7-2019

Securing messaging services through efficient signcryption with designated equality test

Yujue WANG
Guilin University of Electronic Technology

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

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

Yong DING
Guilin University of Electronic Technology

Qianhong WU
Beijing University of Aeronautics and Astronautics (Beihang University)

See next page for additional authors

Citation

WANG, Yujue; PANG, Hwee Hwa; DENG, Robert H.; DING, Yong; WU, Qianhong; and QIN, Bo. Securing messaging services through efficient signcryption with designated equality test. (2019). *Information Sciences*. 490, 145-165.

Available at: https://ink.library.smu.edu.sg/sis_research/4349

Author
Yujue WANG, Hwee Hwa PANG, Robert H. DENG, Yong DING, Qianhong WU, and Bo QIN

7-2019

Securing messaging services through efficient signcryption with designated equality test

Yujue WANG

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

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

Yong DING

Qianhong WU

See next page for additional authors

DOI: https://doi.org/10.1016/j.ins.2019.03.039

Citation

WANG, Yujue; PANG, Hwee Hwa; DENG, Robert H.; DING, Yong; WU, Qianhong; and QIN, Bo. Securing messaging services through efficient signcryption with designated equality test. (2019). *Information Sciences*. 490, 145-165. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/4349

Author
Yujue WANG, Hwee Hwa PANG, Robert H. DENG, Yong DING, Qianhong WU, and Bo QIN

Published in Information Sciences,
Vol 490, July 2019, Pages 145–165
DOI 10.1016/j.ins.2019.03.039

Securing messaging services through efficient signcryption with designated equality test

Yujue Wang[a,b], HweeHwa Pang[c], Robert H. Deng[c], Yong Ding[a,d,*], Qianhong Wu[e,f], Bo Qin[g]

[a] Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
[b] State Key Laboratory of Cryptology, P. O. Box 5159, Beijing 100878, China
[c] School of Information Systems, Singapore Management University, 188065, Singapore
[d] Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518055, China
[e] School of Electronic and Cyber Science and Technology, Beihang University, Beijing 100191, China
[f] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[g] School of Information, Renmin University of China, Beijing 100872, China

ARTICLE INFO

Article history:
Received 27 March 2018
Revised 17 February 2019
Accepted 19 March 2019
Available online 22 March 2019

Keywords:
Messaging system
Encryption
Confidentiality
Authentication
Signcryption
Equality test on ciphertexts
Equijoin
Data outsourcing

ABSTRACT

To address security and privacy issues in messaging services, we present a *public key signcryption scheme with designated equality test on ciphertexts* (PKS-DET) in this paper. The scheme enables a sender to simultaneously encrypt and sign (signcrypt) messages, and to designate a tester to perform equality test on ciphertexts, i.e., to determine whether two ciphertexts signcrypt the same underlying plaintext message. We introduce the PKS-DET framework, present a concrete construction and formally prove its security against three types of adversaries, representing two security requirements on message confidentiality against outsiders and the designated tester, respectively, and a requirement on message unforgeability against the designated tester. We also present three extensions, analyze the efficiency of our PKS-DET construction and extensions, and compare them with related schemes in terms of ciphertext sizes and computation costs of signcryption (encryption), unsigncryption (decryption) and ciphertext equality testing. Experimental results further confirmed the practicality of our construction.

1. Introduction

With increased business and consumer awareness of data security and privacy, popular messaging services have built end-to-end encryption into their platforms in recent years. Yet, this privacy protection can and has been abused, for example for organizing coordinated attacks or spreading fake news. Consequently, the regulatory framework of many jurisdictions mandates a business or individual to release protected messages to law enforcement, on the ground of national/social interests or efficient operation of government functions. It would be desirable for a secure messaging platform to balance between privacy protection and law enforcement, by making the following provisions besides privacy and authentication:

[*] Corresponding author at: Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China.

E-mail address: stone_dingy@126.com (Y. Ding).

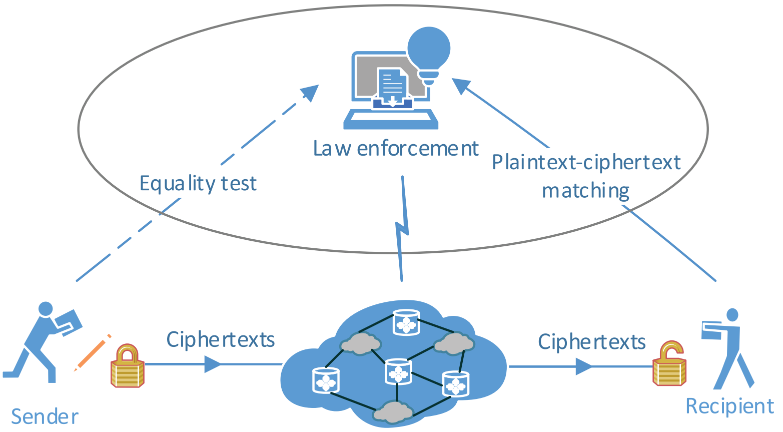

Fig. 1. Secure messaging system.

(a) For users to disclose only requisitioned messages to law enforcement without compromising other messages, as well as (b) for law enforcement to track the communication patterns of encrypted messages that are being monitored.

To the best of our knowledge, there is no existing solution that satisfies all the above requirements. Public key signcryption (PKS) schemes [52] concurrently sign and encrypt messages, thus providing data confidentiality and authentication; however, they do not support equality test on ciphertexts by a third party. On the other hand, existing public key encryption schemes with equality test on ciphertexts (PKEET) [26,31,32,43,46,47,50] do not support data authentication.

1.1. Contributions

In this paper, we present a *public key signcryption scheme with designated equality test on ciphertexts* (PKS-DET), in which a sender can signcrypt (i.e., simultaneously encrypt and sign) messages, and allow a designated tester to perform one or both of the following functions (see Fig. 1):

- Verify that a plaintext message surrendered by a user indeed corresponds to a specific ciphertext. With this function, a user only needs to surrender the requisitioned plaintext messages and law enforcement can verify that the messages are genuine; the user does not need to grant anyone access to her messaging device nor disclose her decryption key, thus keeping her remaining messages protected.
- Test the equality of two ciphertexts, i.e., determine whether two ciphertexts signcrypt the same underlying plaintext message. In a secure messaging service, the same message could be forwarded from user to user, or a user could retweet a message to her friends. The forwarded message or retweet would get re-encrypted to a different ciphertext each time. The ability to compare the plaintext messages behind two ciphertexts, without decrypting them, enables law enforcement to monitor traffic patterns to isolate messages that may warrant further investigation. Note that in a data outsourcing scenario, this functionality can also be delegated to a remote server, so that the server could execute a deduplication procedure on outsourced encrypted data to save storage space.

Our notion of PKS-DET combines the functionalities of PKS and PKEET. PKS-DET allows a sender to generate a ciphertext that simultaneously encrypts and signs a message, while designating a tester to perform equality test on ciphertexts, i.e., test the equality of the underlying plaintext messages of two ciphertexts. We formulate the security model of PKS-DET against *three* types of adversaries, representing two security requirements on message *confidentiality* and a requirement on message *unforgeability*. Specifically, the two message confidentiality requirements are IND-CCA2 security against an outsider and OW-CCA2 security against the designated tester, respectively, and the message unforgeability requirement is EU-CMA security against the designated tester.

We present a concrete PKS-DET construction on bilinear groups, which enables a designated tester to perform equality test on ciphertexts associated with arbitrary sender and recipient pairs. Moreover, the designated tester can match a surrendered plaintext message against a ciphertext without resorting to unsigncryption (which has the undesirable consequence of allowing the tester to unsigncrypt arbitrary ciphertexts), or sequentially signcrypting the message and testing the equality of ciphertexts which is inefficient (as shown in our experiments in Section 4.4, direct plaintext-ciphertext matching requires only 12 msec, compared with 17 msec for signcryption plus 18 msec for ciphertext-ciphertext matching). We then extend our PKS-DET construction to support longer and shorter messages, and to allow more flexible delegation on ciphertext equality test. The security of our construction is proved against the three types of adversaries as defined in our security framework in the random oracle model. Theoretical comparison with related schemes and experimental analysis demonstrate that our construction is practical in applications.

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

Scheme	Confidentiality			Authentication	Token
	Outsider	Tester (Prior to Aut)	Tester (After Aut)		
Yang et al. [50]	–	–	OW-CCA2	✗	–
Tang [41]	IND-CCA2	IND-CCA2	OW-CCA2	✗	Required
Tang [43]	IND-CCA2	IND-CCA2	OW-CCA2	✗	Required
Lee et al. [25]	IND-CCA2	IND-CCA2	OW-CCA2	✗	Required
Lee et al. [27]	IND-CCA2	IND-CCA2	OW-CCA2	✗	Required
Ma et al. [31]	IND-CCA2	IND-CCA2	OW-CCA2	✗	Required
Ma [30]	IND-CCA2	IND-CCA2	OW-ID-CCA2	✗	Required
Ma et al. [32]	IND-CCA2	IND-CCA2	OW-CCA2	✗	Required
Wang and Pang [46]	IND-CCA2	IND-CCA2	OW-CCA2	✗	Required
Slamanig et al. [40]	IND-CCA2	IND-CCA2	OW-CCA2	✗	Required
Wang et al. [47]	IND-CCA2	IND-CCA2	OW-CCA2	✗	Required
Pang and Ding [35]	IND-CPA	IND-CPA	OW-CPA	✗	Required
PKS-DET Section 3	IND-CCA2	–	OW-CCA2	√	No
Section 4.1(1)	IND-CCA2	–	OW-CCA2	√	No
Section 4.1(2)	IND-CCA2	–	OW-CCA2	√	No
Section 4.2	IND-CCA2	IND-CCA2	OW-CCA2	√	Required

1.2. Related work

The notion of public key encryption with equality test (PKEET) and a concrete construction on bilinear groups were introduced by Yang et al. [50]. PKEET allows *anyone* to verify whether two ciphertexts encrypt the same plaintext, even if they are generated with different public keys. Since then, the functionality of *authorized/delegable* equality test on ciphertexts has been incorporated into PKEET schemes [19,31,32,40,41,43], to enable an authorized/delegated tester to compare ciphertexts. Note that the authorized/delegated tester in [19,31,32,43] can be authorized by two users to compare ciphertexts from both users or ciphertexts from one of the users. The security properties of our PKS-DET construction and the existing PKEET schemes are summarized in Table 1, where 'Aut' denotes the authorization/delegation/designation of testing equality on ciphertexts, whereas '1' and '2' denote case 1 and case 2 of extensions of PKS-DET in Section 4.1, respectively.

Tang proposed an all-or-nothing PKEET (AoN-PKEET) [43], in which a tester can be authorized by two users, acting independently, to compare their ciphertexts. In [40], Slamanig et al. proposed an AoN-PKEET* construction on asymmetric bilinear groups based on the ElGamal encryption scheme [14], which is a special case of AoN-PKEET [43] in that the tester is only able to compare ciphertexts generated with the same public key. The AoN-PKEET* construction [40] only offers IND-CPA security for ciphertexts prior to the authorization of the tester; Table 1 refers to an enhanced construction that is IND-CCA2 secure in the random oracle model, derived with the approach in [13,33,34].

Ma [30] studied identity-based encryption supporting outsourced equality test on ciphertexts (IBEET), which combines the functionalities of PKEET and identity-based encryption (IBE). In [25], Lee et al. identified a flaw in the scheme proposed in [19] and provided a solution to enhance its security. Semi-generic constructions for PKEET and IBEET were given in [26]. Lee et al. [27] presented a generic PKEET construction based on 2-level hierarchical IBE and strongly unforgeable one-time signature without using random oracles, and extended it to the IBEET construction. They also noted in [27, Section 6.2] that a PKEET construction can be instantiated from [2] and [5] following their generic framework, which is the construction compared in Table 1. Wang et al. [47] proposed a scheme in Type-3 bilinear groups based on the ElGamal scheme, which guarantees the confidentiality of ciphertexts as well as tokens in standard model.

Pang and Ding [35] for the first time studied controlled equijoin in relational databases and proposed an IND-CPA secure construction in symmetric bilinear groups in the secret key setting. The idea behind [35] is equality test on encrypted fields in outsourced records. In [46], Wang and Pang presented a public key encryption for controlled equijoin in relational databases, which offers IND-CCA2 security for outsourced records before the tester gets the authorization token. Recently, ciphertext comparability was adopted in [9,48,49] to address the problem of deduplication on encrypted data, and in [45] to identify the same road condition reports in clouds.

To save the costs of separately signing and encrypting a message, Zheng [52] introduced the notion of public key signcryption (PKS). Since then, identity-based signcryption [1,6,7] and (threshold/dynamic) attribute-based signcryption [12,15,37,38,51], and their applications [36], have been extensively studied. Li et al. [28] presented a PKS scheme with confidentiality, existential unforgeability and anonymity, and extended it to a ring signcryption scheme. Huang et al. [20] designed a heterogeneous signcryption scheme, where the sender has an identity-based secret key while the recipient holds a certificate-based key pair. In [44], Wang et al. analyzed the security of two signcryption schemes [18,23], and found that [18] cannot provide confidentiality while [23] does not provide unforgeability, coalition-resistance, and traceability.

Herranz et al. [17] presented a PKS that supports threshold unsigncryption in a multi-user setting. Cui et al. [10,11] addressed the security issue regarding related-key attacks in PKS. In [24], Lai et al. presented an online/offline PKS scheme in the random oracle model, where the signcryption procedure is divided into two phases such that most computation can be pre-computed in the offline phase without knowing the plaintext to be signcrypted. Li et al. [29] designed a certificateless signcryption scheme to achieve access control in industrial wireless sensor networks. Karati et al. [21] used identity-based signcryption to secure crowdsourced industrial IoT data in clouds.

1.3. Paper organization

The remainder of this paper is organized as follows. In Section 2, we formulate the PKS-DET framework and the corresponding security requirements. We present a PKS-DET construction and prove its security in Section 3. In Section 4, we describe three extensions of our PKS-DET construction, compare their performance with those of existing schemes in the literature, and conduct experimental analysis. Finally, Section 5 concludes the paper.

2. PKS-DET framework and security definitions

A PKS-DET based secure messaging system should satisfy the following requirements:

- Designated equality test on ciphertexts: Only the designated tester is able to check whether two ciphertexts signcrypt the same plaintext message.
- Plaintext-ciphertext matching: The designated tester is able to match a surrendered plaintext against a given ciphertext without unsigncryption.
- Message/ciphertext authentication: Both the recipient and designated tester can verify whether a given ciphertext is really produced by the claimed sender.
- Message confidentiality: Even though the tester is designated to perform equality test on ciphertexts and plaintext-ciphertext matching, he is unable to infer unsurrendered plaintexts from ciphertexts.

2.1. PKS-DET framework

A PKS-DET scheme consists of the following procedures:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{gp}$: Given a security parameter λ, the system setup procedure produces a global parameter gp.
- $\mathsf{KeyGen}_s(\mathsf{gp}) \to (\mathsf{sk}_s, \mathsf{pk}_s)$: With the global parameter gp, a sender runs the sender key generation procedure to produce a pair of secret key sk_s and public key pk_s.
- $\mathsf{KeyGen}_r(\mathsf{gp}) \to (\mathsf{sk}_r, \mathsf{pk}_r)$: With the global parameter gp, a recipient runs the recipient key generation procedure to produce a pair of secret key sk_r and public key pk_r.
- $\mathsf{KeyGen}_t(\mathsf{gp}) \to (\mathsf{sk}_t, \mathsf{pk}_t)$: With the global parameter gp, a tester runs the tester key generation procedure to produce a pair of secret key sk_t and public key pk_t.
- $\mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m) \to C$: With the global parameter gp, the recipient's public key pk_r, the tester's public key pk_t and the sender's secret key sk_s, the sender runs the signcryption procedure on message $m \in \mathcal{M}$ to produce a ciphertext C.
- $\mathsf{Unsigncrypt}(\mathsf{gp}, \mathsf{pk}_s, \mathsf{pk}_t, \mathsf{sk}_r, C) \to m/\perp$: With the global parameter gp, the sender's public key pk_s, the tester's public key pk_t and the recipient's secret key sk_r, the recipient runs the unsigncryption procedure on ciphertext C to produce a message m or $\perp$ that signifies an error in unsigncryption.
- $\mathsf{PCMatch}(\mathsf{gp}, \mathsf{sk}_t, (\mathsf{pk}_s, \mathsf{pk}_r, C), m') \to 1/0$: With the global parameter gp and the tester's secret key sk_t, the tester runs the plaintext-ciphertext matching procedure on a ciphertext C along with its sender and recipient's public keys pk_s, pk_r, and a message $m' \in \mathcal{M}$. The procedure outputs 1 if the message signcrypted in C is equal to m'; otherwise, the procedure outputs 0.
- $\mathsf{EqTest}(\mathsf{gp}, \mathsf{sk}_t, (\mathsf{pk}_s, \mathsf{pk}_r, C), (\mathsf{pk}'_s, \mathsf{pk}'_r, C')) \to 1/0$: With the global parameter gp and the tester's secret key sk_t, the tester runs the equality test procedure on two ciphertexts C and C' along with their respective sender and recipient's public keys pk_s, pk_r, pk'_s and pk'_r. The procedure outputs 1 if C and C' signcrypt the same message; otherwise, the procedure outputs 0.

A PKS-DET scheme must be *sound* in the sense that: (1) Every ciphertext generated by $\mathsf{Signcrypt}$ is unsigncryptable by $\mathsf{Unsigncrypt}$; (2) For any pair of ciphertext and plaintext, the procedure $\mathsf{PCMatch}$ must output 1 when the ciphertext signcrypts the given plaintext; (3) For any pair of ciphertext and plaintext, the procedure $\mathsf{PCMatch}$ must output 0 with overwhelming probability when the ciphertext does not signcrypt the given plaintext; (4) For any two ciphertexts that signcrypt the same plaintext, which may be *generated by different senders for different recipients* but are designated to the same tester to perform ciphertext equality test, the procedure EqTest must output 1; (5) For any two ciphertexts that signcrypt different plaintexts, the procedure EqTest must output 0 with overwhelming probability.

Definition 2.1 (Soundness)**.** A PKS-DET scheme is *sound* if, for any security parameter $\lambda \in \mathbb{N}$, any global parameter $\mathsf{gp} \leftarrow \mathsf{Setup}(\lambda)$, any secret/public key pairs of two senders $(\mathsf{sk}_s, \mathsf{pk}_s) \leftarrow \mathsf{KeyGen}_s(\mathsf{gp})$, $(\mathsf{sk}'_s, \mathsf{pk}'_s) \leftarrow \mathsf{KeyGen}_s(\mathsf{gp})$, any secret/public key pairs of two recipients $(\mathsf{sk}_r, \mathsf{pk}_r) \leftarrow \mathsf{KeyGen}_r(\mathsf{gp})$, $(\mathsf{sk}'_r, \mathsf{pk}'_r) \leftarrow \mathsf{KeyGen}_r(\mathsf{gp})$, and any secret/public key pair of tester $(\mathsf{sk}_t, \mathsf{pk}_t) \leftarrow \mathsf{KeyGen}_t(\mathsf{gp})$, the following conditions are satisfied:

1. For every $m \in \mathcal{M}$, $\mathsf{Unsigncrypt}(\mathsf{gp}, \mathsf{pk}_s, \mathsf{pk}_t, \mathsf{sk}_r, C) = m$, where $C \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m)$.

2. For any $m, m' \in \mathcal{M}$ such that $C \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m)$, if $m = m'$, then $\mathsf{PCMatch}(\mathsf{gp}, \mathsf{sk}_t, (\mathsf{pk}_s, \mathsf{pk}_r, C), m') = 1$, otherwise $\Pr[\mathsf{PCMatch}(\mathsf{gp}, \mathsf{sk}_t, (\mathsf{pk}_s, \mathsf{pk}_r, C), m') = 1] \leq \epsilon(\lambda)$, where $\epsilon(\cdot)$ is a negligible function.
3. For any $m, m' \in \mathcal{M}$ such that $C \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m)$ and $C' \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}'_r, \mathsf{pk}_t, \mathsf{sk}'_s, m')$, if $m = m'$, then $\mathsf{EqTest}(\mathsf{gp}, \mathsf{sk}_t, (\mathsf{pk}_s, \mathsf{pk}_r, C), (\mathsf{pk}'_s, \mathsf{pk}'_r, C')) = 1$, otherwise $\Pr[\mathsf{EqTest}(\mathsf{gp}, \mathsf{sk}_t, (\mathsf{pk}_s, \mathsf{pk}_r, C), (\mathsf{pk}'_s, \mathsf{pk}'_r, C')) = 1] \leq \epsilon(\lambda)$, where $\epsilon(\cdot)$ is a negligible function.

2.2. Security definitions

In a messaging system, ciphertexts may be subjected to three types of attacks. First, anyone in the system listening to the communication channel may intercept the transmitted ciphertexts and try to deduce the corresponding plaintext messages. Second, the law enforcement (tester) that is monitoring all the ciphertexts may try to infer the plaintext messages of ciphertexts, which is easier than that in the first case since the tester has been designated to perform equality test on ciphertexts. Third, someone may try to forge a ciphertext of a message, in a bid to generate fake news.

Accordingly, we consider three types of adversaries:

- *Type-1 adversary* models a curious outsider who has the global parameter gp and public keys pk_s, pk_r and pk_t of the sender, recipient and tester, and tries to distinguish between ciphertexts.
- *Type-2 adversary* models a curious tester who has been designated by the sender. The tester has the global parameter gp, the public keys pk_s, pk_r of the sender and recipient, and the tester's secret/public key pair $(\mathsf{sk}_t, \mathsf{pk}_t)$, and tries to get the plaintexts corresponding to some ciphertexts.
- *Type-3 adversary* models a malicious tester who has been designated by the sender. The tester has the global parameter gp, the public keys pk_s, pk_r of sender and recipient, and the tester's secret/public key pair $(\mathsf{sk}_t, \mathsf{pk}_t)$, and tries to forge a ciphertext for some message.

The first two types of adversaries capture the requirement of message confidentiality, while the third captures the requirement of message/ciphertext unforgeability. We formally define the security of a PKS-DET scheme in the following three definitions.

Definition 2.2 (IND-CCA2 security against Type-1 adversary)**.** Let Π be a PKS-DET scheme. Suppose $\mathcal{A}$ is a probabilistic polynomial-time (PPT) adversary who interacts with a challenger $\mathcal{C}$ to perform the following security game.

Set-up: Challenger $\mathcal{C}$ runs the Setup procedure to produce global parameter gp and derives $(\mathsf{sk}_s, \mathsf{pk}_s) \leftarrow \mathsf{KeyGen}_s(\mathsf{gp})$, $(\mathsf{sk}_r, \mathsf{pk}_r) \leftarrow \mathsf{KeyGen}_r(\mathsf{gp})$ and $(\mathsf{sk}_t, \mathsf{pk}_t) \leftarrow \mathsf{KeyGen}_t(\mathsf{gp})$. The global parameter gp and public keys pk_s, pk_r and pk_t are given to $\mathcal{A}$.

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

- Signcryption query: For a queried message $m \in \mathcal{M}$, the challenger runs $C \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m)$ and returns C.
- Unsigncryption query: For a queried ciphertext C, the challenger returns m or $\perp$ according to $\mathsf{Unsigncrypt}(\mathsf{gp}, \mathsf{pk}_s, \mathsf{pk}_t, \mathsf{sk}_r, C)$.

Challenge: At the end of Phase 1, the adversary randomly picks two messages $m_0, m_1 \xleftarrow{\$} \mathcal{M}$ with the same length, and sends them to $\mathcal{C}$. The challenger chooses a random value $d \xleftarrow{\$} \{0, 1\}$, computes $C_d \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m_d)$, and gives C_d to the adversary.

Phase 2: The adversary is able to issue queries as in Phase 1, except that C_d cannot be submitted for unsigncryption.

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

Let

$$\mathsf{Adv}^{\text{ind-cca2}}_{\Pi,\mathcal{A}} = \left|\Pr[d' = d] - \frac{1}{2}\right|$$

Π is said to offer indistinguishability under adaptive chosen ciphertext attack (IND-CCA2) against Type-1 adversary if, for all PPT adversary $\mathcal{A}$, there exists a negligible function $\epsilon(\cdot)$ such that $\mathsf{Adv}^{\text{ind-cca2}}_{\Pi,\mathcal{A}} \leq \epsilon(\cdot)$.

Definition 2.3 (OW-CCA2 security against Type-2 adversary)**.** Let Π be a PKS-DET scheme. Suppose $\mathcal{A}$ is a PPT adversary who interacts with a challenger $\mathcal{C}$ to perform the following security game.

Set-up: Challenger $\mathcal{C}$ runs the Setup procedure to produce global parameter gp and derives $(\mathsf{sk}_s, \mathsf{pk}_s) \leftarrow \mathsf{KeyGen}_s(\mathsf{gp})$ and $(\mathsf{sk}_r, \mathsf{pk}_r) \leftarrow \mathsf{KeyGen}_r(\mathsf{gp})$. The global parameter gp and public keys pk_s and pk_r are given to $\mathcal{A}$. The adversary runs the $\mathsf{KeyGen}_t(\mathsf{gp})$ procedure to obtain a key pair $(\mathsf{sk}_t, \mathsf{pk}_t)$, where pk_t is published.

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

- Signcryption query: For a queried message $m \in \mathcal{M}$, the challenger runs $C \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m)$ and returns C.
- Unsigncryption query: For a queried ciphertext C, the challenger returns m or $\perp$ according to $\mathsf{Unsigncrypt}(\mathsf{gp}, \mathsf{pk}_s, \mathsf{pk}_t, \mathsf{sk}_r, C)$.

Challenge: At the end of Phase 1, the challenger randomly picks a message $m^* \xleftarrow{\$} \mathcal{M}$, computes $C^* \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m^*)$, and sends C^* to the adversary.

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

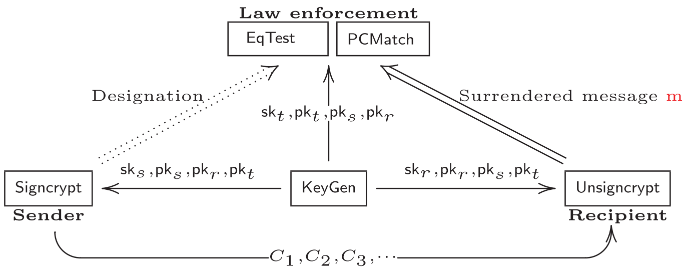
Law enforcement
EqTest
PCMatch
Designation
$\mathsf{sk}_t, \mathsf{pk}_t, \mathsf{pk}_s, \mathsf{pk}_r$
Surrendered message m
Signcrypt
Sender
$\mathsf{sk}_s, \mathsf{pk}_s, \mathsf{pk}_r, \mathsf{pk}_t$
KeyGen
$\mathsf{sk}_r, \mathsf{pk}_r, \mathsf{pk}_s, \mathsf{pk}_t$
Unsigncrypt
Recipient
$C_1, C_2, C_3, \cdots$

Fig. 2. Application of the PKS-DET scheme in securing messaging systems.

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

Let

$$\mathsf{Adv}_{\Pi,\mathcal{A}}^{\text{ow-cca2}} = \Pr[m' = m^*]$$

Π is said to offer one-way confidentiality under adaptive chosen ciphertext attack (OW-CCA2) against Type-2 adversary if, for all PPT adversary $\mathcal{A}$, there exists a negligible function $\epsilon(\cdot)$ such that $\mathsf{Adv}_{\Pi,\mathcal{A}}^{\text{ow-cca2}} \le \epsilon(\cdot)$.

Definition 2.4 (EU-CMA security against Type-3 adversary)**.** Let Π be a PKS-DET scheme. Suppose $\mathcal{A}$ is a PPT adversary who interacts with a challenger $\mathcal{C}$ to perform the following security game.

Set-up: Challenger $\mathcal{C}$ runs the Setup procedure to produce global parameter gp, and derives $(\mathsf{sk}_s, \mathsf{pk}_s) \leftarrow \mathsf{KeyGen}_s(\mathsf{gp})$ and $(\mathsf{sk}_r, \mathsf{pk}_r) \leftarrow \mathsf{KeyGen}_r(\mathsf{gp})$. The global parameter gp and public keys pk_s and pk_r are given to $\mathcal{A}$. The adversary runs $(\mathsf{sk}_t, \mathsf{pk}_t) \leftarrow \mathsf{KeyGen}_t(\mathsf{gp})$, where pk_t is published.

Queries: The adversary is able to adaptively issue the following queries:

- Signcryption query: For a queried message $m_i \in \mathcal{M}$, the challenger returns $C_i \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m_i)$.
- Unsigncryption query: For a queried ciphertext C_i, the challenger returns m_i or $\perp$ according to $\mathsf{Unsigncrypt}(\mathsf{gp}, \mathsf{pk}_s, \mathsf{pk}_t, \mathsf{sk}_r, C_i)$.

Output: Eventually, the adversary outputs a tuple (m^*, C^*).

Adversary $\mathcal{A}$ wins the game if both of the following conditions are satisfied:

1. $m^* \notin \{m_i\}$, that is, m^* has not been submitted in signcryption queries;
2. $\mathsf{Unsigncrypt}(\mathsf{gp}, \mathsf{pk}_s, \mathsf{pk}_t, \mathsf{sk}_r, C^*) = m^*$.

Let

$$\mathsf{Adv}_{\Pi,\mathcal{A}}^{\text{eu-cma}} = \Pr[\mathcal{A} \text{ wins}]$$

Π is said to offer existential unforgeability under adaptive chosen message attack (EU-CMA) against Type-3 adversary if, for all PPT adversary $\mathcal{A}$, there exists a negligible function $\epsilon(\cdot)$ such that $\mathsf{Adv}_{\Pi,\mathcal{A}}^{\text{eu-cma}} \le \epsilon(\cdot)$.

2.3. Application in securing messaging systems

Fig. 2 shows how PKS-DET can be applied in a secure messaging system. There are three types of entities in the system: senders, recipients, and testers (which could be law enforcement agencies). There is also a manager to initiate the system by issuing global parameters, who can be some agency trusted by all system entities. To communicate with a recipient, a sender runs the signcryption procedure $\mathsf{Signcrypt}$ on a message using her secret key, the recipient's public key and a tester's public key. The designated tester collects ciphertexts and tests if they correspond to the same plaintext message by running the EqTest procedure, without additional authorization from the sender or recipient. In the event that a sender or a recipient is required to surrender a plaintext message corresponding to some ciphertext, the designated tester can match the two by invoking the $\mathsf{PCMatch}$ procedure with only the sender and recipient's public keys.

2.4. Mathematical assumptions

Let $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$ and G_T be cyclic groups of prime order p, where $\langle g_x \rangle$ denotes that g_x is a generator of G_x $(x = 1, 2)$. The mapping $\hat{e} : G_1 \times G_2 \to G_T$ is bilinear if the following conditions hold:

- Bilinearity: $\forall u \in G_1$, $v \in G_2$ and $a, b \in Z_p^*$, $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$.
- Non-degeneracy: $\hat{e}(g_1, g_2) \ne 1$.
- Computability: All group operations in G_1, G_2, G_T and bilinear mapping $\hat{e}(\cdot,\cdot)$ can be computed efficiently.

Table 2
Notation.

Symbol	Meaning
G, G_T	Cyclic groups with bilinear mapping $\hat{e}: G \times G \to G_T$
p	Large prime number, the order of G and G_T
g	A generator of G
$H_1(.)$, $H_2(.)$, $H_3(.)$	One-way, collision-resistant hash functions
x_s, X_s	Secret key and public key of sender
$(x_{r,1}, x_{r,2})$, $(X_{r,1}, X_{r,2})$	Secret key and public key of recipient
x_t, X_t	Secret key and public key of tester
m	Message in domain $\mathcal{M}$
$C = (c_1, c_2, c_3, c_4)$	Ciphertext of m
α_1, α_2	Random elements in Z_p^*
$[x]^l$	Substring with length l that is taken from string x
PRG	Pseudo-random bit generator

Moreover, if $G_1 = G_2$, then $\hat{e}$ is a Type 1 (symmetric) bilinear map; if $G_1 \neq G_2$ and there exists an efficiently computable homomorphism from G_2 to G_1, then $\hat{e}$ is a Type 2 (asymmetric) bilinear map; if $G_1 \neq G_2$ and there exists no efficiently computable homomorphism from G_2 to G_1, then $\hat{e}$ is a Type 3 (asymmetric) bilinear map [16].

Our PKS-DET construction relies on the following complexity assumptions.

Computational Diffie-Hellman assumption (CDH). Let $G = \langle g \rangle$ be a cyclic group with bilinear mapping $\hat{e}: G \times G \to G_T$, where G and G_T have prime order p. Given a tuple $(g,\ g^a,\ g^b)$ for random values $a, b \in_R Z_p^*$, any PPT algorithm $\mathcal{E}$ would have negligible probability in computing $g^{ab} \in G$.

Computational bilinear Diffie-Hellman assumption (BDH) [3]. Let $G = \langle g \rangle$ be a cyclic group with bilinear map $\hat{e}: G \times G \to G_T$, where G and G_T have prime order p. Given a tuple $(g,\ g^a,\ g^b,\ g^v)$ for some random values $a, b, v \in_R Z_p^*$, any PPT algorithm $\mathcal{E}$ would have negligible probability in computing $\hat{e}(g, g)^{abv} \in G_T$.

3. Concrete PKS-DET construction

We now present our basic PKS-DET construction in Type 1 bilinear groups. Table 2 summarizes the frequently used notations in our constructions.

Setup: This algorithm picks a symmetric bilinear map: $\hat{e}: G \times G \to G_T$, where $G = \langle g \rangle$ and G_T are cyclic groups with prime order p. Then it picks two cryptographic hash functions H_1: $G_T \to G$ and $H_3: G^4 \to \{0,1\}^{\tau_m + \log p}$, where τ_m denotes the message size in domain $\mathcal{M} = \{0,1\}^{\tau_m}$ and $|\mathcal{M}| = |G|$, and a target collision-resistant hash function H_2 which can be a bijective encoding function from $\mathcal{M}$ to G [8,22]. The global parameters are $\mathsf{gp} = (G, G_T, g, \hat{e}, p, H_1, H_2, H_3)$.

KeyGen_s: The sender randomly picks a secret key $\mathsf{sk}_s = x_s \xleftarrow{\$} Z_p^*$ and computes the corresponding public key $\mathsf{pk}_s = X_s = g^{x_s}$.

KeyGen_r: The recipient randomly picks a secret key $\mathsf{sk}_r = (\mathsf{sk}_{r,1}, \mathsf{sk}_{r,2}) = (x_{r,1}, x_{r,2}) \xleftarrow{\$} (Z_p^*)^2$ and computes the corresponding public key $\mathsf{pk}_r = (X_{r,1} = g^{x_{r,1}}, X_{r,2} = g^{x_{r,2}})$.

KeyGen_t: The tester randomly picks a secret key $\mathsf{sk}_t = x_t \xleftarrow{\$} Z_p^*$ and computes the corresponding public key $\mathsf{pk}_t = X_t = g^{x_t}$.

Signcrypt: For a message $m \in \mathcal{M}$, the sender randomly selects $\alpha_1, \alpha_2 \xleftarrow{\$} Z_p^*$, and generates ciphertext $C = (c_1, c_2, c_3, c_4)$ as follows:

$$c_1 = g^{\alpha_1} \qquad c_2 = g^{\alpha_2}$$
$$c_3 = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2}) \cdot H_2(m)^{\alpha_1 + x_s} \qquad c_4 = H_3(c_1 \| c_2 \| c_3 \| X_{r,2}^{\alpha_2}) \oplus (m \| \alpha_1)$$

Unsigncrypt: Given a ciphertext $C = (c_1, c_2, c_3, c_4)$, the recipient computes

$$m \| \alpha_1 = c_4 \oplus H_3(c_1 \| c_2 \| c_3 \| c_2^{x_{r,2}})$$

then verifies

$$c_1 \stackrel{?}{=} g^{\alpha_1} \tag{1}$$

and

$$\hat{e}\left(\frac{c_3}{H_1(\hat{e}(X_t, c_2)^{x_{r,1}})}, g\right) \stackrel{?}{=} \hat{e}(H_2(m), c_1 \cdot X_s) \tag{2}$$

If both conditions are met, the recipient outputs m.

PCMatch: Given ciphertext C, the tester who is designated to perform equality test may check whether C signcrypts m' as follows:

$$\hat{e}\left(\frac{c_3}{H_1(\hat{e}(X_{r,1}, c_2)^{x_t})}, g\right) \stackrel{?}{=} \hat{e}(H_2(m'), c_1 \cdot X_s) \tag{3}$$

If the condition is met, the tester outputs 1; otherwise he outputs 0.

EqTest: Given ciphertexts C and C', a tester who is designated to perform equality test may check whether they signcrypt the same message (i.e., $m = m'$) as follows:

$$\hat{e}\left(\frac{c_3}{H_1(\hat{e}(X_{r,1}, c_2)^{x_t})}, c_1' \cdot X_s'\right) \overset{?}{=} \hat{e}\left(\frac{c_3'}{H_1(\hat{e}(X_{r,1}', c_2')^{x_t})}, c_1 \cdot X_s\right) \tag{4}$$

If the condition is met, the tester outputs 1; otherwise he outputs 0.

Soundness. For unsigncryption, the equality in (2) holds because

$$\begin{aligned}\hat{e}\left(\frac{c_3}{H_1(\hat{e}(X_t, c_2)^{x_{r,1}})}, g\right) &= \hat{e}\left(\frac{H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2}) \cdot H_2(m)^{\alpha_1 + x_s}}{H_1(\hat{e}(X_t, X_{r,1})^{\alpha_2})}, g\right) \\ &= \hat{e}\big(H_2(m)^{\alpha_1 + x_s}, g\big) \\ &= \hat{e}(H_2(m), c_1 \cdot X_s)\end{aligned}$$

The equality in (3) holds in the same way as the equality in (2) if the message m encrypted by C is equal to m'. On the other hand, if the equality in (3) holds, then $H_2(m) = H_2(m')$ must be true. Since H_2 is bijective, $H_2(m) = H_2(m')$ implies $m = m'$.

For ciphertext equality test, we have

$$\begin{aligned}\hat{e}\left(\frac{c_3}{H_1\big(\hat{e}(X_{r,1}, c_2)^{x_t}\big)}, c_1' \cdot X_s'\right) &= \hat{e}\left(\frac{H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2}) \cdot H_2(m)^{\alpha_1 + x_s}}{H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2})}, g^{\alpha_1'} \cdot g^{x_s'}\right) \\ &= \hat{e}\big(H_2(m)^{\alpha_1 + x_s}, g^{\alpha_1' + x_s'}\big) \\ &= \hat{e}(H_2(m), g)^{(\alpha_1 + x_s)(\alpha_1' + x_s')}\end{aligned}$$

and similarly

$$\begin{aligned}\hat{e}\left(\frac{c_3'}{H_1(\hat{e}(X_{r,1}', c_2')^{x_t})}, c_1 \cdot X_s\right) &= \hat{e}\left(\frac{H_1(\hat{e}(X_{r,1}', X_t)^{\alpha_2'}) \cdot H_2(m')^{\alpha_1' + x_s'}}{H_1(\hat{e}(X_{r,1}', X_t)^{\alpha_2'})}, g^{\alpha_1} \cdot g^{x_s}\right) \\ &= \hat{e}\big(H_2(m')^{\alpha_1' + x_s'}, g^{\alpha_1 + x_s}\big) \\ &= \hat{e}\big(H_2(m'), g\big)^{(\alpha_1' + x_s')(\alpha_1 + x_s)}\end{aligned}$$

Thus, if $m = m'$, then equality in (4) holds. On the other hand, if the equality in (4) holds, then $H_2(m) = H_2(m')$ must hold, which implies $m = m'$.

The proposed PKS-DET construction offers IND-CCA2, OW-CCA2 and EU-CMA security for ciphertexts. To prove the security of our PKS-DET construction, we need the following Difference Lemma [39].

Lemma 3.1. *Let $\mathbb{E}_1$, $\mathbb{E}_2$, and $\mathbb{F}$ be events defined on some probability space. Suppose that 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 3.1. *The above PKS-DET construction is IND-CCA2 secure against Type-1 adversary in the random oracle model assuming that the CDH and BDH assumptions hold.*

The following proof for Theorem 3.1 follows the standard framework established in [32,39,50].

Proof. Let $\mathcal{A}$ be a PPT adversary that has advantage ϵ in attacking the IND-CCA2 security for ciphertexts of the PKS-DET scheme. Suppose $\mathcal{A}$ issues at most q_S signcryption queries, at most q_U unsigncryption queries, at most q_{H_1} hash queries of H_1 and at most q_{H_3} hash queries of H_3 (here, q_S, q_U, q_{H_1} and q_{H_3} are positive). We prove the theorem through a sequence of games.

Game $\mathcal{G}_0$: We define Game $\mathcal{G}_0$ as formulated in Definition 2.2.

1. $x_s, x_{r,1}, x_{r,2}, x_t \overset{\$}{\leftarrow} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$, $X_t = g^{x_t}$.
2. $(m_0, m_1) \in \mathcal{M}^2 \leftarrow \mathcal{A}^{\mathcal{O}_{H_1}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(X_s, X_{r,1}, X_{r,2}, X_t)$ such that $|m_0| = |m_1|$.
3. $d \overset{\$}{\leftarrow} \{0, 1\}$, $\alpha_1^*, \alpha_2^* \overset{\$}{\leftarrow} Z_p^*$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = H_1(\hat{e}(c_2^*, X_t)^{x_{r,1}}) \cdot H_2(m_d)^{x_s + \alpha_1^*}$, and $c_4^* = H_3(c_1^* \| c_2^* \| c_3^* \| (c_2^*)^{x_{r,2}}) \oplus (m_d \| \alpha_1^*)$.
4. $d' \in \{0, 1\} \leftarrow \mathcal{A}^{\mathcal{O}_{H_1}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(m_0, m_1, C^*)$, where the oracles work as follows.
   - H_1 oracle query: For an input element $w \in G_T$, $\mathcal{O}_{H_1}$ responds with a random value $\theta \in G$ in a consistent way, meaning that the same value will be returned for the same input.
   - H_3 oracle query: For an input element $(\varpi_1, \varpi_2, \varpi_3, \varpi_4) \in G^4$, $\mathcal{O}_{H_3}$ responds with a random value $\eta \in \{0, 1\}^{\tau_m + \log p}$ in a consistent way, meaning that the same value will be returned for the same input.
   - $\mathcal{O}_S$ oracle query: For an input message m, $\mathcal{O}_S$ responds with a ciphertext $C = (c_1, c_2, c_3, c_4)$ by running the Signcrypt procedure with x_s.

- $\mathcal{O}_U$ oracle query: For an input ciphertext $C = (c_1, c_2, c_3, c_4)$, $\mathcal{O}_U$ responds with a message m or $\perp$ by running the $\mathsf{Unsigncrypt}$ procedure with $x_{r,1}$ and $x_{r,2}$. Here, unsigncryption queries on C^* are not allowed.

Let $\mathbb{E}_i$ be the event that $d' = d$ in Game $\mathcal{G}_i$. Thus, we have

$$\mathsf{Adv}^{\mathsf{ind\text{-}cca2}}_{\Pi,\mathcal{A}} = \left|\Pr[\mathbb{E}_0] - \frac{1}{2}\right| \tag{5}$$

We then define the following game which is indistinguishable from Game $\mathcal{G}_0$.
<u>Game $\mathcal{G}_1$</u>:

1. $x_s, x_{r,1}, x_{r,2}, x_t \xleftarrow{\$} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$, $X_t = g^{x_t}$, $\mathcal{L}_3 = \varnothing$.
2. $(m_0, m_1) \in \mathcal{M}^2 \leftarrow \mathcal{A}^{\mathcal{O}_{H_1}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(X_s, X_{r,1}, X_{r,2}, X_t)$ such that $|m_0| = |m_1|$.
3. $d \xleftarrow{\$} \{0,1\}$, $\alpha_1^*, \alpha_2^* \xleftarrow{\$} Z_p^*$, $\eta^* \in \{0,1\}^{\tau_m + \log p}$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}) \cdot H_2(m_d)^{x_s + \alpha_1^*}$, and $c_4^* = \eta^* \oplus (m_d \| \alpha_1^*)$. $\mathcal{L}_3 \leftarrow \mathcal{L}_3 \cup \{(c_1^*, c_2^*, c_3^*, X_{r,2}^{\alpha_2^*}, \eta^*)\}$.
4. $d' \in \{0,1\} \leftarrow \mathcal{A}^{\mathcal{O}_{H_1}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(m_0, m_1, C^*)$, where the oracles work as follows.
   - H_1 oracle query: The same as in Game $\mathcal{G}_0$.
   - H_3 oracle query: For answering $\mathcal{O}_{H_3}$ queries, challenger $\mathcal{C}$ maintains a list $\mathcal{L}_3$ which is initially empty. For an input element $(\varpi_1, \varpi_2, \varpi_3, \varpi_4) \in G^4$, if there exists an entry $(\varpi_1, \varpi_2, \varpi_3, \varpi_4, \eta) \in \mathcal{L}_3$, then $\mathcal{O}_{H_3}$ responds with η; otherwise, a random value $\eta \in \{0,1\}^{\tau_m + \log p}$ is picked and returned, and $\mathcal{L}_3$ is updated as $\mathcal{L}_3 \cup (\varpi_1, \varpi_2, \varpi_3, \varpi_4, \eta)$.
   - $\mathcal{O}_S$ oracle query: For an input message m, $\mathcal{O}_S$ randomly picks $\alpha_1, \alpha_2 \xleftarrow{\$} Z_p^*$, queries $\mathcal{O}_{H_1}$ to get $H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2}) = w \in G$, computes $c_1 = g^{\alpha_1}$, $c_2 = g^{\alpha_2}$ and $c_3 = w \cdot H_2(m)^{x_s + \alpha_1}$, queries $\mathcal{O}_{H_3}$ to get $H_3(c_1, c_2, c_3, X_{r,2}^{\alpha_2}) = \eta \in \{0,1\}^{\tau_m + \log p}$, computes $c_4 = \eta \oplus (m \| \alpha_1)$, and responds with the ciphertext $C = (c_1, c_2, c_3, c_4)$.
   - $\mathcal{O}_U$ oracle query: For an input ciphertext $C = (c_1, c_2, c_3, c_4)$, $\mathcal{O}_U$ queries $\mathcal{O}_{H_3}$ to get $H_3(c_1, c_2, c_3, c_2^{x_{r,2}}) = \eta \in \{0,1\}^{\tau_m + \log p}$, computes $m \| \alpha_1 \leftarrow c_4 \oplus \eta$, and checks Equalities (1) and (2), where $\hat{e}(X_t, c_2)^{x_{r,1}}$ is queried to oracle $\mathcal{O}_{H_1}$. If both hold, then m is returned; otherwise, $\perp$ is returned. Also, unsigncryption queries on C^* are not allowed.

Due to the idealness of random oracle, we have

$$\Pr[\mathbb{E}_1] = \Pr[\mathbb{E}_0] \tag{6}$$

We next modify the simulation in an indistinguishable manner.
<u>Game $\mathcal{G}_2$</u>:

1. $x_s, x_{r,1}, x_{r,2}, x_t \xleftarrow{\$} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$, $X_t = g^{x_t}$, $\mathcal{L}_3 = \varnothing$.
2. $(m_0, m_1) \in \mathcal{M}^2 \leftarrow \mathcal{A}^{\mathcal{O}_{H_1}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(X_s, X_{r,1}, X_{r,2}, X_t)$ such that $|m_0| = |m_1|$.
3. $d \xleftarrow{\$} \{0,1\}$, $\alpha_1^*, \alpha_2^* \xleftarrow{\$} Z_p^*$, $\eta^* \in \{0,1\}^{\tau_m + \log p}$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}) \cdot H_2(m_d)^{x_s + \alpha_1^*}$, and $c_4^* = \eta^*$. $\mathcal{L}_3 \leftarrow \mathcal{L}_3 \cup \{(c_1^*, c_2^*, c_3^*, X_{r,2}^{\alpha_2^*}, \eta^* \oplus (m_d \| \alpha_1^*))\}$.
4. $d' \in \{0,1\} \leftarrow \mathcal{A}^{\mathcal{O}_{H_1}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(m_0, m_1, C^*)$, where the oracles work in the same way as in Game $\mathcal{G}_1$ except for the following cases.
   - H_3 oracle query: If $(\cdot, c_2^*, \cdot, (c_2^*)^{x_{r,2}})$ is queried, the game aborts. Let this abortion event be $\mathbb{F}_1$.
   - $\mathcal{O}_U$ oracle query: If $(c_1^*, c_2^*, c_3^*, \bar{c}_4^*)$ is queried such that $\bar{c}_4^* \neq c_4^*$, then $\perp$ is returned.

Since c_4^* is a random value in both games of $\mathcal{G}_1$ and $\mathcal{G}_2$, the challenge ciphertext C^* generated in Game $\mathcal{G}_2$ is identically distributed as in Game $\mathcal{G}_1$. Thus, if event $\mathbb{F}_1$ does not occur, then $\mathcal{G}_2$ is identical to $\mathcal{G}_1$. According to Lemma 3.1, we have

$$|\Pr[\mathbb{E}_2] - \Pr[\mathbb{E}_1]| \leq \Pr[\mathbb{F}_1] \tag{7}$$

Lemma 3.2. $\Pr[\mathbb{F}_1] \leq \mathsf{Adv}^{\mathsf{CDH}} + \frac{q_U}{2^{\tau_m} p}$.

Proof. Suppose that event $\mathbb{F}_1$ happens with non-negligible probability. Using adversary $\mathcal{A}$, we can construct a PPT algorithm $\mathcal{I}$ to break the CDH assumption. At first, algorithm $\mathcal{I}$ is given a CDH instance $(g, g^a, g^b) \in G^3$, with the goal of computing g^{ab}.

Algorithm $\mathcal{I}$ randomly picks $x_s, x_{r,1}, x_t \xleftarrow{\$} Z_p^*$, computes the public keys $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$ and $X_t = g^{x_t}$, and sets $X_{r,2} = g^a$. Algorithm $\mathcal{I}$ invokes adversary $\mathcal{A}$ in Game $\mathcal{G}_2$ on public keys $(X_s, X_{r,1}, X_{r,2}, X_t)$.

Algorithm $\mathcal{I}$ generates a challenge ciphertext $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ for m_d as follows:

$$\alpha_1^* \xleftarrow{\$} Z_p^*, c_1^* = g^{\alpha_1^*}, c_2^* = g^b, c_3^* = H_1(\hat{e}(X_{r,1}, c_2^*)^{x_t}) \cdot H_2(m_d)^{\alpha_1^* + x_s}, \eta^* \in \{0,1\}^{\tau_m + \log p}, c_4 = \eta^*$$

Algorithm $\mathcal{I}$ adds the tuple $(c_1^*, c_2^*, c_3^*, \top, \top)$ to list $\mathcal{L}_3$, where $\top$ denotes an ‘unknown’ value. The challenge ciphertext C^* has the same distribution as that in Game $\mathcal{G}_2$. Algorithm $\mathcal{I}$ simulates the oracles in the same way as in Game $\mathcal{G}_2$ except for the following:

- H_3 oracle query: If $(\cdot, c_2^*, \cdot, \chi)$ is queried, algorithm $\mathcal{I}$ checks $\hat{e}(c_2^*, X_{r,2}) \overset{?}{=} \hat{e}(\chi, g)$. If it holds, algorithm $\mathcal{I}$ outputs χ and aborts the game.

- $\mathcal{O}_U$ oracle query: On input a ciphertext $C=(c_1,c_2,c_3,c_4)$, if $c_j=c_j^*$ holds for $1\le j\le 3$ yet $c_4\ne c_4^*$, algorithm $\mathcal{I}$ returns $\perp$. Otherwise, algorithm $\mathcal{I}$ performs the following steps: Search for an entry $(c_1,\ c_2,\ c_3,\ ^*,\ ^*)$ in $\mathcal{L}_3$. If some entry $(c_1,\ c_2,\ c_3,\ \chi,\ \eta)$ exists, then compute $m\|\alpha_1\leftarrow\eta\oplus c_4$. The unsigncryption m is returned only when both $c_1=g^{\alpha_1}$ and $\hat{e}(\frac{c_3}{H_1(\hat{e}(X_{r,1},c_2)^{x_t})},g)=\hat{e}(H_2(m),c_1\cdot X_s)$ are satisfied, otherwise $\perp$ is returned.

Let $\mathbb{F}_2$ be the event that a tuple $(\cdot,c_2^*,\cdot,g^{ab})$ is queried to oracle $\mathcal{O}_{H_3}$. If event $\mathbb{F}_2$ does not happen by the end of the simulation, then algorithm $\mathcal{I}$ aborts with failure. To show that the unsigncryption queries to $\mathcal{O}_U$ are simulated indistinguishably from Game $\mathcal{G}_2$, we analyze the unsigncryption queries as follows:

- Case 1: (c_1,c_2,c_3,c_2^a) has been queried to $\mathcal{O}_{H_3}$ before an unsigncryption query for $C=(c_1,c_2,c_3,c_4)$ is issued. In this case, c_4 is uniquely determined. Thus, the unsigncryption oracle $\mathcal{O}_U$ is perfectly simulated.
- Case 2: (c_1,c_2,c_3,c_2^a) has not been queried to $\mathcal{O}_{H_3}$ before an unsigncryption query for $C=(c_1,c_2,c_3,c_4)$ is issued. In this case, $\mathcal{O}_U$ will output $\perp$. Thus, the simulations would fail if C is a valid ciphertext. Due to the idealness of $\mathcal{O}_{H_3}$, this happens with probability $1/2^{\tau_m+\log p}$.

Letting $\mathbb{F}_3$ denote the event that a valid ciphertext is rejected in the simulation, we have

$$\Pr[\mathbb{F}_3]\le\frac{q_U}{2^{\tau_m}p}.$$

Thus, if event $\mathbb{F}_3$ does not happen, the simulations are identical to Game $\mathcal{G}_2$. According to Lemma 3.1, we have

$$|\Pr[\mathbb{F}_2]-\Pr[\mathbb{F}_1]|\le\Pr[\mathbb{F}_3].$$

Therefore,

$$\mathsf{Adv}^{\mathsf{CDH}}=\Pr[\mathbb{F}_2]\ge\Pr[\mathbb{F}_1]-\Pr[\mathbb{F}_3]\ge\Pr[\mathbb{F}_1]-\frac{q_U}{2^{\tau_m}p}. \tag{8}$$

This completes the proof of Lemma 3.2. □

We continue to modify the simulation in an indistinguishable manner.

Game $\mathcal{G}_3$:

1. $x_s,x_{r,1},x_{r,2},x_t\overset{\$}{\leftarrow}Z_p^*,\ X_s=g^{x_s},\ X_{r,1}=g^{x_{r,1}},\ X_{r,2}=g^{x_{r,2}},\ X_t=g^{x_t},\ \mathcal{L}_1=\varnothing,\ \mathcal{L}_3=\varnothing.$
2. $(m_0,m_1)\in\mathcal{M}^2\leftarrow\mathcal{A}^{\mathcal{O}_{H_1},\mathcal{O}_{H_3},\mathcal{O}_S,\mathcal{O}_U}(X_s,X_{r,1},X_{r,2},X_t)$ such that $|m_0|=|m_1|$.
3. $d\overset{\$}{\leftarrow}\{0,1\},\ \alpha_1^*,\alpha_2^*\overset{\$}{\leftarrow}Z_p^*,\ \theta^*\overset{\$}{\leftarrow}G,\ \eta^*\in\{0,1\}^{\tau_m+\log p},\ C^*=(c_1^*,c_2^*,c_3^*,c_4^*)$ where $c_1^*=g^{\alpha_1^*},\ c_2^*=g^{\alpha_2^*},\ c_3^*=\theta^*\cdot H_2(m_d)^{x_s+\alpha_1^*}$, and $c_4^*=\eta^*$. $\mathcal{L}_1\leftarrow\mathcal{L}_1\cup\{(\hat{e}(X_{r,1},X_t)^{\alpha_2^*},\theta^*)\},\ \mathcal{L}_3\leftarrow\mathcal{L}_3\cup\{(c_1^*,c_2^*,c_3^*,X_{r,2}^{\alpha_2^*},\eta^*\oplus(m_d\|\alpha_1^*))\}$.
4. $d'\in\{0,1\}\leftarrow\mathcal{A}^{\mathcal{O}_{H_1},\mathcal{O}_{H_3},\mathcal{O}_S,\mathcal{O}_U}(m_0,m_1,C^*)$, where the oracles work as follows.
   - H_1 oracle query: For an input $w\in G_T$, if there exists $(w,\theta)\in\mathcal{L}_1$, then $\mathcal{O}_{H_1}$ returns θ; otherwise, $\mathcal{O}_{H_1}$ randomly picks $\theta\overset{\$}{\leftarrow}G$, inserts $\mathcal{L}_1\leftarrow\mathcal{L}_1\cup\{(w,\theta)\}$, and returns θ.
   - H_3 oracle query: It works in the same way as in Game $\mathcal{G}_2$.
   - $\mathcal{O}_S$ oracle query: Similar to that in Game $\mathcal{G}_2$, where both $\mathcal{O}_{H_1}$ and $\mathcal{O}_{H_3}$ should be queried.
   - $\mathcal{O}_U$ oracle query: Similar to that in Game $\mathcal{G}_2$, where both $\mathcal{O}_{H_1}$ and $\mathcal{O}_{H_3}$ should be queried. If $(c_1^*,c_2^*,c_3^*,\bar{c}_4^*)$ is queried such that $\bar{c}_4^*\ne c_4^*$, then $\perp$ is returned.

Due to the idealness of random oracle, the H_1 oracle $\mathcal{O}_{H_1}$ and signcryption oracle $\mathcal{O}_S$ are identical to those in Game $\mathcal{G}_2$. To show that the unsigncryption queries to $\mathcal{O}_U$ are simulated indistinguishably from Game $\mathcal{G}_2$, we analyze the unsigncryption queries as follows:

- Case 1: $\hat{e}(X_{r,1},c_2)^{x_t}$ has been queried to $\mathcal{O}_{H_1}$ before an unsigncryption query for $(c_1,\ c_2,\ c_3,\ c_4)$ is issued. In this case, $\theta\leftarrow H_1(\hat{e}(X_{r,1},c_2)^{x_t})$ is uniquely determined. Thus, the unsigncryption oracle $\mathcal{O}_U$ is perfectly simulated.
- Case 2: $\hat{e}(X_{r,1},c_2)^{x_t}$ has not been queried to $\mathcal{O}_{H_1}$ before an unsigncryption query for $(c_1,\ c_2,\ c_3,\ c_4)$ is issued. In this case, $\mathcal{O}_U$ will output $\perp$. Thus, the simulations would fail if $(c_1,\ c_2,\ c_3,\ c_4)$ is a valid ciphertext. Due to the idealness of $\mathcal{O}_{H_1}$, this happens with probability $1/p$.

Letting $\mathbb{F}_4$ denote the event that a valid ciphertext is rejected in Case 2, we have

$$\Pr[\mathbb{F}_4]\le\frac{q_U}{p}.$$

If event $\mathbb{F}_4$ does not happen, Game $\mathcal{G}_3$ is identical to Game $\mathcal{G}_2$. According to Lemma 3.1, we have

$$|\Pr[\mathbb{E}_3]-\Pr[\mathbb{E}_2]|\le\Pr[\mathbb{F}_4]\le\frac{q_U}{p}. \tag{9}$$

We continue to modify the simulation in an indistinguishable manner.

Game $\mathcal{G}_4$:

1. $x_s, x_{r,1}, x_{r,2}, x_t \xleftarrow{\$} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$, $X_t = g^{x_t}$, $\mathcal{L}_1 = \varnothing$, $\mathcal{L}_3 = \varnothing$.
2. $(m_0, m_1) \in \mathcal{M}^2 \leftarrow \mathcal{A}^{\mathcal{O}_{H_1}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(X_s, X_{r,1}, X_{r,2}, X_t)$ such that $|m_0| = |m_1|$.
3. $d \xleftarrow{\$} \{0, 1\}$, $\alpha_1^*, \alpha_2^* \xleftarrow{\$} Z_p^*$, $\theta^* \xleftarrow{\$} G$, $\eta^* \in \{0, 1\}^{\tau_m + \log p}$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = \theta^*$, and $c_4^* = \eta^*$. $\mathcal{L}_1 \leftarrow \mathcal{L}_1 \cup \{(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}, \theta^*/H_2(m_d)^{x_s + \alpha_1^*})\}$, $\mathcal{L}_3 \leftarrow \mathcal{L}_3 \cup \{(c_1^*, c_2^*, c_3^*, X_{r,2}^{\alpha_2^*}, \eta^* \oplus (m_d \| \alpha_1^*))\}$.
4. $d' \in \{0, 1\} \leftarrow \mathcal{A}^{\mathcal{O}_{H_1}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(m_0, m_1, C^*)$, where the oracles work as follows.
   - H_1 oracle query: It works in the same way as in Game $\mathcal{G}_3$, except that if $\hat{e}(c_2^*, X_t)^{x_{r,1}}$ is queried, the game aborts. Let this abortion event be $\mathbb{F}_5$.
   - H_3 oracle query: It works in the same way as in Game $\mathcal{G}_3$.
   - $\mathcal{O}_S$ oracle query: It works in the same way as in Game $\mathcal{G}_3$.
   - $\mathcal{O}_U$ oracle query: It works in the same way as in Game $\mathcal{G}_3$.

Since c_3^* is a random value in games $\mathcal{G}_4$ and $\mathcal{G}_3$, the challenge ciphertext C^* generated in these two games are identically distributed. Note that if event $\mathbb{F}_5$ happens, then adversary $\mathcal{A}$ can correctly guess d with probability 1. If event $\mathbb{F}_5$ does not occur, then $\mathcal{G}_4$ is identical to $\mathcal{G}_3$. According to Lemma 3.1, we have

$$|\Pr[\mathbb{E}_4] - \Pr[\mathbb{E}_3]| \leq \Pr[\mathbb{F}_5]. \tag{10}$$

In next lemma, we prove that event $\mathbb{F}_5$ can only happen with negligible probability.

Lemma 3.3. $\Pr[\mathbb{F}_5] \leq \frac{q_{H_1}}{1 - q_U/p} \mathsf{Adv}^{\mathsf{BDH}}$.

Proof. Suppose that event $\mathbb{F}_5$ happens with non-negligible probability. Using adversary $\mathcal{A}$, we can construct a PPT algorithm $\mathcal{I}$ to break the BDH assumption. At first, algorithm $\mathcal{I}$ is given a BDH instance $(g, g^a, g^b, g^v) \in G^4$, with the goal of computing $\hat{e}(g, g)^{abv}$.

Algorithm $\mathcal{I}$ randomly picks $x_s, x_{r,2} \xleftarrow{\$} Z_p^*$, computes $X_s = g^{x_s}$ and $X_{r,2} = g^{x_{r,2}}$, and sets $X_{r,1} = g^a$ and $X_t = g^b$, respectively. Algorithm $\mathcal{I}$ generates a challenge ciphertext $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ as follows:

$$\alpha_1^* \xleftarrow{\$} Z_p^*, c_1^* = g^{\alpha_1^*}, c_2^* = g^v, \theta^* \xleftarrow{\$} G, c_3^* = \theta^*, \eta^* \xleftarrow{\$} \{0, 1\}^{\tau_m + \log p}, c_4 = \eta^*$$

Algorithm $\mathcal{I}$ initiates list $\mathcal{L}_1$ as empty and adds the tuple $(c_1^*, c_2^*, c_3^*, \top, \top)$ to list $\mathcal{L}_3$, where $\top$ denotes an 'unknown' value. The challenge ciphertext C^* has the same distribution as that in Game $\mathcal{G}_4$. Algorithm $\mathcal{I}$ invokes adversary $\mathcal{A}$ in Game $\mathcal{G}_4$ with public keys $(X_s, X_{r,1}, X_{r,2}, X_t)$ and the challenge ciphertext C^*, where the oracles are simulated by $\mathcal{I}$ in the same way as in Game $\mathcal{G}_4$ except for the following:

- $\mathcal{O}_U$ oracle query: On input a ciphertext $C = (c_1, c_2, c_3, c_4)$, if $c_j = c_j^*$ holds for $1 \leq j \leq 3$ yet $c_4 \neq c_4^*$, algorithm $\mathcal{I}$ returns $\perp$. Otherwise, algorithm $\mathcal{I}$ performs the following steps: Query $\mathcal{O}_{H_3}$ to get $H_3(c_1, c_2, c_3, c_2^{x_{r,2}}) = \eta$. Compute $m \| \alpha_1 \leftarrow \eta \oplus c_4$. Continue to search each entry (w, θ) in $\mathcal{L}_1$, if both $c_1 = g^{\alpha_1}$ and $c_3 = \theta \cdot H_2(m)^{\alpha_1 + x_s}$ are satisfied, then m is returned; otherwise $\perp$ is returned.

To show that the unsigncryption queries to $\mathcal{O}_U$ are simulated indistinguishably from Game $\mathcal{G}_4$, we analyze the unsigncryption queries as follows:

- Case 1: $\hat{e}(X_t, c_2)^{x_{r,1}}$ has been queried to $\mathcal{O}_{H_1}$ before an unsigncryption query for $C = (c_1, c_2, c_3, c_4)$ is issued. In this case, θ is uniquely determined. Thus, the unsigncryption oracle $\mathcal{O}_U$ is perfectly simulated.
- Case 2: $\hat{e}(X_t, c_2)^{x_{r,1}}$ has not been queried to $\mathcal{O}_{H_1}$ before an unsigncryption query for $C = (c_1, c_2, c_3, c_4)$ is issued. In this case, $\mathcal{O}_U$ will output $\perp$. Thus, the simulations would fail if C is a valid ciphertext. Due to the idealness of $\mathcal{O}_{H_1}$, this happens with probability $1/p$.

Letting $\mathbb{F}_6$ denote the event that a valid ciphertext is rejected in the simulation, we have

$$\Pr[\mathbb{F}_6] \leq \frac{q_U}{p}.$$

Thus, if event $\mathbb{F}_6$ does not happen, the simulations are identical to Game $\mathcal{G}_4$. Algorithm $\mathcal{I}$ randomly picks a pair (w, θ) from $\mathcal{L}_1$, and sets w as the solution to the given BDH problem instance. Letting $\mathbb{F}_7$ denote the event that $w = \hat{e}(g, g)^{abv}$. We have

$$\Pr[\mathbb{F}_7 | \neg\mathbb{F}_6] = \frac{1}{q_{H_1}} \Pr[\mathbb{F}_5].$$

Therefore,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{BDH}} = \Pr[\mathbb{F}_7] &= \Pr[\mathbb{F}_7|\mathbb{F}_6]\Pr[\mathbb{F}_6] + \Pr[\mathbb{F}_7|\neg\mathbb{F}_6]\Pr[\neg\mathbb{F}_6] \\ &\geq \Pr[\mathbb{F}_7|\neg\mathbb{F}_6]\Pr[\neg\mathbb{F}_6] \\ &= \frac{1}{q_{H_1}}\Pr[\mathbb{F}_5](1-\Pr[\mathbb{F}_6]) \\ &\geq \frac{1-q_U/p}{q_{H_1}}\Pr[\mathbb{F}_5]\end{aligned} \tag{11}$$

This completes the proof of Lemma 3.3. □

In Game $\mathcal{G}_4$, the challenge ciphertext C^* is independent of message m_d, which means that adversary has no advantage in correctly guessing d. Thus,

$$\Pr[\mathbb{E}_4] = 1/2. \tag{12}$$

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

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind\text{-}cca2}}_{\Pi,\mathcal{A}} &\leq \Pr[\mathbb{E}_2] + \Pr[\mathbb{F}_1] - \frac{1}{2} \\ &\leq \Pr[\mathbb{E}_2] + \mathsf{Adv}^{\mathsf{CDH}} + \frac{q_U}{2^{\tau_m}p} - \frac{1}{2} \\ &\leq \Pr[\mathbb{E}_3] + \frac{q_U}{p} + \mathsf{Adv}^{\mathsf{CDH}} + \frac{q_U}{2^{\tau_m}p} - \frac{1}{2} \\ &\leq \Pr[\mathbb{E}_4] + \frac{q_{H_1}}{1-q_U/p}\mathsf{Adv}^{\mathsf{BDH}} + \frac{q_U}{p} + \mathsf{Adv}^{\mathsf{CDH}} + \frac{q_U}{2^{\tau_m}p} - \frac{1}{2} \\ &= \mathsf{Adv}^{\mathsf{CDH}} + \frac{q_U}{2^{\tau_m}p} + \frac{q_U}{p} + \frac{q_{H_1}}{1-q_U/p}\mathsf{Adv}^{\mathsf{BDH}}\end{aligned}$$

This completes the proof of Theorem 3.1. □

Theorem 3.2. *The above PKS-DET construction is OW-CCA2 secure against Type-2 adversary in the random oracle model assuming that the CDH assumption holds.*

The following proof for Theorem 3.2 follows the standard framework established in [32,39,50].

Proof. Let $\mathcal{A}$ be a PPT adversary that has non-negligible advantage in attacking the OW-CCA2 security for ciphertexts of the PKS-DET scheme. Suppose $\mathcal{A}$ issues at most q_S signcryption queries, at most q_U unsigncryption queries, at most q_{H_1} hash queries of H_1 and at most q_{H_3} hash queries of H_3 (here, q_S, q_U, q_{H_1} and q_{H_3} are positive). We prove the theorem through a sequence of games.

Game $\mathcal{G}_0$: We define Game $\mathcal{G}_0$ as formulated in Definition 2.3.

1. $x_s, x_{r,1}, x_{r,2} \xleftarrow{\$} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$.
2. $(x_t \xleftarrow{\$} Z_p^*, X_t = g^{x_t}) \leftarrow \mathcal{A}^{\mathcal{O}_{H_2},\mathcal{O}_{H_3},\mathcal{O}_S,\mathcal{O}_U}(X_s, X_{r,1}, X_{r,2})$.
3. $m^* \xleftarrow{\$} \mathcal{M}$, $\alpha_1^*, \alpha_2^* \xleftarrow{\$} Z_p^*$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}) \cdot H_2(m^*)^{x_s+\alpha_1^*}$, and $c_4^* = H_3(c_1^* \| c_2^* \| c_3^* \| X_{r,2}^{\alpha_2^*}) \oplus (m^* \| \alpha_1^*)$.
4. $m' \leftarrow \mathcal{A}^{\mathcal{O}_{H_2},\mathcal{O}_{H_3},\mathcal{O}_S,\mathcal{O}_U}(C^*)$, where the oracles work in the same way as that in Game $\mathcal{G}_0$ in proving Theorem 3.1, except for $\mathcal{O}_{H_2}$.
   - H_2 oracle query: For an input message $m \in \mathcal{M}$, $\mathcal{O}_{H_2}$ responds with a random value $h \in G$ in a consistent way, meaning that the same value will be returned for the same input.

Let $\mathbb{E}_i$ be the event that $m' = m^*$ in Game $\mathcal{G}_i$ for $0 \leq i \leq 3$. Thus, we have

$$\mathsf{Adv}^{\mathsf{ow\text{-}cca2}}_{\Pi,\mathcal{A}} = \Pr[\mathbb{E}_0] \tag{13}$$

We then define the following game which is indistinguishable from Game $\mathcal{G}_0$.

Game $\mathcal{G}_1$:

1. $x_s, x_{r,1}, x_{r,2} \xleftarrow{\$} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$, $\mathcal{L}_3 = \varnothing$.
2. $(x_t \xleftarrow{\$} Z_p^*, X_t = g^{x_t}) \leftarrow \mathcal{A}^{\mathcal{O}_{H_2},\mathcal{O}_{H_3},\mathcal{O}_S,\mathcal{O}_U}(X_s, X_{r,1}, X_{r,2})$.
3. $m^* \xleftarrow{\$} \mathcal{M}$, $\alpha_1^*, \alpha_2^* \xleftarrow{\$} Z_p^*$, $\eta^* \in \{0,1\}^{\tau_m + \log p}$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}) \cdot H_2(m^*)^{x_s+\alpha_1^*}$, and $c_4^* = \eta^* \oplus (m^* \| \alpha_1^*)$. $\mathcal{L}_3 \leftarrow \mathcal{L}_3 \cup \{(c_1^*, c_2^*, c_3^*, X_{r,2}^{\alpha_2^*}, \eta^*)\}$.
4. $m' \leftarrow \mathcal{A}^{\mathcal{O}_{H_2},\mathcal{O}_{H_3},\mathcal{O}_S,\mathcal{O}_U}(C^*)$, where $\mathcal{O}_{H_2}$ works in the same way as in Game $\mathcal{G}_0$ and the other oracles work in the same way as in Game $\mathcal{G}_1$ in proving Theorem 3.1.

Due to the idealness of random oracle, we have

$$\Pr[\mathbb{E}_1] = \Pr[\mathbb{E}_0] \tag{14}$$

We next modify the simulation in an indistinguishable manner.

Game $\mathcal{G}_2$:

1. $x_s, x_{r,1}, x_{r,2} \overset{\$}{\leftarrow} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$, $\mathcal{L}_3 = \varnothing$.
2. $(x_t \overset{\$}{\leftarrow} Z_p^*, X_t = g^{x_t}) \leftarrow \mathcal{A}^{\mathcal{O}_{H_2}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(X_s, X_{r,1}, X_{r,2})$.
3. $m^* \overset{\$}{\leftarrow} \mathcal{M}$, $\alpha_1^*, \alpha_2^* \overset{\$}{\leftarrow} Z_p^*$, $\eta^* \in \{0,1\}^{\tau_m + \log p}$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}) \cdot H_2(m^*)^{x_s + \alpha_1^*}$, and $c_4^* = \eta^*$. $\mathcal{L}_3 \leftarrow \mathcal{L}_3 \cup \{(c_1^*, c_2^*, c_3^*, X_{r,2}^{\alpha_2^*}, \eta^* \oplus (m^* \| \alpha_1^*))\}$.
4. $m' \leftarrow \mathcal{A}^{\mathcal{O}_{H_2}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(C^*)$, where $\mathcal{O}_{H_2}$ works in the same way as in Game $\mathcal{G}_1$ and the other oracles work in the same way as in Game $\mathcal{G}_2$ in proving Theorem 3.1.

Since c_4^* is a random value in both games of $\mathcal{G}_1$ and $\mathcal{G}_2$, the challenge ciphertext C^* generated in Game $\mathcal{G}_2$ is identically distributed as in Game $\mathcal{G}_1$. Thus, if event $\mathbb{F}_1$ does not occur, then $\mathcal{G}_2$ is identical to $\mathcal{G}_1$. According to Lemma 3.1, we have

$$|\Pr[\mathbb{E}_2] - \Pr[\mathbb{E}_1]| \le \Pr[\mathbb{F}_1] \tag{15}$$

Also, in a way similar to Lemma 3.2, we can prove:

$$\Pr[\mathbb{F}_1] \le \mathsf{Adv}^{\mathsf{CDH}} + \frac{q_U}{2^{\tau_m} p} \tag{16}$$

We modify the simulation in an indistinguishable manner.

Game $\mathcal{G}_3$:

1. $x_s, x_{r,1}, x_{r,2} \overset{\$}{\leftarrow} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$, $\mathcal{L}_2 = \varnothing$, $\mathcal{L}_3 = \varnothing$.
2. $(x_t \overset{\$}{\leftarrow} Z_p^*, X_t = g^{x_t}) \leftarrow \mathcal{A}^{\mathcal{O}_{H_2}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(X_s, X_{r,1}, X_{r,2})$.
3. $m^* \overset{\$}{\leftarrow} \mathcal{M}$, $h^* \overset{\$}{\leftarrow} G$, $\alpha_1^*, \alpha_2^* \overset{\$}{\leftarrow} Z_p^*$, $\eta^* \in \{0,1\}^{\tau_m + \log p}$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}) \cdot (h^*)^{x_s + \alpha_1^*}$, and $c_4^* = \eta^*$. $\mathcal{L}_2 \leftarrow \mathcal{L}_2 \cup \{(m^*, h^*)\}$, $\mathcal{L}_3 \leftarrow \mathcal{L}_3 \cup \{(c_1^*, c_2^*, c_3^*, X_{r,2}^{\alpha_2^*}, \eta^* \oplus (m^* \| \alpha_1^*))\}$.
4. $m' \leftarrow \mathcal{A}^{\mathcal{O}_{H_2}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(C^*)$, where the oracles work in the same way as in Game $\mathcal{G}_2$, except for $\mathcal{O}_{H_2}$.
   - H_2 oracle query: For an input message $m \in \mathcal{M}$, if there exists $(m, h) \in \mathcal{L}_2$, then $\mathcal{O}_{H_2}$ returns h; otherwise, $\mathcal{O}_{H_2}$ randomly picks $h \overset{\$}{\leftarrow} G$, inserts $\mathcal{L}_2 \leftarrow \mathcal{L}_2 \cup \{(m, h)\}$, and returns h.

Due to the idealness of random oracle, we have

$$\Pr[\mathbb{E}_3] = \Pr[\mathbb{E}_2] \tag{17}$$

We then change Game $\mathcal{G}_3$ in letting the adversary output the H_2 hash value of challenge message.

Game $\mathcal{G}_4$:

1. $x_s, x_{r,1}, x_{r,2} \overset{\$}{\leftarrow} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$, $\mathcal{L}_2 = \varnothing$, $\mathcal{L}_3 = \varnothing$.
2. $(x_t \overset{\$}{\leftarrow} Z_p^*, X_t = g^{x_t}) \leftarrow \mathcal{A}^{\mathcal{O}_{H_2}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(X_s, X_{r,1}, X_{r,2})$.
3. $m^* \overset{\$}{\leftarrow} \mathcal{M}$, $h^* \overset{\$}{\leftarrow} G$, $\alpha_1^*, \alpha_2^* \overset{\$}{\leftarrow} Z_p^*$, $\eta^* \in \{0,1\}^{\tau_m + \log p}$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}) \cdot (h^*)^{x_s + \alpha_1^*}$, and $c_4^* = \eta^*$. $\mathcal{L}_2 \leftarrow \mathcal{L}_2 \cup \{(m^*, h^*)\}$, $\mathcal{L}_3 \leftarrow \mathcal{L}_3 \cup \{(c_1^*, c_2^*, c_3^*, X_{r,2}^{\alpha_2^*}, \eta^* \oplus (m^* \| \alpha_1^*))\}$.
4. $h' \leftarrow \mathcal{A}^{\mathcal{O}_{H_2}, \mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(C^*)$, where the oracles work in the same way as in Game $\mathcal{G}_3$.

Let $\mathbb{E}_4$ be the event that $h' = H_2(m^*)$ in Game $\mathcal{G}_4$. Therefore, we have

Lemma 3.4. $\Pr[\mathbb{E}_3] = \Pr[\mathbb{E}_4]$.

We continue to define the following game, where the signcryption and unsigncryption oracles deal with elements in group G.

Game $\mathcal{G}_5$:

1. $x_s, x_{r,1}, x_{r,2} \overset{\$}{\leftarrow} Z_p^*$, $X_s = g^{x_s}$, $X_{r,1} = g^{x_{r,1}}$, $X_{r,2} = g^{x_{r,2}}$, $\mathcal{L}_3 = \varnothing$.
2. $(x_t \overset{\$}{\leftarrow} Z_p^*, X_t = g^{x_t}) \leftarrow \mathcal{A}^{\mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(X_s, X_{r,1}, X_{r,2})$, where $H_3 : G^4 \to \{0,1\}^{\tau_G + \log p}$. Here τ_G denotes the element size of group G.
3. $h^* \overset{\$}{\leftarrow} G$, $\alpha_1^*, \alpha_2^* \overset{\$}{\leftarrow} Z_p^*$, $\eta^* \in \{0,1\}^{\tau_G + \log p}$, $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where $c_1^* = g^{\alpha_1^*}$, $c_2^* = g^{\alpha_2^*}$, $c_3^* = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}) \cdot (h^*)^{x_s + \alpha_1^*}$, $c_4^* = \eta^*$. $\mathcal{L}_3 \leftarrow \mathcal{L}_3 \cup \{(c_1^*, c_2^*, c_3^*, X_{r,2}^{\alpha_2^*}, \eta^* \oplus (h^* \| \alpha_1^*))\}$.
4. $h' \leftarrow \mathcal{A}^{\mathcal{O}_{H_3}, \mathcal{O}_S, \mathcal{O}_U}(C^*)$, where the oracles work in the same way as in Game $\mathcal{G}_4$.

Let $\mathbb{E}_5$ be the event that $h' = h^*$ in Game $\mathcal{G}_5$. The messages in Game $\mathcal{G}_5$ can be seen as $\mathcal{O}_{H_2}$ outputs defined in $\mathcal{G}_4$. The change to the output length of H_3 does not affect its idealness. Thus, we have

Lemma 3.5. $\Pr[\mathbb{E}_4] = \Pr[\mathbb{E}_5]$.

Next, we show that event $\mathbb{E}_5$ can only happen with negligible probability.

Lemma 3.6. $\Pr[\mathbb{E}_5] \leq \mathsf{Adv}^{\mathsf{CDH}}$.

Proof. Suppose that event $\mathbb{E}_5$ happens with non-negligible probability. We construct a PPT algorithm $\mathcal{I}$ to break the CDH assumption. Given a tuple $(g,\, g^a,\, m^a) \in G^3$, where $r \in_R Z_p^*$ and $m \in {}_R G^*$, the goal of algorithm $\mathcal{I}$ is to calculate m.

Algorithm $\mathcal{I}$ randomly picks $x_s, x_{r,1}, x_{r,2} \xleftarrow{\$} Z_p^*$ and computes the public parameters $X_s = g^{x_s},\ X_{r,1} = g^{x_{r,1}},\ X_{r,2} = g^{x_{r,2}}$. Also, algorithm $\mathcal{I}$ randomly selects $\alpha_2^* \xleftarrow{\$} Z_p^*$ and $\eta^* \in \{0,1\}^{\tau_G + \log p}$. It then generates the challenge ciphertext $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ as follows: $c_1^* = g^a/X_s,\ c_2^* = g^{\alpha_2^*},\ c_3^* = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2^*}) \cdot m^a,\ c_4^* = \eta^*$. Algorithm $\mathcal{I}$ adds $\{(c_1^*, c_2^*, c_3^*, X_{r,2}^{\alpha_2^*}, \top)\}$ to list $\mathcal{L}_3$, where $\top$ denotes an 'unknown' value. It invokes adversary $\mathcal{E}$ on input $(X_s,\, X_{r,1},\, X_{r,2})$ and C^*, and simulates the game as described in Game $\mathcal{G}_5$. Eventually, algorithm $\mathcal{I}$ outputs whatever $\mathcal{E}$ outputs. Therefore, Lemma 3.6 follows. □

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

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ow\text{-}cca2}}_{\Pi,\mathcal{A}} &\leq \Pr[\mathbb{E}_2] + \Pr[\mathbb{F}_1] \\
&\leq \Pr[\mathbb{E}_2] + \mathsf{Adv}^{\mathsf{CDH}} + \frac{q_U}{2^{\tau_m} p} \\
&= \Pr[\mathbb{E}_5] + \mathsf{Adv}^{\mathsf{CDH}} + \frac{q_U}{2^{\tau_m} p} \\
&\leq 2\mathsf{Adv}^{\mathsf{CDH}} + \frac{q_U}{2^{\tau_m} p}
\end{aligned}$$

This concludes Theorem 3.2. □

Theorem 3.3. *The above PKS-DET construction is EU-CMA secure against Type-3 adversary in the random oracle model assuming that the CDH assumption holds.*

The proof for Theorem 3.3 follows the standard framework established in [4].

Proof. Suppose that there is a PPT adversary $\mathcal{A}$ who can break the proposed PKS-DET scheme with non-negligible probability ϵ. Suppose $\mathcal{A}$ issues at most q_S signcryption queries, at most q_U unsigncryption queries and at most q_{H_2} hash queries of H_2 (here, q_S, q_U and q_{H_2} are positive). We show how to construct an algorithm $\mathcal{I}$ to solve the CDH problem in a similar way as [4] by manipulating H_2 as a random oracle.

At first, algorithm $\mathcal{I}$ is given a CDH instance $(g,\, g^a,\, h)$. The goal of $\mathcal{I}$ is to compute h^a. Algorithm $\mathcal{I}$ simulates challenger of the PKS-DET scheme and interacts with adversary $\mathcal{A}$ as follows.

Setup: Algorithm $\mathcal{I}$ randomly picks $x_{r,1}, x_{r,2} \xleftarrow{\$} Z_p^*$, computes $X_{r,1} = g^{x_{r,1}}$ and $X_{r,2} = g^{x_{r,2}}$ and sets $X_s = g^a$ which implies $x_s = a$. Adversary $\mathcal{A}$ randomly picks $x_t \xleftarrow{\$} Z_p^*$ and computes $X_t = g^{x_t}$.

Queries: Adversary $\mathcal{A}$ can adaptively submit the following queries.

- $\mathcal{O}_{H_2}$: For such queries with input message m_i, $\mathcal{I}$ maintains a list $\mathcal{L}_2 = \{(m_i,\, \mu_i,\, \nu_i,\, \zeta_i)\}$ and responds as follows. If there is an entry $(m_i,\, \mu_i,\, \nu_i,\, \zeta_i) \in \mathcal{L}_2$, $\mathcal{I}$ returns ζ_i; if not, $\mathcal{I}$ picks a random coin $\mu_i \xleftarrow{\$} \{0,1\}$ such that $\Pr[\mu_i = 0] = \frac{1}{q_S+1}$, picks a random value $\nu_i \xleftarrow{\$} Z_p^*$, computes $\zeta_i = h^{1-\mu_i} g^{\nu_i} \in G$, returns ζ_i and appends $(m_i,\, \mu_i,\, \nu_i,\, \zeta_i)$ to $\mathcal{L}_2$.
- $\mathcal{O}_S$: For each queried message m_i, algorithm $\mathcal{I}$ randomly picks $\alpha_{i,1}, \alpha_{i,2} \xleftarrow{\$} Z_p^*$ and returns a ciphertext $C_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4})$ which is generated as follows. It computes $c_{i,1} = g^{\alpha_{i,1}}$ and $c_{i,2} = g^{\alpha_{i,2}}$, and runs the algorithm in $\mathcal{O}_{H_2}$ for a H_2 query on m_i. Let $(m_i,\, \mu_i,\, \nu_i,\, \zeta_i)$ be the corresponding entry in list $\mathcal{L}_2$. If $\mu_i = 0$, then $\mathcal{I}$ reports failure and aborts the game. Otherwise, algorithm $\mathcal{I}$ computes $c_{i,3} = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_{i,2}}) \cdot (g^{\alpha_{i,1}} \cdot X_s)^{\nu_i}$, where $H_2(m_i) = g^{\nu_i} \in G$. It then computes $c_4 = H_3(c_{i,1} \| c_{i,2} \| c_{i,3} \| X_{r,2}^{\alpha_{i,2}}) \oplus (m_i \| \alpha_{i,1})$. Note that the ciphertexts are perfectly simulated in adversary $\mathcal{A}$'s view when the abortion case does not occur.
- $\mathcal{O}_U$: For each queried ciphertext $C_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4})$, algorithm $\mathcal{I}$ performs the following steps: Compute $m_i \| \alpha_{i,1} \leftarrow c_{i,4} \oplus H_3(c_{i,1} \| c_{i,2} \| c_{i,3} \| c_{i,2}^{x_{r,2}})$, and run the algorithm in $\mathcal{O}_{H_2}$ for a H_2 query on m_i. Let $\zeta_i = H_2(m_i)$. If both $c_{i,1} = g^{\alpha_{i,1}}$ and $\hat{e}(c_{i,3}/H_1(\hat{e}(X_t, c_{i,2})^{x_{r,1}}), g) = \hat{e}(\zeta_i, c_{i,1} \cdot X_s)$ are satisfied, then m_i is returned, otherwise, $\perp$ is returned.

Output: Eventually, adversary $\mathcal{A}$ outputs a tuple $(m^*, C^* = (c_1^*, c_2^*, c_3^*, c_4^*))$ such that m^* has not been queried to $\mathcal{O}_S$. Assume C^* is a valid forged ciphertext of m^*; otherwise, $\mathcal{I}$ reports failure and aborts the game. In the random oracle model, m^* should have been queried to $\mathcal{O}_{H_2}$.

Algorithm $\mathcal{I}$ retrieves the tuple $(m^*,\, \mu^*,\, \nu^*,\, \zeta^*)$ from the list $\mathcal{L}_2$. If $\mu^* = 1$, then $\mathcal{I}$ reports failure and aborts the game. Otherwise, i.e., $\mu^* = 0$, we know $H_2(m^*) = \zeta^* = h \cdot g^{\nu^*} \in G$. Therefore, $c_3^* = H_1(\hat{e}(c_2^*, X_t)^{x_{r,1}}) \cdot (h^a \cdot h^{\alpha_1^*} \cdot X_s^{\nu^*} \cdot (c_1^*)^{\nu^*})$. Next, algorithm $\mathcal{I}$ runs the $\mathsf{Unsigncrypt}$ procedure on C^* to obtain α_1^* and computes $h^a = c_3^*/(H_1(\hat{e}(c_2^*, X_t)^{x_{r,1}}) \cdot h^{\alpha_1^*} \cdot X_s^{\nu^*} \cdot (c_1^*)^{\nu^*})$.

To analyze the probability of solving the given CDH instance, we define three events:

- Let $\mathbb{E}_1$ be the event that algorithm $\mathcal{I}$ does not abort in responding to signcryption queries.
- Let $\mathbb{E}_2$ be the event that C^* is a valid forged ciphertext of m^*.
- Let $\mathbb{E}_3$ be the event that $\mu^* = 1$.

As discussed in [4],

$$\Pr[\mathbb{E}_1] = \left(1 - \frac{1}{q_S + 1}\right)^{q_S} \geq \frac{1}{e}, \quad \Pr[\mathbb{E}_2|\mathbb{E}_1] \geq \varepsilon, \quad \Pr[\mathbb{E}_3|\mathbb{E}_2 \cap \mathbb{E}_1] = \frac{1}{q_S + 1}$$

where e denotes the base of the natural logarithm. Therefore, algorithm $\mathcal{I}$ can correctly solve the given CDH problem with the following probability:

$$\Pr[\mathcal{I}_{success}] = \Pr[\mathbb{E}_1 \cap \mathbb{E}_2 \cap \mathbb{E}_3] = \Pr[\mathbb{E}_1] \cdot \Pr[\mathbb{E}_2|\mathbb{E}_1] \cdot \Pr[\mathbb{E}_3|\mathbb{E}_2 \cap \mathbb{E}_1] \geq \frac{\varepsilon}{e(q_S + 1)}$$

This completes the proof of Theorem 3.3. □

4. Extension, comparison and analysis

In this section, we extend our basic PKS-DET construction in Section 3 to support other message domains and to allow more flexible delegation on ciphertext equality test. We also compare our basic PKS-DET construction and its extensions with existing schemes.

4.1. Extension to other message domains

Our basic PKS-DET construction in Section 3 works for message domains $\mathcal{M}$ such that $|\mathcal{M}| = |G|$. We now extend the construction to work with longer and shorter messages.

Case 1. For signcrypting long messages such that $|\mathcal{M}| > |G|$, our construction can be modified with the technique in [50, Section 4]. In the following, we only describe the parts which are different from the basic PKS-DET construction.

Setup: A pseudo-random bit generator $\mathsf{PRG} : \{0,1\}^{\tau_{seed}} \to \{0,1\}^{\tau_{prg}}$ is chosen, where $\tau_{seed} < \tau_{prg}$. Also, two collision resistant hash functions $H_2 : \mathcal{M} \to G$ and $H_3 : G^4 \to \{0,1\}^{\tau_{seed}}$ are chosen. They are included in the global parameter gp. Let $[x]^l$ denote that a substring with length l is taken from string x.

Signcrypt: Given a message $m \in \mathcal{M}$, the fourth component in ciphertext $C = (c_1, c_2, c_3, c_4)$ is changed to

$$c_4 = \left[\mathsf{PRG}(H_3(c_1 \| c_2 \| c_3 \| X_{r,2}^{\alpha_2}))\right]^{\tau_m + \log p} \oplus (m \| \alpha_1)$$

Unsigncrypt: Given a ciphertext $C = (c_1, c_2, c_3, c_4)$, the recipient computes

$$m \| \alpha_1 = c_4 \oplus \left[\mathsf{PRG}(H_3(c_1 \| c_2 \| c_3 \| c_2^{x_{r,2}}))\right]^{\tau_m + \log p}$$

and verifies Eqs. (1) and (2). If both conditions are met, the recipient outputs m.

Regarding the soundness of the above modified construction, conditions 3 and 5 in Definition 2.1 are bounded by the collision probability of H_2. Similar to Theorems 3.1–3.3, we have the following security results.

Theorem 4.1. *The above modified PKS-DET construction is IND-CCA2 secure against Type-1 adversary in the random oracle model assuming that the CDH and BDH assumptions hold and PRG is a secure pseudo-random bit generator.*

Theorem 4.2. *The above modified PKS-DET construction is OW-CCA2 secure against Type-2 adversary in the random oracle model assuming that the CDH assumption holds and PRG is a secure pseudo-random bit generator.*

Theorem 4.3. *The above modified PKS-DET construction is EU-CMA secure against Type-3 adversary in the random oracle model assuming that the CDH assumption holds and PRG is a secure pseudo-random bit generator.*

Case 2. For the case where $\mathcal{M}$ is a small message domain (i.e., $|\mathcal{M}| < |G|$), for example, $\mathcal{M}$ may contain boolean values, we modify our basic construction by concatenating every message with a random string. That is, in the Signcrypt procedure, the sender picks a random number $z \in \mathcal{R}$ and concatenate it to m as $\bar{m} = m \| z \in \bar{\mathcal{M}} \stackrel{def}{=} \mathcal{M} \times \mathcal{R}$. Accordingly, the other procedures Unsigncrypt, PCMatch and EqTest will work on $\bar{m}$ rather than m. In this case, the EqTest procedure can only check whether two ciphertexts signcrypt the same concatenated message $\bar{m}$ rather than m.

In the basic PKS-DET construction, the designated tester is able to verify whether a surrendered message m matches C by running the PCMatch procedure. However, there is no mechanism to prevent the designated tester from matching some other randomly chosen plaintext against ciphertexts, which is referred to as offline message recovery attack in [32,43] and is unavoidable in PKEET related schemes including our basic PKS-DET construction. Interestingly, our modified construction is not vulnerable to offline message recovery attack, though the functionality of its EqTest procedure is degraded.

In the following, we highlight how the modified construction differs from the basic PKS-DET construction. We let $\mathcal{R} \stackrel{def}{=} \{0,1\}^{\log p - \tau_m}$, which means $\bar{\mathcal{M}} = Z_p$.

Signcrypt: Given a message $m \in \mathcal{M}$, a number $z \xleftarrow{\$} \mathcal{R}$ is randomly chosen, and the last two elements in ciphertext $C = (c_1, c_2, c_3, c_4)$ are changed to

$$c_3 = H_1(\hat{e}(X_{r,1}, X_t)^{\alpha_2}) \cdot H_2(\bar{m})^{\alpha_1 + x_s} \qquad c_4 = H_3(c_1 \| c_2 \| c_3 \| X_{r,2}^{\alpha_2}) \oplus (\bar{m} \| \alpha_1)$$

Unsigncrypt: Given a ciphertext $C = (c_1, c_2, c_3, c_4)$, compute $\bar{m} \| \alpha_1 \leftarrow c_4 \oplus H_3(c_1 \| c_2 \| c_3 \| c_2^{x_{r,2}})$, verify Eq. (1) and

$$\hat{e}\left(\frac{c_3}{H_1(\hat{e}(X_t, c_2)^{x_{r,1}})}, g\right) \stackrel{?}{=} \hat{e}(H_2(\bar{m}), c_1 \cdot X_s) \tag{18}$$

If both conditions are met, output $\bar{m}$.

PCMatch: Check whether C signcrypts $\bar{m}'$ as follows:

$$\hat{e}\left(\frac{c_3}{H_1(\hat{e}(X_{r,1}, c_2)^{x_t})}, g\right) \stackrel{?}{=} \hat{e}(H_2(\bar{m}'), c_1 \cdot X_s) \tag{19}$$

If the condition is met, output 1; otherwise output 0.

EqTest: Check whether ciphertexts C and C' signcrypt the same concatenated message (i.e., $\bar{m} = \bar{m}'$) in the same way as shown in Eq. (4). If the condition is met, output 1; otherwise output 0.

For the soundness requirement formulated in Definition 2.1, all the conditions apply to the concatenated message $\bar{m}$ instead of m. In particular, the third and fifth conditions are changed as follows:

3'. For any $\bar{m} \neq \bar{m}' \in \bar{\mathcal{M}}$ such that $C \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m)$, $\Pr[\mathsf{PCMatch}(\mathsf{gp}, \mathsf{sk}_t, (\mathsf{pk}_s, \mathsf{pk}_r, C), \bar{m}') = 1] \leq \epsilon(\lambda)$, where $\epsilon(\cdot)$ is a negligible function and $z \in_R \mathcal{R}$ is used in Signcrypt to construct $\bar{m}$.
5'. For any $\bar{m} \neq \bar{m}' \in \bar{\mathcal{M}}$ such that $C \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}_r, \mathsf{pk}_t, \mathsf{sk}_s, m)$ and $C' \leftarrow \mathsf{Signcrypt}(\mathsf{gp}, \mathsf{pk}'_r, \mathsf{pk}_t, \mathsf{sk}'_s, m')$, $\Pr[\mathsf{EqTest}(\mathsf{gp}, \mathsf{sk}_t, (\mathsf{pk}_s, \mathsf{pk}_r, C), (\mathsf{pk}'_s, \mathsf{pk}'_r, C')) = 1] \leq \epsilon(\lambda)$ where $\epsilon(\cdot)$ is a negligible function and $z, z' \in_R \mathcal{R}$ are used in Signcrypt to construct $\bar{m}$ and $\bar{m}'$.

Similar to Theorems 3.1–3.3, we have the following security results.

Theorem 4.4. *The above modified PKS-DET construction is IND-CCA2 secure against Type-1 adversary in the random oracle model assuming that the CDH and BDH assumptions hold.*

Theorem 4.5. *The above modified PKS-DET construction is OW-CCA2 secure against Type-2 adversary in the random oracle model assuming that the CDH assumption holds.*

Theorem 4.6. *The above modified PKS-DET construction is EU-CMA secure against Type-3 adversary in the random oracle model assuming that the CDH assumption holds.*

4.2. Extension to flexible delegation

In this section, we show that the basic PKS-DET construction can also be extended to allow a recipient or tester to further delegate to some other party the capability of running the PCMatch and EqTest procedures. This extension is applicable to the scenario where a resource-constrained tester would like to delegate to a well-equipped party to run those two procedures. The delegation mechanism is similar to that in PKEET related schemes [25,26,30–32,42,43,46], where a recipient generates a *token* to enable a third party to perform equality test on his ciphertexts.

Let $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a CCA-secure public key encryption scheme, where the message space is G. Suppose that a party A holds a pair of public/secret keys $(\mathsf{pk}_a, \mathsf{sk}_a) \leftarrow \mathsf{PKE.KGen}(\lambda)$. A recipient is able to generate an encrypted token as

$$\widehat{\mathsf{token}} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_a, X_t^{x_{r,1}})$$

Similarly, a tester can also encrypt the same token as follows

$$\widehat{\mathsf{token}} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_a, X_{r,1}^{x_t})$$

Party A can recover the token by performing the following decryption procedure:

$$\mathsf{token} = g^{x_t x_{r,1}} \leftarrow \mathsf{PKE.Dec}(\mathsf{sk}_a, \widehat{\mathsf{token}})$$

As an example, PKE can be instantiated using Kiltz's key encapsulation mechanism [22], which was proved to be CCA-secure under the Gap Hashed Diffie-Hellman assumption.

With the token, party A is able to perform the following PCMatch and EqTest procedures.

$\mathsf{PCMatch}_a$: Given a ciphertext $C = (c_1, c_2, c_3, c_4)$, party A checks whether C signcrypts m' with the token as follows:

Table 3
Performance comparison with existing encryption schemes supporting equality test on ciphertexts in symmetric bilinear groups.

Scheme	Ciphertext size	Computation cost			
		Signcryption or Encryption	Unsigncryption or Decryption	Equality test	PCMatch
Yang et al. [50]	$3\tau_G + \tau_Z$	$3\pi_G$	$3\pi_G$	$2\pi_{\hat{e}}$	$2\pi_{\hat{e}}$
Tang [43]	$3\tau_{\mathtt{G}} + \tau_Z + \tau_m + \lambda$	$5\pi_{\mathtt{G}}$	$2\pi_{\mathtt{G}}$	$4\pi_{\mathtt{G}}$	$2\pi_{\mathtt{G}}$
Lee et al. [25]	$3\tau_G + \tau_Z$	$4\pi_G$	$3\pi_G$	$2\pi_G + 2\pi_{\hat{e}}$	$\pi_G + 2\pi_{\hat{e}}$
Lee et al. [27]	$(2\lambda + 15)\tau_G + \tau_Z$	$14\pi_G + \pi_{\hat{e}}$	$11\pi_G + 9\pi_{\hat{e}}$	$10\pi_G + 6\pi_{\hat{e}}$	$24\pi_G + 7\pi_{\hat{e}}$
Ma et al. [31]	$5\tau_G + \tau_Z$	$6\pi_G$	$5\pi_G$	$2\pi_G + 2\pi_{\hat{e}}$	$\pi_G + 2\pi_{\hat{e}}$
Ma [30]	$5\tau_G + \tau_Z$	$6\pi_G + 2\pi_{\hat{e}}$	$2\pi_G + 2\pi_{\hat{e}}$	$4\pi_{\hat{e}}$	$3\pi_{\hat{e}}$
Wang and Pang [46]	$5\tau_G + \tau_Z$	$8\pi_G + \pi_{\hat{e}}$	$3\pi_G + 4\pi_{\hat{e}}$	$2\pi_G + 4\pi_{\hat{e}}$	$10\pi_G + 5\pi_{\hat{e}}$
Slamanig et al. [40]	$4\tau_{G_1} + 3\tau_Z$	$6\pi_{G_1}$	$5\pi_{G_1}$	$2\pi_{\hat{de}}$	$2\pi_{\hat{de}}$
Pang and Ding [35]	$7\tau_G + \tau_{G_T}$	$7\pi_G + \pi_{\hat{e}}$	–	$2\pi_G + 5\pi_{\hat{e}}$	–
PKS-DET Section 3	$3\tau_G + 2\tau_Z + \tau_m$	$4\pi_G + \pi_{G_T} + \pi_{\hat{e}}$	$2\pi_G + \pi_{G_T} + 3\pi_{\hat{e}}$	$2\pi_{G_T} + 4\pi_{\hat{e}}$	$\pi_{G_T} + 3\pi_{\hat{e}}$
Section 4.1(1)	$3\tau_G + 2\tau_Z + \tau_m$	$4\pi_G + \pi_{G_T} + \pi_{\hat{e}}$	$2\pi_G + \pi_{G_T} + 3\pi_{\hat{e}}$	$2\pi_{G_T} + 4\pi_{\hat{e}}$	$\pi_{G_T} + 3\pi_{\hat{e}}$
Section 4.1(2)	$3\tau_G + 2\tau_Z + \tau_m$	$4\pi_G + \pi_{G_T} + \pi_{\hat{e}}$	$2\pi_G + \pi_{G_T} + 3\pi_{\hat{e}}$	$2\pi_{G_T} + 4\pi_{\hat{e}}$	$\pi_{G_T} + 3\pi_{\hat{e}}$
Section 4.2	$3\tau_G + 2\tau_Z + \tau_m$	$4\pi_G + \pi_{G_T} + \pi_{\hat{e}}$	$2\pi_G + \pi_{G_T} + 3\pi_{\hat{e}}$	$4\pi_{\hat{e}}$	$3\pi_{\hat{e}}$

$$\hat{e}\left(\frac{c_3}{H_1(\hat{e}(\mathsf{token}, c_2))}, g\right) \stackrel{?}{=} \hat{e}(H_2(m'), c_1 \cdot X_s) \tag{20}$$

EqTest_a: Given ciphertexts $C = (c_1, c_2, c_3, c_4)$ and $C' = (c'_1, c'_2, c'_3, c'_4)$ associated with (sender, recipient, tester) and (sender′, recipient′, tester′), respectively, and given $\mathsf{token} = g^{x_t x_{r,1}}$ and $\mathsf{token}' = g^{x'_t x'_{r,1}}$, party A checks whether C and C' signcrypt the same message (i.e., $m = m'$) as follows:

$$\hat{e}\left(\frac{c_3}{H_1(\hat{e}(\mathsf{token}, c_2))}, c'_1 \cdot X'_s\right) \stackrel{?}{=} \hat{e}\left(\frac{c'_3}{H_1(\hat{e}(\mathsf{token}', c'_2))}, c_1 \cdot X_s\right) \tag{21}$$

The correctness of procedures $\mathsf{PCMatch}_a$ and EqTest_a are straightforward and omitted here.

4.3. Comparison

We now analyze the costs of our PKS-DET construction and its extensions, and compare them in Table 3 with existing schemes supporting equality test on ciphertexts in terms of ciphertext sizes and computation costs of signcryption/encryption, unsigncryption/decryption, matching and equality test. We focus on resource-intensive computations such as exponentiation and bilinear mapping, while lightweight computations including addition, hash evaluation and pseudorandom bit generation are omitted. The cost of the PCMatch procedure is one ciphertext matching for a given surrendered message, and the cost of the EqTest procedure includes the computation in comparing two ciphertexts.

In Table 3, for a symmetric bilinear map $\hat{e} : G \times G \to G_T$, we use τ_G to denote the element size in G, and π_G and $\pi_{\hat{e}}$ to represent the costs of evaluating an exponentiation in G and a symmetric bilinear mapping $\hat{e}(\cdot,\cdot)$, respectively. We use τ_Z and τ_{G_T} to represent the element sizes in Z_p and G_T, respectively, in symmetric bilinear maps. We also use π_{G_T} to denote the cost of an exponentiation on G_T. For Tang's schemes [41,43], $\tau_{\mathtt{G}}$ denotes the element size in an ordinary multiplicative cyclic group $\mathtt{G}$, λ denotes the security parameter, $\pi_{\mathtt{G}}$ represents the evaluation cost of an exponentiation in $\mathtt{G}$. τ_m denotes the message size in domain $\mathcal{M}$ in [41,43] and our PKS-DET constructions; while λ represents the security parameter in [27].

Although the public key schemes in [25,27,30–32,40,41,43,46,47,50] do not include an explicit PCMatch procedure, the (authorized/delegated) tester may match ciphertexts against surrendered messages as follows. Given a surrendered message m, the tester in Yang et al.'s scheme [50] checks $\hat{e}(c_1, m) \stackrel{?}{=} \hat{e}(c_2, g)$ for every ciphertext $C = (c_1, c_2, c_3)$, and concludes that C encrypts m if the condition holds. In Tang's scheme [43], suppose the tester has already received a user U_i's token_i, then the tester verifies $c_4 \stackrel{?}{=} g^{H_2((c_2)^{\mathsf{token}_i})+m}$ for every ciphertext $C = (c_1, c_2, c_3, c_4, c_5)$. In Lee et al.'s scheme [25] that enhanced the security of [19], the authorized tester checks $\hat{e}(c_1, m) \stackrel{?}{=} \hat{e}(c_2/H_1((c_1)^{\alpha}), g)$ for every ciphertext $C = (c_1, c_2, c_3)$, where α denotes the secret key of the recipient that serves as a token in the user's warrant.

For the user level authorization scheme in [31], the tester will obtain a user token token_i from recipient U_i, which is in fact the secret key of U_i. With token_i, the tester computes $\ddot{m}^{r_1} \| (\ddot{m} \cdot Y)^{r_1} \leftarrow c_3 \oplus H_1((c_2)^{\mathsf{token}_i})$ and verifies $\hat{e}(\ddot{m}^{r_1}, g) \stackrel{?}{=} \hat{e}(c_1, m)$ for every ciphertext $C = (c_1, c_2, c_3, c_4)$. If so, then $\ddot{m} = m$, which means C encrypts m. In Ma's scheme [30], upon obtaining a token $\mathsf{token}_{\mathsf{ID}}$ from a recipient ID, the tester checks if $\hat{e}(c_3/H_2(\hat{e}(\mathsf{token}_{\mathsf{ID}}, c_2)), g) \stackrel{?}{=} \hat{e}(c_1, m)$ holds for every ciphertext $C = (c_1, c_2, c_3, c_4, c_5)$. Ma et al.'s scheme [32] allows a delegated tester to verify $\hat{e}(g_1, c_3/H_1(\mathsf{token}, c_2)) \stackrel{?}{=} \hat{e}(c_1, m)$. Note that in [32], the token is encrypted using the ElGamal encryption technique. Thus the tester has to perform 2 and 1 exponentiations in G_1 to recover user token(s) in the EqTest and PCMatch procedures, respectively; however, these recovering steps are performed only once for all ciphertexs in the two procedures.

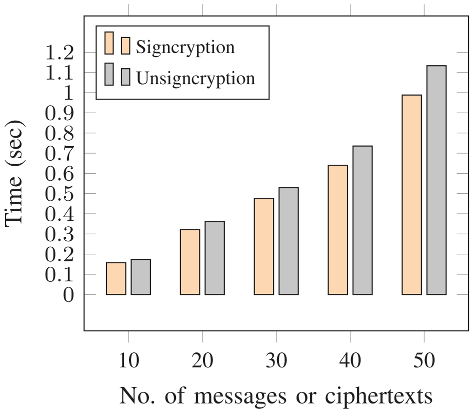

Fig. 3. Performance of Signcrypt and Unsigncrypt.

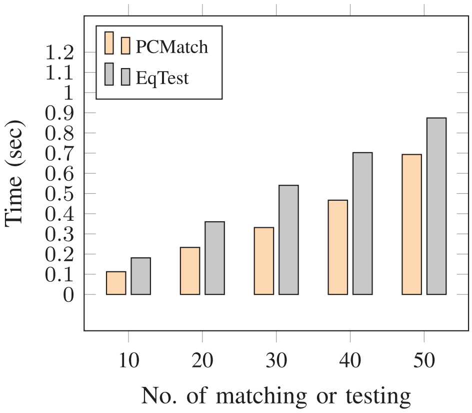

Fig. 4. Performance of PCMatch and EqTest.

The schemes proposed by Slamanig et al. [40] only allow the authorized tester to compare ciphertexts generated by the same user. In their construction based on ElGamal encryption in asymmetric bilinear groups, the token consists of two elements token_1 and token_2, which enables the tester to verify $\hat{e}(c_2/m, \mathsf{token}_1) \overset{?}{=} \hat{e}(c_1, \mathsf{token}_2)$ for every ciphertext $C = (c_1, c_2)$. The scheme presented by Wang et al. [47] has the same implicit PCMatch procedure as [40]. However, in other schemes presented in [27,41,46], to enable a PCMatch procedure on some message m, the tester should sequentially run the Enc and EqTest procedures, that is, he first encrypts m to get C and then matches C against the outsourced ciphertexts. Since Pang and Ding's scheme [35] is designed in the secret key setting, there is no PCMatch procedure on messages for the tester, except when some ciphertext instead of surrendered message is provided.

4.4. Experimental analysis

We conducted the experiments of our PKS-DET construction using the Pairing Based Cryptography Library (PBC, http://crypto.stanford.edu/pbc/). All procedures are executed on a system with Intel(R) Core(TM) i5-5200U CPU at 2.20GHz, 8.00GB RAM and running Windows 7. The elliptic curve is of Type A ($y^2 = x^3 + x$) such that p is a 160-bit prime and $\tau_G = 256$. We obtained the benchmark where each pairing takes roughly 2.4 ms, and an exponentiation in G, G_T and Z_p take roughly 2.7 ms, 0.6 ms and 0.03 ms, respectively. With this benchmark, it is easy to estimate the rough running time of every procedure of the schemes compared in Table 3.

The performance of the signcryption procedure and unsigncryption procedure are shown in Fig. 3, where several cases with different number of inputs to Signcrypt and Unsigncrypt are considered, that is, Signcrypt is run to signcrypt 10, 20, 30, 40 and 50 messages, whereas Unsigncrypt is run to unsigncrypt 10, 20, 30, 40 and 50 ciphertexts. The experiment shows that the average execution time of signcrypting (resp. unsigncrypting) a single message (resp. ciphertext) is roughly 17 msec (resp. 19 msec). Fig. 3 also demonstrates that the performance of Signcrypt and Unsigncrypt are linearly determined by the number of the input elements to be signcrypted or unsigncrypted.

Fig. 4 plots the performance of the plaintext-ciphertext matching procedure and equality test procedure, where several cases with different number of inputs to PCMatch and EqTest are considered, that is, PCMatch is run to match a given message with 10, 20, 30, 40 and 50 ciphertexts, whereas EqTest is run to compare 10, 20, 30, 40 and 50 pairs of ciphertexts. The figure shows that the average execution time of matching a message with a ciphertext is roughly 12 msec, and performing equality test on two ciphertexts needs about 18 msec. Similar to Fig. 3, Fig. 4 also indicates that the cost of PCMatch and EqTest is linear to the number of the input pairs to be matched or tested.

5. Conclusion

Motivated by the problem of simultaneous authentication and monitoring on encrypted data in a secure messaging system, we proposed the notion of public key signcryption with designated equality test on ciphertexts (PKS-DET). We formulated the PKS-DET framework and security model with respect to three types of adversaries, two for data confidentiality and one for data integrity. We then presented a concrete PKS-DET construction in bilinear groups which allows a sender to designate a third-party tester to perform equality test in ciphertexts without any additional explicit authorization. We formally proved the security of our construction in the security model. We also showed how to extend our basic PKS-DET construction to support long and short message domains, and to allow the designated tester to further delegate the equality test functionalities to other parties. Detailed comparison with related schemes showed that our basic PKS-DET construction and its extensions enjoy rich functionalities with reasonable efficiency, and an experimental analysis demonstrated the practicality of our construction.

Acknowledgements

This research is supported by the Singapore National Research Foundation under NCR Award Number NRF2014NCR-NCR001-012. This article is also supported in part by the National Key R&D Program of China through project 2017YFB1400700, the National Natural Science Foundation of China under projects 61772150, 61772538, 61672083, 61532021, 91646203, 61862012, 61862011, and 61602125, the Guangxi Natural Science Foundation under grant 2018GXNSFAA281232, the National Cryptography Development Fund of China under projects MMJJ20170217 and MMJJ20170106, and the Guangxi Young Teachers' Basic Ability Improvement Program under Grant 2018KY0194.

References

[1] P.S.L.M. Barreto, B. Libert, N. McCullagh, J.-J. Quisquater, Efficient and provably-secure identity-based signatures and signcryption from bilinear maps, in: B. Roy (Ed.), Advances in Cryptology - ASIACRYPT 2005: 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4–8, 2005. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 515–532, doi:10.1007/11593447_28.

[2] 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.

[3] D. Boneh, M. Franklin, Identity-based encryption from the weil pairing, SIAM J. Comput. 32 (3) (2003) 586–615, doi:10.1137/S0097539701398521.

[4] D. Boneh, B. Lynn, H. Shacham, Short signatures from the weil pairing, J. Cryptol. 17 (4) (2004) 297–319, doi:10.1007/s00145-004-0314-9.

[5] 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.

[6] X. Boyen, Multipurpose identity-based signcryption, in: D. Boneh (Ed.), Advances in Cryptology - CRYPTO 2003: 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17–21, 2003. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 383–399, doi:10.1007/978-3-540-45146-4_23.

[7] L. Chen, J. Malone-Lee, Improved identity-based signcryption, in: S. Vaudenay (Ed.), Public Key Cryptography - PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23–26, 2005. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 362–379, doi:10.1007/978-3-540-30580-4_25.

[8] S.S.M. Chow, M. Franklin, H. Zhang, Practical dual-receiver encryption, in: J. Benaloh (Ed.), Topics in Cryptology – CT-RSA 2014: The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25–28, 2014. Proceedings, Springer International Publishing, Cham, 2014, pp. 85–105, doi:10.1007/978-3-319-04852-9_5.

[9] H. Cui, R.H. Deng, Y. Li, G. Wu, Attribute-based storage supporting secure deduplication of encrypted data in cloud, IEEE Trans. Big Data PP (99) (2017). 1–1. doi: 10.1109/TBDATA.2017.2656120.

[10] H. Cui, Y. Mu, M.H. Au, Anonymous signcryption against linear related-key attacks, in: W. Susilo, R. Reyhanitabar (Eds.), Provable Security: 7th International Conference, ProvSec 2013, Melaka, Malaysia, October 23–25, 2013. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 165–183, doi:10.1007/978-3-642-41227-1_10.

[11] H. Cui, Y. Mu, M.H. Au, Signcryption secure against linear related-key attacks, Comput. J. 57 (10) (2014) 1472–1483, doi:10.1093/comjnl/bxt076.

[12] P. Datta, R. Dutta, S. Mukhopadhyay, Functional signcryption: notion, construction, and applications, in: M.-H. Au, A. Miyaji (Eds.), Provable Security: 9th International Conference, ProvSec 2015, Kanazawa, Japan, November 24–26, 2015, Proceedings, Springer International Publishing, Cham, 2015, pp. 268–288, doi:10.1007/978-3-319-26059-4_15.

[13] 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.

[14] 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.

[15] M. Gagné, S. Narayan, R. Safavi-Naini, Threshold attribute-based signcryption, in: J.A. Garay, R. De Prisco (Eds.), Security and Cryptography for Networks: 7th International Conference, SCN 2010, Amalfi, Italy, September 13–15, 2010. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 154–171, doi:10.1007/978-3-642-15317-4_11.

[16] S.D. Galbraith, K.G. Paterson, N.P. Smart, Pairings for cryptographers, Discrete Appl. Math. 156 (16) (2008) 3113–3121, doi:10.1016/j.dam.2007.12.010.

[17] J. Herranz, A. Ruiz, G. Sáez, Signcryption schemes with threshold unsigncryption, and applications, Des. Codes Cryptogr. 70 (3) (2014) 323–345, doi:10.1007/s10623-012-9688-0.

[18] H.-F. Huang, C.-C. Chang, An efficient convertible authenticated encryption scheme and its variant, in: S. Qing, D. Gollmann, J. Zhou (Eds.), International Conference on Information and Communications Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 382–392.

[19] 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, doi:10.1093/comjnl/bxv025.

[20] Q. Huang, D.S. Wong, G. Yang, Heterogeneous signcryption with key privacy, Comput. J. 54 (4) (2011) 525–536, doi:10.1093/comjnl/bxq095.

[21] A. Karati, S.H. Islam, G.P. Biswas, M.Z.A. Bhuiyan, P. Vijayakumar, M. Karuppiah, Provably secure identity-based signcryption scheme for crowdsourced industrial internet of things environments, IEEE Internet Things J. PP (99) (2017), doi:10.1109/JIOT.2017.2741580. 1–1

[22] E. Kiltz, Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie-Hellman, in: T. Okamoto, X. Wang (Eds.), Public Key Cryptography – PKC 2007: 10th International Conference on Practice and Theory in Public-Key Cryptography Beijing, China, April 16–20, 2007. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 282–297, doi:10.1007/978-3-540-71677-8_19.

[23] D. Kwak, S. Moon, Efficient distributed signcryption scheme as group signcryption, in: J. Zhou, M. Yung, Y. Han (Eds.), International Conference on Applied Cryptography and Network Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 403–417.

[24] J. Lai, Y. Mu, F. Guo, Efficient identity-based online/offline encryption and signcryption with short ciphertext, Int. J. Inf. Secur. (2016) 1–13, doi:10.1007/s10207-016-0320-6.

[25] 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. (2016), doi:10.1093/comjnl/bxw033.

[26] 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, doi:10.1016/j.ins.2016.09.013.

[27] H.T. Lee, S. Ling, J.H. Seo, H. Wang, T.-Y. Youn, Public key encryption with equality test in the standard model, IACR Cryptology ePrint Archive, 2016. http://eprint.iacr.org/2016/1182.

[28] C.K. Li, G. Yang, D.S. Wong, X. Deng, S.S.M. Chow, An efficient signcryption scheme with key privacy and its extension to ring signcryption, J. Comput. Secur. 18 (3) (2010) 451–473, doi:10.3233/JCS-2009-0374.

[29] F. Li, J. Hong, A.A. Omala, Efficient certificateless access control for industrial internet of things, Future Gener. Comput. Syst. 76 (2017) 285–292, doi:10.1016/j.future.2016.12.036.

[30] S. Ma, Identity-based encryption with outsourced equality test in cloud computing, Inf. Sci. 328 (2016) 389–402, doi:10.1016/j.ins.2015.08.053.

[31] 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, doi:10.1109/TIFS.2014.2378592.

[32] 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, doi:10.1093/comjnl/bxu026.

[33] 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.

[34] 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.

[35] 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.

[36] B. Qin, H. Wang, Q. Wu, J. Liu, J. Domingo-Ferrer, Simultaneous authentication and secrecy in identity-based data upload to cloud, Cluster Comput. 16 (4) (2013) 845–859, doi:10.1007/s10586-013-0258-7.

[37] Y.S. Rao, A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing, Future Gener. Comput. Syst. 67 (2017) 133–151, doi:10.1016/j.future.2016.07.019.

[38] Y.S. Rao, R. Dutta, Efficient attribute-based signature and signcryption realizing expressive access structures, Int. J. Inf. Secur. 15 (1) (2016) 81–109, doi:10.1007/s10207-015-0289-6.

[39] V. Shoup, Sequences of games: a tool for taming complexity in security proofs, (IACR Cryptology ePrint Archive), 2004. http://eprint.iacr.org/2004/332.

[40] 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.

[41] 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, vol. 6812, Springer, Heidelberg, 2011, pp. 389–406, doi:10.1007/978-3-642-22497-3_25.

[42] Q. Tang, Public key encryption schemes supporting equality test with authorisation of different granularity, Int. J. Appl. Cryptogr. 2 (4) (2012) 304–321, doi:10.1504/IJACT.2012.048079.

[43] Q. Tang, Public key encryption supporting plaintext equality test and user-specified authorization, Secur. Commun. Netw. 5 (12) (2012) 1351–1362, doi:10.1002/sec.418.

[44] G. Wang, R.H. Deng, D. Kwak, S. Moon, Security analysis of two signcryption schemes, in: K. Zhang, Y. Zheng (Eds.), International Conference on Information Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 123–133.

[45] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, H. Wang, Privacy-preserving cloud-based road condition monitoring with source authentication in vanets, IEEE Trans. Inf. Forensics Secur. 14 (7) (2018) 1779–1790. doi: 10.1109/TIFS.2018.2885277.

[46] 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.

[47] Y. Wang, H. Pang, N.H. Tran, R.H. Deng, CCA Secure encryption supporting authorized equality test on ciphertexts in standard model and its applications, Inf. Sci. 414 (2017) 289–305, doi:10.1016/j.ins.2017.06.008.

[48] Z. Yan, W. Ding, X. Yu, H. Zhu, R.H. Deng, Deduplication on encrypted big data in cloud, IEEE Trans. Big Data 2 (2) (2016) 138–150, doi:10.1109/TBDATA.2016.2587659.

[49] Z. Yan, M. Wang, Y. Li, A.V. Vasilakos, Encrypted data management with deduplication in cloud computing, IEEE Cloud Comput. 3 (2) (2016) 28–35, doi:10.1109/MCC.2016.29.

[50] 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, vol. 5985, Springer, Heidelberg, 2010, pp. 119–131, doi:10.1007/978-3-642-11925-5_9.

[51] H. Zheng, J. Qin, J. Hu, Q. Wu, Threshold attribute-based signcryption and its application to authenticated key agreement, Secur. Commun. Netw. (2016), doi:10.1002/sec.1664.

[52] Y. Zheng, Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption), in: Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption), Springer Berlin Heidelberg, Berlin, Heidelberg, 1997, pp. 165–179, doi:10.1007/BFb0052234.