6-2017

CCA Secure encryption supporting authorized equality test on ciphertexts in standard model and its applications

Yujue WANG
Singapore Management University, yjwang@smu.edu.sg

Hwee Hwa PANG
Singapore Management University, hhpang@smu.edu.sg

Ngoc Hieu TRAN
Singapore Management University, nhtran.2013@phdis.smu.edu.sg

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Published in Information Sciences, 2017 June, Volume 414, Pages 289-305
http://doi.org/10.1016/j.ins.2017.06.008

CCA Secure encryption supporting authorized equality test on ciphertexts in standard model and its applications

Yujue Wang[a,b,*], HweeHwa Pang[b], Ngoc Hieu Tran[b], Robert H. Deng[b]

[a] *School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, PR China*
[b] *School of Information Systems, Singapore Management University, 188065, Singapore*

A B S T R A C T

We present an *encryption scheme for authorized equality test on ciphertexts* (SEET), which allows the data owner to authorize a tester to compare her ciphertexts without decrypting their values. The security of SEET is formally proved against three types of adversary, two of them for ciphertext confidentiality in the phases before and after authorization respectively, and the third for token privacy. To the best of our knowledge, our SEET construction is the first encryption scheme supporting equality test on ciphertexts that is proven secure against the three types of adversary in the standard model. Our SEET construction outperforms existing schemes in terms of ciphertext size and encryption/decryption/testing costs. To show its application in set operations, we extend it into schemes for *controlled set distance computation*, such that a curious server is able to deduce the similarity/dissimilarity score between two encrypted user sets without knowing their elements.

Keyworas:
Data encryption
Equality test on ciphertexts
Data outsourcing
Private set intersection
Set operation
Implicit authentication

1. Introduction

In this paper, we consider *controlled set-theoretic operations* in data outsourcing scenario. The data owner has many sets of sensitive values and wants to engage some server to host them, for example, the data owner may outsource the sets to some cloud server. Thereafter, the data owner may authorize the server to perform set operations (e.g., intersection, union, and difference) as well as the relations of set equality and set inclusion on these sets without revealing the elements. In this scenario, the data owner would have three security concerns: (1) the privacy of the outsourced sets should be preserved against the server at all times; (2) the server can perform set operations/relations in a non-interactive way after getting authorization from the data owner; and (3) the authorization should be controlled in a way such that only the designated server can recover a token for performing set operations/relations.

To address the above concerns, all set elements and the authorization token should be encrypted before they are sent to the server, and the authorized server must be able to perform set operations on the ciphertexts. The encrypted set elements and token can only be decrypted by the data owner and server, respectively. It is important for the authorization token to be encrypted under the server's public key to prevent interception, otherwise any person who listens on the communication channel could get the token and encrypted sets, and further perform set operations/relations. In fact, *the fundamental problem in this scenario is the capability for the authorized server to compare the outsourced set elements in ciphertext format*.

* Corresponding author.
E-mail address: yjwang@smu.edu.sg (Y. Wang).

Table 1
Comparison with existing encryption schemes supporting equality test on ciphertexts.

Scheme	Element size		Computation cost			Security			Model
	Ciphertext	Token	Enc	Dec	Test	Security-1	Security-2	Security-3	
[14][1]	$3s_G + s_Z$	$s_G + s_Z$	$4E_G$	$3E_G$	$2E_G + 2E_{\hat{e}}$	IND-CCA	OW-CCA	IND-CCA	RO
[16][2]	$(2\lambda + 15)s_G + s_Z$	$3s_G$	$14E_G + E_{\hat{e}}$	$11E_G + 9E_{\hat{e}}$	$10E_G + 6E_{\hat{e}}$	IND-CCA	OW-CCA	No	Standard
[17][3]	$5s_G + s_Z$	$2s_{G_1}$	$6E_G + 2E_{\hat{e}}$	$2E_G + 2E_{\hat{e}}$	$4E_{\hat{e}}$	IND-CCA	OW-ID-CCA	No	RO
[18][4]	$5s_G + s_Z$	$2s_Z$	$6E_G$	$5E_G$	$2E_G + 2E_{\hat{e}}$	IND-CCA	OW-CCA	No	RO
[19]	$s_{G_1} + 3s_{G_2} + s_Z$	$2s_{G_1}$	$E_{G_1} + 3E_{G_2}$ $+E_{G_T} + E_{a\hat{e}}$	$E_{G_1} + 2E_{G_2}$ $+E_{G_T} + E_{a\hat{e}}$	$4E_{a\hat{e}}$	IND-CCA	OW-CCA	IND-CPA	RO
[23][5]	$7s_G + s_{G_T}$	$2s_Z$	$7E_G + E_{\hat{e}}$	–	$2E_G + 5E_{\hat{e}}$	IND-CPA	OW-CPA	No	Standard
[27][6]	$4s_{G_1} + 3s_Z$	$2s_{G_2}$	$6E_{G_1}$	$5E_{G_1}$	$2E_{a\hat{e}}$	IND-CCA	OW-CCA	IND-CPA	RO
[29][7]	$s_{\mathfrak{G}} + 2s_{G_1} + s_Z$ $+\lvert\mathcal{M}\rvert + \ell$	$3s_{G_2}$	$2E_{\mathfrak{G}} + 2E_{G_1}$	$2E_{\mathfrak{G}}$	$4E_{a\hat{e}}$	IND-CCA	OW-CCA	IND-CPA	RO
[30]	$3s_{\mathfrak{G}} + s_Z$ $+\lvert\mathcal{M}\rvert + \ell$	$2s_Z$	$5E_{\mathfrak{G}}$	$2E_{\mathfrak{G}}$	$4E_{\mathfrak{G}}$	IND-CCA	OW-CCA	No	RO
[31]	$5s_G + s_Z$	$2s_Z$	$8E_G + E_{\hat{e}}$	$3E_G + 4E_{\hat{e}}$	$2E_G + 4E_{\hat{e}}$	IND-CCA	OW-CCA	No	RO
[32][8]	$3s_G + s_Z$	–	$3E_G$	$3E_G$	$2E_{\hat{e}}$	No	OW-CCA	No	RO
SEET	$4s_{G_1}$	$4s_{G_2}$	$4E_{G_1}$	$3E_{G_1}$	$2E_{a\hat{e}}$	IND-CCA	OW-CCA	IND-CCA	Standard

Notes: 1. Lee et al. [14] enhanced the security of the scheme proposed by Huang et al. [10]. The token is the secret key of the data owner. The encryption scheme contained in the **Aut** procedure only offers IND-CPA security, which is not proved in [14]. The IND-CCA security relies on the public key for verification. 2. The generic PKEET constructions and instantiation in [16] do not consider the security of token. 3. Security against an adversary without holding a token is not proved in [17]. 4. There are four types of authorization in [18], where the first one directly gives the secret key of the data owner to the tester as token. 5. Pang and Ding's [23] scheme does not have a decryption procedure. 6. The scheme in [27] only offers IND-CPA security for ciphertexts. Here, an IND-CCA2 secure variant of [27] is compared, which is transformed using the technique of [22] in the random oracle model. 7. To generate a token, two data owners should interactively negotiate a common random value. 8. Equality test can be performed on ciphertexts without authorization.

Many studies have been conducted on public key encryption schemes with (authorized) equality test [14,18,19,27,29–32], which allows a sender to encrypt messages under the receiver's public key, and the receiver can authorize some tester to compare the ciphertexts. A rigorous requirement for these schemes is that the message space must be very large, otherwise a curious tester may be able to encrypt any message of his choice under the receiver's public key to compare with the received ciphertexts. Privacy of the data is compromised as long as some equivalence relationship is deduced. Clearly, these schemes are not applicable to the above problem which requires that only the data owner can encrypt and decrypt values. Also, these schemes only offer security in the random oracle model, except the one in [16].

In [23], Pang and Ding designed an encryption scheme that supports ad hoc equijoins on encrypted relations in an outsourced database in a private key setting. With their scheme, data fields in different relations are encrypted using different private parameters. Note that data entries in a relation column are not necessarily unique. With the authorization of the data owner in the form of an equijoin token, the server can only detect equivalence across the columns in two designated relations, and the server is prevented from testing the equivalence of data entries within a relation if there exists no matching value in the other relation. Applied to our problem scenario, their scheme only offers IND-CPA security for outsourced data in the phase prior to equijoin. Also, privacy of the equijoin token is not considered.

Outsourced private set intersection (OPSI) [1,12,13] is different from our problem scenario. In OPSI, two parties (say A and B) outsource their encoded data sets to a server, and the server is able to compute the intersection of their sets. The OPSI schemes presented in [12,13] do not have an authorization mechanism, thus the server is able to deduce the intersection immediately upon receiving the data sets. With the OPSI scheme in [1], when party A wishes to intersect her outsourced set with that of party B, she needs to interact with party B to get permission. Party B also needs to inform the server with a special message, if he agrees to the intersection request from party A. With that message, the server can generate an intermediate result for party A to deduce the final intersection.

1.1. Our contributions

In this paper, we introduce the notion of *secure encryption with authorized equality test* (SEET), which allows a data owner to authorize a tester to compare her ciphertexts without decrypting their values. We present a SEET construction on bilinear groups, for which the security is proved against three types of adversary. Specifically, before the tester is authorized by the data owner, the encrypted data enjoy IND-CCA2 security; after the authorization, SEET offers One-Way CCA2 security protection of the data against the tester; the token is encrypted with an IND-CCA2 secure scheme that permits only the tester to recover it. To the best of our knowledge, SEET is the first encryption scheme with equality test capability that does not require random oracles in its security proof. Analysis shows that our SEET construction is more efficient than existing related schemes in ciphertext size and computation cost (see Table 1 for a comparison).

Based on our SEET construction, we present non-interactive, controllable, and privacy-preserving solutions to distance computation on three distance functions. In the literature, these data sets may represent user profile features [8] and user preferences [2] (see Section 5.1 for a review). The fundamental problem in such controlled computation scenarios is for the

server to compare the data owner's sets in ciphertext format. Our solutions allow the data owner to deposit her encrypted datasets on a server, and later issue a token to enable the server to calculate the distance on the outsourced sets, where the distance reflects the dissimilarity between two sets. The authorization is controlled in a way that only the intended server is able to get the token and the authorized server can only perform distance-related computations. We also discuss how these solutions for controlled set distance computation can be transformed into *symmetric* solutions for implicit authentication with efficient user profile update, such that both initialization and authentication use the same encryption procedure (SEET) to protect the privacy of user profile features.

1.2. Related works

Yang et al. [32] presented a public key encryption scheme that supports equality test (PKEET). For any two ciphertexts which may be generated under different public keys, anyone is able to check whether they encrypt the same message. Thus, their scheme permits no authorization for when a tester may perform equality test on ciphertexts. PKEET schemes with *delegable/authorized* equality test were studied in [10,18,19,27,29,30], such that only a suitably enabled tester (e.g., a server) can perform equality test on the ciphertexts. In particular, the PKEET schemes proposed in [10,18,19,30] allow two users to authorize a tester to compare their ciphertexts. With that authorization, the tester can also compare ciphertexts generated under the same public key.

In [30], Tang proposed an all-or-nothing PKEET (AoN-PKEET) scheme. Slamanig, Spreitzer and Unterluggauer [27] considered a special case of AoN-PKEET such that the equality test is only carried out for ciphertexts under the same public key, which is called AoN-PKEET*. Based on the ElGamal encryption scheme [9,27] proposed an IND-CPA secure AoN-PKEET* scheme in asymmetric bilinear groups. They argued that their scheme can be transformed into an IND-CCA2 secure scheme in the random oracle model following the approach in [7,21,22]. Ma [17] presented an encryption scheme that supports outsourced equality test in identity-based setting. Recently, Lee et al. [15] proposed semi-generic constructions for PKEET and identity-based encryption scheme with equality test.

In [16], a generic PKEET and identity-based PKEET are constructed by employing 2-level hierarchical identity-based encryption scheme and strongly unforgeable one-time signature scheme. They also presented a PKEET instantiation from [3] and [4] by following their generic framework. Although their constructions can be proved in the standard model, efficiency remains a weakness and the security of authorization token is not considered. The basic idea behind controlled equijoin scheme for relational databases is also about the capability to perform equality test on ciphertexts, and was initially investigated by Pang and Ding [23] in a private key setting on symmetric bilinear groups.

1.3. Paper organization

The remainder of this paper is organized as follows. Section 2 covers some preliminaries. We introduce SEET and present a construction in Section 3. The security of SEET is proved in Section 4, where its performance is analyzed and compared with existing schemes as well. Privacy-preserving solutions for three controlled set distance computation cases are presented in Section 5 based on the SEET scheme. Finally, Section 6 concludes the paper.

2. Preliminaries

A hash function $H : \{0,1\}^{l_k} \times \{0,1\}^{l_m} \to \{0,1\}^{l_n}$ is a ε_{tcr}-secure targeted collision-resistant hash function [24,28] if any probabilistic polynomial-time (PPT) adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has only negligible advantage in winning the collision finding game, that is

$$\mathsf{Adv}_{\mathcal{A},H} = \left[m \neq m' \wedge H(m) = H(m') \;\middle|\; \begin{array}{l} (m, \mathsf{state}) \xleftarrow{\$} \mathcal{A}_1() \\ k \xleftarrow{\$} \{0,1\}^{l_k} \\ m' \xleftarrow{\$} \mathcal{A}_2(k, \mathsf{state}) \end{array} \right] \leq \varepsilon_{\text{tcr}}$$

where l_k, l_m, l_n denote the length of key, message, and output of H, respectively.

Suppose $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$ and G_T are cyclic groups with prime order p and efficient group operations. The mapping $\hat{e} : G_1 \times G_2 \to G_T$ is bilinear if the following properties are satisfied:

- Bilinearity: $\forall \mu \in G_1$, $\nu \in G_2$, and $\forall a, b \in Z_p^*$, $\hat{e}(\mu^a, \nu^b) = \hat{e}(\mu, \nu)^{ab}$;
- Non-degeneracy: $\hat{e}(g_1, g_2) \neq 1$;
- Efficiency: $\hat{e}$ is efficiently computable.

If $G_1 = G_2$, then $\hat{e}$ is a symmetric bilinear map; otherwise, it is asymmetric. In this paper, our schemes are constructed using Type 3 (asymmetric) pairing, i.e., there exists no efficiently computable isomorphism from G_2 to G_1.

The security of our constructions relies on the following complexity assumptions [5,27].

Computational Diffie–Hellman Assumption (CDH). Let $G = \langle g \rangle$ be a cyclic group with prime order p. The CDH assumption states that given a tuple $(g, g^x, g^y) \in G^3$, where $x, y \in_R Z_p^*$, any PPT algorithm has negligible advantage ε_{cdh} in computing g^{xy}.

Decisional Diffie–Hellman Assumption (DDH). Let $G = \langle g \rangle$ be a cyclic group with prime order p. The DDH assumption states that given a tuple $(g, g^x, g^y, g^z) \in G^4$, where $x, y, z \in_R Z_p^*$, any PPT algorithm has negligible advantage $\varepsilon_{\mathsf{ddh}}$ in deciding whether $g^{xy} = g^z$.

Computational co-Diffie–Hellman Assumption (co-CDH*). Let $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$ and G_T be three distinct cyclic groups with prime order p, and $\hat{e} : G_1 \times G_2 \to G_T$ be a Type 3 pairing. The co-CDH* assumption states that given a tuple $(g_1^x, g_1^y, g_2^x, g_2^y) \in G_1^2 \times G_2^2$, where $x, y \in_R Z_p^*$, any PPT algorithm has negligible advantage $\varepsilon_{\mathsf{cocdh}}$ in computing g_1^{xy}.

Symmetric eXternal Diffie–Hellman Assumption (SXDH). Let $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$ and G_T be cyclic groups with prime order p and $\hat{e} : G_1 \times G_2 \to G_T$ be a Type 3 bilinear pairing. The SXDH assumption states that the DDH assumption holds in both groups G_1 and G_2.

3. SEET Scheme

In this section, we introduce SEET and formalize its security model. We then present a SEET construction, and prove its security in the standard model in next section. To the best of our knowledge, our SEET construction is the first provably secure encryption scheme with equality test on ciphertexts against three types of adversary that does not require random oracles in the security proof.

3.1. System framework and requirements

The SEET system model consists of a data owner and a curious tester. The tester may be a remote storage server. The owner has many data sets, denoted by **A, B, C**, etc., and engages the tester to host them. All entries in data sets are uniformly distributed. Subsequently, the owner can authorize the tester to compare her outsourced values. Accordingly, a secure SEET scheme should satisfy the following functionalities and requirements.

- Authorized comparison: Without authorization from the data owner, the tester is unable to compare the outsourced data. Also, the authorization token can only be known by the tester, otherwise someone who intercepts the communication between the data owner and tester would be able to compare the outsourced data.
- Non-interactivity: With the authorization, the tester can compare the outsourced data without any interaction with the data owner.
- Privacy: Since the tester may be curious about the outsourced values, all values must be stored in ciphertext, such that only the owner can retrieve them for decryption. That means the tester must perform the authorized comparison on the encrypted elements without knowing their real values. Even with the authorization to perform comparison, the tester should still be unable to infer the real values of outsourced data. Also, the authorization token cannot leak the secret key of data owner.

3.2. Definitions

A SEET scheme consists of the following eight procedures:

- **Setup**$(\lambda) \to \mathsf{pp}$: Given security parameter λ, the system setup procedure produces public parameter pp.
- **KGen**$_O(\mathsf{pp}) \to \mathsf{sk}_O$: With public parameter pp, the data owner executes the key generation procedure to produce a secret key sk_O.
- **KGen**$_T(\mathsf{pp}) \to (\mathsf{sk}_T, \mathsf{pk}_T)$: With public parameter pp, the tester executes the key generation procedure to produce a pair of secret key sk_T and public key pk_T.
- **Enc**$(\mathsf{sk}_O, \mathsf{pp}, m) \to C$: With secret key sk_O and public parameter pp, the data owner runs the encryption procedure on message m to produce a ciphertext C.
- **Dec**$(\mathsf{sk}_O, \mathsf{pp}, C) \to m/\perp$: With secret key sk_O and public parameter pp, the data owner runs the decryption procedure on ciphertext C to produce a message m or $\perp$ that signifies an error in decryption.
- **Aut**$(\mathsf{sk}_O, \mathsf{pp}, \mathsf{pk}_T) \to \mathsf{etk}$: Using secret key sk_O, public parameter pp and the tester's public key pk_T, the data owner runs the authorization procedure to produce an encrypted token etk.
- **Rec**$(\mathsf{sk}_T, \mathsf{pp}, \mathsf{etk}) \to \mathsf{tk}/\perp$: With secret key sk_T and public parameter pp, the tester runs the token recovery procedure on etk to produce a token tk or $\perp$ that signifies an error in recovery.
- **Com**$(\mathsf{pp}, \mathsf{tk}, C_1, C_2) \to 1/0$: With public parameter pp and token tk, the tester runs the comparison procedure on two ciphertexts C_1 and C_2. The procedure outputs 1 if C_1 and C_2 encrypt the same plaintext; otherwise, the procedure outputs 0.

A SEET scheme must be *sound* in the sense that: (1) Every ciphertext generated by **Enc** is decryptable by **Dec**; (2) The token generated by **Aut** is recoverable by **Rec**; (3) For any two ciphertexts that encrypt the same message, the procedure **Com** must output 1; (4) For any two ciphertexts that encrypt different messages, the procedure **Com** must output 0.

Definition 3.1 (Soundness)**.** A SEET scheme is *sound* if, for any security parameter $\lambda \in \mathbb{N}$, any public parameter $\mathsf{pp} \leftarrow$ **Setup**(λ), any secret key of data owner $\mathsf{sk}_O \leftarrow$ **KGen**$_O(\mathsf{pp})$, and any secret/public key pair of tester $(\mathsf{sk}_T, \mathsf{pk}_T) \leftarrow$ **KGen**$_T(\mathsf{pp})$, the following conditions are satisfied:

Set-up: The challenger runs the **Setup** procedure to produce public parameter pp and the $\mathbf{KGen}_O$ procedure to produce a secret key sk_O. The public parameter pp is given to the adversary.

Phase 1: The adversary is able to adaptively issue the following two types of queries.

- Encryption query: For a queried message $m \in \mathcal{M}$, the challenger returns $C \leftarrow \mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}, m)$.
- Decryption query: For a queried ciphertext C, the challenger returns m or $\perp$ according to $\mathbf{Dec}(\mathsf{sk}_O, \mathsf{pp}, C)$.

Challenge: At the end of Phase 1, the adversary randomly picks two messages $m_0, m_1 \xleftarrow{\$} \mathcal{M}$, and sends them to the challenger. The challenger chooses a random value $b \xleftarrow{\$} \{0,1\}$, computes $C_b \leftarrow \mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}, m_b)$, and gives C_b to the adversary.

Phase 2: The adversary is able to issue queries in the same way as in Phase 1, except that C_b cannot be submitted for decryption.

Guess: At the end of Phase 2, the adversary outputs a guess b', and succeeds in the security game if $b' = b$.

Fig. 1. IND-CCA security game for ciphertexts against Type-1 adversary.

1. For every $m \in \mathcal{M}$, $\mathbf{Dec}(\mathsf{sk}_O, \mathsf{pp}, \mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}, m)) = m$.
2. $\mathbf{Rec}(\mathsf{sk}_T, \mathsf{pp}, \mathbf{Aut}(\mathsf{sk}_O, \mathsf{pp}, \mathsf{pk}_T)) = \mathsf{tk}$.
3. For any $m_1 = m_2 \in \mathcal{M}$ such that $C_1 \leftarrow \mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}, m_1)$ and $C_2 \leftarrow \mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}, m_2)$, $\mathbf{Com}(\mathsf{pp}, \mathsf{tk}, C_1, C_2) = 1$ where $\mathsf{tk} \leftarrow \mathbf{Rec}(\mathsf{sk}_T, \mathsf{pp}, \mathbf{Aut}(\mathsf{sk}_O, \mathsf{pp}, \mathsf{pk}_T))$.
4. For any $m_1 \neq m_2 \in \mathcal{M}$ such that $C_1 \leftarrow \mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}, m_1)$ and $C_2 \leftarrow \mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}, m_2)$, $\mathbf{Com}(\mathsf{pp}, \mathsf{tk}, C_1, C_2) = 0$ where $\mathsf{tk} \leftarrow \mathbf{Rec}(\mathsf{sk}_T, \mathsf{pp}, \mathbf{Aut}(\mathsf{sk}_O, \mathsf{pp}, \mathsf{pk}_T))$.

In a SEET scheme, the user can authorize the tester to perform equality test on her ciphertexts. In relation to the confidentiality of ciphertexts and token produced by a SEET scheme, three types of adversary need to be considered:

- *Type-1 adversary*: It models a curious tester who has not been authorized by the user, that is, the tester only has the public parameter pp.
- *Type-2 adversary*: It models a curious tester who has been authorized by the user, that is, the tester has both the public parameter pp, the tester's secret and public keys, and an encrypted token etk.
- *Type-3 adversary*: It models a curious adversary who has both the public parameter pp and the tester's public key pk_T, and intends to recover the token tk from encrypted token etk.

The first two types of adversaries respectively capture the notion of ciphertext confidentiality before and after the authorization procedure is executed, while the third captures the notion of token confidentiality. We formally define the security of a SEET scheme in the following three definitions.

Definition 3.2 (IND-CCA security against Type-1 adversary)**.** Let $\Gamma = (\mathbf{Setup}, \mathbf{KGen}_O, \mathbf{KGen}_T, \mathbf{Enc}, \mathbf{Dec}, \mathbf{Aut}, \mathbf{Rec}, \mathbf{Com})$ be a SEET scheme. Suppose $\mathcal{A}$ is a PPT adversary who interacts with a challenger $\mathcal{C}$ to perform the security game in Fig. 1. Let

$$\mathsf{Adv}^{\mathsf{ind-cca}}_{\Gamma,\mathcal{A}} = \left| \Pr[b' = b] - \frac{1}{2} \right|$$

Γ is said to offer indistinguishability under adaptive chosen ciphertext attack (IND-CCA) for ciphertexts against Type-1 adversary if, for all PPT adversary $\mathcal{A}$, there exists a negligible function $\varepsilon(\cdot)$ such that $\mathsf{Adv}^{\mathsf{ind-cca}}_{\Gamma,\mathcal{A}} \leq \varepsilon(\cdot)$.

Definition 3.3 (OW-CCA security against Type-2 adversary)**.** Let $\Gamma = (\mathbf{Setup}, \mathbf{KGen}_O, \mathbf{KGen}_T, \mathbf{Enc}, \mathbf{Dec}, \mathbf{Aut}, \mathbf{Rec}, \mathbf{Com})$ be a SEET scheme. Suppose $\mathcal{A}$ is a PPT adversary who interacts with a challenger $\mathcal{C}$ to perform the security game in Fig. 2. Let

$$\mathsf{Adv}^{\mathsf{ow-cca}}_{\Gamma,\mathcal{A}} = \Pr[m' = m^*]$$

Γ is said to offer one-way confidentiality under adaptive chosen ciphertext attack (OW-CCA) for ciphertexts against Type-2 adversary if, for all PPT adversary $\mathcal{A}$, there exists a negligible function $\varepsilon(\cdot)$ such that $\mathsf{Adv}^{\mathsf{ow-cca}}_{\Gamma,\mathcal{A}} \leq \varepsilon(\cdot)$.

Note that every data owner only needs to generate one token for the tester to compare her ciphertexts. Thus, the following security game is defined in a multi-data-owner setting.

Set-up: The challenger runs the **Setup** procedure to produce public parameter pp, the $\mathbf{KGen}_O$ procedure to produce a secret key sk_O, and gives pp to the adversary. The adversary runs the $\mathbf{KGen}_T$ procedure to produce a pair of tester's secret/public keys $(\mathsf{sk}_T, \mathsf{pk}_T)$. Next, the challenger runs $\mathbf{Aut}(\mathsf{sk}_O, \mathsf{pp}, \mathsf{pk}_T)$ to get an encrypted token etk and gives etk to the adversary.

Phase 1: The adversary is able to adaptively issue the following two types of queries.

- Encryption query: For a queried message $m \in \mathcal{M}$, the challenger returns $C \leftarrow \mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}, m)$.
- Decryption query: For a queried ciphertext C, the challenger returns m or $\perp$ according to $\mathbf{Dec}(\mathsf{sk}_O, \mathsf{pp}, C)$.

Challenge: At the end of Phase 1, the challenger randomly picks a message $m^* \in_R \mathcal{M}$, computes $C^* \leftarrow \mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}, m^*)$, and sends C^* to the adversary.

Phase 2: The adversary is able to issue queries in the same way as in Phase 1, except that C^* cannot be submitted for decryption.

Guess: At the end of Phase 2, the adversary outputs a guess m', and succeeds in the security game if $m' = m^*$.

Fig. 2. OW-CCA security game for ciphertexts against Type-2 adversary.

Set-up: The challenger runs the **Setup** procedure to produce public parameter pp. Next, the challenger runs the $\mathbf{KGen}_T$ procedure to produce a pair of tester's secret/public keys $(\mathsf{sk}_T, \mathsf{pk}_T)$. The challenger gives pp and pk_T to the adversary.

Phase 1: The adversary is able to adaptively issue the following type of queries.

- Token recovery query: The adversary runs the $\mathbf{KGen}_O$ algorithm to produce a secret key sk_{O_i}, and invokes $\mathsf{etk}_i \leftarrow \mathbf{Aut}(\mathsf{sk}_{O_i}, \mathsf{pp}, \mathsf{pk}_T)$. The encrypted token etk_i is given to the challenger, who returns tk_i or $\perp$ according to $\mathbf{Rec}(\mathsf{sk}_T, \mathsf{pp}, \mathsf{etk}_i)$.

Challenge: At the end of Phase 1, the adversary runs the $\mathbf{KGen}_O$ algorithm twice to produce two secret keys sk_{O_0} and sk_{O_1}, and gives them to the challenger. The challenger chooses a random value $b \xleftarrow{\$} \{0,1\}$, computes $\mathsf{etk}_b \leftarrow \mathbf{Aut}(\mathsf{sk}_{O_b}, \mathsf{pp}, \mathsf{pk}_T)$, and gives etk_b to the adversary.

Phase 2: The adversary is able to issue queries in the same way as in Phase 1, except that etk_b cannot be submitted for recovery.

Guess: At the end of Phase 2, the adversary outputs a guess b', and succeeds in the security game if $b' = b$.

Fig. 3. IND-CCA security game for tokens against Type-3 adversary.

Definition 3.4 (IND-CCA security against Type-3 adversary)**.** Let $\Gamma = (\mathbf{Setup}, \mathbf{KGen}_O, \mathbf{KGen}_T, \mathbf{Enc}, \mathbf{Dec}, \mathbf{Aut}, \mathbf{Rec}, \mathbf{Com})$ be a SEET scheme. Suppose $\mathcal{A}$ is a PPT adversary who interacts with a challenger $\mathcal{C}$ to perform the security game in Fig. 3. Let

$$\mathsf{Adv}^{\mathsf{ind-cca}}_{\mathcal{A},\mathsf{token}} = \left| \Pr[b' = b] - \frac{1}{2} \right|$$

Γ is said to offer indistinguishability under adaptive chosen ciphertext attack (IND-CCA) for tokens against Type-3 adversary if, for all PPT adversary $\mathcal{A}$, there exists a negligible function $\varepsilon(\cdot)$ such that $\mathsf{Adv}^{\mathsf{ind-cca}}_{\mathcal{A},\mathsf{token}} \leq \varepsilon(\cdot)$.

3.3. Construction

In this section, we present a SEET construction using Type 3 bilinear pairing.

Setup: Choose a bilinear map $\hat{e}: G_1 \times G_2 \to G_T$, where $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$, G_T are cyclic groups with prime order p, and the SXDH assumption holds. Let H_1: $G_1 \to Z_p$, $H_2: G_1^3 \to Z_p$, H_3: $G_2 \to Z_p$ and $H_4: G_2^3 \to Z_p$ be four target collision-resistant hash functions. The public parameter is $\mathsf{pp} = (\hat{e}, G_1, G_2, G_T, g_1, g_2, p, H_1, H_2, H_3, H_4)$.

KGen$_O$: Pick random values $x, y, u, v \xleftarrow{\$} Z_p^*$. The data owner's secret key is $\mathsf{sk}_O = (x, y, u, v)$.

KGen$_T$: Pick random values $\tilde{x}, \tilde{y}, \tilde{u}, \tilde{v} \xleftarrow{\$} Z_p^*$ and compute

$$\hbar_1 = g_2^{\tilde{x}},\ \hbar_2 = g_2^{\tilde{y}},\ \hbar_3 = g_2^{\tilde{u}},\ \hbar_4 = g_2^{\tilde{v}}.$$

The tester's secret key and public key are $\mathsf{sk}_T = (\tilde{x}, \tilde{y}, \tilde{u}, \tilde{v})$ and $\mathsf{pk}_T = (\hbar_1, \hbar_2, \hbar_3, \hbar_4)$, respectively.

Enc: Given a message $m \in G_1$, choose a random value $\alpha \xleftarrow{\$} Z_p^*$ and compute the ciphertext $C = (c_1, c_2, c_3, c_4)$ as follows:

$$c_1 = g_1^{\alpha}, c_2 = m g_1^{\alpha x}, c_3 = (g_1^{x\theta+y})^{\alpha}, c_4 = (g_1^{u\vartheta+v})^{\alpha}$$

where $\theta = H_1(c_1)$ and $\vartheta = H_2(c_1, c_2, c_3)$.

Dec: Given a ciphertext $C = (c_1, c_2, c_3, c_4)$, compute

$$\theta = H_1(c_1) \text{ and } \vartheta = H_2(c_1, c_2, c_3),$$

and check

$$c_3 \stackrel{?}{=} c_1^{x\theta+y} \text{ and } c_4 \stackrel{?}{=} c_1^{u\vartheta+v}.$$

If both conditions hold, then output $m = c_2/c_1^x$; otherwise, output $\perp$.

Aut: With sk_O and pk_T, choose a random value $\gamma \xleftarrow{\$} Z_p^*$ and compute an encrypted token $\mathsf{etk} = (t_1, t_2, t_3, t_4)$ as follows:

$$t_1 = g_2^{\gamma}, t_2 = g_2^{x\gamma} \hbar_1^{\gamma}, t_3 = (\hbar_1^{\tilde{\theta}} \hbar_2)^{\gamma}, t_4 = (\hbar_3^{\tilde{\vartheta}} \hbar_4)^{\gamma}$$

where $\tilde{\theta} = H_3(t_1)$ and $\tilde{\vartheta} = H_4(t_1, t_2, t_3)$.

Rec: Given an encrypted token $\mathsf{etk} = (t_1, t_2, t_3, t_4)$, compute

$$\tilde{\theta} = H_3(t_1) \text{ and } \tilde{\vartheta} = H_4(t_1, t_2, t_3),$$

and check

$$t_3 \stackrel{?}{=} t_1^{\tilde{x}\tilde{\theta}+\tilde{y}} \text{ and } t_4 \stackrel{?}{=} t_1^{\tilde{u}\tilde{\vartheta}+\tilde{v}}.$$

If both conditions hold, then output $\mathsf{tk} = (\hat{t}_1, \hat{t}_2) = (t_1, t_2/t_1^{\tilde{x}})$; otherwise, output $\perp$.

Com: Given two ciphertexts $C = (c_1, c_2, c_3, c_4)$ and $C' = (c_1', c_2', c_3', c_4')$ and a token tk, check whether

$$\hat{e}\left(\frac{c_2}{c_2'}, \hat{t}_1\right) \stackrel{?}{=} \hat{e}\left(\frac{c_1}{c_1'}, \hat{t}_2\right) \tag{1}$$

Output "1" if the condition holds, which implies $m = m'$; otherwise output "0" which signifies that $m \neq m'$.

Theorem 3.1. *The SEET scheme proposed above is sound.*

Proof. The correctness of ciphertext decryption and token recovery are straightforward. For ciphertext comparison, we have

$$\hat{e}\left(\frac{c_2}{c_2'}, \hat{t}_1\right) = \hat{e}\left(\frac{m g_1^{\alpha x}}{m' g_1^{\alpha' x}}, g_2^{\gamma}\right) = \hat{e}\left(\frac{m}{m'} \cdot g_1^{(\alpha-\alpha')x}, g_2^{\gamma}\right) \stackrel{m=m'}{\Longleftrightarrow} \hat{e}(g_1^{\alpha-\alpha'}, g_2^{x\gamma}) = \hat{e}\left(\frac{c_1}{c_1'}, \hat{t}_2\right)$$

Thus, Equality (1) holds if and only if $m = m'$. □

4. Analysis

4.1. Security

To prove the security of our proposed SEET construction in the previous section, we need the following lemma [6].

Lemma 4.1. *Let $\mathbb{E}_1$, $\mathbb{E}_2$, and $\mathbb{F}$ be events defined on some probability space. Suppose the event $\mathbb{E}_1 \wedge \neg\mathbb{F}$ occurs if and only if $\mathbb{E}_2 \wedge \neg\mathbb{F}$ occurs. Then $|\Pr[\mathbb{E}_1] - \Pr[\mathbb{E}_2]| \leq \Pr[\mathbb{F}]$.*

Theorem 4.1. *Suppose H_1 and H_2 are secure target collision-resistant hash functions. Prior to authorization, the SEET scheme in Section 3.3 offers IND-CCA security for ciphertexts in the standard model assuming that the SXDH assumption holds.*

Proof. The following proof follows the standard framework established in [28].

Let $\varepsilon_{\mathsf{sxdh}}$ be the advantage of a PPT adversary in deciding the DDH problem on G_1. Let ε_1 and ε_2 be the advantages of a PPT adversary in finding collisions in H_1 and H_2, respectively.

Let $\mathcal{A}$ be a PPT adversary that has non-negligible advantage ε in attacking the IND-CCA2 security for ciphertexts in the SEET scheme. Prior to the authorization phase, the adversary does not have the token $\mathtt{tk}$. Suppose $\mathcal{A}$ issues at most q_E encryption queries and q_D decryption queries. We show that if such an adversary $\mathcal{A}$ exists, then one can construct an algorithm $\mathcal{E}$ to solve the SXDH problem with non-negligible probability.

Let $G_1 = \langle g \rangle, G_2, G_T$ be cycle groups with prime order p and bilinear map $\hat{e} : G_1 \times G_2 \to G_T$. Also, let H_1 and H_2 be two target collision resistant hash functions. At first, algorithm $\mathcal{E}$ is given a SXDH instance $(g_1, g_1^{\alpha^*}, g_1^x, g_1^z) \in (G_1)^4$. The goal of $\mathcal{E}$ is to determine whether $g_1^{\alpha^* x} = g_1^z$. Algorithm $\mathcal{E}$ simulates the challenger and interacts with adversary $\mathcal{A}$ as follows.

Set-up: Algorithm $\mathcal{E}$ sets $c_1^* = g_1^{\alpha^*}$, computes $\theta^* = H_1(c_1^*)$, and randomly picks $u, v, w \xleftarrow{\$} Z_p^*$. The data owner's secret key sk_O is $(x, w - x\theta^*, u, v)$, where x and $y = w - x\theta^*$ are unknown to $\mathcal{E}$.

Phase 1: The adversary adaptively makes the following queries.

- Encryption query: For input message m, algorithm $\mathcal{E}$ randomly picks $\alpha \xleftarrow{\$} Z_p^*$, and computes ciphertext $C = (c_1, c_2, c_3, c_4)$ as follows:

$$c_1 = g_1^{\alpha}, \theta = H_1(c_1), c_2 = m(g_1^x)^{\alpha}, c_3 = (g_1^x)^{\alpha(\theta - \theta^*)} g_1^{\alpha w}, \vartheta = H_2(c_1, c_2, c_3), c_4 = g_1^{\alpha(u\vartheta + v)}$$

  Note that $c_3 = (g_1^{x\theta + (w - x\theta^*)})^{\alpha} = (g_1^{x\theta + y})^{\alpha}$. $\mathcal{E}$ returns C as the ciphertext for m. Here if $c_1 = g_1^{\alpha^*}$ or $c_1 = g_1^x$, then algorithm $\mathcal{E}$ stops the simulation.
- Decryption query: For input ciphertext $C = (c_1, c_2, c_3, c_4)$, algorithm $\mathcal{E}$ first checks $c_1 \stackrel{?}{=} c_1^*$. If so, algorithm $\mathcal{E}$ aborts; otherwise, $\mathcal{E}$ computes

$$\theta = H_1(c_1) \text{ and } \vartheta = H_2(c_1, c_2, c_3)$$

  and checks

$$c_4 \stackrel{?}{=} c_1^{u\vartheta + v}$$

  If the condition does not hold or $\theta = \theta^*$, then algorithm $\mathcal{E}$ outputs $\perp$; otherwise, $\mathcal{E}$ computes

$$m = c_2 / (c_3 c_1^{-w})^{(\theta - \theta^*)^{-1}}$$

  where $c_3 = (g_1^{x\theta} g_1^y)^{\alpha}$ and $c_1^w = (g_1^w)^{\alpha} = (g_1^{x\theta^*} g_1^y)^{\alpha}$. Then, $\mathcal{E}$ returns the message m.

Challenge: Adversary $\mathcal{A}$ picks two random messages m_0 and m_1 of the same length, and gives them to $\mathcal{E}$. Then, algorithm $\mathcal{E}$ randomly picks $b \xleftarrow{\$} \{0, 1\}$, and generates the challenge ciphertext $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ as follows:

$$c_1^* = g_1^{\alpha^*}, \theta^* = H_1(c_1^*), c_2^* = m_b g_1^z, c_3^* = (g_1^{\alpha^*})^w, \vartheta^* = H_2(c_1^*, c_2^*, c_3^*), c_4^* = (g_1^{\alpha^*})^{u\vartheta^* + v}$$

Note that $c_3 = (g_1^{x\theta^* + y})^{\alpha^*}$. Then, algorithm $\mathcal{E}$ returns the challenge ciphertext C^*.

Phase 2: The adversary can continue to make queries except that C^* cannot be submitted for decryption.

Guess: If algorithm $\mathcal{E}$ does not abort in the two query phases, then the simulation is perfect for CCA2 attack and the adversary's view is the same as that in a real attack. Eventually, adversary $\mathcal{A}$ returns a guess b'. Algorithm $\mathcal{E}$ outputs "1" if $b' = b$; otherwise, $\mathcal{E}$ outputs "0".

Analysis. We analyze the success probability of algorithm $\mathcal{E}$ through a sequence of games [26,28]. The first game $\mathcal{G}_0$ is the same as above but has no abortion during queries; while the last one $\mathcal{G}_6$ gives no advantage to adversary $\mathcal{A}$. Let Φ_i denote the event that $b' = b$ in game $\mathcal{G}_i$ for $0 \le i \le 6$. Thus, $\mathsf{Adv}_{\Gamma,\mathcal{A}}^{\mathsf{ind-cca}} = |\Pr[\Phi_0] - \frac{1}{2}| = \varepsilon$.

Game $\mathcal{G}_1$: (Eliminate the correct guess of α^* and x). Game $\mathcal{G}_1$ is similar to game $\mathcal{G}_0$ except that α^* and x would not be chosen in answering encryption queries. Note that if either α^* or x is chosen, then the given DDH problem on G_1 is already solved by algorithm $\mathcal{E}$. Denote this abortion case by event $\mathbb{E}_1$. According to Lemma 4.1, we have $|\Pr[\Phi_1] - \Pr[\Phi_0]| \le \Pr[\mathbb{E}_1]$.

Lemma 4.2. $\Pr[\mathbb{E}_1] \le \frac{2q_E}{p}$.

The proof of Lemma 4.2 is straightforward due to

$$\Pr[\mathbb{E}_1] = \frac{2}{p} + \left(1 - \frac{2}{p}\right)\frac{2}{p} + \cdots + \left(1 - \frac{2}{p}\right)^{q_E - 1}\frac{2}{p} \le \frac{2q_E}{p}$$

Game $\mathcal{G}_2$: (Eliminate the correct guess of c_1). Game $\mathcal{G}_2$ is similar to game $\mathcal{G}_1$ except that a decryption query for ciphertext $C = (c_1, c_2, c_3, c_4)$ is rejected and algorithm $\mathcal{E}$ aborts, if $c_1 = c_1^*$. Denote this rejection case by event $\mathbb{E}_2$. According to Lemma 4.1, we have $|\Pr[\Phi_2] - \Pr[\Phi_1]| \le \Pr[\mathbb{E}_2]$.

Lemma 4.3. $\Pr[\mathbb{E}_2] \le \frac{q_D}{p}$.

The proof of Lemma 4.3 is straightforward due to

$$\Pr[\mathbb{E}_2] = \frac{1}{p} + \left(1 - \frac{1}{p}\right)\frac{1}{p} + \cdots + \left(1 - \frac{1}{p}\right)^{q_D - 1}\frac{1}{p} \le \frac{q_D}{p}$$

Game $\mathcal{G}_3$: (Eliminate the target collision of H_1). Game $\mathcal{G}_3$ is similar to game $\mathcal{G}_2$ except that a decryption query for a valid ciphertext $C = (c_1, c_2, c_3, c_4)$ is rejected and algorithm $\mathcal{E}$ aborts, if $c_1 \neq c_1^*$ and $\theta = \theta^*$ where $\theta = H_1(c_1)$. According to the definition of target collision resistant hash function, the probability of this rejection case is at most some negligible value ε_1. Therefore, according to Lemma 4.1, we have $|\Pr[\Phi_3] - \Pr[\Phi_2]| \leq \varepsilon_1$.

Game $\mathcal{G}_4$: (Eliminate the target collision of H_2). Game $\mathcal{G}_4$ is similar to game $\mathcal{G}_3$ except that a decryption query for a valid ciphertext $C = (c_1, c_2, c_3, c_4)$ is rejected and algorithm $\mathcal{E}$ aborts, if $c_2 \neq c_2^*$ and $\vartheta = \vartheta^*$ where $\vartheta = H_2(c_1, c_2, c_3)$. According to the definition of target collision resistant hash function, the probability of this rejection case is at most some negligible value ε_2. Therefore, according to Lemma 4.1, we have $|\Pr[\Phi_4] - \Pr[\Phi_3]| \leq \varepsilon_2$.

Game $\mathcal{G}_5$: (Eliminate invalid ciphertext). Game $\mathcal{G}_5$ is similar to game $\mathcal{G}_4$ except that all decryption queries for invalid ciphertexts $C = (c_1, c_2, c_3, c_4)$ are rejected. The validity of C can be checked by

$$c_4 \stackrel{?}{=} c_1^{u\vartheta+v}$$

where $\vartheta = H_2(c_1, c_2, c_3)$. With a valid ciphertext C, the adversary may try to generate $C' = (c_1, c_2 m', c_3, c_4')$, where $m', c_4' \stackrel{\$}{\leftarrow} G_1$. The probability of C' not being rejected in decrypting the query is $1/p$. Therefore, according to Lemma 4.1, we have $|\Pr[\Phi_5] - \Pr[\Phi_4]| \leq \frac{q_D}{p}$.

Game $\mathcal{G}_6$: (Modify the challenge ciphertext). In this game, the element c_2^* is replaced by a random value $c_2' \in G_1$ such that C^* is a valid challenge ciphertext. If adversary $\mathcal{A}$ cannot distinguish $g_1^{\alpha^* x}$ from a random value in G_1, then game $\mathcal{G}_6$ is equivalent to game $\mathcal{G}_5$. Thus, we have $|\Pr[\Phi_6] - \Pr[\Phi_5]| \leq \varepsilon_{\text{sxdh}}$. Also, since c_2' is independent of b, it gives no information about b to adversary $\mathcal{A}$. Therefore, $\Pr[\Phi_6] = \frac{1}{2}$.

Combining the above results for games $\mathcal{G}_i$, we have

$$\varepsilon \leq \varepsilon_{\text{sxdh}} + \varepsilon_1 + \varepsilon_2 + \frac{2q_E}{p} + \frac{2q_D}{p}$$

This concludes Theorem 4.1. □

Theorem 4.2. *Suppose H_1 and H_2 are secure target collision-resistant hash functions. After authorization, the SEET scheme in Section 3.3 offers OW-CCA security for ciphertexts in the standard model assuming that the co-CDH* assumption and SXDH assumption hold.*

Proof. The following proof follows the standard frameworks established in [27,28,32].

Let $\varepsilon_{\text{sxdh}}$ be the advantage of a PPT adversary in deciding the SXDH problem, and $\varepsilon_{\text{cocdh}}$ be the advantage if a PPT adversary in solving the co-CDH* problem. Let ε_1 and ε_2 be the advantages of a PPT adversary in finding collisions in H_1 and H_2, respectively.

Let $\mathcal{A}$ be a PPT adversary that has non-negligible advantage ε in attacking the OW-CCA security of the SEET scheme. Note that in this phase, the adversary should hold the token tk. Suppose $\mathcal{A}$ issues at most q_E encryption queries and q_D decryption queries. We show that if such an adversary $\mathcal{A}$ exists, then one can construct an algorithm $\mathcal{E}$ to solve the co-CDH* problem with non-negligible probability.

Let $G_1 = \langle g \rangle, G_2, G_T$ be cycle groups with prime order p and bilinear map $\hat{e} : G_1 \times G_2 \to G_T$. Let H_1 and H_2 be two target collision resistant hash functions. At first, algorithm $\mathcal{E}$ is given a co-CDH* instance $(g_1^{\alpha^*}, g_1^x, g_2^{\alpha^*}, g_2^x) \in G_1^2 \times G_2^2$. The goal of $\mathcal{E}$ is to compute $g_1^{\alpha^* x}$. Algorithm $\mathcal{E}$ simulates the challenger and interacts with adversary $\mathcal{A}$ as follows.

Set-up: Algorithm $\mathcal{E}$ sets $c_1^* = g_1^{\alpha^*}$, computes $\theta^* = H_1(c_1^*)$, and randomly picks $u, v, w \stackrel{\$}{\leftarrow} Z_p^*$. The data owner's secret key sk_O is $(x, w - x\theta^*, u, v)$, where x and $y = w - x\theta^*$ are unknown to $\mathcal{E}$. Algorithm $\mathcal{E}$ randomly chooses $z \stackrel{\$}{\leftarrow} Z_p^*$, sets the token as $\mathsf{tk} \leftarrow (g_2^z, (g_2^x)^z)$, and gives tk to adversary $\mathcal{A}$.

Phase 1: The adversary adaptively makes the following queries.

- Encryption query: For input message m, algorithm $\mathcal{E}$ randomly picks $\alpha \stackrel{\$}{\leftarrow} Z_p^*$, and computes the ciphertext $C = (c_1, c_2, c_3, c_4)$ as follows:

  $$c_1 = g_1^\alpha, \theta = H_1(c_1), c_2 = m \cdot (g_1^x)^\alpha, c_3 = (g_1^x)^{\alpha(\theta-\theta^*)} g_1^{\alpha w}, \vartheta = H_2(c_1, c_2, c_3), c_4 = g_1^{\alpha(u\vartheta+v)}$$

  Note that $c_3 = (g_1^{x\theta+(w-x\theta^*)})^\alpha = (g_1^{x\theta+y})^\alpha$. $\mathcal{E}$ returns C as the ciphertext of m. Here if $c_1 = g_1^{\alpha^*}$ or $c_1 = g_1^x$, then algorithm $\mathcal{E}$ stops the simulation.
- Decryption query: For input ciphertext $C = (c_1, c_2, c_3, c_4)$, algorithm $\mathcal{E}$ first checks $c_1 \stackrel{?}{=} c_1^*$. If so, algorithm $\mathcal{E}$ aborts; otherwise, $\mathcal{E}$ computes

  $$\theta = H_1(c_1) \text{ and } \vartheta = H_2(c_1, c_2, c_3)$$

  and checks

  $$c_4 \stackrel{?}{=} c_1^{u\vartheta+v}$$

If the condition does not hold or $\theta = \theta^*$, then algorithm $\mathcal{E}$ outputs $\perp$; otherwise, $\mathcal{E}$ computes

$$m = c_2/(c_3 c_1^{-w})^{(\theta-\theta^*)^{-1}}$$

where $c_3 = (g_1^{x\theta} g_1^{y})^{\alpha}$ and $c_1^w = (g_1^w)^{\alpha} = (g_1^{x\theta^*} g_1^{y})^{\alpha}$. Then, $\mathcal{E}$ returns the message m.

Challenge: Algorithm $\mathcal{E}$ randomly picks $c_2^* \xleftarrow{\$} G_1$, and generates the challenge ciphertext $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ as follows:

$$c_1^* = g_1^{\alpha^*}, \theta^* = H_1(c_1^*), c_3^* = (g_1^{\alpha^*})^w, \vartheta^* = H_2(c_1^*, c_2^*, c_3^*), c_4^* = (g_1^{\alpha^*})^{u\vartheta^*+v}$$

Note that $c_3 = (g_1^{x\theta^*+y})^{\alpha^*}$. Then, algorithm $\mathcal{E}$ gives the challenge ciphertext C^* to $\mathcal{A}$.

Phase 2: The adversary can continue to make queries except that C^* cannot be submitted for decryption.

Guess: If algorithm $\mathcal{E}$ does not abort in the two query phases, then the simulation is perfect for OW-CCA2 attack and the adversary's view is the same as that in a real attack. Eventually, if adversary $\mathcal{A}$ outputs m', then algorithm $\mathcal{E}$ outputs $g_1^{\alpha^* x} = c_2^*/m'$ as the solution to the co-CDH* instance.

Analysis. We analyze the success probability of algorithm $\mathcal{E}$ through a sequence of games [26,28]. The first game $\mathcal{G}_0$ is the same as above but has no abortion during queries and m' is not involved in any queries; the subsequent games gradually consider abortion cases. Let Φ_i denote the event that m' is a correct output in game $\mathcal{G}_i$ for $0 \le i \le 6$. Thus, $\mathrm{Adv}^{\mathrm{ow-cca}}_{\Gamma,\mathcal{A}} = \varepsilon = \Pr[\Phi_0] = \varepsilon_{\mathrm{cocdh}}$.

Game $\mathcal{G}_1$: (Eliminate the correct guess of α^* and x). Game $\mathcal{G}_1$ is similar to game $\mathcal{G}_0$ except that α^* and x would not be chosen in answering encryption queries. Note that if either α^* or x is chosen, then the given co-CDH* problem is already solved by algorithm $\mathcal{E}$. Denote this abortion case by event $\mathbb{E}_1$. According to Lemmas 4.1 and 4.2, we have $|\Pr[\Phi_1] - \Pr[\Phi_0]| \le \Pr[\mathbb{E}_1]$ and $\Pr[\mathbb{E}_1] \le \frac{2q_E}{p}$.

Game $\mathcal{G}_2$: (Eliminate the correct guess of c_1). Game $\mathcal{G}_2$ is similar to game $\mathcal{G}_1$ except that a decryption query for ciphertext $C = (c_1, c_2, c_3, c_4)$ is rejected and algorithm $\mathcal{E}$ aborts, if $c_1 = c_1^*$. Denote this rejection case by event $\mathbb{E}_2$. According to Lemmas 4.1 and 4.3, we have $|\Pr[\Phi_2] - \Pr[\Phi_1]| \le \Pr[\mathbb{E}_2]$ and $\Pr[\mathbb{E}_2] \le \frac{q_D}{p}$.

Game $\mathcal{G}_3$: (Eliminate the correct guess of m'). If m' is involved in any encryption and decryption query, then the adversary is able to compare its ciphertext with the challenge ciphertext, which leaks the challenge message to the adversary. Thus, game $\mathcal{G}_3$ is similar to game $\mathcal{G}_2$ except that algorithm $\mathcal{E}$ aborts if m' is involved in any query. Denote this rejection case by event $\mathbb{E}_3$. According to Lemma 4.1, we have $|\Pr[\Phi_3] - \Pr[\Phi_2]| \le \Pr[\mathbb{E}_3]$, where $\Pr[\mathbb{E}_3] \le \frac{q_E+q_D}{p}$.

Game $\mathcal{G}_4$: (Eliminate the target collision of H_1). Game $\mathcal{G}_4$ is similar to game $\mathcal{G}_3$ except that a decryption query for a valid ciphertext $C = (c_1, c_2, c_3, c_4)$ is rejected and algorithm $\mathcal{E}$ aborts, if $c_1 \ne c_1^*$ and $\theta = \theta^*$ where $\theta = H_1(c_1)$. According to the definition of target collision resistant hash function, the probability of this rejection case is at most some negligible value ε_1. Therefore, according to Lemma 4.1, we have $|\Pr[\Phi_4] - \Pr[\Phi_3]| \le \varepsilon_1$.

Game $\mathcal{G}_5$: (Eliminate the target collision of H_2). Game $\mathcal{G}_5$ is similar to game $\mathcal{G}_4$ except that a decryption query for a valid ciphertext $C = (c_1, c_2, c_3, c_4)$ is rejected and algorithm $\mathcal{E}$ aborts, if $c_2 \ne c_2^*$ and $\vartheta = \vartheta^*$ where $\vartheta = H_2(c_1, c_2, c_3)$. According to the definition of target collision resistant hash function, the probability of this rejection case is at most some negligible value ε_2. Therefore, according to Lemma 4.1, we have $|\Pr[\Phi_5] - \Pr[\Phi_4]| \le \varepsilon_2$.

Game $\mathcal{G}_6$: (Eliminate invalid ciphertext). Game $\mathcal{G}_6$ is similar to game $\mathcal{G}_5$ except that all decryption queries for invalid ciphertexts $C = (c_1, c_2, c_3, c_4)$ are rejected. The validity of C can be checked by $c_4 \stackrel{?}{=} c_1^{u\vartheta+v}$ where $\vartheta = H_2(c_1, c_2, c_3)$. With a valid ciphertext C, the adversary may try to generate $C' = (c_1, c_2 m', c_3, c_4')$, where $m', c_4' \xleftarrow{\$} G_1$. The probability of C' not being rejected in decrypting the query is $1/p$. Therefore, according to Lemma 4.1, we have $|\Pr[\Phi_6] - \Pr[\Phi_5]| \le \frac{q_D}{p}$.

Combining the above results for games $\mathcal{G}_i$, we have

$$\varepsilon \le \varepsilon_{\mathrm{cocdh}} + \varepsilon_1 + \varepsilon_2 + \frac{3q_E}{p} + \frac{3q_D}{p}$$

This concludes Theorem 4.2. □

Theorem 4.3. *Suppose H_3 and H_4 are secure target collision-resistant hash functions. The SEET scheme in Section 3.3 offers IND-CCA security for tokens in the standard model assuming that the SXDH assumption holds.*

In the **Aut** procedure, g_2^x is in fact randomized by γ and encrypted by the following procedure:

$$t_1 = g_2^{\gamma}, t_2 = m\hbar_1^{\gamma}, t_3 = (\hbar_1^{\tilde{\theta}}\hbar_2)^{\gamma}, t_4 = (\hbar_3^{\tilde{\vartheta}}\hbar_4)^{\gamma}$$

Thus, proving Theorem 4.3 is equivalent to showing that the above public key encryption procedure is IND-CCA2 secure.

Proof. The following proof follows the standard framework established in [28].

Let $\varepsilon_{\mathrm{sxdh}}$ be the advantage of a PPT adversary in deciding the DDH problem on G_2. Let ε_3 and ε_4 be the advantages of a PPT adversary in finding collisions in H_3 and H_4, respectively.

Let $\mathcal{A}$ be a PPT adversary that has non-negligible advantage ε in attacking the IND-CCA2 security of the above scheme. Suppose $\mathcal{A}$ issues at most q_D decryption queries. We show that if such an adversary $\mathcal{A}$ exists, then one can construct an algorithm $\mathcal{E}$ to solve the SXDH problem with non-negligible probability.

Let $G_1 = \langle g \rangle, G_2, G_T$ be cyclic groups with prime order p and bilinear map $\hat{e} : G_1 \times G_2 \to G_T$. Also, let H_3 and H_4 be two target collision resistant hash functions. At first, algorithm $\mathcal{E}$ is given a SXDH instance $(g_2, g_2^{\gamma^*}, g_2^{\tilde{x}}, g_2^{\tilde{z}}) \in (G_2)^4$. The goal of $\mathcal{E}$ is to determine whether $g_2^{\gamma^* \tilde{x}} = g_2^{\tilde{z}}$. Algorithm $\mathcal{E}$ simulates the challenger and interacts with adversary $\mathcal{A}$ as follows.

Set-up: Algorithm $\mathcal{E}$ sets $\hbar_1 = g_2^{\tilde{x}}$ and $t_1^* = g_2^{\gamma^*}$, randomly picks $\tilde{u}, \tilde{v}, \tilde{w} \xleftarrow{\$} Z_p^*$ and computes

$$\tilde{\theta}^* = H_3(t_1^*), \hbar_2 = \hbar_1^{-\tilde{\theta}^*} g_2^{\tilde{w}}, \hbar_3 = g_2^{\tilde{u}}, \hbar_4 = g_2^{\tilde{v}}$$

Algorithm $\mathcal{E}$ gives the public key $\mathsf{pk}_T = (\hbar_1, \hbar_2, \hbar_3, \hbar_4)$ to the adversary. Note that the secret key sk_T is $(\tilde{x}, \tilde{w} - \tilde{x}\tilde{\theta}^*, \tilde{u}, \tilde{v})$, where $\tilde{x}$ and $\tilde{y} = \tilde{w} - \tilde{x}\tilde{\theta}^*$ are unknown to $\mathcal{E}$.

Phase 1: The adversary adaptively makes decryption queries. For input ciphertext $C = (t_1, t_2, t_3, t_4)$, algorithm $\mathcal{E}$ first checks whether $t_1 \stackrel{?}{=} t_1^*$. If so, algorithm $\mathcal{E}$ aborts; otherwise, $\mathcal{E}$ computes

$$\tilde{\theta} = H_3(t_1) \text{ and } \tilde{\vartheta} = H_4(t_1, t_2, t_3)$$

and checks whether

$$t_4 \stackrel{?}{=} t_1^{\tilde{u}\tilde{\vartheta} + \tilde{v}}$$

If it does not hold or $\tilde{\theta} = \tilde{\theta}^*$, then algorithm $\mathcal{E}$ outputs $\perp$; otherwise, $\mathcal{E}$ computes

$$m = t_2 / (t_3 t_1^{-\tilde{w}})^{(\tilde{\theta} - \tilde{\theta}^*)^{-1}}$$

where $t_3 = (g_2^{\tilde{x}\tilde{\theta}} g_2^{\tilde{y}})^{\gamma}$ and $t_1^{\tilde{w}} = (g_2^{\tilde{w}})^{\gamma} = (g_2^{\tilde{x}\tilde{\theta}^*} g_2^{\tilde{y}})^{\gamma}$. Then, $\mathcal{E}$ returns the message m.

Challenge: Adversary $\mathcal{A}$ picks two random messages m_0 and m_1, and gives them to $\mathcal{E}$. Then, algorithm $\mathcal{E}$ randomly picks $b \xleftarrow{\$} \{0, 1\}$, and generates the challenge ciphertext $C^* = (t_1^*, t_2^*, t_3^*, t_4^*)$ as follows:

$$t_1^* = g_2^{\gamma^*}, \theta^* = H_3(t_1^*), t_2^* = m_b g_2^{\tilde{z}}, t_3^* = (g_2^{\gamma^*})^{\tilde{w}}, \vartheta^* = H_4(t_1^*, t_2^*, t_3^*), t_4^* = (g_2^{\gamma^*})^{\tilde{u}\tilde{\vartheta}^* + \tilde{v}}$$

Note that $t_3 = (g_2^{\tilde{x}\tilde{\theta}^* + \tilde{y}})^{\gamma^*}$. Then, algorithm $\mathcal{E}$ returns the challenge ciphertext C^*.

Phase 2: The adversary can continue to make queries except that C^* cannot be submitted for decryption.

Guess: If algorithm $\mathcal{E}$ does not abort in the two query phases, then the simulation is perfect for CCA2 attack and the adversary's view is the same as that in a real attack. Eventually, adversary $\mathcal{A}$ returns a guess b'. Algorithm $\mathcal{E}$ outputs "1" if $b' = b$; otherwise, $\mathcal{E}$ outputs "0".

Analysis. We analyze the success probability of algorithm $\mathcal{E}$ through a sequence of games [26,28]. The first game $\mathcal{G}_0$ is the same as the above but has no abortion during queries; while the last one $\mathcal{G}_6$ gives no advantage to adversary $\mathcal{A}$. Let Φ_i denote the event that $b' = b$ in game $\mathcal{G}_i$ for $0 \le i \le 6$. Thus, $\mathsf{Adv}^{\mathsf{ind-cca}}_{\mathcal{A},\mathsf{token}} = |\Pr[\Phi_0] - \frac{1}{2}| = \varepsilon$.

Game $\mathcal{G}_1$: (Eliminate the correct guess of t_1). Game $\mathcal{G}_1$ is similar to game $\mathcal{G}_0$ except that a decryption query for ciphertext $C = (t_1, t_2, t_3, t_4)$ is rejected and algorithm $\mathcal{E}$ aborts, if $t_1 = t_1^*$. Denote this rejection case by event $\mathbb{E}_1$. According to Lemma 4.1, we have $|\Pr[\Phi_1] - \Pr[\Phi_0]| \le \Pr[\mathbb{E}_1]$.

Lemma 4.4. $\Pr[\mathbb{E}_1] \le \frac{q_D}{p}$.

The proof for Lemma 4.4 is the same as that for Lemma 4.3 and is omitted.

Game $\mathcal{G}_2$: (Eliminate the target collision of H_3). Game $\mathcal{G}_2$ is similar to game $\mathcal{G}_1$ except that a decryption query for a valid ciphertext $C = (t_1, t_2, t_3, t_4)$ is rejected and algorithm $\mathcal{E}$ aborts, if $t_1 \ne t_1^*$ and $\tilde{\theta} = \tilde{\theta}^*$ where $\tilde{\theta} = H_3(t_1)$. According to the definition of target collision resistant hash function, the probability of this rejection case is at most some negligible value ε_3. Therefore, by Lemma 4.1, we have $|\Pr[\Phi_2] - \Pr[\Phi_1]| \le \varepsilon_3$.

Game $\mathcal{G}_3$: (Eliminate the target collision of H_4). Game $\mathcal{G}_3$ is similar to game $\mathcal{G}_2$ except that a decryption query for a valid ciphertext $C = (t_1, t_2, t_3, t_4)$ is rejected and algorithm $\mathcal{E}$ aborts, if $t_2 \ne t_2^*$ and $\tilde{\vartheta} = \tilde{\vartheta}^*$ where $\tilde{\vartheta} = H_4(t_1, t_2, t_3)$. According to the definition of target collision resistant hash function, the probability of this rejection case is at most some negligible value ε_4. Therefore, by Lemma 4.1, we have $|\Pr[\Phi_3] - \Pr[\Phi_2]| \le \varepsilon_4$.

Game $\mathcal{G}_4$: (Eliminate invalid ciphertext). Game $\mathcal{G}_4$ is similar to game $\mathcal{G}_3$ except that all decryption queries for invalid ciphertexts $C = (t_1, t_2, t_3, t_4)$ are rejected. The validity of C can be checked by

$$t_4 \stackrel{?}{=} t_1^{\tilde{u}\tilde{\vartheta} + \tilde{v}}$$

where $\tilde{\vartheta} = H_4(t_1, t_2, t_3)$. With a valid ciphertext C, the adversary may try to generate $C' = (t_1, t_2 m', t_3, t_4')$ where $m', t_4' \xleftarrow{\$} G_2$. The probability of C' not being rejected in decrypting the query is $1/p$. Therefore, by Lemma 4.1, we have $|\Pr[\Phi_4] - \Pr[\Phi_3]| \le \frac{q_D}{p}$.

Game $\mathcal{G}_5$: (Modify the challenge ciphertext). In this game, element t_2^* is replaced by a random value $t_2' \in G_2$ such that C^* is a valid challenge ciphertext. If adversary $\mathcal{A}$ cannot distinguish $g_2^{\gamma^* \tilde{x}}$ from a random value in G_2, then game $\mathcal{G}_5$ is equivalent to game $\mathcal{G}_4$. Thus, we have $|\Pr[\Phi_5] - \Pr[\Phi_4]| \le \varepsilon_{\mathsf{sxdh}}$. Also, since t_2' is independent of b, it gives no information about b to adversary $\mathcal{A}$. Therefore, $\Pr[\Phi_5] = \frac{1}{2}$.

Combining the above results for games $\mathcal{G}_i$, we have

$$\varepsilon \leq \varepsilon_{\text{sxdh}} + \varepsilon_3 + \varepsilon_4 + \frac{2q_D}{p}$$

This concludes Theorem 4.3.

□

4.2. Comparison

We analyze our SEET scheme and compare it with existing encryption schemes supporting equality test on ciphertexts. The comparison is summarized in Table 1 in terms of ciphertext size, computation costs, ciphertext/token security, etc., where security-1/-2 denotes ciphertext security in the phases before/after authorization, and security-3 denote the security of token against Type-3 adversary.

In the table, s_G denotes the element size in G for a symmetric bilinear map $\hat{e} : G \times G \to G_T$; also, E_G and $E_{\hat{e}}$ represent the evaluation costs of an exponentiation in G and a bilinear map $\hat{e}(\cdot,\cdot)$ for this symmetric $\hat{e}$, respectively. Similarly, s_{G_1} and s_{G_2} respective denote the element sizes in G_1 and G_2 for an asymmetric bilinear map $\hat{e} : G_1 \times G_2 \to G_T$; while E_{G_1}, E_{G_2} and $E_{a\hat{e}}$ represent the evaluation costs of an exponentiation in G_1 and G_2 and a bilinear map $\hat{e}(\cdot,\cdot)$ for the asymmetric $\hat{e}$, respectively. We use s_Z and s_{G_T} to denote the element sizes of Z_p and G_T, respectively, for both types of bilinear maps. We also use E_{G_T} to denote the cost of an exponentiation on G_T. Moreover, $s_{\mathfrak{G}}$, $|\mathcal{M}|$ and ℓ respectively denote the size of an ordinary multiplicative cyclic group G, message space $\mathcal{M}$ and security parameter in Tang's schemes [29,30], while $E_{\mathfrak{G}}$ represent the cost of an exponentiation on this multiplicative group G. λ represents the security parameter in [16]. RO is the abbreviation for the random oracle model.

From the table, we see that our SEET scheme is the only one with IND-CCA/OW-CCA security for ciphertexts and IND-CCA security for token in the standard model. Compared to an IND-CCA secure variant of Slamanig et al.'s scheme [27], the encryption and decryption procedures in SEET are much more efficient since they only require 4 and 2 exponentiations in G_1, respectively. Moreover, a ciphertext in our scheme consists of only 4 elements in G_1, which is shorter than those in most existing schemes.

5. Application in controlled set distance computation

As our SEET construction enables controlled set operations/relations, it would also support many real-world applications that build upon set operations/relations. At the same time, the server is prevented from performing other computations. To exemplify such applications, we show in this section that our SEET scheme is applicable to controlled set distance computation, where three distance functions are considered. Suppose that two sets are encrypted by data owner (possibly at different times) under SEET and stored at a server. Subsequently, the data owner can authorize the server to compare the ciphertexts in these two sets. In this way, the server is able to compute the *cardinalities* of operation results of these datasets in ciphertext, e.g., intersection and union.

5.1. Review of dissimilarity functions on sets

Let $\mathbf{A} = \{a_1, a_2, \cdots, a_m\}$ and $\mathbf{B} = \{b_1, b_2, \cdots, b_n\}$ be two sets. The distance between $\mathbf{A}$ and $\mathbf{B}$ is determined by their similarity/dissimilarity score. Here, we review three similarity/dissimilarity functions between $\mathbf{A}$ and $\mathbf{B}$, where the sets concerned may characterize the users' profile in implicit authentication [8] or interests/preferences in social networks and e-commerce systems [2]. These studies showed that the similarities/dissimilarities in all these cases are really determined by the cardinality of the intersection on sets $\mathbf{A}$ and $\mathbf{B}$, or their variant sets. Since the datasets need to be encrypted to protect against the curious server, the server must be able to compute $|\mathbf{A} \cap \mathbf{B}|$ without knowing the real values of their entries.

Case 1: Both sets $\mathbf{A}$ and $\mathbf{B}$ contain independent nominal values, which are binary and the relationship between two values is equality or nothing. The dissimilarity between $\mathbf{A}$ and $\mathbf{B}$ is inversely proportional to the cardinality of their intersection, that is:

$$\text{Dsm}(\mathbf{A}, \mathbf{B}) = \begin{cases} 1/|\mathbf{A} \cap \mathbf{B}| & \text{if } \mathbf{A} \cap \mathbf{B} \neq \varnothing \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

Case 2: Both sets $\mathbf{A}$ and $\mathbf{B}$ comprise qualitative values, where the values may be correlated. Let $\ell : E \times E \to Z^+$ be the correlation function between the values in $\mathbf{A}$ and $\mathbf{B}$, where E denotes their domain. Note that case 1 is a special case such that $\ell(a_i, b_j) = 1$ if $a_i = b_j$, and 0 otherwise. Suppose both the server and data owner know the function ℓ. The dissimilarity between $\mathbf{A}$ and $\mathbf{B}$ is defined as follows:

$$\text{Dsm}(\mathbf{A}, \mathbf{B}) = \begin{cases} 1/\left(\sum_{a_i \in \mathbf{A}} \sum_{b_j \in \mathbf{B}} \ell(a_i, b_j)\right) & \text{if } \sum_{a_i \in \mathbf{A}} \sum_{b_j \in \mathbf{B}} \ell(a_i, b_j) \neq 0 \\ \infty & \text{otherwise} \end{cases} \tag{3}$$

For every $\pi \in E$, $\ell_\pi = \sum_{b_j \in \mathbf{B}} \ell(\pi, b_j)$ reflects the overall similarity between π and $\mathbf{B}$. Thus, Equality (3) equals to:

$$\mathsf{Dsm}(\mathbf{A}, \mathbf{B}) = \begin{cases} 1/\sum_{a_i \in \mathbf{A}} \ell_{a_i} & \text{if } \sum_{a_i \in \mathbf{A}} \ell_{a_i} \neq 0 \\ \infty & \text{otherwise} \end{cases} \tag{4}$$

Case 3: Both sets $\mathbf{A}$ and $\mathbf{B}$ comprise the same number of numerical feature values. Suppose $|\mathbf{A}| = |\mathbf{B}| = m$, the dissimilarity between A and B is evaluated as follows:

$$\mathsf{Dsm}(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{m} |a_i - b_i| \tag{5}$$

Defining

$$\mathbf{A}' = \{(i, j) : a_i \in \mathbf{A}, a_i > 0 \text{ and } 1 \leq j \leq a_i\}$$

and

$$\mathbf{B}' = \{(i, j) : b_i \in \mathbf{B}, b_i > 0 \text{ and } 1 \leq j \leq b_i\},$$

we have

$$|\mathbf{A}' \cap \mathbf{B}'| = |\{(i, j) : a_i > 0, b_i > 0 \text{ and } 1 \leq j \leq \min\{a_i, b_i\}\}| = \sum_{i=1}^{m} \min\{a_i, b_i\}$$

It further implies:

$$\mathsf{Dsm}(\mathbf{A}, \mathbf{B}) = |\mathbf{A}'| + |\mathbf{B}'| - 2|\mathbf{A}' \cap \mathbf{B}'| \tag{6}$$

since

$$\begin{aligned} |\mathbf{A}'| + |\mathbf{B}'| - 2|\mathbf{A}' \cap \mathbf{B}'| &= \sum_{i=1}^{m} (\max\{a_i, b_i\} + \min\{a_i, b_i\}) - 2\sum_{i=1}^{m} \min\{a_i, b_i\} \\ &= \sum_{i=1}^{m} (\max\{a_i, b_i\} - \min\{a_i, b_i\}) = \sum_{i=1}^{m} |a_i - b_i| \end{aligned}$$

In the remainder of this section, let Γ be a SEET scheme introduced in Section 3.3. We present privacy-preserving schemes for controlled set distance computation based on Γ.

5.2. Privacy-preserving scheme for case 1

Set-up: The system runs $\Gamma.\mathbf{Setup}(\lambda)$ to produce $\mathsf{pp}_\Gamma = (\hat{e}, G_1, G_2, G_T, g_1, g_2, p, H_1, H_2, H_3, H_4)$ and chooses a cryptographic hash function H': $\{0, 1\}^* \to G_1$. Thus, the public parameter is $\mathsf{pp} = \mathsf{pp}_\Gamma \cup \{H'\}$. With pp, the data owner invokes $\Gamma.\mathbf{KGen}_O(\mathsf{pp}_\Gamma)$ to get a secret key $\mathsf{sk}_O = (x, y, u, v)$, and the server runs $\Gamma.\mathbf{KGen}_T(\mathsf{pp}_\Gamma)$ to obtain a pair of secret and public keys, that is, $\mathsf{sk}_T = (\tilde{x}, \tilde{y}, \tilde{u}, \tilde{v})$ and $\mathsf{pk}_T = (\hbar_1, \hbar_2, \hbar_3, \hbar_4)$.

Data processing: For every $a_i \in \mathbf{A}$, the data owner runs the encryption procedure

$$C_i^{(a)} = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}) \leftarrow \Gamma.\mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}_\Gamma, H'(a_i)).$$

Similarly, every element $b_j \in \mathbf{B}$ is encrypted as follows

$$C_j^{(b)} = \left(c_{j,1}, c_{j,2}, c_{j,3}, c_{j,4}\right) \leftarrow \Gamma.\mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}_\Gamma, H'(b_j)).$$

The data owner sends $\mathbb{A} = \{C_i^{(a)} : a_i \in \mathbf{A}\}$ and $\mathbb{B} = \{C_j^{(b)} : b_j \in \mathbf{B}\}$ to the server.

Authorization: The data owner runs $\mathsf{etk} \leftarrow \Gamma.\mathbf{Aut}(\mathsf{sk}_O, \mathsf{pp}_\Gamma, \mathsf{pk}_T)$ and gives the encrypted token etk to the server. Then, the server is able to recover the token tk by running $\Gamma.\mathbf{Rec}(\mathsf{sk}_T, \mathsf{pp}_\Gamma, \mathsf{etk})$.

Distance computation: With the token tk, the server computes $|\mathbf{A} \cap \mathbf{B}|$ as follows:

$$|\mathbf{A} \cap \mathbf{B}| = \left|\{C_j^{(b)} : \Gamma.\mathbf{Com}(\mathsf{pp}_\Gamma, \mathsf{tk}, C_i^{(a)}, C_j^{(b)}) = 1 \text{ for every } C_i^{(a)} \in \mathbb{A} \text{ and } C_j^{(b)} \in \mathbb{B}\}\right|$$

By procedure $\Gamma.\mathbf{Com}$, for every $b_j \in \mathbf{B}$, if $b_j = a_i$ for some $a_i \in \mathbf{A}$, then its ciphertext $C_j^{(b)}$ must be counted in $|\mathbf{A} \cap \mathbf{B}|$. Finally, the server determines the distance (dissimilarity) between sets $\mathbf{A}$ and $\mathbf{B}$ according to $\mathsf{Dsm}(\mathbf{A}, \mathbf{B})$ as defined in Equality (2) using $|\mathbf{A} \cap \mathbf{B}|$.

Theorem 5.1. *The above scheme from SEET for case 1 is correct.*

Proof. In the data processing procedure, the data owner provides her encrypted datasets $\mathbb{A}$ and $\mathbb{B}$ to the server, which cannot be decrypted by the server. In the authorization phase, the encrypted token etk is given to the server, which can

be correctly recovered as guaranteed by Theorem 3.1. Also, following Theorem 3.1 for the soundness of SEET, with $\mathtt{tk}$, the server is able to compare every pair of ciphertext in $\mathbb{A}$ and ciphertext in $\mathbb{B}$ to infer $|\mathbf{A} \cap \mathbf{B}|$. Thus, correctness follows. □

Theorem 5.2. *The above scheme from SEET for case 1 is secure against an honest-but-curious server.*

Proof. In the data processing phase, the server only gets the encrypted values in $\mathbb{A}$ and $\mathbb{B}$, which security is guaranteed by Theorem 4.1. With the token $\mathtt{tk}$ recovered from $\mathtt{etk}$, the server runs the **Com** procedure in Γ to calculate $\mathsf{Dsm}(\mathbf{A}, \mathbf{B})$. According to Theorem 4.2, the procedure Γ.**Com** does not leak the data owner's real data values to the server, whether or not their ciphertexts are matched by Γ.**Com**. □

Theorem 5.3. *The authorization procedure in the above scheme from SEET for case 1 is secure against a malicious user on the communication channel.*

Proof. According to Theorem 4.3, the encrypted token $\mathtt{etk}$ cannot be modified by anyone who intercepts the communication between the data owner and server, otherwise $\mathtt{etk}$ would not go through the recovery procedure under the server's secret key. That also means such adversary cannot get the token $\mathtt{tk}$. □

5.3. Privacy-preserving scheme for case 2

Recall that for every $\pi \in E$, $\ell_\pi = \sum_{b_j \in \mathbf{B}} \ell(\pi, b_j)$ (see Section 5.1). Defining $\tilde{\mathbf{B}} = \{\pi \in E : \ell_\pi > 0\}$, we have $\mathbf{B} \subseteq \tilde{\mathbf{B}}$ since $\ell(\pi, \pi) > 0$ for every $\pi \in E$. The scheme for controlled set distance computation is designed as follows, where the data owner produces ℓ_π ciphertexts for every $\pi \in \tilde{\mathbf{B}}$.

Set-up: Same as in Section 5.2.

Data processing: Every $a_i \in \mathbf{A}$ is encrypted in the same way as in Section 5.2. For dataset $\mathbf{B}$, the data owner constructs $\tilde{\mathbf{B}}$. For every $\pi_j \in \tilde{\mathbf{B}}$, the data owner runs the encryption procedure $\Gamma.\mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}_\Gamma, \pi_j)$ for ℓ_{π_j} times, and gets $\{C_{j,l}^{(b)} = (c_{j,l,1}, c_{j,l,2}, c_{j,l,3}, c_{j,l,4}) : 1 \le l \le \ell_{\pi_j}\}$. Then, the data owner gives $\mathbb{A} = \{C_i^{(a)} : a_i \in \mathbf{A}\}$ and $\mathbb{B} = \{C_{j,l}^{(b)} : \pi_j \in \tilde{\mathbf{B}}, 1 \le l \le \ell_\pi\}$ to the server.

Authorization: Same as in Section 5.2.

Distance computation: With the token $\mathtt{tk}$, the server computes:

$$\sum_{a_i \in \mathbf{A}} \ell_{a_i} = \left|\{C_{j,l}^{(b)} : \Gamma.\mathbf{Com}(\mathsf{pp}_\Gamma, \mathtt{tk}, C_i^{(a)}, C_{j,l}^{(b)}) = 1 \text{ for every } C_i^{(a)} \in \mathbb{A} \text{ and } C_{j,l}^{(b)} \in \mathbb{B}\}\right|$$

Note that for every $\pi_j \in \tilde{\mathbf{B}}$, if $\pi_j = a_i$ for some $a_i \in \mathbf{A}$, then all of its ℓ_{π_j} ciphertexts would be matched with $C_i^{(a)}$ by the procedure Γ.**Com**. Finally, the server determines the distance (dissimilarity) between sets $\mathbf{A}$ and $\mathbf{B}$ according to $\mathsf{Dsm}(\mathbf{A}, \mathbf{B})$ as defined in Equality (4).

The correctness for the above scheme for case 2 is straightforward, and Theorem 5.3 is also achieved. Similar to Theorem 5.2, we have the following corollary.

Corollary 1. *The above scheme from SEET for case 2 is secure against an honest-but-curious server.*

5.4. Privacy-preserving scheme for case 3

As discussed in Section 5.1, the distance computation is performed on sets $\mathbf{A}' = \{(i, j) : a_i \in \mathbf{A}, a_i > 0 \text{ and } 1 \le j \le a_i\}$ and $\mathbf{B}' = \{(i, j) : b_i \in \mathbf{B}, b_i > 0 \text{ and } 1 \le j \le b_i\}$, rather than $\mathbf{A}$ and $\mathbf{B}$ directly.

Set-up: Same as in Section 5.2.

Data processing: The data owner constructs $\mathbf{A}'$ and $\mathbf{B}'$ from $\mathbf{A}$ and $\mathbf{B}$, respectively. Every pair $(i, j) \in \mathbf{A}'$ is encrypted as follows

$$C_{i,j}^{(a)} = \left(c_{i,j,1}^{(a)}, c_{i,j,2}^{(a)}, c_{i,j,3}^{(a)}, c_{i,j,4}^{(a)}\right) \leftarrow \Gamma.\mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}_\Gamma, H'(i \| j)).$$

Similarly, every $(i, j) \in \mathbf{B}'$ is encrypted as follows

$$C_{i,j}^{(b)} = \left(c_{i,j,1}^{(b)}, c_{i,j,2}^{(b)}, c_{i,j,3}^{(b)}, c_{i,j,4}^{(b)}\right) \leftarrow \Gamma.\mathbf{Enc}(\mathsf{sk}_O, \mathsf{pp}_\Gamma, H'(i \| j)).$$

Then, the data owner gives $\mathbb{A} = \{C_{i,j}^{(a)} : (i, j) \in \mathbf{A}'\}$ and $\mathbb{B} = \{C_{i,j}^{(b)} : (i, j) \in \mathbf{B}'\}$ to the server.

Authorization: Same as in Section 5.2.

Distance computation: With the token $\mathtt{tk}$, the server computes $|\mathbf{A}' \cap \mathbf{B}'|$ as follows:

$$|\mathbf{A}' \cap \mathbf{B}'| = \left|\{C_{i,j}^{(b)} : \Gamma.\mathbf{Com}(\mathsf{pp}_\Gamma, \mathtt{tk}, C_{i,j}^{(a)}, C_{i,j}^{(b)}) = 1 \text{ for every } C_{i,j}^{(a)} \in \mathbb{A} \text{ and } C_{i,j}^{(b)} \in \mathbb{B}\}\right|$$

Then, the server calculates $\mathsf{Dsm}(\mathbf{A}, \mathbf{B})$ as defined in Equality (5).

The correctness of the above scheme for case 3 is straightforward, and Theorem 5.3 is also achieved. Similar to Theorem 5.2, we have the following corollary.

Corollary 2. *The above scheme from SEET for case 3 is secure against an honest-but-curious server.*

Table 2
Performance for controlled set distance computation for three cases.

		Section 5.2	Section 5.3	Section 5.4
Data process	Computation cost	$(4m+4n)E_{G_1}$	$(4m+4\delta)E_{G_1}$	$8m\hat{a}E_{G_1}$
	Storage cost	$(4m+4n)s_{G_1}$	$(4m+4\delta)s_{G_1}$	$8m\hat{a}s_{G_1}$
Authorization	Computation cost (user)	$7E_{G_2}$	$7E_{G_2}$	$7E_{G_2}$
	Computation cost (server)	$2E_{G_2}$	$2E_{G_2}$	$2E_{G_2}$
	Communication cost	$4s_{G_2}$	$4s_{G_2}$	$4s_{G_2}$
Distance computation	Computation cost	$2\eta E_{a\hat{e}}$	$2m\delta E_{a\hat{e}}$	$2\eta' E_{a\hat{e}}$

5.5. Performance analysis

In this section, we analyze our proposed schemes for controlled set distance computation in terms of computation, storage and communication costs. For computation cost, we only consider time-consuming operations such as exponentiations and bilinear pairings, whereas lightweight computations including additions, multiplications and hash evaluations are omitted. The analyzes are summarized in Table 2 for the three cases. In the table, the notations $s_{G_1}, s_{G_2}, E_{G_1}, E_{G_2}, E_{a\hat{e}}$ have the same meanings as in Table 1.

We first consider the scheme for case 1 (see Section 5.2). In the data processing phase, the user needs to encrypt $a_i \in \mathbf{A}$ and $b_j \in \mathbf{B}$ with the SEET scheme, which requires $(4m+4n)$ exponentiations on G_1. The server will store all ciphertexts in $\mathbb{A}$ and $\mathbb{B}$ which contain in total $(4m+4n)$ group elements in G_1. The authorization procedure allows the user to encrypt a token for the server to decrypt on group G_2, which takes 7 and 2 exponentiations on G_2, respectively. The encrypted token is 4 elements in G_2, which is transmitted over the channel. (Note that the authorization procedure is the same for all three cases and will not be considered again in the following for the other two cases.) Suppose $n \le m$ and let

$$\eta = \lceil \frac{m}{2} \rceil + \lceil \frac{m-1}{2} \rceil + \cdots + \lceil \frac{m-n+1}{2} \rceil \approx mn - (n^2-n)/2$$

In the distance computation phase, the server will compare η pairs of ciphertexts in $\mathbb{A}$ and $\mathbb{B}$ on average.

Case 2 is similar to case 1, except that the user and server deal with a newly constructed set $\tilde{\mathbf{B}}$ rather than $\mathbf{B}$. The computation and communication complexities for the worst case happen when $\tilde{\mathbf{B}} = E$ and the correlation function ℓ takes the maximum value in its range L. Let $\delta = \sum_{\pi_j \in \tilde{\mathbf{B}}} \ell_{\pi_j} = n|E|L$. Thus, the user and server will deal with $(m+\delta)$ values. For case 3, all computations are on sets $\mathbf{A}'$ and $\mathbf{B}'$ that are constructed from $\mathbf{A}$ and $\mathbf{B}$, respectively, where $|\mathbf{A}| = |\mathbf{B}| = m$. Suppose all the a_i and b_j values take the maximum value $\hat{a}$ in domain E, which implies the worst case such that $|\mathbf{A}'|$ is maximum and $|\mathbf{A}'| = |\mathbf{B}'| = m\hat{a}$. The other discussions are similar to case 1, except that the different sets $\mathbf{A}'$ and $\mathbf{B}'$ with $m\hat{a}$ elements are involved and

$$\eta' = \lceil \frac{m\hat{a}}{2} \rceil + \lceil \frac{m\hat{a}-1}{2} \rceil + \cdots + 1 \approx m^2\hat{a}^2/8 + m\hat{a}/4$$

5.6. Extension to implicit authentication

Implicit authentication allows a server to authenticate a user based on his/her usage profile [8,11,25], which can be used to enhance existing knowledge-based authentication systems. In the initialization phase, the user deposits his/her profile with the server, which enables the server to validate the user in the authentication phase based on the similarity/dissimilarity score between the stored profile and the current one. Implicit authentication is different from two-factor authentication [20] in that the latter requires additional hardware in the system as second authentication factor, which would impose additional cost on users.

Following Shahandashti and co-workers [8,25], in an implicit authentication system, the server keeps a set of the user's profile features that consists of the history of the user's action on the device, and frequently interacts with the user to do profile update. The server is honest-but-curious and may be interested in the details of the user profile. The device serves as a data source in the system, which always honestly collects the user's profile features. The server can perform implicit authentication on the user. In doing so, the user together with the device provide the latest profile feature values to the server, so that the server is able to make an authentication decision by comparing with the stored user profile.

Domingo-Ferrer, Wu and Blanco-Justicia [8] presented privacy-preserving implicit authentication schemes for three types of user profile. In fact, our three schemes in Sections 5.2–5.4 for controlled set distance computation from SEET can be transformed into more efficient *symmetric* solutions to the same problem in [8]. In our symmetric solutions, both phases of user profile initialization and authentication apply the *same* encryption technique (i.e., SEET scheme). In the initialization phase, to enable the server to authenticate the user subsequently, the user also provides a token generated by the **Aut** procedure in SEET to the server.

In the authentication phase, the token enables the server to compare the ciphertexts for the current profile feature values with the stored feature values in encrypted format and calculate an authentication score. To protect against replay attacks, in each round of implicit authentication, the server needs to sign a time stamp and send it along with the signature to the

user. If the pair passes the verification, then the device samples the current feature values, and the user encrypts them, signs the string $\mathsf{ciphertexts}\|\mathsf{timestamp}$, and gives the ciphertexts together with the signature to the server for verification and computing an authentication score. Note that in the authentication phase, the server does not give any information on the stored profile to the user, which avoids leaking information on the stored profile to a malicious user. Since each feature value is separately encrypted, our implicit authentication solutions support efficient profile update without involving all feature values. The transformed solutions are straightforward, thus we do not go into details here.

6. Conclusion

In this paper, we studied the problem of controlled operations on sets that are hosted by a curious server. The server can operate on encrypted sets only upon getting an authorization from the data owner, and the authorization is controlled in the sense that only the designated server can recover a token to perform computations without decrypting data sets. To address the problem, we introduced the notion of secure encryption with authorized equality test (SEET) and formalized its security model to capture two phases of data confidentiality along with token privacy. We proposed a SEET construction from Type 3 bilinear pairing, and formally proved its security against three types of adversary without using random oracles as defined in our security model. An efficiency analysis demonstrated that our SEET construction outperforms existing (public/private key and identity-based) encryption schemes that support equality test on ciphertexts. As exemplary applications of SEET, we also developed efficient approaches for controlled set distance computation, and further discussed their application in implicit authentication system.

Acknowledgment

This article is based on research work supported by the Singapore National Research Foundation under NCR Award Number NRF2014NCR-NCR001-012.

References

[1] A. Abadi, S. Terzis, C. Dong, O-PSI: delegated private set intersection on outsourced datasets, in: H. Federrath, D. Gollmann (Eds.), ICT Systems Security and Privacy Protection: 30th IFIP TC 11 International Conference, SEC 2015, Hamburg, Germany, May 26–28, 2015, Proceedings, Springer International Publishing, Cham, 2015, pp. 3–17, doi:10.1007/978-3-319-18467-8_1.
[2] A. Blanco, J. Domingo-Ferrer, O. Farràs, D. Sánchez, Distance computation between two private preference functions, in: N. Cuppens-Boulahia, F. Cuppens, S. Jajodia, A. Abou El Kalam, T. Sans (Eds.), ICT Systems Security and Privacy Protection: 29th IFIP TC 11 International Conference, SEC 2014, Marrakech, Morocco, June 2–4, 2014. Proceedings, Springer Berlin Heidelberg, 2014, pp. 460–470, doi:10.1007/978-3-642-55415-5_39.
[3] D. Boneh, X. Boyen, Efficient selective-ID secure identity-based encryption without random oracles, in: C. Cachin, J.L. Camenisch (Eds.), Advances in Cryptology – EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2–6, 2004. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 223–238, doi:10.1007/978-3-540-24676-3_14.
[4] D. Boneh, E. Shen, B. Waters, Strongly unforgeable signatures based on computational Diffie–Hellman, in: M. Yung, Y. Dodis, A. Kiayias, T. Malkin (Eds.), Public Key Cryptography – PKC 2006: 9th International Conference on Theory and Practice in Public-Key Cryptography, New York, NY, USA, April 24–26, 2006. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 229–240, doi:10.1007/11745853_15.
[5] S. Chatterjee, A. Menezes, On cryptographic protocols employing asymmetric pairings—the role of ψ revisited, Discrete Appl. Math. 159 (13) (2011) 1311–1322. http://dx.doi.org/10.1016/j.dam.2011.04.021.
[6] R. Cramer, V. Shoup, Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack, SIAM J. Comput. 33 (1) (2003) 167–226, doi:10.1137/S0097539702403773.
[7] C. Delerablée, D. Pointcheval, Dynamic fully anonymous short group signatures, in: P.Q. Nguyen (Ed.), Progress in Cryptology – VIETCRYPT 2006: First International Conference on Cryptology in Vietnam, Hanoi, Vietnam, September 25–28, 2006., Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 193–210, doi:10.1007/11958239_13.
[8] J. Domingo-Ferrer, Q. Wu, A. Blanco-Justicia, Flexible and robust privacy-preserving implicit authentication, in: H. Federrath, D. Gollmann (Eds.), ICT Systems Security and Privacy Protection: 30th IFIP TC 11 International Conference, SEC 2015, Hamburg, Germany, May 26–28, 2015, Proceedings, Springer International Publishing, Cham, 2015, pp. 18–34.
[9] T. Elgamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Inf. Theory 31 (4) (1985) 469–472, doi:10.1109/TIT.1985.1057074.
[10] K. Huang, R. Tso, Y.-C. Chen, S.M.M. Rahman, A. Almogren, A. Alamri, PKE-AET: public key encryption with authorized equality test, Comput. J. 58 (10) (2015) 2686–2697.
[11] M. Jakobsson, E. Shi, P. Golle, R. Chow, Implicit authentication for mobile devices, in: Proceedings of the 4th USENIX Conference on Hot Topics in Security, in: HotSec'09, USENIX Association, Berkeley, CA, USA, 2009.
[12] F. Kerschbaum, Collusion-resistant outsourcing of private set intersection, in: Proceedings of the 27th Annual ACM Symposium on Applied Computing, in: SAC'12, ACM, New York, NY, USA, 2012, pp. 1451–1456.
[13] F. Kerschbaum, Outsourced private set intersection using homomorphic encryption, in: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, in: ASIACCS'12, ACM, New York, NY, USA, 2012, pp. 85–86.
[14] H.T. Lee, S. Ling, J.H. Seo, H. Wang, CCA2 Attack and modification of Huang et al.'s public key encryption with authorized equality test, Comput. J. 59 (11) (2016) 1689–1694, doi:10.1093/comjnl/bxw033.
[15] H.T. Lee, S. Ling, J.H. Seo, H. Wang, Semi-generic construction of public key encryption and identity-based encryption with equality test, Inf. Sci. 373 (2016) 419–440. http://dx.doi.org/10.1016/j.ins.2016.09.013.
[16] H.T. Lee, S. Ling, J.H. Seo, H. Wang, T.-Y. Youn, Public key encryption with equality test in the standard model, 2016.
[17] S. Ma, Identity-based encryption with outsourced equality test in cloud computing, Inf. Sci. 328 (2016) 389–402. http://dx.doi.org/10.1016/j.ins.2015.08.053.
[18] S. Ma, Q. Huang, M. Zhang, B. Yang, Efficient public key encryption with equality test supporting flexible authorization, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 458–470.
[19] S. Ma, M. Zhang, Q. Huang, B. Yang, Public key encryption with delegated equality test in a multi-user setting, Comput. J. 58 (4) (2015) 986–1002.
[20] C. Mann, D. Loebenberger, Two-factor authentication for the bitcoin protocol, Int. J. Inf. Secur. 16 (2) (2017) 213–226, doi:10.1007/s10207-016-0325-1.
[21] M. Naor, M. Yung, Public-key cryptosystems provably secure against chosen ciphertext attacks, in: Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing, in: STOC'90, ACM, New York, NY, USA, 1990, pp. 427–437, doi:10.1145/100216.100273.

[22] L. Nguyen, R. Safavi-Naini, Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings, in: P.J. Lee (Ed.), Advances in Cryptology – ASIACRYPT 2004: 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5–9, 2004. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 372–386, doi:10.1007/978-3-540-30539-2_26.
[23] H. Pang, X. Ding, Privacy-preserving ad-hoc equi-join on outsourced data, ACM Trans. Database Syst. 39 (3) (2014) 23:1–23:40, doi:10.1145/2629501.
[24] M.R. Reyhanitabar, W. Susilo, Y. Mu, Enhanced target collision resistant hash functions revisited, in: O. Dunkelman (Ed.), Fast Software Encryption: 16th International Workshop, FSE 2009 Leuven, Belgium, February 22–25, 2009, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 327–344, doi:10.1007/978-3-642-03317-9_20.
[25] S.F. Shahandashti, R. Safavi-Naini, N.A. Safa, Reconciling user privacy and implicit authentication for mobile devices, Comput. Secur. 53 (C) (2015) 215–233.
[26] V. Shoup, Sequences of games: a tool for taming complexity in security proofs, 2004. (IACR Cryptology ePrint Archive).
[27] D. Slamanig, R. Spreitzer, T. Unterluggauer, Adding controllable linkability to pairing-based group signatures for free, in: S.S.M. Chow, J. Camenisch, L.C.K. Hui, S.M. Yiu (Eds.), Information Security: 17th International Conference, ISC 2014, Hong Kong, China, October 12–14, 2014. Proceedings, Springer International Publishing, Cham, 2014, pp. 388–400, doi:10.1007/978-3-319-13257-0_23.
[28] C.H. Tan, Secure public-key encryption scheme without random oracles, Inf. Sci. 178 (17) (2008) 3435–3442. http://dx.doi.org/10.1016/j.ins.2008.04.006.
[29] Q. Tang, Towards public key encryption scheme supporting equality test with fine-grained authorization, in: U. Parampalli, P. Hawkes (Eds.), Proceedings of Information Security and Privacy: 16th Australasian Conference, ACISP 2011, LNCS, 6812, Springer, Heidelberg, 2011, pp. 389–406.
[30] Q. Tang, Public key encryption supporting plaintext equality test and user-specified authorization, Secur. Commun. Netw. 5 (12) (2012) 1351–1362.
[31] Y. Wang, H. Pang, Probabilistic public key encryption for controlled equijoin in relational databases, Comput. J. 60 (4) (2017) 600–612, doi:10.1093/comjnl/bxw083.
[32] G. Yang, C.H. Tan, Q. Huang, D.S. Wong, Probabilistic public key encryption with equality test, in: J. Pieprzyk (Ed.), Topics in Cryptology - CT-RSA 2010, LNCS, 5985, Springer, Heidelberg, 2010, pp. 119–131, doi:10.1007/978-3-642-11925-5_9.