Reinforcing Privacy in Cloud Computing via Adaptively Secure Non-zero Inner Product Encryption and Anonymous Identity-based Revocation in Unbounded Setting

Subhranil Dutta[a,*], Tapas Pal[b], Ratna Dutta[a]

[a]*Department of Mathematics, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal, India*
[b]*KASTEL Security Research Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany*

Abstract

Cloud computing serves as an advanced computing technology to support Internet services and has numerous applications, including medical fields, online data storage, social network, big data analysis, and e-learning platforms. *Inner product encryption* (IPE) is a promising cryptographic primitive that facilitates access control over outsourced encrypted data by restricting decryption capability via an inner product relation. Most of the existing IPEs are built assuming a priori bound on the length of attribute vectors associated with the plane data. A situation may occur when adding new attributes is essential due to upgradation. Thus it is desirable to construct IPE with variable length attribute vectors instead of fixing the length at the setup phase. Such IPEs are more flexible while encrypting data and consequently have more real-life applications. In the following, we make our contributions:

– We design the *first* efficient *adaptively payload-hiding* secure *unbounded non-zero IPE* (UNIPE) scheme from the standard *symmetric external Diffie-Hellman* (SXDH) assumption and prove security in the standard model.

– We propose a generic construction of an *adaptively weak attribute-hiding* UNIPE from an *unbounded inner product functional encryption* (UIPFE) scheme.

*Corresponding author

URL: subhranildutta@iitkgp.ac.in (Subhranil Dutta), tapas.pal@kit.edu (Tapas Pal), ratna@maths.iitkgp.ac.in (Ratna Dutta)

We also instantiate UNIPE from the SXDH assumption and prove security in the standard model.

– Finally, we present the *first unbounded anonymous identity-based revocation* (UAnon-IBRV) scheme using our generic UNIPE as the building block. The anonymity feature ensures that the ciphertext reveals no information about the revoking sets of the system.

Keywords: Unbounded, inner product encryption, attribute-hiding, payload-hiding.

---

1. Introduction

Functional encryption (FE) [1, 2, 3] is a modern cryptographic prototype that is an advanced version of usual *public-key encryption* (PKE). The beauty of FE is that it incorporates nearly every current cryptographic encryption scheme, including *identity-based encryption* (IBE) [4, 5, 6], *attribute-based encryption*(ABE) [7, 8, 1, 9], and *predicate encryption* (PE) [10, 11]. FE generates a legitimate secret key sk_f corresponding to any function f, which decrypts the ciphertext ct_m associated with a message m and outputs $f(m)$ instead of the original message m in contrast to the traditional PKE. Sahai and Waters [5] initiated the idea of ABE, a particular class of FE that produces the user's secret keys only for boolean functions. Over the decades, many variants of ABEs have emerged in realistic scenarios. In ABE, a legitimate secret key corresponds to a predicate v, and ciphertext is associated with an attribute-message pair (u, m) such that the decryptor successfully outputs the message m whenever a pre-defined relationship between u and v holds. In this *key-policy* based ABE (KP-ABE) [12], ciphertext and secret key are associated with attributes and access policies, respectively. *Ciphertext-policy* ABE (CP-ABE) [7] is another variant of ABE in which secret keys are associated with attributes and ciphertext correspond with access policies.

Katz et al. *first* studied a particular class of ABE known as *inner product encryption* (IPE) [10]. In IPE, a secret key $\mathsf{sk}_{\boldsymbol{v}}$ corresponding to a predicate vector $\boldsymbol{v}$, that successfully decrypts the ciphertext $\mathsf{ct}_{\boldsymbol{u}}$ associated with the attribute-message pair $(\boldsymbol{u}, m)$ if a linear relation R holds between the attribute vector $\boldsymbol{u}$ and predicate vector $\boldsymbol{v}$. In the case of *zero* IPE (ZIPE), $R(\boldsymbol{u}, \boldsymbol{v}) = 1$ if $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0$ and for *non-zero* IPE (NIPE), $R(\boldsymbol{u}, \boldsymbol{v}) = 1$ if $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \neq 0$. From the security perspective, *payload-hiding* (PH) is the primary requirement that hides the message m inside the ciphertext. A more

enhanced security notion, known as *attribute-hiding* (AH), was introduced by Katz et al. [10] in which any information about the attribute vector $\boldsymbol{u}$ and the message m to an adversary is not revealed from the ciphertext. Let the adversary submit two attribute-message pairs $(\boldsymbol{u}^{(b)}, m^{(b)})$, for $b \in \{0, 1\}$. Given a ciphertext corresponding to a challenge attribute-message pair $(\boldsymbol{u}^{(b)}, m^{(b)})$ a probabilistic polynomial time adversary can guess the challenge bit b with probability at most $1/2$. In *weakly attribute-hiding* (WAH), the adversary is only allowed to query for the secret keys $\mathsf{sk}_{\boldsymbol{v}}$ associated with predicate vectors $\boldsymbol{v}$ whenever $R(\boldsymbol{u}^{(0)}, \boldsymbol{v}) = R(\boldsymbol{u}^{(1)}, \boldsymbol{v}) = 0$, i.e., the adversary is not permitted to ask for a secret key capable of decrypting the challenge ciphertext. In contrast, *fully attribute-hiding* (FAH) enables security for more robust adversaries, ensuring that the challenger answers polynomially many secret key queries $\mathsf{sk}_{\boldsymbol{v}}$ associated with predicate vector $\boldsymbol{v}$ for which $R(\boldsymbol{u}^{(0)}, \boldsymbol{v}) = R(\boldsymbol{u}^{(1)}, \boldsymbol{v})$ if $m^{(0)} = m^{(1)}$ otherwise $R(\boldsymbol{u}^{(0)}, \boldsymbol{v}) = 0 = R(\boldsymbol{u}^{(1)}, \boldsymbol{v})$.

1.1. Motivation of Unbounded IPE

Almost every prior study on IPE suffered from the size of the system parameters, which depended on the length of the attribute and predicate vectors. It is practically inappropriate since the length n of the corresponding attribute/predicate vectors may not have been previously known while generating the system parameters. A general technique to address this issue is to set a huge upper bound on n while generating these parameters. However, it is not desirable as the parameter size grows linearly with the priori bound on n, but the predicate or attribute vector length could be much less than this upper bound on n. It is challenging to develop an IPE where system parameters are independent of the length n and choose n while generating keys or encrypting messages depending on the requirement. The existing UZIPEs proposed by Okamoto et al. [8] and Dutta et al. [13] are based on the *decisional linear* (DLIN) and the SXDH assumptions, respectively, and provide adaptive full attribute-hiding security. However, there is no NIPE in the unbounded setting with adaptive security.

1.2. Applications of UNIPE

IPE has many practical applications, including evaluations of polynomials, testing membership, and disjunction/conjunctions of equality tests [10, 3], and so on. The unbounded property enables the construction of realistic, storage-friendly and efficient primitives from NIPE, such as *unbounded*

anonymous identity-based revocation (UAnon-IBRV). In UAnon-IBRV ciphertext, a message is encrypted using an arbitrary number of revoked users $\mathcal{R} = \{\mathsf{id}_1, \mathsf{id}_2, \ldots\}$. For successful decryption, the embedded identity id in the secret key requires $\mathsf{id} \notin \mathcal{R}$. A UAnon-IBRV scheme can be built from a UNIPE where the revoked users' set $\mathcal{R}$ is unbounded in the sense that the public parameters are independent of the size of $\mathcal{R}$.

Consider the following two examples that illustrate the practical importance of unbounded attribute-hiding IPE.

(i) Suppose a hospital authority stores in a cloud the medical details, such as name, personal information, health report, etc., in an encrypted form for patients who are undergoing treatment of a drug addict. The identity of the patient is treated as message m and all the health components are represented as an attribute vector $\boldsymbol{u} = (u_1, u_2, \ldots)$. The hospital authority will issue a decryption key so that a selected member of specialist doctors like Psychiatrists and Pathologists can access the encrypted data from the cloud. The doctors' identities are represented as a predicate vector $\boldsymbol{v} = (v_1, v_2, \ldots)$, and the hospital authority issues the decryption key corresponding to $\boldsymbol{v}$ to all the doctors whose identities are included in $\boldsymbol{v}$. While the treatment starts, the medical board may not have a fixed team of specialist physicians. So, the relevant authority should be flexible in adding or updating the medical board of specialist physicians for the treatment required for the drug addict at any time of the diagnostic process. After seeing the medical report, the inner product computation $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ can be modelled to represent the overall conclusion of all the selected specialist doctors about the patient's health. Whenever a patient appears medically fit (i.e., $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \neq 0$) to all the medical board members or the specialist doctors, the patient's identity is disclosed to all such medical board members. Since the doctors' board may be required to get updated over time and it is impossible to predict the number of health components needed to test before the treatment starts, UNIPE is an ideal cryptographic primitive to address this scenario. The attribute-hiding property ensures that the medical report of that drug addict patient is decryptable only by a specified group of doctors and should not be revealed to all. Any doctor outside the medical board, like an eye specialist, cannot decrypt the whole data set. Therefore, one needs to embed the inner product policy into the secret keys. The policy identifies the specially appointed doctors who are eligible to learn about the medical parameters of the

patient's health from these statistics for the treatment of the particular patient. The attribute vector associated with the ciphertext comprises medical test records to compute a specific statistic. The hospital authority encrypts the identity of the patients under the attribute vector and uploads the ciphertext to the cloud. The doctors can download the ciphertext and decrypt it using their secret keys only if the underlying policy is satisfied by the attribute of the ciphertext.

(*ii*) In a defence organization of the territory, the Director-General of Defence wants to pass a secret message to all the sector officers (or major general), such as the navy, marines, armed forces and air forces. It is natural to assume that the size of the revocation list (junior army officers' identities) is not fixed at the time of system parameter generation. The Director-General may use UAnon-IBRV to encrypt the confidential data with the set of all revoked users that varies from time to time in the organization as UAnon-IBRV features public parameters size independent of the revocation list size. The secret data is treated as the message m and the attribute vector $\boldsymbol{u} = (u_1, u_2, \ldots)$, where each $u_i \in \mathcal{R}$. The ciphertext is generated corresponding to the attribute-message pair $(\boldsymbol{u}, m)$. The secret keys are issued corresponding to the predicate vector $\boldsymbol{v} = (v_1, v_2, \ldots)$ which can be the identities of all users in the organization (these identities can be some code name), and the ciphertext is decryptable only by the secret key holders who are not in the revocation list (i.e., $v_i \notin \mathcal{R}$). The revocation list may contain the identities of junior lieutenants or captains who are posted under a major general or sector officer. The anonymity feature of UAnon-IBRV ensures that the identities of junior officers on the revocation list, along with the message, should not be revealed from the ciphertext for the interests of national security. Thus, UAnon-IBRV plays an important role in such scenarios. UAnon-IBRV can be designed from attribute-hiding UNIPE and consequently illustrates an application of attribute-hiding UNIPE.

1.3. Our Contributions

The somewhat unsatisfactory state-of-the-art NIPE in an unbounded setup motivates us to design UNIPE and investigate their applications in real-life scenarios. Our contributions to UNIPE are of independent interest in constructing anonymous identity-based revocation and much more cryptographic primitive in unbounded setup. We list below our contributions to this paper.

• Our UNIPE Scheme: We design a concrete UNIPE scheme that provides adaptive payload-hiding security relying on the standard SXDH assumption. This security ensures that an adversary can query a polynomial number of secret keys at any instant during the experiment. We emphasize that our scheme is the *first* construction of adaptive payload-hiding secure NIPE, which supports unbounded length vectors. We use the framework of dual-pairing vector space (DPVS) [2, 8, 14, 15]. In setup, we choose two pairs of orthonormal dual bases $(\mathbf{W}, \mathbf{W}^*)$ from $\mathbb{Z}_p^{5\times 5} \times \mathbb{Z}_p^{5\times 5}$ and $(\widehat{\mathbf{W}}, \widehat{\mathbf{W}}^*)$ from $\mathbb{Z}_p^{6\times 6} \times \mathbb{Z}_p^{6\times 6}$, where p is a prime. We apply component-wise encoding methodology to encrypt the attribute vector $\boldsymbol{u}$ and use an additional randomness z to bind all encodings so that a secret key can successfully decrypt the ciphertext whenever $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \neq 0$ holds. Let $\boldsymbol{u} = (u_i)_{i\in I_{\boldsymbol{u}}}$ and $\boldsymbol{v} = (v_i)_{i\in I_{\boldsymbol{v}}}$ be the attribute and predicate vectors with the associated non-empty index sets $I_{\boldsymbol{u}}$ and $I_{\boldsymbol{v}}$ respectively. The generalized inner product over the permissive setting is defined as $\sum_{i\in I_{\boldsymbol{v}}} u_i v_i$ whenever $I_{\boldsymbol{v}} \subseteq I_{\boldsymbol{u}}$. We use the notation of double bracket '$[\![s]\!]_\iota$' to express any group element 's' in the exponent power of g_ι, which is a generator of the group G_ι for $\iota \in \{1, 2, T\}$. In the encryption phase, we encrypt the attribute-message pair $(\boldsymbol{u} \in \mathbb{Z}_p^{|I_{\boldsymbol{u}}|}, m)$ and generate the ciphertext components $[\![\boldsymbol{c}_0]\!]_1 = [\![(-\delta, 0, z, 0, \eta)\mathbf{W}]\!]_1, [\![\boldsymbol{c}_i]\!]_1 = [\![(\Psi_i(1, i), \delta u_i, 0, 0, \eta u_i)\widehat{\mathbf{W}}]\!]_1$ and $c = [\![z]\!]_T \cdot m$ where the randomness Ψ_i, z, η, δ are uniformly chosen from $\mathbb{Z}_p$ for all $i \in I_{\boldsymbol{u}}$. The secret key components for the predicate vector $\boldsymbol{v} \in \mathbb{Z}_p^{|I_{\boldsymbol{v}}|}$ are generated as $[\![\boldsymbol{k}_0]\!]_2 = [\![(\omega, 0, 1, \phi, 0)\mathbf{W}^*]\!]_2, [\![\boldsymbol{k}_i]\!]_2 = [\![(\Upsilon_i(-i, 1), \omega v_i, 0, \phi_i, 0)\widehat{\mathbf{W}}^*]\!]_2$ where the randomness ω, ϕ, ϕ_i and Υ_i are chosen uniformly from $\mathbb{Z}_p$. By pairing computation between $[\![\boldsymbol{c}_0]\!]_1, [\![\boldsymbol{k}_0]\!]_2$ and $[\![\boldsymbol{c}_i]\!]_1, [\![\boldsymbol{k}_i]\!]_2$ for all $i \in I_{\boldsymbol{v}}$, the decryptor first gets $[\![z]\!]_T$ using $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \neq 0$ and extracts the message m from the ciphertext component $c = [\![z]\!]_T \cdot m$. Our ciphertext needs $6n + 6$ group elements, whereas the secret key bears $6n' + 5$ group elements, n, n' being the length of attribute and predicate vectors, respectively. For an detailed discussion of our scheme, we refer to Section 3 and security analysis is presented in Theorem 3.1.

• Our UNIPE from UIPFE: We suggest a generic construction of UNIPE employing an *unbounded inner product function encryption* (UIPFE) [16, 17] and prove that it achieves weakly attribute-hiding security. We additionally present a concrete instantiation of UNIPE relying on the SXDH assumption in the standard model. The setup and key extraction algorithms of UNIPE are the same as that of the underlying UIPFE. The ciphertext $\mathsf{ct}_{\boldsymbol{u}}$ of UNIPE corresponding to an attribute-message pair $(\boldsymbol{u}, m)$ consists of two UIPFE ci-

phertexts $\mathsf{ct}_{m\cdot\boldsymbol{u}}$ and $\mathsf{ct}_{\boldsymbol{u}}$, which are encryptions of the vectors $m \cdot \boldsymbol{u}$ and $\boldsymbol{u}$ respectively. In UIPFE, the secret key $\mathsf{sk}_{\boldsymbol{v}}$ can compute $H = m\langle \boldsymbol{u}, \boldsymbol{v}\rangle$, $h = \langle \boldsymbol{u}, \boldsymbol{v}\rangle$ from $\mathsf{ct}_{m\cdot\boldsymbol{u}}$ and $\mathsf{ct}_{\boldsymbol{u}}$ respectively. Computing $H \cdot h^{-1}$, the decryptor outputs the message m if $\langle \boldsymbol{u}, \boldsymbol{v}\rangle \neq 0$. We have shown that the (weakly) attribute-hiding security of our UNIPE follows from the security of the UIPFE and leave a detailed discussion about the construction and security Theorem 4.1 in Section 4. Further, we instantiate our generic UNIPE using the UIPFE of [16] (Section 4.2) based on the SXDH assumption. We emphasize that our NIPE is the *first* construction in an unbounded context that achieves attribute-hiding security. We compare the efficiency of our SXDH-based UNIPE with existing group-based bounded NIPE schemes and observe that our UNIPE ciphertext requires $14n$ group elements, whereas the secret key contains $7n'$ group elements.

• Our UAnon-IBRV: We then integrate our generic UNIPE to build the *first* generic construction of UAnon-IBRV. The anonymity feature follows from attribute-hiding property of the underlying UNIPE scheme.

1.4. Related works

This section comprehensively explains the related works on IPE, IPFE, and broadcast encryption (BE) based on revocation mechanisms.

• IPE: Katz et al. [10] proposed the first attribute-hiding IPE scheme against selective adversaries based on the *general subgroup decisional* (GSD) assumption. In a selective model, the adversary sends the challenge messages to the challenger without seeing the public parameters. Lewko et al. [1] introduced the adaptive WAH security in ZIPE. An adaptive model guarantees stronger security over the selective model since it allows the adversary to query for secret keys and challenge ciphertexts at any arbitrary order during the security experiment. The security of Lewko et al. [1] is based on the non-standard *extended decisional Diffie-Hellman* (eDDH) assumption. Park et al. [18] proposed IPE schemes which are proven secure for the first time based on the standard assumptions known as *decisional bilinear Diffie-Hellman* (DBDH) and DLIN. Next, Okamoto et al. [19] improved WAH security by proposing a stronger adaptive FAH security model. [19] constructed an IPE scheme with the stronger security under the DLIN assumption. Subsequently, Zhenlin et al. [20] extended the notion of IPE with additional multiplicative homomorphic property that is useful in multi-party cloud computing. However, most of the prior works in IPEs [10, 1, 2, 15, 11, 21] can only deal with a

bounded length of vectors. Okamoto et al. [8] proposed the first construction of ZIPE that can handle unbounded length vectors and achieved adaptive FAH security in the standard model relying on the DLIN assumption. After that, Dutta et al. [13] significantly reduced the communication and computation costs compared to the only existing unbounded ZIPE [8] with the same security requirements. In the context of NIPE, Attrapadung et al. [22] presented the first concrete co-selectively secure NIPE construction based on a non-standard *q-type* assumption. In co-selective security, the adversary submits all secret key queries before the setup. However, the challenge attribute might be provided by leveraging the information obtained through secret-key queries. The security was improved by Chen et al. [23], and they developed two selectively secure NIPEs: one is secured under the *n-decisional bilinear Diffie-Hellman exponent* (n-DBDHE) assumption with simultaneously constant-size ciphertext and constant-size secret-key, whereas the second construction contains constant-size secret keys under non-interactive and falsifiable assumptions. Further, Okamoto et al. [15] presented adaptive payload-hiding secure NIPEs, employing the technique of Lewko et al. [1], which either enjoys a constant-size ciphertext or a constant-size secret key. Next, Patranabis et al. [24] suggested an adaptively attribute and function private NIPE construction from *matrix* DDH (MDDH) assumption. A long sequence of efficient NIPE schemes has been proposed [22, 12, 2, 15, 25, 23]. However, constructing a pairing-free NIPE scheme under the standard assumption was open until Katsumata et al. [26] devised a direct approach to constructing a selectively secure NIPE relying on the *learning with error* (LWE) assumption. They provided a generic construction of NIPE from IPFE and instantiated their scheme with the IPFEs of Agrawal et al. [27] to achieve adaptively secure NIPEs from the standard assumptions.

• IPFE: Abdalla et al. [28] introduced the inner product notion into FE and proposed the first IPFE scheme in the selective indistinguishability-based model. Further, Agrawal et al. [27] improved the security and proposed the first adaptively secure IPFE schemes based on the standard assumptions. A long sequence of IPFEs [29, 30, 31] has been proposed in bounded scenarios. Later on, Dufour-Sans et al. [17] and Takashima et al. [16] concurrently introduced the unbounded notion in the context of IPFE. In particular, the unbounded IPFE of [17] achieved selective security in the random oracle model, whereas [16] realized adaptive security in the standard model.

• BE WITH REVOCATION: To enable secure communication to a dynamically changing group of recipients without revealing the individual keys to each re-

cipient, *broadcast encryption* (BE) plays an important role, which was first studied by Fiat and Naor [32]. A BE can be categorized as (I) Subscription-based BE [33, 34] based on the number of subscribed and revoked users. (II) BE-based on revocation [34, 35, 22, 36]. From secure content delivery of pay-TV to content distribution networks (CDN), BE offers a broad range of practical applications. There are other variants of BE explored, such as *identity-based* BE (IBBE) [37, 38, 39, 40], *anonymous* BE (Anon-BE) [41], *hierarchical* BE (HBE) [42]. *Identity-based revocation* (IBRV) is another modification to BE that enables the revocation of private keys. Boldyreva et al. [43] introduced the first efficient IBRV system using a binary tree. Lewko et al. [35] proposed an IBRV with a constant-size of group elements in public and secret keys and linear ciphertext size with the revoked user set. Further, Attrapadung et al. [22] suggested an IBRV scheme from NIPE using a group-based assumption. Recently, Pal et al. [36] showed that an Anon-IBRV could be realized from attribute-hiding NIPE in a semi-generic manner.

For the cloud storage system, revoking unauthenticated users is a primary issue, which was addressed by Li et al. [44]. They provided a CP-ABE system based on the *divisible computation Diffie-Hellman* (DCDH) assumption, where a group manager updates the authenticated users' secret key after the revocation of malicious users. Later on, Zhang et al. [45] proposed a CP-ABE system that partially resolved the key-escrow problem. In this system, a secure key-issuing protocol was employed to calculate the secret key between a trusted authority and the data user. This protocol inhibited key authorities from obtaining the whole user's secret key. In contemporary IoT devices, safeguarding privacy has become a significant challenge, but this can be alleviated by encrypting the data stored in the cloud. Li et al. [46] suggested the first CP-ABE, which allows for fine-grained access control of encrypted IoT data in the cloud. Their approach was to hide the policy associated with the ciphertext, preserving user privacy. Additionally, a white traceable CP-ABE with accountability was developed in order to counter major misuse of the user and authorization centers. Next, Chen et al. [47] designed a CP-ABE based on the DBDH assumption, which supports shared decryption. More explicitly, a group of semi-authorized users can work together to recover the messages in the decryption, while authorized users independently recover the message. Recently, Chen et al. [48] presented a revocable ABE scheme that ensures data integrity after the cloud server performs a revocation. Using this protocol, the data user can detect inappropriate revocations by the cloud.

2. Preliminaries

In this section, we discuss some notations and preliminary definitions.

Notations: Consider a prime p and $\mathbb{Z}_p$ denotes the field $\mathbb{Z}/p\mathbb{Z}$. We use $\lambda \in \mathbb{N}$ as a security parameter and $\mathsf{poly}(\lambda)$ represents a polynomial function. If an element 'a' is uniformly chosen from a set A, we use $a \xleftarrow{\$} A$. Given $n \in \mathbb{N}$, $[n]$ indicates the set $\{1, 2, \ldots, n\}$. We use the bold upper and lower case alphabet to express a matrix $\mathbf{X}$ and a vector $\boldsymbol{u}$, respectively. Also, u_i represents the i-th component of a vector $\boldsymbol{u}$ and $\boldsymbol{u}_i$ indicates the i-th row vector of the matrix $\mathbf{X}$. Let $\mathbf{I}_n$ denote the $n \times n$ identity matrix. Consider g_ι to be a generator of a cyclic group G_ι. For $\mathbf{X} = (x_{ij}) \in \mathsf{GL}_n(\mathbb{Z}_p)$ (collection of all $n \times n$ invertible matrix with the underlying field $\mathbb{Z}_p$), we represent $[\![\mathbf{X}]\!]_k$ as $g_k^{\mathbf{X}}$. For $\boldsymbol{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{F}^n$, we represent $[\![\boldsymbol{u}]\!]_k$ as $[\![\boldsymbol{u}]\!]_k = (g_k^{u_1}, g_k^{u_2}, \ldots, g_k^{u_n})$. For any matrix $\mathbf{X}$, the dual orthonormal and transpose matrices are denoted by $\mathbf{X}^* = (\mathbf{X}^{-1})^\top$ and $\mathbf{X}^\top$ respectively. Consider two vectors $\boldsymbol{u} = (u_i)_{i\in[n]} \in \mathbb{Z}^n$ and $\boldsymbol{v} = (v_i)_{i\in I} \in \mathbb{Z}^{|I|}$, then the inner product is defined as $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i\in I} u_i v_i$ if $I \subseteq [n]$. A *negligible function* is defined as $\mathsf{negl}(\lambda) = \lambda^{-\omega(1)}$ for $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$. The algorithm P is referred to as the *probabilistic polynomial time* (PPT) algorithm if it can be represented as a probabilistic Turing machine with $\mathsf{poly}(\lambda)$ as its running time.

2.1. Basic notions

Definition 1 (Bilinear Groups [8])**.** A bilinear group $\mathbb{BG} = (p, \{G_i\}_{i\in\{1,2,T\}}, g_1, g_2, e)$ consists of the multiplicative groups G_1, G_2 and G_T with prime order p, and g_1, g_2 are the generators of G_1, G_2, respectively. Consider a bilinear map $e : G_1 \times G_2 \to G_T$ satisfying the following properties.
– Bilinearity: $e(x_1^{a_1}, x_2^{a_2}) = e(x_1, x_2)^{a_1 a_2}$ $\forall$ $x_1 \in G_1, x_2 \in G_2$ and $a_1, a_2 \in \mathbb{Z}_p$.
– Non-degeneracy: $e(g_1, g_2) = g_T$ (say) generates G_T.

A bilinear group generator $\mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ generates a prime-order bilinear group $\mathbb{BG}$ using the security parameter λ.

Definition 2 (SXDH Assumption [16])**.** Consider a bilinear group $\mathbb{BG} = (p, \{G_i\}_{i\in\{1,2,T\}}, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ and define the distribution $(D, [\![\mathfrak{t}_\beta]\!]_\iota)$ for $\iota \in \{1, 2\}$ as follows:

$$D = (\mathbb{BG}, [\![\mathfrak{a}]\!]_\iota = g_\iota^{\mathfrak{a}}, [\![\mathfrak{e}]\!]_\iota = g_\iota^{\mathfrak{e}}), [\![\mathfrak{t}_\beta]\!]_\iota = [\![\mathfrak{a}\mathfrak{e} + \beta\mathfrak{f}]\!]_\iota = g_\iota^{\mathfrak{a}\mathfrak{e}+\beta\mathfrak{f}}$$

where $\mathfrak{a}, \mathfrak{e}, \mathfrak{f} \xleftarrow{\$} \mathbb{Z}_p$ with $\beta \in \{0, 1\}$. We say that the SXDH assumption holds over $\mathbb{BG}$ if, for any PPT adversary $\mathcal{A}$ and $\iota \in \{1, 2\}$, the following advantage

$$\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{A}}(\lambda) = \Big| \Pr[\mathcal{A}(D, [\![\mathfrak{t}_0]\!]_\iota) \to 1] - \Pr[\mathcal{A}(D, [\![\mathfrak{t}_1]\!]_\iota) \to 1] \Big|$$

is *negligible* in λ, i.e., $\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda)$.

Definition 3 (Dual pairing vector space (DPVS) [16])**.** We generate a DPVS as $(p, V, V^*, A_1, A_2, G_T, E) \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(n, \mathbb{BG})$ where $n \in \mathbb{N}$ and $\mathbb{BG} = (p, \{G_i\}_{i\in\{1,2,T\}}, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ is a bilinear group of order p (prime) and V, V^*, A_1, A_2, E are defined as follows. We choose a matrix $\mathbf{W} \xleftarrow{\$} \mathsf{GL}_n(\mathbb{Z}_p)$ with random dual orthonormal basis $(\mathbf{W}, \mathbf{W}^*) \leftarrow \mathcal{G}_{\mathsf{OB.Gen}}(\mathbb{Z}_p^n)$. Then $[\![\mathbf{W}]\!]_1$ and $[\![\mathbf{W}^*]\!]_2$ are two dual orthonormal bases corresponding to the vector spaces $V = G_1^n$ and $V^* = G_2^n$ respectively. Let A_1 and A_2 be canonical bases of V, V^* given by $A_\iota = (g_\iota^{\boldsymbol{e}_1}, g_\iota^{\boldsymbol{e}_2} \ldots, g_\iota^{\boldsymbol{e}_n})$ for $\iota \in \{1, 2\}$ where $\boldsymbol{e}_i = (\overbrace{0, ..., 0}^{i-1}, 1, \overbrace{0, ..., 0}^{n-i})$. We now extend the bilinear pairing $e : G_1 \times G_2 \to G_T$ to a mapping $E : V \times V^* \to G_T$ satisfying $E([\![\boldsymbol{u}\mathbf{W}]\!]_1, [\![\boldsymbol{v}\mathbf{W}^*]\!]_2) = e(g_1, g_2)^{\langle \boldsymbol{u}, \boldsymbol{v}\rangle} = g_T^{\langle \boldsymbol{u}, \boldsymbol{v}\rangle}$ for $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{Z}_p^n$. For each arbitrary vectors from $\{\boldsymbol{u}_i\}_{i=1}^k, \{\boldsymbol{v}_j\}_{j=1}^\ell$ and for any $n \times n$ matrix $\mathbf{M} \in \mathsf{GL}_n(\mathbb{Z}_p)$, the following two distributions are identical,

$$\big(\{\boldsymbol{u}_i\mathbf{W}\}_{i\in[k]}, \{\boldsymbol{v}_j\mathbf{W}^*\}_{j\in[\ell]}\big) \equiv \big(\{\boldsymbol{u}_i\mathbf{M}\mathbf{W}\}_{i\in[k]}, \{\boldsymbol{v}_j\mathbf{M}^*\mathbf{W}^*\}_{j\in[\ell]}\big)$$

where $\mathbf{M}^* = (\mathbf{M}^{-1})^\top$. More formally, if a set $U \subseteq [n]$ such that $\forall i \in U, \mathbf{M}\boldsymbol{h}_i = \boldsymbol{w}_i$, then the distributions $\mathcal{D}'_1 = (\{\boldsymbol{w}_i\}_{i\in U}, \{\boldsymbol{u}_i\boldsymbol{w}_i\}_{i\in[k]}, \{\boldsymbol{v}_i\boldsymbol{w}_i^*\}_{i\in[\ell]})$ and $\mathcal{D}'_2 = (\{\boldsymbol{h}_i\}_{i\in U}, \{\boldsymbol{u}_i\mathbf{M}\boldsymbol{h}_i\}_{i\in[k]}, \{\boldsymbol{v}_i\mathbf{M}^*\boldsymbol{h}_i^*\}_{i\in[\ell]})$ are identical. Therefore, $(\mathbf{H}, \mathbf{H}^*) = (\mathbf{M}^{-1}\mathbf{W}, \mathbf{M}^\top\mathbf{W}^*)$ forms random dual orthonormal basis satisfying the following conditions:

$$(\{\boldsymbol{w}_i\}_{i\in U}, \{\boldsymbol{u}_i\mathbf{W}\}_{i\in[k]}, \{\boldsymbol{v}_i\mathbf{W}^*\}_{i\in[\ell]}) \equiv (\{\boldsymbol{h}_i\}_{i\in U}, \{\boldsymbol{u}_i\mathbf{M}\mathbf{D}\}_{i\in[k]}, \{\boldsymbol{v}_i\mathbf{M}^*\mathbf{H}^*\}_{i\in[\ell]})$$

Fig.1 describes the dual orthonormal basis generator $\mathcal{G}_{\mathsf{OB.Gen}}(\mathbb{Z}_p^n)$ over $\mathbb{Z}_p^n$:

$\mathcal{G}_{\mathsf{OB.Gen}}(\mathbb{Z}_p^n)$: This algorithm executes the following steps:

1. Chooses $\mathbf{W} \xleftarrow{\$} \mathsf{GL}_n(\mathbb{Z}_p)$ and computes $\mathbf{W}^* = (\mathbf{W}^{-1})^\top$. Let the i-th row vector of $\mathbf{W}$ be $\boldsymbol{w}_i$, and that of $\mathbf{W}^*$ be $\boldsymbol{w}_i^*$.
2. Sets $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n)$ and $\mathbf{W}^* = (\boldsymbol{w}_1^*, \boldsymbol{w}_2^*, \ldots, \boldsymbol{w}_n^*)$. Note that $(\mathbf{W}, \mathbf{W}^*)$ is dual orthonormal basis as
$$\langle \boldsymbol{w}_i, \boldsymbol{w}_j^* \rangle = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{elsewhere} \end{cases}$$
for all $j, i \in \{1, 2, \ldots, n\}$.
3. Returns $(\mathbf{W}, \mathbf{W}^*)$.

Figure 1: Dual orthonormal basis generator $\mathcal{G}_{\mathsf{OB.Gen}}(\mathbb{Z}_p^n)$

Definition 4 (Unbounded non-zero inner product encryption (UNIPE))**.** The inner product relation of UNIPE is defined over the parameter of the mes-

sage $m \in \mathcal{M}_\lambda$, attribute vector $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathcal{U}_\lambda$, and the predicate vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}} \in \mathcal{V}_\lambda$ where the non-empty index sets $I_{\boldsymbol{u}}, I_{\boldsymbol{v}} \subset \mathbb{N}$. Here $\mathcal{M}_\lambda, \mathcal{U}_\lambda$ and $\mathcal{V}_\lambda$ are the message, attribute, and predicate spaces, respectively. A $\mathsf{UNIPE} = (\mathsf{Setup}, \mathsf{KeyExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ scheme consists of the following four algorithms:

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathbf{Setup}(1^\lambda)$: A trusted authority produces a master public and secret key pair $(\mathsf{mpk}, \mathsf{msk})$ after receiving the security parameter λ as input. Then this authority keeps msk private while mpk is made public.

$\mathsf{sk}_{\boldsymbol{v}} \leftarrow \mathbf{KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v})$: On input $\mathsf{mpk}, \mathsf{msk}$, a predicate vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}} \in \mathcal{V}_\lambda$ such that $\phi \neq I_{\boldsymbol{v}} \subseteq [s_1]$ (where $s_1 = s_1(\lambda)$ is an integer), the trusted authority generates a secret key $\mathsf{sk}_{\boldsymbol{v}}$.

$\mathsf{ct}_{\boldsymbol{u}} \leftarrow \mathbf{Encrypt}(\mathsf{mpk}, \boldsymbol{u}, m)$: On input mpk, a message m and an attribute vector $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathcal{U}_\lambda$ such that $\phi \neq I_{\boldsymbol{u}} \subseteq [s_2]$ (where $s_2 = s_2(\lambda)$ is an integer), the encryptor produces a ciphertext $\mathsf{ct}_{\boldsymbol{u}}$.

$(\alpha \text{ or } \perp) \leftarrow \mathbf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{v}}, \mathsf{ct}_{\boldsymbol{u}})$: The decryptor takes mpk, $\mathsf{sk}_{\boldsymbol{v}}$ and $\mathsf{ct}_{\boldsymbol{u}}$ as input and outputs either $\alpha \in \mathcal{M}_\lambda$ or a special rejection symbol $\perp$.

Correctness: A UNIPE scheme is *correct* if for all attribute vector $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathcal{U}_\lambda$, predicate vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}} \in \mathcal{V}_\lambda$, message $m \in \mathcal{M}_\lambda$ satisfying $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \neq 0$ and $I_{\boldsymbol{v}} \subseteq I_{\boldsymbol{u}}$, the following holds

$$\Pr[\mathsf{UNIPE.Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{v}}, \mathsf{ct}_{\boldsymbol{u}}) = m] = 1$$

where $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{UNIPE.Setup}(1^\lambda)$, $\mathsf{sk}_{\boldsymbol{v}} \leftarrow \mathsf{UNIPE.KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{u})$ and $\mathsf{ct}_{\boldsymbol{u}} \leftarrow \mathsf{UNIPE.Encrypt}(\mathsf{mpk}, \boldsymbol{u}, m)$.

Definition 5. (Adaptively payload-hiding (AdpPH) security [15]) The model of AdpPH indistinguishability security for $\mathsf{UNIPE} = (\mathsf{Setup}, \mathsf{KeyExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ is described below in Fig. 2 as a game played between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$ for $b \in \{0, 1\}$.

$\mathsf{Exp}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpPH}}(1^\lambda)$

1: $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{UNIPE.Setup}(1^\lambda)$.
2: $\{(m^{(0)}, m^{(1)}), \boldsymbol{u}\} \leftarrow \mathcal{A}^{\mathsf{UNIPE.KeyExtract}(\mathsf{mpk},\ \mathsf{msk}, \cdot)}(\mathsf{mpk})$ where $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}}$.
3: $b \xleftarrow{\$} \{0, 1\}$.
4: $\mathsf{ct}^{(b)}_{\boldsymbol{u}} \leftarrow \mathsf{UNIPE.Encrypt}(\mathsf{mpk}, \boldsymbol{u}, m^{(b)})$.
5: $b' \leftarrow \mathcal{A}^{\mathsf{UNIPE.KeyExtract}(\mathsf{mpk},\ \mathsf{msk}, \cdot)}(\mathsf{mpk}, \mathsf{ct}^{(b)}_{\boldsymbol{u}})$.

Figure 2: Experiment $\mathsf{Exp}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpPH}}(1^\lambda)$

For j-th queried predicate vectors $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$, oracle $\mathsf{UNIPE.KeyExtract}$

returns the secret key $\mathsf{sk}_{\boldsymbol{v}^{(j)}} \leftarrow \mathsf{UNIPE.KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v}^{(j)})$. For all secret keys $\mathsf{sk}_{\boldsymbol{v}^{(j)}}$ corresponding to the queried predicate vector $\boldsymbol{v}^{(j)}$, the challenge attribute vector $\boldsymbol{u}$ must satisfy $\langle \boldsymbol{u}, \boldsymbol{v}^{(j)} \rangle = 0$ whenever $I_{\boldsymbol{v}^{(j)}} \subseteq I_{\boldsymbol{u}}$.

For all PPT adversaries $\mathcal{A}$, the UNIPE is AdpPH secure if a negligible function $\mathsf{negl}(\cdot)$ exists such that

$$\mathsf{Adv}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpPH}}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)$$

Definition 6 (Adaptively weakly attribute-hiding (AdpWAH) security [8])**.** The model of AdpWAH indistinguishability security for $\mathsf{UNIPE} = (\mathsf{Setup}, \mathsf{KeyExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ is described below in Fig. 3 as a game played between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$ for $b \in \{0, 1\}$.

$\underline{\mathsf{Exp}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpWAH}}(1^\lambda)}$

1: $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{UNIPE.Setup}(1^\lambda)$.
2: $\{(m^{(0)}, \boldsymbol{u}^{(0)}), (m^{(1)}, \boldsymbol{u}^{(1)})\} \leftarrow \mathcal{A}^{\mathsf{UNIPE.KeyExtract}(\mathsf{mpk},\ \mathsf{msk}, \cdot)}(\mathsf{mpk})$ where $\boldsymbol{u}^{(0)} = (u_i^{(0)})_{i \in I_{\boldsymbol{u}^{(0)}}}$ and $\boldsymbol{u}^{(1)} = (u_i^{(1)})_{i \in I_{\boldsymbol{u}^{(1)}}}$ such that $I_{\boldsymbol{u}^{(0)}} = I_{\boldsymbol{u}^{(1)}} = I_{\boldsymbol{u}}$.
3: $b \xleftarrow{\$} \{0, 1\}$.
4: $\mathsf{ct}_{\boldsymbol{u}^{(b)}} \leftarrow \mathsf{UNIPE.Encrypt}(\mathsf{mpk}, \boldsymbol{u}^{(b)}, m^{(b)})$.
5: $b' \leftarrow \mathcal{A}^{\mathsf{UNIPE.KeyExtract}(\mathsf{mpk},\ \mathsf{msk}, \cdot)}(\mathsf{mpk}, \mathsf{ct}_{\boldsymbol{u}^{(b)}})$.

Figure 3: Experiment $\mathsf{Exp}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpWAH}}(1^\lambda)$

For j-th queried predicate vectors $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$, oracle UNIPE.KeyExtract returns the secret key $\mathsf{sk}_{\boldsymbol{v}^{(j)}} \leftarrow \mathsf{UNIPE.KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v}^{(j)})$. For all the secret key queried vectors $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$ and the challenge attribute-message pairs $\{(m^{(0)}, \boldsymbol{u}^{(0)}), (m^{(1)}, \boldsymbol{u}^{(1)})\}$ must satisfy the following two restrictions:

- If $m^{(0)} \neq m^{(1)}$ then $\langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = 0 = \langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle$.
- If $m^{(0)} = m^{(1)}$ then $\langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle \neq 0$ and $\langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle \neq 0$ with the condition $\langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = \langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle$ (i.e., $\mathsf{sk}_{\boldsymbol{v}^{(j)}}$ is capable to decrypt).

For all PPT adversaries $\mathcal{A}$, the UNIPE is AdpWAH secure if a negligible function $\mathsf{negl}(\cdot)$ exists such that

$$\mathsf{Adv}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpWAH}}(\lambda) = \left| \Pr[b' = b] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda).$$

Remark 1. This weakly attribute-hiding security model for UNIPE is stronger than the security model for NIPE of Okamoto et al. [15] because they do not allow the secret key queries on the predicate vectors $\boldsymbol{v}^{(j)}$ for which successful

decryption is possible (i.e., $\langle \boldsymbol{u}^{(b)}, \boldsymbol{v}^{(j)} \neq 0 \rangle$). In our case, the adversary can query for such secret keys whenever $m^{(0)} = m^{(1)}$.

Definition 7 (Unbounded inner product functional encryption (UIPFE))**.** The UIPFE is defined over the parameters of the message vector $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathcal{U}'_\lambda$ and the key vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}} \in \mathcal{V}'_\lambda$ where the non-empty index sets $I_{\boldsymbol{u}}, I_{\boldsymbol{v}} \subset \mathbb{N}$ and the inner product space $\mathcal{I}_\lambda$. Here, $\mathcal{U}'_\lambda$ and $\mathcal{V}'_\lambda$ are the message and key vector spaces, respectively. A UIPFE = (Setup, KeyExtract, Encrypt, Decrypt) scheme consists of the following four PPT algorithms:

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \textbf{Setup}(1^\lambda)$: A trusted authority produces a master public and secret key pair $(\mathsf{mpk}, \mathsf{msk})$ after receiving the security parameter λ as input. Then this authority keeps msk private while mpk is made public.

$\mathsf{sk}_{\boldsymbol{v}} \leftarrow \textbf{KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v})$: On input $\mathsf{mpk}, \mathsf{msk}$ and a key vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}} \in \mathcal{V}'_\lambda$ such that $\phi \neq I_{\boldsymbol{v}} \subseteq [s_1]$ (where $s_1 = s_1(\lambda)$ is an integer), the trusted authority outputs a secret key $\mathsf{sk}_{\boldsymbol{v}}$.

$\mathsf{ct}_{\boldsymbol{u}} \leftarrow \textbf{Encrypt}(\mathsf{mpk}, \boldsymbol{u})$: On input mpk, a message vector $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathcal{U}'_\lambda$ such that $\phi \neq I_{\boldsymbol{u}} \subseteq [s_2]$ (where $s_2 = s_2(\lambda)$ is an integer), the encryptor outputs a ciphertext $\mathsf{ct}_{\boldsymbol{u}}$.

$(\alpha \text{ or } \perp) \leftarrow \textbf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{v}}, \mathsf{ct}_{\boldsymbol{u}})$: The decryptor takes mpk, $\mathsf{sk}_{\boldsymbol{v}}, \mathsf{ct}_{\boldsymbol{u}}$ as input and outputs either $\alpha \in \mathcal{I}_\lambda$ or a special rejection symbol $\perp$.

Correctness: A UIPFE scheme is *correct* if for all message vector $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathcal{U}'_\lambda$, key vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}} \in \mathcal{V}'_\lambda$ satisfying $I_{\boldsymbol{v}} \subseteq I_{\boldsymbol{u}}$, the following holds

$$\Pr[\mathsf{UIPFE.Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{v}}, \mathsf{ct}_{\boldsymbol{u}}) = \langle \boldsymbol{u}, \boldsymbol{v} \rangle] = 1$$

where $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{UIPFE.Setup}(1^\lambda), \mathsf{sk}_{\boldsymbol{v}} \leftarrow \mathsf{UIPFE.KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v})$ and $\mathsf{ct}_{\boldsymbol{u}} \leftarrow \mathsf{UIPFE.Encrypt}(\mathsf{mpk}, \boldsymbol{u})$.

Definition 8 (Adaptive indistinguishability (AdpIND) security [16])**.** The model of AdpIND security for UIPFE = (Setup, KeyExtract, Encrypt, Decrypt) is described below in Fig. 4 as a game played between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$ for $b \in \{0.1\}$.

$\underline{\mathsf{Exp}^{\mathsf{UIPFE}}_{\mathcal{A},\mathsf{AdpIND}}(1^\lambda)}$

1: $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{UIPFE.Setup}(1^\lambda)$.

2: $\{(\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)})\} \leftarrow \mathcal{A}^{\mathsf{UIPFE.KeyExtract}(\mathsf{mpk},\ \mathsf{msk},\cdot)}(\mathsf{mpk})$ where $\boldsymbol{u}^{(0)} = (u_i^{(0)})_{i\in I_{\boldsymbol{u}^{(0)}}}$, $\boldsymbol{u}^{(1)} = (u_i^{(1)})_{i\in I_{\boldsymbol{u}^{(1)}}}$ with $I_{\boldsymbol{u}^{(0)}} = I_{\boldsymbol{u}^{(1)}} = I_{\boldsymbol{u}}$.

3: $b \xleftarrow{\$} \{0,1\}$.

4: $\mathsf{ct}_{\boldsymbol{u}^{(b)}} \leftarrow \mathsf{UIPFE.Encrypt}(\mathsf{mpk}, \boldsymbol{u}^{(b)})$.

5: $b' \leftarrow \mathcal{A}^{\mathsf{UIPFE.KeyExtract}(\mathsf{mpk},\ \mathsf{msk},\cdot)}(\mathsf{mpk}, \mathsf{ct}_{\boldsymbol{u}^{(b)}})$.

Figure 4: Experiment $\mathsf{Exp}^{\mathsf{UIPFE}}_{\mathcal{A},\mathsf{AdpIND}}(1^\lambda)$

For j-th queried key vectors $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i\in I_{\boldsymbol{v}^{(j)}}}$, the oracle $\mathsf{UIPFE.KeyExtract}$ returns the secret key $\mathsf{sk}_{\boldsymbol{v}^{(j)}} \leftarrow \mathsf{UIPFE.KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v}^{(j)})$. For all secret keys $\mathsf{sk}_{\boldsymbol{v}^{(j)}}$ corresponding to the queried key vector $\boldsymbol{v}^{(j)}$, the challenge message vectors $\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}$ must satisfy $\langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = \langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle$ whenever $I_{\boldsymbol{v}^{(j)}} \subseteq I_{\boldsymbol{u}}$.

For all PPT adversaries $\mathcal{A}$, the UIPFE is AdpIND secure if a negligible function $\mathsf{negl}(\cdot)$ exists such that

$$\mathsf{Adv}^{\mathsf{UIPFE}}_{\mathcal{A},\mathsf{AdpIND}}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda).$$

Definition 9 (Unbounded anonymous identity-based revocation ($\mathsf{UAnon\text{-}IBRV}$))**.** A $\mathsf{UAnon\text{-}IBRV}$ scheme is defined over the parameter of identity $\mathsf{id} \in \mathcal{ID}$ and a message $m \in \mathcal{M}_\lambda$. Here, $\mathcal{ID}$ and $\mathcal{M}_\lambda$ are the identity and message spaces, respectively. A $\mathsf{UAnon\text{-}IBRV} = (\mathsf{Setup}, \mathsf{KeyExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ consists of four PPT algorithms as follows:

$\underline{(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathbf{Setup}(1^\lambda)}$: A trusted authority produces a master public and secret key pair $(\mathsf{mpk}, \mathsf{msk})$ after receiving the security parameter λ as input. Then this authority keeps msk private while mpk is made public.

$\underline{\mathsf{sk}_{\mathsf{id}} \leftarrow \mathbf{KeyExtract}(\mathsf{mpk}, \mathsf{msk}, s, \mathsf{id})}$: On input mpk, msk, the size of revoke user set s and an identity $\mathsf{id} \in \mathcal{ID}$, the trusted authority generates a secret key $\mathsf{sk}_{\mathsf{id}}$.

$\underline{\mathsf{ct}_m \leftarrow \mathbf{Encrypt}(\mathsf{mpk}, \mathcal{R}, m)}$: On input mpk, a message $m \in \mathcal{M}_\lambda$, a revoke user set $\mathcal{R} \subset \mathcal{ID}$ (where $|\mathcal{R}| = s = s(\lambda)$ is an integer), the encryptor outputs a ciphertext ct_m.

$\underline{(\alpha \text{ or } \perp) \leftarrow \mathbf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_m)}$: The decryptor takes $\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}$ and ct_m as input and outputs $\alpha \in \mathcal{M}_\lambda$ or a special rejection symbol $\perp$.

Correctness: A $\mathsf{UAnon\text{-}IBRV}$ scheme is *correct* if for all identity $\mathsf{id} \in \mathcal{ID}$, a revoke user set $\mathcal{R}$ and a message $m \in \mathcal{M}_\lambda$ satisfying $\mathsf{id} \notin \mathcal{R}$, the following

holds

$$\Pr[\mathsf{UAnon\text{-}IBRV.Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_m) = m] = 1$$

where $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{UAnon\text{-}IBRV.Setup}(1^\lambda)$, $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{UAnon\text{-}IBRV.KeyExtract}(\mathsf{mpk}, \mathsf{msk}, s, \mathsf{id})$, $\mathsf{ct}_m \leftarrow \mathsf{UAnon\text{-}IBRV.Encrypt}(\mathsf{mpk}, \mathcal{R}, m)$.

Definition 10 (Adaptive anonymous (AdpAnon) security)**.** The model of AdpAnon security for UAnon-IBRV = (Setup, KeyExtract, Encrypt, Decrypt) is described in Fig. 5 below as a game played between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$ for $b \in \{0, 1\}$,

$\underline{\mathsf{Expt}^{\mathsf{UAnon\text{-}IBRV}}_{\mathcal{A},\mathsf{AdpAnon}}(1^\lambda)}$

1: $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{UAnon\text{-}IBRV.Setup}(1^\lambda)$.
2: $\{(m^{(0)}, \mathcal{R}^{(0)}), (m^{(1)}, \mathcal{R}^{(1)})\} \leftarrow \mathcal{A}^{\mathsf{UAnon\text{-}IBRV.KeyExtract}(\mathsf{mpk},\mathsf{msk},\cdot,\cdot)}(\mathsf{mpk})$ with $|\mathcal{R}^{(0)}| = |\mathcal{R}^{(1)}| = s^*$.
3: $b \xleftarrow{\$} \{0, 1\}$.
4: $\mathsf{ct}_{m^{(b)}} \leftarrow \mathsf{UAnon\text{-}IBRV.Encrypt}(\mathsf{mpk}, \mathcal{R}^{(b)}, m^{(b)})$.
5: $b' \leftarrow \mathcal{A}^{\mathsf{UAnon\text{-}IBRV.KeyExtract}(\mathsf{mpk},\mathsf{msk},\cdot,\cdot)}(\mathsf{mpk}, \mathsf{ct}_{m^{(b)}})$.
6: Outputs: b'.

Figure 5: Experiment $\mathsf{Expt}^{\mathsf{UAnon\text{-}IBRV}}_{\mathcal{A},\mathsf{AdpAnon}}(1^\lambda)$

where $\mathsf{UAnon\text{-}IBRV.KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \cdot, \cdot)$ oracle takes any input $\widehat{s}, \{\mathsf{id}\}$ and returns $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{UAnon\text{-}IBRV.KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \widehat{s}, \mathsf{id})$ with the restriction for all $\mathsf{id} \in \mathcal{R}^{(0)} \cap \mathcal{R}^{(1)}$. For all PPT adversaries $\mathcal{A}$, the UAnon-IBRV is AdpAnon secure if a negligible function $\mathsf{negl}(\cdot)$ exists such that

$$\mathsf{Adv}^{\mathsf{UAnon\text{-}IBRV}}_{\mathcal{A},\mathsf{AdpAnon}}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda).$$

3. Our Payload-hiding UNIPE

We present below our UNIPE = (Setup, KeyExtract, Encrypt, Decrypt) scheme using the framework of DPVS [14].

$\underline{(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathbf{Setup}(1^\lambda)}$: On input the security parameter λ, a trusted authority runs this algorithm using the following steps:

– Generates a bilinear group $\mathbb{BG} = (p, \{G_i\}_{i=1,2,T}, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ as defined in Definition 1.

– Chooses two uniformly random matrices $\mathbf{W} \xleftarrow{\$} \mathsf{GL}_5(\mathbb{Z}_p)$, $\widehat{\mathbf{W}} \xleftarrow{\$} \mathsf{GL}_6(\mathbb{Z}_p)$.

– Generates $(p, V, V^*, A_1, A_2, G_T, E) \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(5, \mathbb{BG})$ and $(p, \widehat{V}, \widehat{V}^*, \widehat{A}_1, \widehat{A}_2, G_T, \widehat{E}) \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(6, \mathbb{BG})$ where $A_1 = (g_1^{\boldsymbol{e}_1}, g_1^{\boldsymbol{e}_2}, \ldots, g_1^{\boldsymbol{e}_5})$, $A_2 = (g_2^{\boldsymbol{e}_1}, g_2^{\boldsymbol{e}_2}, \ldots, g_2^{\boldsymbol{e}_5})$ are the canonical bases for the vector spaces $V = G_1^5, V^* = G_2^5$ respectively with $\boldsymbol{e}_i = (\overbrace{0, \ldots, 0}^{i-1}, 1, \overbrace{0, \ldots, 0}^{5-i})$ and similarly $\widehat{A}_1 = (g_1^{\widehat{\boldsymbol{e}}_1}, g_1^{\widehat{\boldsymbol{e}}_2}, \ldots, g_1^{\widehat{\boldsymbol{e}}_6})$, $\widehat{A}_2$

$= (g_2^{\widehat{e}_1}, g_2^{\widehat{e}_2}, \ldots, g_2^{\widehat{e}_6})$ are the canonical bases for the vector spaces $\widehat{V} = G_1^6, \widehat{V}^* = G_2^6$ respectively with $\widehat{\boldsymbol{e}}_i = (\overbrace{0, \ldots, 0}^{i-1}, 1, \overbrace{0, \ldots, 0}^{6-i})$. Consider $E : V \times V^* \to G_T$ and $\widehat{E} : \widehat{V} \times \widehat{V}^* \to G_T$ are extended maps of the bilinear map e defined as $E([\![\boldsymbol{x}\mathbf{W}]\!]_1, [\![\boldsymbol{y}\mathbf{W}^*]\!]_2) = e(g_1, g_2)^{\langle \boldsymbol{x}, \boldsymbol{y} \rangle} = g_T^{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}_p^5$ and $\widehat{E}([\![\boldsymbol{x}'\widehat{\mathbf{W}}]\!]_1, [\![\boldsymbol{y}'\widehat{\mathbf{W}}^*]\!]_2) = e(g_1, g_2)^{\langle \boldsymbol{x}', \boldsymbol{y}' \rangle} = g_T^{\langle \boldsymbol{x}', \boldsymbol{y}' \rangle}$ for $\boldsymbol{x}', \boldsymbol{y}' \in \mathbb{Z}_p^6$.
– Sets $\mathsf{pp} = (p, \{g_i\}_{i=1,2,T}, V, V^*, \widehat{V}, \widehat{V}^*, E, \widehat{E})$, $g_T = e(g_1, g_2)$ generates G_T.
– Outputs the master key pair as follows:

$$\mathsf{mpk} = (\mathsf{pp}, [\![\boldsymbol{w}_1]\!]_1, [\![\boldsymbol{w}_3]\!]_1, [\![\boldsymbol{w}_5]\!]_1, [\![\widehat{\boldsymbol{w}}_1]\!]_1, [\![\widehat{\boldsymbol{w}}_2]\!]_1, [\![\widehat{\boldsymbol{w}}_3]\!]_1, [\![\widehat{\boldsymbol{w}}_6]\!]_1)$$
$$\mathsf{msk} = (\boldsymbol{w}_1^*, \boldsymbol{w}_3^*, \boldsymbol{w}_4^*, \widehat{\boldsymbol{w}}_1^*, \widehat{\boldsymbol{w}}_2^*, \widehat{\boldsymbol{w}}_3^*, \widehat{\boldsymbol{w}}_5^*)$$

where $(\boldsymbol{w}_i, \boldsymbol{w}_i^*), (\widehat{\boldsymbol{w}}_i, \widehat{\boldsymbol{w}}_i^*)$ are the i-th rows of the matrices $(\mathbf{W}, \mathbf{W}^* = (\mathbf{W}^{-1})^\top)$ and $(\widehat{\mathbf{W}}, \widehat{\mathbf{W}}^* = (\widehat{\mathbf{W}}^{-1})^\top)$ respectively.
– The trusted authority keeps msk private while mpk is made public.

$\underline{\mathsf{sk}_{\boldsymbol{v}} \leftarrow \textbf{KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v})}$: On input mpk, msk, a predicate vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}} \in \mathbb{Z}_p^{|I_{\boldsymbol{v}}|}$ such that $\phi \neq I_{\boldsymbol{v}} \subseteq [s_1]$ (where $s_1 = s_1(\lambda)$ is an integer), the trusted authority proceeds as follows:
– Computes

$$\boldsymbol{k}_0 = (\omega, 0, 1, \phi_0, 0)\mathbf{W}^*; \quad \boldsymbol{k}_i = (\Upsilon_i(-i, 1), \omega v_i, 0, \phi_i, 0)\widehat{\mathbf{W}}^* \ \forall i \in I_{\boldsymbol{v}}$$

where $\omega, \phi_0 \xleftarrow{\$} \mathbb{Z}_p$ and $\Upsilon_i, \phi_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in I_{\boldsymbol{v}}$.
– Outputs the secret key $\mathsf{sk}_{\boldsymbol{v}} = (\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}}, [\![\boldsymbol{k}_0]\!]_2, \{[\![\boldsymbol{k}_i]\!]_2\}_{i \in I_{\boldsymbol{v}}})$.

$\underline{\mathsf{ct}_{\boldsymbol{u}} \leftarrow \textbf{Encrypt}(\mathsf{mpk}, \boldsymbol{u}, m)}$: On input mpk, a message $m \in G_T$ and an attribute $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathbb{Z}_p^{|I_{\boldsymbol{u}}|}$ such that $\phi \neq I_{\boldsymbol{u}} \subseteq [s_2]$ (where $s_2 = s_2(\lambda)$ is an integer), the encryptor runs the following steps:
– Computes

$$[\![\boldsymbol{c}_0]\!]_1 = [\![(-\delta, 0, z, 0, \eta_0)\mathbf{W}]\!]_1, \ [\![\boldsymbol{c}_i]\!]_1 = [\![(\Psi_i(1, i), \delta u_i, 0, 0, \eta u_i)\widehat{\mathbf{W}}]\!]_1$$

where $\delta, z, \eta, \eta_0 \xleftarrow{\$} \mathbb{Z}_p$ and $\Psi_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in I_{\boldsymbol{u}}$.
– Outputs the ciphertext $\mathsf{ct}_{\boldsymbol{u}} = (\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}}, [\![\boldsymbol{c}_0]\!]_1, \{[\![\boldsymbol{c}_i]\!]_1\}_{i \in I_{\boldsymbol{u}}}, c = [\![z]\!]_T \cdot m)$.

$\underline{(\alpha \text{ or } \perp) \leftarrow \textbf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{v}}, \mathsf{ct}_{\boldsymbol{u}})}$: The decryptor takes input mpk, $\mathsf{sk}_{\boldsymbol{v}}$, $\mathsf{ct}_{\boldsymbol{u}}$ and performs the following operations:
– For $I_{\boldsymbol{v}} \subseteq I_{\boldsymbol{u}}$ with $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \neq 0$, it computes

$$h = E([\![\boldsymbol{c}_0]\!]_1, [\![\boldsymbol{k}_0]\!]_2) \prod_{i \in I_{\boldsymbol{v}}} \widehat{E}\Big([\![\boldsymbol{c}_i]\!]_1, [\![\boldsymbol{k}_i]\!]_2^{\langle \boldsymbol{u}, \boldsymbol{v} \rangle^{-1}}\Big)$$

otherwise outputs $\perp$.
– Returns $\alpha = c/h$.

Correctness: For $I_{\boldsymbol{v}} \subseteq I_{\boldsymbol{u}}$ with $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \neq 0$, we have

$$h = E([\![\boldsymbol{c}_0]\!]_1, [\![\boldsymbol{k}_0]\!]_2) \prod_{i \in I_{\boldsymbol{v}}} \widehat{E}\Big([\![\boldsymbol{c}_i]\!]_1, [\![\boldsymbol{k}_i]\!]_2^{\langle \boldsymbol{u}, \boldsymbol{v} \rangle^{-1}}\Big)$$

$$= g_T^{\langle (-\delta, 0, z, 0, \eta_0), (\omega, 0, 1, \phi_0, 0) \rangle} \cdot \prod_{i \in I_{\boldsymbol{v}}} g_T^{\langle \boldsymbol{u}, \boldsymbol{v} \rangle^{-1} \Big\langle \big(\Psi_i(1,i), \delta u_i, 0, 0, \eta u_i\big), \big(\Upsilon_i(-i,1), \omega v_i, 0, \phi_i, 0\big) \Big\rangle}$$

$$= g_T^{-\omega\delta + z} \cdot e(g_1, g_2)^{\langle \boldsymbol{u}, \boldsymbol{v} \rangle^{-1} \omega\delta \langle \boldsymbol{u}, \boldsymbol{v} \rangle} = g_T^{-\omega\delta + z + \omega\delta} = g_T^z = (g_T)^z = [\![z]\!]_T$$

and $\frac{c}{h} = \frac{[\![z]\!]_T \cdot m}{[\![z]\!]_T} = m$.

3.1. Security

Theorem 3.1. *Under the SXDH assumption, our* UNIPE *scheme is adaptive payload-hiding (AdpPH) secure as per Definition 5. More specifically, for all $\lambda \in \mathbb{N}$ as security parameters and PPT adversaries $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}$ against the SXDH assumption such that*

$$\mathsf{Adv}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpPH}}(\lambda) \leq (3\nu + 1)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + \frac{1}{p} + 2^{-\Omega(\lambda)}$$

where the adversaries make ν many secret key queries.

Proof sketch. At a very high level, our technique is inspired by Water's [49] dual-system encryption methodology. Consider the challenge attribute $\boldsymbol{u}$ with the challenge message pair $(m^{(0)}, m^{(1)})$. Note that any queried predicate vector $\boldsymbol{v}$ for the secret key must satisfy $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0$ whenever $I_{\boldsymbol{v}} \subseteq I_{\boldsymbol{u}}$. Here, we consider $(3\nu + 2)$ many games to convert an honestly generated challenge ciphertext/normal ciphertext into a *randomized* ciphertext (where the challenge bit is completely hidden). We do this by transforming the *normal* ciphertext into two intermediate forms, the first of which is *semi-functional* and the second of which is *randomized*. Similarly, the secret keys demand two kinds of forms: one is *normal*, and the other is *semi-functional*. Here, we transform the normal ciphertext $\mathsf{ct}^{\mathsf{norml}}$ into semi-functional ciphertext $\mathsf{ct}^{\mathsf{semi}}$, where the advantage gap is bounded by the advantage of the SXDH problem. Next, a loop has been incorporated to convert a normal secret key $\mathsf{sk}^{\mathsf{norml}}$ to a semi-functional secret key $\mathsf{sk}^{\mathsf{semi}}$ via *1st semi-functional* and *2nd semi-functional* forms of secret keys, i.e., $\mathsf{sk}^{\mathsf{1st\ pre\text{-}semi}}$ and $\mathsf{sk}^{\mathsf{2nd\ pre\text{-}semi}}$. Note

that none of these secret keys will decode the challenge ciphertext due to the restrictions of the payload-hiding game. In this loop, we have shown that the joint distributions $(\mathsf{sk}^{\text{1st-pre semi}}, \mathsf{ct}^{\mathsf{semi}})$, $(\mathsf{sk}^{\text{2nd-pre semi}}, \mathsf{ct}^{\mathsf{semi}})$, and $(\mathsf{sk}^{\mathsf{semi}}, \mathsf{ct}^{\mathsf{semi}})$ are computationally indistinguishable using the SXDH assumption. Finally, the semi-functional ciphertext $\mathsf{ct}^{\mathsf{semi}}$ is conceptually changed to a randomized ciphertext $\mathsf{ct}^{\mathsf{rand}}$ via the well-known basis-transforming technique of DPVS. We have depicted the security in more detail as follows.

Proof. Consider a PPT adversary $\mathcal{A}$ against AdpPH security for our UNIPE scheme. We design an algorithm $\mathcal{B}$ employing $\mathcal{A}$ as a subroutine to break the SXDH assumption. We define $(3\nu+2)$ games between the adversary and challenger. It starts with real Game 0 and ultimately reaches Game 3, where $\mathcal{A}$ obtains no information about the challenge bit. Let R_i represent the event that $\mathcal{A}$ wins in Game i. We use a part frame box to indicate coefficients that changed from the previous game.

Game 0: This is the AdpPH security experiment as described in Definition 5 where $\mathcal{B}$ serves as the challenger.

- **Setup**: In the initial phase, $\mathcal{B}$ produces a master keys pair $(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{UNIPE.Setup}(1^\lambda)$ and passes $\mathsf{mpk} = (\mathsf{pp}, [\![\boldsymbol{w}_1]\!]_1, [\![\boldsymbol{w}_3]\!]_1, [\![\boldsymbol{w}_5]\!]_1, [\![\widehat{\boldsymbol{w}}_1]\!]_1, [\![\widehat{\boldsymbol{w}}_2]\!]_1, [\![\widehat{\boldsymbol{w}}_3]\!]_1, [\![\widehat{\boldsymbol{w}}_6]\!]_1)$ to $\mathcal{A}$ and keeps $\mathsf{msk} = (\boldsymbol{w}_1^*, \boldsymbol{w}_3^*, \boldsymbol{w}_4^*, \widehat{\boldsymbol{w}}_1^*, \widehat{\boldsymbol{w}}_2^*, \widehat{\boldsymbol{w}}_3^*, \widehat{\boldsymbol{w}}_5^*)$ secret to itself. Here, $(\boldsymbol{w}_i, \boldsymbol{w}_i^*), (\widehat{\boldsymbol{w}}_i, \widehat{\boldsymbol{w}}_i^*)$ are i-th rows of the matrices $(\mathbf{W}, \mathbf{W}^*)$ and $(\widehat{\mathbf{W}}, \widehat{\mathbf{W}}^*)$ respectively and $\mathsf{pp} = (p, \{g_i\}_{i=1,2,T}, V, V^*, \widehat{V}, \widehat{V}^*, E, \widehat{E})$ is generated from the bilinear group $\mathbb{BG} = (p, \{G_i\}_{i=\{1,2,T\}}, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$, $(p, V, V^*, G_T, A_1, A_2, E) \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(5, \mathbb{BG})$, $(p, \widehat{V}, \widehat{V}^*, \widehat{A}_1, \widehat{A}_2, G_T, \widehat{E}) \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(6, \mathbb{BG})$ with $g_T = e(g_1, g_2)$ as described in Definitions 3 and 1.
- **Pre-key query**: Adversary $\mathcal{A}$ can query polynomially many secret keys for the predicate vectors. Corresponding to the j-th secret key query on the predicate vector $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$ with non-empty index set $I_{\boldsymbol{v}^{(j)}}$, $\mathcal{B}$ issues the j-th secret key $\mathsf{sk}_{\boldsymbol{v}^{(j)}} = (\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}, [\![\boldsymbol{k}_0]\!]_2, ([\![\boldsymbol{k}_i]\!]_2)_{i \in I_{\boldsymbol{v}^{(j)}}}) \leftarrow \mathsf{UNIPE.KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v}^{(j)})$ where

  $$[\![\boldsymbol{k}_0]\!]_2 = [\![(\omega, 0, 1, \phi_0, 0)\mathbf{W}^*]\!]_2;\ [\![\boldsymbol{k}_i]\!]_2 = [\![(\Upsilon_i(-i, 1), \omega v_i^{(j)}, 0, \phi_i, 0)\widehat{\mathbf{W}}^*]\!]_2$$

  with $\omega, \phi_0 \xleftarrow{\$} \mathbb{Z}_p$ and $\Upsilon_i, \phi_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in I_{\boldsymbol{v}^{(j)}}$. These secret keys are known as *normal secret keys*.

– **Challenge query:** The challenge messages pair $(m^{(0)}, m^{(1)})$ along with the challenge attribute vector $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}}$ are submitted by $\mathcal{A}$ to $\mathcal{B}$ satisfying $\langle \boldsymbol{u}, \boldsymbol{v}^{(j)} \rangle = 0$ for all predicate vectors $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$ whenever $I_{\boldsymbol{v}^{(j)}} \subseteq I_{\boldsymbol{u}}$ on which pre-key query has been made by $\mathcal{A}$. The challenger $\mathcal{B}$ chooses $b \xleftarrow{\$} \{0,1\}$ and produces the challenge ciphertext $\mathsf{ct}_{\boldsymbol{u}}^{(b)} = \left(\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}}, [\![\boldsymbol{c}_0]\!]_1, ([\![\boldsymbol{c}_i]\!]_1)_{i \in I_{\boldsymbol{u}}}, c^{(b)}\right) \leftarrow \mathsf{UNIPE.Encrypt}(\mathsf{mpk}, \boldsymbol{u}, m^{(b)})$ and sends $\mathsf{ct}_{\boldsymbol{u}}^{(b)}$ to $\mathcal{A}$ where

$$[\![\boldsymbol{c}_0]\!]_1 = [\![(-\delta, 0, z, 0, \eta_0)\mathbf{W}]\!]_1,\ [\![\boldsymbol{c}_i]\!]_1 = [\![(\Psi_i(1,i), \delta u_i, 0, 0, \eta u_i)\widehat{\mathbf{W}}]\!]_1,$$
$$c^{(b)} = [\![z]\!]_T \cdot m^{(b)}$$

with $\delta, z, \eta_0, \eta \xleftarrow{\$} \mathbb{Z}_p$ and $\Psi_i \xleftarrow{\$} \mathbb{Z}_p\ \forall i \in I_{\boldsymbol{u}}$.

– **Post-key query:** All predicate vectors $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$ on which post-key queries must satisfy $\langle \boldsymbol{u}, \boldsymbol{v}^{(j)} \rangle = 0$ whenever $I_{\boldsymbol{v}^{(j)}} \subseteq I_{\boldsymbol{u}}$.

– **Guess**: Finally, $\mathcal{A}$ guesses a bit b' and forwards it to $\mathcal{B}$. The challenger $\mathcal{B}$ outputs $\beta = 1$ if $b = b'$, otherwise outputs $\beta = 0$.

Game 1: This game is similar to Game 0 except that the challenge ciphertext components $[\![\boldsymbol{c}_0]\!]_1, [\![\boldsymbol{c}_i]\!]_1$ of $\mathsf{ct}_{\boldsymbol{u}}^{(b)}$ corresponding to the message, attribute pair $(m^{(b)}, \boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}})$ are set as

$$[\![\boldsymbol{c}_0]\!]_1 = [\![(-\delta, \boxed{-\tau}, z, 0, \eta_0)\mathbf{W}]\!]_1;\ [\![\boldsymbol{c}_i]\!]_1 = [\![(\Psi_i(1,i), \delta u_i, \boxed{\tau u_i}, 0, \eta u_i)\widehat{\mathbf{W}}]\!]_1,$$
$$c^{(b)} = [\![z]\!]_T \cdot m^{(b)}$$

where $b \xleftarrow{\$} \{0,1\}$ and $\tau \xleftarrow{\$} \mathbb{Z}_p$. The remaining randomness $\Psi_i, z, \eta_0, \delta$ are chosen by $\mathcal{B}$ as in Game 0. This type of ciphertext is called *semi-functional ciphertext.*

Game 2-h-1: For $h \in [\nu]$, this game is similar to Game 2-$(h-1)$-3 except that the h-th secret key components $[\![\boldsymbol{k}_0]\!]_2, [\![\boldsymbol{k}_i]\!]_2$ of $\mathsf{sk}_{\boldsymbol{v}^{(h)}} = (\boldsymbol{v}^{(h)} = (v_i^{(h)})_{i \in I_{\boldsymbol{v}^{(h)}}}, [\![\boldsymbol{k}_0]\!]_2, ([\![\boldsymbol{k}_i]\!]_2)_{i \in I_{\boldsymbol{v}^{(h)}}})$ corresponding to the predicate vector $\boldsymbol{v}^{(h)} = (v_i^{(h)})_{i \in I_{\boldsymbol{v}^{(h)}}}$ are set as

$$[\![\boldsymbol{k}_0]\!]_2 = [\![(\omega, \boxed{\rho}, 1, \phi_0, 0)\mathbf{W}^*]\!]_2;\ [\![\boldsymbol{k}_i]\!]_2 = [\![(\Upsilon_i(-i,1), \omega v_i^{(h)}, \boxed{\rho v_i^{(h)}}, \phi_i, 0)\widehat{\mathbf{W}}^*]\!]_2$$

where $\rho \xleftarrow{\$} \mathbb{Z}_p$. Other randomness $\Upsilon_i, \phi_0, \phi_i, \omega$ are chosen by $\mathcal{B}$ as in Game 2-$(h-1)$-3. These secret keys $\mathsf{sk}_{\boldsymbol{v}^{(h)}}$ are referred *1st pre semi-functional secret keys.* Recall that. Game 2-0-3 is identical to Game 1.

Game 2-h-2: For $h \in [\nu]$, this game is similar to Game 2-h-1 except that the h-th secret key components $[\![\boldsymbol{k}_0]\!]_2, [\![\boldsymbol{k}_i]\!]_2$ of $\mathsf{sk}_{\boldsymbol{v}^{(h)}} = (\boldsymbol{v}^{(h)} = (v_i^{(h)})_{i \in I_{\boldsymbol{v}^{(h)}}}, [\![\boldsymbol{k}_0]\!]_2, ([\![\boldsymbol{k}_i]\!]_2)_{i \in I_{\boldsymbol{v}^{(h)}}})$ for a predicate vector $\boldsymbol{v}^{(h)} = (v_i^{(h)})_{i \in I_{\boldsymbol{v}^{(h)}}}$ are set as

$$[\![\boldsymbol{k}_0]\!]_2 = [\![(\omega, \boxed{\zeta}, 1, \phi_0, 0)\mathbf{W}^*]\!]_2;\ [\![\boldsymbol{k}_i]\!]_2 = [\![(\Upsilon_i(-i, 1), \omega v_i^{(h)}, \rho v_i^{(h)}, \phi_i, 0)\widehat{\mathbf{W}}^*]\!]_2$$

where $\zeta, \rho \xleftarrow{\$} \mathbb{Z}_p$. The rest of the randomness $\Upsilon_i, \phi_0, \phi_i, \omega$ are chosen by $\mathcal{B}$ as in Game 2-h-1, These secret keys $\mathsf{sk}_{\boldsymbol{v}^{(h)}}$ are termed *2nd pre semi-functional secret keys.*

Game 2-h-3: For $h \in [\nu]$, this game is similar to Game 2-h-2 except that the h-th secret key components $[\![\boldsymbol{k}_0]\!]_2, [\![\boldsymbol{k}_i]\!]_2$ of $\mathsf{sk}_{\boldsymbol{v}^{(h)}} = (\boldsymbol{v}^{(h)} = (v_i^{(h)})_{i \in I_{\boldsymbol{v}^{(h)}}}, [\![\boldsymbol{k}_0]\!]_2, ([\![\boldsymbol{k}_i]\!]_2)_{i \in I_{\boldsymbol{v}^{(h)}}})$ for a predicate vector $\boldsymbol{v}^{(h)} = (v_i^{(h)})_{i \in I_{\boldsymbol{v}^{(h)}}}$ are set as

$$[\![\boldsymbol{k}_0]\!]_2 = [\![(\omega, \zeta, 1, \phi_0, 0)\mathbf{W}^*]\!]_2;\ [\![\boldsymbol{k}_i]\!]_2 = [\![(\Upsilon_i(-i, 1), \omega v_i^{(h)}, \boxed{0}, \phi_i, 0)\widehat{\mathbf{W}}^*]\!]_2$$

where $\zeta \xleftarrow{\$} \mathbb{Z}_p$ and the remaining randomness $\Upsilon_i, \phi_0, \phi_i, \omega$ are similar as in Game 2-h-2. These secret keys $\mathsf{sk}_{\boldsymbol{v}^{(h)}}$ are called *semi-functional secret keys.*

Game 3: Game 3 is identical to Game 2-ν-3 except that the challenge ciphertext components $[\![\boldsymbol{c}_0]\!]_1, [\![\boldsymbol{c}_i]\!]_1, c^{(b)}$ of $\mathsf{ct}_{\boldsymbol{u}}^{(b)}$ associated with the challenge pair $(m^{(b)}, \boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}})$ are set as

$$\begin{aligned}
[\![\boldsymbol{c}_0]\!]_1 &= [\![(-\delta, -\tau, \boxed{z'}, 0, \eta_0)\mathbf{W}]\!]_1;\ [\![\boldsymbol{c}_i]\!]_1 = [\![(\Psi_i(1, i), \delta u_i, \tau u_i, 0, \eta u_i)\widehat{\mathbf{W}}]\!]_1, \\
c^{(b)} &= [\![z]\!]_T \cdot m^{(b)}
\end{aligned}$$

where $b \xleftarrow{\$} \{0, 1\}$, $z' \xleftarrow{\$} \mathbb{Z}_p$ and the other randomness $\Psi_i, z, \eta_0, \delta$ are as in Game 2-ν-3.

Note that the advantage of Game 0 is equivalent to payload-hiding game, i.e., $\mathsf{Adv}^{\mathsf{UNIPE}}_{\mathcal{A},\text{Game }0}(\lambda) = \mathsf{Adv}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpPH}}(\lambda)$. The advantage of Game i can be represented as $\mathsf{Adv}^{\mathsf{UNIPE}}_{\mathcal{A},\text{Game }i}(\lambda) = |\Pr[R_i] - \frac{1}{2}|$. Since b is independently chosen from $\mathcal{A}$'s view in Game 3, we have $\mathsf{Adv}^{\mathsf{UNIPE}}_{\mathcal{A},\text{Game }3}(\lambda) = 0$ which implies $\Pr[R_3] = \frac{1}{2}$. Thus

$$\mathsf{Adv}^{\mathsf{UNIPE}}_{\mathcal{A},\text{Game }0}(\lambda) = \left|\Pr[R_0] - \frac{1}{2}\right| = |\Pr[R_0] - \Pr[R_3]|$$

$$= \Big| (\Pr[R_0] - \Pr[R_1]) + \textstyle\sum_{h=1}^{\nu} \big(\Pr[R_{(2\text{-}(h-1)\text{-}3)}] - \Pr[R_{(2\text{-}h\text{-}1)}]\big) +$$
$$\sum_{h=1}^{\nu} (\Pr[R_{(2\text{-}h\text{-}1)}] - \Pr[R_{(2\text{-}h\text{-}2)}]) + \sum_{h=1}^{\nu} \Big(\Pr[R_{(2\text{-}h\text{-}2)}] - \Pr[R_{(2\text{-}h\text{-}3)}]\Big)$$
$$+ \big(\Pr[R_{(2\text{-}\nu\text{-}3)}] - \Pr[R_3]\big) \Big|$$
$$\leq (3\nu + 1)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + \frac{1}{p} + 2^{-\Omega(\lambda)}$$

by Claim 1 to 5, proven below. □

Claim 1 : $\Big| \Pr[R_0] - \Pr[R_1] \Big| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$

Proof. We devise the reduction algorithm $\mathcal{B}$, that utilises the subroutine $\mathcal{A}$ to solve the SXDH problem. We show below how $\mathcal{B}$ uses the given SXDH instance that interpolate the Game 0 and Game 1 on receiving an instance $(\mathbb{BG}, [\![\mathfrak{a}]\!]_\iota = g_\iota^{\mathfrak{a}}, [\![\mathfrak{e}]\!]_\iota = g_\iota^{\mathfrak{e}}, [\![\mathfrak{t}_\beta]\!]_\iota = g_\iota^{\mathfrak{a}\mathfrak{e}+\beta\mathfrak{f}})$ of the SXDH assumption for $\iota = 1$ where $\mathfrak{a}, \mathfrak{f}, \mathfrak{e} \xleftarrow{\$} \mathbb{Z}_p$ and $\beta \xleftarrow{\$} \{0,1\}$. It chooses two random matrices $\mathbf{H} \xleftarrow{\$} \mathsf{GL}_5(\mathbb{Z}_p), \widehat{\mathbf{H}} \xleftarrow{\$} \mathsf{GL}_6(\mathbb{Z}_p)$ and implicitly sets the random dual orthonormal bases $(\mathbf{W}, \mathbf{W}^*)$ and $(\widehat{\mathbf{W}}, \widehat{\mathbf{W}}^*)$ as:

$$\mathbf{W} = \begin{bmatrix} 1 & & & & \\ & -1 & & & \\ & & \mathbf{I}_2 & & \\ & -\mathfrak{a} & & -1 & \end{bmatrix} \mathbf{H},\ \mathbf{W}^* = \begin{bmatrix} 1 & & & & \\ & -1 & & & \mathfrak{a} \\ & & \mathbf{I}_2 & & \\ & & & -1 & \end{bmatrix} \mathbf{H}^*,$$

$$\widehat{\mathbf{W}} = \begin{bmatrix} \mathbf{I}_3 & & & & \\ & -1 & & & \\ & & 1 & & \\ & -\mathfrak{a} & & -1 & \end{bmatrix} \widehat{\mathbf{H}},\ \widehat{\mathbf{W}}^* = \begin{bmatrix} \mathbf{I}_3 & & & & \\ & -1 & & & \mathfrak{a} \\ & & 1 & & \\ & & & -1 & \end{bmatrix} \widehat{\mathbf{H}}^*$$

where $\mathbf{H}^* = (\mathbf{H}^{-1})^\top, \widehat{\mathbf{H}}^* = (\widehat{\mathbf{H}}^{-1})^\top$. Since $[\![\mathfrak{a}]\!]_1$ is provided through the SXDH challenge, the algorithm $\mathcal{B}$ can compute $[\![\mathbf{W}]\!]_1$ and $[\![\widehat{\mathbf{W}}]\!]_1$. As the first, third and fourth rows $\boldsymbol{w}_1^*, \boldsymbol{w}_3^*, \boldsymbol{w}_4^*$ of $\mathbf{W}^*$ and the first three rows and fifth row $\widehat{\boldsymbol{w}}_1^*, \widehat{\boldsymbol{w}}_2^*, \widehat{\boldsymbol{w}}_3^*, \widehat{\boldsymbol{w}}_5^*$ of $\widehat{\mathbf{W}}^*$ are independent of a, the algorithm $\mathcal{B}$ can compute $[\![\boldsymbol{w}_1^*]\!]_2, [\![\boldsymbol{w}_3^*]\!]_2, [\![\boldsymbol{w}_4^*]\!]_2$ of $[\![\mathbf{W}^*]\!]_2$ and $[\![\widehat{\boldsymbol{w}}_1^*]\!]_2, [\![\widehat{\boldsymbol{w}}_2^*]\!]_2, [\![\widehat{\boldsymbol{w}}_3^*]\!]_2, [\![\widehat{\boldsymbol{w}}_5^*]\!]_2$ of $[\![\widehat{\mathbf{W}}^*]\!]_2$ and sets $\mathsf{msk} = (\boldsymbol{w}_1^*, \boldsymbol{w}_3^*, \boldsymbol{w}_4^*, \widehat{\boldsymbol{w}}_1^*, \widehat{\boldsymbol{w}}_2^*, \widehat{\boldsymbol{w}}_3^*, \widehat{\boldsymbol{w}}_5^*)$. To answer the j-th secret key query corresponding to the predicate vector $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$, $\mathcal{B}$ generates the components $[\![\boldsymbol{k}_0]\!]_2$ and $[\![\boldsymbol{k}_i]\!]_2$ of the secret key $\mathsf{sk}_{\boldsymbol{v}^{(j)}} = (\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}, [\![\boldsymbol{k}_0]\!]_2, ([\![\boldsymbol{k}_i]\!]_2)_{i \in I_{\boldsymbol{v}}^{(j)}})$ as:

$$[\![\boldsymbol{k}_0]\!]_2 = [\![(\omega, 0, 1, \phi_0, 0)\mathbf{W}^*]\!]_2;\ [\![\boldsymbol{k}_i]\!]_2 = [\![(\Upsilon_i(-i,1), \omega v_i^{(j)}, 0, \phi_i, 0)\widehat{\mathbf{W}}^*]\!]_2$$

where $\omega, \phi_0 \xleftarrow{\$} \mathbb{Z}_p$ and $\Upsilon_i, \phi_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in I_{\boldsymbol{v}^{(j)}}$. Note that $\mathcal{B}$ can compute $[\![\boldsymbol{k}_0]\!]_2, [\![\boldsymbol{k}_i]\!]_2$ using $[\![\boldsymbol{w}_1^*]\!]_2, [\![\boldsymbol{w}_3^*]\!]_2, [\![\boldsymbol{w}_4^*]\!]_2$ of $[\![\mathbf{W}^*]\!]_2$ and $[\![\widehat{\boldsymbol{w}}_1^*]\!]_2, [\![\widehat{\boldsymbol{w}}_2^*]\!]_2, [\![\widehat{\boldsymbol{w}}_3^*]\!]_2, [\![\widehat{\boldsymbol{w}}_5^*]\!]_2$ of $[\![\widehat{\mathbf{W}}^*]\!]_2$. To generate the challenge ciphertext components $[\![\boldsymbol{c}_0]\!]_1, [\![\boldsymbol{c}_i]\!]_1$ of $\mathsf{ct}_{\boldsymbol{u}}^{(b)} = \big(\boldsymbol{u}, [\![\boldsymbol{c}_0]\!]_1, ([\![\boldsymbol{c}_i]\!]_1)_{i\in I_{\boldsymbol{u}}}, c^{(b)} = [\![z]\!]_T \cdot m^{(b)}\big)$, $\mathcal{B}$ simulates as follows:

$$\begin{aligned}
[\![\boldsymbol{c}_0]\!]_1 &= [\![(-\delta, 0, z, 0, \eta')\mathbf{W} + (0, \mathfrak{t}_\beta, 0, 0, \mathfrak{e})\mathbf{H}]\!]_1 \\
&= [\![(-\delta, \beta\mathfrak{f}, z, 0, \eta' - \mathfrak{e})\mathbf{W}]\!]_1 \\
[\![\boldsymbol{c}_i]\!]_1 &= [\![(\Psi_i(1,i), \delta u_i, 0, 0, 0)\widehat{\mathbf{W}} + u_i(0, 0, 0, \mathfrak{t}_\beta, 0, \mathfrak{e})\widehat{\mathbf{H}}]\!]_1 \\
&= [\![(\Psi_i(1,i), \delta u_i, -\beta\mathfrak{f} u_i, 0, -\mathfrak{e} u_i)\widehat{\mathbf{W}}]\!]_1
\end{aligned}$$

where $\delta, z, \eta' \xleftarrow{\$} \mathbb{Z}_p$ and $\Psi_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in I_{\boldsymbol{u}}$. To compute $[\![\boldsymbol{c}_0]\!]_1, [\![\boldsymbol{c}_i]\!]_1$, the rows $[\![\boldsymbol{w}_1]\!]_1, [\![\boldsymbol{w}_3]\!]_1, [\![\boldsymbol{w}_5]\!]_1$ of $[\![\mathbf{W}]\!]_1$ and $[\![\widehat{\boldsymbol{w}}_1]\!]_1, [\![\widehat{\boldsymbol{w}}_2]\!]_1, [\![\widehat{\boldsymbol{w}}_3]\!]_1, [\![\widehat{\boldsymbol{w}}_6]\!]_1$ of $[\![\widehat{\mathbf{W}}]\!]_1$ are sufficient. The second row of $[\![\mathbf{W}^*]\!]_2$ and the fourth row of $[\![\widehat{\mathbf{W}}]\!]_2^*$ contain the unknown element $[\![\mathfrak{a}]\!]_2$. However, for secret key simulation, the second row of $[\![\mathbf{W}]\!]_2^*$ and fourth row of $[\![\widehat{\mathbf{W}}]\!]_2$ are not required as the second entry of the vector $[\![\boldsymbol{k}_0]\!]_2$ and the fourth entry of the vector $[\![\boldsymbol{k}_i]\!]_2$ are set as zero in both Game 0 and Game 1. Letting $\eta_0 = \eta' - \mathfrak{e}$ and $\eta = -\mathfrak{e}$, the $\mathcal{A}$'s view is the same as in Game 0 for $\beta = 0$ since the second and fourth entries of the vectors $[\![\boldsymbol{c}_0]\!]_1, [\![\boldsymbol{c}_i]\!]_1$ are zero in Game 0. Letting $\beta\mathfrak{f} = -\tau, -\beta\mathfrak{f} u_i = \tau u_i$, the $\mathcal{A}$'s view is the same as in Game 1 for $\beta = 1$ unless $f = 0$. Hence, $|\Pr[R_0] - \Pr[R_1]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$. □

Claim 2: $\Big|\Pr[R_{2\text{-}(h-1)\text{-}3}] - \Pr[R_{2\text{-}h\text{-}1}]\Big| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$

Proof. Suppose $\mathcal{B}$ obtains an SXDH instance $(\mathbb{BG}, [\![\mathfrak{a}]\!]_\iota = g_\iota^{\mathfrak{a}}, [\![\mathfrak{e}]\!]_\iota = g_\iota^{\mathfrak{e}}, [\![\mathfrak{t}_\beta]\!]_\iota = g_\iota^{\mathfrak{a}\mathfrak{e}+\beta\mathfrak{f}})$ for $\iota = 2$ where $\mathfrak{a}, \mathfrak{f}, \mathfrak{e} \xleftarrow{\$} \mathbb{Z}_p$ and $\beta \xleftarrow{\$} \{0,1\}$. We explicitly show below how $\mathcal{B}$ uses this instance to interpolate between Game 2-$(h-1)$-3 and Game 2-h-1. The reduction algorithm $\mathcal{B}$ picks $\widehat{\mathbf{H}} \xleftarrow{\$} \mathsf{GL}_6(\mathbb{Z}_p)$ and implicitly sets the random dual orthonormal bases $(\widehat{\mathbf{W}}, \widehat{\mathbf{W}}^*)$ as:

$$\widehat{\mathbf{W}} = \begin{bmatrix} I_2 & & & \\ & 0 & -1 & \\ & 1 & \mathfrak{a} & \\ & & & \mathbf{I}_2 \end{bmatrix} \widehat{\mathbf{H}}, \quad \widehat{\mathbf{W}}^* = \begin{bmatrix} I_2 & & & \\ & \mathfrak{a} & -1 & \\ & 1 & 0 & \\ & & & \mathbf{I}_2 \end{bmatrix} \widehat{\mathbf{H}}^*$$

where $\widehat{\mathbf{H}}^* = (\widehat{\mathbf{H}}^{-1})^\top$. Observe that $[\![\mathfrak{a}]\!]_2 = g_2^{\mathfrak{a}}$ and the reduction algorithm $\mathcal{B}$ can efficiently calculate the first three rows and the fifth row of $[\![\widehat{\mathbf{W}}^*]\!]_2$

using the given SXDH challenge instance. To simulate the h-th secret key $\mathsf{sk}_{\boldsymbol{v}^{(h)}} = (\boldsymbol{v}^{(h)} = (v_i^{(h)})_{i \in I_{\boldsymbol{v}^{(h)}}}, [\![\boldsymbol{k}_0]\!]_2, ([\![\boldsymbol{k}_i]\!]_2)_{i \in I_{\boldsymbol{v}^{(h)}}})$ associated with the h-th predicate vector $\boldsymbol{v}^{(h)} = (v_i^{(h)})_{i \in I_{\boldsymbol{v}^{(h)}}}$, $\mathcal{B}$ computes the secret key component $[\![\boldsymbol{k}_i]\!]_2$ as follows:

$$\begin{aligned}[\![\boldsymbol{k}_i]\!]_2 &= [\![(\Upsilon_i(-i,1), \omega' v_i^{(h)}, 0, \phi_i, 0)\widehat{\mathbf{W}}^* + v_i^{(h)}(0,0,\mathfrak{t}_\beta, -\mathfrak{e}, 0, 0)\widehat{\mathbf{H}}^*]\!]_2 \\ &= [\![(\Upsilon_i(-i,1), (\omega' + \mathfrak{e})v_i^{(h)}, \beta\mathfrak{f} v_i^{(h)}, \phi_i, 0)\widehat{\mathbf{W}}^*]\!]_2\end{aligned}$$

where $\omega' \xleftarrow{\$} \mathbb{Z}_p$ and $\Upsilon_i, \phi_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in I_{\boldsymbol{v}^{(h)}}$. To simulate the challenge ciphertext $\mathsf{ct}_{\boldsymbol{u}}^{(b)} = \big(\boldsymbol{u}, [\![\boldsymbol{c}_0]\!]_1, ([\![\boldsymbol{c}_i]\!]_1)_{i \in I_{\boldsymbol{u}}}, c^{(b)} = [\![z]\!]_T \cdot m^{(b)}\big)$, $\mathcal{B}$ computes the component $[\![\boldsymbol{c}_i]\!]_1$ as:

$$\begin{aligned}[\![\boldsymbol{c}_i]\!]_1 &= [\![(\Psi_i(1,i), \delta' u_i, 0, 0, \eta u_i)\widehat{\mathbf{W}} + u_i(0,0,\tau,0,0,0)\widehat{\mathbf{H}}]\!]_1 \\ &= [\![(\Psi_i(1,i), (\delta' + \mathfrak{a}\tau)u_i, \tau u_i, 0, \eta u_i)\widehat{\mathbf{W}}]\!]_1\end{aligned}$$

where $\tau, \delta', \eta \xleftarrow{\$} \mathbb{Z}_p$ and $\Psi_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in I_{\boldsymbol{u}}$. The fourth row $[\![\widehat{\boldsymbol{w}}_4]\!]_1$ of $[\![\widehat{\mathbf{W}}]\!]_1$ is not computable by $\mathcal{B}$ as it contains $[\![\mathfrak{a}]\!]_1$. We implicitly set $\delta = \delta' + \mathfrak{a}\tau, \omega = \omega' + \mathfrak{e}$. For $\beta = 0$, the $\mathcal{A}$'s view is the same as in Game 2-$(h-1)$-3 as the fourth entry of the vector $\boldsymbol{k}_i$ is zero and for $\beta = 1$, the $\mathcal{A}$'s view is similar to that of Game 2-h-1 as the entry of the fourth position of the vector $\boldsymbol{k}_i$ is $\rho v_i^{(h)} = \mathfrak{f} v_i^{(h)}$ unless $\mathfrak{f} = 0$. Therefore, $\mathcal{B}$ can interpolate between the two Game 2-$(h-1)$-3 and Game 2-h-1 and hence $|\Pr[R_{\text{2-}(h-1)\text{-3}}] - \Pr[R_{\text{2-}h\text{-1}}]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$. □

Claim 3: $\Big|\Pr[R_{\text{2-}h\text{-1}}] - \Pr[R_{\text{2-}h\text{-2}}]\Big| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$

Proof. Suppose $\mathcal{B}$ obtains the SXDH challenge instance $(\mathbb{BG}, [\![\mathfrak{a}]\!]_\iota = g_\iota^{\mathfrak{a}}, [\![\mathfrak{e}]\!]_\iota = g_\iota^{\mathfrak{e}}, [\![\mathfrak{t}_\beta]\!]_\iota = g_\iota^{\mathfrak{a}\mathfrak{e}+\beta\mathfrak{f}})$ for $\iota = 2$ where $\mathfrak{a}, \mathfrak{f}, \mathfrak{e} \xleftarrow{\$} \mathbb{Z}_p$ and $\beta \xleftarrow{\$} \{0,1\}$. The reduction algorithm $\mathcal{B}$ picks $\mathbf{H} \xleftarrow{\$} \mathsf{GL}_5(\mathbb{Z}_p)$ and implicitly sets the random dual orthonormal bases $(\mathbf{W}, \mathbf{W}^*)$ as:

$$\mathbf{W} = \begin{bmatrix} 1 & & & & \\ a & 1 & & & \\ & & & & \\ & & & & \\ & & & & \mathbf{I}_3 \end{bmatrix} \mathbf{H}, \quad \mathbf{W}^* = \begin{bmatrix} 1 & -a & & & \\ & 1 & & & \\ & & & & \\ & & & & \\ & & & & \mathbf{I}_3 \end{bmatrix} \mathbf{H}^*$$

where $\mathbf{H}^* = (\mathbf{H}^{-1})^\top$. Observe that $[\![\mathfrak{a}]\!]_2 = g_2^{\mathfrak{a}}$ and the reduction algorithm $\mathcal{B}$ can efficiently calculate the first, third and fourth rows of $[\![\mathbf{W}^*]\!]_2$ using the given SXDH challenge instances. To simulate the h-th secret key $\mathsf{sk}_{\boldsymbol{v}^{(h)}} = (\boldsymbol{v}^{(h)} = (v_i^{(h)})_{i \in I_{\boldsymbol{v}^{(h)}}}, [\![\boldsymbol{k}_0]\!]_2, ([\![\boldsymbol{k}_i]\!]_2)_{i \in I_{\boldsymbol{v}^{(h)}}})$, $\mathcal{B}$ computes the secret key component $[\![\boldsymbol{k}_0]\!]_2$ as:

$$[\![\boldsymbol{k}_0]\!]_2 = [\![(0,\rho,1,\phi_0,0)\mathbf{W}^* + (-\mathfrak{e},\mathfrak{t}_\beta,0,0,0)\mathbf{H}^*]\!]_2 = [\![(-\mathfrak{e},(\rho+\beta\mathfrak{f}),1,\phi_0,0)\mathbf{W}^*]\!]_2$$

where $\rho, \phi_0 \xleftarrow{\$} \mathbb{Z}_p$. The challenger $\mathcal{B}$ sets the ciphertext component $[\![\boldsymbol{c}_0]\!]_1$ of the challenge ciphertext $\mathsf{ct}_{\boldsymbol{u}}^{(b)} = \big(\boldsymbol{u}, [\![\boldsymbol{c}_0]\!]_1, ([\![\boldsymbol{c}_i]\!]_1)_{i\in I_{\boldsymbol{u}}}, c^{(b)} = [\![z]\!]_T \cdot m^{(b)}\big)$ as:

$$[\![\boldsymbol{c}_0]\!]_1 = [\![(\delta',0,z,0,\eta_0)\mathbf{W} + (0,-\tau,0,0,0)\mathbf{H}]\!]_1 = [\![(\delta'+\mathfrak{a}\tau,-\tau,z,0,\eta_0)\mathbf{W}]\!]_1$$

where $\delta', \eta_0, z \xleftarrow{\$} \mathbb{Z}_p$. The second row of $[\![\mathbf{W}]\!]_1$ contains $[\![\mathfrak{a}]\!]_1$ and so that $\mathcal{B}$ cannot compute $[\![\boldsymbol{w}_2]\!]_1$. We implicitly set $\omega = -\mathfrak{e}, -\delta = \delta' + \mathfrak{a}\tau$. For $\beta = 0$, the $\mathcal{A}$'s view is the same as in Game 2-h-1 since the second entry of $\boldsymbol{k}_0$ is ρ. For $\beta = 1$, the $\mathcal{A}$'s view is identical to that in Game 2-h-2 as the second entry of $[\![\boldsymbol{k}_0]\!]_2$ is $\zeta = \rho + \mathfrak{f}$ which is a uniform element from $\mathbb{Z}_p$ unless $\mathfrak{f} = 0$. Therefore, the challenger can interpolate between the two games Game 2-h-1 to Game 2-h-2 and hence $|\Pr[R_{2\text{-}h\text{-}1}] - \Pr[R_{2\text{-}h\text{-}2}]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$. □

Claim 4: $\Big|\Pr[R_{2\text{-}h\text{-}2}] - \Pr[R_{2\text{-}h\text{-}3}]\Big| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$

Similarly follows as Claim 3.

Claim 5: $\Big|\Pr[R_{2\text{-}\nu\text{-}3}] - \Pr[R_3]\Big| \leq \dfrac{1}{p}$

Proof. We now show that the distribution of the ν-th secret key $\mathsf{sk}_{\boldsymbol{v}^{(\nu)}} = (\boldsymbol{v}^{(\nu)} = (v_i^{(\nu)})_{i\in I_{\boldsymbol{v}^{(\nu)}}}, [\![\boldsymbol{k}_0]\!]_2, ([\![\boldsymbol{k}_i]\!]_2)_{i\in I_{\boldsymbol{v}^{(\nu)}}})$ corresponding to the ν-th predicate vector $\boldsymbol{v}^{(\nu)} = (v_i^{(\nu)})_{i\in I_{\boldsymbol{v}^{(\nu)}}}$ are identical in Game 2-ν-3 and Game 3. Also the challenge ciphertext $\mathsf{ct}_{\boldsymbol{u}}^{(b)} = \big(\boldsymbol{u}, [\![\boldsymbol{c}_0]\!]_1, ([\![\boldsymbol{c}_i]\!]_1)_{i\in I_{\boldsymbol{u}}}, c^{(b)} = [\![z]\!]_T \cdot m^{(b)}\big)$ corresponding to the challenge attribute-message pair $(\boldsymbol{u}, m^{(b)})$ are also identically distributed as in the two games Game 2-ν-3 and Game 3. We replace the dual orthonormal basis $(\mathbf{W}, \mathbf{W}^*)$ by another dual orthonormal basis $(\mathbf{H}, \mathbf{H}^*)$ by choosing $\chi \xleftarrow{\$} \mathbb{Z}_p$, setting $\boldsymbol{h}_1 = \boldsymbol{w}_1, \boldsymbol{h}_2 = \boldsymbol{w}_2 + \chi\boldsymbol{w}_3, \boldsymbol{h}_3 = \boldsymbol{w}_3, \boldsymbol{h}_4 = \boldsymbol{w}_4, \boldsymbol{h}_5 = \boldsymbol{w}_5$, $\boldsymbol{h}_1^* = \boldsymbol{w}_1^*, \boldsymbol{h}_2^* = \boldsymbol{w}_2^*, \boldsymbol{h}_3 = \boldsymbol{w}_3^* - \chi\boldsymbol{w}_2^*, \boldsymbol{h}_4^* = \boldsymbol{w}_4^*, \boldsymbol{h}_5^* = \boldsymbol{w}_5^*$ and defining

$$\mathbf{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3, \boldsymbol{h}_4, \boldsymbol{h}_5), \quad \mathbf{H}^* = (\boldsymbol{h}_1^*, \boldsymbol{h}_2^*, \boldsymbol{h}_3^*, \boldsymbol{h}_4^*, \boldsymbol{h}_5^*)$$

Observe that, $(\mathbf{H}, \mathbf{H}^*)$ satisfies the properties of dual orthonormality and it is identically distributed as $(\mathbf{W}, \mathbf{W}^*)$ and consequently $(\mathbf{H}, \mathbf{H}^*)$ and $(\mathbf{W}, \mathbf{W}^*)$ are identically distributed. Note that the ν-th secret key $\mathsf{sk}_{\boldsymbol{v}^{(\nu)}} = (\boldsymbol{v}^{(\nu)} = (v_i^{(\nu)})_{i\in I_{\boldsymbol{v}^{(\nu)}}}, [\![\boldsymbol{k}_0]\!]_2, ([\![\boldsymbol{k}_i]\!]_2)_{i\in I_{\boldsymbol{v}^{(\nu)}}})$ corresponding to the ν-th predicate vector $\boldsymbol{v}^{(\nu)} = (v_i^{(\nu)})_{i\in I_{\boldsymbol{v}^{(\nu)}}}$ where $[\![\boldsymbol{k}_0]\!]_2 = [\![(\omega,\zeta,1,\phi_0,0)\mathbf{W}^*]\!]_2, [\![\boldsymbol{k}_i]\!]_2 = [\![(\Upsilon_i(-i,1), \omega y_i^{(\nu)},$

$0, \phi_i, 0)\widehat{\mathbf{W}}^*]\!]_2$ for both Game 2-ν-3 and Game 3. On the other hand, in Game 2-ν-3, the challenge ciphertext is $\mathsf{ct}_{\boldsymbol{u}}^{(b)} = (\boldsymbol{u}, [\![\boldsymbol{c}_0]\!]_1, ([\![\boldsymbol{c}_i]\!]_1)_{i\in I_{\boldsymbol{u}}}, c^{(b)} = [\![z]\!]_T \cdot m^{(b)})$ where $[\![\boldsymbol{c}_0]\!]_1 = [\![(-\delta, -\tau, z, \eta_0, 0)\mathbf{W}]\!]_1, [\![\boldsymbol{c}_i]\!]_1 = [\![(\Psi_i(1,i), \delta u_i, \tau u_i, 0, \eta u_i)\widehat{\mathbf{W}}]\!]_1$ and in Game 3, the challenge ciphertext is $\mathsf{ct}_{\boldsymbol{u}}^{(b)} = (\boldsymbol{u}, [\![\boldsymbol{c}_0]\!]_1, ([\![\boldsymbol{c}_i]\!]_1)_{i\in I_{\boldsymbol{u}}}, c^{(b)} = [\![z]\!]_T \cdot m^{(b)})$ where $[\![\boldsymbol{c}_0]\!]_1 = [\![(-\delta, -\tau, z', \eta_0, 0)\mathbf{W}]\!]_1, [\![\boldsymbol{c}_i]\!]_1 = [\![(\Psi_i(1,i), \delta u_i, \tau u_i, 0, \eta u_i)\widehat{\mathbf{W}}]\!]_1$. The ν-th secret key component $[\![\boldsymbol{k}_0]\!]_2$ and the challenge ciphertext component $[\![\boldsymbol{c}_0]\!]_1$ uses $\mathbf{W}^*$ and $\mathbf{W}$ respectively in Game 2-ν-3 and the orthonormal basis $(\mathbf{H}, \mathbf{H}^*)$ in Game 3 as follows:

$$
\begin{aligned}
[\![\boldsymbol{k}_0]\!]_2 &= [\![(\omega, \zeta', 1, \phi_0, 0)\mathbf{W}^*]\!]_2 \\
&= [\![\omega \cdot \boldsymbol{w}_1^* + \zeta' \cdot \boldsymbol{w}_2^* + 1 \cdot (\boldsymbol{h}_3^* + \chi \boldsymbol{w}_2^*) + \phi_0 \cdot \boldsymbol{w}_4^* + 0 \cdot \boldsymbol{w}_5^*]\!]_2 \\
&= [\![\omega \cdot \boldsymbol{w}_1^* + (\zeta' + \chi) \cdot \boldsymbol{w}_2^* + 1 \cdot \boldsymbol{h}_3^* + \phi_0 \cdot \boldsymbol{w}_4^* + 0 \cdot \boldsymbol{w}_5^*]\!]_2 \\
&= [\![(\omega, \zeta, 1, \phi_0, 0)\mathbf{H}^*]\!]_2 \\
[\![\boldsymbol{c}_0]\!]_1 &= [\![(-\delta, -\tau, z, 0, \eta_0)\mathbf{W}]\!]_1 \\
&= [\![-\delta \cdot \boldsymbol{w}_1 - \tau \cdot (\boldsymbol{h}_2 - \chi \boldsymbol{w}_3) + z \cdot \boldsymbol{w}_3 + 0 \cdot \boldsymbol{w}_4 + \eta_0 \cdot \boldsymbol{w}_5]\!]_1 \\
&= [\![-\delta \cdot \boldsymbol{w}_1 - \tau \cdot \boldsymbol{h}_2 + (z + \tau\chi) \boldsymbol{w}_3 + 0 \cdot \boldsymbol{w}_4 + \eta_0 \cdot \boldsymbol{w}_5]\!]_1 \\
&= [\![(-\delta, -\tau, z', 0, \eta_0)\mathbf{H}]\!]_1
\end{aligned}
$$

We implicitly set $z + \tau\chi = z'$ and $\zeta = \zeta' + \chi$. Therefore, $\mathcal{A}$'s view for both the bases $(\mathbf{W}, \mathbf{W}^*)$ and $(\mathbf{H}, \mathbf{H}^*)$ are consistent with respect to the master public key since $\omega, \phi_0, \delta, z, \chi, \tau, \eta_0, \zeta' \xleftarrow{\$} \mathbb{Z}_p$ and follows similar distribution except with the probability $\frac{1}{p}$ for $\chi = 0$. More precisely, $\mathsf{mpk} = (\mathsf{pp}, [\![\boldsymbol{w}_1]\!]_1, [\![\boldsymbol{w}_3]\!]_1, [\![\boldsymbol{w}_5]\!]_1, [\![\widehat{\boldsymbol{w}}_1]\!]_1, [\![\widehat{\boldsymbol{w}}_2]\!]_1, [\![\widehat{\boldsymbol{w}}_3]\!]_1, [\![\widehat{\boldsymbol{w}}_6]\!]_1)$ uses dual orthonormal basis $(\mathbf{W}, \mathbf{W}^*)$ and $\mathsf{mpk}' = (\mathsf{pp}, [\![\boldsymbol{h}_1]\!]_1, [\![\boldsymbol{h}_3]\!]_1, [\![\boldsymbol{h}_5]\!]_1, [\![\widehat{\boldsymbol{w}}_1]\!]_1, [\![\widehat{\boldsymbol{w}}_2]\!]_1, [\![\widehat{\boldsymbol{w}}_3]\!]_1, [\![\widehat{\boldsymbol{w}}_6]\!]_1)$ uses dual orthonormal basis $(\mathbf{H}, \mathbf{H}^*)$ which are identically distributed and $(\widehat{\mathbf{W}}, \widehat{\mathbf{W}}^*)$ are defined in Game 0. Also, the challenge ciphertext and the secret key components can be expressed as in Game 2-ν-3 depending on the basis $(\mathbf{W}, \mathbf{W}^*)$ and Game 3 preserves the same distribution depending on the basis $(\mathbf{H}, \mathbf{H}^*)$. Therefore, replacing $(\mathbf{W}, \mathbf{W}^*)$ by $(\mathbf{H}, \mathbf{H}^*)$ converts Game 2-ν-3 to Game 3 except with probability $\frac{1}{p}$. Hence $|\Pr[R_{2\text{-}\nu\text{-}3}] - \Pr[R_3]| \leq \frac{1}{p}$. □ □

4. Our Attribute-hiding UNIPE

In this section, we provide a generic transformation to construct a weakly attribute-hiding unbounded non-zero inner product encryption $\mathsf{UNIPE} = (\mathsf{Setup}, \mathsf{KeyExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ from an adaptive indistinguishability (AdpIND) secure $\mathsf{UIPFE} = (\mathsf{Setup}, \mathsf{KeyExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ with the mes-

sage vector space $\mathcal{U}'_\lambda$, key vector space $\mathcal{V}'_\lambda$ and an inner product space $\mathcal{I}_\lambda$.

We describe below our $\mathsf{UNIPE} = (\mathsf{Setup}, \mathsf{KeyExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ with the attribute space $\mathcal{U}_\lambda = \mathcal{U}'_\lambda$, the predicate space $\mathcal{V}_\lambda = \mathcal{V}'_\lambda$, and a polynomially bounded message space $\mathcal{M}_\lambda \subseteq \mathcal{I}_\lambda$ such that for any $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathcal{U}_\lambda$ such that $\phi \neq I_{\boldsymbol{u}}$ and $m \in \mathcal{M}_\lambda$, the vector $m\boldsymbol{u} = (mu_i)_{i \in I_{\boldsymbol{u}}} \in \mathcal{U}_\lambda$. We describe below our generic UNIPE scheme:

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathbf{Setup}(1^\lambda)$: On input of the security parameter λ, a trusted authority generates $(\mathsf{mpk}^{\mathsf{UIPFE}}, \mathsf{msk}^{\mathsf{UIPFE}}) \leftarrow \mathsf{UIPFE.Setup}(1^\lambda)$ and outputs the master public key $\mathsf{mpk} = \mathsf{mpk}^{\mathsf{UIPFE}}$ and the master secret key $\mathsf{msk} = \mathsf{msk}^{\mathsf{UIPFE}}$. The trusted authority keeps msk private while mpk is made public.

$\mathsf{sk}_{\boldsymbol{v}} \leftarrow \mathbf{KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v})$: On input $\mathsf{mpk}, \mathsf{msk}$ and a predicate vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}} \in \mathcal{V}_\lambda$ such that $\phi \neq I_{\boldsymbol{v}} \subseteq [s_1]$ (where $s_1 = s_1(\lambda)$ is an integer), the trusted authority computes $\mathsf{sk}_{\boldsymbol{v}}^{\mathsf{UIPFE}} \leftarrow \mathsf{UIPFE.KeyExtract}(\mathsf{mpk}^{\mathsf{UIPFE}}, \mathsf{msk}^{\mathsf{UIPFE}}, \boldsymbol{v})$ and outputs the secret key $\mathsf{sk}_{\boldsymbol{v}} = \mathsf{sk}_{\boldsymbol{v}}^{\mathsf{UIPFE}}$.

$\mathsf{ct}_{\boldsymbol{u}} \leftarrow \mathbf{Encrypt}(\mathsf{mpk}, \boldsymbol{u}, m)$: On input mpk, an attribute vector $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathcal{U}_\lambda$ such that $\phi \neq I_{\boldsymbol{u}} \subseteq [s_2]$ (where $s_2 = s_2(\lambda)$ is an integer) and a message $m \in \mathcal{M}_\lambda$, the encryptor computes $\mathsf{ct}_{m\cdot\boldsymbol{u}}^{\mathsf{UIPFE}} \leftarrow \mathsf{UIPFE.Encrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, m\boldsymbol{u})$ and $\mathsf{ct}_{\boldsymbol{u}}^{\mathsf{UIPFE}} \leftarrow \mathsf{UIPFE.Encrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, \boldsymbol{u})$. It outputs the ciphertext $\mathsf{ct}_{\boldsymbol{u}} = (\mathsf{ct}_{m\cdot\boldsymbol{u}}^{\mathsf{UIPFE}}, \mathsf{ct}_{\boldsymbol{u}}^{\mathsf{UIPFE}})$.

$(\alpha \text{ or } \perp) \leftarrow \mathbf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{v}}, \mathsf{ct}_{\boldsymbol{u}})$: The decryptor takes as input mpk, $\mathsf{sk}_{\boldsymbol{v}}$ associated with a predicate vector $\boldsymbol{v}$ for an index set $I_{\boldsymbol{v}} \neq \phi$, the ciphertext $\mathsf{ct}_{\boldsymbol{u}}$ corresponding to an attribute vector $\boldsymbol{u}$ with an index set $I_{\boldsymbol{u}} \neq \phi$ and a message m. The decryption proceeds by computing

$$H \leftarrow \mathsf{UIPFE.Decrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, \mathsf{sk}_{\boldsymbol{v}}^{\mathsf{UIPFE}}, \mathsf{ct}_{m\cdot\boldsymbol{u}}^{\mathsf{UIPFE}})$$
$$h \leftarrow \mathsf{UIPFE.Decrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, \mathsf{sk}_{\boldsymbol{v}}^{\mathsf{UIPFE}}, \mathsf{ct}_{\boldsymbol{u}}^{\mathsf{UIPFE}}).$$

and returning $\alpha = H \cdot h^{-1} \in \mathcal{M}_\lambda$ if $h \neq 0$ and $\perp$ if $h = 0$.

Correctness: Our UNIPE's correctness is derived from the correctness of the underlying UIPFE scheme. It is observed that

$$\mathsf{UIPFE.Decrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, \mathsf{sk}_{\boldsymbol{v}}^{\mathsf{UIPFE}}, \mathsf{ct}_{m\cdot\boldsymbol{u}}^{\mathsf{UIPFE}}) \rightarrow H = \langle m\boldsymbol{u}, \boldsymbol{v} \rangle = m\langle \boldsymbol{u}, \boldsymbol{v} \rangle$$
$$\mathsf{UIPFE.Decrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, \mathsf{sk}_{\boldsymbol{v}}^{\mathsf{UIPFE}}, \mathsf{ct}_{\boldsymbol{u}}^{\mathsf{UIPFE}}) \rightarrow h = \langle \boldsymbol{u}, \boldsymbol{v} \rangle.$$

The decryption is successful for $h = \langle \boldsymbol{u}, \boldsymbol{v} \rangle \neq 0$ as $\alpha = H \cdot h^{-1} = m \cdot \langle \boldsymbol{u}, \boldsymbol{v} \rangle \cdot \langle \boldsymbol{u}, \boldsymbol{v} \rangle^{-1} = m$. For $h = \langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0$, the decryption fails to extract m.

4.1. Security

Theorem 4.1. *Assuming the underlying* UIPFE *scheme is adaptively indistinguishable (*AdpIND*) secure as per Definition 8, our generic construction of* UNIPE *described above achieves adaptive weakly attribute-hiding (*AdpWAH*) security as per Definition 6.*

Proof. We consider the following game of sequence to prove the above theorem. We begin with Game 0, the real AdpWAH security game according to Definition 6, in which the challenger chooses a challenge bit $b = 0$. We transform Game 0 for the challenge bit $b = 0$ to Game 2 for the challenge bit $b = 1$ using Game 1 as an intermediate game. Now, we employ the AdpIND security of the underlying UIPFE scheme to achieve indistinguishability between the consecutive games. Let E_i represents the event where $b = b'$ in Game i, where b' is the bit emitted by the UNIPE adversary $\mathcal{A}$ during the guessing phase. Here is a description of the games:

Game 0: In Game 0, the UIPFE adversary $\mathcal{B}$ servers as UNIPE challenger $\mathcal{C}$.

1. **Setup:** Firstly, UIPFE-challenger $\mathcal{C}$ generates $(\mathsf{mpk}^{\mathsf{UIPFE}}, \mathsf{msk}^{\mathsf{UIPFE}}) \leftarrow \mathsf{UIPFE.Setup}(1^\lambda)$ and sends the master public key $\mathsf{mpk}^{\mathsf{UIPFE}}$ to $\mathcal{B}$, who forwards $\mathsf{mpk}^{\mathsf{UIPFE}}$ as master public key mpk of UNIPE scheme to $\mathcal{A}$.
2. **Pre-key query:** $\mathcal{A}$ can query polynomially many secret keys on the predicate vectors. Corresponding to the j-th secret key query on the predicate vector $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$ with non-empty index set $I_{\boldsymbol{v}^{(j)}}$, the challenger $\mathcal{B}$ sends the same predicate vector $\boldsymbol{v}^{(j)}$ to the UIPFE-challenger $\mathcal{C}$. The UIPFE-challenger $\mathcal{C}$ issues to $\mathcal{B}$ the secret key $\mathsf{sk}_{\boldsymbol{v}^{(j)}}^{\mathsf{UIPFE}} \leftarrow \mathsf{UIPFE.KeyExtract}(\mathsf{mpk}^{\mathsf{UIPFE}}, \mathsf{msk}^{\mathsf{UIPFE}}, \boldsymbol{v}^{(j)})$ and $\mathcal{B}$ forwards the secret key $\mathsf{sk}_{\boldsymbol{v}^{(j)}} = \mathsf{sk}_{\boldsymbol{v}^{(j)}}^{\mathsf{UIPFE}}$ to $\mathcal{A}$.
3. **Challenge query:** The challenge messages pair $(m^{(0)}, m^{(1)})$ along with the challenge attributes pair $(\boldsymbol{u}^{(0)} = (u_i^{(0)})_{i \in I_{\boldsymbol{u}^{(0)}}}, \boldsymbol{u}^{(1)} = (u_i^{(0)})_{i \in I_{\boldsymbol{u}^{(0)}}})$ (where the non-empty index sets $I_{\boldsymbol{u}^{(0)}} = I_{\boldsymbol{u}^{(1)}} = I_{\boldsymbol{u}}$) are submitted by $\mathcal{A}$ to $\mathcal{B}$ satisfying the following
   (a) if $m^{(0)} \neq m^{(1)}$ then $\langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = 0 = \langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle$
   (b) if $m^{(0)} = m^{(1)}$ then $\langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = \langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle \neq 0$

   for all predicate vectors $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$ on which pre-key query has been made by $\mathcal{A}$. The challenger $\mathcal{B}$ computes $m^{(0)}\boldsymbol{u}^{(0)}$ and $m^{(1)}\boldsymbol{u}^{(1)}$ and sends the pairs $(\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}), (m^{(0)}\boldsymbol{u}^{(0)}, m^{(1)}\boldsymbol{u}^{(1)})$ to the UIPFE-challenger $\mathcal{C}$. The UIPFE-challenger $\mathcal{C}$ fixes a bit $b = 0$ and generates the challenge ciphertext $\mathsf{ct}^{(0,0)} = (\mathsf{ct}_{m^{(0)}\boldsymbol{u}^{(0)}}^{\mathsf{UIPFE}}, \mathsf{ct}_{\boldsymbol{u}^{(0)}}^{\mathsf{UIPFE}})$ where the ciphertext components $\mathsf{ct}_{m^{(0)}\boldsymbol{u}^{(0)}}^{\mathsf{UIPFE}} \leftarrow \mathsf{UIPFE.Encrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, m^{(0)}\boldsymbol{u}^{(0)})$ and $\mathsf{ct}_{\boldsymbol{u}^{(0)}}^{\mathsf{UIPFE}} \leftarrow$

$\mathsf{UIPFE.Encrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, \boldsymbol{u}^{(0)})$[1] and forwards $\mathsf{ct}^{(0,0)}$ to $\mathcal{B}$. Finally, $\mathcal{B}$ sets the challenge ciphertext $\mathsf{ct}_{\boldsymbol{u}^{(0)}} = \mathsf{ct}^{(0,0)}$ and forwards this to $\mathcal{A}$.

4. **Post-key query:** The pre-key query phase is repeated with the same constraint for any key query as specified in the challenge queries phase on the predicate vector $\boldsymbol{v}^{(j)}$, the challenge attributes pair $(\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)})$, and the messages pair $(m^{(0)}, m^{(1)})$.

Therefore, Game 0 is identical with the experiment $\mathsf{Exp}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpWAH}}(1^\lambda)$ described in Definition 6 for $b = 0$.

Game 1: This game is the same as Game 0 except that the second component $\mathsf{ct}^{\mathsf{UIPFE}}_{\boldsymbol{u}^{(0)}}$ of the challenge ciphertext $\mathsf{ct}^{(0,0)} = (\mathsf{ct}^{\mathsf{UIPFE}}_{m^{(0)}\boldsymbol{u}^{(0)}}, \mathsf{ct}^{\mathsf{UIPFE}}_{\boldsymbol{u}^{(0)}})$ is replaced by $\mathsf{ct}^{\mathsf{UIPFE}}_{\boldsymbol{u}^{(1)}} \leftarrow \mathsf{UIPFE.Encrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, \boldsymbol{u}^{(1)})$. Thus, the challenger ciphertext becomes $\mathsf{ct}^{(0,1)} = (\mathsf{ct}^{\mathsf{UIPFE}}_{m^{(0)}\boldsymbol{u}^{(0)}}, \mathsf{ct}^{\mathsf{UIPFE}}_{\boldsymbol{u}^{(1)}})$. For the AdpIND security game of UIPFE, consider $\mathcal{B}$ an admissible adversary. Thus we have $\langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = \langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle$, for j-th secret key query on the predicate vector $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$ with non-empty index set $I_{\boldsymbol{v}^{(j)}}$ which holds from the admissible condition of AdpWAH security of UNIPE. As a result, the advantage of $\mathcal{A}$ in distinguishing between Game 0 and Game 1 is the same as the advantage in distinguishing between the experiments $\mathsf{Exp}^{\mathsf{UIPFE}}_{\mathcal{B}_1,\mathsf{AdpIND}}(1^\lambda)$ for $b = 0$ and $\mathsf{Exp}^{\mathsf{UIPFE}}_{\mathcal{B}_1,\mathsf{AdpIND}}(1^\lambda)$ for $b = 1$. Thus, we have $|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_1]| \leq \mathsf{Adv}^{\mathsf{UIPFE}}_{\mathcal{B}_1,\mathsf{AdpIND}}(\lambda)$.

Game 2: This game is identical to Game 1 except that the first component of the challenge ciphertext is switched by $\mathsf{ct}_{m^{(1)}\boldsymbol{u}^{(1)},\mathsf{UIPFE}} \leftarrow \mathsf{UIPFE.Encrypt}(\mathsf{mpk}^{\mathsf{UIPFE}}, m^{(1)}\boldsymbol{u}^{(1)})$. Thus, the challenger ciphertext becomes $\mathsf{ct}^{(1,1)} = (\mathsf{ct}^{\mathsf{UIPFE}}_{m^{(1)}\boldsymbol{u}^{(1)}}, \mathsf{ct}^{\mathsf{UIPFE}}_{\boldsymbol{u}^{(1)}})$. As previously mentioned, we consider $\mathcal{B}_1$ the admissible adversary for the AdpIND security of UIPFE. From the admissible condition of AdpWAH security of UNIPE, it holds that $\langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = \langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle = 0$ if $m^{(0)} \neq m^{(1)}$ and $\langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = \langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle \neq 0$ whenever $m^{(0)} = m^{(1)}$ for j-th secret key query on the predicate vector $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i \in I_{\boldsymbol{v}^{(j)}}}$ with index set $I_{\boldsymbol{v}^{(j)}}$. If $m^{(0)} \neq m^{(1)}$,

$$\langle m^{(0)}\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = m^{(0)} \langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = 0 = m^{(1)} \langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle = \langle m^{(1)}\boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle$$

and for $m^{(0)} = m^{(1)}$,

$$\langle m^{(0)}\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = m^{(0)} \langle \boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)} \rangle = m^{(1)} \langle \boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle = \langle m^{(1)}\boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)} \rangle$$

[1]Here we note that $\mathcal{B}$ makes two challenge queries to the UIPFE-challenger $\mathcal{C}$. This is not an issue since in indistinguishability based model, as a scheme which is secure on single challenge query is also secure on multiple challenge queries as long as the game restriction does not change with respect to the multiple challenge messages.

Therefore, $\langle m^{(0)}\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(j)}\rangle = \langle m^{(1)}\boldsymbol{u}^{(1)}, \boldsymbol{v}^{(j)}\rangle$ for j-th secret key query on predicate vector $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i\in I_{\boldsymbol{v}^{(j)}}}$ with non-empty index set $I_{\boldsymbol{v}^{(j)}}$. Thus, the advantage in distinguishing between the experiments $\mathsf{Exp}^{\mathsf{UIPFE}}_{\mathcal{B}_1,\mathsf{AdpIND}}(1^\lambda)$ for $b=0$ and $\mathsf{Exp}^{\mathsf{UIPFE}}_{\mathcal{B}_1,\mathsf{AdpIND}}(1^\lambda)$ for $b=1$ is identical to the advantage of $\mathcal{A}$ in distinguishing Game 2 and Game 3. Hence, we have $|\Pr[\mathsf{E}_1]-\Pr[\mathsf{E}_2]| \leq \mathsf{Adv}^{\mathsf{UIPFE}}_{\mathcal{B}_1,\mathsf{AdpIND}}(\lambda)$.

It is apparent that Game 2 and the initial AdpWAH security experiment with the challenge bit $b=1$ are identical. It is easy to show that the advantage of AdpWAH security, according to Definition 6, is equal to the difference in probabilities of hybrid games where the challenge bits are $b=0$ and $b=1$, respectively. Therefore, we have

$$\mathsf{Adv}^{\mathsf{UNIPE}}_{\mathcal{A},\mathsf{AdpWAH}}(\lambda) \leq |\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_2]| \leq 2\cdot \mathsf{Adv}^{\mathsf{UIPFE}}_{\mathcal{B}_1,\mathsf{AdpIND}}(\lambda)$$

which is $\mathsf{negl}(\lambda)$ according our assumption. This completes the proof. □

4.2. Concrete instantiation of UNIPE *based on SXDH*

Let us instantiate the framework with the SXDH-based UNIPE is defined in Section 4. Tomida and Takashima [16] proposed a novel technique to provide an AdpIND secure UIPFE based on the SXDH assumption over an asymmetric bilinear pairing group of prime order p using the framework of DPVS. Their scheme outputs inner products over $\mathbb{Z}_p$, and the components of the key or message vector are in $\mathbb{Z}_p$. A discrete logarithm of $[\![\langle \boldsymbol{u}, \boldsymbol{v}\rangle]\!]_T$ over a group G_T is computed at the final step of the decryption phase. We adopt a modified decryption of this UIPFE [16] that outputs $[\![\langle \boldsymbol{u}, \boldsymbol{v}\rangle]\!]_T$ instead of this discrete logarithm of $[\![\langle \boldsymbol{u}, \boldsymbol{v}\rangle]\!]_T$. Accordingly, the decryption of our SXDH-based NIPE differs from that of the generic construction. Using this AdpIND secure UIPFE scheme, we achieve an AdpWAH based on the SXDH assumption via our generic transforming with the domain $\mathcal{U}_\lambda = \mathbb{Z}_p^{I_1}, \mathcal{V}_\lambda = \mathbb{Z}_p^{I_2}, \mathcal{M}_\lambda \subseteq \mathbb{Z}_p$ where I_1, I_2 are non-empty index sets. We consider the message space $\mathcal{M}_\lambda$ to be polynomially bounded in λ. We now describe our concrete $\mathsf{UNIPE} = (\mathsf{Setup}, \mathsf{KeyExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ instantiation as described below.

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathbf{Setup}(1^\lambda)$: On input the security parameter λ, a trusted authority executes the following steps:

– Generates a bilinear group $\mathbb{BG} \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ where $\mathbb{BG} = (p, \{G_i\}_{i=1,2,T}, g_1, g_2, e)$ where the order of each multiplicative group G_1, G_2, G_T is prime p, the elements g_1, g_2 are the generators of the groups G_1, G_2 respectively and $e : G_1 \times G_2 \to G_T$ is a bilinear map.

– Chooses a uniformly random matrix $\mathbf{W} \xleftarrow{\$} \mathsf{GL}_7(\mathbb{Z}_p)$.

– Generates $\mathsf{params}_V = (p, V, V^*, A_1, A_2, G_T, E) \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(7, \mathbb{BG})$ as defined in Definition 3.
– Sets $\mathsf{pp} = (p, \{g_i\}_{i=1,2,T}, V, V^*, E)$ where $g_T = e(g_1, g_2)$ generates G_T.
– Outputs the master public key mpk and the master secret key msk as

$$\mathsf{mpk} = (\mathsf{pp}, [\![\boldsymbol{w}_1]\!]_1, [\![\boldsymbol{w}_2]\!]_1, [\![\boldsymbol{w}_3]\!]_1, [\![\boldsymbol{w}_4]\!]_1)\,;\ \mathsf{msk} = (\boldsymbol{w}_1^*, \boldsymbol{w}_2^*, \boldsymbol{w}_3^*, \boldsymbol{w}_4^*)$$

where $(\boldsymbol{w}_i, \boldsymbol{w}_i^*)$ are the i-th row of the matrix $(\mathbf{W}, \mathbf{W}^*)$.

$\underline{\mathsf{sk}_{\boldsymbol{v}} \leftarrow \mathbf{KeyExtract}(\mathsf{mpk}, \mathsf{msk}, \boldsymbol{v})}$: On input msk, mpk and a key vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}} \in \mathbb{Z}_p^{|I_{\boldsymbol{v}}|}$ such that $\phi \neq I_{\boldsymbol{v}} \subseteq [s_1]$ (where $s_1 = s_1(\lambda)$ is an integer), the trusted authority proceeds as follows.
– Computes

$$\boldsymbol{k}_i = (\Upsilon_i(-i, 1), v_i, r_i, 0, 0, 0)\mathbf{W}^*$$

where $\Upsilon_i, r_i \xleftarrow{\$} \mathbb{Z}_p$ satisfying $\sum_{i \in I_{\boldsymbol{v}}} r_i = 0$.
– Outputs the secret key $\mathsf{sk}_{\boldsymbol{v}} = (\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}}, \{[\![\boldsymbol{k}_i]\!]_2\}_{i \in I_{\boldsymbol{v}}})$.

$\underline{\mathsf{ct}_{\boldsymbol{u}} \leftarrow \mathbf{Encrypt}(\mathsf{mpk}, \boldsymbol{u}, m)}$: On input mpk, an attribute $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}} \in \mathbb{Z}_p^{|I_{\boldsymbol{u}}|}$ such that $\phi \neq I_{\boldsymbol{u}} \subseteq [s_2]$ (where $s_2 = s_2(\lambda)$ is an integer) and a message $m \in \mathcal{M}_\lambda \subseteq \mathbb{Z}_p$, the encryptor works as follows:
– Computes

$$[\![\mathsf{ct}_{1,i}]\!]_1 = [\![(\Psi_i(1, i), mu_i, z_1, 0, 0, 0)\mathbf{W}]\!]_1;\ [\![\mathsf{ct}_{2,i}]\!]_1 = [\![(\widetilde{\Psi}_i(1, i), u_i, z_2, 0, 0, 0)\mathbf{W}]\!]_1$$

where $z_1, z_2 \xleftarrow{\$} \mathbb{Z}_p$ and $\Psi_i, \widetilde{\Psi}_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in I_{\boldsymbol{u}}$.
– Outputs the ciphertext $\mathsf{ct}_{\boldsymbol{u}} = (\{[\![\mathsf{ct}_{1,i}]\!]_1\}_{i \in I_{\boldsymbol{u}}}, \{[\![\mathsf{ct}_{2,i}]\!]_1\}_{i \in I_{\boldsymbol{u}}})$.

$\underline{(\alpha \text{ or } \perp) \leftarrow \mathbf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{v}}, \mathsf{ct}_{\boldsymbol{u}})}$: The decryptor on input mpk, $\mathsf{sk}_{\boldsymbol{v}}$ associated with the vector $\boldsymbol{v} = (v_i)_{i \in I_{\boldsymbol{v}}}$, the ciphertext $\mathsf{ct}_{\boldsymbol{u}}$ corresponding to an attribute vector $\boldsymbol{u} = (u_i)_{i \in I_{\boldsymbol{u}}}$ and message m, performs the following step:
– If $I_{\boldsymbol{v}} \nsubseteq I_{\boldsymbol{u}}$, outputs $\perp$ otherwise, it computes

$$H = \prod_{i \in I_{\boldsymbol{v}}} E([\![\mathsf{ct}_{1,i}]\!]_1, [\![\boldsymbol{k}_i]\!]_2), \quad h = \prod_{i \in I_{\boldsymbol{v}}} E([\![\mathsf{ct}_{2,i}]\!]_1, [\![\boldsymbol{k}_i]\!]_2)$$

– Exhaustively searches for α over the message space in polynomial time such that $h^\alpha = H$ holds. If such α found, outputs α otherwise, outputs $\perp$.

Correctness. For $I_{\boldsymbol{v}} \subseteq I_{\boldsymbol{u}}$, computes

$$H = \prod_{i \in I_{\boldsymbol{v}}} E([\![\mathsf{ct}_{1,i}]\!]_1, [\![\boldsymbol{k}_i]\!]_2) = \prod_{i \in I_{\boldsymbol{v}}} g_T^{\langle (\Psi_i(1,i), mu_i, z_1, 0,0,0), (\Upsilon_i(-i,1), v_i, r_i, 0,0,0)\rangle} = g_T^{m\langle \boldsymbol{u}, \boldsymbol{v}\rangle},$$

$$h = \prod_{i \in I_{\boldsymbol{v}}} E([\![\mathsf{ct}_{2,i}]\!]_1, [\![\boldsymbol{k}_i]\!]_2) = \prod_{i \in I_{\boldsymbol{v}}} g_T^{\langle (\widetilde{\Psi}_i(1,i), u_i, z_2, 0,0,0), (\Upsilon_i(-i,1), v_i, r_i, 0,0,0)\rangle} = g_T^{\langle \boldsymbol{u}, \boldsymbol{v}\rangle}$$

Now, it searches exhaustively for m over the message space $\mathcal{M}_\lambda$ such that $h^m = H$. The correctness follows if and only if $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \neq 0$.

Theorem 4.2. *Under the SXDH assumption, our UNIPE scheme is adaptively weak attribute-hiding (AdpWAH) secure as per Definition 6.*

Assuming the adaptive security of [16, Theorem 4.1], the proof of the above theorem follows similarly to Theorem 4.1 of our UNIPE scheme.

5. UAnon-IBRV from UNIPE

In this section, we provide a generic transformation to construct adaptive anonymous (AdpAnon) secure unbounded anonymous identity-based revocation UAnon-IBRV = (Setup, KeyExtract, Encrypt, Decrypt) with the message space $\mathcal{M}_\lambda$ and identity space $\mathcal{ID}$. The identity-based revocation (IBRV) system encrypts data for a revoked user set $\mathcal{R}$, and a legitimate secret-key $\mathsf{sk}_{\mathsf{id}}$ corresponding to an identity id can decrypt the message if and only if $\mathsf{id} \notin \mathcal{R}$. Attrapadung et al.[22] constructed an IBRV scheme from NIPE with constant-size ciphertext. We use an attribute-hiding UNIPE and present an unbounded IBRV where the setup phase does not specify the size of the revoked user set featuring a more robust anonymous security notion of unbounded IBRV using attribute-hiding UNIPE as the building block.

Let us consider a UNIPE = (Setup, KeyExtract, Encrypt, Decrypt) where each attribute vector and predicate vector components belong to $\mathbb{Z}_p$ and message space $\mathcal{M}_\lambda \subseteq \mathbb{Z}_p$. We describe below our UAnon-IBRV = (Setup, KeyExtract, Encrypt, Decrypt) scheme for $\mathcal{ID} = \mathbb{Z}_p$ with the same message space $\mathcal{M}_\lambda$.

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \textbf{Setup}(1^\lambda)$: On input of a security parameter λ, the trusted authority computes $(\mathsf{mpk}^{\mathsf{UNIPE}}, \mathsf{msk}^{\mathsf{UNIPE}}) \leftarrow \mathsf{UNIPE.Setup}(1^\lambda)$ and outputs a master public key $\mathsf{mpk} = \mathsf{mpk}^{\mathsf{UNIPE}}$ and master secret key $\mathsf{msk} = \mathsf{msk}^{\mathsf{UNIPE}}$. The trusted authority keeps msk private while mpk is made public.

$\mathsf{sk}_{\mathsf{id}} \leftarrow \textbf{KeyExtract}(\mathsf{mpk}, \mathsf{msk}, s, \mathsf{id})$: On input mpk, msk, an identity $\mathsf{id} \in \mathbb{Z}_p$ and size s of a revoked user set, the trusted authority sets $\boldsymbol{z}_{\mathsf{id}} = (1, \mathsf{id}, \mathsf{id}^2, \ldots, \mathsf{id}^s) \in \mathbb{Z}_p^{s+1}$ and computes the secret key $\mathsf{sk}_{\boldsymbol{z}_{\mathsf{id}}}^{\mathsf{UNIPE}} \leftarrow \mathsf{UNIPE.KeyExtract}(\mathsf{mpk}^{\mathsf{UNIPE}}, \mathsf{msk}^{\mathsf{UNIPE}}, \boldsymbol{z}_{\mathsf{id}})$. It outputs the secret key $\mathsf{sk}_{\mathsf{id}} = \mathsf{sk}_{\boldsymbol{z}_{\mathsf{id}}}^{\mathsf{UNIPE}}$ corresponding to the identity id.

$\mathsf{ct}_m \leftarrow \textbf{Encrypt}(\mathsf{mpk}, \mathcal{R}, m)$: On input mpk, a set of revoked users identities $\mathcal{R} = \{\mathsf{id}_1, \mathsf{id}_2, \ldots, \mathsf{id}_s\}$ and a message $m \in \mathcal{M}_\lambda$, the encryptor computes a polynomial $P(T) = \prod_{i=0}^{s}(t - \mathsf{id}_i) = \sum_{i=0}^{s} t_i T^i \in \mathbb{Z}_p[T]$ and sets $\boldsymbol{t}_{\mathcal{R}} = (t_0, t_1, \ldots, t_s) \in \mathbb{Z}_p^{s+1}$. It computes $\mathsf{ct}_{\boldsymbol{t}_{\mathcal{R}}}^{\mathsf{UNIPE}} \leftarrow \mathsf{UNIPE.Encrypt}(\mathsf{mpk}^{\mathsf{UNIPE}}, \boldsymbol{t}_{\mathcal{R}}, m)$

and outputs the ciphertext $\mathsf{ct}_m = \mathsf{ct}^{\mathsf{UNIPE}}_{\boldsymbol{t}_{\mathcal{R}}}$ corresponding to the message m.
$(M \text{ or } \perp) \leftarrow \textbf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_m)$: The decryptor takes input $\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_m$ and outputs $m \leftarrow \mathsf{UNIPE.Decrypt}(\mathsf{mpk}^{\mathsf{UNIPE}}, \mathsf{sk}^{\mathsf{UNIPE}}_{\boldsymbol{z}_{\mathsf{id}}}, \mathsf{ct}^{\mathsf{UNIPE}}_{\boldsymbol{w}_{\mathcal{R}}})$ if $\mathsf{id} \notin \mathcal{R}$.

Observe that $\langle \boldsymbol{t}_{\mathcal{R}}, \boldsymbol{z}_{\mathsf{id}} \rangle = P(\mathsf{id}) = 0$ when $\mathsf{id} \in \mathcal{R}$ and the revoked user with identities id cannot recover the message m as the underlying UNIPE scheme fails to decrypt when $\langle \boldsymbol{t}_{\mathcal{R}}, \boldsymbol{z}_{\mathsf{id}} \rangle = 0$. On the other hand, as $\langle \boldsymbol{t}_{\mathcal{R}}, \boldsymbol{z}_{\mathsf{id}} \rangle \neq 0$ for $\mathsf{id} \notin \mathcal{R}$, the non-revoked user in the identity id will be able to recover the message m as the underlying NIPE can recover m when $\langle \boldsymbol{t}_{\mathcal{R}}, \boldsymbol{z}_{\mathsf{id}} \rangle \neq 0$. Therefore, the correctness of the above UAnon-IBRV directly follows from the correctness of the underlying UNIPE, i.e., $\langle \boldsymbol{t}_{\mathcal{R}}, \boldsymbol{z}_{\mathsf{id}} \rangle \neq 0$ implies $\mathsf{id} \notin \mathcal{R}$. For the security of UAnon-IBRV, we assume that the adversary $\mathcal{A}$ submits a challenge tuple $(m^{(0)}, \mathcal{R}^{(0)})$ and $(m^{(1)}, \mathcal{R}^{(1)})$ such that $\mathsf{id} \in \mathcal{R}^{(0)} \cap \mathcal{R}^{(1)}$ for all $\widehat{s}, \{\mathsf{id}\}$ queried by $\mathcal{A}$ to the key extraction oracle. Therefore, it satisfies $\langle \boldsymbol{t}_{\mathcal{R}^{(0)}}, \boldsymbol{z}_{\mathsf{id}} \rangle = \langle \boldsymbol{t}_{\mathcal{R}^{(1)}}, \boldsymbol{z}_{\mathsf{id}} \rangle = 0$ for all id queried by $\mathcal{A}$ to the key extraction oracle. Therefore, AdpWAH security of UNIPE ensures that the challenge ciphertext $\mathsf{ct}^{\mathsf{UNIPE}}_{\boldsymbol{w}_{\mathcal{R}^{(b)}}} \leftarrow \mathsf{UNIPE.Encrypt}(\mathsf{mpk}^{\mathsf{UNIPE}}, \boldsymbol{t}_{\mathcal{R}^{(b)}}, m^{(b)})$ hides b from $\mathcal{A}$'s view. The following theorem proves the security of the UAnon-IBRV scheme, and the proof follows from AdpWAH security of the underlying UNIPE scheme.

Theorem 5.1. *Assuming the underlying UNIPE scheme is adaptively weak attribute-hiding (AdpWAH) secure as per Definition 6, our generic construction of UAnon-IBRV scheme achieves adaptive anonymous (AdpAnon) security as per Definition 10.*

6. Comparison & Analysis

In Table 1, we compare the existing NIPEs with our proposed SXDH-based payload-hiding UNIPE-I and attribute-hiding UNIPE-II. Table 1 demonstrates that our UNIPE exhibits unbounded property, and the master secret and public keys are more efficient in terms of sizes than the existing NIPE schemes. In contrast to the existing NIPEs, the public parameter sizes in our UNIPEs are constant, which is more desirable from a practical point of view. Each of our UNIPEs requires 39 and 28 group elements in the master public key and the master secret key, respectively. The ciphertext size in our UNIPE-I and UNIPE-II are $6n_1 + 6$ and $14n_1$ group elements, respectively, whereas the secret keys contain $7n_2$ and $6n_2 + 5$ group elements, respectively.

Table 1: Comparison between existing NIPE schemes

Scheme	$\lvert\mathsf{mpk}\rvert$	$\lvert\mathsf{msk}\rvert$	$\lvert\mathsf{sk}\rvert$	$\lvert\mathsf{ct}\rvert$	Data size	Security model
NIPE-I of [15]	$(8n_1+8)\lvert G\rvert$	$\mathcal{O}(1)\lvert\mathbb{Z}_p\rvert$	$(4n_2{+}5)\lvert G\rvert$	$13\lvert G\rvert{+}\lvert G_T\rvert$	bund	AdpPH
NIPE-II of [15]	$(8n_1+8)\lvert G\rvert$	$\mathcal{O}(n_1)\lvert G\rvert$	$13\lvert G\rvert$	$(4n_1{+}5)\lvert G\rvert$ $+\lvert G_T\rvert$	bund	AdpPH
[24]	$(2n_1{+}2)\lvert G_1\rvert$	$4n_1\lvert\mathbb{Z}_p\rvert$	$(n_2{+}4)\lvert G_2\rvert$	$(2n_1+4)\lvert G_1\rvert$	bund	AdpWAH
[36]	$(2n_1+6)\lvert G_1\rvert+6\lvert G_2\rvert$	$(4n_1+8)\lvert\mathbb{Z}_p\rvert$	$4\lvert\mathbb{Z}_p\rvert$	$(2n_1+4)\lvert G_1\rvert$	bund	AdpWAH
Our UNIPE-I	$39\lvert G_1\rvert$	$39\lvert\mathbb{Z}_p\rvert$	$(6n_2{+}5)\lvert G_2\rvert$	$(6n_1{+}5)\lvert G_1\rvert$ $+\lvert G_T\rvert$	unbund	AdpPH
Our UNIPE-II	$28\lvert G_1\rvert$	$28\lvert\mathbb{Z}_p\rvert$	$7n_2\lvert G_2\rvert$	$14n_1\lvert G_1\rvert$	unbund	AdpWAH

$\lvert\mathsf{mpk}\rvert, \lvert\mathsf{msk}\rvert, \lvert\mathsf{sk}\rvert, \lvert\mathsf{ct}\rvert$: sizes of the master public key, master secret key, secret key and ciphertext respectively; bund: bounded; unbund: unbounded; n_1, n_2 : sizes of attribute vector and predicate vector respectively; for exiting bounded NIPE, $n_1 = n_2$ holds; AdpWAH : adaptively weakly attribute-hiding; AdpPH: adaptively payload-hiding; $|G|$, $|G_\iota|$: size of elements of groups G and G_ι for $\iota \in \{1, 2, T\}$ respectively. Consider symmetric pairing $G \times G$ to G_T and asymmetric pairing $G_1 \times G_2$ to G_T.

Table 2: Comparison between existing NIPE parameters in terms of kilo-bits

Scheme	att. vec. length	pred vec. length	128-bit security			256-bit security		
	n_1	n_2	mpk	ct	sk	mpk	ct	sk
NIPE-I	100	100	404	9.5	202.5	2020	47.5	1012.5
of [15]	200	200	804	9.5	402.5	4020	47.5	2012.5
NIPE-II	100	100	404	205.5	6.5	2020	1027.5	32.5
of [15]	200	200	804	405.5	6.5	4020	2027.5	32.5
[24]	100	100	50.5	51	52	126.25	127.5	260
	200	200	100.5	101	102	251.25	252.5	510
[36]	100	100	54.5	51	1	143.75	127.5	2.5
	200	200	104.5	502.5	1	268.75	252.5	2.5
Our	100	100	9.75	154.25	302.5	24.375	393.125	1512.5
UNIPE-I	200	200	9.75	304.25	602.5	24.375	768.125	3012.5
Our	100	100	7	350	350	70	875	1750
UNIPE-II	200	200	7	700	700	70	1750	3500

att vec.: attribute vector; pred vec.: predicate vector; Group sizes of asymmetric pairing follows from 2007 NIST recommendations of [51]. Descriptions of an elliptic curves are in [52]. We consider a 256-bit Barreto-Naehrig curve [53] with embedding degree 12 for 128 bit security and a 640-bit Brezing-Weng curve [54] with embedding degree 24 for 256-bit security.

Table 2 compares the existing NIPEs with our two UNIPE instantiations (i.e., payload-hiding UNIPE of section 3 and weak-attribute-hiding SXDH-

based UNIPE of section 4.2) concerning 128-bit and 256-bit security levels using the group sizes described in [50]. It demonstrates that proposed UNIPEs contain very short sizes of public keys than other existing NIPEs. Also, the secret key and ciphertext sizes are well comparable with other bounded NIPEs. However, our schemes are designed in an unbounded setting applicable in more practical situations that bounded NIPEs cannot address.

7. Conclusion

We have presented the *first* UNIPE scheme that achieves adaptively payload-hiding security based on the SXDH assumption in the standard model, featuring constant-size master key pairs. Further, we have developed a generic construction of the dual of unbounded zero inner product encryption (UZIPE), namely UNIPE, utilizing UIPFE as the fundamental building block. The proposed UNIPE exhibits weak attribute-hiding security, which relies on the adaptive security of underlying UIPFE. More positively, we have provided a precise instantiation of our UNIPE based on the standard SXDH assumption. Additionally, we have designed a generic construction of UAono-IBRV from UNIPE. Realizing an adaptive, fully attribute-hiding UNIPE from standard assumption is still an open problem.

CRediT Authorship Contribution Statement. Subhranil Dutta: Conceptualization, Methodology, Formal Analysis, Writing–original draft, Writing–review & editing. Tapas Pal: Conceptualization, Methodology, Writing–review & editing. Ratna Dutta: Conceptualization, Methodology, Formal analysis, Writing–review & editing.

Declaration of Completing Interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability. No data was used for the research described in the article.

Acknowledgements. The first author was partially supported by the Council of Scientific & Industrial Research (CSIR) fellowship of the Government of India. All authors approved the version of the manuscript to be submitted.

References

[1] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters, Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption, in: H. Gilbert (Ed.), Advances in Cryptology – EUROCRYPT 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 62–91.

[2] T. Okamoto, K. Takashima, Fully secure functional encryption with general relations from the decisional linear assumption, in: T. Rabin (Ed.), Advances in Cryptology – CRYPTO 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 191–208.

[3] D. Boneh, B. Waters, Conjunctive, subset, and range queries on encrypted data, in: Theory of cryptography conference, Springer, 2007, pp. 535–554.

[4] D. Boneh, X. Boyen, Secure identity based encryption without random oracles, in: Annual International Cryptology Conference, Springer, 2004, pp. 443–459.

[5] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: Annual international conference on the theory and applications of cryptographic techniques, Springer, 2005, pp. 457–473.

[6] A. Lewko, B. Waters, Unbounded hibe and attribute-based encryption, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2011, pp. 547–567.

[7] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: 2007 IEEE symposium on security and privacy (SP'07), IEEE, 2007, pp. 321–334.

[8] T. Okamoto, K. Takashima, Fully secure unbounded inner-product and attribute-based encryption, in: X. Wang, K. Sako (Eds.), Advances in Cryptology – ASIACRYPT 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 349–366.

[9] R. Ostrovsky, A. Sahai, B. Waters, Attribute-based encryption with non-monotonic access structures, in: Proceedings of the 14th ACM conference on Computer and communications security, 2007, pp. 195–203.

[10] J. Katz, A. Sahai, B. Waters, Predicate encryption supporting disjunctions, polynomial equations, and inner products, in: N. Smart (Ed.), Advances in Cryptology – EUROCRYPT 2008, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 146–162.

[11] T. Okamoto, K. Takashima, Hierarchical predicate encryption for inner-products, in: M. Matsui (Ed.), Advances in Cryptology – ASIACRYPT 2009, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 214–231.

[12] N. Attrapadung, B. Libert, E. d. Panafieu, Expressive key-policy attribute-based encryption with constant-size ciphertexts, in: D. Catalano, N. Fazio, R. Gennaro, A. Nicolosi (Eds.), Public Key Cryptography – PKC 2011, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 90–108.

[13] S. Dutta, T. Pal, R. Dutta, Fully secure unbounded zero inner product encryption with short ciphertexts and keys, in: Q. Huang, Y. Yu (Eds.), Provable and Practical Security, Springer International Publishing, Cham, 2021, pp. 241–258.

[14] T. OKAMOTO, K. TAKASHIMA, Dual pairing vector spaces and their applications, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E98.A (1) (2015) 3–15. doi:10.1587/transfun.E98.A.3.

[15] T. Okamoto, K. Takashima, Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption, Designs, Codes and Cryptography 77 (2) (2015) 725–771.

[16] J. Tomida, K. Takashima, Unbounded inner product functional encryption from bilinear maps, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2018, pp. 609–639.

[17] E. Dufour-Sans, D. Pointcheval, Unbounded inner-product functional encryption with succinct keys, in: International Conference on Applied Cryptography and Network Security, Springer, 2019, pp. 426–441.

[18] J. H. Park, Inner-product encryption under standard assumptions, Designs, Codes and Cryptography 58 (3) (2011) 235–257.

[19] T. Okamoto, K. Takashima, Adaptively attribute-hiding (hierarchical) inner product encryption, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2012, pp. 591–608.

[20] T. Zhenlin, Z. Wei, A predicate encryption scheme supporting multi-party cloud computation, in: 2015 international conference on intelligent networking and collaborative systems, IEEE, 2015, pp. 252–256.

[21] Y. Kawai, K. Takashima, Predicate-and attribute-hiding inner product encryption in a public key setting, in: International Conference on Pairing-Based Cryptography, Springer, 2013, pp. 113–130.

[22] N. Attrapadung, B. Libert, Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation, in: P. Q. Nguyen, D. Pointcheval (Eds.), Public Key Cryptography – PKC 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 384–402.

[23] J. Chen, H. Wee, Doubly spatial encryption from dbdh, Theoretical Computer Science 543 (2014) 79–89.

[24] S. Patranabis, D. Mukhopadhyay, S. C. Ramanna, Function private predicate encryption for low min-entropy predicates, in: IACR International Workshop on Public Key Cryptography, Springer, 2019, pp. 189–219.

[25] S. Yamada, N. Attrapadung, G. Hanaoka, N. Kunihiro, A framework and compact constructions for non-monotonic attribute-based encryption, in: International Workshop on Public Key Cryptography, Springer, 2014, pp. 275–292.

[26] S. Katsumata, S. Yamada, Non-zero inner product encryption schemes from various assumptions: Lwe, ddh and dcr, in: IACR International Workshop on Public Key Cryptography, Springer, 2019, pp. 158–188.

[27] S. Agrawal, B. Libert, D. Stehlé, Fully secure functional encryption for inner products, from standard assumptions, in: Annual International Cryptology Conference, Springer, 2016, pp. 333–362.

[28] M. Abdalla, F. Bourse, A. D. Caro, D. Pointcheval, Simple functional encryption schemes for inner products, in: IACR International Workshop on Public Key Cryptography, Springer, 2015, pp. 733–751.

[29] P. Datta, R. Dutta, S. Mukhopadhyay, Strongly full-hiding inner product encryption, Theoretical Computer Science 667 (2017) 16–50.

[30] J. Tomida, M. Abe, T. Okamoto, Efficient functional encryption for inner-product values with full-hiding security, in: International Conference on Information Security, Springer, 2016, pp. 408–425.

[31] A. Bishop, A. Jain, L. Kowalczyk, Function-hiding inner product encryption, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2015, pp. 470–491.

[32] D. Naor, M. Naor, J. Lotspiech, Revocation and tracing schemes for stateless receivers, in: Advances in Cryptology—CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23, 2001 Proceedings 21, Springer, 2001, pp. 41–62.

[33] D. Boneh, C. Gentry, B. Waters, Collusion resistant broadcast encryption with short ciphertexts and private keys, in: Annual international cryptology conference, Springer, 2005, pp. 258–275.

[34] C. Delerablée, P. Paillier, D. Pointcheval, Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys, in: International Conference on Pairing-Based Cryptography, Springer, 2007, pp. 39–59.

[35] A. Lewko, A. Sahai, B. Waters, Revocation systems with very small private keys, in: 2010 IEEE Symposium on Security and Privacy, IEEE, 2010, pp. 273–285.

[36] T. Pal, R. Dutta, Cca secure attribute-hiding inner product encryption from minimal assumption, in: Australasian Conference on Information Security and Privacy, Springer, 2021, pp. 254–274.

[37] C. Delerablée, Identity-based broadcast encryption with constant size ciphertexts and private keys, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2007, pp. 200–215.

[38] C. Gentry, B. Waters, Adaptive security in broadcast encryption systems (with short ciphertexts), in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2009, pp. 171–188.

[39] J. Kim, W. Susilo, M. Au, J. Seberry, Adaptively secure identity-based broadcast encryption with a constant-sized ciphertext, IEEE Transactions on Information Forensics and Security 10 (3) (2015) 679–693.

[40] L. Zhang, Q. Wu, Y. Mu, Anonymous identity-based broadcast encryption with adaptive security, in: International Symposium on Cyberspace Safety and Security, Springer, 2013, pp. 258–271.

[41] A. Barth, D. Boneh, B. Waters, Privacy in encrypted content distribution using private broadcast encryption, in: International conference on financial cryptography and data security, Springer, 2006, pp. 52–64.

[42] W. Liu, J. Liu, Q. Wu, B. Qin, Hierarchical identity-based broadcast encryption, in: Information Security and Privacy: 19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7-9, 2014. Proceedings 19, Springer, 2014, pp. 242–257.

[43] A. Boldyreva, V. Goyal, V. Kumar, Identity-based encryption with efficient revocation, in: Proceedings of the 15th ACM conference on Computer and communications security, 2008, pp. 417–426.

[44] J. Li, W. Yao, Y. Zhang, H. Qian, J. Han, Flexible and fine-grained attribute-based data storage in cloud computing, IEEE Transactions on Services Computing 10 (5) (2016) 785–796.

[45] R. Zhang, J. Li, Y. Lu, J. Han, Y. Zhang, Key escrow-free attribute based encryption with user revocation, Information Sciences 600 (2022) 59–72.

[46] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, D. Wang, Attribute based encryption with privacy protection and accountability for cloudiot, IEEE Transactions on Cloud Computing 10 (2) (2020) 762–773.

[47] N. Chen, J. Li, Y. Zhang, Y. Guo, Efficient cp-abe scheme with shared decryption in cloud storage, IEEE Transactions on Computers 71 (1) (2020) 175–184.

[48] S. Chen, J. Li, Y. Zhang, J. Han, Efficient revocable attribute-based encryption with verifiable data integrity, IEEE Internet of Things Journal.

[49] B. Waters, Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions, in: Annual International Cryptology Conference, Springer, 2009, pp. 619–636.

[50] J. Chen, H. W. Lim, S. Ling, H. Wang, H. Wee, Shorter ibe and signatures via asymmetric pairings, in: Pairing-Based Cryptography–Pairing 2012: 5th International Conference, Cologne, Germany, May 16-18, 2012, Revised Selected Papers 5, Springer, 2013, pp. 122–140.

[51] E. Barker, W. Barker, Recommendation for key management, part 2: best practices for key management organization, Tech. rep., National Institute of Standards and Technology (2018).

[52] D. Freeman, M. Scott, E. Teske, A taxonomy of pairing-friendly elliptic curves, Journal of cryptology 23 (2010) 224–280.

[53] P. S. Barreto, M. Naehrig, Pairing-friendly elliptic curves of prime order, in: Selected Areas in Cryptography: 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers 12, Springer, 2006, pp. 319–331.

[54] F. Brezing, A. Weng, Elliptic curves suitable for pairing based cryptography, Designs, Codes and Cryptography 37 (1) (2005) 133–141.