Cross-Domain Identity-based Matchmaking Encryption

Axin Wu, Jian Weng (✉), Weiqi Luo, Anjia Yang, Jia-Nan Liu, and Zike Jiang

College of Cyber Security, Jinan University, Guangzhou, 510632, China

Abstract. Recently, Ateniese et al. (CRYPTO 2019) proposed a new cryptographic primitive called matchmaking encryption (ME), which provides fine-grained access control over encrypted data by allowing both the sender and receiver to specify access control policies. The encrypted message can be decrypted correctly if and only if the attributes of the sender and receiver simultaneously meet each other's specified policies. In current ME, when users from different organizations need secret communication, they need to be managed by a single-authority center. However, it is more reasonable if users from different domains obtain secret keys from their own authority centers, respectively. Inspired by this, we extend ME to cross-domain scenarios. Specifically, we introduce the concept of the cross-domain ME and instantiate it in the identity-based setting (i.e., cross-domain identity-based ME). Then, we first formulate and design a cross-domain identity-based ME (IB-ME) scheme and prove its privacy and authenticity in the random oracle model. Further, we extend the cross-domain IB-ME to the multi-receiver setting and give the formal definition, concrete scheme and security proof. Finally, we analyze and implement the schemes, which confirms the efficiency feasibility.

Keywords: Access Control · Matchmaking Encryption · Cross-Domain IB-ME · Cross-Domain Multi-receiver IB-ME.

1 Introduction

The concept of matchmaking encryption (ME) is introduced by Ateniese et al [1]. In ME, the attributes of the sender and receiver must meet the access control policies specified by the other party at the same time in order to decrypt the ciphertext correctly. If the decryption fails, nothing is revealed except the fact that the match does not occur. Concretely, a trusted authority center generates an encryption key and a decryption key according to the attributes of sender and receiver. Additionally, the receiver also receives a policy decryption key that captures a policy the sender's should satisfy. Then, the sender encrypts the message using the encryption key and dynamically specifying the policy that the receiver attribute must meet. The ciphertext can be decrypted correctly if and only if the attributes of the sender and receiver simultaneously meet the access control policy selected by each other. Informally, ME provides the privacy and authenticity of messages while guaranteeing the privacy of the attribute and access control policy of the sender and receiver.

ME opens up a novel way of secret communication. For example, intelligence agents communicate secretly through the anonymous bulletin board [1]. If one party wants to communicate secretly with the other party, they can leverage ME for intelligence interaction without necessity of being real-time online and multiple rounds of interaction, which provides a more secure method of communication owing to avoiding traffic analysis. When two parties of communication are users intended by each other, the information will be recovered correctly. If not, the decryption fails without revealing the reason of failure. The same application can be used in the auction scenario [2], where the matching parties are bidders and the collector. ME can also be applied to social matchmaking to match the ideal partner [3] and marginalized and dissident communities in authoritarian countries [4] to support freedom of speech.

The current ME, however, is not very satisfactory in dealing with secret communication between users of different organizations. In current ME, private keys of the sender and receiver are generated by a single-authority center. In other words, these users belong to the same organization since they are managed by the same authority center. If users belonging to two different organizations (e.g., CIA agents and FBI agents) want to communicate secretly, they need a trust third party to be the authority center, which may be very cumbersome and sometimes difficult. In contrast, if improved with the property that private keys for users are generated by two authority centers in their respective domains, ME can be applied in more practical setting where users in different domains communicate secretly, such as, secret communications between CIA agents and FBI agents in intelligence operations. Based on previous analysis, a natural question follows:

Is there is an efficient ME scheme supporting cross-domain secret communication without a common trust third party?

Inspired by this, we extend the concept of ME to the cross-domain setting. Then, we propose a concrete instantiation of cross-domain ME in the identity-based setting (i.e., cross-domain identity-based ME (IB-ME)) and prove its privacy and authenticity in the random oracle model based on the standard computational bilinear Diffie-Hellman ($CBDH$) assumption. Further, we extend it to the multi-receiver scenario (i.e., cross-domain multi-receiver IB-ME). Finally, concrete contributions are summarized as follows:

- We introduce the concept of cross-domain ME. In cross-domain ME, there is a sender-authority center and a receiver-authority center, which can also be regarded as ME with multiple authorities, as pointed out by Ateniese et al. [1] at CRYPTO 2019. Then, we first instantiate the cross-domain ME in the identity-based setting and propose a cross-domain IB-ME, where the sender-authority center (or the receiver-authority center) can generate the encryption key (or decryption private key) for users in the respective domain. As a result, secret communication can be carried out between users in different domains without looking for an authority center trusted by both sender and receiver. Besides, we also provide the formal security definition and prove its privacy and authenticity in the random oracle model based on the *CBDH* assumption over the bilinear group. Finally, performance analysis demonstrates that the proposed scheme can be implemented efficiently.

– We extend the cross-domain IB-ME to the multi-receiver scenario and propose a cross-domain multi-receiver IB-ME, where the sender can specify a receivers' identity set when encrypting. Besides, we also give its formal security definition and prove its privacy and authenticity in the random oracle model based on *CBDH* assumption over the bilinear group. Finally, performance analysis demonstrates that the proposed scheme can be implemented efficiently.

1.1 Related works

We will distinguish ME from secret handshake (SH), attribute-based encryption (ABE), attribute-based key exchange (AB-KE), and access control encryption ACE, which are similar to ME to some extent.

ME [1] can be regarded as an extension of the non-interactive secret handshake (SH). SH [5, 6] allows two parties belonging to the same group to authenticate each other secretly and negotiate a symmetric key at the same time. In SH, one party can additionally specify the precise identity of the other party. Indeed, ME can not only provide the privacy guarantee similar to that of SH but also give more flexible communication way.

Sahai and Waters [7] proposed the concept of ABE in the setting of fuzzy identity-based encryption. Then, the concept was generalized by Bethencourt et al. [8], where a user is described by multiple attributes. In ABE [9–12], only one party specifies the access control policies that the other party must meet. To overcome this limitation, the dual-policy ABE [13] was proposed. In dual-policy ABE [14], when encrypting a message, the sender specifies an access control policy and attribute set. The receiver can obtain a private key associated with the attribute set and access control policy. Similar to ME, only when the policies in the ciphertext and the key are simultaneously matched can the ciphertext be decrypted correctly. However, there are differences between dual-policy ABE and ME in terms of the syntactical level and security level. As for syntax definition, ME can generate separate keys for attributes and access control policies, which will be more flexible than dual-policy ABE. As for security definition, ME can not only ensure the confidentiality of information but also ensure the authentication of information, which will provide stronger security assurance than dual-policy ABE.

The concept of attribute-based authentication key exchange was introduced by Gorantla et al. [15], which allows participants whose attributes satisfy a fixed access policy to share a secret key. Note that this access control policy which is the same for all participants must be negotiated before running the protocol. Then, a different AB-KE [16] was proposed without bilateral authentication. In their protocol, a client with some attributes certified by an authority can share a private key with a server that defines an access control policy if and only if the user's attributes satisfy the access control policy specified by the server. In contrast in ME, both sender and receiver can specify access control policies, which is more fine-grained than AB-KE [15, 16].

The concept of ACE was proposed by Damgaard et al. [17], which provides flexible access control over encrypted information flow according to the sender's and receiver's identities. In ACE [18, 19], there are three participants: a set of senders, a set of receivers, and a sanitizer. The security goal of ACE is to enforce *non-read* and *non-write* rules described by a policy on encrypted flow. The flow enforcement is performed by the sanitizer, which processes the incoming ciphertext through a random algorithm. As a result, only the receivers allowed to communicate with the source can decrypt the sanitized ciphertext correctly (*no-read* rule). On the other hand, if the source does not have permission to communicate with the receiver, the decryption of the sanitized ciphertext received by the receiver looks like a random message (*no-write* rule). ACE [20] enforces fine-grained access control of information flow according to a single policy. In contrast, ME enables fine-grained access control on encrypted data according to the policies specified by the sender and receiver.

ME allows both sender and receiver to perform fine-grained access control on encrypted data at the same time. Inspired by this idea, related applications of bilateral access control were proposed. For example, Xu et al. [21] proposed the expressive bilateral access control in cloud-fog computing. Wang et al. [22] achieved flexible bidirectional access control in the context of public key encryption with keyword search. Badertscher et al. [23] introduced the policy-compliant signature with policies considering attributes of both sender and receiver. Besides, Francati et al. [24] constructed a IB-ME scheme without random oracles. These schemes extend the application of ME. However, the users' private keys in these schemes are generated by a single-authority center. As far as we know, there is no cross-domain ME scheme, which can be applied to secret communication between users in different organizations.

The remainder of the paper is organized as follows: In Section 2, we introduce preliminaries that will be used later. Then, in Section 3, we give the relevant definitions including syntax and security definition. Next, the construction and proof of cross-domain IB-ME are given in Section 4 and Section 5, respectively. In Section 6 we extend the cross-domain IB-ME to the multi-receiver scenario. Section 7 presents the performance evaluation of schemes. Finally, we summarize this work in Section 8.

2 Preliminaries

2.1 Notations

Throughout the paper, the security parameter is represented by λ. For $a, b \in \mathbb{N}$, let $[a, b] := \{a, a+1, a+2, ..., b\}$. $\mathcal{O}$ represents the random oracle. $\mathbb{O}$ records inputs of $\mathcal{O}$. If a random algorithm can terminate in a polynomial number of steps in λ for any input, the algorithm is probabilistic polynomial-time (PPT). If a function $negl(\lambda)$ in the security parameter λ vanishes faster than the inverse of any polynomial, $negl(\lambda)$ is said to be negligible in λ.

2.2 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p. Let g be the generator of $\mathbb{G}$. The map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is considered to be an admissible bilinear map if it has the following properties:

– Bilinear: For $\forall\ g, h \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, $e(g^a, h^b) = e(g, h)^{ab}$.
– Non-degenerate: $e(g, g) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ represents the unit element of $\mathbb{G}_T$.
– Computable: For any $g, h \in \mathbb{G}$, there is an efficient algorithm to compute $e(g, h)$.

For ease of representation, the above bilinear group system is described as $(p, \mathbb{G}, \mathbb{G}_T, e(\cdot), g) \leftarrow \mathcal{G}(1^\lambda)$.

2.3 Computational Bilinear Diffie-Hellman Assumption

For two cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p, an admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and a generator g of $\mathbb{G}$, the *CBDH* problem is hard in $(\mathbb{G}, \mathbb{G}_T, e)$ if for any PPT adversary $\mathcal{A}$ such that:

$$Pr[\mathcal{A}(p, \mathbb{G}, \mathbb{G}_T, e, g, g^a, g^b, g^c) = e(g, g)^{abc}] \leq negl(\lambda),$$

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

3 Definition

3.1 Concept of Cross-Domain ME

As Ateniese et al. [1] pointed out at the end of their work, there are several important questions about ME including the construction of ME from simpler assumption, the instantiation of black-box construction on better assumptions, the efficient IB-ME without relying on random oracles, ME with multiple authorities, and so on. In this work, we focus on ME with multiple authorities. In current ME, when users from different organizations need secret communication, they have to be managed by a single-authority center. However, it is sometimes difficult to find a trust third party for both organizations, like for two transnational organizations, which will impede the secret communication. To meet this demand, ME needs to be further extended. Specifically, ME with a sender-authority center and a receiver-authority center makes its application more flexible. Especially, it will also be more practical for secret communication between users across organizations, such as FBI agents and CBI agents. Thus, with the extension of original ME, the new ME can be used for cross-domain setting. Users managed by the different authority center can communicate secretly across domains. Obviously, cross-domain ME implies ME [1], because it is totally a single-authority case when an authority center includes a sender-authority center and a receiver-authority center. Actually, cross-domain ME will be more applicable for secret communication between users from two distinct organizations.

In the instantiation of ME, Ateniese et al. [1] proposed IB-ME. Similarly, we also give the instantiation of cross-domain ME (i.e., cross-domain IB-ME) in our work, where the access control policy is a single identity rather than a general policy (expressed as a circuit) and satisfying the policy means that the identity bit strings are equal. When encrypting, the sender only specifies a single receiver, which is a one-to-one secret communication model. If a sender wants to communicate with multiple receivers, he has to encrypt the same message multiple times, which will bring a great burden of computation and communication to the sender. To alleviate the consumption of the sender's resources, we extend cross-domain IB-ME to cross-domain multi-receiver IB-ME, where the sender can specify a receivers' identity set when encrypting the message. The message can be decrypted correctly only when the identity of the receiver belongs to the receivers' identity set. Due to the one-to-many feature, cross-domain multi-receiver IB-ME can be applied to cross-domain data sharing.

3.2 Syntax Definition of Cross-Domain IB-ME

In this subsection, we first formulate the syntax definition of cross-domain IB-ME consisting of the following polynomial-time algorithms:

- **Initialization**$(1^\lambda) \rightarrow pp$. The algorithm is executed by the sender-authority center and the receiver-authority together, which negotiate some common parameters. After inputting the security parameter λ, it outputs the system parameter pp, which is the implicit input of other algorithms.
- **RASetup**$(pp) \rightarrow (mpk_R, msk_R)$. This algorithm is executed by the receiver-authority center. After getting the system parameter pp, it outputs the master public key mpk_R and the master decryption key msk_R of the receiver-authority center.
- **SASetup**$(pp) \rightarrow (mpk_S, msk_S)$. This algorithm is operated by the sender-authority center. After inputting the system parameter pp, it outputs the master public key mpk_S and the master encryption key msk_S of the sender-authority center.
- **EKGen**$(mpk_R, msk_S, ID_S) \rightarrow ek_{ID_S}$. This algorithm is operated by the sender-authority center. After inputting the master public key mpk_R of the receiver-authority center, the master encryption key msk_S of the sender-authority center, and the sender's identity ID_S, it outputs the encryption key ek_{ID_S} of the sender.
- **DKGen**$(msk_R, ID_R) \rightarrow dk_{ID_R}$. This algorithm is run by the receiver-authority center. After inputting the master decryption key msk_R of the receiver-authority center and the receiver's identity ID_R, it outputs the decryption key dk_{ID_R} of the receiver.
- **Enc**$(mpk_S, mpk_R, ek_{ID_S}, M, ID_{Rev}) \rightarrow C$. This algorithm is run by the sender. After inputting the master public key mpk_R of the receiver authority center, the master public key mpk_S of the sender-authority center, the

sender's encryption key ek_{ID_S}, the plaintext message M and the the target identity ID_{Rev}, it outputs the ciphertext C.

– **Dec**$(mpk_S, dk_{ID_R}, C, ID_{Snd}) \rightarrow M$ or $\perp$. This algorithm is run by the receiver. After inputting the master public key mpk_S of the sender-authority center, the receiver's decryption key dk_{ID_R} corresponding to the identity ID_R, the ciphertext C and the target identity ID_{Snd}, it outputs the plaintext M or $\perp$.

Correctness. The intuitive description of correctness is that the ciphertext generated by the sender ID_S with the target identity ID_{Rev} can be correctly encrypted by the receiver with the identity ID_R by setting the target identity ID_{Snd} if and only if $ID_S = ID_{Snd}$ and $ID_R = ID_{Rev}$. Besides, in the case of the mismatch, the privacy of the sender's identity and message is hidden. More formally, a cross-domain IB-ME with message space $\mathcal{M}$ is correct if $\forall\ \lambda \in \mathbb{N}$, $\forall\ ID_S, ID_R, ID_{Snd}, ID_{Rev} \in \{0,1\}^*$, when $pp \leftarrow$ **Initialization**(1^λ), $(mpk_R, msk_R) \leftarrow$ **RASetup**(pp), $msk_S \leftarrow$ **SASetup**(pp), $ek_{ID_S} \leftarrow$ **EKGen**(mpk_R, msk_S, ID_S), $dk_{ID_R} \leftarrow$ **DKGen**(msk_R, ID_R):

$$Pr[\mathbf{Dec}(mpk_S, dk_{ID_R}, ID_{Snd}, \mathbf{Enc}(mpk_S, mpk_R, ek_{ID_S}, ID_{Rev}, M)) = M] \geq 1 - negl(\lambda),$$

whenever $ID_S = ID_{Snd}$ and $ID_R = ID_{Rev}$.

3.3 Security Definition of Cross-Domain IB-ME

In this subsection, we present the security definition of cross-domain IB-ME, which potentially inherits privacy and authenticity of IB-ME [1].

Privacy of Cross-Domain IB-ME. In the case of matching, the receiver decrypts the ciphertext by the decryption key and the target identity. In this case, the receiver can infer the sender's identity. Privacy requires that nothing is revealed to a non-intended receiver, which includes the sender's identity and the message. The formal privacy game $G^{Priv}_{\Pi,\mathcal{A}}(\lambda)$ is presented in **Table** 1. $\mathcal{O}_1$ and $\mathcal{O}_2$ represent encryption key oracle and decryption key oracle. $\mathbb{O}_1$ and $\mathbb{O}_2$ respectively record the identity of accessing the random oracles $\mathcal{O}_1$ and $\mathcal{O}_2$.

Definition 1 (Privacy of Cross-Domain IB-ME). We say that a cross-domain IB-ME satisfies privacy if the above game $G^{Priv}_{\Pi,\mathcal{A}}$ presented in **Table** 1 satisfies the following conditions for any valid PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$:

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

where $\mathcal{A}$ is valid if $\forall\ ID_R \in \mathbb{O}_2$, it satisfies $ID_R \neq ID_{Rev_0} \wedge ID_R \neq ID_{Rev_1}$.

Authenticity of Cross-Domain IB-ME. Authenticity captures security against the malicious sender who does not have the encryption key of the corresponding identity (e.g., *Alice*), but generates a valid ciphertext corresponding to the identity (i.e., $ID_S = Alice$). The forgery (C, ID_R, ID_{Snd}) such that

Table 1. Security Games of Privacy and Authenticity for Cross-Domain IB-ME

$G^{Priv}_{\Pi,\mathcal{A}}(\lambda)$	$G^{Auth}_{\Pi,\mathcal{A}}(\lambda)$
$pp \leftarrow$ **Initialization**(1^λ)	
$(mpk_R, msk_R) \leftarrow$ **RASetup**(pp)	$pp \leftarrow$ **Initialization**(1^λ)
$(mpk_S, msk_S) \leftarrow$ **SASetup**(pp)	$(mpk_R, msk_R) \leftarrow$ **RASetup**(pp)
$(M_0, M_1, ID_{Rev_0}, ID_{Rev_1}, ID_{S_0}, ID_{S_1}, \alpha) \leftarrow \mathcal{A}_1^{\mathcal{O}_1,\mathcal{O}_2}(pp, mpk_R, mpk_S)$	$(mpk_S, msk_S) \leftarrow$ **SASetup**(pp)
$b \leftarrow \{0,1\}$	$(C, ID_R, ID_{Snd}) \leftarrow \mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2}(pp, mpk_R, mpk_S)$
$ek_{ID_{S_b}} \leftarrow$ **EKGen**(mpk_R, msk_S, ID_{S_b})	$dk_{ID_R} \leftarrow$ **DKGen**(msk_R, ID_R)
$C \leftarrow$ **Enc**$(mpk_S, mpk_R, ek_{ID_{S_b}}, ID_{Rev_b}, M_b)$	$M \leftarrow$ **Dec**$(mpk_S, dk_{ID_R}, ID_{Snd}, C)$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}_1,\mathcal{O}_2}(pp, C, \alpha)$	$If\ \forall\ ID_S \in \mathbb{O}_1\ :\ (ID_S \neq ID_{Snd})\ \wedge\ (ID_R \notin \mathbb{O}_2)$
$If\ (b = b')$ **return** 1 *Else* **return** 0	$\wedge\ (M \neq \perp)$ **return** 1 *Else* **return** 0

$Dec(C, dk_{ID_R}, ID_{Snd}) \neq \perp$ is considered valid if $ID_{Snd} \neq ID_S$ for all encryption key ek_{ID_S} acquired by $\mathcal{A}$ and the identity ID_R not held by $\mathcal{A}$ (i.e., $\mathcal{A}$ cannot "forge to itself").

Definition 2 (Authenticity of Cross-Domain IB-ME). We say that a cross-domain IB-ME satisfies authenticity if the above game $G^{Auth}_{\Pi,\mathcal{A}}$ presented in **Table** 1 satisfies the following conditions for any valid PPT adversary $\mathcal{A}$:

$$Pr[G^{Auth}_{\Pi,\mathcal{A}}(\lambda) = 1] \leq negl(\lambda).$$

Definition 3 (Secure Cross-Domain IB-ME). We say that a cross-domain IB-ME Π is semantically secure if Π meets the privacy (*Def. 1*) and authenticity (*Def. 2*).

4 Concrete Construction of Cross-domain IB-ME

4.1 Overview

The overview of the scheme is described as follows: the framework of cross-domain IB-ME is similar to that of [25], which implies that the sender-authority center and the receiver-authority center trust each other. Therefore, the system initialization can be completed by the receiver-authority center, or by the sender-authority center and the receiver-authority center together. Without loss of generality, the system parameter *pp* is jointly negotiated by the sender-authority center and the receiver-authority center. Then, the sender-authority center and the receiver-authority center generate their public keys $\{(P, \Upsilon), (K, \Theta)\}$ and secret keys $\{(\rho, \gamma), (k, \theta)\}$ according to *pp*, respectively. Next, the sender-authority center and the receiver-authority center generate the encryption key ek_{ID_S} and the decryption key dk_{ID_R} for users in their domain, respectively, where Υ is embedded in ek_{ID_S}. When encrypting, the sender obtains the ciphertext by employing the encryption key ek_{ID_S} and the receiver's identity ID_{Rev}. Since Υ and ID_{Rev} are embedded in the encryption process, it can ensure that only the specified receiver can decrypt the ciphertext. When decrypting, the receiver obtains the plaintext message by using the decryption key dk_{ID_R} and the sender's identity ID_{Snd}. Because ek_{ID_S} is used in the ciphertext generation process, and the sender-authority center and the receiver-authority center trust each other, it can ensure that the ciphertext comes from the domain managed by the sender-authority center. In addition, since the sender's identity is required during decryption, this ensures that only the specified sender can generate the ciphertext. Through the above process, the cross-domain secret communication is completed.

4.2 The Concrete Construction

The cross-domain IB-ME consists of the following polynomial-time algorithms:

- **Initialization.** After inputting the security parameter λ, it randomly selects a bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^\lambda)$ with the order $p > 2^\lambda$, where g is a generator of $\mathbb{G}$. Then it also picks three cryptographic hash functions $H : \{0,1\}^* \rightarrow \mathbb{G}$, $H_0 : \{0,1\}^* \rightarrow \mathbb{G}$, $H_1 : \{0,1\}^* \rightarrow \{0,1\}^n$. Finally, the public parameter of the system is $pp = (p, \mathbb{G}, \mathbb{G}_T, e, g, n, H, H_0, H_1)$. Note that here we imply that message length in the plaintext space is n-bit length. If the bit length in the plaintext space is less than n, we can use a polynomial-time computable padding function to expand it to n bits, and this function can also efficiently recover the original plaintext message from the expanded bits.
- **Receiver-Authority Setup.** The receiver-authority randomly selects $\rho, \gamma \in \mathbb{Z}_p^*$. Then, it calculates $P = g^\rho$ and $\Upsilon = g^\gamma$. Finally, it publishes $mpk_R = (P, \Upsilon)$ as the receiver-authority public key and keeps $msk_R = (\rho, \gamma)$ in secret as the master decryption key.
- **Sender-Authority Setup.** The sender-authority randomly selects $k, \theta \in \mathbb{Z}_p^*$ and calculates $K = g^k$ and $\Theta = g^\theta$. Then, the master encryption key and the public key of the sender-authority center are $msk_S = (k, \theta)$ and $mpk_S = (K, \Theta)$, respectively.
- **Encryption-Key Generation.** After inputting the sender identity ID_S, the sender-authority calculates $ek_1 = H_0(ID_S)^k, ek_2 = \Upsilon^\theta$. Finally, the encryption key of the user ID_S is $ek_{ID_S} = (ek_1, ek_2)$.
- **Decryption-Key Generation.** After inputting the receiver identity ID_R, the receiver-authority computes $dk_1 = H(ID_R)^\rho, dk_2 = H(ID_R)^\gamma$. Finally, the decryption key of the user ID_R is $dk_{ID_R} = (dk_1, dk_2)$.
- **Encryption.** Once the public keys mpk_R and mpk_S, the encryption key ek_{ID_S}, the target identity ID_{Rev} and the message $M \in \{0,1\}^n$ are input, the sender randomly picks $\eta_0, \eta_1 \in \mathbb{Z}_p^*$ and computes

$$\mu_1 = e(H(ID_{Rev}), P)^{\eta_0}, \mu_2 = e(H(ID_{Rev}), ek_1 ek_2 g^{\eta_1}),$$

and

$$C_1 = M \oplus H_1(\mu_1) \oplus H_1(\mu_2), C_2 = g^{\eta_0}, C_3 = ek_1 H_0(ID_S)^{\eta_1}, C_4 = H(ID_{Rev})^{-\eta_1}.$$

Finally, the ciphertext is $C = (C_1, C_2, C_3, C_4)$.

– **Decryption.** After inputting the ciphertext $C = (C_1, C_2, C_3, C_4)$ from ID_{Snd} and the decrypting key $dk_{ID_R} = (dk_1, dk_2)$, the receiver computes

$$\mu_1' = e(C_2, dk_1), \mu_2' = e(dk_2, \Theta)e(g, C_4)^{-1}e(H(ID_R), C_3)e(H_0(ID_{Snd}), C_4).$$

Then, he can obtain the plaintext $M = C_1 \oplus H_1(\mu_1') \oplus H_1(\mu_2')$ if the sender identity is $ID_{Snd} = ID_S$ and the receiver identity is $ID_{Rev} = ID_R$.

1. If $ID_{Snd} = ID_S$ and $ID_{Rev} = ID_R$:

$$\begin{aligned}
\mu_1' &= e(C_2, dk_1) = e(g^{\eta_0}, H(ID_R)^{\rho}) \\
&= e(g, H(ID_R))^{\rho\eta_0} = e(P, H(ID_{Rev}))^{\eta_0} = \mu_1, \ and \\
\mu_2' &= e(dk_2, \Theta)e(g, C_4)^{-1}e(H(ID_R), C_3)e(H_0(ID_{Snd}), C_4) \\
&= e(H(ID_R), g)^{\gamma\cdot\theta+\eta_1}e(H(ID_R), H_0(ID_{Snd}))^k \\
&= e(H(ID_{Rev}), ek_2)e(H(ID_{Rev}), g)^{\eta_1}e(H(ID_{Rev}), ek_1) \\
&= e(H(ID_{Rev}), ek_1 ek_2 g^{\eta_1}) = \mu_2.
\end{aligned}$$

2. If $ID_{Snd} \neq ID_S$ and $ID_{Rev} \neq ID_R$:

$$\begin{aligned}
\mu_1' &= e(C_2, dk_1) = e(g^{\eta_0}, H(ID_R)^{\rho}) \\
&= e(g, H(ID_R))^{\rho\eta_0} \neq e(P, H(ID_{Rev}))^{\eta_0} = \mu_1, \ and \\
\mu_2' &= e(dk_2, \Theta)e(g, C_4)^{-1}e(H(ID_R), C_3)e(H_0(ID_{Snd}), C_4) \\
&= e(H(ID_R), g)^{\gamma\cdot\theta+\eta_1}e(H(ID_R), H_0(ID_{Snd}))^k \\
&\neq e(H(ID_{Rev}), g)^{\gamma\cdot\theta+\eta_1}e(H(ID_{Rev}), H_0(ID_S))^k \\
&= e(H(ID_{Rev}), ek_2)e(H(ID_{Rev}), g^{\eta_1})e(H(ID_{Rev}), ek_1) \\
&= e(H(ID_{Rev}), ek_1 ek_2 g^{\eta_1}) = \mu_2.
\end{aligned}$$

5 Security Proof of Cross-domain IB-ME

In this section, we give the security proof of cross-domain IB-ME. The concrete proof is presented as follows:

Theorem 1. *For two groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order q and an admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, if the* CBDH assumption *holds in $(\mathbb{G}, \mathbb{G}_T, e)$, the cross-domain IB-ME is semantically secure in the random oracle model.*

Lemma 1. If an adversary $\mathcal{A}$ issuing at most q_R, q_{H_1} queries, respectively, to the decryption key oracle $\mathcal{O}_2$ and the random oracle H_1 has the advantage ε to break the privacy property of cross-domain IB-ME, we can construct an algorithm that solves the *CBDH* problem with the advantage $4\varepsilon/e^2(q_R+2)^2q_{H_1}$.

The proven framework is similar to the schemes [1, 26]. In their proof, for simplification, an intermediate public-key encryption scheme **IBasicPub** is introduced. More in detail, the privacy of their scheme can be reduced to the security of **IBasicPub**, and then the security of **IBasicPub** can be reduced to the *CBDH* assumption. As a result, if the *CBDH* assumption holds, the privacy property of the scheme can be guaranteed. Here, we employ the same tactic. Specifically, we define two games indirectly reducing our scheme to the *CBDH* assumption. First, we define a public key encryption scheme **BasicPub**, a variant of **IBasicPub**, which includes the following polynomial-time algorithms

- **Setup**(1^λ). According to the security parameter λ, two cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p and a symmetric bilinear mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ are generated. Then the algorithm selects a random generator g, a random number $\rho \in \mathbb{Z}_p^*$, and calculate $P = g^\rho$. Next, it chooses a hash function simulated as a random oracle $H_1 : \{0,1\}^* \to \{0,1\}^n$. Finally, the master public key is $mpk = (p, \mathbb{G}, \mathbb{G}_T, e, g, n, H_1, P = g^\rho)$ and the master secret key is $msk = \rho$.
- **KGen**(mpk, msk). This algorithm randomly samples $T \in \mathbb{G}$ as the public key $pk = T$ and calculates $sk = T^\rho$ as the private key.
- **Enc**(mpk, pk, M). The algorithm takes the message $M \in \{0,1\}^n$ and the public key $pk = T$ as input, then selects a random number $\varpi \in \mathbb{Z}_p^*$ and outputs the ciphertext $C = (U, V) = (M \oplus H_1(e(pk, P)^\varpi), g^\varpi)$.
- **Dec**(mpk, sk, C). This algorithm takes the ciphertext C and private key sk as input, and then outputs the plaintext $M = U \oplus H_1(e(sk, V))$.

In the first game, we demonstrate that our scheme satisfies privacy if **BasicPub** is *IND-CPA*-secure. Specifically, the security definition of basic scheme **BasicPub** is the same as that defined by Ateniese et al. [1]. That is, in the adversary challenge phase, the adversary not only submits a pair of challenge messages m_0 and m_1 but also inputs a pair of public keys $pk_{j,0}$ and $pk_{j,1}$. This security game can be regarded as a hybrid between the classic IND-CPA experiment and the key-privacy experiment for public-key encryption defined in [27]

Definition 4 (**IND-CPA**$^+$). A public-key encryption scheme $\mathit{\Pi}$ is **IND-CPA**$^+$ semantically secure if

$$|Pr[G_{\mathit{\Pi},\mathcal{A}}^{CPA^+}] - 1/2| \leq negl(\lambda)$$

holds for any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, where $G_{\mathit{\Pi},\mathcal{A}}^{CPA^+}$ is defined as follows: $(mpk, msk) \leftarrow Setup(1^\lambda)$. $(M_0, M_1, pk_{j,0}, pk_{j,1}, \alpha) \leftarrow \mathcal{A}_1^{\mathbf{KGEen}(msk,\cdot)}(1^\lambda, mpk)$. $C \leftarrow Enc(mpk, pk_{j,b}, M_b)$ where $b \in \{0,1\}$. $b' \leftarrow \mathcal{A}_2^{\mathbf{KGEen}(msk,\cdot)}(1^\lambda, C, \alpha)$. If $b = b'$ and $pk_{j,0}, pk_{j,1} \in \mathbb{O}_{KGen}$, the game outputs 1, otherwise outputs 0. In $G_{\mathit{\Pi},\mathcal{A}}^{CPA^+}$, the key generation oracle $\mathcal{O}_{KGen}$ generates pairs (pk, sk), but only outputs the public key pk.

Claim 1. If an adversary $\mathcal{A}$ issuing at most q_R the decryption key oracle $\mathcal{O}_2$ has the advantage ε to break the privacy property of cross-domain IB-ME, we can construct an algorithm $\mathcal{A}'$ that can break **IND-CPA**$^+$ security of **BasicPub** with the advantage $2\varepsilon/e^2(q_R + 2)^2$.

Proof. In the process of proof, $\mathcal{A}'$ is both the challenger of $G_{\mathit{\Pi},\mathcal{A}}^{Priv}(\lambda)$ and the attacker of **BasicPub**. The challenger of **BasicPub** first runs the **Setup** algorithm to obtain the master public key $mpk = (p, \mathbb{G}, \mathbb{G}_T, e, g, n, H_1, P = g^\rho)$, and then sends mpk to $\mathcal{A}'$. The master secret key of **BasicPub** is ρ, which is unknown to $\mathcal{A}'$. Then, $\mathcal{A}$ and $\mathcal{A}'$ interact as follows:

- **Setup:** $\mathcal{A}'$ randomly selects $\gamma, k, \theta \in \mathbb{Z}_p^*$, and then computes $\Upsilon = g^\gamma$, $\Theta = g^\theta$ and $K = g^k$. Next, $\mathcal{A}'$ sends the above public parameter mpk, Υ, Θ, K and two hash functions H and H_0 modeled as random oracles under its control to $\mathcal{A}$.
  - H **queries:** $\mathcal{A}'$ respond to the query as follows:
    - ∗ If the tuple $(ID_i, Q_i, \beta_i, d_i) \in \mathcal{L}_1$ corresponding to query ID_i already exists, Q_i is returned. Otherwise, a random coin $d_i \in \{0, 1\}$ is generated so that $Pr[d_i = 0] = \delta$.
    - ∗ Then, $\beta_i \in \mathbb{Z}_p^*$ is randomly selected. If $d_i = 0$, then $Q_i = g^{\beta_i}$ is calculated, and the tuple $(ID_i, Q_i, \beta_i, 0)$ is stored in $\mathcal{L}_1$. Otherwise, the public key generation oracle of **BasicPub** is called to obtain pk_i, set $Q_i = pk_i$, and then the tuple $(ID_i, Q_i, \beta_i, 1)$ is added to $\mathcal{L}_1$.
    - ∗ Return Q_i.
  - H_0 **queries:** $\mathcal{A}'$ keeps a list $\mathcal{L}_2$ of the history of calls to H_0 in which each tuple is (ID_i, Z_i). If the query ID_i already exists, Z_i is returned. Otherwise, a random number $Z_i \in \mathbb{G}$ is selected, (ID_i, Z_i) is added to $\mathcal{L}_2$, and then Z_i is returned.
- **Phase 1**: $\mathcal{A}$ can issue the polynomial queries for the encryption key and decryption key as follows:
  - **SKGen queries:** For the input ID_i of the encryption key oracle $\mathcal{O}_1$, $\mathcal{A}'$ obtains $H_0(ID_i) = Z_i$ from the list $\mathcal{L}_2$, and then returns $ek_{ID_i} = (Z_i^k, \Upsilon^\theta)$.
  - **RKGen queries:** For the input ID_i of the decryption key oracle $\mathcal{O}_2$, $\mathcal{A}'$ obtains the tuple $(ID_i, Q_i, \beta_i, d_i)$ according to $H(ID_i) = Q_i$ from the list $\mathcal{L}_1$. If $d_i = 1$, $\mathcal{A}'$ aborts, otherwise returns $dk_{ID_i} = (P^{\beta_i}, Q_i^\gamma)$, where we have $H(ID_i) = Q_i$ and $dk_1 = (g^\rho)^{\beta_i} = Q_i^\rho$ since $P = g^\rho$.
- **Challenge:** After the first query phase, $\mathcal{A}$ submits $(M_0, M_1, ID_{Rev_0}, ID_{Rev_1}, ID_{S_0}, ID_{S_1})$ to $\mathcal{A}'$. Then $\mathcal{A}'$ performs the following steps:
  - $\mathcal{A}'$ obtain the tuples $(ID_{Rev_0}, Q_0, \beta_0, d_0)$ and $(ID_{Rev_1}, Q_1, \beta_1, d_1)$ according to queries $H(ID_{Rev_0}) = Q_0$ and $H(ID_{Rev_1}) = Q_1$ from $\mathcal{L}_1$. If $d_0 \neq 1$ or $d_1 \neq 1$ in both tuples, $\mathcal{A}'$ terminates the process. Otherwise, we know $d_0 = 1$ and $d_1 = 1$, and $Q_0 = pk_0$ and $Q_1 = pk_1$.
  - $\mathcal{A}'$ obtain the tuples $H_0(ID_{S_0}) = Z_0$ and $H_0(ID_{S_1}) = Z_1$ according to queries ID_{S_0} and ID_{S_1}. Then, $\mathcal{A}'$ selects a random number $\sigma_0, \sigma_1 \in \mathbb{Z}_p^*$ and calculates $M_0^* = M_0 \oplus H_1(e(Q_0, Z_0^k g^{\gamma \cdot \theta + \sigma_0}))$ and $M_1^* = M_1 \oplus H_1(e(Q_1, Z_1^k g^{\gamma \cdot \theta + \sigma_1}))$. Note that $ek_{ID_{S_i}} = (Z_i^k, \Upsilon^\theta)$.
  - $\mathcal{A}'$ sends (M_0^*, M_1^*, Q_0, Q_1) to the challenger of **BasicPub** and gets $C = (U, V)$ as response.
  - $\mathcal{A}'$ computes $C_{3,0} = Z_0^{k+\sigma_0}$, $C_{3,1} = Z_1^{k+\sigma_1}$ and $C_{4,0} = Q_0^{-\sigma_0}$, $C_{4,1} = Q_1^{-\sigma_1}$. Then, $\mathcal{A}'$ randomly returns $C_0' = (U, V, C_{3,0}, C_{4,0})$ or $C_1' = (U, V, C_{3,1}, C_{4,1})$ to $\mathcal{A}$.
- **Second query phase:** The query at this stage is the same as the first stage.
- **Guess:** $\mathcal{A}'$ outputs the same guess as $\mathcal{A}$.

Probability Analysis. Suppose $\mathcal{A}$ issues q_R queries to the oracle $\mathcal{O}_2$, and the probability that $\mathcal{A}'$'s query is not terminated is δ^{q_R}. Similarly, in the challenge phase, the probability that $\mathcal{A}'$'s query does not abort is $(1-\delta)^2$. For the probability $\delta^{q_R}(1-\delta)^2$ that does not abort, the maximum value of this probability is at $\delta_{opt} = q_R/(q_R+2)$. If we leverage δ_{opt} as the probability that $\mathcal{A}'$'s query does not terminate to obtain coins $d_i = 0$ in H query, the probability that $\mathcal{A}'$'s query does not abort is at least $4/e^2(q_R+2)^2$. In the challenge phase, after obtaining the challenge ciphertext $C = (U, V)$ from the challenger of **BasicPub**, $\mathcal{A}'$ will randomly return C'_0 or C'_1 to $\mathcal{A}$. As a result, $\mathcal{A}$ will lose 1/2 of its advantage against the **BasicPub**.

Claim 2. If an adversary $\mathcal{A}$ issuing at most q_{H_1} queries to the random oracle H_1 has the advantage ε against **IND-CPA$^+$** security of **BasicPub**, we can construct an algorithm against the *CBDH* problem with the advantage $2\varepsilon/q_{H_1}$.

Proof. $\mathcal{A}'$ obtains a *CBDH* tuple (g, g^a, g^b, g^c) from the challenger of **BasicPub**, whose correct solution is $D = e(g,g)^{abc}$. During **Setup** phase, $\mathcal{A}'$ sets $P = g^a$. The master secret key of **BasicPub** is $msk = a$, which is unknown to $\mathcal{A}'$. Then, $\mathcal{A}$ and $\mathcal{A}'$ interact as follows:

- **KGen queries:** $\mathcal{A}'$ randomly selects $x_i \in \mathbb{Z}_p^*$, then calculates $pk_i = (g^b)^{x_i}$ and returns pk_i to $\mathcal{A}$. Note that the private key corresponding to pk_i is $sk_i = g^{abx_i}$, which is unknown to $\mathcal{A}'$.
  - H_1 **oracle:** $\mathcal{A}'$ keeps a list $\mathcal{L}$ of the history of calls to H_1 in which each tuple is (X_i, h_i). If the query X_i already exists, h_i is returned. Otherwise, a random bit string $h_i \in \{0,1\}^n$ is selected, (X_i, h_i) is added to $\mathcal{L}$, and then h_i is returned.
- **Challenge:** $\mathcal{A}$ sends $(M_0, M_1, pk_{j,0}, pk_{j,1})$ to $\mathcal{A}'$. Then a random string $Z \in \{0,1\}^n$ is selected by $\mathcal{A}'$ and the challenge ciphertext is set by $C = (Z, g^c)$. Next, $\mathcal{A}'$ returns C to $\mathcal{A}$. Note that the plaintext hidden by C is $Z \oplus H_1(e(g^c, sk_{j,b}))$ for some $b \in \{0,1\}$, which can also be expressed as $Z \oplus H_1(D^{x_{j,b}})$, where $x_{j,b}$ is the secret key corresponding to $pk_{j,b}$.
- **Guess:** $\mathcal{A}$ sends the guess $b' \in \{0,1\}$ to $\mathcal{A}'$. Then $\mathcal{A}'$ sets $z = 1/x_{j,b'}$, then it selects a random tuple (X_i, h_i) from $\mathcal{L}$, and then outputs $D = X_i^z$ as the solution of the received *CBDH* instance.

Probability Analysis. The probability that $\mathcal{A}'$ outputs the correct solution is at least $2\varepsilon/q_{H_1}$ if $\mathcal{A}$ breaks **IND-CPA$^+$** security of **BasicPub** with the advantage ε. This analysis is similar to that in [26]. A detailed analysis can be found in [26].

Proof of Lemma 1. By combining **Claim 1** and **Claim 2**, we can conclude that if there is an adversary who can break the privacy property of cross-domain IB-ME with advantage ε, there will be an algorithm solving the *CBDH* problem with advantage $4\varepsilon/e^2(q_R+2)^2 q_{H_1}$.

Lemma 2. If an adversary $\mathcal{A}$ issuing at most q_R queries to the decryption key oracle $\mathcal{O}_2$, q_S queries to the encryption key oracle $\mathcal{O}_1$, and q_{H_1} queries to the random oracle H_1, has the advantage ε to break the authenticity property of cross-domain IB-ME, we can construct an algorithm that can solve the *CBDH* problem with the advantage $8\varepsilon/e^2(q_R+q_S+2)^2 q_{H_1}$.

Proof. $\mathcal{A}'$ obtains a *CBDH* tuple (g, g^a, g^b, g^c) from the challenger of $\textbf{BasicPub}^+$, whose correct solution is $D = e(g,g)^{abc}$. $\mathcal{A}'$ randomly selects random numbers $\rho, \gamma, \theta \in \mathbb{Z}_p^*$ and takes $(\rho, \gamma, \theta, c, H_1)$ as the master secret key of the cross-domain IB-ME, where c is unknown to $\mathcal{A}'$. Then, $\mathcal{A}$ and $\mathcal{A}'$ interact as follows:

- **Setup:** $\mathcal{A}'$ sends to $\mathcal{A}$ $(p, \mathbb{G}, \mathbb{G}_T, e, g, n, H, H_0, H_1, P = g^\rho, \Upsilon = g^\gamma, K = g^c, \Theta = g^\theta)$, where H, H' and H_1 are modeled as random oracles under its control.
  - H **queries:** $\mathcal{A}'$ respond to the query as follows:
    - If the tuple $(ID_i, Q_i, \beta_i, d_i) \in \mathcal{L}_1$ corresponding to query ID_i already exists, Q_i is returned. Otherwise, a random coin $d_i \in \{0,1\}$ is generated so that $Pr[d_i = 0] = \delta$.
    - Then, $\beta_i \in \mathbb{Z}_p^*$ is randomly selected. If $d_i = 0$, then $Q_i = g^{\beta_i}$ is calculated, and the tuple $(ID_i, Q_i, \beta_i, 0)$ is stored in $\mathcal{L}_1$. Otherwise, set $Q_i = g^{a\beta_i}$, and then the tuple $(ID_i, Q_i, \beta_i, 1)$ is added to $\mathcal{L}_1$.
    - Return Q_i to $\mathcal{A}$.
  - H_0 **queries:** $\mathcal{A}'$ respond to the query as follows:
    - If the tuple $(ID_i, Q_i, \beta_i, d_i) \in \mathcal{L}_2$ corresponding to query ID_i already exists, Q_i is returned. Otherwise, a random coin $d_i \in \{0,1\}$ is generated so that $Pr[d_i] = \delta$.
    - Then, $\beta_i \in \mathbb{Z}_p^*$ is randomly selected. If $d_i = 0$, then $Q_i = g^{\beta_i}$ is calculated, and the tuple $(ID_i, Q_i, \beta_i, 0)$ is stored in $\mathcal{L}_2$. Otherwise, set $Q_i = g^{b\beta_i}$, and then the tuple $(ID_i, Q_i, \beta_i, 1)$ is added to $\mathcal{L}_2$.
    - Return Q_i to $\mathcal{A}$.
  - H_1 **queries:** $\mathcal{A}'$ keeps a list $\mathcal{L}$ of the history of calls to H_1 in which each tuple is (X_i, h_i). If the query X_i already exists, return h_i. Otherwise, a random bit string $h_i \in \{0,1\}^n$ is selected, (X_i, h_i) is added to $\mathcal{L}$, and then h_i is returned.
- **Phase 1**: $\mathcal{A}$ can issue the polynomial queries for the encryption key and decryption key as follows:
  - **SKGen queries:** For the input ID_i of the encryption key oracle $\mathcal{O}_1$, $\mathcal{A}'$ obtains the tuple $(ID_i, Q_i, \beta_i, d_i)$ according to $H_0(ID_i) = Q_i$ from the list $\mathcal{L}_2$. If $d_i = 1$, $\mathcal{A}'$ terminates the query, otherwise returns $ek_{ID_i} = (K^{\beta_i}, \Upsilon^\theta)$.
  - **RKGen queries:** For the input ID_i of the decryption key oracle $\mathcal{O}_2$, $\mathcal{A}'$ obtains the tuple $(ID_i, Q_i, \beta_i, d_i)$ according to $H(ID_i) = Q_i$ from the list $\mathcal{L}_1$. If $d_i = 1$, $\mathcal{A}'$ terminates the query, otherwise returns $dk_{ID_i} = (P^{\beta_i}, \Upsilon^{\beta_i})$, where $H(ID_i) = Q_i = g^{\beta_i}$.
- **Forgery:** At the moment, $\mathcal{A}$ submits (C, ID_R, ID_{Snd}) to $\mathcal{A}'$, where $C = (C_1, C_2, C_3, C_4)$. Then $\mathcal{A}'$ performs the following steps:
  - $\mathcal{A}'$ obtain the tuples $(ID_R, Q, \beta, d) \in \mathcal{L}_1$ and $(ID_{Snd}, Q', \beta', d') \in \mathcal{L}_2$ according to queries $H(ID_R) = Q$ and $H_0(ID_{Snd}) = Q'$ from $\mathcal{L}_1$ and $\mathcal{L}_2$. If

$d \neq 1$ or $d' \neq 1$ in both tuples, $\mathcal{A}'$ terminates the process. Otherwise, we know $H(ID_R) = g^{a\beta}$, $dk_2 = g^{\gamma \cdot a\beta}$, $H_0(ID_{Snd}) = g^{b\beta'}$ and $ek_1 = g^{bc\beta'}$. Hence, $H_1(\mu_2') = H_1(e(dk_2, \Theta)e(g, C_4)^{-1}e(H(ID_R), C_3)e(H_0(ID_{Snd}), C_4))$, where $e(H(ID_R), C_3)e(H_0(ID_{Snd}), C_4) = e(H(ID_R), ek_1) = e(g^{a\beta}, g^{bc\beta'})$ $= e(g, g)^{abc\beta\beta'} = D^{\beta\beta'}$.

- $\mathcal{A}'$ sets $z = 1/(\beta\beta')$, then it randomly selects a random tuple (X_i, h_i) from $\mathcal{L}$, and then outputs $D' = (X_i \cdot e(g, C_4)e(dk_2, \Theta)^{-1})^z$ as the solution of the received $CBDH$ instance.

Probability Analysis. We require the challenge ciphertext (C, ID_R, ID_{Snd}) to meet $ID_R \notin \mathbb{O}_2$ and $\forall\ ID_S \in \mathbb{O}_1,\ ID_S \neq ID_{Snd}$, which can perfectly simulate the authentication game. Suppose $\mathcal{A}$ issues q_R queries to the oracle $\mathcal{O}_2$ and q_S queries to the oracle $\mathcal{O}_1$, and the probability that $\mathcal{A}$'s query will not be terminated is $\delta^{q_R+q_S}$. Similarly, in the challenge phase, the probability that $\mathcal{A}$'s query does not abort is $(1-\delta)^2$. For the probability $\delta^{q_R+q_S}(1-\delta)^2$ that does not abort, the maximum value of this probability is at $\delta_{opt} = (q_R+q_S)/(q_R+q_S+2)$. If we use δ_{opt} as the probability that $\mathcal{A}$'s query does not abort to obtain coins $d_i = 0$ in H query and H_0 query, the probability that $\mathcal{A}'$s query does not abort is at least $4/e^2(q_R+2)^2$. If $\mathcal{A}$'s query is not terminated, the probability that the correct solution is outputted by $\mathcal{A}'$ is at least $2\varepsilon/q_{H_1}$. As a result, the $CBDH$ problem can be solved by $\mathcal{A}'$ with advantage $8\varepsilon/e^2(q_R + q_S + 2)^2 q_{H_1}$.

By setting $\varepsilon \geq 1/poly(\lambda)$, $q_R = poly(\lambda)$, $q_S = poly(\lambda)$ and $q_{H_1} = poly(\lambda)$ in **Lemma 1** and **Lemma 2**, we can conclude that the proposed cross-domain IB-ME is semantically secure.

6 Cross-domain Multi-receiver IB-ME

6.1 Syntax of Cross-Domain Multi-receiver IB-ME.

The syntax of cross-domain multi-receiver IB-ME is same as cross-domain IB-ME, except that a receivers' identity set rather than a single receiver is specified during encryption and the receiver's identity must belong to the identity set specified by the sender when decrypting.

A cross-domain multi-receiver IB-ME with message space $\mathcal{M}$ is correct if $\forall\ \lambda \in \mathbb{N}$, $\forall\ ID_S, ID_R, ID_{Snd}, ID_{Rev_i} \in \{0,1\}^*$, when $pp \leftarrow$ **Initialization**(1^λ), $(mpk_R, msk_R) \leftarrow$ **RASetup**(pp), $msk_S \leftarrow$ **SASetup**(pp), $ek_{ID_S} \leftarrow$ **EKGen**(mpk_R, msk_S, ID_S), $dk_{ID_{Rev_i}} \leftarrow$ **DKGen**(msk_R, ID_{Rev_i}):

$$Pr[\mathbf{Dec}(mpk_S, dk_{ID_{Rev_i}}, ID_{Snd}, \mathbf{Enc}(mpk_S, mpk_R, ek_{ID_S}, ID_{Rev}, M)) = M] \geq 1 - negl(\lambda),$$

whenever $ID_S = ID_{Snd}$ and $ID_{Rev_i} \in ID_{Rev}$.

6.2 Security of Cross-Domain Multi-receiver IB-ME.

Cross-domain multi-receiver IB-ME also potentially inherits the security properties of IB-ME (i.e., privacy and authenticity).

Privacy of Cross-Domain Multi-receiver IB-ME. In the privacy game $G^{Priv}_{\Pi,\mathcal{A}}(\lambda)$ of **Table** 2, $ID^*_{Rev_0}$ and $ID^*_{Rev_1}$ represent two distinct receivers' identity set. After accessing the oracles $\mathcal{O}_1$ and $\mathcal{O}_2$, $\mathcal{A}_1$ outputs $(M_0, M_1, ID^*_{Rev_0}, ID^*_{Rev_1}, ID_{S_0}, ID^*_{S_1}, \alpha)$ with the restriction $ID_i \notin (ID^*_{Rev_0} \cup ID^*_{Rev_1})$ for $ID_i \in \mathbb{O}_2$, where $\mathbb{O}_2$ records the identity set accessed by $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. In the next phase, $\mathcal{A}_2$ cannot issue the decryption key on $ID_i \in (ID^*_{Rev_0} \cup ID^*_{Rev_1})$.

Table 2. Security Games of Cross-Domain Multi-receiver IB-ME

$G^{Priv}_{\Pi,\mathcal{A}}(\lambda)$	$G^{Auth}_{\Pi,\mathcal{A}}(\lambda)$
$pp \leftarrow$ **Initialization**(1^λ)	
$(mpk_R, msk_R) \leftarrow$ **RASetup**(pp)	$pp \leftarrow$ **Initialization**(1^λ)
$(mpk_S, msk_S) \leftarrow$ **SASetup**(pp)	$(mpk_R, msk_R) \leftarrow$ **RASetup**(pp)
$(M_0, M_1, ID^*_{Rev_0}, ID^*_{Rev_1}, ID_{S_0}, ID_{S_1}, \alpha)$	$(mpk_S, msk_S) \leftarrow$ **SASetup**(pp)
$\leftarrow \mathcal{A}_1^{\mathcal{O}_1,\mathcal{O}_2}(pp, mpk_R, mpk_S)$	$(C, ID_{Rev}, ID_{Snd}) \leftarrow \mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2}(pp, mpk_R, mpk_S)$
$b \leftarrow \{0,1\}$	$dk_{ID_{Rev_i}} \leftarrow$ **DKGen**(msk_R, ID_{Rev_i})
$ek_{ID_{S_b}} \leftarrow$ **EKGen**(mpk_R, msk_S, ID_{S_b})	$M \leftarrow$ **Dec**$(mpk_S, dk_{ID_{Rev_i}}, ID_{Snd}, C)$
$C \leftarrow$ **Enc**$(mpk_S, mpk_R, ek_{ID_{S_b}}, ID^*_{Rev_b}, M_b)$	$If \forall ID_S \in \mathbb{O}_1, ID_{Rev_i} \in ID_{Rev} : (ID_S \neq ID_{Snd}) \wedge$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}_1,\mathcal{O}_2}(pp, C, \alpha)$	$(ID_{Rev_i} \notin \mathbb{O}_2) \wedge (M \neq \perp)$ **return** 1 *Else* **return** 0
$If\ (b = b')$ **return** 1 *Else* **return** 0	

Definition 5 (Privacy of Cross-Domain Multi-receiver IB-ME). We say that a cross-domain multi-receiver IB-ME provides privacy if for any valid PPT adversary $\mathcal{A}$:

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

where $\mathcal{A}$ is valid if $\forall\ ID_R \in \mathbb{O}_2$, it satisfies $ID_R \notin (ID^*_{Rev_0} \cup ID^*_{Rev_1})$.

Authenticity of Cross-Domain Multi-receiver IB-ME. In the authenticity game $G^{Auth}_{\Pi,\mathcal{A}}(\lambda)$ in **Table** 2, after the adversary $\mathcal{A}$ accesses the random oracles $\mathcal{O}_1$ and $\mathcal{O}_2$, $\mathcal{A}$ outputs a forgery ciphertext (C, ID_{Rev}, ID_{Snd}) so that $Dec(dk_{ID_{Rev_i}}, ID_{Snd}, C) \neq \perp$ for any $ID_{Rev_i} \in ID_{Rev}$, where $\forall\ ID_S \in \mathbb{O}_1, ID_S \neq ID_{Snd}$ and $\forall\ ID_R \in \mathbb{O}_2, ID_R \notin ID_{Rev}$, which is considered valid.

Definition 6 (Authenticity of Cross-Domain Multi-receiver IB-ME). We say that a cross-domain multi-receiver IB-ME provides authenticity if for any valid PPT adversary $\mathcal{A}$:

$$Pr[G^{Auth}_{\Pi,\mathcal{A}}(\lambda) = 1] \leq negl(\lambda).$$

Definition 7 (Secure Cross-Domain Multi-receiver IB-ME). We say that a cross-domain multi-receiver IB-ME Π is semantically secure if Π meets the privacy (*Def. 5*) and authenticity (*Def. 6*).

6.3 Construction of Cross-Domain Multi-receiver IB-ME

The construction of cross-domain multi-receiver IB-ME has many of the same processes as cross-domain IB-ME. During **Initialization**, this process is the same as that of cross-domain IB-ME except that the public parameter additionally contains other cryptographic hash function $H_2 : \mathbb{G}_T \to \mathbb{Z}_p^*$. The processes of **Receiver-authority Setup**, **Sender-authority Setup**, **Encryption-key Generation** and **Decryption-key Generation** are the same as that of cross-domain IB-ME. Different **Encryption** and **Decryption** processes are as follows:

- **Encryption.** Once mpk_R, mpk_S, ek_{ID_S}, the target identity set $ID_{Rev} = \{ID_{Rev_1}, ID_{Rev_2}, ..., ID_{Rev_d}\}$ and the message $M \in \{0,1\}^n$ are input, the sender randomly picks $\eta_0, \eta_1, z, v \in \mathbb{Z}_p^*$ and computes

$$\mu_{Rev_i} = e(H(ID_{Rev_i}), P)^{\eta_0}, \hat{\mu}_{Rev_i} = e(H(ID_{Rev_i}), ek_1 ek_2 g^{\eta_1}),$$

  $K_{ID_{Rev_i}} = H_2(\mu_{Rev_i})$ and $V_{ID_{Rev_i}} = H_2(\hat{\mu}_{Rev_i})$ for $ID_{Rev_i} \in ID_{Rev}$. Then, he also computes

$$f(x) = \Pi_{i=1}^{d}(x - K_{ID_{Rev_i}}) + z = \sum_{i=0}^{d} a_i x^i \ (mod\ p)$$

  and

$$J(x) = \Pi_{i=1}^{d}(x - V_{ID_{Rev_i}}) + v = \sum_{i=0}^{d} b_i x^i \ (mod\ p),$$

  where a_i and b_i are coefficients of polynomials. Next, he computes

$$C_1 = M \oplus H_1(z||g^{\eta_0}) \oplus H_1(v||g^{\eta_1}), C_2 = g^{\eta_0}, C_3 = ek_1 H_0(ID_S)^{\eta_1},$$

  $C_{4,i} = H(ID_{Rev_i})^{-\eta_1}$ for $ID_{Rev_i} \in ID_{Rev}$ and $C_5 = g^{\eta_1}$. Finally, the ciphertext is $C = (C_1, C_2, C_3, \{C_{4,i}\}_{i\in[1,d]}, \{a_i, b_i\}_{i\in[0,d]}\}, C_5)$.
- **Decryption.** After inputting the ciphertext $C = (C_1, C_2, C_3, \{C_{4,i}\}_{i\in[1,d]}, \{a_i, b_i\}_{i\in[0,d]}\}, C_5)$ from ID_{Snd} and the decrypting key $dk_{ID_{Rev_i}} = (dk_{i,1}, dk_{i,2})$ corresponding to the receiver identity ID_{Rev_i}, the receiver computes $\mu'_{Rev_i} = e(C_2, dk_{i,1})$,

$$\hat{\mu}'_{Rev_i} = e(dk_{i,2}, \Theta)e(H(ID_{Rev_i}), C_5)e(H(ID_{Rev_i}), C_3)e(H_0(ID_{Snd}), C_{4,i}).$$

  Then, he also computes $K_{ID_{Rev_i}} = H_2(\mu'_{Rev_i})$ and $V_{ID_{Rev_i}} = H_2(\hat{\mu}'_{Rev_i})$. If the sender identity $ID_{Snd} = ID_S$ and the receiver identity $ID_{Rev_i} \in ID_{Rev}$, there will be a $C_{4,i}$ so that the receiver can calculate z, v. Specifically, he computes

$$z = \sum_{i=1}^{d} a_i(K_{ID_{R_i}})^i \ (mod\ p), v = \sum_{i=1}^{d} b_i(V_{ID_{R_i}})^i \ (mod\ p).$$

  Finally, the receiver can obtain the plaintext $M = C_1 \oplus H_1(z||C_2) \oplus H_1(v||C_5)$.

Note that: To enable the receiver to determine the correct message, the plaintext message can be encoded (e.g., the hash value of a message can be concatenated after the message to determine the correct plaintext). Besides, the decryption process can be speeded by anonymous hint technology [28].

6.4 Proof of Cross-Domain Multi-receiver IB-ME

Theorem 2. *For two groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order q and an admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, if the* CBDH *assumption holds in $(\mathbb{G}, \mathbb{G}_T, e)$, the cross-domain multi-receiver IB-ME is semantically secure in the random oracle model.*

Lemma 3. If an adversary $\mathcal{A}$ issuing at most q_R, q_{H_2} queries, respectively, to the decryption key oracle $\mathcal{O}_2$ and the random oracle H_2 has the advantage ε to break the privacy property of cross-domain multi-receiver IB-ME, we can construct an algorithm that solves the *CBDH* problem with the advantage $\varepsilon/(q_{H_2} \cdot e^{2d}(q_R/2d)^{2d})$, where d is the number of receivers.

The proof framework is similar to the cross-domain IB-ME. Its security reduces the privacy of cross-domain multi-receiver IB-ME to the privacy of an intermediate multi-receiver public-key encryption **BasicPub**$^+$, which is proved to be semantically secure in the random oracle model on *CBDH* assumption. Specifically, if there exists an adversary $\mathcal{A}$ who can break privacy of the proposed cross-domain multi-receiver IB-ME, there will be an adversary $\mathcal{A}'$ to break *CBDH* assumption.

Similarly, before the proof, we define a multi-receiver public-key encryption scheme, which is familiar with the public-key encryption scheme proved in cross-domain IB-ME. Specifically, **Setup** algorithm is similar except that a hash function $H_2 : G_T \to \mathbb{Z}_p^*$ is also added. **KGen** algorithm is the same. Different **Encryption** and **Decryption** algorithms are presented as follows:

- **Enc**(mpk, PK, M). The algorithm takes the message $M \in \{0,1\}^n$ and the public-key set PK as input, then selects a random number $\varpi, z \in \mathbb{Z}_p^*$ and computes $l_i = H_2(e(pk_i, P)^{\varpi})$ for $pk_i \in PK$, where $d = |PK|$. Then, it also computes $f(x) = \Pi_{i=1}^{d}(x - l_i) + z = \Sigma_{i=0}^{d} a_i x^i$, $U = M \oplus H_1(z||g^{\varpi})$ and $V = g^{\varpi}$. Finally, it outputs the ciphertext $C = (U, V, \{a_i\}_{i\in[0,d]})$.
- **Dec**(mpk, sk, C). This algorithm takes the ciphertext C and private key sk_i corresponding to $pk_i \in PK$ as input, then computes $l' = H_2(e(sk_i, V))$ and $z = \Sigma_{i=0}^{d} a_i (l')^i$. Finally, it outputs the plaintext $M = U \oplus H_1(z||V)$.

The security definition of **BasicPub**$^+$ is similar to **BasicPub** presented in cross-domain IB-ME. After the key query, $\mathcal{A}$ outputs $(M_0, M_1, PK_0, PK_1, \alpha)$ instead of $(M_0, M_1, pk_{j,0}, pk_{j,1}, \alpha)$, where the public key pk_i in PK_0 and PK_1 is generated by the keg generation oracle $\mathcal{O}_{KGen}$.

Claim 3. If an adversary $\mathcal{A}$ issuing at most q_R the decryption key oracle $\mathcal{O}_2$ has the advantage ε to break the privacy property of cross-domain multi-receiver IB-ME, we can construct an algorithm $\mathcal{A}'$ that can break *IND-CPA*$^+$ security of **BasicPub**$^+$ with the advantage $\varepsilon/(2 \cdot e^{2d}(q_R/2d)^{2d})$, where d is the number of receivers.

Proof. In the process of proof, $\mathcal{A}'$ is not only the challenger of privacy game of the proposed cross-domain multi-receiver IB-ME, but also the attacker of privacy game of **BasicPub**$^+$. After $\mathcal{A}'$ obtains the master public key $mpk = (p, \mathbb{G}, \mathbb{G}_T, e, g, n, H_1, H_2, P)$ of **BasicPub**$^+$. The interaction among $\mathcal{A}$, $\mathcal{A}'$ and the challenger $\mathcal{C}$ of **BasicPub**$^+$ is as follows:

- **Setup:** This process is similar to that of cross-domain IB-ME.
- **Phase 1**: $\mathcal{A}$ can issue the polynomial queries for the encryption key and decryption key. The response process is similar to that of cross-domain IB-ME.
- **Challenge:** At this moment, $\mathcal{A}$ submits $(M_0, M_1, ID^*_{Rev_0}, ID^*_{Rev_1}, ID_{S_0}, ID_{S_1})$ to $\mathcal{A}'$, where $ID^*_{Rev_0} = \{ID^*_{0,1}, ID^*_{0,2}, ..., ID^*_{0,d}\}$ and $ID^*_{Rev_1} = \{ID^*_{1,1}, ID^*_{1,2}, ..., ID^*_{1,d}\}$. For the two distinct receivers' identity sets $ID^*_{Rev_0}$ and $ID^*_{Rev_1}$, $\mathcal{A}'$ computes $PK_0 = \{H(ID^*_{0,i})\}_{i\in[1,d]}$ for $ID^*_{0,i} \in ID^*_{Rev_0}$ and $PK_1 = \{H(ID^*_{1,i})\}_{i\in[1,d]}$ for $ID^*_{1,i} \in ID^*_{Rev_1}$. For $ID_i \in ID^*_{Rev_0}$ and $ID_i \in ID^*_{Rev_1}$, $\mathcal{A}'$ looks for the tuple $(ID_i, Q_i, \beta_i, d_i)$, where $H(ID_i) = Q_i$. If $d_i = 0$, $\mathcal{A}'$ aborts. Otherwise, $\mathcal{A}'$ performs the following process:
  - $\mathcal{A}'$ obtain the tuples $H_0(ID_{S_0}) = Z_0$ and $H_0(ID_{S_1}) = Z_1$ according to queries ID_{S_0} and ID_{S_1}. Then, $\mathcal{A}'$ selects a random number $\sigma_0, \sigma_1, v \in \mathbb{Z}^*_p$ and calculates $V_{0,i} = H_2(e(Q_i, ek_1 ek_2 g^{\sigma_0}))$ for $ID_i \in ID^*_{Rev_0}$ and $V_{1,i} = H_2(e(Q_i, ek_1 ek_2 g^{\sigma_1}))$ for $ID_i \in ID^*_{Rev_1}$. Next, $\mathcal{A}'$ computes $C_{3,0} = ek_1 Z_0^{\sigma_0}$ and $C_{3,1} = ek_1 Z_1^{\sigma_1}$. After that, $\mathcal{A}'$ computes $J_0(x) = \Pi^d_{i=1}(x - V_{0,i}) + v = \sum^d_{i=0} b_{0,i} x^i \pmod p$ and $J_1(x) = \Pi^d_{i=1}(x - V_{1,i}) + v = \sum^d_{i=0} b_{1,i} x^i \pmod p$. Next, $\mathcal{A}'$ computes $C_{4,0,i} = e(H(ID_i), ek_1 \cdot ek_2 g^{\sigma_0})$ for $ID_i \in ID^*_{Rev_0}$ and $C_{4,1,i} = e(H(ID_i), ek_1 \cdot ek_2 g^{\sigma_1})$ for $ID_i \in ID^*_{Rev_1}$. And, $\mathcal{A}'$ also computes $C_{5,0} = g^{\sigma_0}$ and $C_{5,1} = g^{\sigma_1}$. Note that $ek_{ID_{S_i}} = (Z_i^k, \Upsilon^\theta)$.
  - $\mathcal{A}'$ computes $M^*_0 = M_0 \oplus H_1(v||g^{\sigma_0})$ and $M^*_1 = M_1 \oplus H_1(v||g^{\sigma_1})$. Then, $\mathcal{A}'$ will forward $(M^*_0, M^*_1, PK_0, PK_1)$ to the challenger $\mathcal{C}$. Then $\mathcal{C}$ randomly selects a bit $b \in \{0,1\}$ to generate the ciphertext $C' = Enc(mpk, PK_b, M_b)$ and returns $C' = (U', V', \{a_i\}_{i\in[0,d]})$ to $\mathcal{A}'$. In addition, $\mathcal{C}$ also returns $K_{b,i} = H_2(e(PK_{b,i}, P)^{\eta_0})$ and $C_2 = g^{\eta_0}$, where η_0 are the random numbers selected in the encryption process. Since H_2 is a hash function, it can be seen from the property of the hash function that $K_{b,i}$ will not disclose the input of the hash function, that is, $e(pk_{b,i}, P)^{\eta_0}$.
  - Finally, $\mathcal{A}'$ randomly returns ciphertext $C = (U', V', C_{3,0}, \{C_{4,0,i}\}_{i\in[1,d]}, \{a_i, b_{0,i}\}_{i\in[0,d]}, C_{5,0})$ or $C = (U', V', C_{3,1}, \{C_{4,1,i}\}_{i\in[1,d]}, \{a_i, b_{1,i}\}_{i\in[0,d]}, C_{5,1})$ to $\mathcal{A}$.
- **Second query phase:** $\mathcal{A}$ can continue to issue queries, but $\mathcal{A}$ cannot issue the decryption key on identity $ID_i \in (ID^*_{Rev_0} \cup ID^*_{Rev_1})$.
- **Guess:** Finally, $\mathcal{A}$ outputs a bit b'. $\mathcal{A}'$ outputs the same bit. If $b' = b$, then 1 is output; otherwise, 0 is output.

Probability Analysis. Suppose $\mathcal{A}$ issues q_R queries to the oracle $\mathcal{O}_2$, and the probability that $\mathcal{A}'$'s query will not be terminated is δ^{q_R}. Similarly, in the challenge phase, the probability that $\mathcal{A}'$'s query does not abort is $(1-\delta)^{2d}$. For the probability $\delta^{q_R}(1-\delta)^{2d}$ that does not abort, the maximum value of this probability is at $\delta_{opt} = q_R/(q_R + 2d)$. If we use δ_{opt} as the probability that $\mathcal{A}'$'s query does not abort to obtain coins $d_i = 0$ in H query, the probability that $\mathcal{A}'$'s query does not abort is at least $1/e^{2d}(q_R/2d)^{2d}$. In the challenge phase, after obtaining the challenge ciphertext $C' = (U', V', \{a_i\}_{i\in[0,d]})$ from the challenger of **BasicPub**$^+$, $\mathcal{A}'$ will randomly return C'_0 or C'_1 to $\mathcal{A}$. As a result, $\mathcal{A}$ will lose 1/2 of its advantage against the **BasicPub**$^+$.

Claim 4. If an adversary $\mathcal{A}$ issuing at most q_{H_2} queries to the random oracle H_2 has the advantage ε against **IND-CPA**$^+$ security of **BasicPub**$^+$, we can construct an algorithm against the *CBDH* problem with the advantage $2\varepsilon/q_{H_2}$.

Proof. $\mathcal{A}'$ obtains a *CBDH* tuple (g, g^a, g^b, g^c) from the challenger of **BasicPub**, whose correct solution is $D = e(g, g)^{abc}$. During **Setup** phase, $\mathcal{A}'$ sets $P = g^a$. The master secret key of **BasicPub**$^+$ is $msk = a$, which is unknown to $\mathcal{A}'$. Then, $\mathcal{A}$ and $\mathcal{A}'$ interact as follows:

- **KGen queries:** $\mathcal{A}'$ randomly selects $x_i \in \mathbb{Z}_p^*$, then calculates $pk_i = (g^b)^{x_i}$ and returns pk_i to $\mathcal{A}$. Note that the private key corresponding to pk_i is $sk_i = g^{abx_i}$, which is unknown to $\mathcal{A}'$.
  - H_2 **oracle:** $\mathcal{A}'$ keeps a list $\mathcal{L}_{H_2}$ of the history of calls to H_2 in which each tuple is $(\hat{X}_i, \hat{h}_i)$. If the query $\hat{X}_i \in \mathbb{G}_T$ already exists, $\hat{h}_i$ is returned. Otherwise, a random number $\hat{h}_i \in \mathbb{Z}_p^*$ is selected, $(\hat{X}_i, \hat{h}_i)$ is added to $\mathcal{L}_{H_2}$, and then $\hat{h}_i$ is returned.
- **Challenge:** After receiving (M_0, M_1, PK_0, PK_1) from $\mathcal{A}$. $\mathcal{A}'$ selects a random number $z, l_1, l_2, ..., l_d \in \mathbb{Z}_p^*$ and $b \in \{0, 1\}$. Then, it computes $f(x) = \Pi_{i=1}^{d}(x - l_i) + z = \Sigma_{i=0}^{d} a_i x^i$, $U = M_b \oplus H_1(z||g^c)$ and $V = g^c$. Finally, it outputs the challenge ciphertext $C = (U, V, \{a_i\}_{i\in[0,d]})$. Note that $l_i = H_2(e(sk_{pk_i}, V)) = H_2(e(g^{abx_i}, g^c)) = H_2(D^{x_i})$.
- **Guess:** $\mathcal{A}$ sends the guess $b' \in \{0, 1\}$ to $\mathcal{A}'$. Then $\mathcal{A}'$ randomly selects pk_i from $PK_{b'}$ and sets $z = 1/x_i$, then it randomly selects a random tuple $(\hat{X}_i, \hat{h}_i)$ from $\mathcal{L}_{H_2}$, and then outputs $D = \hat{X}_i^z$ as the solution of the received *CBDH* instance.

Probability Analysis. The probability that $\mathcal{A}'$ outputs the correct solution is at least $2\varepsilon/q_{H_2}$ if $\mathcal{A}$ has the advantage ε against **IND-CPA**$^+$ security of **BasicPub**$^+$. This analysis is similar to that in [26]. A detailed analysis can be found in [26].

Proof of Lemma 3. Through combining **Claim 3** and **Claim 4**, we can reduce the privacy of the proposed cross-domain multi-receiver IB-ME to the *CBDH* assumption. Specifically, if the adversary can break the privacy of the proposed cross-domain multi-receiver IB-ME, there will be an algorithm to break the *CBDH* assumption. If the *CBDH* assumption holds, the proposed cross-domain multi-receiver IB-ME can meet the privacy security.

Lemma 4. If an adversary $\mathcal{A}$ issuing at most q_R queries to the decryption key oracle $\mathcal{O}_2$, q_S queries to the encryption key oracle $\mathcal{O}_1$, and q_{H_2} queries to the random oracle H_2 has the advantage ε to break the authenticity property of cross-domain IB-ME, we can construct an algorithm that can solve the *CBDH* problem with the advantage $2\varepsilon/(q_{H_2} \cdot e^{2d+1} \cdot ((q_R + q_S)/(2d + 1))^{2d+1})$, where d is the number of receivers.

Proof. $\mathcal{A}'$ receives a *CBDH* tuple (g, g^a, g^b, g^c) whose solution is $D = e(g,g)^{abc}$. $\mathcal{A}'$ determines that the master secret key is $msk = (a, b, H_2)$, although (a, b) are unknown to $\mathcal{A}'$. Then, the interaction between $\mathcal{A}$ and $\mathcal{A}'$ is as follows:

- **Setup:** This process is similar to that of cross-domain IB-ME. In addition, $\mathcal{A}'$ will answer the queries of random oracle H_2.
  - H_2 **queries:** When $\mathcal{A}'$ receives a query about $\hat{X}_i \in \mathbb{G}_T$, the following process will be performed: If there is a record $(\hat{X}_i, \hat{h}_i)$ in the list $\mathcal{L}_{H_2}$, which the list is empty at initialization, $\hat{h}_i$ is returned; otherwise, a random number $\hat{h}_i \in \mathbb{Z}_p^*$ is selected. Next, $(\hat{X}_i, \hat{h}_i)$ is added in the list $\mathcal{L}_{H_2}$, and then $\hat{h}_i$ is returned.
- **Phase 1**: $\mathcal{A}$ can issue the polynomial queries for the encryption key and decryption key. Then, the response process is the same as that of cross-domain IB-ME.
- **Forgery:** At this moment, $\mathcal{A}'$ submits (C, ID_{Rev}, ID_{Snd}) to $\mathcal{A}'$, where $C = (C_1, C_2, C_3, \{C_{4,i}\}_{i\in[1,d]}\}, \{a_i, b_i\}_{i\in[0,d]}\}, C_5)$. Let $ID_S = ID_{Snd}$. Then, $\mathcal{A}'$ performs the following procedure:
  - For any $ID_{Rev_i} \in ID_{Rev}$, compute $H(ID_{Rev_i}) = Q_i$ and $H_0(ID_S) = Q'$. If $d_i \neq 1$ and $d' \neq 1$ in tuples $(ID_{Rev_i}, Q_i, \beta_i, d_i) \in \mathcal{L}_1$ and (ID_S, Q', β', d') $\in \mathcal{L}_2$, $\mathcal{A}'$ aborts. Otherwise, we know $H(ID_{Rev_i}) = g^{a\beta_i}$, $dk_{i,2} = g^{\gamma a \beta_i}$, $H_0(ID_{Snd}) = g^{b\beta'}$ and $ek_1 = g^{bc\beta'}$. Hence, $V_{ID_{Rev_i}} = H_2(e(dk_{i,2}, \Theta)$ $e(H(ID_{Rev_i}), C_5)e(H(ID_{Rev_i}), C_3)e(H_0(ID_{Snd}), C_{4,i}))$, where $e(H(ID_{Rev_i}), C_3)e(H_0(ID_{Snd}), C_4) = e(H(ID_{Rev_i}), ek_1) = e(g^{a\beta_i}, g^{bc\beta'})$ $= D^{\beta_i \beta'}$.
  - Compute $z = 1/(\beta_i\beta')$, select a random tuple $(\hat{X}_i, \hat{h}_i)$ from the list $\mathcal{L}_{H_2}$, and then return $D' = (X \cdot e(H(ID_{Rev_i}), C_5)^{-1} e(dk_{i,2}, \Theta)^{-1})^z$.

Probability Analysis. First of all, the game perfectly simulates the authenticity game where (C, ID_{Rev}, ID_{Snd}) satisfies (1) $\forall ID_{Rev_i} \in ID_{Rev}, ID_{Rev_i} \notin \mathbb{O}_2$, (2) $\forall ID'_S \in \mathbb{O}_1, ID'_S \neq ID_S$. Suppose $\mathcal{A}$ issues q_R queries to the oracle $\mathcal{O}_2$ and q_S queries to the oracle $\mathcal{O}_1$, and the probability that $\mathcal{A}$'s query will not be terminated is $\delta^{q_R+q_S}$. Similarly, in the challenge phase, the probability that $\mathcal{A}$'s query does not abort is $(1-\delta)^{2d+1}$. For the probability $\delta^{q_R+q_S}(1-\delta)^{2d+1}$ that does not abort, the maximum value of this probability is at $\delta_{opt} = (q_R + q_S)/(q_R + q_S + 2d + 1)$. If we leverage δ_{opt} as the probability that $\mathcal{A}$'s query does not terminate to obtain coins $d_i = 0$ in H query and H_0 query, the probability that $\mathcal{A}$'s query does not abort is at least $1/e^{2d+1}((q_R + q_S)/(2d+1))^{2d+1}$. If $\mathcal{A}$'s query will not be terminated, the probability that the correct solution is outputted by $\mathcal{A}'$ is at least $2\varepsilon/q_{H_2}$. As a result, the *CBDH* problem can be solved by $\mathcal{A}'$ with advantage $2\varepsilon/(q_{H_2} \cdot e^{2d+1} \cdot ((q_R + q_S)/(2d+1))^{2d+1})$.

When the *CBDH* assumption holds, by integrating **Lemma 3** and **Lemma 4**, we can conclude that the proposed cross-domain multi-receiver IB-ME is semantically secure in the random oracle model.

Table 3. Average Performance (ms) of Each Subprocess

Operation	Cross-domain IB-ME	Cross-domain Multi-receiver IB-ME
Initialization	1.67	1.74
Receiver-Authority Setup	17.93	19.35
Sender-Authority Setup	17.81	19.23
Encryption-Key Generation	39.35	42.33
Decryption-Key Generation	38.90	42.38
Encryption	88.52	652.6
Decryption	46.52	534.7

7 Performance Evaluation

In this section, we evaluate the performance of the proposed schemes. The experimental platform is performed on a personal computer running 64-bit Windows 10 with Intel(R) Core(TM) i5-10210U CPU with 1.60GHz based on JPBC 2.0.0 with the default Type A elliptic curve. Table 3 manifests the average running time of each subprocess of cross-domain IB-ME and cross-domain multi-receiver IB-ME. To eliminate the influence of random factors, we perform 50 times each subprocess and average them. As shown on the table, the running time of each subprocess in cross-domain IB-ME can be completed in milliseconds, which is acceptable. In cross-domain multi-receiver IB-ME, we assume that the number of receivers is 10. It can be seen from the table that when there are multiple receivers, the computation cost of the sender is less than that of making encryption for each receiver respectively, although the cost of our multi-receiver scheme is slightly higher in the decryption process. In short, each operation can be completed in milliseconds.

Table 4 shows the space costs of different elements in cross-domain IB-ME and cross-domain multi-receiver IB-ME from the theoretical aspect. It can be seen from the table that the size of the encryption key and the decryption key is constant. The storage cost of ciphertext is small in cross-domain IB-ME with constant size, while it is not much larger in multi-receiver case with ciphertext size that is linear with reverivers' number. In conclusion, the storage cost in both schemes is acceptable.

Table 4. Space Costs (bits) of Each Element

Element	Cross-Domain IB-ME	Cross-Domain Multi-receiver IB-ME
Encryption Key	$2\|\mathbb{G}\|$	$2\|\mathbb{G}\|$
Decryption Key	$2\|\mathbb{G}\|$	$2\|\mathbb{G}\|$
Message	n	n
Ciphertext	$3\|\mathbb{G}\| + n$	$d\|\mathbb{G}\| + 2(d+1)\|\mathbb{Z}_p^*\| + 3\|\mathbb{G}\| + n$

$|\mathbb{G}|$ represents the bit size of group $\mathbb{G}$. d represents the number of receivers.

From the above analysis, we conclude that the proposed schemes can run efficiently while providing privacy and authenticity of messages as well as the sender's identity privacy against the non-intended receiver.

8 Conclusion

In this paper, we extend the concept of ME to cross-domain scenarios, which is closer to the scenario of cross-domain secret communication such as intelligence operations. Specifically, we introduce the concept of cross-domain ME and propose a concrete instantiation (i.e., cross-domain IB-ME), where there is a sender-authority center and a receiver-authority center. The sender and receiver are managed by the authority center in their respective domain rather than by the same authority center. Then, we prove the privacy and authenticity of cross-domain IB-ME in the random oracle model on the *CBDH* assumption over bilinear groups. Additionally, we further extend the cross-domain IB-ME to the multi-receiver scenario, and propose cross-domain multi-receiver IB-ME and prove its security. Next, we implement these two schemes and give their performance evaluations, which show that the schemes can run efficiently.

Our work leaves several interesting questions. First, the black-box construction of cross-domain ME is a meaningful work. Second, it will be very interesting to construct efficient cross-domain multi-receiver ME without relying on random oracle. Third, constructing cross-domain IB-ME from lattice-based assumptions will also be very challenging.

References

1. Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. Match me if you can: Matchmaking encryption and its applications. In *CRYPTO*, pages 701–731. Springer, 2019.
2. Claude Castelluccia, Lukasz Olejnik, and Tran Minh-Dung. Selling off privacy at auction. In *NDSS*, 2014.
3. Elena Zheleva and Lise Getoor. Privacy in social networks: A survey. In *Social network data analytics*, pages 277–306. Springer, 2011.
4. Michael Nekrasov, Danny Iland, Miriam Metzger, Lisa Parks, and Elizabeth Belding. A user-driven free speech application for anonymous and verified online, public group discourse. *J. Internet Serv. Appl.*, 9(1):1–23, 2018.
5. Dirk Balfanz, Glenn Durfee, Narendar Shankar, Diana Smetters, Jessica Staddon, and Hao-Chi Wong. Secret handshakes from pairing-based key agreements. In *IEEE S&P*, pages 180–196. IEEE, 2003.
6. Yangguang Tian, Shiwei Zhang, Guomin Yang, Yi Mu, and Yong Yu. Privacy-preserving k-time authenticated secret handshakes. In *ACISP*, pages 281–300. Springer, 2017.
7. Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473. Springer, 2005.
8. John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE S&P*, pages 321–334. IEEE, 2007.

9. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS*, pages 89–98, 2006.
10. Junzuo Lai and Qiang Tang. Making any attribute-based encryption accountable, efficiently. In *ESORICS*, pages 527–547. Springer, 2018.
11. Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In *CCS*, pages 195–203, 2007.
12. Yinghui Zhang, Robert H Deng, Shengmin Xu, Jianfei Sun, Qi Li, and Dong Zheng. Attribute-based encryption for cloud computing access control: A survey. *ACM Computing Surveys (CSUR)*, 53(4):1–41, 2020.
13. Nuttapong Attrapadung and Hideki Imai. Dual-policy attribute based encryption. In *ACNS*, pages 168–185. Springer, 2009.
14. Y Sreenivasa Rao and Ratna Dutta. Computationally efficient dual-policy attribute based encryption with short ciphertext. In *ProvSec*, pages 288–308. Springer, 2013.
15. M Choudary Gorantla, Colin Boyd, and Juan Manuel González Nieto. Attribute-based authenticated key exchange. In *ACISP*, pages 300–317. Springer, 2010.
16. Vladimir Kolesnikov, Hugo Krawczyk, Yehuda Lindell, Alex Malozemoff, and Tal Rabin. Attribute-based key exchange with general policies. In *CCS*, pages 1451–1463, 2016.
17. Ivan Damgård, Helene Haagh, and Claudio Orlandi. Access control encryption: Enforcing information flow with cryptography. In *TCC*, pages 547–576. Springer, 2016.
18. Georg Fuchsbauer, Romain Gay, Lucas Kowalczyk, and Claudio Orlandi. Access control encryption for equality, comparison, and more. In *PKC*, pages 88–118. Springer, 2017.
19. Sam Kim and David J Wu. Access control encryption for general policies from standard assumptions. In *ASIACRYPT*, pages 471–501. Springer, 2017.
20. Xiuhua Wang, Harry WH Wong, and Sherman SM Chow. Access control encryption from group encryption. In *ACNS*, pages 417–441. Springer, 2021.
21. Shengmin Xu, Jianting Ning, Jinhua Ma, Xinyi Huang, Hwee Hwa Pang, and Robert H Deng. Expressive bilateral access control for internet-of-things in cloud-fog computing. In *SACMAT*, pages 143–154, 2021.
22. Zhongming Wang, Biwen Chen, Tao Xiang, Lu Zhou, Hongyang Yan, and Jin Li. Public key based searchable encryption with fine-grained sender permission control. In *ProvSec*, pages 3–18. Springer, 2021.
23. Christian Badertscher, Christian Matt, and Hendrik Waldner. Policy-compliant signatures. In *TCC*, pages 350–381. Springer, 2021.
24. Danilo Francati, Alessio Guidi, Luigi Russo, and Daniele Venturi. Identity-based matchmaking encryption without random oracles. In *INDOCRYPT*, pages 415–435. Springer, 2021.
25. Xiuhua Wang and Sherman SM Chow. Cross-domain access control encryption: arbitrary-policy, constant-size, efficient. In *IEEE S&P*, pages 388–401, 2021.
26. Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229. Springer, 2001.
27. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT*, pages 566–582. Springer, 2001.
28. Benoît Libert, Kenneth G Paterson, and Elizabeth A Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In *PKC*, pages 206–224. Springer, 2012.