Candidate Matchmaking Encryption from Attribute-Based Encryption Schemes

Zhuang Shan[1][0000−0003−4309−4505], Leyou Zhang[1,*][0000−0003−4950−1140], Fuchun Guo[2][0000−0001−6939−7710], and Yong Yu[3]

[1] School of Mathematics and Statistics, Xidian University, Xi'an 710126, China, arcsec30@163com, lyzhang@mail.xidian.edu.cn

[2] Centre for Computer and Information Security Research, University of Wollongong, Wollongong, NSW 2522, Australia, fuchun@uow.edu.au

[3] School of Computer Science, Shaanxi Normal University, Xi'an 710121, China, yuyong@snnu.edu.cn

Abstract. We were deeply impressed by the paper by Ateniese et al., published in Crypto 2019. In it, they presented a black-box construction of matchmaking encryption (ME) based on functional encryption. In our work, we propose an ME scheme based on standard assumptions in the standard model. This scheme has been proven to be secure under the learning with error (LWE) assumption. Our ME scheme is achieved through a novel framework of *bilateral-policy attribute-based encryption (BP-ABE)* and a new intermediate primitive termed a *perturbed pseudorandom generator (PPRG)*, which facilitates the implementation of authentication functionality by replacing non-interactive zero-knowledge proof functionality. In the scheme presented in this paper, the user's "public key" is generated using Hamming correlation robustness and user attributes. Note that the 'public key' is not public. In order to preserve the privacy of the two parties involved in matchmaking encryption, our BP-ABE scheme does not use the 'public key' directly to encrypt the plaintext. Instead, the message sender selects matching attributes and uses a Hamming correlation robustness and *homomorphic pseudorandom function (HPRF)* to generate temporary public keys and hide the public key and user attributes.
When these temporary public keys satisfy the access policy, the receiver can decrypt the data using their private key. Regarding the authentication function of matchmaking encryption, this paper proposes a non-interactive privacy set intersection (PSI) scheme based on HPRF and PPRG. The message sender encrypts their 'public key' using the proposed PSI scheme as part of the ciphertext. The receiver also encrypts their 'public key' using the proposed PSI scheme and matches the attributes, thereby completing the message authentication function. We consider our approach to be a significant departure from existing constructions, despite its simplicity.

Keywords: matchmaking encryption · attribute-based encryption · standard assumptions.

1 Introduction

Matchmaking encryption (ME) is a new form of encryption proposed by Ateniese, Francati, Nuñez and Venturi [2, 3] in Crypto 2019, in which both the sender and the receiver (each with its own attributes) can specify fine-grained access policies the other party must satisfy in order for the message to be revealed. Using ME, a sender with attributes $\sigma \in \{0,1\}^*$ encrypts messages after generating the policy $\mathbb{R}$ of the intended receiver, and a receiver with attributes $\rho \in \{0,1\}^*$ obtains a decryption key $dk_{\mathbb{S}}$ from an authority before decrypting the ciphertext from a sender satisfying

the specified policy $\mathbb{S}$ specified by the receiver, and at the same time the receiver's attributes ρ specified by the sender.

Roughly speaking, ME is an encryption scheme that integrates bilateral access control, message confidentiality, access policy privacy, and message authenticity (integrity) into one. It is also a non-interactive secret handshake (SH) protocol[5, 4]. Ateniese et al. provide generic frameworks for constructing ME from functional encryption, which is a black-box construction [2, 3]. Therefore, these leave the following problem:

Can we construct matchmaking encryption under the standard assumptions?

1.1 Our Contributions

We propose a new approach to construct general-purpose matchmaking encryption. Our approach is heuristic but without using any new sources of computational hardness. In other words, our constructions use well-studied cryptographic primitives in a generic way to realize ME, while still being heuristic in the sense that our constructions are not provably secure and make implicit assumptions about the interplay of the underlying primitives. The primitives we use can themselves be securely realized based on standard assumptions, namely the hardness of the learning with errors (LWE, [16, 17, 15, 8]).

Bilateral-Policy Attribute-Based Encryption The starting point of our work is the fact that ME can provably be based on *bilateral-policy attribute-based encryption(BP-ABE)*, a new primitive that we introduce in this work. A BP-ABE is an ABE scheme [6, 18] that requires the message sender to also satisfy the recipient's access control in order to send a message to the recipient. Additionally, it provides two special features beyond the original ABE:

i. **Public Key Hiding** To ensure the anonymity of the message recipient, the message sender selects a subset of attributes to form a matrix. This matrix is then encrypted using a homomorphic pseudorandom function, resulting in a temporary public key. The sender then uses this temporary public key to encrypt the message.
ii. **Message Authentication** Since the encryption process does not use the recipient's public key, message authentication is required before decryption. We have constructed a non-interactive Private Set Intersection(PSI) scheme, which replaces Ateniese et al.'s non-interactive zero-knowledge proof for completing the message authentication.

As for the security of the BP-ABE scheme, we present a general LWE-based ABE scheme and prove that the BP-ABE scheme is indistinguishable from this general LWE-based ABE scheme, thereby proving the security of the BP-ABE scheme.

A Generic Candidate Next, we show a *generic candidate* construction of BP-ABE based on three building blocks: (i) a standard ABE scheme with bilateral access control (which can be instantiated with essentially all LWE-based constructions), (ii) a homomorphic pseudorandom function is used to generate a temporary key (which can be replaced by homomorphic encryption), and (iii) private set intersection from perturbed pseudorandom generator is used to complete the verification function (such as [19]).

We expect that there will exist instantiations of the underlying primitives (though contrived) for which this construction is insecure. For example, if the underlying schemes are not circular secure to begin with, then the resulting BP-ABE would also be insecure. However, for natural instantiations of these primitives, security can be conjectured.

Evidence of security To build confidence in our construction, we provide evidence that the aforementioned conjecture regarding the interplay between security holds in an appropriate initial ABE scheme based on LWE, inspired by the LWE-based initial ABE scheme. Thus, we push all the heuristic aspects of the construction to an initial ABE scheme. In fact, we demonstrate that security can be proven in this ABE scheme.

Conceptual Simplicity Another positive feature of our construction is its conceptual simplicity, which makes it much easier to analyze and thus have condence in. Finally, we remark that our construction is a big departure from the previously-mentioned functional encryption based ME construction and will be accessible to readers without first understanding prior ME construction.

The Envisioned Application Scenarios of ME The application scenarios of ME are very broad, especially on network platforms with high privacy requirements. For example, in smart healthcare, a patient may wish to communicate with another patient who has the same symptoms; in social media, a user may want to find someone with a similar personality for online communication; and people with shared interests may wish to exchange experiences without others knowing. Both the message sender and receiver do not want either party or any authoritative institution to know their true identity, but they are certain that the other person shares the same specific attribute.

1.2 Technical Overview

In the following we give an informal overview of the techniques we develop in this work and we refer the reader to the technical sections for more precise statements.

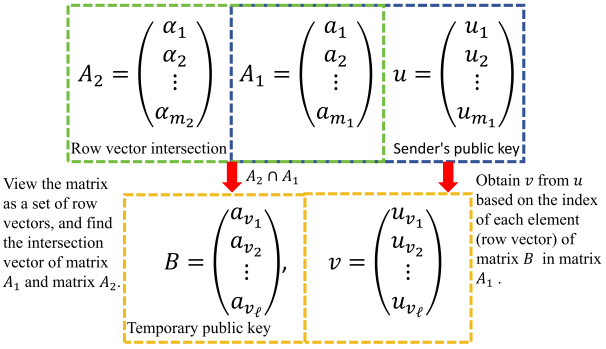
BP-ABE of First Attempt How to implement bilateral matching in an ABE scheme? We achieve this through matrix operations and the properties of homomorphic pseudo-random functions, i.e., assuming $A = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} \in \mathbb{Z}_q^{m \times n}$, $e \in \mathbb{Z}_q^n$, $u \in \mathbb{Z}_q^m$, obtain

$$\begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} \cdot e = \begin{pmatrix} a_1 e \\ \vdots \\ a_m e \end{pmatrix}.$$

Fix e, and arbitrarily choose the row vectors $a_{v_1}, \ldots, a_{v_\ell}$ from A, where $\ell \leq m$, then we have

$$\begin{pmatrix} b_{v_1} \\ \vdots \\ b_{v_\ell} \end{pmatrix} = \begin{pmatrix} a_{v_1} \\ \vdots \\ a_{v_\ell} \end{pmatrix} \cdot e = \begin{pmatrix} a_{v_1} e \\ \vdots \\ a_{v_\ell} e \end{pmatrix}.$$

This property is key to helping us implement BP-ABE. Suppose the public and private keys of the message sender are $pk = (A_1, u_1)$ and $sk = e_1$, respectively, and the public and private keys of the receiver are $pk = (A_2, u_2)$ and $sk = e_2$, respectively. Let the matrix B be formed by the row vectors that are common to both A_1 and A_2, and then apply the following operation to u_1 to obtain w. Here $A_1 \in \mathbb{Z}_q^{m_1 \times n}$, $A_2 \in \mathbb{Z}_q^{m_2 \times n}$.



Thus, based on the learning with error problem, construct a public-key encryption scheme. To encrypt the plaintext $\mathfrak{m} \in \{0, 1\}$, obtain

$$c = (c_1, c_2), c_1 = B^T s + \mathtt{x}, c_2 = v^T s + x + \mathfrak{m}\lfloor q/2 \rfloor.$$

In this case, $B \in \mathbb{Z}_q^{v_\ell \times n}$, v and $s \in \mathbb{Z}_q^{v_\ell}$, $\mathtt{x} \in \mathbb{Z}_q^n$, and $x \in \mathbb{Z}_q$. Clearly, when the sender's attributes satisfy the receiver's access control, the receiver can use their private key to decrypt and obtain the plaintext.

However, this requires interaction between the sender and receiver before the message is sent, which does not align with the definition of ME. This is because ME requires non-interactive communication.

BP-ABE of Second Attempt We no longer require the generation of B to be constructed from row vectors that are the same in both A_1 and A_2. Instead, the message sender derives the corresponding attributes based on the receiver's access control $\mathbb{R}$ and generates the matrix B and vector v accordingly. However, this approach introduces security risks, as selecting different numbers of row vectors from A_1 by the message sender causes the dimension of c_1 in the ciphertext to be variable, which poses a security threat.

BP-ABE of Third Attempt To address this issue, we want the message sender to choose exactly t row vectors from A_1 to construct the matrix B and generate the corresponding vector v each time.

With this approach, regardless of whether the receiver satisfies the sender's access policy, the receiver can always decrypt. Therefore, we need to use a homomorphic pseudorandom function to wrap the temporary key, and both the receiver and the sender use their respective attributes and access policies to generate the homomorphic pseudorandom function keys. When both parties

satisfy the access policy, the two keys will be equal, ensuring that the receiver can only decrypt when the access policy is met.

So, where do the "attributes" come into play in the above scheme? In fact, the matrix A in the above scheme is not randomly generated; it is constructed based on user (here, without distinguishing between the message sender and the receiver) attributes and hamming correlation robustness. First, define $\mathcal{H}_1$ and $\mathcal{H}_2$ as hamming correlation robustness, which essentially represents a hash function, whose output is pseudorandom like a pseudorandom function. Suppose the user's attributes are $Att = (att_1, \ldots, att_m)$, and using $\mathcal{H}_1$, we obtain

$$A = \begin{pmatrix} \mathcal{H}_1(att_1) \\ \vdots \\ \mathcal{H}_1(att_m) \end{pmatrix} \in \mathbb{Z}_q^{m \times n}, u = \begin{pmatrix} \mathcal{H}_2(att_1) \\ \vdots \\ \mathcal{H}_2(att_m) \end{pmatrix} \in \mathbb{Z}_q^m$$

As for the generation of the private key, it can simply be done using a trapdoor.

Authentication Function of BP-ABE Regarding the authentication function of BP-ABE, we use non-interactive private set intersection techniques to help implement the authentication function. More specifically, the sender utilizes the receiver's access control $\mathbb{R}$ and their own attributes to generate a homomorphic pseudorandom key $ek_{\mathbb{R}}$, which encrypts all of the sender's attributes, i.e.

$$D = \{d | d = F_{ek_{\mathbb{R}}}(att), att \in A_\sigma\}.$$

Then, the perturbed pseudorandom generator G_γ is used to further obscure D, i.e.

$$\begin{aligned} \Psi = \{\psi | \psi = G_\gamma(d), d \in D\}, \text{ here} \\ \text{If } d_1 = d_2, \|G_\gamma(d_1) - G_\gamma(d_2)\| \leq \gamma, \\ \text{Else } \|G_\gamma(d_1) - G_\gamma(d_2)\| > \gamma. \end{aligned}$$

In addition to sending the ciphertext, the sender also needs to send Ψ to the receiver. The receiver similarly uses the sender's access control $\mathbb{S}$ and their own attributes to generate a homomorphic pseudorandom key $dk_{\mathbb{S}}$, which encrypts all of the receiver's attributes, and then uses the pseudorandom generator to further encrypt them. The encrypted attributes are compared element by element with the elements in Ψ. When the number of matching elements exceeds the threshold t, the receiver outputs 1; otherwise, it outputs 0.

The private set intersection described above differs from other private intersection protocols in that it is non-interactive, while others are interactive. Since we require the BP-ABE to be non-interactive, a novel private set intersection protocol is adopted.

1.3 Related Work

In 2019, Ateniese, Francati, Nuñez and Venturi first proposed the concept of ME and gave black-box constructions for ME based on functional encryption for randomized functionalities [1], digital signatures, and non-interactive zero-knowledge proofs. In addition, Ateniese, Francati, Nuñez and Venturi also gave the concepts of identity-based matchmaking encryption and arranged matchmaking encryption.

In 2021, Francati, Guidi, Russo, and Venturi constructed an identity-based matchmaking encryption without random oracles in the plain model[12].

In 2022, Chen, Li, Wen and Weng gave identity-based matchmaking encryption from a variant of two-level anonymous IBE scheme, and the IBE scheme is based on the SXDH assumption in prime order bilinear pairing groups [10]. Xu et al. gave a new cryptographic primitive based on the concept of ME, namely matchmaking attribute-based encryption[21], which is also a bilateral access-controlled ABE. Unlike the BP-ABE in this article, MABE does not require *kpol* to participate in decryption.

In 2025, the identity-based matchmaking encryption schemes by Wang et al.[20] and Chihu et al.[11] respectively considered security models under quantum adversaries and quantum random oracle.

1.4 Organization

The structure of this paper is as follows. Section 2 will introduce some necessary basic concepts, including some basic definitions, concepts, etc.; Section 3 introduces the LWE-based ABE scheme and BP-ABE scheme proposed in this paper, among which the LWE-based ABE scheme is the basis for constructing BP-ABE, and proves that the two schemes are IND-CPA secure; Section 4 gives an example of the authentication function of the BP-ABE scheme, that is, an example of non-interactive PSI and a security proof; Section 5 gives this paper's matchmaking encryption scheme and its security proof; The last section summarizes the work and shortcomings of this paper.

2 Preliminaries

We begin by presenting some necessary definitions, including some given in this paper, for constructing matchmaking encryption. We follow with a brief review of lattice, and then state the complexity assumptions we use for our proofs of security. We conclude with the IND-CPA security model for BP-ABE and the security definition of ME.

2.1 Definitions

We state definitions below.

Definition 1 ([9, 14], Hamming Correlation Robustness(HCR)). *Let $\mathcal{H}$ be a hash function with input length n. Then H is d-Hamming correlation robust if, for any $a_1, \ldots, a_m, b_1, \ldots, b_m \in \{0,1\}^n$ with $\|b_i\|_H \geq d$ for each $i \in [m]$, the following distribution, induced by random sampling of $s \leftarrow_R \{0,1\}^n$, is pseudorandom. Namely,*

$$(\mathcal{H}(a_1 \oplus [b_1 \cdot s]), \ldots, \mathcal{H}(a_1 \oplus [b_1 \cdot s])) \approx_C (F(a_1 \oplus [b_1 \cdot s]), \ldots, F(a_1 \oplus [b_1 \cdot s])).$$

where $\cdot$ denotes bitwise-AND and $\oplus$ denotes bitwise-XOR, F is a random function.

Definition 2. *The notation $\mathbb{P}_{\mathcal{X}} = \{\Pr(x_1), \ldots, \Pr(x_n)\}$ represents the probability of the occurrence of element x_i in the set $\mathcal{X}$, where $i = 1, \ldots, n$ and $\mathcal{X} = \{x_1, \ldots, x_n\}$.*

Definition 3 (Homomorphic pseudo-random function). *For a pseudo-random function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, assuming that $k \in \mathcal{K}$ is the pseudo-random function key of F, for $x_1, x_2 \in \mathcal{X}$, there are $F_k(x_1) + F_k(x_2) = F_k(x_1 + x_2)$, $F_k(x_1) \cdot F_k(x_2) = F_k(x_1 \cdot x_2) \in \mathcal{Y}$.*

Definition 4 (Perturbed PRG). *A perturbed pseudorandom generator, denoted as $G_\gamma(\cdot)$, is defined such that for $x_1, x_2 \in \mathcal{X}$, there exists $\gamma > 0$ satisfying the following conditions:*

1. *When $x_1 = x_2$, $\Pr(G_\gamma(x_1) = G_\gamma(x_2)) \leq \varepsilon(n)$ and $\|G_\gamma(x_1) - G_\gamma(x_2)\| < \gamma$, the practical meaning is that for the same input, the output is different.*
2. *When $x_1 \neq x_2$, there is $\|G_\gamma(x_1) - G_\gamma(x_2)\| \geq \gamma$, the practical meaning is that the adversary finds γ.*

2.2 Lattice

Each element of a lattice in $\mathbb{R}^n$ can be expressed linearly by n linearly independent vector integer coefficients. This set of linearly independent vectors is called a lattice basis, and we know that the lattice basis is not unique. Given a set of lattice bases $(v_1, \ldots, v_n)$ in the lattice $\mathcal{L}$, then the fundamental parallelelepiped is

$$\mathcal{P}(v_1, \ldots, v_n) = \left\{ \sum_{i=1}^{n} k_i v_i \middle| k_i \in [0, 1) \right\}.$$

If the lattice base $(v_1, \ldots, v_n)$ is determined, use the symbol $\mathcal{P}(\mathcal{L})$ to replace $\mathcal{P}(v_1, \ldots, v_n)$. $\forall x \in \mathbb{R}^n$, project it onto $\mathcal{P}(\mathcal{L})$. According to the properties of projection, there is a unique $y \in \mathcal{P}(\mathcal{L})$ makes $y - x \in \mathcal{L}$. Use the symbol $\det(\mathcal{L})$ to represent the volume of the fundamental parallelelepiped of the lattice $\mathcal{L}$. In other words, the symbol $\det(\mathcal{L})$ represents the determinant of a matrix composed of a set of lattice bases $(v_1, \ldots, v_n)$. For a given n dimensional lattice, the $\det(\mathcal{L})$ size of any set of lattice bases of the lattice is constant.

2.3 Complexity Assumptions

We state our complexity assumptions below.

Lemma 1 ([7]). *If the adversary has no advantage in breaking the challenge ciphertext without making the challenge hash query to the random oracle, the adversary will make the challenge hash query to the random oracle with probability ε.*

Definition 5 (Decisional Learning with Error (LWE) Assumption). *Suppose a challenger chooses $A \in \mathbb{Z}_q^{(t+1)\times n}, s \in \mathbb{Z}_q^n$, $z \in \mathbb{Z}_q^{t+1}$, $e \in \chi_q^{t+1} \subset \mathbb{Z}_q^{t+1}$ at random. The Decisional LWE assumption is that no polynomial-time adversary is to be able to distinguish the tuple $(A, b = As + e)$ from the tuple (A, z) with more than a negligible advantage.*

2.4 IND-CPA Security Model

In this section, we present IND-CPA security model for BP-ABE[13], defined as follows.

IND-CPA security model

Setup. The challenger runs the Setup algorithm and gives the public parameters, pk to the adversary.

Phase 1. The adversary makes repeated private keys corresponding to sets of attributes $S_1, \ldots, S_{q_1}$.

Challenge. The adversary submits two equal length messages m_0 and m_1. In addition the adversary gives a challenge access structure $\mathbb{A}^*$ such that none of the sets $S_1, \ldots, S_{q_1}$ from Phase 1 satisfy the access structure. The challenger flips a random coin b, and encrypts m_b under $\mathbb{A}^*$. The ciphertext c^* is given to the adversary.

Phase 2. Phase 1 is repeated with the restriction that none of sets of attributes $S_{q_1+1}, ..., S_q$ satisfy the access structure corresponding to the challenge.

Guess. The adversary outputs a guess b' of b.

The advantage of an adversary $\mathcal{A}$ in this game is defined as $\Pr[b' = b] - \frac{1}{2}$. We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase 1 and Phase 2

Definition 6. *A scheme is secure in the IND-CPA model of security if all polynomial-time adversaries have at most a negligible advantage in the above game.*

3 Construct Bilateral-Policy Attribute-Based Encryption

First, we present the ABE based on LWE scheme, which is the basic framework of the BP-ABE scheme.

3.1 LWE-based ABE

ABESetup(n) hash functions $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{Z}_q^{1\times m}$ and $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{Z}_q$.

ABEGen$(\mathcal{H}_1, \mathcal{H}_2, \{att_i\}_{i=1,\ldots,n})$ $A = \begin{pmatrix} \mathcal{H}_1(att_1) \\ \vdots \\ \mathcal{H}_1(att_n) \end{pmatrix} \in \mathbb{Z}_q^{n\times m}$, $u = \begin{pmatrix} \mathcal{H}_2(att_1) \\ \vdots \\ \mathcal{H}_2(att_n) \end{pmatrix} \in \mathbb{Z}_q^n$, $e \leftarrow f_A^{-1}(u)$ such that $Ae = u$, output

$$pk = (A, u), sk = e$$

ABEEnc$(pk, \mathfrak{m}, \{w_i\}_{i=1,\ldots,t}, t \le n)$ Input message $\mathfrak{m} \in \{0,1\}$, set $pk = (B, v)$, and $B^T = \begin{pmatrix} \mathcal{H}_1(w_1) \\ \vdots \\ \mathcal{H}_1(w_t) \end{pmatrix} \in \mathbb{Z}_q^{t\times n}$, and $v = \begin{pmatrix} \mathcal{H}_2(w_1) \\ \vdots \\ \mathcal{H}_2(w_t) \end{pmatrix} \in \mathbb{Z}_q^t$, output

$$c = (c_1, c_2), c_1 = B^T s + \mathbf{x}, c_2 = v^T s + x + \mathfrak{m}\lfloor q/2 \rfloor.$$

ABEDec(sk, c) if $|\{att_i\}_{i=1,\ldots,n} \cap \{w_i\}_{i=1,\ldots,t}| \ge t$, input ciphertext c, compute $\mathfrak{m}' = c_2 - e^T c_1$, then $\mathfrak{m}' < q/4$, $\mathfrak{m} = 0$; else $\mathfrak{m} = 1$.

Fig. 1. ABE based on LWE

Remark 1. The access control of ABE in Fig 1 is very simple. The reason for this simplicity is mainly to facilitate the introduction of more complex access control from other perspectives when constructing BP-ABE later.

Theorem 1. *If an adversary can break LWE-based ABE scheme in the IND-CPA model, then a simulator can be constructed to play the Decisional LWE game with a non-negligible advantage* $(n \geq t)$.

Proof. Suppose there exists a polynomial-time adversary, $\mathcal{A}$, that can attack LWE-based ABE scheme in the IND-CPA model with advantage ε. We build a simulator $\mathcal{B}$ that can play the Decisional LWE game with advantage $\frac{\varepsilon}{2}$. The simulation proceeds as follows:

First, the challenger flips a fair binary coin, μ, outside of $\mathcal{B}$'s view. If $\mu = 0$, the challenger sets $(A, b) = (A, A^T s + x)$; otherwise it sets $(A, b) = (A, z)$ for random $A = (a_1, \cdots, a_m, a_{m+1}) \in \mathbb{Z}_q^{t\times(m+1)}, s \in \mathbb{Z}_q^t$, $z \in \mathbb{Z}_q^{m+1}$, $x \in \chi_q^{m+1}$. And

Setup. Construct a trapdoor algorithm $g_{(\cdot)}^{-1}(\cdot) : \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n \to \mathbb{Z}_q^m$, and $g_W^{-1}(w) = e$, such that $w = W^T e$.

$\mathcal{H}$-Query. The adversary performs a hash query. Before the hash query, $\mathcal{B}$ prepares a hash list to store the hash query results. Let $Att_i = \{att_j\}_{j=1,\ldots,n}^{(i)}$ be the attribute of the i-th query $\mathcal{H}$. If Att_i has been queried, then look up the hash value in the list and output it. Otherwise $\mathcal{B}$ calculates

$$\mathcal{H}_1(att_i) = \left\{ \begin{array}{l} W_i, \text{ if } i \neq i^*, \\ A^*, \text{ otherwise,} \end{array} \right\} \in \mathbb{Z}_q^{n\times m}, \mathcal{H}_2(att_i) = \left\{ \begin{array}{l} w_i, \text{ if } i \neq i^*, \\ u^*, \text{ otherwise,} \end{array} \right\} \in \mathbb{Z}_q^n.$$

Among them, any row vector $((a_1, \cdots, a_m), a_{m+1})_j$ of $((a_1, \cdots, a_m), a_{m+1})$ is randomly distributed in a row of (A^*, u^*), that is, $((a_1, \cdots, a_m), a_{m+1}) \subset (A^*, u^*)$. In the simulation, (A^*, u^*) is not used to reply to the query. The simulator $\mathcal{B}$ replies to the query and stores (i, Att_i, W_i, w_i) and (i^*, Att^*, A^*, u^*) in the hash list.

Phase 1. The adversary initiates a key generation query. For the key query of attribute Att_i, let (i, Att_i, W_i, w_i) be a group. If $i = i^*$, terminate. Otherwise, according to the simulation of the previous stage, get W_i and w_i, let $e_i \leftarrow f_{W_i}^{-1}(w_i)$, e_i is the secret key.

Challenge. The adversary, $\mathcal{A}$, will submit two challenge messages $\mathfrak{m}_0$ and $\mathfrak{m}_1$ to the simulator. The simulator flips a fair binary coin, ν, and returns an encryption of $\mathfrak{m}_\nu$. The ciphertext is output as:

$$c^* = (c_1^*, c_2^*), c_1^* = b^*, c_2^* = b^{**} + \lfloor q/2 \rfloor \mathfrak{m}_\nu, b = \begin{pmatrix} b^* \\ b^{**} \end{pmatrix} \in \begin{pmatrix} \mathbb{Z}_q^m \\ \mathbb{Z}_q \end{pmatrix}.$$

If $\mu = 0$, then $b = A^T s + e$. And get $c_1^* = b^* = (a_1, \cdots, a_m)^T s + \mathfrak{x}^*$, $c_2^* = b^{**} + \lfloor q/2 \rfloor \mathfrak{m}_\nu = (a_{m+1})^T s + x^* + \lfloor q/2 \rfloor \mathfrak{m}_\nu$, here $e = \begin{pmatrix} \mathfrak{x}^* \\ x^* \end{pmatrix} \in \begin{pmatrix} \chi_q^m \\ \chi_q \end{pmatrix}$. Therefore, the ciphertext is a random encryption of the message $\mathfrak{m}_\nu$ under the $((a_1, \cdots, a_m), a_{m+1})$.

Otherwise, $\mu = 1$, then $b = z$. We then have $c_1^* = z^*$, $c_2^* = z^{**} + \lfloor q/2 \rfloor \mathfrak{m}_\nu$. Since z is random, (c_1^*, c_2^*) will be a random element of $\mathbb{Z}_q^m \times \mathbb{Z}_q$ from the adversaries view and the message contains no information about $\mathfrak{m}_\nu$.

Phase 2. The simulator acts exactly as it did in Phase 1.

Guess. $\mathcal{A}$ will submit a guess ν' of ν. If $\nu = \nu'$ the simulator will output $\mu' = 0$ to indicate that it was given a LWE otherwise it will output $\mu' = 1$ to indicate it was given a random vector.

As shown in the construction the simulator's generation of scheme is identical to that of the actual scheme.

In the case where $\mu = 1$ the adversary gains no information about ν. Therefore, we have $\Pr[\nu \neq \nu' | \mu = 1] = \frac{1}{2}$. Since the simulator guesses $\mu' = 1$ when $\nu \neq \nu'$,we have $\Pr[\mu \neq \mu' | \mu = 1] = \frac{1}{2}$.

If $\mu = 0$ then the adversary sees an encryption of $\mathfrak{m}_\delta$. The adversary's advantage in this situation is ε by definition. Therefore, we have $\Pr[\nu = \nu' | \mu = 0] = \frac{1}{2} + \varepsilon$. Since the simulator guesses $\mu' = 0$ when $\nu = \nu'$, we have $\Pr[\mu \neq \mu' | \mu = 0] = \frac{1}{2} + \varepsilon$.

The overall advantage of the simulator in the Decisional LWE game is $\frac{1}{2}\Pr[\mu \neq \mu' | \mu = 0] + \frac{1}{2}\Pr[\mu \neq \mu' | \mu = 1] - \frac{1}{2} = \frac{1}{2}(\frac{1}{2} + \varepsilon) + \frac{1}{2}\frac{1}{2} - \frac{1}{2} = \frac{1}{2}\varepsilon$.

3.2 Constructing BP-ABE based on LWE

We transform ABE based on LWE into a BP-ABE scheme. In order to use BP-ABE to achieve matchmarking encryption, we hope that the sender of the message only uses part of his or her "public key" to ensure the confidentiality of the identities of both users. Since the user's "public key" is essentially the hash function value of its attributes, the user selects a certain number of attribute hash values from his or her public key, combines them into a temporary public key, encrypts the plaintext with the temporary public key and sends it to the receiver. The receiver first performs a homomorphic pseudo-random function operation on his or her private key and then decrypts it.

Definition 7 (BP-ABE).

Setup(1^λ)**:** *Upon input the security parameter* 1^λ *the randomized setup algorithm outputs the master public key* mpk*, the master policy key* $kpol$*, and the master secret key* msk*. We implicitly assume that all other algorithms take* mpk *as input.*

SKGen(msk, σ)**:** *The randomized sender-key generator takes as input the master secret key* msk*, and attributes* $\sigma \in \{0,1\}^*$*. The algorithm outputs a secret encryption key* ek_σ *for attributes* σ*.*

RKGen(msk, ρ)**:** *The randomized receiver-key generator takes as input the master secret key* msk*, and attributes* $\rho \in \{0,1\}^*$*. The algorithm outputs a secret decryption key* dk_ρ *for attributes* ρ*.*

PolGen$(kpol, \mathbb{S})$**:** *The randomized receiver policy generator takes as input the master policy key* $kpol$*, and a policy* $\mathbb{S} : \{0,1\}^* \to \{0,1\}$ *represented as a circuit. The algorithm outputs a secret decryption key* $dk_\mathbb{S}$ *for the circuit* $\mathbb{S}$*.*

Enc$(ek_\sigma, \mathbb{R}, m)$**:** *The randomized encryption algorithm takes as input a secret encryption key* ek_σ *for attributes* $\sigma \in \{0,1\}^*$*, a policy* $\mathbb{R} : \{0,1\}^* \to \{0,1\}$ *represented as a circuit, and a message* $m \in \mathcal{M}$*. The algorithm produces a ciphertext* c *linked to both and* $\mathbb{R}$*.*

Dec$(dk_\rho, dk_\mathbb{S}, c)$**:** *The deterministic decryption algorithm takes as input a secret decryption key* dk_ρ *for attributes* $\rho \in \{0,1\}^*$*, a secret decryption key* $dk_\mathbb{S}$ *for a circuit* $\mathbb{S} : \{0,1\}^* \to \{0,1\}$*, and a ciphertext* c*. The algorithm outputs either a message* m *or* $\perp$ *(denoting an error).*

Note that the decryption keys dk_ρ and $dk_\mathbb{S}$ are independent, thus allowing a receiver with attributes ρ to obtain decryption keys for different policies $\mathbb{S}$. We also remark that the master policy key $kpol$ could be considered as part of the master secret key msk, but we preferred to use distinct keys for clarity.

BP-ABE based on LWE

Setup(n) HCR functions $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{Z}_q^{1\times m}$ and $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{Z}_q$, generate master polic key $kpol$.

SKGen(σ) Generate the sender's public and secret keys, where $A_\sigma \in \mathbb{Z}_q^{\ell_1 \times m}$ and trapdoor $S_\sigma \leftarrow \Lambda^\perp(A_\sigma, q)$, $ek_\sigma = (A_\sigma, u_\sigma, ek_\sigma)$

$$A_\sigma = \begin{pmatrix} \alpha_1 = \mathcal{H}_1(att_1) \\ \vdots \\ \alpha_{\ell_1} = \mathcal{H}_1(att_{\ell_1}) \end{pmatrix}, u_\sigma = \begin{pmatrix} u_1 = \mathcal{H}_2(att_1) \\ \vdots \\ u_{\ell_1} = \mathcal{H}_2(att_{\ell_1}) \end{pmatrix}, e_\sigma \leftarrow f_{A_\sigma}^{-1}(u_\sigma).$$

RKGen(ρ) Generate the receiver's public and secret keys, where $A_\rho \in \mathbb{Z}_q^{\ell_2 \times m}$ and trapdoor $S_\rho \leftarrow \Lambda^\perp(A_\rho, q)$, $dk_\rho = (A_\rho, u_\rho, ek_\rho)$

$$A_\rho = \begin{pmatrix} \alpha_1 = \mathcal{H}_1(att_1) \\ \vdots \\ \alpha_{\ell_2} = \mathcal{H}_1(att_{\ell_2}) \end{pmatrix}, u_\rho = \begin{pmatrix} u_1 = \mathcal{H}_2(att_1) \\ \vdots \\ u_{\ell_2} = \mathcal{H}_2(att_{\ell_2}) \end{pmatrix}, e_\rho \leftarrow f_{A_\rho}^{-1}(u_\rho).$$

PolGen$(kpol, \mathbb{S})$ Generate a homomorphic pseudo-random function key $dk_\mathbb{S}$ based on the access policy $\mathbb{S}$.

Enc$(ek_\sigma, \mathbb{R}, \mathfrak{m})$ According to the receiver's access control $\mathbb{R}$ and the sender's attribute A_ρ, select t sender's own attributes and generate the sender's homomorphic pseudo-random function key, that is, $(ek_\mathbb{R}, (B, v)) \leftarrow (ek_\sigma = (A_\rho, u_\rho), \mathbb{R})$, calculate the temporary public key $F_{ek_\mathbb{R}}^T(B)$ and $F_{ek_\mathbb{R}}^T(v)$, input the plaintext $\mathfrak{m} \in \{0,1\}$, compute $c = (c_1, c_2, c_3)$,,

$$c_1 = F_{ek_\mathbb{R}}^T(B)s + \mathfrak{x}, c_2 = F_{ek_\mathbb{R}}^T(v)s + x + \mathfrak{m}\lfloor q/2 \rfloor, c_3 \leftarrow psi_pprg(A_\sigma).$$

When the access control $\mathbb{R}$ cannot select t attributes of the sender itself, the temporary public key $= \emptyset$ and the encryption is invalid.

Dec$(dk_\rho, dk_\mathbb{S}, c)$ First, verify that $\mathsf{PSI}(psi_pprg(A_\sigma), psi_pprg(A_\rho)) = 1$(explain later). Input ciphertext c, compute $\mathfrak{m}' = c_2 - F_{dk_\mathbb{S}}^T(dk_\rho)c_1 = c_2 - F_{dk_\mathbb{S}}^T(e)c_1$, when $\mathfrak{m}' < \frac{q}{4}$, $\mathfrak{m} = 0$; otherwise $\mathfrak{m} = 1$.

Note 1. $\mathbb{R}$ has two responsibilities: one is to select the attributes of the message sender and generate a temporary public key; the other is to generate a homomorphic pseudo-random function key $ek_\mathbb{R}$, in which case $ek_\mathbb{R} = dk_\mathbb{S}$. $\mathbb{S}$ has only one responsibility, which is to generate a homomorphic pseudo-random function key $dk_\mathbb{S}$.

3.3 Security Proof

We will now show that the above construction satisfies the IND-CPA security definition. First, prove the security of the following intermediate scheme.

Intermediate Scheme

Setup(n) HCR functions $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{Z}_q^{1 \times m}$ and $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{Z}_q$ ~~, generate master polic key $kpol$~~.

SKGen(σ) Generate the sender's public and private keys $A_\sigma \in \mathbb{Z}_q^{\ell_1 \times m}$ and trapdoor $S_\sigma \leftarrow \Lambda^\perp(A_\sigma, q)$, $ek_\sigma = (A_\sigma, u_\sigma, ek_\sigma)$

$$A_\sigma = \begin{pmatrix} \alpha_1 = \mathcal{H}_1(att_1) \\ \vdots \\ \alpha_{\ell_1} = \mathcal{H}_1(att_{\ell_1}) \end{pmatrix}, u_\sigma = \begin{pmatrix} u_1 = \mathcal{H}_2(att_1) \\ \vdots \\ u_{\ell_1} = \mathcal{H}_2(att_{\ell_1}) \end{pmatrix}, e_\sigma \leftarrow f_{A_\sigma}^{-1}(u_\sigma).$$

RKGen(ρ) Generate the receiver's public and private keys, where$A_\rho \in \mathbb{Z}_q^{\ell_2 \times m}$ and trapdoor $S_\rho \leftarrow \Lambda^\perp(A_\rho, q)$, $dk_\rho = (A_\rho, u_\rho, ek_\rho)$

$$A_\rho = \begin{pmatrix} \alpha_1 = \mathcal{H}_1(att_1) \\ \vdots \\ \alpha_{\ell_2} = \mathcal{H}_1(att_{\ell_2}) \end{pmatrix}, u_\rho = \begin{pmatrix} u_1 = \mathcal{H}_2(att_1) \\ \vdots \\ u_{\ell_2} = \mathcal{H}_2(att_{\ell_2}) \end{pmatrix}, e_\rho \leftarrow f_{A_\rho}^{-1}(u_\rho).$$

~~**PolGen**$(kpol, \mathbb{S})$ Generate a homomorphic pseudo-random function key $dk_{\mathbb{S}}$ based on the access policy $\mathbb{S}$.~~

Enc$(ek_\sigma, \mathbb{R}, \mathfrak{m})$ According to the receiver's access control $\mathbb{R}$ and the sender's attribute A_ρ, select t of the sender's own attributes ~~and generate the sender's homomorphic pseudo-random function key~~, that is ~~$(ek_{\mathbb{R}},$~~(B, v)~~$)$~~$\leftarrow (ek_\sigma = (A_\rho, u_\rho), \mathbb{R})$, ~~calculate temporary public key $F_{ek_{\mathbb{R}}}^T(B)$ and $F_{ek_{\mathbb{R}}}^T(v)$,~~ input message $\mathfrak{m} \in \{0,1\}$, compute $c = (c_1, c_2$~~$, c_3$~~$)$,

$$c_1 = B^T s + \mathfrak{x}, c_2 = v^T s + x + \mathfrak{m}\lfloor q/2 \rfloor.$$

When the access control $\mathbb{R}$ cannot select t attributes of the sender itself, the temporary public key $= \emptyset$ and the encryption is invalid.

Dec$(dk_\rho$~~$, dk_{\mathbb{S}}$~~$, c)$ ~~First, verify that PSI$(psi_pprg(A_\sigma), psi_pprg(A_\rho)) = 1$.~~ Input ciphertext c, compute $\mathfrak{m}' = c_2 - F_{dk_{\mathbb{S}}}^T(dk_\rho)c_1 = c_2 - F_{dk_{\mathbb{S}}}^T(e)c_1$, when $\mathfrak{m}' < \frac{q}{4}$, $\mathfrak{m} = 0$; otherwise $\mathfrak{m} = 1$.

Theorem 2. *If an adversary can break intermediate scheme in the IND-CPA model, then a simulator can be constructed to play the Decisional LWE game with a non-negligible advantage.*

Proof. Suppose there exists a polynomial-time adversary, $\mathcal{A}$, that can attack LWE-based ABE scheme in the IND-CPA model with advantage ε. We build a simulator $\mathcal{B}$ that can play the Decisional LWE game with advantage $\frac{\varepsilon}{2}$. The simulation proceeds as follows:

First, the challenger flips a fair binary coin, μ, outside of $\mathcal{B}$'s view. If $\mu = 0$, the challenger sets $(A, b) = (A, A^T s + e)$; otherwise it sets $(A, b) = (A, z)$ for random $A \in \mathbb{Z}_q^{t \times (m+1)}$, $s \in \mathbb{Z}_q^t$, $z \in \mathbb{Z}_q^{m+1}$, $e \in \chi_q^{m+1}$. And

Setup. Constructing a trapdoor algorithm $g_{(\cdot)}^{-1}(\cdot) : \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n \to \mathbb{Z}_q^m$, and $g_W^{-1}(w) = e$, such that $w = We$.

$\mathcal{H}$-Query. The adversary performs a hash query. Before the hash query, $\mathcal{B}$ prepares a hash list to store the hash query results. $Att_i^1 = \{att_j^1\}_{j=1,\ldots,\ell_1}^{(i)}$, $Att_i^2 = \{att_j^2\}_{j=1,\ldots,\ell_2}^{(i)}$ is the i-th query property of $\mathcal{H}$. Among them, the adversary $\mathcal{A}$ only queries $Att_i^1 = \{att_j^1\}_{j=1,\ldots,\ell_1}^{(i)}$. If Att_i^1 has been queried, then look up the hash value in the list and output it. Otherwise $\mathcal{B}$ calculates

$$\mathcal{H}_1(att_i^\tau) = \begin{cases} W_i^\tau \& B_i \subset W_i^1 \cap W_i^2, \text{ if } i \neq i^*, \\ A^\tau \& A^* \subset A^1 \cap A^2, \text{ otherwise}, \end{cases} \tau = 1, 2$$

$$\mathcal{H}_2(att_i^\tau) = \begin{cases} w_i^\tau \& w_i^1 \cap w_i^2 = v_i, \text{ if } i \neq i^*, \\ u^\tau \& u^1 \cap u^2 = u^*, \text{ otherwise}, \end{cases}$$

In the simulation, (A^τ, u^τ) is not used to reply to the query. The simulator $\mathcal{B}$ replies to the query, stores $(i, Att_i^\tau, W_i^\tau, w_i^\tau)$ and $(i^*, Att^\tau, A^\tau, u^\tau)$ in the hash list.

Phase 1. The adversary initiates a key generation query. For the key query of attribute Att_i^1, let $(i, Att_i^1, W_i^1, w_i^1)$ be a group. If $i = i^*$, terminate. Otherwise, according to the simulation of the previous stage, get W_i^1 and w_i^1, let $e_i \leftarrow f_{W_i^1}^{-1}(w_i^1)$, e_i is the private key.

Challenge. The adversary, $\mathcal{A}$, will submit two challenge messages $\mathfrak{m}_0$ and $\mathfrak{m}_1$ to the simulator. The simulator flips a fair binary coin, ν, and returns an encryption of $\mathfrak{m}_\nu$. The ciphertext is output as:

$$c^* = (c_1^*, c_2^*), c_1^* = b^*, c_2^* = b^{**} + \lfloor q/2 \rfloor \mathfrak{m}_\nu, b = \begin{pmatrix} b^* \\ b^{**} \end{pmatrix} \in \begin{pmatrix} \mathbb{Z}_q^m \\ \mathbb{Z}_q \end{pmatrix}.$$

If $\mu = 0$, then $b = As + e$. And get $c_1^* = b^* = (a_1, \cdots, a_m)^T s + \mathtt{x}^*$, $c_2^* = b^{**} + \lfloor q/2 \rfloor \mathfrak{m}_\nu = (a_{m+1})^T s + x^* + \lfloor q/2 \rfloor \mathfrak{m}_\nu$, here $e = \begin{pmatrix} \mathtt{x}^* \\ x^* \end{pmatrix} \in \begin{pmatrix} \chi_q^m \\ \chi_q \end{pmatrix}$. Therefore, the ciphertext is a random encryption of the message $\mathfrak{m}_\nu$ under $((a_1, \cdots, a_m), a_{m+1})$.

Otherwise, $\mu = 1$, then $b = z$. We then have $c_1^* = z^*$, $c_2^* = z^{**} + \lfloor q/2 \rfloor \mathfrak{m}_\nu$. Since z is random, (c_1^*, c_2^*) will be a random element of $\mathbb{Z}_q^m \times \mathbb{Z}_q$ from the adversaries view and the message contains no information about $\mathfrak{m}_\nu$.

Phase 2. The simulator acts exactly as it did in Phase 1.

Guess. $\mathcal{A}$ will submit a guess ν' of ν. If $\nu = \nu'$ the simulator will output $\mu' = 0$ to indicate that it was given a LWE otherwise it will output $\mu' = 1$ to indicate it was given a random vector.

As shown in the construction the simulator's generation of scheme is identical to that of the actual scheme.

In the case where $\mu = 1$ the adversary gains no information about ν. Therefore, we have $\Pr[\nu \neq \nu' | \mu = 1] = \frac{1}{2}$. Since the simulator guesses $\mu' = 1$ when $\nu \neq \nu'$,we have $\Pr[\mu \neq \mu' | \mu = 1] = \frac{1}{2}$.

If $\mu = 0$ then the adversary sees an encryption of $\mathfrak{m}_\delta$. The adversary's advantage in this situation is ε by definition. Therefore, we have $\Pr[\nu = \nu' | \mu = 0] = \frac{1}{2} + \varepsilon$. Since the simulator guesses $\mu' = 0$ when $\nu = \nu'$, we have $\Pr[\mu \neq \mu' | \mu = 0] = \frac{1}{2} + \varepsilon$.

The overall advantage of the simulator in the Decisional LWE game is $\frac{1}{2}\Pr[\mu \neq \mu' | \mu = 0] + \frac{1}{2}\Pr[\mu \neq \mu' | \mu = 1] - \frac{1}{2} = \frac{1}{2}(\frac{1}{2} + \varepsilon) + \frac{1}{2}\frac{1}{2} - \frac{1}{2} = \frac{1}{2}\varepsilon$.

Next, we prove that the BP-ABE scheme without authentication function is indistinguishable from the intermediate scheme.

Lemma 2. *Assuming that $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{Z}_q^{1 \times m}$ and $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{Z}_q$ are HCR functions, and F is a secure homomorphic pseudo-random function, the intermediate scheme is indistinguishable from the BP-ABE scheme without verification function from the perspective of the adversary $\mathcal{A}$.*

Proof. The difference between the two lies in the encryption stage. The intermediate scheme directly uses the temporary public key (B, v) to encrypt the plaintext, while the BP-ABE scheme without authentication function uses the temporary public key $(F_k(B), F_k(v))$ to encrypt the plaintext. Based on the assumption that F is a secure homomorphic pseudorandom function and the properties of the HCR function (pseudorandomness), we have

$$F_{ek_{\mathbb{R}}}(B) \approx_C B, F_{ek_{\mathbb{R}}}(v) \approx_C v.$$

Therefore, it is easy to prove that from the perspective of adversary $\mathcal{A}$, the intermediate scheme is indistinguishable from the BP-ABE scheme without authentication function.

Theorem 3. *In BP-ABE based on LWE, if $c = (c_1, c_2)$ is secure and $psi_pprg(\cdot)$ is pseudo-random, then $c = (c_1, c_2, c_3)$ is also secure.*

Proof. If $c = (c_1, c_2)$ is secure, if the adversary wants to attack BP-ABE based on LWE, then he needs to crack the plaintext of (c_1, c_2) and c_3, or directly crack the user's attributes from c_3. If the adversary wants to crack the plaintext of (c_1, c_2) and c_3, since it is assumed that (c_1, c_2) is secure and (c_1, c_2) and c_3 are two independent encryption systems, it is impossible to crack the ciphertext of (c_1, c_2) through c_3.

If the adversary wants to crack the user's attributes from c_3, since it is assumed that $psi_pprg(\cdot)$ is pseudo-random, the adversary cannot crack the user's attributes.

4 Using PPRG-based PSI to implement authentication

4.1 Constructing PPRG

Algorithm 1 Perturbed Pseudorandom Generator $G_\gamma(\cdot)$

1: **Input:** $x \in \mathbb{Z}_{\{0,1\}}^n$
2: **Output:** $G_\gamma(x) \in \mathbb{Z}_4^m$, $\gamma = \sqrt{m+1}$
3: **Setup:** Initialize $A \in \mathbb{Z}_2^{m \times n}$, $e \in \mathbb{Z}_2^m$
4: **Enc:** Compute

$$G_\gamma(x) = (Ax) \bmod 3 + e$$

4.2 Pseudorandomness proof of PPRG

Theorem 4 (Main theorem). *Assume the construction of the $G_\gamma(\cdot)$ as depicted in Algorithm 1, then $G_\gamma(\cdot)$ is indistinguishable from $u \in \mathbb{Z}_4^m$.*

Proof. To prove this theorem, we divide it into two parts: The first part proves that the indistinguishability probability between $(As) \bmod 3 + e$ and u is bounded by $\frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n$ from Corollary 2.

The second part, Corollary 4, reduces the difficulty of solving the $G_\gamma(\cdot)$ as depicted in Algorithm 1 to the Extrapolated Dihedral Coset Problem (Definition 8). Lemma 4 and Lemma 5 further reduce the Extrapolated Dihedral Coset Problem to the Dihedral Coset Problem, thereby proving the difficulty of solving the $G_\gamma(\cdot)$.

Part 1: The first part demonstrates that the indistinguishability probability between $(As) \bmod 3 + e$ and u is upper-bounded by $\frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n$.

Claim. Suppose that $\mathbb{P}_{\{0,1\}} = \{\Pr(0), \Pr(1)\}$, $\mathbb{P}_{\{0,1\}}^{(1)} = \{\frac{1}{2}, \frac{1}{2}\}$, $\mathbb{P}_{\{0,1\}}^{(2)} = \{\frac{1}{\omega}, \frac{\omega-1}{\omega}\}$, then

$$\begin{aligned}\mathbb{P}_{\{0,1\}}^{(1)} \odot \mathbb{P}_{\{0,1\}}^{(2)} = \mathbb{P}_{\{0,1\}}^{(3)} &= \left\{\frac{1}{2}\cdot\frac{1}{\omega} + \frac{1}{2}\cdot\frac{1}{\omega} + \frac{1}{2}\cdot\frac{\omega-1}{\omega}, \frac{1}{2}\cdot\frac{\omega-1}{\omega}\right\} \\ &= \left\{\frac{1}{2} + \frac{1}{2\omega}, \frac{1}{2} - \frac{1}{2\omega}\right\}.\end{aligned}$$

Claim. Suppose that $\mathbb{P}^{(3)}_{\{0,1\}} = \{\frac{1}{2}+\frac{1}{2\omega}, \frac{1}{2}-\frac{1}{2\omega}\} = \{\Pr(0), \Pr(1)\}$, then

$$\begin{aligned}
\mathbb{P}_{\{0,1,2\}} =& \{\overset{'}{\Pr}(0), \overset{'}{\Pr}(1), \overset{'}{\Pr}(2)\} = \mathbb{P}^{(3)}_{\{0,1\}} \uplus \mathbb{P}^{(3)}_{\{0,1\}}. \text{ Here,} \\
&\overset{'}{\Pr}(0) = \Pr(0)\Pr(0) = \left(\frac{1}{2}+\frac{1}{2\omega}\right)^2 = \frac{1}{3}+\frac{1}{2\omega}+\frac{1}{4\omega^2}-\frac{1}{12}, \\
&\overset{'}{\Pr}(1) = \Pr(0)\Pr(1)+\Pr(1)\Pr(0) = 2\left(\frac{1}{2}+\frac{1}{2\omega}\right)\left(\frac{1}{2}-\frac{1}{2\omega}\right) \\
&\quad = \frac{1}{3}+\frac{1}{6}-\frac{1}{2\omega^2}, \\
&\overset{'}{\Pr}(2) = \Pr(1)\Pr(1) = \left(\frac{1}{2}-\frac{1}{2\omega}\right)^2 = \frac{1}{3}-\frac{1}{2\omega}+\frac{1}{4\omega^2}-\frac{1}{12}.
\end{aligned}$$

Furthermore, there is also

$$\begin{aligned}
\mathbb{P}'_{\{0,1,2\}} =& \{\overset{''}{\Pr}(0), \overset{''}{\Pr}(1), \overset{''}{\Pr}(2)\} = \mathbb{P}_{\{0,1,2\}} \uplus \mathbb{P}_{\{0,1,2\}} \bmod 3. \text{ Here,} \\
&\overset{''}{\Pr}(0) = \overset{'}{\Pr}(0)\overset{'}{\Pr}(0)+\overset{'}{\Pr}(1)\overset{'}{\Pr}(2)+\overset{'}{\Pr}(2)\overset{'}{\Pr}(1), \\
&\overset{''}{\Pr}(1) = \overset{'}{\Pr}(0)\overset{'}{\Pr}(1)+\overset{'}{\Pr}(1)\overset{'}{\Pr}(0)+\overset{'}{\Pr}(2)\overset{'}{\Pr}(2), \\
&\overset{''}{\Pr}(2) = \overset{'}{\Pr}(0)\overset{'}{\Pr}(2)+\overset{'}{\Pr}(1)\overset{'}{\Pr}(1)+\overset{'}{\Pr}(2)\overset{'}{\Pr}(0).
\end{aligned} \tag{1}$$

The equation (2) can be rewritten as

$$\mathbb{P}'_{\{0,1,2\}} = \begin{pmatrix} \Pr'(0) & \Pr'(2) & \Pr'(1) \\ \Pr'(1) & \Pr'(0) & \Pr'(2) \\ \Pr'(2) & \Pr'(1) & \Pr'(0) \end{pmatrix} \begin{pmatrix} \Pr'(0) \\ \Pr'(1) \\ \Pr'(2) \end{pmatrix} = M_{\mathbb{P}_{\{0,1,2\}}} \mathbb{P}_{\{0,1,2\}}.$$

Claim. For the sequence

$$\mathbb{P}^{(n)}_{\{0,1,2\}} = M_{\mathbb{P}^{(n-1)}_{\{0,1,2\}}} \mathbb{P}^{(n-1)}_{\{0,1,2\}},$$

define

$$\mathbb{P}^{(n)}_{\{0,1,2\}} = (a^{(0)}_n, a^{(1)}_n, a^{(2)}_n) = \left(\frac{1}{3}+\Delta^{(0)}_n, \frac{1}{3}+\Delta^{(1)}_n, \frac{1}{3}+\Delta^{(2)}_n\right).$$

Claim. The sequence

$$\mathbb{P}^{(n)}_{\{0,1,2\}} = M_{\mathbb{P}^{(n-1)}_{\{0,1,2\}}} \mathbb{P}^{(n-1)}_{\{0,1,2\}}$$

is a Cauchy sequence.

Proof. To prove that

$$\mathbb{P}^{(n)}_{\{0,1,2\}} = M_{\mathbb{P}^{(n-1)}_{\{0,1,2\}}} \mathbb{P}^{(n-1)}_{\{0,1,2\}}$$

forms a Cauchy sequence, it suffices to show that for any $\delta > 0$, there exists an$N > 0$ such that for any $n > N$,

$$\left\| \mathbb{P}^{(n)}_{\{0,1,2\}} - \mathbb{P}^{(n-1)}_{\{0,1,2\}} \right\| \leq \delta.$$

Because

$$\mathbb{P}^{(n)}_{\{0,1,2\}} - \mathbb{P}^{(n-1)}_{\{0,1,2\}} = \begin{pmatrix} (\Delta_0^{(n-1)})^2 + 2\Delta_1^{(n-1)}\Delta_2^{(n-1)} \\ (\Delta_2^{(n-1)})^2 + 2\Delta_0^{(n-1)}\Delta_1^{(n-1)} \\ (\Delta_0^{(n-1)})^2 + 2\Delta_0^{(n-1)}\Delta_2^{(n-1)} \end{pmatrix} = \begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)} \\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)} \\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)} \end{pmatrix} \begin{pmatrix} \Delta_0^{(n-1)} \\ \Delta_1^{(n-1)} \\ \Delta_2^{(n-1)} \end{pmatrix}.$$

And

$$\left\| \begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)} \\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)} \\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)} \end{pmatrix} \right\| \le \left\| \begin{pmatrix} \frac{1}{6} - \frac{1}{2\varphi^2} & \frac{1}{6} - \frac{1}{2\varphi^2} & \frac{1}{6} - \frac{1}{2\varphi^2} \\ \frac{1}{6} - \frac{1}{2\varphi^2} & \frac{1}{6} - \frac{1}{2\varphi^2} & \frac{1}{6} - \frac{1}{2\varphi^2} \\ \frac{1}{6} - \frac{1}{2\omega^2} & \frac{1}{6} - \frac{1}{2\omega^2} & \frac{1}{6} - \frac{1}{2\omega^2} \end{pmatrix} \right\| = \frac{1}{2} - \frac{3}{2\omega^2}.$$

So, it is obtained that

$$\left\| \mathbb{P}^{(n)}_{\{0,1,2\}} - \mathbb{P}^{(n-1)}_{\{0,1,2\}} \right\| \le \left(\frac{1}{2} - \frac{3}{2\omega^2} \right)^{n-1} \left\| \mathbb{P}^{(1)}_{\{0,1,2\}} - \mathbb{P}^{(0)}_{\{0,1,2\}} \right\| \le \frac{1}{\sqrt{3}} \left(\frac{1}{2} - \frac{3}{2\omega^2} \right)^{n}.$$

Lemma 3. *For any initial vector $a_0 = (a_0^{(0)}, a_0^{(1)}, a_0^{(2)})$, where $a_0^{(0)}, a_0^{(1)}, a_0^{(2)} \in [(\frac{1}{2} - \frac{1}{2\omega})^2, (\frac{1}{2} + \frac{1}{2\omega})^2]$ and $\sum_{i=0}^{2} a_0^{(i)} = 1$, and $\omega > 2$, the matrix M_{a_0} is generated as follows:*

$$M_{a_0} = \begin{pmatrix} a_0^{(0)} & a_0^{(2)} & a_0^{(1)} \\ a_0^{(1)} & a_0^{(0)} & a_0^{(2)} \\ a_0^{(2)} & a_0^{(1)} & a_0^{(0)} \end{pmatrix}.$$

Then, let $a_{n+1} = M_{a_n} a_n = T a_n$, then $\{a_n\}_{n=0}^{\infty}$ is a Cauchy sequence and converges to $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

Proof. According to Claim 4.2, we know that

$$\begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)} \\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)} \\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)} \end{pmatrix}$$

is a contraction operator, and

$$\left\| \begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)} \\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)} \\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)} \end{pmatrix} \right\| \le \frac{1}{2} - \frac{3}{2\omega^2}.$$

Therefore, the matrix

$$\begin{pmatrix} \Delta_0^{(n-1)} & \Delta_2^{(n-1)} & \Delta_1^{(n-1)} \\ \Delta_1^{(n-1)} & \Delta_0^{(n-1)} & \Delta_2^{(n-1)} \\ \Delta_2^{(n-1)} & \Delta_1^{(n-1)} & \Delta_0^{(n-1)} \end{pmatrix}$$

is contractive, with $(0, 0, 0)$ being both a convergent point and a fixed point of this matrix sequence. Moreover, since $a_{n+1} = M_{a_n} a_n$ has been proven to be a Cauchy sequence, the sequence $\{a_n\}_{n=0}^{\infty}$ converges, and it converges to the fixed point of $T(\cdot)$.

Theorem 5. *Given* $\{a_j\}_{j=0}^{2}$ *and* $\{s_j\}_{j=1}^{2}$ *such that* $a_j \in_R \mathbb{Z}_{\{0,1\}}$, $s \in \mathbb{Z}_{\{0,1\}}$, *and the probability that the components of* s *equal* 0 *is* $\frac{1}{\omega}$, *while the probability that they equal* 1 *is* $\frac{\omega-1}{\omega}$, *where* $\omega > 2$. *Then for any* $i = 0, 1, 2$, *we have*

$$\max_{i=0,1,2}\left|\Pr\left(\sum_{j=0}^{2}(a_j s_j) = i\right) - \Pr(u = i)\right| \leq \frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n.$$

Corollary 1. *For any* $A \in \mathbb{Z}_{\{0,1\}}^{m\times n}$, $s \in \mathbb{Z}_{\{0,1\}}^{n}$ *where the probability that the components of* s *equal* 0 *is* $\frac{1}{\omega}$ *and the probability that they equal* 1 *is* $\frac{\omega-1}{\omega}$, *with* $\omega > 2$, *and* $u \in \mathbb{Z}_{\{0,1,2\}}^{m}$, *then the indistinguishability probability between* $As \bmod 3$ *and* u *is bounded by* $\frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n$.

Corollary 2. *For any* $A \in \mathbb{Z}_{\{0,1\}}^{m\times n}$, $s \in \mathbb{Z}_{\{0,1\}}^{n}$ *where the probability that the components of* s *equal* 0 *is* $\frac{1}{\omega}$ *and the probability that they equal* 1 *is* $\frac{\omega-1}{\omega}$, $e \in_R \mathbb{Z}_{\{0,1\}}^{n}$, $\omega > 2$, *and* $u \in \mathbb{Z}_{\{0,1,2,3\}}^{m}$, *then the indistinguishability probability between* $(As) \bmod 3 + e$ *and* u *is bounded by* $\frac{1}{\sqrt{3}}\left(\frac{1}{2} - \frac{3}{2\omega^2}\right)^n$.

Part 2: The second part mainly reduces the difficulty of solving the $G_\gamma(\cdot)$ as depicted in Algorithm 1 to the Dihedral Coset Problem, thereby proving the difficulty of solving the $G_\gamma(\cdot)$.

Claim. The dihedral coset problem is a challenging problem in quantum computing, hence the time complexity for solving it is $O(e^n)$ or $O(n!)$.

Lemma 4. *Currently, there is no efficient algorithm that can solve* DCP_3^ℓ *in polynomial time.*

Proof. By way of contradiction, assume $q = 2^n$ and there exists an efficient algorithm $\mathcal{W}$ that can solve DCP_3^ℓ in polynomial time. Then for an instance of DCP_9^ℓ, we have

$$\begin{aligned}|0\rangle|x_i\rangle + |1\rangle|(x_i + s) \bmod 9\rangle = |0\rangle|x_i'\rangle + |1\rangle|(x_i' + s') \bmod 3\rangle \\ + 3(|0\rangle|x_i''\rangle + |1\rangle|(x_i'' + s'') \bmod 3\rangle), \quad i \leq \ell,\end{aligned}$$

which implies running $\mathcal{W}$ twice can solve $DCP_{9=3^2}^\ell$. Similarly, running $\mathcal{W}$ four times can solve $DCP_{81=3^4}^\ell$, and so forth, requiring $\frac{n}{2}$ executions of $\mathcal{W}$ to solve DCP_q^ℓ. Let $O(\mathcal{W})$ denote the time complexity of $\mathcal{W}$. Thus, we have

$$\frac{n}{2}O(\mathcal{W}) \geq O(e^n) \Rightarrow O(\mathcal{W}) \geq \frac{2O(e^n)}{n},$$

or

$$\frac{n}{2}O(\mathcal{W}) \geq O(n!) \Rightarrow O(\mathcal{W}) \geq \frac{2O(n!)}{n}.$$

This contradicts the hypothesis that $\mathcal{W}$ can solve DCP_3^ℓ in polynomial time, thus proving the theorem.

Definition 8 (Extrapolated Dihedral Coset Problem with Model 3, [8]). *Given security parameter* κ, *input instance* $EDCP_{n,3,\rho}^{\ell}$ *where* 3 *denotes the modulus,* ρ *denotes the probability density function, and* ℓ *denotes the number of states, each state is represented as*

$$\sum_{j\in \mathrm{supp}(\rho)} \rho(j)|j\rangle|(x_i + js) \bmod 3\rangle, \quad i \leq \ell,$$

with 3 *bits stored, where* $x_i \in_R \mathbb{Z}_3^n$, $s \in \mathbb{Z}_4^n$. *The problem* $EDCP_{n,3,\rho}^{\ell}$ *is considered broken if* s *can be computed with probability* $\mathrm{poly}(1/(n\log 3))$ *within* $\mathrm{poly}(n\log 3)$ *time.*

Lemma 5. *If there exists an algorithm to solve $EDCP^{\ell}_{n,3,\rho}$, then there exists an algorithm to solve DCP^{ℓ}_{3}.*

Proof. Let

$$|b\rangle = \frac{1}{\sqrt{2}}|0\rangle|x_i\rangle + \frac{1}{\sqrt{2}}|1\rangle|(x_i + s) \bmod 3\rangle.$$

Thus, $\rho(0)|0\rangle = \frac{1}{\sqrt{2}}|0\rangle$, $\rho(1)|1\rangle = \frac{1}{\sqrt{2}}|1\rangle$, implying DCP^{ℓ}_{3} is a special case of $EDCP^{\ell}_{n,3,\rho}$. Therefore, if there exists an algorithm to solve $EDCP^{\ell}_{n,3,\rho}$, then there exists an algorithm to solve DCP^{ℓ}_{3}.

Lemma 6 ([8]). *Let $(n, q, r = \Omega(\sqrt{\kappa}))$ be an instance of G-EDCP, and (n, q, α) an instance of LWE. If there exists an algorithm to solve $LWE_{n,q,\alpha}$, then there exists an algorithm to solve $G\text{-}EDCP^{\ell}_{n,q,\rho_r}$.*

Corollary 3. *Let $(n, 3, r = \Omega(\sqrt{\kappa}))$ be an instance of G-EDCP, and $(n, 3, \alpha)$ an instance of LWE. If there exists an algorithm to solve $LWE_{n,3,\alpha}$, then there exists an algorithm to solve $G\text{-}EDCP^{\ell}_{n,3,\rho_r}$.*

Corollary 4. *Let $(n, 3, r = \Omega(\sqrt{\kappa}))$ be an instance of G-EDCP, and $G(x) = As \bmod 3 + e$, where $A \in \mathbb{Z}^{m\times n}_{\{0,1\}}$, $x \in \mathbb{Z}^{n}_{\{0,1\}}$, $e \in \mathbb{Z}^{m}_{\{0,1\}}$. If there exists an algorithm to recover x from $G(x)$, then there exists an algorithm to solve $G\text{-}EDCP^{\ell}_{n,3,\rho_r}$.*

4.3 Correctness proof of PPRG

Theorem 6. *Assume the construction of the $G_\gamma(\cdot)$ as depicted in Algorithm 1, then $G_\gamma(\cdot)$ satisfies Definition 4, and $\gamma = \sqrt{m+1}$.*

Proof. Prove each statement separately. First, when $x_1 = x_2$, it be known that

$$\Pr(G_\gamma(x_1) = G_\gamma(x_2)) = \Pr(e_1 = e_2) = \frac{1}{2^m}.$$

Additionally, set $\gamma = \sqrt{m+1}$, so

$$\|(Ax_1 + e_1) - (Ax_2 + e_2)\| = \|e_1 - e_2\| < \gamma.$$

When $x_1 \neq x_2$, set $v_1 = G_\gamma(x_1)$, $v_2 = G_\gamma(x_2)$, and know that

$$\Pr(\|v_1 - v_2\| \leq \sqrt{m}) = \sum_{k=0}^{m} C^k_m \left(\frac{1}{3}\right)^k \left(\frac{1}{2}\right)^{m-k} + \sum_{k=0}^{n/2} C^k_n \left(\frac{1}{3}\right)^k \left(\frac{1}{6}\right)^k \left(\frac{1}{2}\right)^{m-2k}.$$

Because

$$\begin{aligned}\sum_{k=0}^{m} C^k_m \left(\frac{1}{3}\right)^k \left(\frac{1}{2}\right)^{m-k} &= \frac{1}{2^m}\left(\frac{2}{3} + \left(\frac{2}{3}\right)^2 + \cdots + \left(\frac{2}{3}\right)^m\right) \\ &= \frac{3}{2^m}\left(1 - \left(\frac{2}{3}\right)^m\right),\end{aligned}$$

and

$$\sum_{k=0}^{m/2} C_m^k \left(\frac{1}{3}\right)^k \left(\frac{1}{6}\right)^k \left(\frac{1}{2}\right)^{m-2k} \leq \frac{3\cdot 6}{17} \frac{1}{2^{m-\frac{m}{2}}} \left(1 - \left(\frac{1}{3\cdot 6}\right)^{\frac{m}{2}}\right).$$

Therefore

$$\Pr(\|v_1 - v_2\| \leq \sqrt{m} < \sqrt{m+1}) \leq \frac{1}{2^m}.$$

Thus, there is a very high probability that $\|v_1 - v_2\| \geq \sqrt{m+1}$.

4.4 Our PSI Protocol from PPRG

0. P_1 and P_2 agree on security parameters λ, σ, protocol parameters $m, \omega, \ell_1, \ell_2$, a hash function $H_1 : \mathbb{Z}_3^* \to \{0,1\}^{\ell_1}$ and a $G_\gamma : \{0,1\}^\omega \to \mathbb{Z}_4^{\ell_2}$, a pseudorandom function $F : \{0,1\}^\lambda \times \{0,1\}^{\ell_1} \to [m]^\omega$.
1. **Precomputation**
   - P_2 performs the following steps:
     - Initialize an $m \times w$ binary matrix D, with all entries set to 1. Let the column vectors of D be denoted as $D_1, D_2, \ldots, D_w$, and initialize them as $D_1 = D_2 = \cdots = D_w = 1^m$.
     - Sample a random key for a pseudorandom function $k \leftarrow \{0,1\}^\lambda$.
     - For each $y \in \mathcal{Y}$, compute $v = F_k(H_1(y))$. Then, for each $i \in [w]$, set $D_i[v[i]] = 0$.
2. **PPRG Evaluation**
   (a) P_2 sends the PRF key k to P_1.
   (b) $\forall x \in \mathcal{X}$, P_1 computes $v = F_k(H_1(x))$ and its PRF value $\psi = G_\gamma(D_1) \circ \cdots \circ G_\gamma(D_w)$ and sends ψ to P_2.
   (c) Let Ψ be the set of PRF values received from P_1. $\forall y \in \mathcal{Y}$, P_2 computes $v = F_k(y)$ and its PPRG value $\|\psi - G_\gamma(D_1) \circ \cdots \circ G_\gamma(D_w)\| < \sqrt{\omega}\gamma$, $\gamma = \sqrt{m+1}$ and outputs y iff $\psi \in \Psi$.

Fig. 2. Private Set Intersection from Perturbed Pseudorandom Generator

Definition 9 (IND-CPA Security Model of $PrOM_\gamma$). *Assume there exists a pseudorandom oracle ($\Pr\mathcal{O}_\gamma$) model (where γ is the upper bound on the norm of the perturbation in $\Pr\mathcal{O}_\gamma$ model), such that for an input x, it outputs two values: one is a random value y_0, and the other is a pseudorandom value y_1 with x as its input.*

- **Setup** *The simulator $\mathcal{B}$ generates the necessary parameters for the algorithms. The adversary $\mathcal{A}$ chooses s and sends it to the simulator $\mathcal{S}$ using OT.*
- **PRF Queries** *The adversary $\mathcal{A}$ performs pseudorandom function queries.*
- **Challenge** *The adversary $\mathcal{A}$ selects a private message m and sends it to the simulator $\mathcal{B}$. The simulator queries the pseudorandom function values of the real scheme, inputs these results into the pseudorandom oracle $\Pr\mathcal{O}_\gamma$ model, obtains two ciphertexts c_0 and c_1, and sends them to the adversary $\mathcal{A}$.*
- **Guessing** *After receiving the two ciphertexts c_0 and c_1, $\mathcal{A}$ guesses which ciphertext corresponds to the encryption of m and sends the guess back to the simulator $\mathcal{B}$.*

The advantage of the adversary $\mathcal{A}$ is defined as the advantage of the simulator $\mathcal{B}$ in distinguishing the outputs of $\Pr\mathcal{O}_\gamma$ model.

Theorem 7. *Suppose the PRF F is a random oracle. If it is hard to distinguish between the two outputs of the pseudorandom oracle* $\Pr\mathcal{O}_\gamma$ *model with disturbed boundary γ, then our protocol is provably secure in the definition 9 with reduction loss $L = q_H$, where q_H is the number of PRF queries to the random oracle.*

Proof. Suppose the adversary $\mathcal{A}$ can (t, ε)-break the scheme with non-negligible advantage. Now, the simulator $\mathcal{S}$ simulates the scheme. Suppose there exists a $\Pr\mathcal{O}_\gamma$ model such that

$$\begin{aligned}\Pr\mathcal{O}_\gamma(x) \to (y_0, y_1),\ &\text{here,}\\ y_0 &= G_\gamma^{black-box}(x) \in \mathbb{Z}_4^{\ell_2},\\ y_1 &\in_R \mathbb{Z}_4^{\ell_2}.\end{aligned}$$

$\mathcal{S}$ controls the random oracle, runs $\mathcal{A}$, and works as follows.

- **Setup** The simulator $\mathcal{S}$ generates some necessary parameters for the algorithms and selects a PRF $F : \{0,1\}^\lambda \times \{0,1\}^{\ell_1} \to [m]^\omega$ with key $k \in \{0,1\}^\lambda$.
- **PRF-Query** The adversary $\mathcal{A}$ makes query about the PRF. The simulator $\mathcal{S}$ pre-establishes lists for handling PRF-Query. That is, for the i^{th} queries $x_i \in \{0,1\}^{\ell_1}$ corresponding to the value of PRF, the simulator $\mathcal{S}$ selects from the PRF $D_i \in \{0,1\}^{m\times w}$ list if available, otherwise selects a random $v_i \in [m]^\omega$, let $D_1 = \cdots = D_\omega = 1^m$, and set $D_i[v[i]] = 0$ for all $i \in [\omega]$ and adds (x_i, D_i) to the PRF list.
- **Challenge** $\mathcal{A}$ selects $x^* \in \mathcal{X}$ and sends it to $\mathcal{S}$. $\mathcal{S}$ using the corresponding pseudorandom function queries, inputs the queried values into the $\Pr\mathcal{O}_\gamma$ model, obtaining ψ_0 and ψ_1, and then sends ψ_0, ψ_1 to $\mathcal{A}$.
- **Guess** Based on the received ψ_0 and ψ_1, $\mathcal{A}$ guesses whether ψ_0 or ψ_1 is the ciphertext of the encrypted message m.

The simulator randomly selects one value x from the PRF list $(x_1, D_1), (x_2, D_2), \ldots, (x_{q_H}, D_{q_H})$ as the challenge PRF query. The simulator can immediately use the adversary's guess result to distinguish between the two outputs of the $\Pr\mathcal{O}_\gamma$ model. This completes the simulation and the solution. The correctness is analyzed as follows.

Indistinguishable simulation. The correctness of the simulation has been explained above. The randomness of the simulation includes all random numbers in the responses to PRF queries. They are

$$D_1, D_2, \ldots, D_{q_H}.$$

According to the setting of the simulation, where D_i are randomly chosen, it is easy to see that the randomness property holds, and thus the simulation is indistinguishable from the real attack.

Probability of successful simulation. There is no abort in the simulation, and thus the probability of successful simulation is 1.

Advantage of breaking the challenge ciphertext. If the adversary guesses with a non-negligible advantage that ψ_b is the encryption of x^*, where $b = 0$ or 1, then the simulator can use the adversary's guess to distinguish between the two outputs of the $\Pr\mathcal{O}_\gamma$ model with a non-negligible advantage.

Advantage and time cost. Let T_s denote the time cost of the simulation. We have $T_s = O(1)$. Therefore, the simulator $\mathcal{B}$ will distinguish between the two outputs of the $\Pr\mathcal{O}_\gamma$ model with $(t + T_s, \frac{\varepsilon}{q_H})$.

This completes the proof of the theorem.

4.5 Implement authentication function based on PPRG (*psi_pprg*)

Authentication function

Setup. Set security parameters λ, σ, protocol parameters $m, \omega, \ell_1, \ell_2$, a hash function $H_1 : \mathbb{Z}_3^* \to \{0,1\}^{\ell_1}$ and a $G_\gamma : \{0,1\}^\omega \to \mathbb{Z}_4^{\ell_2}$, a pseudorandom function $F : \{0,1\}^\lambda \times \{0,1\}^{\ell_1} \to [m]^\omega$.

Precomputation. Receiver performs the following steps:

- Initialize an $m \times w$ binary matrix D, with all entries set to 1. Let the column vectors of D be denoted as $D_1, D_2, \ldots, D_w$, and initialize them as $D_1 = D_2 = \cdots = D_w = 1^m$.
- For each $y \in A_\rho$, compute $v = F_{dk_\mathbb{R}}(y)$. Then, for each $i \in [w]$, set $D_i[v[i]] = 0$.

Verify. (a) $\forall x \in A_\sigma$, Sender computes $v = F_{ek_\mathbb{S}}(x)$ and its PRF value $\psi = G_\gamma(D_1) \circ \cdots \circ G_\gamma(D_w)$ and sends ψ to Receiver.

(b) Let Ψ be the set of PRF values received from Sender. $\forall y \in A_\rho$, Receiver computes $v = F_{dk_\mathbb{R}}(y)$ and its PPRG value $\|\psi - G_\gamma(D_1) \circ \cdots \circ G_\gamma(D_w)\| < \sqrt{\omega}\gamma$, $\gamma = \sqrt{m+1}$ and outputs y iff $\psi \in \Psi$. When the number of matches reaches a certain number (t), output 1; otherwise output 0.

5 BP-ABE ⇒ Matchmaking encryption

In order to construct matchmaking encryption from BP-ABE, we need to prove that BP-ABE satisfies the definition in the literature. Roughly speaking, the theorem states that, under the assumption that the Decisional LWE problem is hard, then BP-ABE satisfies Ateniese et al.'s definition of privacy and authenticity for matchmaking encryption.

$\mathbf{G}^{\mathbf{priv}}_{\mathbf{\Pi},\mathcal{A}}(\lambda)$	$\mathbf{G}^{\mathbf{auth}}_{\mathbf{\Pi},\mathcal{A}}(\lambda)$
$(mpk, k_{pol}, msk) \leftarrow \text{Setup}(1^\lambda)$	$(mpk, k_{pol}, msk) \leftarrow \text{Setup}(1^\lambda)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathbb{R}_0, \mathbb{R}_1, \sigma_0, \sigma_1, \alpha) \leftarrow \mathcal{A}_1^{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3}(1^\lambda, mpk)$	$(c, \rho, \mathbb{S}) \leftarrow \mathcal{A}_1^{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3}(1^\lambda, mpk)$
$b \leftarrow \{0,1\}$	$dk_\rho \leftarrow \text{RKGen}(msk, \rho)$
$ek_{\sigma_b} \leftarrow \text{SKGen}(msk, \sigma_b)$	$dk_\mathbb{S} \leftarrow \text{PolGen}(k_{pol}, \mathbb{S})$
$c \leftarrow \text{Enc}(ek_{\sigma_b}, \mathbb{R}_b, \mathfrak{m}_b)$	$m = \text{Dec}(dk_\rho, dk_\mathbb{S}, c)$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3}(1^\lambda, c, \alpha)$	If $\forall \sigma \in \mathcal{Q}_{\mathcal{O}_1} : (\mathbb{S}(\sigma) = 0) \wedge (\mathfrak{m} \neq \perp)$ return 1
If $(b' = b)$ return 1	Else return 0
Else return 0	

Fig. 3. Games defining privacy and authenticity of ME. Oracles $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ are implemented by $\text{KGen}(msk, \cdot)$, $\text{RKGen}(msk, \cdot)$, $\text{PolGen}(k_{pol}, \cdot)$.

Correctness. The correctness of matchmarking encryption is naturally described as the receiver using the decryption keys dk_ρ and $dk_\mathbb{S}$ to decrypt the ciphertext c and obtain the corresponding plaintext $\mathfrak{m}$. In addition, the sender can encrypt the message with the attribute and send it to the receiver if and only if the access policy $\mathbb{S}$ and $\mathbb{R}$ match. The canonical definition is as follows:

Definition 10 (Correctness of Matchmarking Encryption). *An ME with message space $\mathcal{M}$ is correct if for any $\lambda \in \mathbb{N}$, the corresponding $(mpk, kpol, msk)$ is output, for any $\mathfrak{m} \in \mathcal{M}$, $\sigma, \rho \in \{0,1\}^*$ and $\mathbb{R}, \mathbb{S} : \{0,1\}^* \to \{0,1\}$, such that $\mathbb{S}(\sigma) = 1 \wedge \mathbb{R}(\rho) = 1$:*

$$\Pr[Dec(dk_\rho, dk_\mathbb{S}, Enc(ek_\sigma, \mathbb{R}, \mathfrak{m})) = \mathfrak{m}] \geq 1 - negl(\lambda),$$

where $ek_\sigma \leftarrow SKGen(msk, \sigma)$, $dk_\rho \leftarrow RKGen(msk, \rho)$ and $dk_\mathbb{S} \leftarrow PolGen(kpol, \mathbb{S})$.

Security. There are two cases: In the first case, when the attributes and access controls of the sender and the receiver do not match, the adversary cannot crack the plaintext. There are four cases of mismatch, namely, the receiver attribute ρ does not satisfy the sender access control $\mathbb{R}_0$ and $\mathbb{R}_1$; the sender attribute σ_0, σ_1 does not satisfy the receiver access control $\mathbb{S}$; the receiver attribute ρ does not satisfy the sender access control $\mathbb{R}_0$ and the sender attribute σ_1 does not satisfy the receiver access control $\mathbb{S}$; the receiver attribute ρ does not satisfy the sender access control $\mathbb{R}_1$ and the sender attribute σ_0 does not satisfy the receiver access control $\mathbb{S}$.

Definition 11 (Security of Matchmarking Encryption). *Matchmarking encryption Π is secure if for all effective PPT adversaries $\mathcal{A}$:*

$$\left| \Pr\left[G^{\text{priv}}_{\Pi,\mathcal{A}} = 1 \right] - \frac{1}{2} \right| \leq \text{negl}(\lambda),$$

where game $\mathbf{G}^{\mathbf{auth}}_{\mathbf{\Pi},\mathcal{A}}(\lambda)$ is depicted in Fig.3. Adversary $\mathcal{A}$ is called valid if $\forall \rho \in \mathcal{Q}_{\mathcal{O}_2}$, $\forall \mathbb{S} \in \mathcal{Q}_{\mathcal{O}_3}$ it satisfies the following invariant:

- *(Mismatch condition). Either*

$$\begin{aligned} &(\mathbb{R}_0(\rho) = \mathbb{R}_1(\rho) = 0) \vee (\mathbb{S}(\sigma_0) = \mathbb{S}(\sigma_1) = 0) \\ &\quad \vee (\mathbb{R}_0(\rho) = \mathbb{S}(\sigma_1) = 0) \vee (\mathbb{R}_1(\rho) = \mathbb{S}(\sigma_0) = 0); \end{aligned} \tag{2}$$

- *(Match condition). Or (i.e., $\exists \hat{\rho} \in \mathcal{Q}_{\mathcal{O}_2}$, $\exists \hat{\mathbb{S}} \in \mathcal{Q}_{\mathcal{O}_3}$ s.t. Eq. (2) does not hold)*

$$(\mathfrak{m}_0 = \mathfrak{m}_1) \wedge (\mathbb{R}_0(\rho) = \mathbb{R}_1(\rho)0) \wedge (\mathbb{S}(\sigma_0) = \mathbb{S}(\sigma_1)).$$

Theorem 8. *If an adversary can break matchmaking encryption scheme in the privacy of Fig3, then a simulator can be constructed to play the Decisional LWE game with a non-negligible advantage.*

Proof. Because BP-ABE is based on LWE, encryption and decryption can be performed correctly when both parties meet the access policy. Because the essence of our matchmaking encryption is BP-ABE, its security also depends on BP-ABE. If BP-ABE is secure, then so is matchmaking encryption.

6 Conclusion

This paper presents a matchmaking encryption scheme under the Decisional LWE assumptions in the IND-CPA model. We construct our ME scheme using bilateral-policy attribute-based encryption and non-interactive private set intersection, which is based on an LWE-based scheme and a perturbed pseudorandom generator.

From a theoretical perspective, the scheme in this paper indeed realizes the description of matchmaking encryption by Ateniese et al. However, from a practical perspective, the efficiency of the scheme is quite low, as it can only encrypt bit by bit. Therefore, improving the efficiency of the ME scheme is the direction for future modifications in this paper.

References

1. Agrawal, S., Wu, D.J.: Functional encryption: Deterministic to randomized functions from simple assumptions. In: Coron, J.S., Nielsen, J.B. (eds.) Advances in Cryptology – EUROCRYPT 2017. pp. 30–61. Springer International Publishing, Cham (2017)
2. Ateniese, G., Francati, D., Nuñez, D., Venturi, D.: Match me if you can: Matchmaking encryption and its applications. Journal of Cryptology **34**(3), 16 (2021). https://doi.org/10.1007/s00145-021-09381-4, https://doi.org/10.1007/s00145-021-09381-4
3. Ateniese, G., Francati, D., Nuñez, D., Venturi, D.: Match me if you can: Matchmaking encryption and its applications. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019. pp. 701–731. Springer International Publishing, Cham (2019)
4. Ateniese, G., Kirsch, J., Blanton, M.: Secret handshakes with dynamic and fuzzy matching. In: Network and Distributed System Security Symposium (2007), https://api.semanticscholar.org/CorpusID:2062405
5. Balfanz, D., Durfee, G., Shankar, N., Smetters, D., Staddon, J., Wong, H.C.: Secret handshakes from pairing-based key agreements. In: 2003 Symposium on Security and Privacy, 2003. pp. 180–196 (2003). https://doi.org/10.1109/SECPRI.2003.1199336
6. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP '07). pp. 321–334 (2007). https://doi.org/10.1109/SP.2007.11
7. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) Advances in Cryptology — CRYPTO 2001. pp. 213–229. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
8. Brakerski, Z., Kirshanova, E., Stehlé, D., Wen, W.: Learning with errors and extrapolated dihedral cosets. In: Public-Key Cryptography – PKC 2018. pp. 702–727. Springer International Publishing (2018)
9. Chase, M., Miao, P.: Private set intersection in the internet setting from lightweight oblivious prf. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 34–63. Springer International Publishing, Cham (2020)
10. Chen, J., Li, Y., Wen, J., Weng, J.: Identity-based matchmaking encryption from standard assumptions. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology – ASIACRYPT 2022. pp. 394–422. Springer Nature Switzerland, Cham (2022)
11. Chiku, S., Hashimoto, K., Hara, K., Shikata, J.: Identity-based matchmaking encryption, revisited: Efficient constructions with strong security. Cryptology ePrint Archive, Paper 2023/1435 (2023), https://eprint.iacr.org/2023/1435
12. Francati, D., Guidi, A., Russo, L., Venturi, D.: Identity-based matchmaking encryption without random oracles. In: Adhikari, A., Küsters, R., Preneel, B. (eds.) Progress in Cryptology – INDOCRYPT 2021. pp. 415–435. Springer International Publishing, Cham (2021)
13. Guo, F., Susilo, W., Mu, Y.: Introduction to Security Reduction (2018), https://api.semanticscholar.org/CorpusID:49426405
14. Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: Spot-light: Lightweight private set intersection from sparse ot extension. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019. pp. 401–431. Springer International Publishing, Cham (2019)
15. Regev, O.: Quantum computation and lattice problems. In: Proceedings of the 43rd Symposium on Foundations of Computer Science. pp. 520–529. FOCS '02, IEEE Computer Society, USA (2002)
16. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing. pp. 84–93. STOC '05, Association for Computing Machinery, New York, NY, USA (2005). https://doi.org/10.1145/1060590.1060603, https://doi.org/10.1145/1060590.1060603
17. Regev, O.: The learning with errors problem (invited survey). 2010 IEEE 25th Annual Conference on Computational Complexity pp. 191–204 (2010), https://api.semanticscholar.org/CorpusID:1752854
18. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) Advances in Cryptology – EUROCRYPT 2005. pp. 457–473. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)

19. Shan, Z., Zhang, L., Wu, Q., Lai, Q.: Probabilistic attacks and enhanced security for "private set intersection in the internet setting from lightweight oblivious PRF". Cryptology ePrint Archive, Paper 2024/969 (2024), https://eprint.iacr.org/2024/969
20. Wang, Y., Wang, B., Lai, Q., Zhan, Y.: Identity-based matchmaking encryption with stronger security and instantiation on lattices. Theoretical Computer Science **1029**, 115048 (2025). https://doi.org/https://doi.org/10.1016/j.tcs.2024.115048, https://www.sciencedirect.com/science/article/pii/S0304397524006650
21. Xu, S., Ning, J., Li, Y., Zhang, Y., Xu, G., Huang, X., Deng, R.H.: Match in my way: Fine-grained bilateral access control for secure cloud-fog computing. IEEE Transactions on Dependable and Secure Computing **19**(2), 1064–1077 (2022). https://doi.org/10.1109/TDSC.2020.3001557