SACfe: Secure Access Control in Functional Encryption with Unbounded Data

Uddipana Dowerah
University of St. Gallen
St. Gallen, Switzerland
uddipana.dowerah@unisg.ch

Subhranil Dutta
Indian Institute of Technology Kharagpur
Kharagpur, India
subhranildutta@iitkgp.ac.in

Frank Hartmann
University of St. Gallen
St. Gallen, Switzerland
frank.hartmann@unisg.ch

Aikaterini Mitrokotsa
University of St. Gallen
St. Gallen, Switzerland
katerina.mitrokotsa@unisg.ch

Sayantan Mukherjee
Indian Institute of Technology Jammu
Jammu, India
csayantan.mukherjee@gmail.com

Tapas Pal
Karlsruhe Institute of Technology,
Karlsruhe, Germany
tapas.pal@kit.edu

***Abstract*—Privacy is a major concern in large-scale digital applications, such as cloud-computing, machine learning services, and access control. Users want to protect not only their plain data but also their associated attributes (*e.g.,* age, location, etc). Functional encryption (FE) is a cryptographic tool that allows fine-grained access control over encrypted data. However, existing FE fall short as they are either inefficient and far from reality or they leak sensitive user-specific information.**

We propose SACfe, a novel attribute-based FE scheme that provides *secure*, fine-grained access control and hides both the user's attributes and the function applied to the data, while preserving the data's confidentiality. Moreover, it enables users to encrypt unbounded-length messages along with an arbitrary number of hidden attributes into ciphertexts. We design SACfe, a protocol for performing linear computation on encrypted data while enforcing access control based on inner product predicates. We show how SACfe can be used for online biometric authentication for privacy-preserving access control. As an additional contribution, we introduce an attribute-based linear FE for unbounded length of messages and functions where access control is realized by monotone span programs. We implement our protocols using the CiFEr cryptographic library and show its efficiency for practical settings.

1. Introduction

Functional Encryption (FE) is a cryptographic primitive that allows fine-grained access control over encrypted data. Unlike the traditional *all-or-nothing* encryption, FE allows to recover specific functions of the input messages with secret keys associated to these functions. Specifically, an FE scheme that allows to evaluate a set of functions $\mathcal{F}$ on its input messages, issues secret keys SK_f corresponding to specific functions $f \in \mathcal{F}$. If CT_m denotes a ciphertext that encrypts a message m, then decrypting CT_m with a secret key SK_f reveals $f(m)$. The security of an FE scheme ensures that no information apart from $f(m)$ is revealed from $\mathsf{SK}_f, \mathsf{CT}_m$ about the message m.

Functional Encryption for Inner Products. Although there have been significant efforts [17], [21] on constructing FE schemes for general functionalities, they rely on complex tools, assumptions, and hence, are far from reality. As a result, there is more focus on constructing FE schemes for specific functionalities like linear or quadratic functions [3], [7], [8], [38]. In this paper, we focus on linear functions, more specifically *inner product functional encryption* (IPFE). In an IPFE scheme, the ciphertext $\mathsf{CT}_{\mathbf{x}}$ and the secret key $\mathsf{SK}_{\mathbf{y}}$ are computed for message and key vectors $\mathbf{x}$, $\mathbf{y}$ respectively and decryption of $\mathsf{CT}_{\mathbf{x}}$ with $\mathsf{SK}_{\mathbf{y}}$ recovers the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$. The linear functionality, although simple, already captures a wide range of applications in the domain of cloud computing [3], [33], [36], machine learning [11], [27], [39] and federated learning [11], [39], [40]. However, the main limitation of an IPFE system is that each secret key inherently leaks new information about the plain message. Specifically, if an IPFE scheme encrypts messages of length n then releasing n secret keys corresponding to vectors forming a basis of the message space enables complete recovery of the plaintext.

Access Control over IPFE. To address the inherent leakage of IPFE, Abdalla et al. [4] proposed a novel approach that embeds access policies in the secret keys and user attributes in the ciphertext while also facilitating the computation of weighted sums on

This work was done while Tapas Pal was a postdoc at NTT Social Informatics Laboratories, Japan.

the data. They named the primitive *attribute-based IPFE* or AB-IPFE which provides access control by *attribute-based encryption* (ABE) [19] and performs linear computation on the encrypted data akin to IPFE. More formally, a secret key is now additionally associated with an access policy P and an input message is encrypted under user-specific attributes att. The recovery of $\langle \mathbf{x}, \mathbf{y} \rangle$ now depends on whether the user's attributes att satisfy the policy P, i.e., when $P(\mathsf{att}) = 1$. For example if the list of attributes are $\mathsf{att} = \{\mathsf{age}, \mathsf{location}, \mathsf{smoking}\}$, an example of policy P could be '$(\mathsf{age} > 18$ AND $\mathsf{location} = \mathsf{Europe}$ OR $\mathsf{Smoking} = \mathsf{yes})$'.

Previous works [4], [25], [32] have presented various constructions of AB-IPFE, however these are solely focused on hiding only the message vectors. In contrast, several ABE and AB-IPFE schemes are built [10], [15], [20], [23], [31] under the name of predicate encryption or predicate based IPFE with a focus on *hiding attributes* (or predicates) associated with ciphertexts since attributes may reveal sensitive information such as user's identity, credit card information, health-related contents. On the other hand, *hiding the function* f or the weight vector $\mathbf{y}$ in an IPFE system turns out to be crucial in several applications related to biometric authentication, nearest neighbour search on encrypted data and privacy-preserving machine learning [9], [12], [24], [33]. In the context of privacy preserving biometric authentication, f corresponds to the reference biometric template of the client that is collected at the time of registration and stored in a remote server. At the time of authentication, a live biometric template of the client is collected and compared with the reference template. This is usually done by a third party service provider and giving the biometric templates in clear could lead to identity theft. Therefore, we need to hide both the reference and the live templates of the client from the server. This is achieved by using a function-hiding IPFE scheme.

Furthermore, in almost all applications of IPFE, secret keys $\mathsf{SK}_{\mathbf{y}}$ are sent to a public server to enable the decryption of ciphertexts by the cloud and, in fact, this has been one of the main motivations for using IPFE in place of a more complex and less efficient tool called Fully Homomorphic Encryption (FHE) [18], particularly in tasks such as training machine learning models [27], [33], [39] or privacy-preserving biometric authentication [24]. A crucial difference between FHE and FE lies in their decryption process: FHE requires the (master) secret key for decryption, whereas FE can generate decryption keys tailored to reveal specific plaintext information. In privacy-preserving biometric authentication, the use of FHE presents challenges because the server requires the secret key to decrypt the authentication decision. This is because the result of the evaluation of a function on encrypted data is also encrypted in FHE. On the contrary, functional encryption, particularly IPFE, can send a functional decryption key to the server, allowing it to learn computation results without needing the master secret key for decryption. However, there's a concern that any information encoded into the weight vector $\mathbf{y}$, such as the activation function of neural networks [33] or a live biometric template of users [24], may be revealed to the server. Therefore, it is crucial to preserve the privacy of user-specific information along with plain data.

Unfortunately, existing AB-IPFEs which could provide access control or better security than IPFE cannot be used in the above-mentioned applications since they are not designed to simultaneously hide user-specific attributes and functions *i.e.,* the weight vectors $\mathbf{y}$. This motivates the following question. *Can we ensure secure access control by protecting user-specific information (*i.e., *attributes and functions) in FE?*

Our Contributions. We present SACfe a protocol that guarantees secure access control by *hiding* (a) user-specific attributes associated with the ciphertext and (b) function embedded into the secret key of an FE system. In particular, when a server receives ciphertext $\mathsf{CT}_{\mathbf{x},\mathsf{att}}$ and secret key $\mathsf{SK}_{\mathbf{y},P}$ (see Figure 1 where EK denotes the encryption key) the maximum information it can extract about the message $\mathbf{x}$, function $\mathbf{y}$ and attributes att is the inner product value $\langle \mathbf{x}, \mathbf{y} \rangle$ and $P(\mathsf{att})$. More precisely, when employing SACfe in the context of privacy-preserving attribute-based biometric authentication, the message $\mathbf{x}$ denotes the fresh biometric template, the attributes att could include $\mathsf{att} = \{\text{age, location, paid subscription}\}$, while the reference template is associated with the vector $\mathbf{y}$. SACfe allows us to compute the distance between the fresh and stored template *i.e.,* $\langle \mathbf{x}, \mathbf{y} \rangle$, while hiding $\mathbf{x}$, $\mathbf{y}$, the attributes of the users as long as the policy P is satisfied *i.e.,* $P(\mathsf{att}) = 1$. This is done without setting any limits on the length of $\mathbf{x}$, $\mathbf{y}$ and the number of associated attributes.

More precisely, in this paper, we construct SACfe a protocol for inner product policies. We represent the predicate P and attributes att as vectors $\mathbf{v}$ and $\mathbf{w}$ respectively and the attributes are said to satisfy the predicate if $\langle \mathbf{w}, \mathbf{v} \rangle = 0$. It is called the zero-predicate. We note that such an inner product predicate delivers a wide variety of access control corresponding to equality tests, disjunctions or conjunctions of equality tests, polynomials, CNF/DNF formulae, or threshold predicates [23], [26].

Additionally, the scheme supports *unbounded* (*i.e.,* unlimited) length vectors for accommodating more data representing predicates, attributes, functions and messages. By *unbounded vectors* we mean that the sizes of those vectors are not fixed during the system setup and the lengths of the vectors are known only when they appear at the time of encryption or key generation. The unboundedness property yields

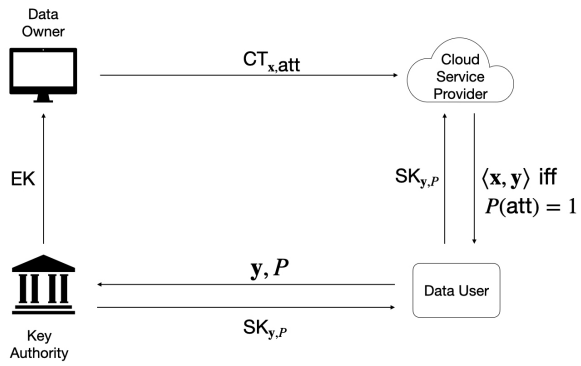

Figure 1. The general framework of SACfe

a more efficient system delivering constant size public keys, input-specific (both message and attribute vectors) ciphertext size and function-specific (both function and predicate vectors) secret key size. It not only enables a user to encrypt an arbitrary length message vector but also a variable length attributes can be associated with the ciphertext. We achieve full-hiding indistinguishability based security with semi-adaptive attributes under the standard and well-known *Symmetric eXternel Diffie-Hellman* (SXDH) assumption.

As an additional contribution, we present an unbounded AB-IPFE scheme supporting more expressive access policies realized by monotone span programs [22]. The secret key size remains constant with respect to the embedded function size. However, we emphasize that in our second contribution *i.e.,* the unbounded AB-IPFE scheme it only hides the message vectors, but *not* the attributes and functions. Finally, we have implemented both our schemes in C using the CiFEr cryptographic library [27] and analyse their performance in different conditions in Section 6.

Applications: Secure & privacy-preserving Access Control in Biometric Authentication. Let us consider the scenario of a service provider that may aim to replace password-based authentication with a biometric authentication (BA) system to eliminate password sharing and enforce access control based on a user's attributes. This could include services such as purchasing online products (*e.g.,* alcohol), access streaming services (*e.g.* Netflix content), access newspaper articles. It is very relevant to the recent announcement[1] made by the giant streaming platform NETFLIX which is consistently putting some efforts to stop password sharing. Unlike passwords, which can be shared/changed if compromised, biometrics are unique and permanent, making the breach of a biometric database potentially more severe. In a BA system, a live template of the user is compared against a reference template stored in the server and successful authentication is determined by a close match between the two templates. In such applications, we want to authenticate a user based on her biometrics in a privacy-preserving way, while also checking that a policy of attributes is satisfied; For instance, a possible scenario is that we want to enable online alcohol purchase for users who meet the following criteria: they are at least 18 years old, they have a paid subscription, they are located in Europe, and they have no medical history of alcoholism.

However, biometric data as well as associated attributes is highly privacy-sensitive, as it may reveal sensitive information such as ethnic origin or health conditions and therefore, should remain private. We propose SACfe as a tool for implementing a BA system which preserves the privacy of a user's attributes and biometric information. This scheme offers an efficient and secure way to compute the distance between the reference and the live biometric templates - without revealing the actual biometric information. In addition to secure authentication, SACfe enables content regulation based on specific attributes. In such a BA system, during the enrollment or registration phase, reference templates of users' biometric data (such as face, iris-scan, or fingerprint), are collected from the user's device, encrypted under the user's attributes (such as age, geographical location, IP address, subscription plan) using SACfe and stored in a database. During the authentication process, a live biometric template is captured from the user's device, and a SACfe secret key corresponding to the live biometric template and a certain policy is provided to the user. Since the objective of this application is to regulate content based on user attributes, the policy can be defined as P: age $>$ 18 and country located in Europe. The user now sends the secret key to the cloud server. The server can successfully compare the already stored encrypted reference template with the live template if the user's attributes satisfy the policy embedded into the secret key. Note that, the server can only compute the (Hamming) distance between the templates while remain oblivious about the biometric templates since SACfe guarantees that both the templates are encrypted in the ciphertext and secret key respectively. Moreover, SACfe ensures that the user's attributes are hidden from the server which only knows whether the policy is satisfied by the attributes or not. Furthermore, the unboundedness property of SACfe protocol allows the platform to accommodate various biometric types for different users and incorporate additional attributes for future access control, ensuring the system's scalability and adaptability as the platform grows.

Related Works. The first IPFE schemes for unbounded length vectors are proposed simultaneously and independently by Tomida and Takashima [37],

1. https://about.netflix.com/en/news/update-on-sharing-may-us

TABLE 1. COMPARISON WITH EXISTING AB-IPFE

Work	Access Control	Privacy	Data Size
[4]	MSP	Msg	bund
[28]	MSP	Msg	bund
[15]	Z-IP	Att, Msg	Unbd
	N-IP	Att, Msg	Unbd
Ours	Z-IP	Att, Func, Msg	Unbd
	MSP	Msg	Unbd

- Z-IP, N-IP, MSP: zero inner product, non-zero inner product, monotone span program.
- Att, Func, Msg: attribute, function, message.
- bund,unbd: bounded, unbounded.

and Dufour-Sans and Pointcheval [16]. The IPFEs [16], [37] do not offer access control features like our SACfe or unbounded AB-IPFE. An advanced variant of IPFE with access control is built by Datta et al. [14] which can handle unbounded length data, but it neither hides the full attribute nor the function. Recently, Dowerah et al. [15] constructed an unbounded ZP-IPFE scheme relying on the SXDH assumption, which hides attributes of the ciphertext, but secret keys reveal the embedded function vectors. Therefore, our SACfe ensures stronger security than [15] under the same assumption and concurrently, it provides better efficiency metrics than [15] as can be seen in Table 3 and 4.

There are other variants of AB-IPFE explored in the multi-users settings such as multi-client [5], [28] and multi-authority [6], [13]. However, all these works neither hide the users' attributes nor user-specific functions. Tomida [35] presented a partially hiding unbounded quadratic FE scheme where only a part of the function is hidden. Recently, Shi and Vanjani [34] proposed a function hiding multi-client IPFE without any access control. A comparison of our scheme with existing AB-IPFEs is provided in Table 1.

2. Technical Overview

2.1. SACfe: UZP-IPFE with Full-hiding Security

Our first contribution in this work is an unbounded inner-product IPFE with full-hiding security. Given a ciphertext computed on two vectors $(\mathbf{x} = (x_i)_{i\in[m_1]}, \mathbf{w} = (w_i)_{i\in[m_2]})$ and a key generated for two vectors $(\mathbf{y} = (y_i)_{i\in I_\mathbf{y}}, \mathbf{v} = (v_i)_{i\in I_\mathbf{v}})$ where all the vectors $\mathbf{w}, \mathbf{v}, \mathbf{x}, \mathbf{y}$ are *unbounded vectors* defined by their index sets $[m_1], [m_2], I_\mathbf{y}, I_\mathbf{v}$ respectively, this primitive computes $\langle \mathbf{x}, \mathbf{y}\rangle$ if $R(\mathbf{w}, \mathbf{v}) = 1$ (given by $\langle \mathbf{w}, \mathbf{v}\rangle = 0$ in this case) and the index sets satisfy a certain relation. Our construction achieves full-hiding security in the so-called *permissive setting* [37] which means that the index sets satisfy a *permissive relation*. In other words, the index sets of the vectors $\mathbf{y}$ and $\mathbf{v}$ are contained in the index sets of $\mathbf{x}$ and $\mathbf{w}$ respectively. Before we describe the full-hiding security model, we recall the permissive setting of the vectors:

- $\mathbf{x} \in \mathbb{Z}^{m_1}, \mathbf{w} \in \mathbb{Z}^{m_2}$.
- $\mathbf{y} = (y_i)_{i\in I_\mathbf{y}} \in \mathbb{Z}^{|I_\mathbf{y}|}, \mathbf{v} = (v_i)_{i\in I_\mathbf{v}} \in \mathbb{Z}^{|I_\mathbf{v}|}$.
- *Permissive relation* $\mathcal{R}_p$: $((\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v})) \in \mathcal{R}_p$ if and only if $I_\mathbf{y} \subseteq [m_1]$ and $I_\mathbf{v} \subseteq [m_2]$. Then, $\langle \mathbf{x}, \mathbf{y}\rangle = \sum_{i\in I_\mathbf{y}} x_i y_i$, $\langle \mathbf{w}, \mathbf{v}\rangle = \sum_{i\in I_\mathbf{v}} w_i v_i$.

Our work, for the first time in the literature, considers unbounded ZP-IPFE (UZP-IPFE) with *full-hiding security* (i.e. attribute-hiding and function-hiding simultaneously). The channel of access control is kept secret from the server by hiding attribute $\mathbf{w}$ in the ciphertext and hiding $\mathbf{y}$ in the secret keys simultaneously. As discussed above, full-hiding and unboundedness properties make ZP-IPFE significantly more relevant in practice. We, therefore, introduce FH-IND (Full-Hiding Indistinguishability) security which allows all efficient adversaries to pose challenges to both the encryption and key generation oracles in an interleaved manner. That being said, we mention that the FH-IND security in this paper is *non-adaptive* in nature on the challenge attribute vector $\mathbf{w}$. In particular, the FH-IND adversary in this paper, commits to a sequence of challenge attributes $\{(\mathbf{w}_\mu^{(0)}, \mathbf{w}_\mu^{(1)})\}_\mu$ just after the setup phase. The adversary is allowed to adaptively select *many* functions of the form $((\mathbf{y}_\ell^{(0)}, \mathbf{v}_\ell), (\mathbf{y}_\ell^{(1)}, \mathbf{v}_\ell))$ as a part of key generation challenge and can also choose *many* pair of challenge messages of the form $(\mathbf{x}_\mu^{(0)}, \mathbf{x}_\mu^{(1)})$ adaptively for encryption. Moreover, the adversary is allowed to choose any arbitrary length vectors for key generation and encryption oracles, but the vectors appearing in the same pair must have the same index sets. It is non-trivial and challenging to handle such queries since we aim to achieve the unboundedness property along with full security. As the challenger returns a secret key on $(\mathbf{y}_\ell^{(b)}, \mathbf{v}_\ell)$ and a ciphertext on $(\mathbf{x}_\mu^{(b)}, \mathbf{w}_\mu^{(b)})$, the adversary has to guess b chosen uniformly at random by the challenger. To restrict the adversary from trivially winning the security game, if the queried vectors are *permissive* then they must satisfy $\langle \mathbf{x}_\mu^{(0)}, \mathbf{y}_\ell^{(0)}\rangle = \langle \mathbf{x}_\mu^{(1)}, \mathbf{y}_\ell^{(1)}\rangle$ whenever it holds that $\langle \mathbf{w}_\mu^{(0)}, \mathbf{v}_\ell\rangle = \langle \mathbf{w}_\mu^{(1)}, \mathbf{v}_\ell\rangle = 0$.

Construction Overview. The starting point of our construction is the function-hiding unbounded (non-attribute-based) IPFE of [37], which we call TT-IPFE. TT-IPFE encodes a message vector $\mathbf{x} = (x_1, \ldots, x_{m_1})$ into $([\![\mathbf{c}_i^1]\!]_1)_{i\in[m_1]}$ and a key vector $\mathbf{y} = (y_j)_{j\in I_\mathbf{y}}$ into $([\![\mathbf{k}_j^1]\!]_2)_{j\in I_\mathbf{y}}$ as

$$[\![\mathbf{c}_i^1]\!]_1 = [\![(x_i, 0, \alpha, 0)\mathbf{B}_i]\!]_1,\ [\![\mathbf{k}_j^1]\!]_2 = [\![(y_j, 0, \gamma_j, 0)\mathbf{B}_j^*]\!]_2$$

where $\mathbf{B}_i, \mathbf{B}_i^*$ are the orthonormal bases of $\mathsf{GL}_4(\mathbb{Z}_p)$ and $\mathbf{B}_i$ encodes the i^{th} component x_i, $\mathbf{B}_j^*$ encodes the j^{th} component y_j.

A naive combination of two independent copies of TT-IPFE— one strain encodes $\mathbf{x}$ to $([\![\mathbf{c}_i^1]\!]_1)_{i\in[m_1]}$ and $\mathbf{y}$ by $\mathsf{sk}_\mathbf{y} = ([\![\mathbf{k}_j^1]\!]_2)_{j\in I_\mathbf{y}}$ whereas the second strain encodes $\mathbf{w}$ to $([\![\mathbf{c}_i^2]\!]_1)_{i\in[m_2]}$ and $\mathbf{v}$ by $\mathsf{sk}_\mathbf{v} = ([\![\mathbf{k}_j^2]\!]_2)_{j\in I_\mathbf{v}}$— is insecure mainly due to natural mix-and-match attacks. In particular, given secret keys $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\mathsf{sk}_\mathbf{y}, \mathsf{sk}_\mathbf{v}), \mathsf{SK}_{\mathbf{z},\mathbf{u}} = (\mathsf{sk}_\mathbf{z}, \mathsf{sk}_\mathbf{u})$ for $((\mathbf{y}, I_\mathbf{y}), (\mathbf{v}, I_\mathbf{v}))$ and $((\mathbf{z}, I_\mathbf{z}), (\mathbf{u}, I_\mathbf{u}))$ respectively, one can easily compute a legitimate secret key $\mathsf{SK}_{\mathbf{z},\mathbf{v}} = (\mathsf{sk}_\mathbf{z}, \mathsf{sk}_\mathbf{v})$ corresponding to the function $((\mathbf{z}, I_\mathbf{z}), (\mathbf{v}, I_\mathbf{v}))$. The mixing of secret key components would lead to an attack that breaks the security of SACfe. To combine the two strains securely, a secret share of zero is carefully distributed to bind the secret key components $[\![\mathbf{k}_j^1]\!]_2$ and $[\![\mathbf{k}_j^2]\!]_2$. Although the construction idea of our SACfe is adopted from [15], the proof technique is quite complicated as we aim to achieve full-hiding security. We give further details below about how we achieved full-hiding security. Our SACfe encodes the key and ciphertext vectors as follows:

$$[\![\mathbf{c}_i^1]\!]_1 = [\![(x_i, 0, \alpha, 0)\mathbf{B}_i]\!]_1,\ [\![\mathbf{k}_i^1]\!]_2 = [\![(y_i, 0, \gamma_i, 0)\mathbf{B}_i^*]\!]_2$$
$$[\![\mathbf{c}_j^2]\!]_1 = [\![(\delta w_j, \alpha, 0)\widetilde{\mathbf{B}}_j]\!]_1,\ [\![\mathbf{k}_j^2]\!]_2 = [\![(\omega v_j, \widetilde{\gamma}_j, 0)\widetilde{\mathbf{B}}_j^*]\!]_2$$

where $\{\{\gamma_i\}_i, \{\widetilde{\gamma}_j\}_j\}$ forms a secret share of zero and the bases are sampled via a pseudorandom function depending on the indices i and j of the vectors. Another type of mix-and-match attack would arise if we used the same basis pair $(\mathbf{B}, \mathbf{B}^*)$ across all $\mathbf{c}_i^1$ and $\mathbf{k}_i^1$ (or $(\widetilde{\mathbf{B}}, \widetilde{\mathbf{B}}^*)$ across all $\mathbf{c}_j^2$ and $\mathbf{k}_j^2$). In that case, the adversary can pair the ciphertext component $[\![\mathbf{c}_i^1]\!]_1$ with the unmatching key component $[\![\mathbf{k}_{i'}^1]\!]_2$ where $i' \neq i$ (or $[\![\mathbf{c}_j^2]\!]_1$ with the unmatching key component $[\![\mathbf{k}_{j'}^2]\!]_2$ where $j' \neq j$). This would reveal unwanted information about the message or the function. To prevent the computation of such pairing operation between the ciphertext and key components, we utilize a pseudorandom function (with two independent keys) to sample independent and pseudorandom bases $(\mathbf{B}_i, \mathbf{B}_i^*)$ associated with index i (or $(\widetilde{\mathbf{B}}_j, \widetilde{\mathbf{B}}_j^*)$ associated with index j).

The decryption algorithm works as follows. First, it recovers $[\![\omega\delta\langle\mathbf{w}, \mathbf{v}\rangle + \alpha\sum_{j\in I_\mathbf{v}} \widetilde{\gamma}_j]\!]_T$ from $\{\mathbf{c}_j^2, \mathbf{k}_j^2\}_j$. If the zero-predicate relation is satisfied then this yields $[\![\alpha\sum_{j\in I_\mathbf{v}} \widetilde{\gamma}_j]\!]_T$. Secondly, the other inner product is computed as $[\![\langle\mathbf{x}, \mathbf{y}\rangle + \alpha\sum_{i\in I_\mathbf{y}} \gamma_i]\!]_T$ from $\{\mathbf{c}_i^1, \mathbf{k}_i^1\}_i$. Now, if the key generator sets $\sum_{i\in I_\mathbf{y}} \gamma_i + \sum_{j\in I_\mathbf{v}} \widetilde{\gamma}_j = 0$, that is $\{\gamma_i, \widetilde{\gamma}_j\}$ forms a secret shares of 0, then we can recover $[\![\langle\mathbf{x}, \mathbf{y}\rangle]\!]_T$ by combining the outputs.

Full-hiding Security. We briefly outline the proof here. Suppose $(\mathbf{x}_\mu^{(0)}, \mathbf{w}_\mu^{(0)})$ and $(\mathbf{x}_\mu^{(1)}, \mathbf{w}_\mu^{(1)})$ are the μ^{th} challenge message-attribute vector pairs. The adversary can ask mainly the following three types of secret keys for the ℓ^{th} key-predicate pairs $(\mathbf{y}_\ell^{(0)}, \mathbf{v}_\ell)$ or $(\mathbf{y}_\ell^{(1)}, \mathbf{v}_\ell)$:

1) $(\mathbf{x}_\mu^{(b)}, \mathbf{y}_\ell^{(b)}) \notin \mathcal{R}_p$ or $(\mathbf{w}_\mu^{(b)}, \mathbf{v}_\ell) \notin \mathcal{R}_p$.
2) $(\mathbf{x}_\mu^{(b)}, \mathbf{y}_\ell^{(b)}), (\mathbf{w}_\mu^{(b)}, \mathbf{v}_\ell) \in \mathcal{R}_p$, but $R(\mathbf{w}_\mu^{(0)}, \mathbf{v}_\ell) \neq 1$ and $R(\mathbf{w}_\mu^{(1)}, \mathbf{v}_\ell) \neq 1$.
3) $(\mathbf{x}_\mu^{(b)}, \mathbf{y}_\ell^{(b)}), (\mathbf{w}_\mu^{(b)}, \mathbf{v}_\ell) \in \mathcal{R}_p$ and $R(\mathbf{w}_\mu^{(0)}, \mathbf{v}_\ell) = R(\mathbf{w}_\mu^{(1)}, \mathbf{v}_\ell) = 1$ and $\langle\mathbf{x}_\mu^{(0)}, \mathbf{y}_\ell^{(0)}\rangle = \langle\mathbf{x}_\mu^{(1)}, \mathbf{y}_\ell^{(1)}\rangle$.

Why Dowerah et al. [15] Technique does not work? The main difference between the proof techniques of ours and Dowerah et al. [15] is the fact that in our case the adversary queries many challenge functional vectors $\mathbf{y}_\ell^{(0)}$ and $\mathbf{y}_\ell^{(1)}$, whereas there was no challenge on such functional vectors in case of [15]. Therefore, a direct application of [15] will not suffice for our purpose of hiding information encoded into the functional vectors associated with secret keys. To replace $\mathbf{y}_\ell^{(0)}$ with $\mathbf{y}_\ell^{(1)}$ in $([\![\mathbf{k}_j^1]\!]_2)_{j\in I_\mathbf{y}}$ while keeping $\mathbf{v}$ the same in $([\![\mathbf{k}_j^2]\!]_2)_{j\in I_\mathbf{v}}$ and at the same time changing $\mathbf{x}_\mu^{(0)}$ with $\mathbf{x}_\mu^{(1)}$ require more delicate hybrid approach than in [15].

Our Ideas. At a very high level, we develop an amalgamation of the proof techniques of TT-IPFE and [15]. To achieve the requirement of full-hiding security of SACfe, we design the following sequence of hybrids. During the argument, we first change the encoding $(y_{\ell,i}^{(0)}, 0, \gamma_{\ell,i}, 0)\mathbf{B}_i^*$ to $(y_{\ell,i}^{(0)}, y_{\ell,i}^{(1)}, \gamma_{\ell,i}, 0)\mathbf{B}_i^*$ following TT-IPFE. After this, we change the ciphertext from $(x_{\mu,i}^{(0)}, 0, \alpha_\mu, 0)\mathbf{B}_i, (\delta_\mu w_{\mu,j}^{(0)}, \alpha_\mu, 0)\widetilde{\mathbf{B}}_j$ to $(0, x_{\mu,i}^{(1)}, \alpha_\mu, 0)\mathbf{B}_i, (\delta_\mu w_{\mu,j}^{(1)}, \alpha_\mu, 0)\widetilde{\mathbf{B}}_j$. This step is crucially handled by integrating the proof techniques of TT-IPFE and [15]. More specifically, we use techniques from TT-IPFE to change the message-encoding component from $(x_{\mu,i}^{(0)}, 0, \alpha_\mu, 0)\mathbf{B}_i$ to $(0, x_{\mu,i}^{(1)}, \alpha_\mu, 0)\mathbf{B}_i$ and the proof ideas of [15] to change the attribute-encoding component from $(\delta_\mu w_{\mu,j}^{(0)}, \alpha_\mu, 0)\widetilde{\mathbf{B}}_j$ to $(\delta_\mu w_{\mu,j}^{(1)}, \alpha_\mu, 0)\widetilde{\mathbf{B}}_j$. It is important to note that since at this stage, the function-encoding component $(y_{\ell,i}^{(0)}, y_{\ell,i}^{(1)}, \gamma_{\ell,i}, 0)\mathbf{B}_i^*$ is independent of the challenge bit, we can freely change the attribute vector from $w_{\mu,j}^{(0)}$ to $w_{\mu,j}^{(1)}$ in parallel with the switching of $x_{\mu,i}^{(0)}$ with $x_{\mu,i}^{(1)}$. We also utilize the additional subspaces to encode secret shares of a uniform value into the components of non-permissive secret keys to restrict the adversary from extracting useful information about the challenge bit.

Next, we swap function and message challenges together. That is, the function-encoding and message-encoding components are changed from $(y_{\ell,i}^{(0)}, y_{\ell,i}^{(1)}, \gamma_{\ell,i}, 0)\mathbf{B}_i^*, (0, x_{\mu,i}^{(1)}, \alpha_\mu, 0)\mathbf{B}_i$ to $(y_{\ell,i}^{(1)}, y_{\ell,i}^{(0)}, \gamma_{\ell,i}, 0)\mathbf{B}_i^*, (x_{\mu,i}^{(1)}, 0, \alpha_\mu, 0)\mathbf{B}_i$ respectively. Finally, the function-encoding component is altered to $(y_{\ell,i}^{(1)}, 0, \gamma_{\ell,i}, 0)\mathbf{B}_i^*$. Although the core technical idea discussed above provides a very high-level intuition on how the full-hiding security of SACfe is achieved, there are several subtle challenges. We present a complete and formal security analysis in

Section 4.2.

2.2. Our Strict UAB-IPFE

Our second contribution is an UAB-IPFE in the *strict setting* where decryption is successful if the index sets of the key and message vectors are equal. The UAB-IPFE is inspired by the recently proposed bounded AB-IPFE of Nguyen et al. [28]. We extend their framework to handle unbounded length data without compromising efficiency of their scheme. Our UAB-IPFE is built in the public-key setting with constant size public keys and succinct secret keys that offers a more expressive access control realized by LSSS. Previously, the only UAB-IPFE with succinct secret keys is known to handle (non-zero) inner product policies [15].

Bounded AB-IPFE of [28]. We start by recalling the bounded AB-IPFE scheme of Nguyen et al. [28]. Their scheme is based on the framework of DPVS [30]. To embed an access structure $\mathbb{A}$ into the secret key, a set of random secret shares $(a_j)_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ of $a_0 \leftarrow \mathbb{Z}_p$ is sampled via LSSS based on $\mathbb{A}$ where $\mathsf{List\text{-}Att}(\mathbb{A})$ denotes the list of attributes appearing in the access structure. On the other hand, an attribute set S is embedded in the ciphertext component in a way that the secret a_0 is recovered during decryption if there exists $A \subseteq \mathsf{S}$ such that $A \subseteq \mathsf{List\text{-}Att}(\mathbb{A})$. This functionality is implemented using the secret key and ciphertext components

$$[\![\mathbf{k}_{\mathsf{ac},j}]\!]_2 = [\![(\pi_j(j,1), za_j)\mathbf{F}^*]\!]_2;$$
$$[\![\mathbf{c}_{\mathsf{ac},j}]\!]_1 = [\![(\sigma_j(1,-j), \psi)\mathbf{F}]\!]_1$$

where $(\mathbf{F}, \mathbf{F}^*)$ are the bases of DPVS, π_j, z, σ_j, ψ are random integers. For the authorized set of attributes S, i.e., $A \subseteq \mathsf{S}$, we can now use the reconstruction coefficients $\{c_j\}_{j\in A}$ satisfying $\sum_{j\in A} c_j a_j = a_0$ to compute $[\![z\psi a_0]\!]_T$ by pairing the vectors $[\![\mathbf{k}_{\mathsf{ac},j}]\!]_2$ and $[\![\mathbf{c}_{\mathsf{ac},j}]\!]_1$.

Next, the remaining components of secret key and ciphertext related to the key and message vectors respectively are generated based on the ALS-style [7] encoding techniques using a master secret key $(\mathbf{U} = (u_i)_i, \mathbf{S} = (s_i)_i)$. The key and message vectors $\mathbf{y}, \mathbf{x}$ are encoded as follows:

$$[\![\mathbf{k}_{\mathsf{fe}}]\!]_2 = [\![(\langle \mathbf{U}, \mathbf{y}\rangle, \langle \mathbf{S}, \mathbf{y}\rangle, za_0)\mathbf{H}^*]\!]_2$$
$$[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 = [\![(\omega, \mu\omega, \psi)\mathbf{H}]\!]_1$$
$$[\![t_i]\!]_1 = [\![\omega(u_i + \mu s_i) + x_i]\!]_1$$

where $(\mathbf{H}, \mathbf{H}^*)$ are the bases of DPVS and ω is the encryption randomness. In order to compute $[\![t_i]\!]_1$ at the time of encryption the master public key of the system must contain $\{[\![u_i + \mu s_i]\!]_1\}_i$ where $\mu \leftarrow \mathbb{Z}_p$ is kept secret. Now, the pairing between $[\![\mathbf{k}_{\mathsf{fe}}]\!]_2$ and $[\![\mathbf{c}_{\mathsf{fe}}]\!]_1$ yields the masking term $[\![\omega \cdot (\langle \mathbf{U}, \mathbf{y}\rangle + \mu\langle \mathbf{S}, \mathbf{y}\rangle) + z\phi a_0]\!]_T$ and hence decryption follows.

Towards Unboundedness. As we can see from the above bounded AB-IPFE, the master key components $\mathbf{U}, \mathbf{S}$ and $\{[\![u_i + \mu s_i]\!]_1\}_{i\in[n]}$ are generated in setup depending on the vector length n. Our first observation is that if we provide $\mathbf{U}$ and $\mathbf{S}$ in the exponent of $\mathbb{G}_1$ along with $[\![\mu]\!]_1$ as a part of the master public key then any one can compute $\{[\![u_i + \mu s_i]\!]_i\}_{i\in[n]}$ during encryption. Our second observation is that the master secret key components u_i and s_i, once sampled, are fixed for each index i throughout the scheme.

Based on these observations, it seems that we can generate u_i, s_i on the fly deterministically using hash functions. More precisely, for each index i, the hash functions $\mathcal{H}_1$, $\mathcal{H}_2$ generate $\mathcal{H}_1(i) = [\![u_i]\!]_2$ and $\mathcal{H}_2(i) = [\![s_i]\!]_2$ on the fly. Therefore, the master keys in the transformed scheme does not depend on n as only μ plays the role of master secret key and $[\![\mu]\!]_1$ is sufficient to generate the public key component $[\![u_i + \mu s_i]\!]_T$ deterministically with the help of a *hash-and-pairing* mechanism at the time of encryption. The IPFE-related parts of secret key $\mathbf{k}_{\mathsf{fe}}$ and ciphertext $\mathbf{c}_{\mathsf{fe}}, t_i$ are now computed as follows:

$$[\![\mathbf{k}_{\mathsf{fe}}]\!]_2 = [\![(\sum_{i\in I_{\mathbf{y}}} u_i y_i, \sum_{i\in I_{\mathbf{y}}} s_i y_i, za_0)\mathbf{H}^*]\!]_2; \quad \begin{matrix}\mathcal{H}_1(i) = [\![u_i]\!]_2 \\ \mathcal{H}_2(i) = [\![s_i]\!]_2\end{matrix}$$
$$[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 = [\![(\omega, \mu\omega, \psi)\mathbf{H}]\!]_1;$$
$$[\![t_i]\!]_1 = [\![x_i]\!]_T \cdot e(g_1, \omega\mathcal{H}_1(i)) \cdot e([\![\mu]\!]_1, \omega\mathcal{H}_2(i)) \;\forall\; i \in I_{\mathbf{x}}$$

If the index sets $I_{\mathbf{x}}, I_{\mathbf{y}}$ are equal then $\prod_i y_i [\![t_i]\!]_T = [\![\langle \mathbf{x}, \mathbf{y}\rangle + \omega \sum_{i\in I_{\mathbf{y}}} u_i y_i + \mu\omega \sum_{i\in I_{\mathbf{y}}} s_i y_i]\!]_T$. Therefore, the decryption follows similar to the bounded scheme. We now briefly sketch the selective message-hiding security of the above UAB-IPFE. Before going to the discussion, we would like to point out that additional hidden spaces may be required to add in order to achieve our security goal.

Overview of Security. We first recall that for all secret key queries $(\mathbf{y}_\ell, \mathbb{A}_\ell)$ where the challenge attribute set S satisfies the policies $\mathbb{A}_\ell$, it must hold $\langle \mathbf{x}^{(0)}, \mathbf{y}_\ell\rangle = \langle \mathbf{x}^{(1)}, \mathbf{y}_\ell\rangle$. At a very high level, our security argument follows two steps – in the *first* step, we randomize $\langle \mathbf{c}_{\mathsf{fe}}, \mathbf{k}_{\ell,\mathsf{fe}}\rangle$ by introducing an additional term $r'_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle$ where $\Delta\mathbf{x} = \mathbf{x}^{(0)} - \mathbf{x}^{(1)}$, in the *second* step, we apply the DBDH assumption and program the random oracle model $\mathcal{H}_1, \mathcal{H}_2$ depending on the decryption criteria to hide the challenge bit from the adversary's view. For the first step we use a masking strategy similar to Nguyen et al. [28]. The main idea is to introduce additional hidden subspaces to the vectors (using SXDH assumption) as follows:

$$[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 = [\![(\sum_{i\in I_{\mathbf{y}_\ell}} u_i y_{\ell,i}, \sum_{i\in I_{\mathbf{y}_\ell}} s_i y_{\ell,i}, za_{\ell,0}, r'_{\ell,0}\langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle)\mathbf{H}^*]\!]_2;$$
$$[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 = [\![(\omega, \mu\omega, \psi, \tau)\mathbf{H}]\!]_1$$

Note that, if decryption is not successful then we always have $\delta_\ell = \langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle \neq 0$ for all such secret keys. It ensures that the secret component $a_{\ell,0}\psi z$ is masked with $r'_{\ell,0}\tau\delta_\ell$ which prevents the adversary to gain any unnecessary information about the message vector. Then before going to the second step, we relocate

the master secret key component μ from ciphertext to secret key via a simple basis transformation. The updated IPFE-realted vector takes the form

$$[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 = [\![(\langle \mathbf{U}, \mathbf{y}_\ell \rangle, \mu\langle \mathbf{S}, \mathbf{y}_\ell \rangle + r'_{\ell,0}\delta_\ell, za_{\ell,0}, r'_{\ell,0}\delta_\ell)\mathbf{H}^*]\!]_2;$$
$$[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 = [\![(\omega, \omega, \psi, \tau)\mathbf{H}]\!]_1.$$

We observe that the implicit IPFE encryption mechanism of the transformed scheme in the current hybrid coincides with the UIPFE of Dufour Sans et al. [16]. Consequently, we can now use the DBDH assumption and program the hash functions following the ideas of Dufour Sans et al. [16] and Datta et al. [13] to hide the challenge bit. Although the overall techniques of the security analysis are inspired from [13], [16], [29] there are crucial technical challenges which we overcome during the security analysis.

3. Preliminaries

Notations. For some prime p, $\mathbb{Z}_p$ denotes a finite field of order p and for $n \in \mathbb{N}$, the set $\mathsf{GL}_n(\mathbb{Z}_p)$ denotes all $n \times n$ invertible matrices with entries from $\mathbb{Z}_p$. We indicate by $a \leftarrow S$ the process of random sampling of an element a from the finite set S. We consider a bold uppercase letter to represent a matrix, e.g., $\mathbf{A}$, a bold lowercase letter to indicate a vector, e.g., $\mathbf{x}$ and $I_{\mathbf{x}}$ denotes the index set of the vector $\mathbf{x}$. For example, if $\mathbf{x} = (x_1, x_3, x_8)$ then we write $I_{\mathbf{x}} = \{1, 3, 8\}$. Consider g_ι is a generator of the cyclic group $\mathbb{G}_\iota$. If $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is an n-tuple vector then $[\![\mathbf{x}]\!]_\iota = (g_\iota^{x_1}, g_\iota^{x_2}, \ldots, g_\iota^{x_n})$. For $a, u \in \mathbb{Z}_p$, we represent $a[\![u]\!]_\iota = g_\iota^{au}$. For a matrix $\mathbf{A} = (a_{ij}) \in \mathsf{GL}_n(\mathbb{Z}_p)$, we define $[\![\mathbf{A}]\!]_\iota = g_\iota^{\mathbf{A}}$ where exponentiation is carried out component-wise and $\mathbf{a}_i$ represents i-th row vector of $\mathbf{A}$. For $n \in \mathbb{N}$, we choose random dual orthonormal bases $(\mathbf{B}, \mathbf{B}^* = (\mathbf{B}^{-1})^\top)$ [15] as $\mathbf{B} \leftarrow \mathsf{GL}_n(\mathbb{Z}_p)$ and $[\![\mathbf{B}]\!]_1, [\![\mathbf{B}^* = (\mathbf{B}^{-1})^\top]\!]_2$ are dual orthonormal bases of vector spaces $V = \mathbb{G}_1^n, V^* = \mathbb{G}_2^n$ respectively and E be extended bilinear map defined as $E([\![\mathbf{x}\mathbf{B}]\!]_1, [\![\mathbf{y}\mathbf{B}^*]\!]_2) = [\![\langle \mathbf{x}, \mathbf{y} \rangle]\!]_T$. A function $\mathsf{negl} : \mathbb{N} \to [0, 1]$ is said to be *negligible* if for every $c \in \mathbb{N}$ there exists a $\lambda_c \in \mathbb{N}$ such that $\mathsf{negl}(\lambda) \leq \frac{1}{\lambda^c}$ for all $\lambda > \lambda_c$.

3.1. Bilinear Group

A bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ consists of a prime p, two multiplicative source groups $\mathbb{G}_1, \mathbb{G}_2$ and a target group $\mathbb{G}_T$ with the order $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$ where g_1, g_2 are the generators of the group $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Let us consider a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. It satisfies the following:

- *bilinearity:* $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2,\ a, b \in \mathbb{Z}_p$ and
- *non-degeneracy:* $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$.

A bilinear group generator $\mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ takes the security parameter λ and outputs a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ with a λ-bit prime integer p.

3.2. Pseudo-Random Function

A pseudo-random function (PRF) family $\mathcal{F} = \{F_K\}_{K \in \mathcal{K}_\lambda}$ with a keyspace $\mathcal{K}_\lambda$, a domain $\mathcal{X}_\lambda$ and a range $\mathcal{Y}_\lambda$ is a function family that consists of functions $F_K : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$. Let Rand_λ be the set of random functions with domain $\mathcal{X}_\lambda$ and co-domain $\mathcal{Y}_\lambda$. Then for all PPT adversaries $\mathcal{A}$, the following holds:

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}}(1^\lambda) = \Big|\Pr[\mathcal{A}^{F_K(\cdot)}(\lambda) = 1] - \Pr[\mathcal{A}^{\mathsf{Rand}(\cdot)}(\lambda) = 1]\Big| \leq \mathsf{negl}(\lambda)$$

with $K \leftarrow \mathcal{K}_\lambda$ and $\mathsf{Rand}(\cdot) \leftarrow \mathsf{Rand}_\lambda$.

3.3. Complexity Assumptions

Assumption 1 (Symmetric External Diffie-Hellman (SXDH) Assumption). For $\iota \in \{1, 2\}$, we define the distribution $(D, [\![t_\beta]\!]_\iota)$ on a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ as

$$D = (\mathsf{G}, [\![a]\!]_\iota, [\![u]\!]_\iota) \text{ for } a, u \leftarrow \mathbb{Z}_p$$
$$[\![t_\beta]\!]_\iota = [\![au + \beta f]\!]_\iota \text{ for } \beta \in \{0, 1\} \text{ and } f \leftarrow \mathbb{Z}_p.$$

We say that the SXDH assumption holds in G if for all PPT adversaries $\mathcal{A}$, if there exists a *negligible* function $\mathsf{negl}(\cdot)$ satisfying the following:

$$\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{A}}(\lambda) = \Big|\Pr[\mathcal{A}(D, [\![t_0]\!]_\iota) = 1] - \Pr[\mathcal{A}(D, [\![t_1]\!]_\iota) = 1]\Big| \leq \mathsf{negl}(\lambda)$$

Assumption 2 (Decisional Bilinear Diffie-Hellman (DBDH) Assumption). Consider a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$. The DBDH assumption holds in G if for all PPT adversaries $\mathcal{A}$, there exists a non-negligible function $\mathsf{negl}(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{A}}(\lambda) = \Big|\Pr[\mathcal{A}(\mathsf{G}, [\![a]\!]_1, [\![b]\!]_1, [\![a]\!]_2, [\![c]\!]_2, [\![abc]\!]_T) = 1] - \Pr[\mathcal{A}(\mathsf{G}, [\![a]\!]_1, [\![b]\!]_1, [\![a]\!]_2, [\![c]\!]_2, [\![d]\!]_T) = 1]\Big| \leq \mathsf{negl}(\lambda)$$

where $a, b, c, d \leftarrow \mathbb{Z}_p$.

3.4. Access Structures and Linear Secret Sharing Schemes

Definition 1 (Access Structure [28]). Let $\mathsf{Att} = \{\mathsf{att}_1, \ldots, \mathsf{att}_n\}$ be a finite set of attributes. An access structure over Att is a collection $\mathbb{A}$ of non-empty subsets of $\{\mathsf{Att}\}$, i.e., $\mathbb{A} \subseteq 2^{\{\mathsf{Att}\}} \setminus \{\emptyset\}$. A set contained in $\mathbb{A}$ is called authorized, otherwise it is called unauthorized. An access structure $\mathbb{A}$ is *monotone* if $\mathsf{S}_1 \subseteq \mathsf{S}_2 \subseteq \mathbb{A}$ and $\mathsf{S}_1 \in \mathbb{A}$ imply $\mathsf{S}_2 \in \mathbb{A}$. Given a set of attributes $\mathsf{S} \subseteq \mathsf{Att}$, we write $\mathbb{A}(\mathsf{S}) = 1$ if and only if there exists $A \subseteq \mathsf{S}$ such that A is authorized. Note that, $\mathsf{List\text{-}Att}(\mathbb{A})$ is the list of attributes appearing in the access structure $\mathbb{A}$.

We are interested in *linear secret sharing schemes* (LSSS) defined below.

Definition 2 (Linear Secret Sharing Schemes [28]). Let K be a field, $d, f \in \mathbb{N}$, and Att be a finite

universe of attributes. A Linear Secret Sharing Scheme LSSS over K for an access structure $\mathbb{A}$ over Att is specified by a share-generating matrix $\mathbf{A} \in K^{d \times f}$ such that for any $I \subset [d]$, there exists a vector $\mathbf{c} \in K^d$ with support I and $\mathbf{c} \cdot \mathbf{A} = (1, 0, ..., 0)$ if and only if $\{\mathsf{att}_i \mid i \in I\} \in \mathbb{A}$.

To share a secret s, pick uniformly random values $v_2, \ldots, v_d \leftarrow K$ and generate a vector of n shares as $\mathbf{s} := (s, v_2, \ldots, v_d) \cdot \mathbf{A}^\top$ such that the share for attribute att_i is the i-th coordinate s_i of $\mathbf{s}$. Only an authorized set $\{\mathsf{att}_i \mid i \in I\} \in \mathbb{A}$ can recover $\mathbf{c}$ to reconstruct s by computing $\mathbf{c} \cdot \mathbf{s}^\top = \mathbf{c} \cdot (\mathbf{A} \cdot (s, v_2, \ldots, v_d)^\top) = s$. For any unauthorized set, reconstructing the secret will result in a closely random value.

3.5. SACfe: Secure Access Control in FE

We present the syntax of our SACfe protocol. First, we define an unbounded AB-IPFE (UAB-IPFE) with general access structure and then we discuss the property it should satisfy for a SACfe. We use notations similar to the work of [15], [28]. The functionality class is $\mathcal{F}_{\mathsf{ip}} \times \mathsf{AC\text{-}K}$. The evaluation functions $\mathcal{F}_{\mathsf{ip}}$ is the class of inner product functions given by $\mathcal{F}_{\mathsf{ip}} = \{F_{\mathbf{y} \in \mathcal{Y}_\lambda} : \mathcal{X}_\lambda \to Z_p\}_\lambda$ where $\{\mathcal{X}_\lambda\}_\lambda$ and $\{\mathcal{Y}_\lambda\}_\lambda$ denote the message space and key space respectively. The access control is represented by a relation $\mathcal{R}_{\mathsf{ac}} : \mathsf{AC\text{-}K} \times \mathsf{AC\text{-}CT} \to \{0, 1\}$ for some sets $\mathsf{AC\text{-}K}$ and $\mathsf{AC\text{-}CT}$. For any two vectors $\mathbf{x} = (x_i)_{i \in I_\mathbf{x}}, \mathbf{y} = (y_i)_{i \in I_\mathbf{y}}$ with associated index sets $I_\mathbf{x}$ and $I_\mathbf{y}$, we define a *permissive* relation $\mathcal{R}_p$ such that

$$(\mathbf{x}, \mathbf{y}) \in \mathcal{R}_p \text{ if and only if } I_\mathbf{y} \subseteq I_\mathbf{x}$$

and, in this case, the inner product is defined as $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i \in I_\mathbf{y}} x_i y_i$. On the other hand, a *strict* relation $\mathcal{R}_s$ between the vectors $\mathbf{x}$, $\mathbf{y}$ is defined as

$$(\mathbf{x}, \mathbf{y}) \in \mathcal{R}_s \text{ if and only if } I_\mathbf{y} = I_\mathbf{x} = I(\text{say})$$

and, in this case, the inner product is given by $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i \in I} x_i y_i$. Let us consider a toy example. If $\mathbf{x} = (x_i)_{i \in I_\mathbf{x}}$ and $\mathbf{y} = (y_i)_{i \in I_\mathbf{y}}$ with $I_\mathbf{x} = \{1, 2, 3, 4\}$ and $I_\mathbf{y} = \{1, 3, 4\}$, then it is easy to verify that $(\mathbf{x}, \mathbf{y}) \in \mathcal{R}_p$ since $I_\mathbf{y} \subseteq I_\mathbf{x}$ and the inner product value is $\langle \mathbf{x}, \mathbf{y} \rangle = x_1 y_1 + x_3 y_3 + x_4 y_4$. However, we see that $(\mathbf{x}, \mathbf{y}) \notin \mathcal{R}_s$ since $I_\mathbf{x} \neq I_\mathbf{y}$.

We define our UAB-IPFE protocol for the function class $\mathcal{F}_{\mathsf{ip}}$ and access control relation $\mathcal{R}_{\mathsf{ac}}$. Here, a ciphertext is associated with $(\mathbf{x}, \mathsf{ac\text{-}ct}) \in \mathcal{X}_\lambda \times \mathsf{AC\text{-}CT}$ and a secret key is associated with $(\mathbf{y}, \mathsf{ac\text{-}k}) \in \mathcal{Y}_\lambda \times \mathsf{AC\text{-}K}$. In the permissive (or strict) setting, decryption outputs the inner product between $\mathbf{x}$ and $\mathbf{y}$ if and only if $\mathcal{R}_{\mathsf{ac}}(\mathsf{ac\text{-}ct}, \mathsf{ac\text{-}k}) = 1$ and $(\mathbf{x}, \mathbf{y}) \in \mathcal{R}_p$ (or $((\mathbf{x}, \mathbf{y}) \in \mathcal{R}_s)$). The permissive setting is more expressive and appealing as it allows computation over a large encrypted database (e.g., $\mathbf{x}$) using a secret key associated with a variable length functions (e.g., $\mathbf{y}$) as long as the function size remains smaller than the size of the actual database.

Definition 3 (UAB-IPFE for $(\mathcal{F}_{\mathsf{ip}}, \mathcal{R}_{\mathsf{ac}})$***).*** A UAB-IPFE scheme for $(\mathcal{F}_{\mathsf{ip}}, \mathcal{R}_{\mathsf{ac}})$ consists of the following algorithms:

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{PP}, \mathsf{EK}, \mathsf{MSK})$: It takes the security parameter λ as input and outputs public parameters PP, an encryption key EK and a master secret key MSK. The pubic parameters PP is an implicit input to the rest of the algorithms.
- $\mathsf{Enc}(\mathsf{EK}, \mathbf{x}, \mathsf{ac\text{-}ct}) \to \mathsf{CT}_{\mathbf{x}, \mathsf{ac\text{-}ct}}$: It takes as input $\mathsf{ac\text{-}ct} \in \mathsf{AC\text{-}CT}$, the encryption key EK, a message vector $\mathbf{x} = (x_i)_{i \in I_\mathbf{x}} \in \mathcal{X}_\lambda$ and outputs a ciphertext $\mathsf{CT}_{\mathbf{x}, \mathsf{ac\text{-}ct}}$.
- $\mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y}, \mathsf{ac\text{-}k}) \to \mathsf{SK}_{\mathbf{y}, \mathsf{ac\text{-}k}}$: Given $\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}$, the master secret key MSK and a key vector $\mathbf{y} = (y_i)_{i \in I_\mathbf{y}} \in \mathcal{Y}_\lambda$, it outputs a secret key $\mathsf{SK}_{\mathbf{y}, \mathsf{ac\text{-}k}}$.
- $\mathsf{Dec}(\mathsf{SK}_{\mathbf{y}, \mathsf{ac\text{-}k}}, \mathsf{CT}_{\mathbf{x}, \mathsf{ac\text{-}ct}}) \to d / \perp$: It takes the secret key $\mathsf{SK}_{\mathbf{y}, \mathsf{ac\text{-}k}}$ and the ciphertext $\mathsf{CT}_{\mathbf{x}, \mathsf{ac\text{-}ct}}$ and outputs either a decrypted value d or a special symbol $\perp$ indicating failure.

Correctness. For all $\lambda \in \mathbb{N}$, $\mathbf{x} \in \mathcal{X}_\lambda$, $\mathbf{y} \in \mathcal{Y}_\lambda$, $\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}$, $\mathsf{ac\text{-}ct} \in \mathsf{AC\text{-}CT}$ and $(\mathbf{x}, \mathbf{y}) \in \mathcal{R}_p$ (or $\mathcal{R}_s$) satisfying $\mathcal{R}_{\mathsf{ac}}(\mathsf{ac\text{-}ct}, \mathsf{ac\text{-}k}) = 1$, the above scheme is correct if the following condition holds:

$$\Pr\left[d = \langle \mathbf{x}, \mathbf{y} \rangle : \begin{array}{l} (\mathsf{PP}, \mathsf{EK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{CT}_{\mathbf{x}, \mathsf{ac\text{-}ct}} \leftarrow \mathsf{Enc}(\mathsf{EK}, \mathbf{x}, \mathsf{ac\text{-}ct}) \\ \mathsf{SK}_{\mathbf{y}, \mathsf{ac\text{-}k}} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y}, \mathsf{ac\text{-}k}) \\ d \leftarrow \mathsf{Dec}(\mathsf{SK}_{\mathbf{y}, \mathsf{ac\text{-}k}}, \mathsf{CT}_{\mathbf{x}, \mathsf{ac\text{-}ct}}) \end{array} \right] = 1$$

Security. We define two security notions 1) full-hiding indistinguishability ($\mathsf{FH\text{-}IND}$) based security with semi-adaptive attributes — in this model, the adversary is allowed to query challenges to both the encryption and the key generation oracles, however, the adversary needs to submit the challenge attributes before submitting queries to the key generation oracle; 2) selective message-hiding indistinguishability ($\mathsf{MH\text{-}IND}$) based security which restricts the adversary to submit the challenge message pairs and the attribute at the start of the experiment.

Definition 4 (FH-IND security). An UAB-IPFE scheme $\mathcal{E} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ for $(\mathcal{F}_{\mathsf{ip}}, \mathcal{R}_{\mathsf{ac}})$ is said to satisfy $\mathsf{FH\text{-}IND}$ security if for any security parameter λ, any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that the following holds

$$\mathsf{Adv}^{\mathsf{FH\text{-}IND}}_{\mathcal{A}, \mathcal{E}}(\lambda) = \Big| \Pr\left[\mathsf{Expt}^{\mathsf{FH\text{-}IND}}_{0, \mathcal{A}, \mathcal{E}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{FH\text{-}IND}}_{1, \mathcal{A}, \mathcal{E}}(\lambda) = 1\right] \Big| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{FH\text{-}IND}}_{\beta, \mathcal{A}, \mathcal{E}}(\lambda)$ is defined for $\beta \in \{0, 1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{FH\text{-}IND}}_{\beta,\mathcal{A},\mathcal{E}}(\lambda)}$

1) $(\mathsf{PP}, \mathsf{EK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2) $\{(\mathsf{ac\text{-}ct}_i^{(0)}, \mathsf{ac\text{-}ct}_i^{(1)})\}_{i \in \mathsf{Q_{CT}}} \leftarrow \mathcal{A}(1^\lambda, \mathsf{PP})$.
3) $\beta' \leftarrow \mathcal{A}^{\mathsf{OKeyGen}_\beta(\mathsf{MSK},\cdot,\cdot),\mathsf{OEnc}_\beta(\mathsf{EK},\cdot,\cdot,\cdot)}(\mathsf{PP})$.
4) Outputs: β'.

The oracle $\mathsf{OKeyGen}_\beta(\mathsf{MSK},\cdot,\cdot)$ takes as input a key vector pair $(\mathbf{y}^{(0)}, \mathbf{y}^{(1)}) \in \mathcal{Y}_\lambda^2$ with the same index set $I_{\mathbf{y}}$ and $\mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K}$ and outputs the secret key $\mathsf{SK}_{\mathbf{y}^{(\beta)},\mathsf{ac\text{-}k}} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y}^{(\beta)}, \mathsf{ac\text{-}k})$. The oracle $\mathsf{OEnc}_\beta(\mathsf{MSK},\cdot,\cdot)$ takes as input a message vector pair $(\mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)}, i) \in \mathcal{X}_\lambda^2 \times [\mathsf{Q_{CT}}]$ such that $|I_{\mathbf{x}^{(0)}}| = |I_{\mathbf{x}^{(1)}}| = m_1$ and $(\mathsf{ac\text{-}ct}_i^{(0)}, \mathsf{ac\text{-}ct}_i^{(1)}) \in \mathsf{AC\text{-}CT}^2$ and outputs $\mathsf{CT}_{\mathbf{x}_i^{(\beta)},\mathsf{ac\text{-}ct}_i^{(\beta)}} \leftarrow \mathsf{Enc}(\mathsf{EK}, \mathbf{x}_i^{(\beta)}, \mathsf{ac\text{-}ct}_i^{(\beta)})$. Let $\mathsf{Q_{CT}}, \mathsf{Q_{SK}}$ be the total number of ciphertext and secret key queries of $\mathcal{A}$, then for all $i \in [\mathsf{Q_{CT}}]$ and $\ell \in [\mathsf{Q_{SK}}]$, we have $\langle \mathbf{x}_i^{(0)}, \mathbf{y}_\ell^{(0)} \rangle = \langle \mathbf{x}_i^{(1)}, \mathbf{y}_\ell^{(1)} \rangle$ when $\mathcal{R}_{\mathsf{ac}}(\mathsf{ac\text{-}ct}_i^{(0)}, \mathsf{ac\text{-}k}_\ell) = \mathcal{R}_{\mathsf{ac}}(\mathsf{ac\text{-}ct}_i^{(1)}, \mathsf{ac\text{-}k}_\ell) = 1$ with $(\mathbf{x}_i^{(0)}, \mathbf{y}_\ell^{(0)}), (\mathbf{x}_i^{(1)}, \mathbf{y}_\ell^{(1)}) \in \mathcal{R}_p$ (or $\mathcal{R}_s$).

Definition 5 (MH-IND Security). A UAB-IPFE scheme $\mathcal{E} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ for $(\mathcal{F}_{\mathsf{ip}}, \mathcal{R}_{\mathsf{ac}})$ is said to satisfy MH-IND security if for any security parameter λ, any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that the following holds

$$\mathsf{Adv}^{\mathsf{MH\text{-}IND}}_{\mathcal{A},\mathcal{E}}(\lambda) = \Big| \Pr\left[\mathsf{Expt}^{\mathsf{MH\text{-}IND}}_{0,\mathcal{A},\mathcal{E}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{MH\text{-}IND}}_{1,\mathcal{A},\mathcal{E}}(\lambda) = 1\right] \Big| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{MH\text{-}IND}}_{\beta,\mathcal{A},\mathcal{E}}(\lambda)$ is defined for $\beta \in \{0,1\}$ as follows:

$\underline{\mathsf{Expt}^{\mathsf{MH\text{-}IND}}_{\beta,\mathcal{A},\mathcal{E}}(\lambda)}$

1) $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathsf{ac\text{-}ct}) \leftarrow \mathcal{A}(1^\lambda)$ s.t $I_{\mathbf{x}^{(0)}} = I_{\mathbf{x}^{(1)}}$.
2) $(\mathsf{PP}, \mathsf{EK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$.
3) $\beta' \leftarrow \mathcal{A}^{\mathsf{OKeyGen}(\mathsf{MSK},\cdot,\cdot),\mathsf{OEnc}_\beta(\mathsf{EK},\cdot,\cdot)}(\mathsf{PP})$.
4) Outputs: β'.

In this experiment, $\mathsf{OKeyGen}(\mathsf{MSK},\cdot,\cdot)$ is an oracle that takes as input $(\mathbf{y} \in \mathcal{Y}_\lambda, \mathsf{ac\text{-}k} \in \mathsf{AC\text{-}K})$ with associated index set $I_{\mathbf{y}}$ of $\mathbf{y}$ and outputs the secret key $\mathsf{SK}_{\mathbf{y},\mathsf{ac\text{-}k}} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y}, \mathsf{ac\text{-}k})$. The oracle $\mathsf{OEnc}_\beta(\mathsf{EK},\cdot,\cdot)$, queried only once, takes as input a message vector pair $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}) \in \mathcal{X}_\lambda^2$ such that $|I_{\mathbf{x}^{(0)}}| = |I_{\mathbf{x}^{(1)}}| = m_1$ and outputs $\mathsf{CT}_{\mathbf{x}^{(\beta)},\mathsf{ac\text{-}ct}} \leftarrow \mathsf{Enc}(\mathsf{EK}, \mathbf{x}^{(\beta)}, \mathsf{ac\text{-}ct})$. For all queried $(\mathbf{y}, \mathsf{ac\text{-}k})$ pair satisfying $\mathcal{R}_{\mathsf{ac}}(\mathsf{ac\text{-}ct}, \mathsf{ac\text{-}k}) = 1$ and $(\mathbf{x}^{(\beta)}, \mathbf{y}) \in \mathcal{R}_p$ (or $\mathcal{R}_s$), we have $\langle \mathbf{x}^{(0)}, \mathbf{y} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle$.

Remark 1 (SACfe). For our concrete constructions, the encryption key is either a private key, i.e., $\mathsf{EK} = \mathsf{MSK}$, or a public key, i.e., $\mathsf{EK} = \mathsf{PP}$. We call a FH-IND secure UAB-IPFE scheme a SACfe. We emphasize that function-hiding security in IPFE can only be achieved in the setting of $\mathsf{EK} = \mathsf{MSK}$ due to the linear functionality as discussed in previous works [9], [12], [24]. In Section 4, we present a SACfe scheme (with $\mathsf{EK} = \mathsf{MSK}$) where AC-K and AC-CT represents sets of vectors over $\mathbb{Z}_p$ and the relation $\mathcal{R}_{\mathsf{ac}}(\mathbf{u} \in \mathsf{AC\text{-}CT}, \mathbf{v} \in \mathsf{AC\text{-}K})$ holds if and only if $\langle \mathbf{u}, \mathbf{v} \rangle = 0$ and $(\mathbf{u}, \mathbf{v}) \in \mathcal{R}_p$.
In our second construction of (strict) UAB-IPFE of Section 5, AC-K and AC-CT represent an LSSS access structure and a set of related attributes respectively, and the relation $\mathcal{R}_{\mathsf{ac}}(\mathbb{A} \in \mathsf{AC\text{-}K}, \mathsf{S} \in \mathsf{AC\text{-}CT})$ holds if and only if $\mathbb{A}(\mathsf{S}) = 1$.

4. SACfe: Function Hiding UZP-IPFE

This section presents our SACfe protocol which is a private key function-hiding UZP-IPFE scheme in the permissive setting based on the DPVS framework of [30].

4.1. Construction

Let $\mathcal{F}_1 := \{F_K\}_{K \in \mathcal{K}_\lambda}, \mathcal{F}_2 := \{F_{\widetilde{K}}\}_{\widetilde{K} \in \mathcal{K}_\lambda}$ be two PRF families with a key space $\mathcal{K}_\lambda$ consisting of functions $F_K : \mathbb{Z} \to \mathsf{GL}_4(\mathbb{Z}_p)$ and $F_{\widetilde{K}} : \mathbb{Z} \to \mathsf{GL}_3(\mathbb{Z}_p)$ respectively. As all pairing-based IPFE in the literature, our required inner product values come from a polynomial range so that at the end of the decryption phase, we can efficiently perform an exhaustive search to obtain the value. We present our UZP-IPFE scheme in Figure 2.

Correctness. The decryption succeeds if $(\mathbf{x}, \mathbf{y})$, $(\mathbf{w}, \mathbf{v}) \in \mathcal{R}_p$ and $\langle \mathbf{w}, \mathbf{v} \rangle = 0$ as shown below

$$\prod_{i \in I_{\mathbf{y}}} E\Big([\![\mathbf{c}_i^1]\!]_1, [\![\mathbf{k}_i^1]\!]_2\Big) = [\![\langle \mathbf{x}, \mathbf{y} \rangle + \alpha \sum_{i \in I_{\mathbf{y}}} \gamma_i]\!]_T.$$

$$\prod_{j \in I_{\mathbf{v}}} E\Big([\![\mathbf{c}_j^2]\!]_1, [\![\mathbf{k}_j^2]\!]_2\Big) = [\![\omega\delta(\langle \mathbf{w}, \mathbf{v} \rangle) + \alpha \sum_{j \in I_{\mathbf{v}}} \widetilde{\gamma}_j]\!]_T.$$

$$h = [\![\langle \mathbf{x}, \mathbf{y} \rangle + \omega\delta\langle \mathbf{w}, \mathbf{v} \rangle + \alpha(\sum_{i \in I_{\mathbf{y}}} \gamma_i + \sum_{j \in I_{\mathbf{v}}} \widetilde{\gamma}_j)]\!]_T = [\![\langle \mathbf{x}, \mathbf{y} \rangle + \omega\delta\langle \mathbf{w}, \mathbf{v} \rangle]\!]_T. \quad (1)$$

Using $\langle \mathbf{w}, \mathbf{v} \rangle = 0$, it can be seen that the correctness follows from Eq. (1).

4.2. Security

Theorem 1. Assuming the SXDH assumption holds over the bilinear group G, our UZP-IPFE scheme achieves FH-IND security as per Def. 4.

Proof of Theorem 1. Suppose $\mathcal{A}$ is a PPT adversary against FH-IND security of our UZP-IPFE scheme. We construct an algorithm $\mathcal{B}$ for breaking the SXDH assumption that uses $\mathcal{A}$ as a subroutine. We prove

$\mathsf{Setup}(1^\lambda) \to (\mathsf{PP}, \mathsf{MSK})$:
- Generate a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ and a pair of PRF keys $K, \widetilde{K} \leftarrow \mathcal{K}_\lambda$
- Output $\mathsf{PP} = \mathsf{G}, \mathsf{MSK} = (K, \widetilde{K})$

$\mathsf{Enc}(\mathsf{MSK}, \mathbf{x} = (x_i)_{i\in[m_1]} \in \mathbb{Z}^{m_1}, \mathbf{w} = (w_i)_{i\in[m_2]} \in \mathbb{Z}^{m_2}) \to \mathsf{CT}_{\mathbf{x},\mathbf{w}}$:
- Compute $F_K(i) = \mathbf{B}_i \ \forall i \in [m_1]$ and $F_{\widetilde{K}}(j) = \widetilde{\mathbf{B}}_j \ \forall j \in [m_2]$
- Sample $\delta, \alpha \leftarrow \mathbb{Z}_p$ and compute

$$[\![\mathbf{c}_i^1]\!]_1 = [\![(x_i, 0, \alpha, 0)\mathbf{B}_i]\!]_1 \ \forall i \in [m_1],$$
$$[\![\mathbf{c}_j^2]\!]_1 = [\![(\delta w_j, \alpha, 0)\widetilde{\mathbf{B}}_j]\!]_1 \ \ \forall j \in [m_2]$$

- Output $\mathsf{CT}_{\mathbf{x},\mathbf{w}} = (\{[\![\mathbf{c}_i^1]\!]_1\}_{i\in[m_1]}, \{[\![\mathbf{c}_j^2]\!]_1\}_{j\in[m_2]})$

$\mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y} = (y_i)_{i\in I_\mathbf{y}} \in \mathbb{Z}^{|I_\mathbf{y}|}, \mathbf{v} = (v_i)_{i\in I_\mathbf{v}} \in \mathbb{Z}^{|I_\mathbf{v}|}) \to \mathsf{SK}_{\mathbf{y},\mathbf{v}}$:
- Compute $F_K(i) = \mathbf{B}_i \ \forall i \in I_\mathbf{y}$ and $F_{\widetilde{K}}(j) = \widetilde{\mathbf{B}}_j \ \forall j \in I_\mathbf{v}$.
- Sample $\omega \leftarrow \mathbb{Z}_p$, $\gamma_i, \widetilde{\gamma}_j \leftarrow \mathbb{Z}_p$ for all $i \in I_\mathbf{y}$, $j \in I_\mathbf{v}$ such that $\sum_{i\in I_\mathbf{y}} \gamma_i + \sum_{j\in I_\mathbf{v}} \widetilde{\gamma}_j = 0$ and compute

$$\mathbf{k}_i^1 = (y_i, 0, \gamma_i, 0)\mathbf{B}_i^* \ \ \forall i \in I_\mathbf{y},$$
$$\mathbf{k}_j^2 = (\omega v_j, \widetilde{\gamma}_j, 0)\widetilde{\mathbf{B}}_j^* \ \ \forall j \in I_\mathbf{v}$$

- Output $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i\in I_\mathbf{y}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j\in I_\mathbf{v}}, I_\mathbf{y}, I_\mathbf{v})$

$\mathsf{Dec}(\mathsf{SK}_{\mathbf{y},\mathbf{v}}, \mathsf{CT}_{\mathbf{x},\mathbf{w}}) \to d/\perp$:
- If $(\mathbf{x}, \mathbf{y}) \notin \mathcal{R}_p$ or $(\mathbf{w}, \mathbf{v}) \notin \mathcal{R}_p$, output $\perp$ else compute

$$h = \prod_{i\in I_\mathbf{y}} \prod_{j\in I_\mathbf{v}} E\Big([\![\mathbf{c}_i^1]\!]_1, [\![\mathbf{k}_i^1]\!]_2\Big) \cdot E\Big([\![\mathbf{c}_j^2]\!]_1, [\![\mathbf{k}_j^2]\!]_2\Big)$$

- Output $d = \log_{e(g_1,g_2)} h$

Figure 2. Our SACfe: UZP-IPFE

Theorem 1 by a series of games. For each game transition, we calculate the difference of probabilities that $\mathcal{A}$ outputs 1 in the corresponding games. In every game, the challenger chooses a random element $m_1' \leftarrow [m_{1,\mathsf{max}}]$, as a guess of m_1^* at the beginning of the games. As we consider the semi-adaptive model here, we set $m_2^* = m_{2,\mathsf{max}}$ where $m_{1,\mathsf{max}}, m_{2,\mathsf{max}}$ as the maximum length of the challenge message and attribute vectors (i.e., $\mathbf{x}$ and $\mathbf{w}$) respectively and consider $s_{\mathsf{max}}, t_{\mathsf{max}}$ to be the maximum indices of the queried key and predicate vectors (i.e., $\mathbf{y}$ and $\mathbf{v}$) respectively to the key generation oracle. Here E_ι denotes the event that $\mathcal{A}$ outputs 1 in Game ι.

Game 0: This game is the same as the real security game where the challenge ciphertext $\mathsf{CT}_{k,\mathbf{x},\mathbf{w}}^{(0)}$ is the encryption of $(\mathbf{x}_k^{(0)}, \mathbf{w}_k^{(0)})$ as described in Def. 4 i.e.,

$$[\![\mathbf{c}_{k,i}^1]\!]_1 = [\![(x_{k,i}^{(0)}, 0, \alpha_k, 0)\mathbf{B}_i]\!]_1 \ \ \forall i \in [m_{1,k}^*],$$
$$[\![\mathbf{c}_{k,j}^2]\!]_1 = [\![(\delta_k w_{k,j}^{(0)}, \alpha_k, 0)\widetilde{\mathbf{B}}_j]\!]_1 \ \ \forall j \in [m_{2,k}^*]$$

with $\alpha_k \leftarrow \mathbb{Z}_p$ and $F_K(i) = \mathbf{B}_i, F_{\widetilde{K}}(j) = \widetilde{\mathbf{B}}_j$. The ℓ-th secret key $\mathsf{SK}_{\mathbf{y}_\ell,\mathbf{v}_\ell}$ for $\mathbf{y}_\ell^{(0)}, \mathbf{v}_\ell$ is replied as

$$[\![\mathbf{k}_{\ell,i}^1]\!]_2 = [\![(y_{\ell,i}^{(0)}, 0, \gamma_{\ell,i}, 0)\mathbf{B}_i^*]\!]_2 \ \ \forall i \in I_{\mathbf{y}_\ell},$$
$$[\![\mathbf{k}_{\ell,j}^2]\!]_2 = [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, 0)\widetilde{\mathbf{B}}_j^*]\!]_2 \ \ \forall j \in I_{\mathbf{v}_\ell}$$

where $\gamma_{\ell,i}, \widetilde{\gamma}_{\ell,j}, \omega_\ell \leftarrow \mathbb{Z}_p$ satisfying $\sum_{i\in I_{\mathbf{y}_\ell}} \gamma_{\ell,i} + \sum_{j\in I_{\mathbf{v}_\ell}} \widetilde{\gamma}_{\ell,j} = 0$.

Game $0'$: Game $0'$ is the same as Game 0, except we use $\mathbf{B}_i \leftarrow \mathsf{GL}_4(\mathbb{Z}_p)$, $\widetilde{\mathbf{B}}_j \leftarrow \mathsf{GL}_3(\mathbb{Z}_p)$ to generate ciphertext and secret key components. Note that, Game 1-0-3 $\equiv$ Game $0'$.

Game (1-μ-1): For $\mu \in [\mathsf{Q}_\mathsf{SK}]$, same as Game 1-$(\mu-1)$-3 except for the following components

$$[\![\mathbf{k}_{\mu,i}^1]\!]_2 = [\![(y_{\mu,i}^{(0)}, 0, \gamma_{\mu,i}, \boxed{\eta\gamma_{\mu,i}})\mathbf{B}_i^*]\!]_2 \ \ \forall i \in I_{\mathbf{y}_\mu},$$
$$[\![\mathbf{k}_{\mu,j}^2]\!]_2 = [\![(\omega_\mu v_{\mu,j}, \widetilde{\gamma}_{\mu,j}, \boxed{\eta\widetilde{\gamma}_{\mu,j}})\widetilde{\mathbf{B}}_j^*]\!]_2 \ \ \forall j \in I_{\mathbf{v}_\mu}$$

where $\eta \leftarrow \mathbb{Z}_p$. Note that Game 1-0-3 $\equiv$ Game $0'$.

Game (1-μ-2): For $\mu \in [\mathsf{Q}_\mathsf{SK}]$, same as Game 1-$\mu$-1 except for the following components

$$[\![\mathbf{k}_{\mu,i}^1]\!]_2 = [\![(y_{\mu,i}^{(0)}, \boxed{y_{\mu,i}^{(1)}}, \gamma_{\mu,i}, \eta\gamma_{\mu,i})\mathbf{B}_i^*]\!]_2 \ \ \forall i \in I_{\mathbf{y}_\mu}$$
$$[\![\mathbf{k}_{\mu,j}^2]\!]_2 = [\![(\omega_\mu v_{\mu,j}, \widetilde{\gamma}_{\mu,j}, \eta\widetilde{\gamma}_{\mu,j})\widetilde{\mathbf{B}}_j^*]\!]_2 \ \ \forall j \in I_{\mathbf{v}_\mu}.$$

Game (1-μ-3): For $\mu \in [\mathsf{Q}_\mathsf{SK}]$, same as Game 1-$\mu$-2 except for the following components

$$[\![\mathbf{k}_{\mu,i}^1]\!]_2 = [\![(y_{\mu,i}^{(0)}, y_{\mu,i}^{(1)}, \gamma_{\mu,i}, \boxed{0})\mathbf{B}_i^*]\!]_2 \ \ \forall i \in I_{\mathbf{y}_\mu},$$
$$[\![\mathbf{k}_{\mu,j}^2]\!]_2 = [\![(\omega_\mu v_{\mu,j}, \widetilde{\gamma}_{\mu,j}, \boxed{0})\widetilde{\mathbf{B}}_j^*]\!]_2 \ \ \forall j \in I_{\mathbf{v}_\mu}.$$

Game 2: Same as Game 1-Q_SK-3 except for the following components

$$[\![\mathbf{k}_{\ell,i}^1]\!]_2 = [\![(y_{\ell,i}^{(0)}, y_{\ell,i}^{(1)}, \gamma_{\ell,i}, \boxed{s_{\ell,i}})\mathbf{B}_i^*]\!]_2 \ \ \forall i \in I_{\mathbf{y}_\ell},$$
$$[\![\mathbf{k}_{\ell,j}^2]\!]_2 = [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{t_{\ell,j}})\widetilde{\mathbf{B}}_j^*]\!]_2 \ \ \forall j \in I_{\mathbf{v}_\ell}$$

where $s_{\ell,i}, t_{\ell,j} \leftarrow \mathbb{Z}_p : \sum_{i\in I_{\mathbf{y}_\ell}} s_{\ell,i} + \sum_{j\in I_{\mathbf{v}_\ell}} t_{\ell,j} = 0$.

Game (3-ν-1): For $\nu \in [\mathsf{Q}_\mathsf{CT}]$, same as Game 3-$(\nu-1)$-3 except for the following components

$$[\![\mathbf{c}_{\nu,i}^1]\!]_1 = [\![(x_{\nu,i}^{(0)}, 0, \alpha_\nu, \boxed{\widehat{\alpha}_\nu})\mathbf{B}_i]\!]_1 \ \ \forall i \in [m_{1,\nu}^*],$$
$$[\![\mathbf{c}_{\nu,j}^2]\!]_1 = [\![(\delta_\nu w_{\nu,j}^{(0)}, \alpha_\nu, \boxed{\widehat{\alpha}_\nu})\widetilde{\mathbf{B}}_j]\!]_1 \ \ \forall j \in [m_{2,\nu}^*]$$

where $\alpha_\nu, \widehat{\alpha}_\nu \leftarrow \mathbb{Z}_p$. Note that Game 3-0-3 $\equiv$ Game 2.

Game (3-ν-2): For $\nu \in [\mathsf{Q}_\mathsf{CT}]$, same as Game 3-$\nu$-1 except for the following components

$$[\![\mathbf{c}_{\nu,i}^1]\!]_1 = [\![(\boxed{0, x_{\nu,i}^{(1)}}, \alpha_\nu, \widehat{\alpha}_\nu)\mathbf{B}_i]\!]_1 \ \ \forall i \in [m_{1,\nu}^*],$$
$$[\![\mathbf{c}_{\nu,j}^2]\!]_1 = [\![(\boxed{\delta_\nu w_{\nu,j}^{(1)}}, \alpha_\nu, \widehat{\alpha}_\nu)\widetilde{\mathbf{B}}_j]\!]_1 \ \ \forall j \in [m_{2,\nu}^*].$$

Game (3-ν-3): For $\nu \in [\mathsf{Q_{CT}}]$, same as Game 3-$\nu$-2 except for the following components

$$[\![\mathbf{c}^1_{\nu,i}]\!]_1 = [\![(0, x^{(1)}_{\nu,i}, \alpha_\nu, \boxed{0})\mathbf{B}_i]\!]_1 \quad \forall i \in [m^*_{1,\nu}],$$
$$[\![\mathbf{c}^2_j]\!]_1 = [\![(\delta_\nu w^{(1)}_{\nu,j}, \alpha_\nu, \boxed{0})\widetilde{\mathbf{B}}_j]\!]_1 \quad \forall j \in [m^*_{2,\nu}].$$

Game 4: Same as Game 3-$\mathsf{Q_{CT}}$-3 except for the k-th ciphertext and ℓ-th secret key component as follows

$$[\![\mathbf{c}^1_{k,i}]\!]_1 = [\![(\boxed{x^{(1)}_{k,i}, 0}, \alpha_k, 0)\mathbf{B}_i]\!]_1 \quad \forall i \in [m^*_{1,k}],$$
$$[\![\mathbf{k}^1_{\ell,i}]\!]_2 = [\![(\boxed{y^{(1)}_{\ell,i}, y^{(0)}_{\ell,i}}, \gamma_{\ell,i}, s_{\ell,i})\mathbf{B}^*_i]\!]_2 \quad \forall i \in I_{\mathbf{y}_\ell}.$$

Game 5: Same as Game 4 except for the k-th ciphertext and ℓ-th secret key components as follows

$$[\![\mathbf{c}^1_{k,i}]\!]_1 = [\![(x^{(1)}_{k,i}, 0, \alpha_k, 0)\boxed{\mathbf{B}_i}]\!]_1 \quad \forall i \in [m^*_{1,k}],$$
$$[\![\mathbf{c}^2_{k,j}]\!]_1 = [\![(\delta_k w^{(1)}_{k,j}, \alpha_k, 0)\boxed{\widetilde{\mathbf{B}}_j}]\!]_1 \quad \forall j \in [m^*_{2,k}]$$
$$[\![\mathbf{k}^1_{\ell,i}]\!]_2 = [\![(y^{(1)}_{\ell,i}, \boxed{0}, \gamma_{\ell,i}, \boxed{0})\boxed{\mathbf{B}^*_i}]\!]_2 \quad \forall i \in I_{\mathbf{y}_\ell},$$
$$[\![\mathbf{k}^2_{\ell,j}]\!]_2 = [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{0})\boxed{\widetilde{\mathbf{B}}^*_j}]\!]_2 \quad \forall j \in I_{\mathbf{v}_\ell}$$

where $F_K(i) = \mathbf{B}_i, F_{\widetilde{K}}(j) = \widetilde{\mathbf{B}}_j$.

We prove the indistinguishability arguments of the above games in Appendix A. For each game transition, we prove that the difference between probabilities that the adversary A outputs 1 in both games is negligible. Combining the arguments, the above Theorem follows.

5. Strict UAB-IPFE

In this section, we construct a strict public key UAB-IPFE scheme using the DPVS framework.

5.1. Construction

Our $\mathsf{UAB\text{-}IPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ scheme can be described in terms of the following algorithms. Let two full-domain hash functions $\mathcal{H}_1, \mathcal{H}_2$ into $\mathbb{G}_2$. As with all pairing-based IPFE in the literature, our required inner product values come from a polynomial range so that at the end of the decryption phase, we can efficiently perform an exhaustive search to obtain the value. We present our UAB-IPFE in Figure 3.

Correctness: If $(\mathbb{A}(\mathsf{S}) = 0) \vee ((\mathbf{x}, \mathbf{y}) \notin \mathcal{R}_s)$, it outputs $\perp$, otherwise it computes

$$\prod_{j \in A} E([\![\mathbf{c}_{\mathsf{ac},j}]\!]_1, c_j[\![\mathbf{k}_{\mathsf{ac},j}]\!]_2) = [\![\psi z a_0]\!]_T \quad (2)$$
$$E([\![\mathbf{c}_{\mathsf{fe}}]\!]_1, [\![\mathbf{k}_{\mathsf{fe}}]\!]_2) = [\![a_0 z\psi - \omega \sum_{i \in I_{\mathbf{y}}} y_i(u_i + \mu s_i)]\!]_T \quad (3)$$
$$\prod_{i \in I_{\mathbf{y}}} y_i[\![t_i]\!]_T = [\![\sum_{i \in I_{\mathbf{y}}} (x_i y_i - \omega u_i y_i - \omega \mu s_i y_i)]\!]_T \quad (4)$$

From Equ. (4),(3) and (2), it outputs $[\![\langle \mathbf{x}, \mathbf{y} \rangle]\!]_T$.

$\mathsf{Setup}(1^\lambda) \to (\mathsf{PP}, \mathsf{MSK})$:
- Generate a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$.
- Sample $z, \mu \leftarrow \mathbb{Z}_p$, $\mathbf{F} \leftarrow \mathsf{GL}_8(\mathbb{Z}_p)$, $\mathbf{H} \leftarrow \mathsf{GL}_4(\mathbb{Z}_p)$ and output

$$\mathsf{PP} = (\{[\![\mathbf{f}_i]\!]_1\}_{i=1}^3, [\![\mathbf{h}_1 + \mu\mathbf{h}_2]\!]_1, [\![\mathbf{h}_3]\!]_1, [\![\mu]\!]_1)$$
$$\mathsf{MSK} = (\{\mathbf{f}^*_i\}_{i=1}^3, \{\mathbf{h}^*_i\}_{i=1}^3, z)$$

$\mathsf{Enc}(\mathsf{PP}, \mathbf{x} = (x_i)_{i \in I_{\mathbf{x}}} \in \mathbb{Z}_p^{|I_{\mathbf{x}}|}, \mathsf{S}) \to \mathsf{CT}_{\mathbf{x},\mathsf{S}}$:
- Compute $\mathcal{H}_1(i|I_{\mathbf{x}}) = [\![u_i]\!]_2$, $\mathcal{H}_2(i|I_{\mathbf{x}}) = [\![s_i]\!]_2 \; \forall i \in I_{\mathbf{x}}$
- Sample $\psi, \omega, \sigma_j \leftarrow \mathbb{Z}_p \; \forall \; j \in \mathsf{S}$ and compute

$$[\![\mathbf{c}_{\mathsf{ac},j}]\!]_1 = [\![(\sigma_j(1, -j), \psi, 0, 0, 0, 0, 0)\,\mathbf{F}]\!]_1$$
$$[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 = [\![(\omega, \mu\omega, \psi, 0)\mathbf{H}]\!]_1$$
$$[\![t_i]\!]_T = [\![x_i]\!]_T \cdot e\,(g_1, \omega[\![u_i]\!]_2)^{-1} \cdot e([\![\mu]\!]_1, \omega[\![s_i]\!]_2)^{-1}$$

- Output $\mathsf{CT}_{\mathbf{x},\mathsf{S}} = (\{[\![\mathbf{c}_{\mathsf{ac},j}]\!]_1\}_{j \in \mathsf{S}}, [\![\mathbf{c}_{\mathsf{fe}}]\!]_1, \{[\![t_i]\!]_T\}_{i \in I_{\mathbf{x}}})$

$\mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y} = (y_i)_{i \in I_{\mathbf{y}}} \in \mathbb{Z}_p^{|I_{\mathbf{y}}|}, \mathbb{A}) \to \mathsf{SK}_{\mathbf{y},\mathbb{A}}$:
- Sample $a_0 \leftarrow \mathbb{Z}_p$, use the secret sharing scheme based on $\mathbb{A}$ to create the shares $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}$ of a_0 as defined in Section 3.4.
- Compute $\mathcal{H}_1(i|I_{\mathbf{y}}) = [\![u_i]\!]_2$, and $\mathcal{H}_2(i|I_{\mathbf{y}}) = [\![s_i]\!]_2$ for all $i \in I_{\mathbf{y}}$
- Sample $\pi_j \leftarrow \mathbb{Z}_p$ for all $j \in \mathsf{List\text{-}Att}(\mathbb{A})$ and compute

$$[\![\mathbf{k}_{\mathsf{ac},j}]\!]_2 = [\![(\pi_j(j, 1), a_j z, 0, 0, 0, 0, 0)\,\mathbf{F}^*]\!]_2$$
$$[\![\mathbf{k}_{\mathsf{fe}}]\!]_2 = [\![(-\sum_{i \in I_{\mathbf{y}}} y_i u_i, -\sum_{i \in I_{\mathbf{y}}} y_i s_i, a_0 z, 0)\mathbf{H}^*]\!]_2$$

- Output $\mathsf{SK}_{\mathbf{y},\mathbb{A}} = (\{[\![\mathbf{k}_{\mathsf{ac},j}]\!]_2\}_{j \in \mathsf{List\text{-}Att}(\mathbb{A})}, [\![\mathbf{k}_{\mathsf{fe}}]\!]_2)$

$\mathsf{Dec}(\mathsf{SK}_{\mathbf{y},\mathbb{A}}, \mathsf{CT}_{\mathbf{x},\mathsf{S}}) \to d/\perp$:
- If $(\mathbf{x}, \mathbf{y}) \notin \mathcal{R}_s \vee \mathbb{A}(\mathsf{S}) = 0$, output $\perp$
- Else, there exists $A \subseteq \mathsf{S}$ and $A \in \mathbb{A}$, then find the reconstruction vector $\mathbf{c} = (c_j)_j$ of the LSSS for A and compute the following

$$h = \prod_{i \in I_{\mathbf{y}}} y_i[\![t_i]\!]_T \cdot E([\![\mathbf{c}_{\mathsf{fe}}]\!]_1, [\![\mathbf{k}_{\mathsf{fe}}]\!]_2)^{-1} \cdot \prod_{j \in A} E([\![\mathbf{c}_{\mathsf{ac},j}]\!]_1, c_j[\![\mathbf{k}_{\mathsf{ac},j}]\!]_2)$$

Finally, output $d = \log_{e(g_1,g_2)} h$

Figure 3. Our UAB-IPFE

5.2. Security

Theorem 2. Assuming the SXDH, DBDH assumptions hold over the bilinear group G, then our UAB-IPFE scheme achieves MH-IND security in the random oracle model as per Def. 5.

Proof of Theorem 2. Suppose $\mathcal{A}$ is a PPT adversary against MH-IND security of our UAB-IPFE scheme. We construct an algorithm $\mathcal{B}$ for breaking the SXDH and DBDH assumptions that uses $\mathcal{A}$ as a subroutine.

We prove Theorem 2 by a series of games. For each game transition, we calculate the difference of probabilities that $\mathcal{A}$ outputs 1 in the corresponding games. We represent E_ι as the event that $\mathcal{A}$ outputs 1 in Game ι.

Game 0**:** This game is the same as the real security game as presented in Def. 5 where the challenge ciphertext and the ℓ-th secret key components are given below:

$$\begin{aligned}
[\![\mathbf{c}_{\mathsf{ac},j}]\!]_1 &= [\![(\sigma_j(1,-j),\psi,0,0,0,0,0)\mathbf{F}]\!]_1\\
[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 &= [\![(\omega,\mu\omega,\psi,0)\mathbf{H}]\!]_1\\
[\![t_i^{(\beta)}]\!]_T &= [\![x_i^{(\beta)}]\!]_T \cdot e(g_1,\omega[\![u_i]\!]_2)^{-1} \cdot e([\![\mu]\!]_1,\omega[\![s_i]\!]_2)^{-1}\\
[\![\mathbf{k}_{\ell,\mathsf{ac},j}]\!]_2 &= [\![\pi_{\ell,j}(j,1),a_{\ell,j}z,0,0,0,0,0)\mathbf{F}^*]\!]_2\\
[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 &= [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}, a_{\ell,0}z,0)\mathbf{H}^*]\!]_2
\end{aligned}$$

Here, $a_{\ell,0} \leftarrow \mathbb{Z}_p$, $(a_{\ell,j})_{j\in\mathsf{List\text{-}att}(\mathbb{A})} \leftarrow \Lambda_{a_{\ell,0}}(\mathbb{A})$.

Game 1: Same as Game 0 except for the following secret key component whenever $I_{\mathbf{x}} = I_{\mathbf{y}_\ell}$

$$\begin{aligned}
[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 &= [\![(\omega,\mu\omega,\psi,\boxed{\tau})\mathbf{H}]\!]_1\\
[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 &= [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}, a_{\ell,0}z, \boxed{r'_{\ell,0}\delta_\ell})\mathbf{H}^*]\!]_2
\end{aligned}$$

where $\delta_\ell = \langle \Delta\mathbf{x}, \mathbf{y}_\ell\rangle, \tau, r'_{\ell,0} \leftarrow \mathbb{Z}_p$, $\Delta\mathbf{x} := \mathbf{x}^{(0)} - \mathbf{x}^{(1)}$. Others secret keys for $I_{\mathbf{x}} \neq I_{\mathbf{y}_\ell}$ remain unaltered.
Indistinguishability follows between Game 0 and Game 1 from the SXDH assumption.

Game 2: Same as Game 1 except for the following components

$$\begin{aligned}
[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 &= [\![(\omega,\boxed{\omega},\psi,\tau)\mathbf{H}]\!]_1\\
[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 &= [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, \boxed{-\mu\textstyle\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}},\\
&\qquad a_{\ell,0}z, r'_{\ell,0}\delta_\ell)\mathbf{H}^*]\!]_2
\end{aligned}$$

Game 3: Same as Game 2 except for the following components

$$\begin{aligned}
[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 = [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, &\boxed{-\mu\textstyle\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i} + r'_{\ell,0}\delta_\ell},\\
&a_{\ell,0}z, r'_{\ell,0}\delta_\ell)\mathbf{H}^*]\!]_2
\end{aligned}$$

Game 4: Same as Game 3 except for the following ciphertext component

$$\begin{aligned}
[\![t_i^{(\beta)}]\!]_T = [\![x_i^{(\beta)}]\!]_T \cdot \boxed{\alpha_{m(i)}[\![d]\!]_T} \cdot e(g_1, \boxed{\omega[\![u'_i]\!]_2})^{-1}\\
\cdot e([\![\mu]\!]_1, \boxed{\omega[\![s'_i]\!]_2})^{-1}
\end{aligned}$$

$$[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 = \begin{cases}
[\![(\boxed{-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u'_{\ell,i}}, \boxed{-\mu\textstyle\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s'_{\ell,i}},\\
\qquad a_{\ell,0}z, r'_{\ell,0}\delta_\ell)\mathbf{H}^*]\!]_2 \text{ if } I_{\mathbf{x}} = I_{\mathbf{y}_\ell}\\
[\![(\boxed{-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u'_{\ell,i}}, \boxed{-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s'_{\ell,i}},\\
\qquad a_{\ell,0}z, 0)\mathbf{H}^*]\!]_2 \text{ if } I_{\mathbf{x}} \neq I_{\mathbf{y}_\ell}
\end{cases}$$

where $d \leftarrow \mathbb{Z}_p$. Here the random oracles $\mathcal{H}_1, \mathcal{H}_2$ are programmed for $i \in I_{\mathbf{x}}$ as follows:

$$\begin{aligned}
\mathcal{H}_1(i|I_{\mathbf{x}}) = [\![u'_{\ell,i}]\!]_2 &:= \alpha_{m(i)}[\![a]\!]_2 \cdot \prod_{\kappa\in[n-1]} \lambda_{m(i),\kappa}[\![\rho_\kappa]\!]_2\\
\mathcal{H}_2(i|I_{\mathbf{x}}) = [\![s'_{\ell,i}]\!]_2 &:= \alpha_{m(i)}[\![c]\!]_2 \cdot \prod_{\kappa\in[n-1]} \lambda_{m(i),\kappa}[\![\rho_\kappa]\!]_2
\end{aligned}$$

where $\alpha_i, \lambda_{i,j} \in \mathbb{Z}_p, a, c \leftarrow \mathbb{Z}_p$ and $\rho_\kappa \leftarrow \mathbb{Z}_p$ with $m : I_{\mathbf{x}} \to [n]$ such that $|I_{\mathbf{x}}| = n$. Otherwise for $i \notin I_{\mathbf{x}}$, but $i \in I_{\mathbf{y}_\ell}$, oracles output $[\![u'_{\ell,i}]\!]_2, [\![s'_{\ell,i}]\!]_2 \leftarrow \mathbb{G}_2$.

We show that the adversary's advantage in this game is negligible relying on the DBDH assumption. We provide detailed analysis of the indistinguishability between each consecutive games in Appendix B.

6. Implementations and Evaluations

We report the ciphertext and secret key sizes of our schemes in Table 3 which also provides the comparison with the recent work of [15]. It shows that the ciphertext and secret key sizes are much smaller in SACfe compared to [15] and these improvements are achieved with the additional property of function hiding. We implemented both schemes using the CiFEr cryptographic library [27]. This library uses the GNU GMP library [2] to represent arbitrary big numbers. For the pairing CiFEr uses the Apache Milagro Cryptographic Library (AMCL) [1] configured with the BN254 curve. All benchmarks were conducted on a MacBook M1 with 16 GB of RAM. The heap consumption was evaluated using the tool valgrind. The code of the implementation is available at https://anonymous.4open.science/r/ipfe-impl-FFEF/README.md.

UZP-IPFE. We evaluated the performance of the UZP-IPFE scheme in terms of execution time and memory consumption for various values of m_1, while keeping m_2 constant at 1000, which means we used 1000 attributes. Figure 4 shows the execution times for encryption, key generation, and decryption. We observe that encryption was faster than key generation, which is expected because key generation requires sampling γ_i and $\tilde{\gamma}_j$ such that their sum is zero. Decryption was the slowest operation, likely due to the pairing computations involved. Compared to the scheme in [15], our UZP-IPFE scheme achieved faster execution times, as shown in Figure 5. The ciphertext and FE key sizes increased linearly with m_1, as shown in Table 4. This is consistent with our analysis, since each additional element in $\mathbf{x}$ adds one element in $\mathbb{G}_1$ to the ciphertext, and each additional element in $\mathbf{y}$ adds one element in $\mathbb{G}_2$ to the FE key. Table 4 also shows that our UZP-IPFE scheme used less memory than the scheme in [15].

UAB-IPFE. We implemented the UAB-IPFE scheme using monotone access programs and linear secret sharing scheme to define the access structure. The

TABLE 3. COMPARISON WITH EXISTING UAB-IPFE

Work	Scheme	|CT|	|SK|	Function Hiding	Assumption
[15]	UZP-IPFE	$7(m_1+m_2)\|\mathbb{G}_1\|$	$7(n_1+n_2)\|\mathbb{G}_2\|$	×	SXDH
Ours	SACfe: UZP-IPFE	$(4m_1+3m_2)\|\mathbb{G}_1\|$	$(4n_1+3n_2)\|\mathbb{G}_2\|$	✓	SXDH
	UAB-IPFE	$m_1\|\mathbb{G}_T\|+(8\|\mathsf{S}\|+4)\|\mathbb{G}_1\|$	$(8\|\mathsf{List\text{-}Att}(\mathbb{A})\|+4)\|\mathbb{G}_2\|$	×	SXDH, DBDH

– m_1, m_2: the lengths of the message and attribute vectors associated with the ciphertext.
– n_1, n_2: the lengths of the vectors associated with the secret key.
– $|\mathsf{CT}|$, $|\mathsf{SK}|$: the size of the ciphertext and the secret key, respectively.
– $|\mathsf{List\text{-}Att}(\mathbb{A})|$, $|\mathsf{S}|$: number of attributes associated with secret key and ciphertext respectively.
– SXDH, bi-k-Lin, DBDH: symmetric external Diffie-Hellman (or 1-Lin), bilateral k-Lin, decisional bilinear Diffie-Hellman.

TABLE 4. UZP-IPFE: SIZE OF THE CIPHER TEXT AND FE KEY FOR DIFFERENT SIZES OF m_1.

	m_1	1	5	10	50	10^2	$5\cdot10^2$	10^3	$5\cdot10^3$	10^4
[15]	Cipher text	919.9 kB	923.9 kB	928.5 kB	976.8 kB	1,020.6 kB	1,377.3 kB	1,845.3 kB	5,589.3 kB	10,169.3 kB
	FE key	1,809.3 kB	1,809.5 kB	1,908.7 kB	1,956.1 kB	2,102.9 kB	2,877.3 kB	3,845.3 kB	11,389.3 kB	20,969.3 kB
Ours	Cipher text	448.6 kB	451.0 kB	454.0 kB	477.6 kB	507.2 kB	744.0 kB	1,040.0 kB	3,408.0 kB	6,368.0 kB
	FE key	937.6 kB	942.5 kB	948.6 kB	997.5 kB	1,058.7 kB	1,548.3 kB	2,160.3 kB	7,056.3 kB	13,176.3 kB

TABLE 5. UAB-IPFE: SIZE OF THE CIPHER TEXT AND FE KEY FOR DIFFERENT SIZES OF m_1.

m_1	1	5	10	50	10^2	$5\cdot10^2$	10^3	$5\cdot10^3$	10^4
Cipher text	8.5 kB	11.0 kB	14.1 kB	38.9 kB	69.7 kB	317.2 kB	625.2 kB	310.1 kB	618.5 kB
FE key	11.4 kB	11.5 kB	11.6 kB	12.4 kB	13.3 kB	19.5	27.7	35.8	44.2

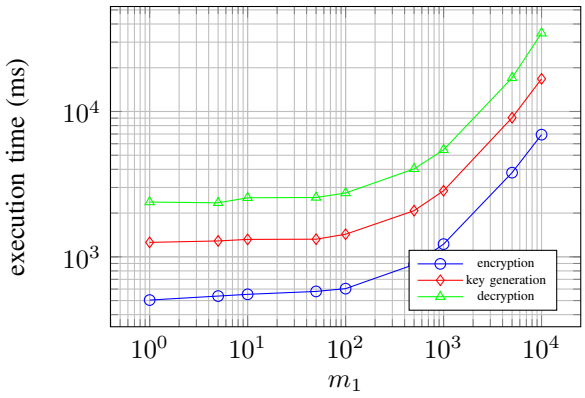

Figure 4. UZP-IPFE: Execution time of Enc, KeyGen, Dec for different sizes of m_1.

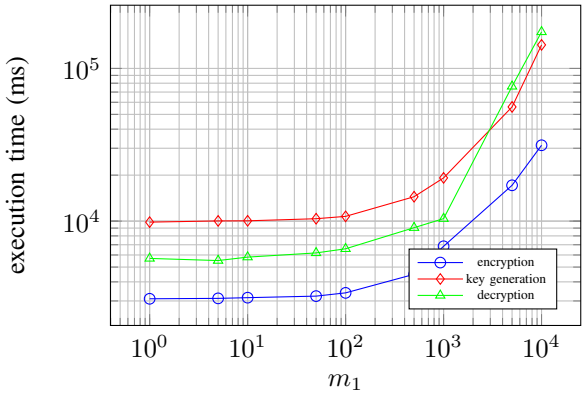

Figure 5. UZP-IPFE [15]: Execution time of Enc, KeyGen, Dec for different sizes of m_1.

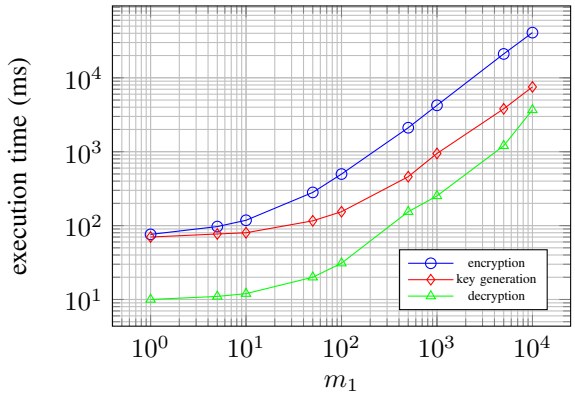

Figure 6. UAB-IPFE: Execution time of Enc, KeyGen, Dec for different sizes of m_1.

access policy can be specified with attributes that are connected by **OR** and **AND** clauses. We evaluated the performance of the scheme in terms of execution time and memory consumption for different values of m_1, while keeping the access policy fixed with six attributes. Figure 6 shows the execution times for encryption, decryption, and key generation. Encryption is the most costly operation, while decryption is the least. Table 5 shows the sizes of the ciphertext and the FE key for different values of m_1. The ciphertext size increases with m_1, as each additional element in $\mathbf{x}$ adds one more group element in $\mathbb{G}_T$ to the ciphertext. The FE key size should be constant in this experiment, as it only depends on the number of attributes and not on the size of the vector $\mathbf{y}$. However, our measurement was not accurate enough

to confirm this. The measured values are given in Table 5.

Acknowledgements

This work was partially supported by the SNSF grant IZSEZ0_220423 titled "Multi-client Attribute-Based Inner Product Functional Encryption" and the ICT small project grant 23080 supported by the Hasler foundation.

References

[1] Apache milagro crypto library:. https://github.com/miracl/amcl.

[2] The gnu multiple precision arithmetic library:. https://gmplib.org.

[3] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, volume 9020 of *Lecture Notes in Computer Science*, pages 733–751. Springer, 2015.

[4] Michel Abdalla, Dario Catalano, Romain Gay, and Bogdan Ursu. Inner-product functional encryption with fine-grained access control. *IACR Cryptol. ePrint Arch.*, page 577, 2020.

[5] Michel Abdalla, Junqing Gong, and Hoeteck Wee. Functional encryption for attribute-weighted sums from k-lin. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 685–716. Springer, 2020.

[6] Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-party functional encryption. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part II*, volume 13043 of *Lecture Notes in Computer Science*, pages 224–255. Springer, 2021.

[7] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 333–362. Springer, 2016.

[8] Carmen Elisabetta Zaira Baltico, Dario Catalano, Dario Fiore, and Romain Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 67–98. Springer, 2017.

[9] Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I*, volume 9452 of *Lecture Notes in Computer Science*, pages 470–491. Springer, 2015.

[10] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, 2007.

[11] Yansong Chang, Kai Zhang, Junqing Gong, and Haifeng Qian. Privacy-preserving federated learning via functional encryption, revisited. *IEEE Trans. Inf. Forensics Secur.*, 18:1855–1869, 2023.

[12] Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Functional encryption for inner product with full function privacy. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part I*, volume 9614 of *Lecture Notes in Computer Science*, pages 164–195. Springer, 2016.

[13] Pratish Datta and Tapas Pal. Decentralized multi-authority attribute-based inner-product FE: large universe and unbounded. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography - PKC 2023 - 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7-10, 2023, Proceedings, Part I*, volume 13940 of *Lecture Notes in Computer Science*, pages 587–621. Springer, 2023.

[14] Pratish Datta, Tapas Pal, and Katsuyuki Takashima. Compact FE for unbounded attribute-weighted sums for logspace from SXDH. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part I*, volume 13791 of *Lecture Notes in Computer Science*, pages 126–159. Springer, 2022.

[15] Uddipana Dowerah, Subhranil Dutta, Aikaterini Mitrokotsa, Sayantan Mukherjee, and Tapas Pal. Unbounded predicate inner product functional encryption from pairings. *J. Cryptol.*, 36(3):29, 2023.

[16] Edouard Dufour-Sans and David Pointcheval. Unbounded inner-product functional encryption with succinct keys. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *Applied Cryptography and Network Security - 17th International Conference, ACNS 2019, Bogota, Colombia, June 5-7, 2019, Proceedings*, volume 11464 of *Lecture Notes in Computer Science*, pages 426–441. Springer, 2019.

[17] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016.

[18] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013.

[19] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 545–554. ACM, 2013.

[20] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 503–523. Springer, 2015.

[21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 60–73. ACM, 2021.

[22] Mauricio Karchmer and Avi Wigderson. On span programs. In *Proceedings of the Eigth Annual Structure in Complexity Theory Conference, San Diego, CA, USA, May 18-21, 1993*, pages 102–111. IEEE Computer Society, 1993.

[23] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2008.

[24] Sam Kim, Kevin Lewi, Avradip Mandal, Hart Montgomery, Arnab Roy, and David J. Wu. Function-hiding inner product encryption is practical. In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 544–562. Springer, 2018.

[25] Qiqi Lai, Feng-Hao Liu, and Zhedong Wang. New lattice two-stage sampling technique and its applications to functional encryption - stronger security and smaller ciphertexts. *IACR Cryptol. ePrint Arch.*, page 779, 2022.

[26] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. *IACR Cryptol. ePrint Arch.*, page 110, 2010.

[27] Tilen Marc, Miha Stopar, Jan Hartman, Manca Bizjak, and Jolanda Modic. Privacy-enhanced machine learning with functional encryption. In Kazue Sako, Steve A. Schneider, and Peter Y. A. Ryan, editors, *Computer Security - ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part I*, volume 11735 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 2019.

[28] Ky Nguyen, Duong Hieu Phan, and David Pointcheval. Multi-client functional encryption with fine-grained access control. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part I*, volume 13791 of *Lecture Notes in Computer Science*, pages 95–125. Springer, 2022.

[29] Ky Nguyen, David Pointcheval, and Robert Schädlich. Function-hiding decentralized multi-client functional encryption for inner products. *IACR Cryptol. ePrint Arch.*, page 1532, 2022.

[30] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 191–208. Springer, 2010.

[31] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 349–366. Springer, 2012.

[32] Tapas Pal and Ratna Dutta. CCA secure attribute-hiding inner product encryption from minimal assumption. In Joonsang Baek and Sushmita Ruj, editors, *Information Security and Privacy - 26th Australasian Conference, ACISP 2021, Virtual Event, December 1-3, 2021, Proceedings*, volume 13083 of *Lecture Notes in Computer Science*, pages 254–274. Springer, 2021.

[33] Prajwal Panzade and Daniel Takabi. Towards faster functional encryption for privacy-preserving machine learning. In *3rd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2021, Atlanta, GA, USA, December 13-15, 2021*, pages 21–30. IEEE, 2021.

[34] Elaine Shi and Nikhil Vanjani. Multi-client inner product encryption: Function-hiding instantiations without random oracles. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography - PKC 2023 - 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7-10, 2023, Proceedings, Part I*, volume 13940 of *Lecture Notes in Computer Science*, pages 622–651. Springer, 2023.

[35] Junichi Tomida. Unbounded quadratic functional encryption and more from pairings. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part III*, volume 14006 of *Lecture Notes in Computer Science*, pages 543–572. Springer, 2023.

[36] Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. Efficient functional encryption for inner-product values with full-hiding security. In Matt Bishop and Anderson C. A. Nascimento, editors, *Information Security - 19th International Conference, ISC 2016, Honolulu, HI, USA, September 3-6, 2016, Proceedings*, volume 9866 of *Lecture Notes in Computer Science*, pages 408–425. Springer, 2016.

[37] Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 609–639. Springer, 2018.

[38] Hoeteck Wee. Functional encryption for quadratic functions from k-lin, revisited. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part I*, volume 12550 of *Lecture Notes in Computer Science*, pages 210–228. Springer, 2020.

[39] Runhua Xu, Nathalie Baracaldo, and James Joshi. Privacy-preserving machine learning: Methods, challenges and directions. *CoRR*, abs/2108.04417, 2021.

[40] Lihua Yin, Jiyuan Feng, Hao Xun, Zhe Sun, and Xiaochun Cheng. A privacy-preserving federated learning for multiparty data sharing in social iots. *IEEE Trans. Netw. Sci. Eng.*, 8(3):2706–2718, 2021.

Appendix A. Security Analysis of Our SACfe

In this section, we provide the details of the indistinguishability argument of the consecutive hybrid of Theorem 1.

Lemma 1. $|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_{0'}]| \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof 1. Follows from the security of PRF function.

Lemma 2. $|\Pr[\mathsf{E}_{1\text{-}(\mu-1)\text{-}3}] - \Pr[\mathsf{E}_{1\text{-}\mu\text{-}1}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda)$.

Proof 2. We now construct a reduction algorithm $\mathcal{B}$ from the SXDH challenges using $\mathcal{A}$ as a subroutine. $\mathcal{B}$ gets the challenge instances from the adversary $\mathcal{A}$ for $\iota = 2$, i.e., $(\mathsf{G}, [\![a]\!]_2, [\![u]\!]_2, [\![t_\beta]\!]_2)$. Now we consider two matrices $\mathbf{W}_i \leftarrow \mathsf{GL}_4(\mathbb{Z}_p)$, $\widetilde{\mathbf{W}}_j \leftarrow \mathsf{GL}_3(\mathbb{Z}_p)$ and construct the matrices $(\mathbf{B}_i, \mathbf{B}_i^*)$, $(\widetilde{\mathbf{B}}_j, \widetilde{\mathbf{B}}_j^*)$ as follows:

$$\mathbf{B}_i = \begin{bmatrix} \mathbf{I}_2 & & \\ & 0 & 1 \\ & 1 & -a \end{bmatrix} \mathbf{W}_i; \mathbf{B}_i^* = \begin{bmatrix} \mathbf{I}_2 & & \\ & a & 1 \\ & 1 & 0 \end{bmatrix} \mathbf{W}_i^*;$$

$$\widetilde{\mathbf{B}}_j = \begin{bmatrix} 1 & & \\ & 0 & 1 \\ & 1 & -a \end{bmatrix} \widetilde{\mathbf{W}}_j; \widetilde{\mathbf{B}}_i^* = \begin{bmatrix} 1 & & \\ & a & 1 \\ & 1 & 0 \end{bmatrix} \widetilde{\mathbf{W}}_j^*$$

Now depending on the $\mu \in [\mathsf{Q}_{\mathsf{SK}}]$, we can simulate the ℓ-th secret key (where $\ell \neq \mu$) component $[\![\mathbf{k}^1_{\ell,i}]\!]_2$ corresponding to the vectors $\mathbf{y}_\ell$ as follows:

$$[\![\mathbf{k}^1_{\ell,i}]\!]_2 = \begin{cases} [\![(y^{(0)}_{\ell,i}, y^{(1)}_{\ell,i}, \gamma_{\ell,i}, 0)\mathbf{B}_i^*]\!]_2 & \text{if } \ell < \mu \\ [\![(y^{(0)}_{\ell,i}, 0, \gamma_{\ell,i}, 0)\mathbf{B}_i^*]\!]_2 & \text{if } \ell > \mu \end{cases};$$

$$[\![\mathbf{k}^2_{\ell,j}]\!]_2 = \begin{cases} [\![(\omega_\ell v^{(0)}_{\ell,j}, \widetilde{\gamma}_{\ell,j}, 0)\widetilde{\mathbf{B}}_j^*]\!]_2 & \text{if } \ell \neq \mu \end{cases}$$

where $\gamma_{\ell,i} \leftarrow \mathbb{Z}_p$ and $\sum_{i \in I_{\mathbf{y}_\ell}} \gamma_{\ell,i} + \sum_{j \in I_{\mathbf{v}_\ell}} \widetilde{\gamma}_{\ell,j} = 0$. For $\ell = \mu$, the secret key components $[\![\mathbf{k}^1_{\mu,i}]\!]_2$, $[\![\mathbf{k}^2_{\mu,j}]\!]_2$ are generated as:

$$\begin{aligned} [\![\mathbf{k}^1_{\mu,i}]\!]_2 &= [\![(y^{(0)}_{\mu,i}, 0, 0, 0)\mathbf{B}_i^* + \gamma_{\mu,i}(0, 0, t_\beta, u)\mathbf{W}_i^*]\!]_2 \\ &= [\![(y^{(0)}_{\mu,i}, 0, u\gamma_{\mu,i}, \beta f\gamma_{\mu,i})\mathbf{B}_i^*]\!]_2 \qquad (5) \\ [\![\mathbf{k}^2_{\mu,i}]\!]_2 &= [\![(\omega_\mu v_{\mu,j}, 0, 0)\widetilde{\mathbf{B}}_j^* + \widetilde{\gamma}_{\mu,j}(0, t_\beta, u)\widetilde{\mathbf{W}}_j^*]\!]_2 \\ &= [\![(\omega_\mu v_{\ell,j}, u\widetilde{\gamma}_{\mu,j}, \beta f\widetilde{\gamma}_{\mu,j})\widetilde{\mathbf{B}}_j^*]\!]_2 \qquad (6) \end{aligned}$$

where $\gamma_{\mu,i}, \widetilde{\gamma}_{\ell,j} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}_\mu}} \gamma_{\mu,i} + \sum_{j \in I_{\mathbf{v}_\mu}} \widetilde{\gamma}_{\mu,j} = 0$. The challenge ciphertext components $[\![\mathbf{c}^1_{k,i}]\!]_1$ simulate as follows:

$$\begin{aligned} [\![\mathbf{c}^1_{k,i}]\!]_1 &= [\![(x^{(0)}_{k,i}, 0, \alpha_k, 0)\mathbf{B}_i]\!]_1; \\ [\![\mathbf{c}^2_{k,j}]\!]_1 &= [\![(\delta_k w^{(0)}_{k,j}, \alpha_k, 0)\widetilde{\mathbf{B}}_j]\!]_1 \end{aligned}$$

where $\alpha_k \leftarrow \mathbb{Z}_p$. The adversarial view is the same as Game 1-μ-1 if $\beta = 1$ and Game 1-$(\mu-1)$-3 for $\beta = 0$. Thus the claim follows.

Lemma 3. $|\Pr[\mathsf{E}_{1\text{-}\mu\text{-}1}] - \Pr[\mathsf{E}_{1\text{-}\mu\text{-}2}]| \leq 2^{-\Omega(\lambda)}$.

Proof 3. We now construct the matrix $(\mathbf{D}_i, \mathbf{D}_i^*)$ as follows:

$$\mathbf{D}_i = \begin{bmatrix} 1 & & & \\ & 1 & & \frac{y^{(1)}_{\mu,i}}{\eta\gamma_{\mu,i}} \\ & & 1 & \\ & & & 1 \end{bmatrix} \mathbf{B}_i; \mathbf{D}_i^* = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & -\frac{y^{(1)}_{\mu,i}}{\eta\gamma_{\mu,i}} & & 1 \end{bmatrix} \mathbf{B}_i^*$$

Therefore, for all $k \in [\mathsf{Q}_{\mathsf{CT}}], \ell \in [\mathsf{Q}_{\mathsf{SK}}]$, the challenge ciphertext component $[\![\mathbf{c}^1_{k,i}]\!]_1$ is simulated as follows:

$$\mathbf{c}^1_{k,i} = (x^{(0)}_{k,i}, 0, \alpha_k, 0)\mathbf{B}_i = (x^{(0)}_{k,i}, 0, \alpha_k, 0)\mathbf{D}_i$$

where $\alpha_k \leftarrow \mathbb{Z}_p$. For all $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$, the secret key component $[\![\mathbf{k}^1_{\ell,i}]\!]_2$ is simulated as follows:

$$\begin{aligned} \mathbf{k}^1_{\ell,i} &= (y^{(0)}_{\ell,i}, \beta_\ell y^{(1)}_{\ell,i}, \gamma_{\ell,i}, \widehat{\beta}_\ell\eta\gamma_{\ell,i})\mathbf{B}_i^* \\ &= (y^{(0)}_{\ell,i}, (\beta_\ell + \widehat{\beta}_\ell)y^{(1)}_{\ell,i}, \gamma_{\ell,i}, \widehat{\beta}_\ell\eta\gamma_{\ell,i}) \end{aligned}$$

where $\gamma_{\ell,i}, \widetilde{\gamma}_{\ell,j} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}_\ell}} \gamma_{\ell,i} + \sum_{j \in I_{\mathbf{v}_\ell}} \widetilde{\gamma}_{\ell,j} = 0$. Now, we set

$$\beta_\ell = \begin{cases} 0 & \text{if } \ell \geq \mu \\ 1 & \text{if } \ell < \mu \end{cases}, \qquad \widehat{\beta}_\ell = \begin{cases} 0 & \text{if } \ell \neq \mu \\ 1 & \text{if } \ell = \mu \end{cases}$$

Therefore, both the Game 1-μ-2 and Game 1-μ-3 are identically distributed.

Lemma 4. $|\Pr[\mathsf{E}_{1\text{-}\mu\text{-}2}] - \Pr[\mathsf{E}_{1\text{-}\mu\text{-}3}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof 4. Follows similarly as Lemma 2 using SXDH instance over $\mathbb{G}_2$.

Lemma 5. $|\Pr[\mathsf{E}_{1\text{-}\mathsf{Q}_{\mathsf{SK}}\text{-}3}] - \Pr[\mathsf{E}_2]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$

Proof 5. Follows similarly as Lemma 2 using SXDH instance over $\mathbb{G}_2$.

Lemma 6. $|\Pr[\mathsf{E}_{3\text{-}(\nu-1)\text{-}3}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$

Proof 6. To achieve the indistinguishibility, the technique of Lemma 2 needs to apply over the ciphertext components $\mathbf{c}^1_{k,i}$, $\mathbf{c}^2_{k,j}$. Thus, we consider the SXDH instance over the group $\mathbb{G}_1$.

Lemma 7. $|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}2}]| \leq 2m_{1,\max}\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$

Proof 7. Chooses $m'_{1,\nu} \leftarrow [m_{1,\max}]$ as a guess of $m^*_{1,\nu}$ at the initial phase and consider intermediate games between Game 3-ν-1 to Game 3-ν-2.

Game 3-ν-1-1 ($\nu \in [\mathsf{Q}_{\mathsf{CT}}]$): Same as Game 3-$\nu$-1, except that the game will abort if $m^*_{1,\nu} \neq m'_{1,\nu}$.

Game 3-ν-1-2 ($\nu \in [\mathsf{Q}_{\mathsf{CT}}]$): Same as Game 3-$\nu$-1-1 except for the following components whenever $(\max(I_{\mathbf{y}_\ell}) > m'_{1,\nu}) \wedge (\min(I_{\mathbf{y}_\ell}) \leq m'_{1,\nu})$ and $(\max(I_{\mathbf{v}_\ell}) > m^*_{2,\nu}) \wedge (\min(I_{\mathbf{v}_\ell}) \leq m^*_{2,\nu})$ are satisfied

$$\mathbf{k}^1_{\ell,i} = \begin{cases} (y^{(0)}_{\ell,i}, y^{(1)}_{\ell,i}, \gamma_{\ell,i}, s_{\ell,i})\mathbf{B}_i^* & i \leq m'_{1,\nu} \\ (y^{(0)}_{\ell,i}, y^{(1)}_{\ell,i}, \gamma_{\ell,i}, \boxed{as_{\ell,i}})\mathbf{B}_i^* & i > m'_{1,\nu} \end{cases}$$

$$\mathbf{k}^2_{\ell,j} = \begin{cases} (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, t_{\ell,j})\widetilde{\mathbf{B}}_j^* & j \leq m^*_{2,\nu} \\ (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{at_{\ell,j}})\widetilde{\mathbf{B}}_j^* & j > m^*_{2,\nu} \end{cases}$$

with $a \leftarrow \mathbb{Z}_p$.

Game 3-ν-1-3 ($\nu \in [\mathsf{Q}_{\mathsf{CT}}]$): Same as Game 3-$\nu$-1-2 except for the following components

$$\mathbf{k}^1_{\ell,i} = (y^{(0)}_{\ell,i}, y^{(1)}_{\ell,i}, \gamma_{\ell,i}, \boxed{\widehat{s}_{\ell,i}})\mathbf{B}_i^* \quad \forall i \in I_{\mathbf{y}_\ell}$$

$$\mathbf{k}^2_{\ell,j} = (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{\widehat{t}_{\ell,j}})\widetilde{\mathbf{B}}_j^* \quad \forall j \in I_{\mathbf{v}_\ell}$$

where $\widehat{s}_{\ell,i}, \widehat{t}_{\ell,j} \leftarrow \mathbb{Z}_p$.

Game 3-ν-1-4 ($\nu \in [\mathsf{Q}_{\mathsf{CT}}]$): Same as Game 3-$\nu$-1-3 except for the following components

$$[\![\mathbf{c}^2_j]\!]_1 = [\![(\boxed{\delta_\nu w^{(0)}_{\nu,j} + \widehat{\xi}_j \widehat{\alpha}_\nu}, \alpha_\nu, \widehat{\alpha}_\nu)\widetilde{\mathbf{B}}_j]\!]_1 \quad \forall j \in [m^*_{2,\nu}]$$

$$\mathbf{k}^2_{\ell,j} = \begin{cases} (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{t_{\ell,j} - \widehat{\xi}_j \omega_\ell v_{\ell,j}})\widetilde{\mathbf{B}}^*_j & \max(I_{\mathbf{v}_\ell}) \le m^*_{2,\nu} \\ (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{\widehat{t}_{\ell,j} - \widehat{\xi}_j \omega_\ell v_{\ell,j}})\widetilde{\mathbf{B}}^*_j & \max(I_{\mathbf{v}_\ell}) > m^*_{2,\nu} \end{cases}$$

whenever ℓ-th quried predicate vector $\mathbf{v}_\ell$ satisfy $\mathsf{min}(I_{\mathbf{v}_\ell}) \le m^*_{2,\nu}$ with $\widehat{\xi}_j \leftarrow \mathbb{Z}_p$.

Game 3-ν-1-5 ($\nu \in [\mathsf{Q}_{\mathsf{CT}}]$): Same as Game 3-$\nu$-1-4 except for the following components whenever $(\max(I_{\mathbf{y}_\ell}) \le m'_{1,\nu}) \wedge (\max(I_{\mathbf{v}_\ell}) \le m^*_{2,\nu})$ such that $\langle \mathbf{w}^{(0)}_\nu, \mathbf{v}_\ell \rangle \neq 0, \langle \mathbf{w}^{(1)}_\nu, \mathbf{v}_\ell \rangle \neq 0$

$$\mathbf{k}^2_{\ell,j} = (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{\mathfrak{t}_{\ell,j}})\widetilde{\mathbf{B}}^*_j \quad \forall j \in I_{\mathbf{v}_\ell}$$

where $\mathfrak{t}_{\ell,j} \leftarrow \mathbb{Z}_p$.

Game 3-ν-1-6 ($\nu \in [\mathsf{Q}_{\mathsf{CT}}]$): Same as Game 3-$\nu$-1-5 except for the following components

$$[\![\mathbf{c}^1_{\nu,i}]\!]_1 = [\![(\boxed{0, x^{(1)}_{\nu,i}}, \alpha_\nu, \widehat{\alpha}_\nu)\mathbf{B}_i]\!]_1 \quad \forall i \in [m'_{1,\nu}],$$

$$[\![\mathbf{c}^2_{\nu,j}]\!]_1 = [\![(\boxed{\delta_\nu w^{(1)}_{\nu,j} + \widehat{\xi}_j \widehat{\alpha}_\nu}, \alpha_\nu, \widehat{\alpha}_\nu)\widetilde{\mathbf{B}}_j]\!]_1 \ \forall j \in [m^*_{2,\nu}].$$

Game 3-ν-1-7 ($\nu \in [\mathsf{Q}_{\mathsf{CT}}]$): Same as Game 3-$\nu$-1-6 except for the following components

$$[\![\mathbf{c}^2_j]\!]_1 = [\![(\delta_\nu w^{(1)}_{\nu,j} + \widehat{\xi}_j \widehat{\alpha}_\nu, \alpha_\nu, \widehat{\alpha}_\nu)\widetilde{\mathbf{B}}_j]\!]_1 \ \forall j \in [m^*_{2,\nu}]$$

$$\mathbf{k}^2_{\ell,j} = \begin{cases} (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{t_{\ell,j} - \widehat{\xi}_j \omega_\ell v_{\ell,j}})\widetilde{\mathbf{B}}^*_j & \max(I_{\mathbf{v}_\ell}) \le m^*_{2,\nu} \\ (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{\widehat{t}_{\ell,j} - \widehat{\xi}_j \omega_\ell v_{\ell,j}})\widetilde{\mathbf{B}}^*_j & \max(I_{\mathbf{v}_\ell}) > m^*_{2,\nu} \end{cases}$$

whenever ℓ-th queried predicate vector $\mathbf{v}_\ell$ satisfy $\mathsf{min}(I_{\mathbf{v}_\ell}) \le m^*_{2,\nu}$ for $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$ and $\widehat{\xi}_j \leftarrow \mathbb{Z}_p$.

Game 3-ν-1-8 ($\nu \in [\mathsf{Q}_{\mathsf{CT}}]$): Same as Game 3-$\nu$-1-7 except for the following components

$$\mathbf{k}^1_{\ell,i} = (y^{(0)}_{\ell,i}, y^{(1)}_{\ell,i}, \gamma_{\ell,i}, \boxed{s_{\ell,i}})\mathbf{B}^*_i \quad \forall i \in I_{\mathbf{y}_\ell}$$

$$\mathbf{k}^2_{\ell,j} = (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \boxed{t_{\ell,j}})\widetilde{\mathbf{B}}^*_j \quad \forall j \in I_{\mathbf{v}_\ell}$$

$$[\![\mathbf{c}^1_{\nu,i}]\!]_1 = [\![(0, x^{(1)}_{\nu,i}, \alpha_\nu, \widehat{\alpha}_\nu)\mathbf{B}_i]\!]_1 \ \forall i \in [m'_{1,\nu}],$$

$$[\![\mathbf{c}^2_{\nu,j}]\!]_1 = [\![(\boxed{\delta_\nu w^{(1)}_{\nu,j}}, \alpha_\nu, \widehat{\alpha}_\nu)\widetilde{\mathbf{B}}_j]\!]_1 \ \forall j \in [m^*_{2,\nu}]$$

Claim 1. $\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] = \frac{1}{m_{1,\max}} \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1}]$.

Proof 8. Let $m_{1,\max}, m_{2,\max}$ be the maximum length of the challenge vector and challenge attribute vector respectively. Note that Game 3-ν-1 is similar to Game 3-ν-1-1 except that $\mathcal{A}$'s output is $\perp$ if $m'_{1,\nu} \neq m^*_{1,\nu}$ where $m'_{1,\nu}$ is the guess of $m^*_{1,\nu}$. Now, we have

$$\begin{aligned} \Pr(\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}) &= \sum_{i \in [m_{1,\max}]} \Pr[m'_{1,\nu} = i] \cdot \Pr[m^*_{1,\nu} = i \wedge \mathsf{E}_{3\text{-}\nu\text{-}1} | m'_{1,\nu} = i] \\ &= \frac{1}{m_{1,\max}} \cdot \Pr(\mathsf{E}_{3\text{-}\nu\text{-}1}). \end{aligned}$$

Claim 2. $|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}1}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}2}]| \le 2^{-\Omega(\lambda)}$.

Proof 9. For $i > m'_{1,\nu} \wedge j > m^*_{2,\nu}$, we construct the matrices $(\mathbf{D}_i, \mathbf{D}^*_i)$, $(\widetilde{\mathbf{D}}_j, \widetilde{\mathbf{D}}^*_j)$ as follows:

$$\mathbf{D}_i = \begin{bmatrix} \mathbf{I}_3 & \\ & a \end{bmatrix} \mathbf{B}_i, \ \mathbf{D}^*_i = \begin{bmatrix} \mathbf{I}_3 & \\ & \frac{1}{a} \end{bmatrix} \mathbf{B}^*_i;$$

$$\widetilde{\mathbf{D}}_j = \begin{bmatrix} \mathbf{I}_2 & \\ & a \end{bmatrix} \widetilde{\mathbf{B}}_j, \ \widetilde{\mathbf{D}}^*_j = \begin{bmatrix} \mathbf{I}_2 & \\ & \frac{1}{a} \end{bmatrix} \widetilde{\mathbf{B}}^*_j$$

For $i > m'_{1,\nu}, j > m^*_{2,\nu}$, the secret key $[\![\mathbf{k}^1_{\ell,i}]\!]_2, [\![\mathbf{k}^2_{\ell,j}]\!]_2$ and k-th ($k \neq \nu$) ciphertext components $[\![\mathbf{c}^1_{k,i}]\!]_1, [\![\mathbf{c}^2_{k,j}]\!]_1$ are generated as follows:

$$\begin{aligned} [\![\mathbf{c}^1_{k,i}]\!]_1 &= [\![(\beta_k x^{(0)}_{k,i}, (1-\beta_k) x^{(1)}_{k,i}, \alpha_k, 0)\mathbf{B}_i]\!]_1 \\ &= [\![(\beta_k x^{(0)}_{k,i}, (1-\beta_k) x^{(1)}_{k,i}, \alpha_k, 0)\mathbf{D}_i]\!]_1 \\ [\![\mathbf{c}^2_{k,j}]\!]_1 &= [\![(\beta_k \delta_k w^{(0)}_{k,j} + (1-\beta_k)\delta_k w^{(1)}_{k,j}, \alpha_k, 0)\widetilde{\mathbf{B}}_j]\!]_1 \\ &= [\![(\beta_k \delta_k w^{(0)}_{k,j} + (1-\beta_k)\delta_k w^{(1)}_{k,j}, \alpha_k, 0)\widetilde{\mathbf{D}}_j]\!]_1 \\ \mathbf{k}^1_{\ell,i} &= (y^{(0)}_{\ell,i}, y^{(1)}_{\ell,i}, \gamma_{\ell,i}, s_{\ell,i})\mathbf{B}^*_i = (y^{(0)}_{\ell,i}, y^{(1)}_{\ell,i}, \gamma_{\ell,i}, a s_{\ell,i})\mathbf{D}^*_i \\ \mathbf{k}^2_{\ell,j} &= (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, t_{\ell,i})\widetilde{\mathbf{B}}^*_j = (\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, a t_{\ell,i})\widetilde{\mathbf{D}}^*_j \end{aligned}$$

where $\omega_\ell, a \leftarrow \mathbb{Z}_p$ and define $\beta_k = 0$ if $k < \nu$ elsewhere $\beta_k = 1$. Observe that there is no change for the distribution of third entries in the both secret keys components $\mathbf{k}^1_{\ell,i}$ and $\mathbf{k}^2_{\ell,j}$ since $\sum_{i \in I_{\mathbf{y}_\ell}} a s_{\ell,i} + \sum_{j \in I_{\mathbf{v}_\ell}} a t_{\ell,i} = a(\sum_{i \in I_{\mathbf{y}_\ell}} s_{\ell,i} + \sum_{j \in I_{\mathbf{v}_\ell}} t_{\ell,i}) = 0$. Note that the ciphertext component does not change its distribution because the basis technique approach is only applied for $i > m'_{1,\nu}, j > m^*_{2,\nu}$. Hence, Game 1-$\nu$-1-2 and Game 1-$\nu$-1-1 are identically close unless $a = 0$.

Claim 3. $|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}2}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}]| \le \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}} + 2^{-\Omega(\lambda)}$.

Proof 10. This proof basically sets $s_{\ell,i}$ (resp. $t_{\ell,i}$) as an affine function $u\bar{s}_{\ell,i} + s'_{\ell,i}$ (resp. $u\bar{t}_{\ell,i} + t'_{\ell,i}$) for all $i > m_{1,\nu}$ (resp. $i > m^*_{2,mu}$). It then replaces $[\![au]\!]_2$ with $[\![f]\!]_2$ by the SXDH assumption. Due to its simple structure, we remove this proof due to space limitations. We further note that this proof is similar to that of [37, Claim 3].

Claim 4. $|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}3}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}]| \le 2^{-\Omega(\lambda)}$.

Proof 11. We construct the matrix $(\widetilde{\mathbf{W}}_j, \widetilde{\mathbf{W}}^*_j)$ for all $j \in I_{\mathbf{v}_\nu}$ as follows:

$$\widetilde{\mathbf{W}}_j = \begin{bmatrix} 1 & & \\ & 1 & \\ -\widehat{\xi}_j & & 1 \end{bmatrix} \widetilde{\mathbf{B}}_j, \ \widetilde{\mathbf{W}}^*_j = \begin{bmatrix} 1 & & \widehat{\xi}_j \\ & 1 & \\ & & 1 \end{bmatrix} \widetilde{\mathbf{B}}^*_j$$

where $\widetilde{\mathbf{B}}_j \leftarrow \mathsf{M}_3(\mathbb{Z}_p)$ and $\widehat{\xi}_j \leftarrow \mathbb{Z}_p$. The challenger $\mathcal{B}$ generates ν-th challenge ciphertext components corresponding to the message, attribute vectors pair $(\mathbf{x}_\nu, \mathbf{w}_\nu)$ as follows:

$$\begin{aligned} [\![\mathbf{c}^2_{k,j}]\!]_1 &= [\![((1-\beta_k)\delta_k w^{(0)}_{k,j} + \beta_k \delta_k w^{(1)}_{k,j}, \alpha_k, \widehat{\beta}_k \widehat{\alpha}_k)\widetilde{\mathbf{B}}_j]\!]_1 \\ &= [\![((1-\beta_k)\delta_k w^{(0)}_{k,j} + \beta_k \delta_k w^{(1)}_{k,j} + \widehat{\xi}_j \widehat{\beta}_k \widehat{\alpha}_k, \\ &\qquad \alpha_k, \widehat{\beta}_k \widehat{\alpha}_k)\widetilde{\mathbf{W}}_j]\!]_1 \end{aligned}$$

set $\beta_k = \begin{cases} 0 & \text{if } k \geq \nu \\ 1 & \text{if } k \leq \nu \end{cases}$ and $\widehat{\beta}_k = \begin{cases} 0 & \text{if } k \neq \nu \\ 1 & \text{if } k = \nu \end{cases}$

Note that the challenge ciphertext component $[\![\mathbf{c}^1_{k,i}]\!]_1$ for all $k \in [\mathsf{Q}_{\mathsf{CT}}]$ are generated as previous Game 3-ν-1-2. The above changes does not effect the ciphertext component $[\![\mathbf{c}^1_{k,i}]\!]_1$ as the basis $(\mathbf{B}, \mathbf{B}^*)$ remain unaltered. Now the second secret key component $[\![\mathbf{k}^2_{\ell,i}]\!]_2$ are generated by using the basis $(\widetilde{\mathbf{W}}_j, \widetilde{\mathbf{W}}^*_j)$ in Game 3-1-ν-3 as follows:
For $j \in I_{\mathbf{v}_\ell} : \max(I_{\mathbf{v}_\ell}) \leq m^*_{2,\nu}$, we have

$$\begin{aligned}[\![\mathbf{k}^2_{\ell,j}]\!]_1 &= [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, t_{\ell,j})\widetilde{\mathbf{B}}^*_j]\!]_2 \\ &= [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, t_{\ell,j} - \widehat{\xi}_j \omega_\ell v_{\ell,j})\widetilde{\mathbf{W}}^*_j]\!]_2\end{aligned}$$

Also for $j \in I_{\mathbf{v}_\ell} : \max(I_{\mathbf{v}_\ell}) > m^*_{2,\nu}$, we have

$$\begin{aligned}[\![\mathbf{k}^2_{\ell,j}]\!]_1 &= [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j})\widetilde{\mathbf{B}}^*_j]\!]_2 \\ &= [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j} - \widehat{\xi}_j \omega_\ell v_{\ell,j})\widetilde{\mathbf{W}}^*_j]\!]_2\end{aligned}$$

where $t_{\ell,j} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}_\ell}} s_{\ell,i} + \sum_{j \in I_{\mathbf{v}_\ell}} t_{\ell,j} = 0$ and $\widehat{t}_{\ell,j} \leftarrow \mathbb{Z}_p$. Therefore, Game 3-$\nu$-1-3 and Game 3-$\nu$-1-4 are identically distributed unless $\widehat{\xi}_{\nu,j} = 0$.

Claim 5. $|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}4}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5}]| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof 12. We will show that $\mathcal{B}$ can utilize the instance $(\mathsf{G}, [\![a]\!]_2, [\![u]\!]_2, [\![t_\beta]\!]_2)$ of the SXDH assumption to interpolate between Game 3-ν-1-4 and Game 3-ν-1-5 using $\mathcal{A}$ as a subroutine. The algorithm $\mathcal{B}$ implicitly define orthonormal dual bases $(\widetilde{\mathbf{B}}_j, \widetilde{\mathbf{B}}^*_j)$ by choosing $\widetilde{\mathbf{D}}_j, \widetilde{\mathbf{D}}^*_j \leftarrow \mathsf{GL}_3(\mathbb{Z}_p)$ and setting

$$\widetilde{\mathbf{B}}_j = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & a \end{bmatrix} \widetilde{\mathbf{D}}_j, \quad \widetilde{\mathbf{B}}^*_j = \begin{bmatrix} a & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \widetilde{\mathbf{D}}^*_j$$

for all $j \in [m'_{1,\nu}]$ and a is implicitly provided through the SXDH instance. For $\langle \mathbf{w}^{(0)}_{k,j}, \mathbf{v}_\ell \rangle \neq 0, \langle \mathbf{w}^{(1)}_{k,j}, \mathbf{v}_\ell \rangle \neq 0$, $\mathcal{B}$ simulates the ℓ-th secret key component $[\![\mathbf{k}^2_{\ell,j}]\!]_2$ as follows:

$$\begin{aligned}[\![\mathbf{k}^2_{\ell,j}]\!]_2 = [\![&(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, t_{\ell,j} - \widetilde{\xi}_j(\omega_\ell + u\langle \mathbf{w}^{(0)}, \mathbf{v}_\ell \rangle) v_{\ell,j}) \\ &\widetilde{\mathbf{B}}^*_j + v_{\ell,j}\langle \mathbf{w}^{(0)}, \mathbf{v}_\ell \rangle (t_\beta, 0, -u)\widetilde{\mathbf{D}}^*_j]\!]_2 \\ = [\![&((\omega_\ell + u\langle \mathbf{w}^{(0)}, \mathbf{v}_\ell \rangle) v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, t_{\ell,j} - \widetilde{\xi}_j \\ &(\omega_\ell + u\langle \mathbf{w}^{(0)}, \mathbf{v}_\ell \rangle) v_{\ell,j} + \beta f v_{\ell,j}\langle \mathbf{w}^{(0)}, \mathbf{v}_\ell \rangle)\widetilde{\mathbf{B}}^*_j]\!]_2\end{aligned}$$

with $\widetilde{\gamma}_{\ell,j}, t_{\ell,j} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}_\ell}} \gamma_{\ell,i} + \sum_{j \in I_{\mathbf{v}_\ell}} \widetilde{\gamma}_{\ell,i} = 0$ and $\sum_{i \in I_{\mathbf{y}_\ell}} s_{\ell,j} + \sum_{j \in I_{\mathbf{v}_\ell}} t_{\ell,j} = 0$ where $\gamma_{\ell,i}, s_{\ell,i} \leftarrow \mathbb{Z}_p$. As $\langle \mathbf{w}^{(0)}, \mathbf{v}_\ell \rangle \neq 0$, we can implicitly set $\omega'_\ell = \omega_\ell + u\langle \mathbf{w}^{(0)}, \mathbf{v}_\ell \rangle, \mathfrak{r}_{\ell,j} = t_{\ell,j} - \widetilde{\xi}_j(\omega_\ell + u\langle \mathbf{w}^{(0)}, \mathbf{v}_\ell \rangle) v_{\ell,j} + f v_{\ell,j}\langle \mathbf{w}^{(0)}, \mathbf{v}_\ell \rangle$ which are random elements in $\mathbb{Z}_p$ for $f \neq 0$. Therefore, the third component of $[\![\mathbf{k}^2_j]\!]_2$ is a random element for $\beta = 1$. Here, we use the fact that $\widetilde{\mathfrak{r}}_{\ell,j} + s_{\ell,i} + \xi_i y_{\ell,i} \neq 0$ with high probability. Hence, the adversarial view is the same as in Game 3-ν-1-5 for $\beta = 1$, otherwise, the view is similar as in Game 3-ν-1-4 if $\beta = 0$.
Now, the ν^{th} challenge ciphertext components are constructed as follows:

$$\begin{aligned}[\![\mathbf{c}^{(2)}_{\nu,j}]\!]_1 &= [\![(\delta_\nu w^{(0)}_{\nu,j} + \widehat{\alpha}_\nu \widetilde{\xi}'_j, \alpha_\nu, 0)\widetilde{\mathbf{B}}_j + (\widehat{\alpha}_\nu, 0, 0)\widetilde{\mathbf{D}}_j]\!]_1 \\ &= [\![(\delta_\nu w^{(0)}_{\nu,j} + \widehat{\alpha}_\nu \widetilde{\xi}_j, \alpha_\nu, \widehat{\alpha}_\nu)\widetilde{\mathbf{B}}_j]\!]_1 \ \forall j \in [m^*_{2,\nu}]\end{aligned}$$

where $\delta_\nu, \widehat{\alpha}_\nu, \widetilde{\xi}'_j \leftarrow \mathbb{Z}_p$. Thus, the distribution of the challenge ciphertext components in Game 3-ν-1-4 is identical with the distribution of Game 3-ν-1-5. Hence, $\mathcal{B}$ interpolates between Game 3-ν-1-5 and Game 3-ν-1-4 and the claim follows.

Claim 6. $|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}6}]| \leq 2^{-\Omega(\lambda)}$.

Proof 13. Let $\widetilde{\mathsf{E}}_\iota$ be the event that denotes $m'_{1,\nu} = m^*_{1,\nu}$ in Game ι where $m'_{1,\nu}$ is the guess of the length $m^*_{1,\nu}$ of ν^{th} message vector. Since $\mathcal{A}$'s view are equivalent for all previous ciphertext query, we have $\Pr(\widetilde{\mathsf{E}}_{3\text{-}\nu\text{-}1\text{-}5}) = \Pr(\widetilde{\mathsf{E}}_{3\text{-}\nu\text{-}1\text{-}6})$. Let us define $(\mathbf{D}_i, \mathbf{D}^*_i)$ for all $i \in [m'_{1,\nu}]$, and ξ'_j for all $j \in [m^*_{2,\nu}]$ as follows:

$$\widetilde{\mathbf{D}}_i = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ \frac{x^{(0)}_{\nu,i}}{\widehat{\alpha}_\nu} & -\frac{x^{(1)}_{\nu,i}}{\widehat{\alpha}_\nu} & 0 & 1 \end{bmatrix} \widetilde{\mathbf{B}}_i; \widetilde{\mathbf{D}}^*_i = \begin{bmatrix} 1 & & -\frac{x^{(0)}_{\nu,i}}{\widehat{\alpha}_\nu} & \\ & 1 & \frac{x^{(1)}_{\nu,i}}{\widehat{\alpha}_\nu} & \\ & & 1 & 0 \\ & & & 1 \end{bmatrix} \widetilde{\mathbf{B}}^*_i$$

$$\widetilde{\xi}'_j = \widetilde{\xi}_j - \frac{\delta_\nu(w^{(1)}_{\nu,j} - w^{(0)}_{\nu,j})}{\widehat{\alpha}_\nu}$$

where $\widehat{\alpha}_\nu, \delta_\nu, \xi_j \leftarrow \mathbb{Z}_p$ and $(\mathbf{x}^{(0)}, \mathbf{w}^{(0)})$, $(\mathbf{x}^{(1)}, \mathbf{w}^{(1)})$ are challenge message and attribute pairs. Note that, $\widetilde{\xi}'_j$ are independently random elements in $\mathbb{Z}_p$ unless $\widehat{\alpha}_\nu = 0$. Then the challenge ciphertext components $[\![\mathbf{c}^1_{k,i}]\!]_1$ and $[\![\mathbf{c}^2_{k,j}]\!]_1$ are indistinguishable in Game 3-ν-1-5 and Game 3-ν-1-6 as shown below,

$$\begin{aligned}[\![\mathbf{c}^1_{k,i}]\!]_1 &= [\![(\beta_k x^{(0)}_{k,i}, (1-\beta_k)x^{(1)}_{k,i}, \alpha_k, \hat{\beta}_k\widehat{\alpha}_k)\mathbf{B}_i]\!]_1 \\ &= [\![((\beta_k - \hat{\beta}_k)x^{(0)}_{k,i}, (1 - \beta_k + \hat{\beta}_k)x^{(1)}_{k,i}, \alpha_k, \\ &\quad \hat{\beta}_k\widehat{\alpha}_k)\mathbf{D}_i]\!]_1 \quad \forall i \in [m^*_{1,\nu}]\end{aligned}$$

$$\text{set } \beta_k = \begin{cases} 0 & \text{if } k \leq \nu \\ 1 & \text{if } k > \nu \end{cases}, \ \widehat{\beta}_k = \begin{cases} 0 & \text{if } k \neq \nu \\ 1 & \text{if } k = \nu \end{cases}$$

$$[\![\mathbf{c}^2_{k,j}]\!]_1 = \begin{cases} [\![(\delta_k w^{(1)}_{k,j}, \alpha_k, 0)\widetilde{\mathbf{B}}_j]\!]_1 & \text{if } k < \nu \\ [\![(\delta_\nu w^{(1)}_{\nu,j} + \widetilde{\xi}'_j\widehat{\alpha}_\nu, \alpha_\nu, \widehat{\alpha}_\nu)\widetilde{\mathbf{B}}_j]\!]_1 & \text{if } k = \nu \\ [\![(\delta_k w^{(0)}_{k,j}, \alpha_k, 0)\widetilde{\mathbf{B}}_j]\!]_1 & \text{if } k > \nu \end{cases}$$

where $\widehat{\alpha}_\nu, \alpha_k \leftarrow \mathbb{Z}_p$ for all $k \in [\mathsf{Q}_{\mathsf{CT}}]$. For all $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$ we categorise adversary's queries to the ℓ-th oracle secret key on $\mathbf{y}_\ell = (y_{\ell,i})_{i \in I_{\mathbf{y}_\ell}}, \mathbf{v}_\ell = (v_{\ell,j})_{j \in I_{\mathbf{v}_\ell}}$ and show that in each cases the ℓ-th secret key components $\{[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2\}$ are indistinguishable in Game 3-ν-1-5 and Game 3-ν-1-6.

Case I when $\langle \mathbf{w}^{(0)}_\nu, \mathbf{v}_\ell \rangle \neq 0, \langle \mathbf{w}^{(1)}_\nu, \mathbf{v}_\ell \rangle \neq 0$.

(i) If $(\mathsf{max}(I_{\mathbf{y}_\ell}) \leq m'_{1,\nu}) \wedge (\mathsf{max}(I_{\mathbf{v}_\ell}) \leq m^*_{2,\nu})$, then

$$[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2 = [\![(y^0_{\ell,i}, y^1_{\ell,i}, \gamma_{\ell,i}, \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i})\mathbf{B}^*_i]\!]_2$$
$$[\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2 = [\![\left(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widetilde{\mathfrak{t}}_{\ell,j}\right)\widetilde{\mathbf{B}}^*_j]\!]_2$$

where $\widehat{s}_{\ell,i}, \widetilde{\mathfrak{t}}_{\ell,j} \leftarrow \mathbb{Z}_p$ for all $j \in I_{\mathbf{v}_\ell}, i \in I_{\mathbf{y}_\ell}$. Observe that, $\sum_{i\in I_{\mathbf{y}_\ell}}(y^0_{\ell,i}x^0_{\nu,i} - y^1_{\ell,i}x^1_{\nu,i}) = 0$. Thus, we can set $s_{\ell,i} = \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i}$ such that $s_{\ell,i}$ is randomly distributed. Thus, $\{[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2\}$ are distributed properly.

(ii) If $(\mathsf{max}(I_{\mathbf{y}_\ell}) > m'_{1,\nu}) \wedge (\mathsf{max}(I_{\mathbf{v}_\ell}) \leq m^*_{2,\nu})$, then

$$\forall i \leq m'_{1,\nu},$$
$$[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2 = [\![(y^0_{\ell,i}, y^1_{\ell,i}, \gamma_{\ell,i}, \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i})\mathbf{B}^*_i]\!]_2$$
$$\forall i > m'_{1,\nu}, [\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2 = [\![\left(y^0_{\ell,i}, y^1_{\ell,i}, \gamma_{\ell,i}, \widehat{s}_{\ell,i}\right)\mathbf{B}^*_i]\!]_2$$
$$\forall j \leq m^*_{2,\nu}, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2 = [\![\left(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widetilde{\mathfrak{t}}_{\ell,j}\right)\widetilde{\mathbf{B}}^*_j]\!]_2$$

where $\widehat{s}_{\ell,i}, \widetilde{\mathfrak{t}}_{\ell,j} \leftarrow \mathbb{Z}_p$ for all $j \in I_{\mathbf{v}_\ell}, i \in I_{\mathbf{y}_\ell}$. Here we set $s_{\ell,i} = \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i}$ for $i \leq m'_{1,\nu}$ which are independently random elements from $\mathbb{Z}_p$ as there are no condition on $(y^0_{\ell,i}x^0_{\nu,i} - y^1_{\ell,i}x^1_{\nu,i})$ and $\widehat{s}_{\ell,i}$ are independently random elements in $\mathbb{Z}_p$. Also, $\widehat{s}_{\ell,i}$ are random elements from $i > m'_{1,\nu}$, the fourth component of $\mathbf{k}^{(1)}_{\ell,i}$ is uniform element from $\mathbb{Z}_p$. Thus, $\{[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2\}$ are distributed properly.

Case II when $\langle \mathbf{w}^{(0)}_\nu, \mathbf{v}_\ell \rangle = \langle \mathbf{w}^{(1)}_\nu, \mathbf{v}_\ell \rangle = 0$.

(iii) If $(\mathsf{max}(I_{\mathbf{y}_\ell}) \leq m'_{1,\nu}) \wedge (\mathsf{max}(I_{\mathbf{v}_\ell}) \leq m^*_{2,\nu})$, then

$$[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2 = [\![(y^0_{\ell,i}, y^1_{\ell,i}, \gamma_{\ell,i}, \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i})\mathbf{B}^*_i]\!]_2$$
$$[\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2 = [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j} - \xi_j\omega_\ell v_{\ell,j})\widetilde{\mathbf{B}}^*_j]\!]_2$$
$$= [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j} - \widetilde{\xi}'_j\omega_\ell v_{\ell,j} - \frac{\delta_\nu\omega_\ell(w^{(1)}_{\nu,j}v_{\ell,j} - w^{(0)}_{\nu,j}v_{\ell,j})}{\widehat{\alpha}_\nu})\widetilde{\mathbf{B}}^*_j]\!]_2$$

where $\widehat{s}_{\ell,i}, \widehat{t}_{\ell,j} \leftarrow \mathbb{Z}_p$ for all $j \in I_{\mathbf{v}_\ell}, i \in I_{\mathbf{y}_\ell}$ s.t. $\sum_{i\in I_{\mathbf{y}_\ell}}\widehat{s}_{\ell,i} + \sum_{j\in I_{\mathbf{v}_\ell}}\widehat{t}_{\ell,j} = 0$. Hence, we set $s_{\ell,i} = \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i}$ for $i \leq m'_{1,\nu}$ and $t_{\ell,j} = \widehat{t}_{\ell,j} - \delta_\nu\omega_\ell\frac{(w^{(1)}_{\nu,j} - w^{(0)}_{\nu,j})}{\widehat{\alpha}_\nu}v_{\ell,j}$ for $j \in [m^*_{2,\nu}]$. Observe that, $\sum_{i\in I_{\mathbf{y}_\ell}}(y^0_{\ell,i}x^0_{\nu,i} - y^1_{\ell,i}x^1_{\nu,i}) + \sum_{i\in I_{\mathbf{v}_\ell}}(w^{(1)}_{\nu,j} - w^{(0)}_{\nu,j})v_{\ell,j} = 0$. Observe that $s_{\ell,i}$ and $t_{\ell,j}$ are randomly distributed s.t. $\sum_{i\in I_{\mathbf{y}_\ell}} s_{\ell,i} + \sum_{j\in I_{\mathbf{v}_\ell}} t_{\ell,j} = 0$. Thus, $\{[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2\}$ are distributed properly.

(iv) If $(\mathsf{max}(I_{\mathbf{y}_\ell}) > m'_{1,\nu}) \wedge (\mathsf{max}(I_{\mathbf{v}_\ell}) \leq m^*_{2,\nu})$, then

$$\forall i \leq m'_{1,\nu},$$
$$[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2 = [\![(y^0_{\ell,i}, y^1_{\ell,i}, \gamma_{\ell,i}, \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i})\mathbf{B}^*_i]\!]_2$$
$$\forall i > m'_{1,\nu}, [\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2 = [\![\left(y^0_{\ell,i}, y^1_{\ell,i}, \gamma_{\ell,i}, \widehat{s}_{\ell,i}\right)\mathbf{B}^*_i]\!]_2$$
$$\forall j \leq m^*_{2,\nu}, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2 = [\![\left(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j} - \xi_j\omega_\ell v_{\ell,j}\right)\widetilde{\mathbf{B}}^*_j]\!]_2$$
$$= [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j} - \widetilde{\xi}'_j\omega_\ell v_{\ell,j} - \frac{\delta_\nu\omega_\ell(w^{(1)}_{\nu,j}v_{\ell,j} - w^{(0)}_{\nu,j}v_{\ell,j})}{\widehat{\alpha}_\nu})\widetilde{\mathbf{B}}^*_j]\!]_2$$

where $\widehat{s}_{\ell,i}, \widehat{t}_{\ell,j} \leftarrow \mathbb{Z}_p$ for all $j \in I_{\mathbf{v}_\ell}, i \in I_{\mathbf{y}_\ell}$. Here, we set $s_{\ell,i} = \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i}$ for $i \leq m'_{1,\nu}$ which are independently random elements from $\mathbb{Z}_p$ as there are no condition on $(y^0_{\ell,i}x^0_{\nu,i} - y^1_{\ell,i}x^1_{\nu,i})$ and $\widehat{s}_{\ell,i}$ are independently random elements in $\mathbb{Z}_p$. Also, $\widehat{s}_{\ell,i}$ are random elements from $i > m'_{1,\nu}$, the fourth component of $\mathbf{k}^{(1)}_{\ell,i}$ is uniform element from $\mathbb{Z}_p$ for all $i \in I_{\mathbf{y}_\ell}$. Considering $t_{\ell,j} = \widehat{t}_{\ell,j} - \delta_\nu\omega_\ell\frac{(w^{(1)}_{\nu,j} - w^{(0)}_{\nu,j})}{\widehat{\alpha}_\nu}v_{\ell,j}$ for $j \in [m^*_{2,\nu}]$ so $t_{\ell,j}$ are uniformly random. Thus, $\{[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2\}$ are distributed properly.

Case III when $\mathsf{max}(I_{\mathbf{v}_\ell}) > m^*_{2,\nu}$.

(v) If $(\mathsf{max}(I_{\mathbf{y}_\ell}) \leq m'_{1,\nu}) \wedge (\mathsf{max}(I_{\mathbf{v}_\ell}) > m^*_{2,\nu})$, then

$$\forall i \leq m'_{1,\nu},$$
$$[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2 = [\![(y^0_{\ell,i}, y^1_{\ell,i}, \gamma_{\ell,i}, \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i})\mathbf{B}^*_i]\!]_2$$
$$\forall j \leq m^*_{2,\nu}, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2 = [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j} - \xi_j\omega_\ell v_{\ell,j})\widetilde{\mathbf{B}}^*_j]\!]_2$$
$$= [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j} - \widetilde{\xi}'_j\omega_\ell v_{\ell,j} - \frac{\delta_\nu\omega_\ell(w^{(1)}_{\nu,j}v_{\ell,j} - w^{(0)}_{\nu,j}v_{\ell,j})}{\widehat{\alpha}_\nu})\widetilde{\mathbf{B}}^*_j]\!]_2$$
$$\forall j > m^*_{2,\nu}, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2 = [\![\left(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j}\right)\widetilde{\mathbf{B}}^*_j]\!]_2$$

where $\widehat{s}_{\ell,i}, \widehat{t}_{\ell,j} \leftarrow \mathbb{Z}_p$ for all $j \in I_{\mathbf{v}_\ell}, i \in I_{\mathbf{y}_\ell}$. Hence, we set $t_{\ell,j} = \widehat{t}_{\ell,j} - \delta_\nu\omega_\ell\frac{(w^{(1)}_{\nu,j} - w^{(0)}_{\nu,j})}{\widehat{\alpha}_\nu}v_{\ell,j}$ for $j \in [m_{2,\nu}]$ which are independently random elements from $\mathbb{Z}_p$ as there are no condition on $(w^{(1)}_{\nu,j} - w^{(0)}_{\nu,j})v_{\ell,j}$. Moreover, the third component of $\mathbf{k}^{(2)}_{\ell,j}$ are independently random elements in $\mathbb{Z}_p$ since $\widehat{t}_{\ell,j}$ for $j > [m^*_{2,\nu}]$ are independent of $t_{\ell,j}$ for $j \leq [m^*_{2,\nu}]$. Thus, $\{[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2\}$ are distributed properly.

(vi) If $\mathsf{max}(I_{\mathbf{y}_\ell}) > m'_{1,\nu} \wedge \mathsf{max}(I_{\mathbf{v}_\ell}) > m^*_{2,\nu}$, then

$$\forall i \leq m'_{1,\nu},$$
$$[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2 = [\![(y^0_{\ell,i}, y^1_{\ell,i}, \gamma_{\ell,i}, \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha_\nu}} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha_\nu}} + \widehat{s}_{\ell,i})\mathbf{B}^*_i]\!]_2$$
$$\forall i > m'_{1,\nu}, [\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2 = [\![\left(y^0_{\ell,i}, y^1_{\ell,i}, \gamma_{\ell,i}, \widehat{s}_{\ell,i}\right)\mathbf{B}^*_i]\!]_2$$
$$\forall j \leq m^*_{2,\nu}, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2 = [\![\left(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j} - \xi_j\omega_\ell v_{\ell,j}\right)\widetilde{\mathbf{B}}^*_j]\!]_2$$
$$= [\![(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j} - \widetilde{\xi}'_j\omega_\ell v_{\ell,j} - \frac{\delta_\nu\omega_\ell(w^{(1)}_{\nu,j}v_{\ell,j} - w^{(0)}_{\nu,j}v_{\ell,j})}{\widehat{\alpha}_\nu})\widetilde{\mathbf{B}}^*_j]\!]_2$$
$$\forall j > m^*_{2,\nu}, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2 = [\![\left(\omega_\ell v_{\ell,j}, \widetilde{\gamma}_{\ell,j}, \widehat{t}_{\ell,j}\right)\widetilde{\mathbf{B}}^*_j]\!]_2$$

where $\widehat{s}_{\ell,i}, \widehat{t}_{\ell,j} \leftarrow \mathbb{Z}_p$ for all $j \in I_{\mathbf{v}_\ell}, i \in I_{\mathbf{y}_\ell}$. Hence, we set $s_{\ell,i} = \frac{y^0_{\ell,i}x^0_{\nu,i}}{\widehat{\alpha}_\nu} - \frac{y^1_{\ell,i}x^1_{\nu,i}}{\widehat{\alpha}_\nu} + \widehat{s}_{\ell,i}$ for $i \le m'_{1,\nu}$ and $t_{\ell,j} = \widehat{t}_{\ell,j} - \delta_\nu \omega_\ell \frac{(w^{(1)}_{\nu,j} - w^{(0)}_{\nu,j})}{\widehat{\alpha}_\nu} v_{\ell,j}$ for $j \in [m^*_{2,\nu}]$. Observe that, $s_{\ell,i}$ for $i \le m'_{1,\nu}$ and $t_{\ell,j}$ for $j \in [m^*_{2,\nu}]$ are independently random elements from $\mathbb{Z}_p$ as there are no condition on $(y^0_{\ell,i}x^0_{\nu,i} - y^1_{\ell,i}x^1_{\nu,i})$ and $(w^{(1)}_{\nu,j} - w^{(0)}_{\nu,j})v_{\ell,j}$ where $\widehat{s}_{\ell,i}$ and $\widehat{t}_{\ell,j}$ are independently random elements in $\mathbb{Z}_p$. Thus, the fourth component of $\mathbf{k}^{(1)}_{\ell,i}$ and the third component of $\mathbf{k}^{(2)}_{\ell,j}$ are independently random elements in $\mathbb{Z}_p$ since $\widehat{s}_{\ell,i}, \widehat{t}_{\ell,j}$ for $i > [m'_{1,\nu}], j > [m^*_{2,\nu}]$ are independent of $s_{\ell,i}$ for $i \le [m'_{1,\nu}]$ and $t_{\ell,j}$ for $j \le [m^*_{2,\nu}]$. Thus, $\{[\![\mathbf{k}^{(1)}_{\ell,i}]\!]_2, [\![\mathbf{k}^{(2)}_{\ell,j}]\!]_2\}$ are distributed properly.

Therefore, Game 3-ν-1-5 and Game 3-ν-1-6 are indistinguishable except a negligible probability i.e., $|\Pr(\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}6}) - \Pr(\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5})| = \Pr(\widetilde{\mathsf{E}}_{3\text{-}\nu\text{-}1\text{-}5}) \cdot \Pr(\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}5}|\widetilde{\mathsf{E}}_{3\text{-}\nu\text{-}1\text{-}5}) - \Pr(\widetilde{\mathsf{E}}_{3\text{-}\nu\text{-}1\text{-}6}) \cdot \Pr(\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}6}|\widetilde{\mathsf{E}}_{3\text{-}\nu\text{-}1\text{-}6}) \le 2^{\Omega(\lambda)}$. Here we utilize the fact that $\mathcal{A}$'s view is identical before ν^{th} ciphertext query (i.e. $\Pr(\widetilde{\mathsf{E}}_{3\text{-}\nu\text{-}1\text{-}5}) = \Pr(\widetilde{\mathsf{E}}_{3\text{-}\nu\text{-}1\text{-}6})$). This establishes the claim.

Claim 7. $|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}6}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}1\text{-}7}]| \le \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof 14. Follows from Claims 5, 4, 3 and 2.

Lemma 8. $|\Pr[\mathsf{E}_{3\text{-}\nu\text{-}2}] - \Pr[\mathsf{E}_{3\text{-}\nu\text{-}3}]| \le \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof 15. The proof is similar to Lemma 6.

Lemma 9. $|\Pr[\mathsf{E}_{3\text{-}\mathsf{Q}_{\mathsf{CT}}\text{-}3}] - \Pr[\mathsf{E}_4]| \le 2^{-\Omega(\lambda)}$.

By a simple basis transformation, this Lemma holds.

Lemma 10. $|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_5]| \le \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof 16. The proof is similar to Lemma 1.

This completes the proof of Theorem 1.

Appendix B. Security Analysis of Our UAB-IPFE

To prove the above Theorem 2, we use the following Lemma from [28].

Lemma 11 (Masking Lemma). [28] Let A be an LSSS-realizable over a set of attributes $\mathsf{Att} \subseteq \mathbb{Z}_q$. We denote by $\mathsf{List\text{-}Att}(\mathbb{A})$ the list of attributes appearing in $\mathbb{A}$ and by P the cardinality of $\mathsf{List\text{-}Att}(\mathbb{A})$. Let $\mathsf{S} \subseteq \mathsf{Att}$ be a set of attributes with $(\mathbf{H}, \mathbf{H}^*) \leftarrow \mathcal{G}_{\mathsf{OB.Gen}}(\mathbb{Z}_p^8)$ and $(\mathbf{F}, \mathbf{F}^*) \leftarrow \mathcal{G}_{\mathsf{OB.Gen}}(\mathbb{Z}_p^2)$. The vectors $([\![\mathbf{h}_1]\!]_1, [\![\mathbf{f}_1]\!]_1, [\![\mathbf{f}_2]\!]_1, [\![\mathbf{f}_3]\!]_1)$ are public, while all other vectors are secret. Suppose we have two random labeling $(a_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a_0}(\mathbb{A})$ and $(a'_j)_{j \in \mathsf{List\text{-}Att}(\mathbb{A})} \leftarrow \Lambda_{a'_0}(\mathbb{A})$ for $a_0, a'_0 \leftarrow \mathbb{Z}_p$. Then Masking Lemma is to guess the bit β, given the following distribution

$$\begin{aligned}
\mathcal{D} &= (\mathsf{G} \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(\lambda), ([\![\mathbf{h}_1]\!]_1, [\![\mathbf{f}_1]\!]_1, [\![\mathbf{f}_2]\!]_1, [\![\mathbf{f}_3]\!]_1), \\
&\qquad [\![\mathbf{h}^*_1]\!]_2, [\![\mathbf{f}^*_1]\!]_2, [\![\mathbf{f}^*_2]\!]_2, [\![\mathbf{f}^*_3]\!]_2) \\
\mathbf{k}^\beta_j &= (\pi_j(j,1), a_i z, 0, 0, \beta \cdot a'_j yz/v_j, 0, 0)\mathbf{F}^* \\
&\qquad \forall j \in \mathsf{List\text{-}Att}(\mathbb{A}) \\
\mathbf{c}^\beta_j &= (\sigma_j(1,-j), \psi, 0, 0, \beta \cdot \tau v_j x, 0, 0)\mathbf{F} \ \forall j \in \mathsf{S} \\
\mathbf{k}^\beta_{\mathsf{root}} &= (a_0 z, \beta \cdot a'_0 yz)\mathbf{H}^* \\
\mathbf{c}^\beta_{\mathsf{root}} &= (\psi, \beta \cdot \tau x)\mathbf{H} \\
\mathcal{U}_\beta &= \Big(\{[\![\mathbf{k}^\beta_j]\!]_2, [\![\mathbf{c}^\beta_j]\!]_1\}_j, ([\![\mathbf{k}^\beta_{\mathsf{root}}]\!]_2, [\![\mathbf{c}^\beta_{\mathsf{root}}]\!]_1)\Big)
\end{aligned}$$

where $x, y \in \mathbb{Z}_p$, $\sigma_j, z, \tau, \pi_j, v_j, \tau, \psi \leftarrow \mathbb{Z}_p$. For any PPT adversary $\mathcal{A}$, $\exists$ a PPT adversary $\mathcal{B}$ for the SXDH assumption such that

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ML}}_{\mathcal{A}}(\lambda) &= |\Pr[\mathcal{A}(\mathcal{D}, \mathcal{U}_0) = 1] - \Pr[\mathcal{A}(\mathcal{D}, \mathcal{U}_1) = 1]| \\
&\le P \cdot (6P+3) + 2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda).
\end{aligned}$$

In the following, we prove that the adversary's advantage for all the consecutive games is negligible in the security parameter λ which completes the proof of the Theorem 2.

Lemma 12. $|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_1]| \le 2\mathsf{Q}_{\mathsf{SK}} \cdot (P(6P+3)+2) \cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda)$

Proof 17. This proof follows from AB-IPFE of Nguyen et al. [28] using Masking Lemma 11 for $x = 1$ and $y = \langle \Delta \mathbf{x}, \mathbf{y}_\ell \rangle$ to achieve the selective security. We consider a sequence of games indexed by $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$ corresponding to Q_{SK} many functional key queries. We consider Game 0-ℓ be the first semi-functional secret key form of Game 1 and denotes Game 0 $\equiv$ Game 0-0 $\ldots$ Game 0-Q_{SK} $\equiv$ Game 1. Consequently, for $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$, the Game 0-$(\ell-1)$ is understood as predecessor of Game 0-ℓ in the Game sequence $\{$Game 0-0, Game 0-1, $\ldots$, Game 0-$\mathsf{Q}_{\mathsf{SK}}\}$. The sequence of games from Game 0-$(\ell-1)$ to Game 0-ℓ is depicted in following.

Game 0-$(\ell-1)$-0: As previously mentioned, Game 0-$(\ell-1)$-0 is the same as Game 0-$(\ell-1)$ where the challenge ciphertext and secret keys components are as follows:

$$\begin{aligned}
[\![\mathbf{c}_{\mathsf{ac},j}]\!]_1 &= [\![(\sigma_j(1,-j), \psi, 0, 0, 0, 0, 0)\mathbf{F}]\!]_1 \\
[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 &= [\![(\omega, \mu\omega, \psi, o)\mathbf{H}]\!]_1 \\
[\![t^{(\beta)}_i]\!]_T &= [\![x^{(\beta)}_i]\!]_T - e(g_1, \omega[\![u_i]\!]_2) - e([\![\mu]\!]_1, \omega[\![s_i]\!]_2) \\
[\![\mathbf{k}_{\ell,\mathsf{ac},j}]\!]_2 &= [\![\pi_{\ell,j}(j,1), a_{\ell,j} \cdot z, 0, 0, 0, 0, 0)\mathbf{F}^*]\!]_2 \\
[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 &= [\![(-\sum_{i \in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\sum_{i \in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}, a_{\ell,0}z, 0)\mathbf{H}^*]\!]_2
\end{aligned}$$

with $a_{\ell,0} \leftarrow \mathbb{Z}_p$, $(a_{\ell,j})_{j \in \mathsf{List\text{-}att}(\mathbb{A})} \leftarrow \Lambda_{a_{\ell,0}}(\mathbb{A})$.

Game 0-$(\ell-1)$-1: Game 0-$(\ell-1)$-1 is identical with Game 0-$(\ell-1)$-0 except that the following changes in the challenge ciphertext and secret keys components as follows:

$$[\![\mathbf{c}_{\mathsf{ac},j}]\!]_1 = [\![(\sigma_j(1,-j),\psi,0,0,\boxed{\tau z_j},0,0)\mathbf{F}]\!]_1$$
$$[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 = [\![(\omega,\mu\omega,\psi,\boxed{\tau})\mathbf{H}]\!]_1$$
$$[\![\mathbf{k}_{\ell,\mathsf{ac},j}]\!]_2 = [\![\pi_{\ell,j}(j,1),a_{\ell,j}\cdot z,0,0,\boxed{a'_{\ell,j}\delta_\ell/z_j},0,0)\mathbf{F}^*]\!]_2$$
$$[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 = [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}, a_{\ell,0}z, \boxed{a'_{\ell,0}\delta_\ell})\mathbf{H}^*]\!]_2$$

where $\tau, z_j, a'_{\ell,0}, a'_{\ell,j} \leftarrow \mathbb{Z}_p$ with $\Delta\mathbf{x} = \mathbf{x}^{(0)} - \mathbf{x}^{(1)}$.

Game 0-$(\ell-1)$-2**:** This Game 0-$(\ell-1)$-2 is similar to Game 0-$(\ell-1)$-1 except that the following secret key component as follows:

$$[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 = [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}, a_{\ell,0}z, \boxed{(a'_{\ell,0}+r'_{\ell,0})\delta_\ell})\mathbf{H}^*]\!]_2$$

where $r'_{\ell,0} \leftarrow \mathbb{Z}_p$.

Game 0-$(\ell-1)$-3**:** Game 0-$(\ell-1)$-3 is the same as Game 0-$(\ell-1)$-2 except that the challenge ciphertext and the secret key components are generated as follows:

$$[\![\mathbf{c}_{\mathsf{ac},j}]\!]_1 = [\![(\sigma_j(1,-j),\psi,0,0,\boxed{0},0,0)\mathbf{F}]\!]_1$$
$$[\![\mathbf{k}_{\ell,\mathsf{ac},j}]\!]_2 = [\![\pi_{\ell,j}(j,1),a_{\ell,j}\cdot z,0,0,\boxed{0},0,0)\mathbf{F}^*]\!]_2$$
$$[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 = [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}, a_{\ell,0}z, \boxed{r'_{\ell,0}\delta_\ell})\mathbf{H}^*]\!]_2$$

where $r'_{\ell,0} \leftarrow \mathbb{Z}_p$. In the following, we show that the intermediate game transition between Game 0-$(\ell-1)$-0 to Game 0-$(\ell-1)$-3 relying on the hardness of SXDH assumption G.

Game 0-$(\ell-1)$-$0 \approx$ **Game** 0-$(\ell-1)$-1**:** In this Game the functional key is still capable to decrypt the challenge ciphertext if the key policy is satisfied. By applying the Masking Lemma 11 as described in that context, we get

$$|\mathsf{Adv}_{0\text{-}(\ell-1)\text{-}1}(\lambda) - \mathsf{Adv}_{0\text{-}(\ell-1)\text{-}0}(\lambda)| \leq [P\cdot(6P+3)+2]\cdot \mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(\lambda)$$

Game 0-$(\ell-1)$-$1 \approx$ **Game** 0-$(\ell-1)$-2**:** In this game, we randomize $a'_{\ell,0}$ in the secret key component $\mathbf{k}_{\ell,\mathsf{fe}}$ by uniform value $r_{\ell,0} \leftarrow \mathbb{Z}_p$. Here, we categorize two cases based on the inner product value $\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle$ zero or non-zero.

- **For** $\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle \neq 0$**:** From the security definition, we have $\mathbb{A}(\mathsf{S}) = 0$, i.e., there is no way to find a reconstruction vector $\mathbf{c} = (c_j)_j$ for an authorized set $A \subseteq \mathsf{S}$. More precisely, there are not enough $a'_{\ell,j}\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle/z_j$ from the ℓ-th functional key to recover

$$\sum_{j\in A}\frac{c_j a'_{\ell,j}}{z_j}\cdot\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle\tau z_j = \tau a'_{\ell,0}\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle$$

Thus, $(a'_{\ell,j})_j$ is a random labeling of $a'_{\ell,0}$ using LSSS of the access structure $\mathbb{A}$ and $\tau, z_j \leftarrow \mathbb{Z}_p$. Therefore, for all $(a'_{\ell,j})_j$ are randomized into $a'_{\ell,j}/z_j$ and become independent uniformly random values. In this case, masking $a'_{\ell,0}$ by $r'_{\ell,0}$ are perfectly indistinguishable under $\mathcal{A}$'s view.

- **For** $\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle = 0$**:** Changing $a'_{\ell,0}$ to $a'_{\ell,0}+r'_{\ell,0}$ does not affect the view of $\mathcal{A}$. The given keys are successful decrypting the challenge ciphertext in both games which we discuss in the following.

$$\begin{aligned}
&e([\![\mathbf{c}_{\mathsf{fe}}]\!]_1, [\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2)\\
&= [\![-\omega\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i} - \mu\omega\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i} + \psi a_{\ell,0}z + \tau(a'_{\ell,0}+r'_{\ell,0})\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle]\!]_T\\
&= [\![-\omega\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i} - \mu\omega\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i} + \psi a_{\ell,0}z]\!]_T\\
&e([\![\mathbf{c}_{\mathsf{ac},j}]\!]_1, \sum_{j\in A} c_j[\![\mathbf{k}_{\ell,\mathsf{ac},j}]\!]_2)\\
&= [\![\psi z\sum_{j\in A} a_{\ell,j} + \sum_{j\in A}\tau z_j\frac{a'_{\ell,j}c_j}{z_j}\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle]\!]_T\\
&= [\![\psi z a_{\ell,0}]\!]_T
\end{aligned}$$

where $A \subseteq \mathsf{S}$ and $(c_j)_j$ is obtained from the LSSS. In total changing $a'_{\ell,0}\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle$ to $(a'_{\ell,0}+r'_{\ell,0})\langle\Delta\mathbf{x},\mathbf{y}_\ell\rangle$ is perfectly indistinguishable under adversarial view. Thus, we have

$$\mathsf{Adv}_{0\text{-}(\ell-1)\text{-}1}(\lambda) = \mathsf{Adv}_{0\text{-}(\ell-1)\text{-}2}(\lambda)$$

Game 0-$(\ell-1)$-$2 \approx$ **Game** 0-$(\ell-1)$-3**:** Similar to the game transformation of Game 0-$(\ell-1)$-1 to Game 0-$(\ell-1)$-0, the game follows the same transformation strategy. Therefore,

$$|\mathsf{Adv}_{0\text{-}(\ell-1)\text{-}3}(\lambda) - \mathsf{Adv}_{0\text{-}(\ell-1)\text{-}2}(\lambda)| \leq (P\cdot(6P+3)+2)\cdot\mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(\lambda)$$

We perform the above sequence of games for each ℓ-th functional key. At the end, we arrive at Game 0-$\mathsf{Q} \equiv$ Game 1. Thus the difference between Game 0 and Game 1 is:

$$|\mathsf{Adv}_1(\lambda) - \mathsf{Adv}_0(\lambda)| \leq 2\mathsf{Q}_{\mathsf{SK}}\cdot(P(6P+3)+2)\cdot\mathsf{Adv}^{\mathsf{SXDH}}_{\mathbb{G}_1,\mathbb{G}_2}(\lambda)$$

Lemma 13. $|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| \leq \frac{1}{p}$

Proof 18. Let us choose a matrix $\mathbf{J} \leftarrow \mathsf{GL}_4(\mathbb{Z}_p)$ and set the random dual orthonormal bases $(\mathbf{H},\mathbf{H}^*)$ such that

$$\mathbf{H} = \begin{bmatrix} 1 & & \\ & \mu^{-1} & \\ & & \mathbf{I}_2 \end{bmatrix}\mathbf{J};\quad \mathbf{H}^* = \begin{bmatrix} 1 & & \\ & \mu & \\ & & \mathbf{I}_2 \end{bmatrix}\mathbf{J}^*$$

where $\mu \leftarrow \mathbb{Z}_p$. In the following, we simulate the ℓ-th secret key component $[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2$ and the challenge ciphertext component $[\![\mathbf{c}_{\mathsf{fe}}]\!]_1$. Now

$$
\begin{aligned}
[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 &= [\![(\omega, \mu\omega, \psi, \tau)\mathbf{H}]\!]_1 = [\![(\omega, \omega, \psi, \tau)\mathbf{J}]\!]_1 \\
[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_1 &= [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}, a_{\ell,0}z, \\
&\qquad\qquad r'_{\ell,0}\langle\Delta\mathbf{x}, \mathbf{y}_\ell\rangle)\mathbf{H}^*]\!]_2 \\
&= [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\mu\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}, a_{\ell,0}z, \\
&\qquad\qquad r'_{\ell,0}\langle\Delta\mathbf{x}, \mathbf{y}_\ell\rangle)\mathbf{J}^*]\!]_2
\end{aligned}
$$

From the above, it is clear that the ciphertext and ℓ-th functional secret keys are simulated the same as Game 1 except for $\mu = 0$, i.e., except for the probability $\frac{1}{p}$. Thus, $|\mathsf{Adv}_1(\lambda) - \mathsf{Adv}_2(\lambda)| \leq \frac{1}{p}$.

Lemma 14. $|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_2]| \leq \mathsf{negl}(\lambda)$

Proof 19. Let $\mathbf{B} \leftarrow \mathsf{GL}_4(\mathbb{Z}_p)$ and set the matrix

$$
\mathbf{H} = \begin{bmatrix} 1 & & & \\ & 1 & & -1 \\ & & 1 & \\ & & & 1 \end{bmatrix}\mathbf{B}, \quad \mathbf{H}^* = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & 1 & & 1 \end{bmatrix}\mathbf{B}^*
$$

where $\mathbf{B}^* = (\mathbf{B}^{-1})^\top$. For $I_{\mathbf{x}} = I_{\mathbf{y}_\ell}$, then using the basis $(\mathbf{B}, \mathbf{B}^*)$, the secret key component and ciphertext component $\mathbf{k}_{\ell,\mathsf{fe}}, \mathbf{c}_{\mathsf{fe}}$ are simulated as follows:

$$
\begin{aligned}
[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 &= [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\mu\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i}, a_{\ell,0}z, \\
&\qquad\qquad r'_{\ell,0}\langle\Delta\mathbf{x}, \mathbf{y}_\ell\rangle)\mathbf{H}^*]\!]_2 \\
&= [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u_{\ell,i}, -\mu\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s_{\ell,i} + r'_{\ell,0}\langle\Delta\mathbf{x}, \mathbf{y}_\ell\rangle, \\
&\qquad\qquad a_{\ell,0}z, r'_{\ell,0}\langle\Delta\mathbf{x}, \mathbf{y}_\ell\rangle)\mathbf{B}^*]\!]_2 \\
[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 &= [\![(\omega, \omega, \psi, \tau)\mathbf{H}]\!]_1 \\
&= [\![(\omega, \omega, \psi, \tau - \omega)\mathbf{B}]\!]_1 \\
&= [\![(\omega, \omega, \psi, \tau')\mathbf{B}]\!]_1
\end{aligned}
$$

where implicitly set $\tau' = \tau - \omega$.

Lemma 15. $|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_3]| \leq \mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{B}}(\lambda)$

Proof 20. To show the game transitions between Game 2 and Game 3, we construct an adversary $\mathcal{B}$ that break the DBDH assumption. Let $\mathcal{B}$ obtains the DBDH challenge instance $(\mathsf{G}, [\![a]\!]_1, [\![b]\!]_1, [\![a]\!]_2, [\![c]\!]_2, [\![d]\!]_T)$ where $a, b, c \leftarrow \mathbb{Z}_p$ and d is either abc or $d \leftarrow \mathbb{Z}_p$. Let the challenge index set $|I_{\mathbf{x}}| = n$ and consider a set $\{\mathbf{w} = (w_i)_i : i \in I_{\mathbf{x}}, w_i \in \mathbb{Z}_p\}$ with the vector space $\mathbb{Z}_q^n$ where $m : I_{\mathbf{x}} \to [n]$ maps the challenge indices to those in $\mathbb{Z}_p^n$.

Now, we apply the proof techniques of Abdalla et al. [3], [16], using the information of $\Delta\mathbf{x} = \mathbf{x}^{(0)} - \mathbf{x}^{(1)}$, the challenger $\mathcal{B}$ generates a basis $(\mathbf{z}_\kappa)_{\kappa\in[n-1]}$ of $\Delta\mathbf{x}^\perp = \{\mathbf{z}_\kappa : \langle\Delta\mathbf{x}, \mathbf{z}_\kappa\rangle = 0\}$. Based on these $\mathbf{z}_i$'s, $\mathcal{B}$ picks $n-1$ random scalars $(\rho_i)_{i\in[n-1]}$ and $(\Delta\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{n-1})$ is basis of $\mathbb{Z}_p^n$. Thus, any canonical vectors $\mathbf{e}_i$ can be represented as $\mathbf{e}_i = \alpha_i \cdot \Delta\mathbf{x} + \sum_{\kappa\in[n-1]} \lambda_{i,\kappa} \cdot \mathbf{z}_\kappa$ where $\alpha_i, \lambda_{i,\kappa} \in \mathbb{Z}_p$ for all $i \in [n], \kappa \in [n-1]$. Now, the challenger $\mathcal{B}$ can simulate the adversarial view as follows:

Public key simulation. Using the challenge DBDH instances, we set the master public key as $\mathsf{MPK} = \big(\{[\![\mathbf{f}_i]\!]_1\}_{i=1}^3, [\![\mathbf{h}_1 + a\mathbf{h}_2]\!]_1, [\![\mathbf{h}_3]\!]_1, [\![a]\!]_1\big)$.

Random Oracle Calls: On input $i \in I_{\mathbf{y}_\ell}$, if $i \notin I_{\mathbf{x}}$, (i.e., $I_{\mathbf{x}} \neq I_{\mathbf{y}_\ell}$) returns two random group elements of $\mathbb{G}_2$ using both the random oracle $\mathcal{H}_1$ and $\mathcal{H}_2$ and set these as $[\![u'_i]\!]_2$ and $[\![s'_i]\!]_2$ respectively. On input $i \in I_{\mathbf{x}}$, $\mathcal{B}$ responses as

$$
\begin{aligned}
\mathcal{H}_1(i|I_{\mathbf{x}}) &= \alpha_{m(i)}[\![a]\!]_2 \cdot \Big(\prod_{\kappa\in[n-1]} \lambda_{m(i),\kappa}[\![\rho_\kappa]\!]_2\Big); \\
\mathcal{H}_2(i|I_{\mathbf{x}}) &= \alpha_{m(i)}[\![c]\!]_2 \cdot \Big(\prod_{\kappa\in[n-1]} \lambda_{m(i),\kappa}[\![\rho_\kappa]\!]_2\Big)
\end{aligned}
$$

where $\rho_\kappa \leftarrow \mathbb{Z}_p$ for $\kappa \in [n-1]$.

Ciphertext simulation: The challenger $\mathcal{B}$ uniformly chooses $\beta \leftarrow \{0, 1\}$ and generates the challenge ciphertext corresponding to the challenge message vector $\mathbf{x}^{(\beta)} = (x_i^{(\beta)})_{i\in I_{\mathbf{x}}}$ from the given DBDH instances as follows:

$$
\begin{aligned}
[\![\mathbf{c}_{\mathsf{ac},j}]\!]_1 &= [\![(\sigma_j(1, -j), \psi, 0, 0, 0, 0, 0)\mathbf{F}]\!]_1 \\
[\![\mathbf{c}_{\mathsf{fe}}]\!]_1 &= [\![(b, b, \psi, \tau)\mathbf{H}]\!]_1 \\
[\![t_i^{(\beta)}]\!]_T &= [\![x_i^{(\beta)}]\!]_T \cdot \alpha_{m(i)} e([\![b]\!]_1, [\![a]\!]_2)^{-1} \cdot \\
&\quad \Big(\sum_{\kappa\in[n-1]} \lambda_{m(i),\kappa}\Big) e([\![b]\!]_1, [\![\rho_\kappa]\!]_2)^{-1} \cdot \alpha_{m(i)}([\![d]\!]_T)^{-1} \\
&\quad \cdot \Big(\sum_{\kappa\in[n-1]} \lambda_{m(i),\kappa}\rho_k\Big) e([\![b]\!]_1, [\![a]\!]_2)^{-1}
\end{aligned}
$$

Decryption keys simulation: The adversary $\mathcal{A}$ can query ℓ-th secret keys $\mathsf{SK}_{\mathbf{y}_\ell,\mathbb{A}} = (\{[\![\mathbf{k}_{\ell,\mathsf{ac},j}]\!]_2\}_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}, [\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2)$ corresponding to $(\mathbf{y}_\ell = (y_{\ell,i})_{i\in I_{\mathbf{y}_\ell}}, \mathbb{A})$. Make those calls to the random oracles $\mathcal{H}_1$, $\mathcal{H}_2$ that haven't been made for inputs $i \in I_{\mathbf{y}_\ell}$. For $I_{\mathbf{x}} \neq I_{\mathbf{y}_\ell}$ or $\mathbb{A}(\mathsf{S}) = 0$, the challenger simply returns the secret keys components $(\{[\![\mathbf{k}_{\ell,\mathsf{ac},j}]\!]_2\}_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}, \{[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2\})$ as

$$
\begin{aligned}
[\![\mathbf{k}_{\ell,\mathsf{ac},j}]\!]_2 &= [\![\pi_{\ell,j}(j, 1), a_{\ell,j} \cdot z, 0, 0, 0, 0, 0)\mathbf{F}^*]\!]_2 \\
[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 &= [\![(-\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}u'_{\ell,i}, -a\sum_{i\in I_{\mathbf{y}_\ell}} y_{\ell,i}s'_{\ell,i}, \\
&\qquad\qquad a_{\ell,0}z, 0)\mathbf{H}^*]\!]_2
\end{aligned}
$$

For $(I_{\mathbf{x}} = I_{\mathbf{y}_\ell}) \wedge (\mathbb{A}(\mathsf{S}) = 0)$, we simualte the

secret key components as follows:

$$[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 = [\![(-a\delta_\ell - \sum_{\kappa\in[n-1]} \lambda_{m(i),\kappa}\rho_\kappa y_i, r''_{\ell,0}\delta_\ell - a\sum_{\kappa\in[n-1]} \lambda_{m(i),\kappa}\rho_\kappa y_i, a_{\ell,0}z, 0)\mathbf{H}^*]\!]_2$$

where we implicitly set $r''_{\ell,0} = r'_{\ell,0} - ac$, i.e., uniformly random in $\mathbb{Z}_p$. Otherwise for $I_{\mathbf{x}} = I_{\mathbf{y}_\ell}$ and $\mathbb{A}(\mathsf{S}) = 1$, we can express the key vector $\mathbf{y}_\ell = (y_{\ell,i})_{i\in I_{\mathbf{y}_\ell}}$ as $(y_{\ell,i})_{i\in I_{\mathbf{y}_\ell}} = \vartheta\cdot(\mathbf{x}^{(0)} - \mathbf{x}^{(1)}) + \sum_{\iota\in[n-1]}\varepsilon_\iota\cdot\mathbf{z}_\iota$ where $\vartheta \leftarrow \mathbb{Z}_p$ and $\varepsilon_\kappa \leftarrow \mathbb{Z}_p$ for all $\kappa\in[n-1]$. Here the coefficient ϑ of $\mathbf{x}^{(0)} - \mathbf{x}^{(1)}$ in the decomposition of $\mathbf{y}_\ell$ for which a key has been queried is zero, i.e.,

$$\langle \mathbf{x}^{(0)} - \mathbf{x}^{(1)}, \mathbf{y}_\ell\rangle = \vartheta\cdot\langle \mathbf{x}^{(0)} - \mathbf{x}^{(1)}, \mathbf{x}^{(0)} - \mathbf{x}^{(1)}\rangle$$
$$\implies \vartheta = 0 \ (\text{as } \langle\Delta\mathbf{x}, \mathbf{y}_\ell\rangle = \delta_\ell = 0)$$

Then the challenger simply returns the secret keys components $\{[\![\mathbf{k}_{\ell,j}]\!]_2\}_{j\in\mathsf{List\text{-}Att}(\mathbb{A})}$ and $\{[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2\}$ as

$$[\![\mathbf{k}_{\ell,\mathsf{ac},j}]\!]_2 = [\![\pi_{\ell,j}(j,1), a_{\ell,j}\cdot z, 0,0,0,0,0)\mathbf{F}^*]\!]_2$$
$$[\![\mathbf{k}_{\ell,\mathsf{fe}}]\!]_2 = [\![(\Big(-\sum_{\iota\in[n-1]}\varepsilon_\iota\Big(\sum_{\kappa\in[n-1]}\lambda_{\iota,\kappa}\rho_\kappa\Big)\Big),$$
$$\Big(-\sum_{\iota\in[n-1]}\varepsilon_\iota\Big(\sum_{\kappa\in[n-1]}\lambda_{\iota,\kappa}\rho_\kappa\Big)\Big)a, a_{\ell,0}z, r'_{\ell,0}\delta_\ell)\mathbf{H}^*]\!]_2$$

The simulation of the secret keys are correctly computed unless the adversary is not admissible. At the end of the simulation if $\mathcal{A}$ correctly guesses the challenge bit β, (the tuple is a proper BDH tuple) $\mathcal{B}$ guesses that $d = abc$ otherwise, it guesses that d is uniformly random. According to the DBDH assumption, $\mathcal{A}$ is unable to differentiate between the these scenarios, and as a result, the adversary $\mathcal{A}$ does not possess any information on the challenge bit β.
This completes the proof of Theorem 2.