bounded by B. Output

$$
\begin{aligned}
&\mathsf{pk}_{\mathsf{ctr}} = ([\mathbf{A}\mathbf{U}_i]_1, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{H}\mathbf{f}_j]_2\}_{t\neq i,j\in[|\mathcal{F}|]}, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{B}\mathbf{f}_j]_2\}_{t\neq i,j\in[|\mathcal{F}|]});\\
&\mathsf{sk}_{\mathsf{ctr}} = \mathbf{U}_i.
\end{aligned}
$$

Upon receiving $(\{\mathbf{f}^{i*}, \mathsf{pk}_i^*\}_{i\in[L]})$ from $\mathcal{A}$, for each $i \in [L]$, $\mathcal{B}_1$ generates the output of $\mathsf{IsValid}$:

Check the validity of pk_i^*. If $0 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_1$ aborts. Otherwise, $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_1$ computes $\mathsf{sk}_i = \mathbf{U}_i$ with msk and brute-force discrete log (If $\mathbf{U}_i$ cannot be computed, $\mathcal{B}_1$ aborts. And note that only with negligible probability, $[\mathbf{U}_i]_T$ is not unique, in this case, $\mathcal{B}_1$ also aborts.).

$\underline{\mathcal{B}_1 \text{ generates the output of } \mathsf{Aggr}\text{:}}$

Output

$$\mathsf{mpk} = ([\mathbf{A}\sum_{i\in[L]}\mathbf{U}_i]_1, [\sum_{i\in[L]}\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1,$$
$$[\mathbf{A}\mathbf{V}_2]_1);$$

$$\{\mathsf{hsk}_i = (i, \mathbf{f}^{i*}, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{i*}]_2,$$
$$[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2)\}_{i\in[L]}.$$

$\mathcal{B}_1$ sends $(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$ to $\mathcal{A}$.

$\underline{\mathcal{B}_1 \text{ generates the output of } \mathsf{Enc}\text{:}}$

Sample $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}, \mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, and $\eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\mathsf{ct}^* = (C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^*}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^*}_{\mathbf{y}_2}]_2, C_3 = [\boxed{T}]_1,$$
$$C_4 = [\underbrace{\boxed{T}\mathbf{V} + (\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^* \| \mathbf{y}_1 \otimes \mathbf{s}_2)}_{\mathbf{y}_0}]_1, C_5 = [\boxed{T}\mathbf{V}_1 + \boxed{T}\sum_{t\in[L]}\mathbf{U}_t]_1,$$
$$C_6 = [\boxed{T}\mathbf{V}_2]_1, C_7 = [-\boxed{T}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{t*} - \eta_1\cdot\eta_2]_T).$$

$\mathcal{B}_1$ sends ct^* to $\mathcal{A}$.

Observe when $T = \mathbf{s}\mathbf{A}$, where $\mathbf{s} \leftarrow_R \mathbb{Z}_p^{1\times k}$, the distributions are as Game_0; when $T = \mathbf{c}$, where $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$ satisfying $\mathbf{c}\mathbf{a}^{\perp\top} = 1$, the distributions are as Game_1. Then if $\mathcal{A}$ can distinguish Game_0 and Game_1, $\mathcal{B}_1$ can utilize $\mathcal{A}$ to break the MDDH assumption in G_1. Thus lead to contradiction.

Lemma 18. *We have* $|\mathsf{Adv}_{\mathsf{Game}_1,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_2,\mathcal{A}}(\lambda)| = 0.$

Proof. Sample $\widetilde{\mathbf{V}} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}$. Then change the variables $\mathbf{V}$ as follows:

$$\mathbf{V} = \widetilde{\mathbf{V}} - \mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \mathbf{x}_2^* \| \mathbf{y}_1 \otimes \mathbf{s}_2),$$

where the distributions are taken over the random choices of $\mathbf{V}$ and $\widetilde{\mathbf{V}}$.

We have $\mathbf{A}\mathbf{V} = \mathbf{A}(\widetilde{\mathbf{V}} - \mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \mathbf{x}_2^* \| \mathbf{y}_1 \otimes \mathbf{s}_2)) = \mathbf{A}\widetilde{\mathbf{V}}$. Then crs is changed into

$$\mathsf{crs} = (\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\boxed{\widetilde{\mathbf{V}}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,$$
$$\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]},$$
$$\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \boxed{\widetilde{\mathbf{V}}}\mathbf{H}\hat{\mathbf{f}}_j\boxed{-\mathbf{a}^{\perp\top}(\mathbf{s}_1 \otimes \hat{\mathbf{x}}_2^* \| \mathbf{y}_1 \otimes \mathbf{s}_2)}\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L], j\in[|\mathcal{F}|]}));$$

ct^* is changed into

$$\begin{aligned}\mathsf{ct}^* = (&C_1 = [\underbrace{\mathbf{s}_1\mathbf{A}_1 + \hat{\mathbf{x}}_1^*}_{\mathbf{y}_1}]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^*}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ &C_4 = [\underbrace{\mathbf{c}\boxed{\widetilde{\mathbf{V}}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ &C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{t*} - \eta_1\cdot\eta_2]_T).\end{aligned}$$

Therefore, Game_1 and Game_2 are identically distributed.

Lemma 19. *We have* $|\mathsf{Adv}_{\mathsf{Game}_2,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_3,\mathcal{A}}(\lambda)| = 0$.

Proof. It is easy to see

$$\begin{aligned}&(\mathbf{s}_1\otimes\hat{\mathbf{x}}_2^*\|\mathbf{y}_1\otimes\mathbf{s}_2)\mathbf{H}\hat{\mathbf{f}}_j\\ =&(\mathbf{s}_1\otimes\hat{\mathbf{x}}_2^*)(\mathbf{A}_1\otimes\mathbf{I}_{n+1}) + (\mathbf{y}_1\otimes\mathbf{s}_2)(\mathbf{I}_{n+1}\otimes\mathbf{A}_2)\\ =&(\mathbf{s}_1\mathbf{A}_1)\otimes\hat{\mathbf{x}}_2^* + (\mathbf{s}_1\mathbf{A}_1)\otimes(\mathbf{s}_2\mathbf{A}_2) + \hat{\mathbf{x}}_1^*\otimes(\mathbf{s}_2\mathbf{A}_2)\\ =&(\mathbf{y}_1\otimes\mathbf{y}_2 - \hat{\mathbf{x}}_1^*\otimes\hat{\mathbf{x}}_2^*)\mathbf{f}_j.\end{aligned}$$

Therefore, Game_2 and Game_3 are identically distributed.

Lemma 20. *We have* $|\mathsf{Adv}_{\mathsf{Game}_3,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_4,\mathcal{A}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_2}^{\textit{bi-MDDH}}(\lambda)$*, where* $\mathcal{B}_2$ *is the adversary for the bi-MDDH assumption.*

Proof. Suppose a challenger $\mathcal{B}_2$. The adversary $\mathcal{A}$ sends $(\mathbf{x}_1^*, \mathbf{x}_2^*)$ and the number of slots L to $\mathcal{B}_2$. $\mathcal{B}_2$ receives $([\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [T]_1, [T]_2)$ from the underlying bi-MDDH assumption. Then $\mathcal{B}_2$ interacts with $\mathcal{A}$ as follows:

$\mathcal{B}_2$ generates the output of Setup:

Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, and sample $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}, \mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \mathbf{A}_2 \leftarrow_R \mathbb{Z}_p^{k\times(n+1)}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. For $\mathbf{A}$, there exist $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. For each $\mathbf{f} \in \mathcal{F}$, set $\hat{\mathbf{f}}$ as equation (2). Sample $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}, \eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}((\boxed{T} + \hat{\mathbf{x}}_1^*)\otimes\mathbf{y}_2 - \hat{\mathbf{x}}_1^*\otimes\hat{\mathbf{x}}_2^*)\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_2, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

$\mathcal{B}_2$ sends crs to $\mathcal{A}$.

$\mathcal{B}_2$ generates the output of KeyGen for honest users:

Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, which satisfies each component of $\mathbf{U}_i$ is bounded by B. Output

$$\begin{aligned}&\mathsf{pk}_{\mathsf{ctr}} = ([\mathbf{A}\mathbf{U}_i]_1, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{H}\mathbf{f}_j]_2\}_{t\neq i, j\in[|\mathcal{F}|]}, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{B}\mathbf{f}_j]_2\}_{t\neq i, j\in[|\mathcal{F}|]});\\ &\mathsf{sk}_{\mathsf{ctr}} = \mathbf{U}_i.\end{aligned}$$

Upon receiving $(\{\mathbf{f}^{i*}, \mathsf{pk}_i^*\}_{i\in[L]})$ from $\mathcal{A}$, for each $i \in [L]$, $\mathcal{B}_2$ generates the output of $\mathsf{IsValid}$:

Check the validity of pk_i^*. If $0 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_2$ aborts. Otherwise, $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_2$ computes $\mathsf{sk}_i = \mathbf{U}_i$ with msk and brute-force discrete log (If $\mathbf{U}_i$ cannot be computed, $\mathcal{B}_2$ aborts. And note that only with negligible probability, $[\mathbf{U}_i]_T$ is not unique, in this case, $\mathcal{B}_2$ also aborts.).

$\mathcal{B}_2$ generates the output of Aggr:

Output

$$\begin{aligned}\mathsf{mpk} = ([\mathbf{A}\sum_{i\in[L]}\mathbf{U}_i]_1, [\sum_{i\in[L]}\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1,\\ [\mathbf{A}\mathbf{V}_2]_1);\\ \{\mathsf{hsk}_i = (i, \mathbf{f}^{i*}, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\sum_{t\neq i, t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\sum_{t\neq i, t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{i*}]_2,\\ [\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2)\}_{i\in[L]}.\end{aligned}$$

$\mathcal{B}_2$ sends $(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$ to $\mathcal{A}$.

$\mathcal{B}_2$ generates the output of Enc:

Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct}^* = (C_1 = [\boxed{T} + \hat{\mathbf{x}}_1^*]_1, C_2 = [\underbrace{\mathbf{s}_2\mathbf{A}_2 + \hat{\mathbf{x}}_2^*}_{\mathbf{y}_2}]_2, C_3 = [\mathbf{c}]_1,\\ C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]}\mathbf{U}_t]_1,\\ C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2\sum_{t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{t*} - \eta_1\cdot\eta_2]_T).\end{aligned}$$

$\mathcal{B}_2$ sends ct^* to $\mathcal{A}$.

Observe when $T = \mathbf{s}_1\mathbf{A}_1$, where $\mathbf{s}_1 \leftarrow_R \mathbb{Z}_p^{1\times k'}$, the distributions are as Game_3; when $T = \mathbf{y}_1'$, where $\mathbf{y}_1' \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$ satisfying $\mathbf{y}_1' + \hat{\mathbf{x}}_1^{(*)} = \mathbf{y}_1$ and $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, the distributions are as Game_4. Then if $\mathcal{A}$ can distinguish Game_3 and Game_4, $\mathcal{B}_2$ can utilize $\mathcal{A}$ to break the bi-MDDH assumption. Thus lead to contradiction.

Lemma 21. *We have* $|\mathsf{Adv}_{\mathsf{Game}_4,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{Game}_5,\mathcal{A}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_3,G_2}^{MDDH}(\lambda)$, *where* $\mathcal{B}_3$ *is the adversary for the MDDH assumption in* G_2.

Proof. Suppose a challenger $\mathcal{B}_3$. The adversary $\mathcal{A}$ sends $(\mathbf{x}_1^*, \mathbf{x}_2^*)$ and the number of slots L to $\mathcal{B}_3$. $\mathcal{B}_3$ receives $([\mathbf{A}_2]_2, [T]_2)$ from the underlying MDDH assumption in G_2. Then $\mathcal{B}_3$ interacts with $\mathcal{A}$ as follows:

$\mathcal{B}_3$ generates the output of Setup:

Run the group generator $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, and sample $\widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+k')(n+1)}, \mathbf{A} \leftarrow_R \mathbb{Z}_p^{k\times(k+1)}, \mathbf{A}_1 \leftarrow_R \mathbb{Z}_p^{k'\times(n+1)}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+k')\times(n+1)^2}$, and $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{n\times n}$ for each $i \in [L]$. For $\mathbf{A}$, there exist $\mathbf{a}^\perp \in \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{A}\mathbf{a}^{\perp\top} = \mathbf{0} \in \mathbb{Z}_p^{k\times 1}$. For each $\mathbf{f} \in \mathcal{F}$, set $\hat{\mathbf{f}}$ as equation (2). Sample $\mathbf{y}_1 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}, \eta_1, \eta_2 \leftarrow_R \mathbb{Z}_p$. Output

$$\begin{aligned}\mathsf{crs} = (&\mathbb{G}, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\widetilde{\mathbf{V}}]_1, [\mathbf{A}\mathbf{V}_1]_1, [\mathbf{A}\mathbf{V}_2]_1,\\ &\{[\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}, \{[\mathbf{W}_i\mathbf{B}\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]},\\ &\{[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}_j + \widetilde{\mathbf{V}}\mathbf{H}\hat{\mathbf{f}}_j - \mathbf{a}^{\perp\top}(\mathbf{y}_1 \otimes (\boxed{T} + \hat{\mathbf{x}}_2^*) - \hat{\mathbf{x}}_1^* \otimes \hat{\mathbf{x}}_2^*)\hat{\mathbf{f}}_j]_2\}_{i\in[L],j\in[|\mathcal{F}|]}));\\ \mathsf{msk} = (&\mathbf{a}^\perp, \mathbf{A}, \mathbf{A}_1, \widetilde{\mathbf{V}}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{B}, \{\mathbf{W}_i\}_{i\in[L]}).\end{aligned}$$

$\mathcal{B}_3$ sends crs to $\mathcal{A}$.

$\mathcal{B}_3$ generates the output of KeyGen for honest users:

Sample $\mathbf{U}_i \leftarrow_R \mathbb{Z}_p^{(k+1)\times n}$, which satisfies each component of $\mathbf{U}_i$ is bounded by B. Output

$$\begin{aligned}&\mathsf{pk}_{\mathsf{ctr}} = ([\mathbf{A}\mathbf{U}_i]_1, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{H}\mathbf{f}_j]_2\}_{t\neq i,j\in[|\mathcal{F}|]}, \{[\mathbf{U}_i\mathbf{W}_t\mathbf{B}\mathbf{f}_j]_2\}_{t\neq i,j\in[|\mathcal{F}|]});\\ &\mathsf{sk}_{\mathsf{ctr}} = \mathbf{U}_i.\end{aligned}$$

Upon receiving $(\{\mathbf{f}^{i*}, \mathsf{pk}_i^*\}_{i\in[L]})$ from $\mathcal{A}$, for each $i \in [L]$, $\mathcal{B}_3$ generates the output of $\mathsf{IsValid}$:

Check the validity of pk_i^*. If $0 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_3$ aborts. Otherwise, $1 \leftarrow \mathsf{IsValid}(\mathsf{crs}, i, \mathsf{pk}_i^*)$, $\mathcal{B}_3$ computes $\mathsf{sk}_i = \mathbf{U}_i$ with msk and brute-force discrete log (If $\mathbf{U}_i$ cannot be computed, $\mathcal{B}_3$ aborts. And note that only with negligible probability, $[\mathbf{U}_i]_T$ is not unique, in this case, $\mathcal{B}_3$ also aborts.).

$\mathcal{B}_3$ generates the output of Aggr:

Output

$$\begin{aligned}\mathsf{mpk} = (&[\mathbf{A}\sum_{i\in[L]}\mathbf{U}_i]_1, [\sum_{i\in[L]}\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{A}]_1, [\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\mathbf{A}\mathbf{V}]_1, [\mathbf{A}\mathbf{V}_1]_1,\\ &[\mathbf{A}\mathbf{V}_2]_1);\\ \{\mathsf{hsk}_i = (&i, \mathbf{f}^{i*}, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, [\sum_{t\neq i,t\in[L]}\mathbf{U}_t\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\sum_{t\neq i,t\in[L]}\mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{i*}]_2,\\ &[\mathbf{V}_1\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}_2\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*} + \mathbf{V}\mathbf{H}\hat{\mathbf{f}}^{i*}]_2, [\mathbf{W}_i\mathbf{H}\hat{\mathbf{f}}^{i*}]_2)\}_{i\in[L]}.\end{aligned}$$

$\mathcal{B}_3$ sends $(\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$ to $\mathcal{A}$.

$\mathcal{B}_3$ generates the output of Enc:

Sample $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{1\times(k+1)}$, which satisfies $\mathbf{c}\mathbf{a}^{\perp\top} = 1$. Output

$$\begin{aligned}\mathsf{ct}^* = (&C_1 = [\mathbf{y}_1]_1, C_2 = [\boxed{T} + \hat{\mathbf{x}}_2^*]_2, C_3 = [\mathbf{c}]_1,\\ &C_4 = [\underbrace{\mathbf{c}\widetilde{\mathbf{V}}}_{\mathbf{y}_0}]_1, C_5 = [\mathbf{c}\mathbf{V}_1 + \mathbf{c}\sum_{t\in[L]} \mathbf{U}_t]_1,\\ &C_6 = [\mathbf{c}\mathbf{V}_2]_1, C_7 = [-\mathbf{c}\mathbf{V}_2 \sum_{t\in[L]} \mathbf{W}_t\mathbf{H}\hat{\mathbf{f}}^{t*} - \eta_1 \cdot \eta_2]_T).\end{aligned}$$

$\mathcal{B}_3$ sends ct^* to $\mathcal{A}$.

Observe when $T = \mathbf{s}_2\mathbf{A}_2$, where $\mathbf{s}_2 \leftarrow_R \mathbb{Z}_p^{1\times k}$, the distributions are as Game_4; when $T = \mathbf{y}_2'$, where $\mathbf{y}_2' \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$ satisfying $\mathbf{y}_2' + \hat{\mathbf{x}}_2^{(*)} = \mathbf{y}_2$ and $\mathbf{y}_2 \leftarrow_R \mathbb{Z}_p^{1\times(n+1)}$, the distributions are as Game_5. Then if $\mathcal{A}$ can distinguish Game_4 and Game_5, $\mathcal{B}_3$ can utilize $\mathcal{A}$ to break the MDDH assumption in G_2. Thus lead to contradiction.

Note that $(\hat{\mathbf{x}}_1^* \otimes \hat{\mathbf{x}}_2^*)\hat{\mathbf{f}}_j$ is the function value.

6 From Slotted RFE to RFE

The transformation from slotted RFE to RFE follows the generic compiler in [28,18]. Here, we omit the concrete constructions of our RFE, the readers can refer [28,18] for details.

Theorem 11 (Perfect Correctness of Construction Π_{RIPFE}). *If the slotted RIPFE Π_{sRIPFE} in Section 4.1 is complete and perfectly correct, then our RIPFE Π_{RIPFE}, which follows the generic compiler in [28,18], is perfectly correct.*

Proof. This follows an identical argument of [28,18]. We omit it here.

Theorem 12 (Compactness of Construction Π_{RIPFE}). *If the slotted RIPFE Π_{sRIPFE} in Section 4.1 is compact, then our RIPFE Π_{RIPFE}, which follows the generic compiler in [28,18], is compact.*

Proof. Since in RIPFE Π_{RIPFE}, there are $(\log L+1)$ instances of slotted RIPFE Π_{sRIPFE}. For each instance of slotted RIPFE Π_{sRIPFE}, the size of master public key is bounded by $n\cdot\mathsf{poly}(\lambda)$. Thus, for RIPFE Π_{RIPFE}, the size of master public key is bounded by $\log L\cdot n\cdot\mathsf{poly}(\lambda)$. Similarly, for each instance of slotted RIPFE Π_{sRIPFE}, the size of helper decryption key is bounded by $n\cdot\mathsf{poly}(\lambda)+O(\log L)$. Thus, for RIPFE Π_{RIPFE}, the size of helper decryption key is bounded by $\log L\cdot n\cdot\mathsf{poly}(\lambda)+\log L\cdot O(\log L)$.

Theorem 13 (Update Efficiency of Construction Π_{RIPFE}). *If the slotted RIPFE Π_{sRIPFE} in Section 4.1 is compact, then our RIPFE Π_{RIPFE}, which follows the generic compiler in [28,18], satisfies update efficiency.*

Proof. By construction of $\mathit{\Pi}_{RIPFE}$, the number of updates is at most $(\log L+1)$. And since each helper decryption key is of size $\log L \cdot n \cdot \mathsf{poly}(\lambda) + \log L \cdot O(\log L)$ (which follows the compactness of slotted RIPFE $\mathit{\Pi}_{sRIPFE}$), thus the update operation can be implemented in $\log L \cdot n \cdot \mathsf{poly}(\lambda) + \log L \cdot O(\log L)$ time in the RAM model of computation. Therefore, our RIPFE $\mathit{\Pi}_{RIPFE}$ satisfies update efficiency.

Theorem 14 (Security of Construction $\mathit{\Pi}_{RIPFE}$). *If the slotted RIPFE $\mathit{\Pi}_{sRIPFE}$ in Section 4.1 is weakly selective-IND secure (resp. weakly selective-SIM secure), then our RIPFE $\mathit{\Pi}_{RIPFE}$, which follows the generic compiler in [28,18], is weakly selective-IND secure (resp. weakly selective-SIM secure).*

Remark 3. When the functions of the users that are corrupted by the adversary cover the entire function space, for $\mathit{\Pi}_{RIPFE}$, weakly selective-SIM security is equivalent to selective-SIM security.

Proof. This follows an identical argument of [28,18]. A notable difference is that we require the efficient algorithm $\mathcal{B}$ also sends the challenge messages chosen by $\mathcal{A}$ to the underlying challenger, together with the number of slots, 2^{k^*}, of the k^*-th slotted RIPFE, in the setup phase. We omit the proof here.

Theorem 15 (Perfect Correctness of Construction $\mathit{\Pi}_{RQFE}$). *If the slotted RQFE $\mathit{\Pi}_{sRQFE}$ in Section 5.1 is complete and perfectly correct, then our RQFE $\mathit{\Pi}_{RQFE}$, which follows the generic compiler in [28,18], is perfectly correct.*

Proof. This follows an identical argument of [28,18]. We omit it here.

Theorem 16 (Compactness of Construction $\mathit{\Pi}_{RQFE}$). *If the slotted RQFE $\mathit{\Pi}_{sRQFE}$ in Section 5.1 is compact, then our RQFE $\mathit{\Pi}_{RQFE}$, which follows the generic compiler in [28,18], is compact.*

Proof. Since in RQFE $\mathit{\Pi}_{RQFE}$, there are $(\log L + 1)$ instances of slotted RQFE $\mathit{\Pi}_{sRQFE}$. For each instance of slotted RQFE $\mathit{\Pi}_{sRQFE}$, the size of master public key is bounded by $(n+1) \cdot \mathsf{poly}(\lambda)$. Thus, for RQFE $\mathit{\Pi}_{RQFE}$, the size of master public key is bounded by $\log L \cdot (n+1) \cdot \mathsf{poly}(\lambda)$. Similarly, for each instance of slotted RQFE $\mathit{\Pi}_{sRQFE}$, the size of helper decryption key is bounded by $(n+1) \cdot \mathsf{poly}(\lambda) + n^2 \cdot \mathsf{poly}(\lambda) + O(\log L)$. Thus, for RQFE $\mathit{\Pi}_{RQFE}$, the size of helper decryption key is bounded by $\log L \cdot (n+1) \cdot \mathsf{poly}(\lambda) + \log L \cdot n^2 \cdot \mathsf{poly}(\lambda) + \log L \cdot O(\log L)$.

Theorem 17 (Update Efficiency of Construction $\mathit{\Pi}_{RQFE}$). *If the slotted RQFE $\mathit{\Pi}_{sRQFE}$ in Section 5.1 is compact, then our RQFE $\mathit{\Pi}_{RQFE}$, which follows the generic compiler in [28,18], satisfies update efficiency.*

Proof. By construction of $\mathit{\Pi}_{RQFE}$, the number of updates is at most $(\log L + 1)$. And since each helper decryption key is of size $\log L \cdot (n+1) \cdot \mathsf{poly}(\lambda) + \log L \cdot n^2 \cdot \mathsf{poly}(\lambda) + \log L \cdot O(\log L)$ (which follows the compactness of slotted RQFE $\mathit{\Pi}_{sRQFE}$), thus the update operation can be implemented in $\log L \cdot (n+1) \cdot \mathsf{poly}(\lambda) + \log L \cdot n^2 \cdot \mathsf{poly}(\lambda) + \log L \cdot O(\log L)$ time in the RAM model of computation. Therefore, our RQFE $\mathit{\Pi}_{RQFE}$ satisfies update efficiency.

Theorem 18 (Security of Construction Π_{RQFE}). *If the slotted RQFE Π_{sRQFE} in Section 5.1 is weakly selective-IND secure (resp. weakly selective-SIM secure), then our RQFE Π_{RQFE}, which follows the generic compiler in [28,18], is weakly selective-IND secure (resp. weakly selective-SIM secure).*

Remark 4. When the functions of the users that are corrupted by the adversary cover the entire function space, for Π_{RQFE}, weakly selective-SIM security is equivalent to selective-SIM security.

Proof. This follows an identical argument of [28,18]. A notable difference is that we require the efficient algorithm $\mathcal{B}$ also sends the challenge messages chosen by $\mathcal{A}$ to the underlying challenger, together with the number of slots, 2^{k^*}, of the k^*-th slotted RQFE, in the setup phase. We omit the proof here.

References

1. Abdalla, M., Benhamouda, F., Gay, R.: From single-input to multi-client inner-product functional encryption. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 552–582. Springer (2019)
2. Abdalla, M., Benhamouda, F., Kohlweiss, M., Waldner, H.: Decentralizing inner-product functional encryption. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 128–157. Springer (2019)
3. Abdalla, M., Bourse, F., Caro, A.D., Pointcheval, D.: Simple functional encryption schemes for inner products. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 733–751. Springer (2015)
4. Agrawal, S., Libert, B., Maitra, M., Titiu, R.: Adaptive simulation security for inner product functional encryption. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part I. LNCS, vol. 12110, pp. 34–64. Springer (2020)
5. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 333–362. Springer (2016)
6. Agrawal, S., Rosen, A.: Functional encryption for bounded collusions, revisited. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017,Part I. LNCS, vol. 10677, pp. 173–205. Springer (2017)
7. Ananth, P., Lombardi, A.: Succinct garbling schemes from functional encryption through a local simulation paradigm. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018,Part II. LNCS, vol. 11240, pp. 455–472. Springer (2018)
8. Ananth, P.V., Sahai, A.: Functional encryption for turing machines. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part I. LNCS, vol. 9562, pp. 125–153. Springer (2016)
9. Baltico, C.E.Z., Catalano, D., Fiore, D., Gay, R.: Practical functional encryption for quadratic functions with applications to predicate encryption. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 67–98. Springer (2017)
10. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer (2011)
11. Boyle, E., Chung, K., Pass, R.: On extractability obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 52–73. Springer (2014)

12. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 595–624. Springer (2015)
13. Chen, J., Gong, J., Kowalczyk, L., Wee, H.: Unbounded ABE via bilinear entropy expansion, revisited. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 503–534. Springer (2018)
14. Chotard, J., Dufour-Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Dynamic decentralized functional encryption. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 747–775. Springer (2020)
15. Chotard, J., Sans, E.D., Gay, R., Phan, D.H., Pointcheval, D.: Decentralized multi-client functional encryption for inner product. In: Peyrin, T., Galbraith, S.D. (eds.) ASIACRYPT 2018, Part II. LNCS, vol. 11273, pp. 703–732. Springer (2018)
16. Cong, K., Eldefrawy, K., Smart, N.P.: Optimizing registration based encryption. In: Paterson, M.B. (ed.) IMACC 2021. LNCS, vol. 13129, pp. 129–157. Springer (2021)
17. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An algebraic framework for diffie-hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer (2013)
18. Francati, D., Friolo, D., Maitra, M., Malavolta, G., Rahimi, A., Venturi, D.: Registered (inner-product) functional encryption. IACR Cryptol. ePrint Arch. p. 395 (2023)
19. Freitag, C., Waters, B., Wu, D.J.: How to use (plain) witness encryption: Registered abe, flexible broadcast, and more. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part IV. LNCS, vol. 14084, pp. 498–531. Springer (2023)
20. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: FOCS 2013. pp. 40–49. IEEE Computer Society (2013)
21. Garg, S., Hajiabadi, M., Mahmoody, M., Rahimi, A.: Registration-based encryption: Removing private-key generator from IBE. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018,Part I. LNCS, vol. 11239, pp. 689–718. Springer (2018)
22. Garg, S., Hajiabadi, M., Mahmoody, M., Rahimi, A., Sekar, S.: Registration-based encryption from standard assumptions. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 63–93. Springer (2019)
23. Gay, R.: A new paradigm for public-key functional encryption for degree-2 polynomials. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part I. LNCS, vol. 12110, pp. 95–120. Springer (2020)
24. Glaeser, N., Kolonelos, D., Malavolta, G., Rahimi, A.: Efficient registration-based encryption. IACR Cryptol. ePrint Arch. p. 1505 (2022), `https://eprint.iacr.org/2022/1505`
25. Gong, J., Qian, H.: Simple and efficient FE for quadratic functions. Des. Codes Cryptogr. **89**(8), 1757–1786 (2021)
26. Goyal, R., Vusirikala, S.: Verifiable registration-based encryption. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 621–651. Springer (2020)
27. Guan, J., Wichs, D., Zhandry, M.: Incompressible cryptography. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part I. LNCS, vol. 13275, pp. 700–730. Springer (2022)
28. Hohenberger, S., Lu, G., Waters, B., Wu, D.J.: Registered attribute-based encryption. IACR Cryptol. ePrint Arch. p. 1500 (2022)

29. Hohenberger, S., Lu, G., Waters, B., Wu, D.J.: Registered attribute-based encryption. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part III. LNCS, vol. 14006, pp. 511–542. Springer (2023). https://doi.org/10.1007/978-3-031-30620-4_17
30. Kitagawa, F., Nishimaki, R., Tanaka, K., Yamakawa, T.: Adaptively secure and succinct functional encryption: Improving security and efficiency, simultaneously. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 521–551. Springer (2019)
31. Kolonelos, D., Malavolta, G., Wee, H.: Distributed broadcast encryption from bilinear groups. IACR Cryptol. ePrint Arch. p. 874 (2023), https://eprint.iacr.org/2023/874
32. Libert, B., Titiu, R.: Multi-client functional encryption for linear functions in the standard model from LWE. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 520–551. Springer (2019)
33. O'Neill, A.: Definitional issues in functional encryption. IACR Cryptol. ePrint Arch. p. 556 (2010), http://eprint.iacr.org/2010/556
34. Waters, B.: A punctured programming approach to adaptively secure functional encryption. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 678–697. Springer (2015)
35. Wee, H.: Attribute-hiding predicate encryption in bilinear groups, revisited. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10677, pp. 206–233. Springer (2017). https://doi.org/10.1007/978-3-319-70500-2_8, https://doi.org/10.1007/978-3-319-70500-2_8
36. Wee, H.: Functional encryption for quadratic functions from k-lin, revisited. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 210–228. Springer (2020)