INFORMATICA, 2015, Vol. 26, No. 3, 543–556

DOI: http://dx.doi.org/10.15388/Informatica.2015.63

A CCA2-Secure Multi-Decrypter Encryption Scheme Without Random Oracles

Shengbao WANG[1], Peng ZENG[2*], Kim-Kwang Raymond CHOO[3], Hongbing WANG[4]

[1]*School of Information Science and Engineering, Hangzhou Normal University Hangzhou 310012, PR China*
[2]*Shanghai Key Laboratory of Trustworthy Computing, East China Normal University Shanghai 200062, PR China*
[3]*Information Assurance Research Group, Advanced Computing Research Centre University of South Australia, Adelaide SA 5000, Australia*
[4]*Department of Computer Science and Engineering, Shanghai Jiao Tong University Shanghai 200240, PR China*
e-mail: shengbaowang@gmail.com, pzeng@sei.ecnu.edu.cn, raymond.choo@fulbrightmail.org, florawang.2011@gmail.com

Received: August 2013; accepted: March 2014

Abstract. In a multi-decrypter encryption (MDE) scheme, a message encrypted under the public keys of multiple receivers can be recovered only when all the receivers designated by the sender are available (e.g. in a national security setting where a "Top Secret" document can only be decrypted and recovered when all the designated "keyholders" present the respective keys). Despite its effectiveness (i.e. without heavy computational overheads) in ensuring a message can only be read when all the designated parties are available, this is an under-researched topic (there are only two published MDE schemes in the literature, to the best of our knowledge). In this paper, we propose an efficient MDE scheme and prove its CCA2 security in the standard model under the decisional bilinear Diffie–Hellman assumption.

Key words: multi-user cryptography, multi-decrypter encryption, bilinear pairing, chosen ciphertext security, decisional bilinear Diffie–Hellman assumption, standard model.

1. Introduction

In a multi-decrypter encryption (MDE) scheme, the message is encrypted under the public keys of n decrypters designated by the sender and the plaintext can only be recovered by combining all the n decryption shares. As one of the multi-user cryptographic schemes (Bellare *et al.*, 2000, 2003; Kurosawa, 2002; Smart, 2004; Hwang and Lee, 2007; Qin *et al.*, 2008; Selvi *et al.*, 2009), MDE can be very useful in scenarios where a confidential message should only be opened when all the decrypters designated by the encrypter are available. One example application is in public electronic bidding, where a bidder only wants his digital bid document to be read by a group of qualified bid inviters.

*Corresponding author

Chai *et al.* (2007) introduced the first concept of MDE and proposed two concrete MDE schemes. It was claimed that any encryption scheme could be trivially converted to an MDE scheme by splitting the plaintext into n shadows, before applying n times the standard encryption scheme to encrypt each piece of shadows. However, this results in higher computation costs and the size of the ciphertext expands with the number of the decrypters. To achieve the same goal, we can use threshold decryption schemes (Shoup and Gennaro, 2007; Baek and Zheng, 2004) or encrypting the message n rounds (the output of the first round is fed as input to the second round and so on) with each round under the public key of the respective decrypter. However, both methods are impractical (Chai *et al.*, 2007). In the former setting, one has to register a new public-private key pair for the group which is formed by all the n decrypters, prior to distributing the group private key shares among these decrypters by applying an (n, n) secret sharing scheme. This is impractical in some dynamic environments (e.g. mobile/vehicle ad hoc networks: Papadimitratos, 2005; Bresson *et al.*, 2002; Dötzer, 2005), where nodes join and leave the network freely. Meanwhile, in addition to their own public-private pairs, nodes have to store additional group private key shares in their precious key-storage spaces (e.g. smart card) (Chai *et al.*, 2007). The challenge is compounded when a node is involved in $r \gg 1$ different groups (in this case it needs to store r shares). Another deficiency of the former setting is that it enables a dealer, who is responsible to combine the n group private key shares, to recover all ciphertexts in the future as long as the ciphertexts are generated for the same n receivers. The latter setting requires the decryption to be performed sequentially, i.e., the decrypters have to decipher the message one by one, and the last decrypter in the last round recovers the message. One could easily see that the MDE scheme is an elegant solution to the problems mentioned in the preceding sentences: the number of pairing, multiplication and exponentiation computations are significantly reduced; the size of ciphertexts is constant no matter how many decrypters are involved; and all the decrypters perform the decryption in parallel.

Despite the potential of MDE in providing an efficient way of ensuring a message is only read by the designated parties, there are relatively few published MDE schemes. To the best of our knowledge, Chai et al.'s schemes (2007) are the only two MDE schemes published in the literature and both schemes are proven secure in the random oracle model. It is a known fact that a cryptographic scheme proven secure in the random oracle model may be insecure in the real world with any instantiation of the random oracle (Canetti *et al.*, 1998). Not surprisingly in recent years, cryptographic protocol(s) and schemes are generally proven secure in the standard model (e.g. Wang *et al.*, 2012; Ren *et al.*, 2010; Sakalauskas and Mihalkovich, 2014; Tsai *et al.*, 2012, 2014) rather than in the random oracle model (e.g. Choo, 2007; Tang and Choo, 2007; Wang *et al.*, 2009).

Our contribution: In this paper, we construct a new MDE scheme which is semantically secure against adaptive chosen-ciphertext attacks (CCA2-secure) under the decisional bilinear Diffie–Hellman assumption in the standard model. Our scheme is as efficient as Chai et al.'s (2007) in terms of computation cost and ciphertext size, yet achieves a higher security level.

The rest of this paper is organized as follows. In Section 2, we introduce the definitions and preliminaries necessary for understanding the remainder of this paper. We also

present the formal definition of MDE and the security model we work in. In Section 3, we present our proposed MDE scheme and its security, as well as a brief discussion of its performance. Section 4 concludes this paper.

2. Preliminaries

2.1. *Bilinear Map*

Definition 1. Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be two multiplicative groups of the same prime order q, and g be a generator of $\mathbb{G}_1$. Assume that the discrete logarithm problems in both $\mathbb{G}_1$ and $\mathbb{G}_T$ are intractable. We say that e: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ is a bilinear map or pairing if it satisfies the following properties:

1. Bilinear: For all $a, b \in \mathbb{Z}_q$, $e(g^a, g^b) = e(g, g)^{ab}$.
2. Non-degenerate: $e(g, g) \neq 1_{\mathbb{G}_T}$, i.e., $e(g, g)$ generates $\mathbb{G}_T$.
3. Computable: The map e is efficiently computable.

We denote **BSetup** as an algorithm that, on input the security parameter 1^k, outputs the parameters for a bilinear map as $(q, g, \mathbb{G}_1, \mathbb{G}_T, e)$, where $q \in \Theta(2^k)$.

2.2. *Decisional Bilinear Diffie–Hellman (DBDH) Problem*

Definition 2. Let $(q, g, \mathbb{G}_1, \mathbb{G}_T, e)$ be the parameters for a bilinear mapping. The DBDH problem in $\langle \mathbb{G}_1, \mathbb{G}_T, e \rangle$ is to decide, given a tuple of values $(g, g^a, g^b, g^c, T) \in \mathbb{G}_1^4 \times \mathbb{G}_T$ (where $a, b, c \in_R \mathbb{Z}_q^*$), whether $T = e(g, g)^{abc}$ holds.

Let k be a security parameter of sufficient size. Formally, we say that the DBDH assumption holds in $\langle \mathbb{G}_1, \mathbb{G}_T, e \rangle$ if for all probabilistic polynomial time (PPT) algorithms $\mathcal{A}$, the following condition is true:

$$\left| \begin{array}{c} \Pr[a, b, c \leftarrow_R \mathbb{Z}_q^*; 1 \leftarrow \mathcal{A}(g, g^a, g^b, g^c, e(g, g)^{abc})] \\ -\Pr[a, b, c \leftarrow_R \mathbb{Z}_q^*; T \leftarrow_R \mathbb{G}_T; 1 \leftarrow \mathcal{A}(g, g^a, g^b, g^c, T)] \end{array} \right| \leqslant v(k),$$

where $v(\cdot)$ is defined as a negligible function, i.e., for all polynomial functions $p(\cdot)$, $v(k) < 1/p(k)$.

2.3. *Multi-Decrypter Encryption (MDE)*

Definition 3. Let 1^k be a security parameter and *par* be the system's public parameters generated on input k. An MDE scheme is defined to comprise the following four algorithms:

– `KeyGen`$(par) \to (pk, sk)$: A probabilistic algorithm that on input the system's public parameters *par*, generates a public-private key pair (pk, sk).

- $\texttt{Enc}(par, pk_1, \ldots, pk_n, m) \rightarrow C$: A probabilistic algorithm that on input par, n decrypters' public keys $pk_1, \ldots, pk_n$, and a plaintext m from the message space $\mathcal{M}$, outputs a ciphertext C.
- $\texttt{Dec}(par, sk_i, C) \rightarrow (\delta_i, \perp)$: A deterministic algorithm that on input par, a private key sk_i, $1 \leqslant i \leqslant n$, and a ciphertext C from ciphertext space $\mathcal{C}$, outputs a decryption share δ_i or $\perp$ (a symbol indicating an invalid ciphertext).
- $\texttt{Combin}(par, \delta_1, \ldots, \delta_n, C) \rightarrow (m, \perp)$: A deterministic algorithm that on input par, n decryption shares δ_i, $1 \leqslant i \leqslant n$, and a ciphertext $C \in \mathcal{C}$, outputs a plaintext m or $\perp$.

Consistency. We say that an MDE scheme $\mathcal{E} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Dec}, \texttt{Combin})$ is *consistent* if for any valid n public-private key pairs (pk_i, sk_i), $1 \leqslant i \leqslant n$, generated by $\texttt{KeyGen}$, and any plaintext $m \in \mathcal{M}$, the following equation holds:

$$m = \texttt{Combin}\big(par, \delta_1, \ldots, \delta_n, \texttt{Enc}(par, pk_1, \ldots, pk_n, m)\big),$$

where $\delta_i = \texttt{Dec}(par, sk_i, \texttt{Enc}(par, pk_1, \ldots, pk_n, m))$ for any $1 \leqslant i \leqslant n$.

2.4. *Security Model*

Definition 4 **(CCA2-MDE Game).** Let $\mathcal{E} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Dec}, \texttt{Combin})$ be an MDE scheme defined as above. We consider the following game, denoted by $Game_{\mathcal{E},\mathcal{A}}^{CCA2\text{-}MDE}$, in which a PPT adversary $\mathcal{A}$ is involved.

The challenger generates the system's public parameters par and simulates the game with $\mathcal{A}$ as follows:

1. **SELECT.** The challenger chooses $b \leftarrow_R \{0, 1\}$.
2. **PHASE 1.** $\mathcal{A}$ makes the following queries adaptively:
   (a) **Public key generation oracle** $\mathcal{O}_{pk}$: On input an index i,[2] the challenger responds by running algorithm $\texttt{KeyGen}(par)$ to generate a key pair (pk_i, sk_i). Then, the challenger returns pk_i to $\mathcal{A}$, and records (pk_i, sk_i) in the table T_{pk}.
   (b) **Private key generation oracle** $\mathcal{O}_{sk}$: On input a public key pk by $\mathcal{A}$, where pk is from $\mathcal{O}_{pk}$, the challenger searches pk in the table T_{pk}. Finally, the challenger returns sk to $\mathcal{A}$, and records pk in the table T_{sk}.
   (c) **Decryption oracle** $\mathcal{O}_{dec}$: On input (pk_i, C) by $\mathcal{A}$, where pk_i is from $\mathcal{O}_{pk}$, the challenger runs algorithm $\texttt{Dec}$ to decrypt the ciphertext C using the private key sk_i, where sk_i is the private key corresponding to pk_i. The challenger returns the resulting decryption share δ_i to $\mathcal{A}$.
3. **CHOICE AND CHALLENGE.** Once $\mathcal{A}$ decides that the **PHASE 1** is over, he presents his choice $(pk_1^*, \ldots, pk_n^*, m_0, m_1)$, where m_0 and m_1 are two plaintext with identical length from message space, and $pk_1^*, \ldots, pk_n^*$ are n target

[2] This index is only used to distinguish the different public keys.

entities on which it wishes to be challenged.[3] The challenger computes $C^* = \texttt{Enc}(par, pk_1^*, \ldots, pk_n^*, m_b)$, and gives C^* to $\mathcal{A}$ as the challenge ciphertext.

4. **PHASE 2.** $\mathcal{A}$ continues to make queries as in **PHASE 1** with the following restrictions:
   (a) $\mathcal{A}$ is not permitted to launch all private key queries on the n entities $pk_1^*, \ldots, pk_n^*$.
   (b) There is at least one entity pk_i^*, $1 \leqslant i \leqslant n$, that $\mathcal{A}$ is allowed to make neither an $\mathcal{O}_{dec}(pk_i^*, C^*)$ nor an $\mathcal{O}_{sk}(pk_i^*)$.
5. **GUESS.** At the end of **PHASE 2**, $\mathcal{A}$ outputs his guess bit $b' \in \{0, 1\}$. If $b' = b$, then $\mathcal{A}$ wins the game.

Let $(\mathcal{O}'_{pk}, \mathcal{O}'_{sk}, \mathcal{O}'_{dec})$ be the corresponding public key generation oracle, private key generation oracle, and decryption oracle modified in **PHASE 2**. With respect to the system's security parameter k, $\mathcal{A}$'s advantage, denoted by $\text{Adv}_{\mathcal{E},\mathcal{A}}^{CCA2\text{-}MDE}(k)$, is defined as:

$$\left| \Pr\left[b' = b \;\middle|\; \begin{array}{c} b \leftarrow_R \{0,1\}; \\ (pk_1^*, \ldots, pk_n^*, m_0, m_1) \\ \leftarrow \mathcal{A}^{\mathcal{O}_{pk}, \mathcal{O}_{sk}, \mathcal{O}_{dec}}(1^k); \\ C^* = \texttt{Enc}(par, pk_1^*, \ldots, pk_n^*, m_b); \\ b' \leftarrow \mathcal{A}^{\mathcal{O}'_{pk}, \mathcal{O}'_{sk}, \mathcal{O}'_{dec}}(C^*) \end{array} \right] - \frac{1}{2} \right|. \tag{1}$$

We say that $\mathcal{E}$ is CCA2-secure if for all PPT algorithms $\mathcal{A}$, $\text{Adv}_{\mathcal{E},\mathcal{A}}^{CCA2\text{-}MDE}(k)$ is negligible with respect to k.

3. Our Construction

In this section, we construct a new MDE scheme which is CCA2-secure in the standard model assuming that the DBDH assumption holds.

3.1. *Scheme Description*

Let 1^k be the security parameter, $(q, g, \mathbb{G}_1, \mathbb{G}_T, e) \leftarrow \textbf{BSetup}(1^k)$, and $H : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ be a one-way, collision-resistant cryptographic hash function. The system's public parameters are

$$par = (q, g, \mathbb{G}_1, \mathbb{G}_T, e, g_1, g_2, h_1, h_2, h_3, H),$$

where g_1, g_2, h_1, h_2, and h_3 are random elements in $\mathbb{G}_1 \setminus \{g\}$.

The four algorithms `KeyGen`, `Enc`, `Dec`, and `Combin` in our proposed MDE scheme are as follows:

[3] Here we request that there is at least one entity that $\mathcal{A}$ has not made a private key query among these n entities, $pk_1^*, \ldots, pk_n^*$, in **PHASE 1**.

- `KeyGen`$(par) \to (pk, sk)$: On input par, select $x \in_R \mathbb{Z}_q^*$ and set $(pk, sk) = (g^x, x)$.
- `Enc`$(par, pk_1, \ldots, pk_n, m) \to C$: To encrypt a message $m \in \mathbb{G}_T$ with $pk_1, \ldots, pk_n$, do the following:
  1. Select $r, r' \leftarrow_R \mathbb{Z}_q^*$.
  2. Compute

$$\begin{cases} C_1 = g^r, \\ C_2 = \left(\prod_{i=1}^{n} pk_i\right)^r g_1^{r'} g_2^r, \\ C_3 = m \cdot e(g_1, g_1)^{r'} \cdot e(g_1, g_2)^r, \\ C_4 = \left(h_1^{H(C_1)} h_2^{H(C_1 \| C_2 \| C_3)} h_3\right)^r, \end{cases}$$

  and output the ciphertext

$$C = (C_1, C_2, C_3, C_4).$$

- `Dec`$(par, sk_i, C) \to \{\delta_i, \perp\}$: To compute the decryption share δ_i of the ciphertext C using its private key sk_i, the receiver pk_i:
  1. Parses C as (C_1, C_2, C_3, C_4).
  2. Verifies that

$$e(g, C_4) = e\left(C_1, h_1^{H(C_1)} h_2^{H(C_1 \| C_2 \| C_3)} h_3\right).$$

  If not, returns $\perp$. Otherwise, computes

$$\delta_i \leftarrow e\left(C_1, g_1^{sk_i}\right).$$

- `Combin`$(par, \delta_1, \ldots, \delta_n, C)$: To recover the plaintext m with all the n decryption shares $\delta_i \in \mathbb{G}_T$, $i = 1, \ldots, n$, a dealer (maybe one of the receivers) computes:

$$m = C_3 \left(\prod_{i=1}^{n} \delta_i\right) \Big/ e(g_1, C_2). \tag{2}$$

  m is returned.

Consistency.

1. For a decryption share δ_i, we have

$$\delta_i = e\left(C_1, g_1^{sk_i}\right) = e\left(g^r, g_1^{x_i}\right) = e(g, g_1)^{r x_i}. \tag{3}$$

2. In the `Combin` algorithm, we have

$$C_3 \left(\prod_{i=1}^{n} \delta_i\right) \Big/ e(g_1, C_2) = \frac{m \cdot e(g_1, g_1)^{r'} \cdot e(g_1, g_2)^r \prod_{i=1}^{n} e(g, g_1)^{r x_i}}{e(g_1, (\prod_{i=1}^{n} pk_i)^r g_1^{r'} g_2^r)}$$

$$= \frac{m \cdot e(g_1, g_1)^{r'} \cdot e(g_1, g_2)^r \prod_{i=1}^n e(g^{x_i}, g_1)^r}{e(g_1, \prod_{i=1}^n g^{x_i})^r \cdot e(g_1, g_1)^{r'} \cdot e(g_1, g_2)^r}$$
$$= m. \tag{4}$$

Equations (3) and (4) illustrate that the decryption is consistent.

3.2. *Security Proof*

Theorem 1. *If there exists a PPT adversary $\mathcal{A}$ that can break our MDE scheme in the sense of CCA2-MDE Game with non-negligible advantage ϵ, then there exists a PPT algorithm $\mathcal{B}$ that can solve the DBDH problem with the advantage*

$$\epsilon' \geqslant \frac{\epsilon}{q_{sk}} \left(1 - \frac{1}{q_{sk}+1}\right)^{q_{sk}+1},$$

where q_{sk} is the number of private key generation query $\mathcal{O}_{sk}$ requested by $\mathcal{A}$. For large q_{sk}, $\epsilon' \geqslant \epsilon/\exp(1)q_{sk}$.

Proof. Assume that $(q, g, \mathbb{G}_1, \mathbb{G}_T, e)$ are the parameters for a bilinear mapping obtained by running algorithm **BSetup** and the algorithm $\mathcal{B}$ accepts as input a properly-distributed tuple $(g, g^a, g^b, g^c, T) \in \mathbb{G}_1^4 \times \mathbb{G}_T$. With the help of $\mathcal{A}$, as shown below, $\mathcal{B}$ can solve the DBDH problem (i.e., output 1 if $T = e(g, g)^{abc}$ and 0 otherwise) with the advantage more than ϵ.

$\mathcal{B}$ first maintains two initially empty tables T_{pk} and T_{sk}, and generates the MDE scheme's system parameters *par* as follows:

(i) chooses a one-way, collision-resistant cryptographic hash function $H : \{0, 1\}^* \to \mathbb{Z}_q^*$;

(ii) sets

$$g_1 = g^a, \qquad g_2 = g^b;$$

(iii) selects $s_1, s_2, s_3 \in_R \mathbb{Z}_q^*$ and sets

$$h_1 = (g^a)^{s_1}, \qquad h_2 = g^{s_2}, \qquad h_3 = g^{s_3} \cdot (g^a)^{-s_1 H(g^c)}.$$

The system parameters are

$$par = (q, g, \mathbb{G}_1, \mathbb{G}_T, e, g_1, g_2, h_1, h_2, h_3, H).$$

$\mathcal{B}$ sends *par* to $\mathcal{A}$ and interacts with $\mathcal{A}$ as follows.

1. **SELECT.** $\mathcal{B}$ chooses $b' \leftarrow_R \{0, 1\}$.
2. **PHASE 1.** $\mathcal{B}$ responds $\mathcal{A}$'s queries as below:

– $\mathcal{O}_{pk}$: On input an index i, $\mathcal{B}$ selects $x_i \leftarrow_R \mathbb{Z}_q^*$. Using the techniques of Coron (2008), $\mathcal{B}$ flips a biased coin $\alpha_i \in \{0, 1\}$ that $\alpha_i = 1$ with probability γ and 0 otherwise, where γ is a fixed probability to be determined later. If $\alpha_i = 1$, $\mathcal{B}$ sets $pk_i = g^{x_i}$ which means that the private key of the this user is x_i. Otherwise, if $\alpha_i = 0$, $\mathcal{B}$ sets $pk_i = (g^b)^{x_i}$ which means that the private key of this user is bx_i (in this case the private key is unknown to $\mathcal{B}$ too). At last, $\mathcal{B}$ records the tuple (pk_i, x_i, α_i) in table T_{pk}, and responds $\mathcal{A}$ with pk_i. Note that pk_i is correctly distributed.

  We assume that $\mathcal{A}$ has made the appropriate $\mathcal{O}_{pk}$ queries before making one of the following queries.
– $\mathcal{O}_{sk}$: On input pk_i, $\mathcal{B}$ obtains the corresponding value of α_i by accessing table T_{pk}. If $\alpha_i = 0$, $\mathcal{B}$ reports failure and aborts the simulation. Otherwise, $\mathcal{B}$ responds $\mathcal{A}$ with x_i, and records pk_i in table T_{sk}.
– $\mathcal{O}_{dec}$: On input (pk_i, C), $\mathcal{B}$ first parses C as (C_1, C_2, C_3, C_4) and checks whether

$$e(g, C_4) = e\big(C_1, h_1^{H(C_1)} h_2^{H(C_1 \| C_2 \| C_3)} h_3\big).$$

  If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ obtains the corresponding value of pk_i and computes the decryption share δ_i as below:

  (a) If $\alpha_i = 1$, $\mathcal{B}$ computes

$$\delta_i = e(C_1, g_1)^{x_i};$$

  (b) Otherwise $\alpha_i = 0$ (and in this case the private key is $sk_i = bx_i$), $\mathcal{B}$ computes

$$\delta_i = \left[\frac{e(g_2, C_4)}{e(g_2, C_1^{s_2 H(C_1 \| C_2 \| C_3) + s_3})}\right]^{\frac{x_i}{s_1 (H(C_1) - H(g^c))}}. \tag{5}$$

  At last, δ_i is returned to $\mathcal{A}$. Note by Eq. (5), we have

$$\begin{aligned}
\delta_i &= \left[\frac{e(g_2, C_4)}{e(g_2, C_1^{s_2 H(C_1 \| C_2 \| C_3) + s_3})}\right]^{\frac{x_i}{s_1 (H(C_1) - H(g^c))}} \\
&= \left[\frac{e(g, C_4)}{e(g, C_1^{s_2 H(C_1 \| C_2 \| C_3) + s_3})}\right]^{\frac{bx_i}{s_1 (H(C_1) - H(g^c))}} \\
&= \left[\frac{e(C_1, h_1^{H(C_1)} h_2^{H(C_1 \| C_2 \| C_3)} h_3)}{e(g, C_1^{s_2 H(C_1 \| C_2 \| C_3) + s_3})}\right]^{\frac{bx_i}{s_1 (H(C_1) - H(g^c))}} \\
&= e(C_1, g)^{as_1 (H(C_1) - H(g^c)) \frac{bx_i}{s_1 (H(C_1) - H(g^c))}} \\
&= e(C_1, g_1)^{bx_i}.
\end{aligned}$$

  That is, $\mathcal{B}$ can respond $\mathcal{A}$ with the correct decryption query for both the cases $\alpha_i = 1$ and $\alpha_i = 0$.

3. **CHOICE AND CHALLENGE.** At the end of the **PHASE 1**, $\mathcal{A}$ submits $(pk_1^*, \ldots, pk_n^*, m_0, m_1)$ to $\mathcal{B}$ with the restrictions: (i) At least one pk_i^* did not appear in T_{sk}, $i = 1, \ldots, n$; (ii) $m_0, m_1 \in \mathbb{G}_T$ are of the same length. Then $\mathcal{B}$ generates the challenge ciphertext as follows:

(a) $\mathcal{B}$ obtains the corresponding value of pk_i^* by accessing T_{pk}, i.e., $(pk_i^*, \alpha_i^*, x_i^*)$, $i = 1, \ldots, n$.

(b) If $\alpha_i^* = 1$ for any $1 \leqslant i \leqslant n$, then $\mathcal{B}$ aborts the simulation. Otherwise, $\mathcal{B}$ selects $r^* \leftarrow_R \mathbb{Z}_q^*$ and computes

$$\begin{cases} C_1^* = g^c, \\ C_2^* = \left(g^c\right)^{\sum_{\alpha_i^*=1} x_i^*} \left(g^a\right)^{r^*}, \\ C_3^* = m_{b'} \cdot T^{-\sum_{\alpha_i^*=0} x_i^*} \cdot e\left(g^a, g^a\right)^{r^*}, \\ C_4^* = \left(g^c\right)^{s_2 H(C_1^* \| C_2^* \| C_3^*) + s_3}. \end{cases}$$

(c) $\mathcal{B}$ gives $C^* = (C_1^*, C_2^*, C_3^*, C_4^*)$ to $\mathcal{A}$ as the challenge ciphertext.

4. **PHASE 2.** In this phase, the adversary $\mathcal{A}$ is allowed to make queries as in **PHASE 1** with the two restrictions: (i) $\mathcal{A}$ is not permitted to query the private keys of the n entities $pk_1^*, \ldots, pk_n^*$; (ii) There is at least one entity pk_i^* that $\mathcal{A}$ is not allowed to make an $\mathcal{O}_{dec}(pk_i^*, C^*)$ query or an $\mathcal{O}_{sk}(pk_i^*)$ query.

5. **GUESS.** At last, $\mathcal{A}$ outputs a guess $b'' \in \{0, 1\}$. If $b'' = b'$, then $\mathcal{B}$ returns 1 meaning that $T = e(g, g)^{abc}$. Otherwise, $\mathcal{B}$ returns 0 which means that T is a random element from $\mathbb{G}_T$.

It is easy to check that for the above ciphertext $C^* = (C_1^*, C_2^*, C_3^*, C_4^*)$, we have

$$\begin{aligned} e(g, C_4^*) &= e\left(g, \left(g^c\right)^{s_2 H(C_1^* \| C_2^* \| C_3^*) + s_3}\right) \\ &= e\left(g^c, g^{s_2 H(C_1^* \| C_2^* \| C_3^*) + s_3}\right) \\ &= e\left(C_1^*, h_1^{H(C_1^*)} h_2^{H(C_1^* \| C_2^* \| C_3^*)} h_3\right), \end{aligned}$$

and thus

$$\delta_i = \begin{cases} e(g, g)^{acx_i^*}, & \alpha_i^* = 1; \\ e(g, g)^{abcx_i^*}, & \alpha_i^* = 0. \end{cases}$$

If $T = e(g, g)^{abc}$, then

$$\begin{aligned} C_3^* \left(\prod_{i=1}^{n} \delta_i \right) \Big/ e\left(g_1, C_2^*\right) &= \frac{m_{b'} \cdot T^{-\sum_{\alpha_i^*=0} x_i^*} \cdot e(g^a, g^a)^{r^*} \left(\prod_{i=1}^{n} \delta_i\right)}{e\left(g_1, (g^c)^{\sum_{\alpha_i^*=1} x_i^*} (g^a)^{r^*}\right)} \\ &= \frac{m_{b'} \cdot e(g, g)^{-abc \sum_{\alpha_i^*=0} x_i^*} \cdot \left(\prod_{\alpha_i^*=1} \delta_i\right)\left(\prod_{\alpha_i^*=0} \delta_i\right)}{e(g, g)^{ac \sum_{\alpha_i^*=1} x_i^*}} \\ &= m_{b'}. \end{aligned}$$

Table 1
Performance comparison.

	MDE_{Ch1}	MDE_{Ch2}	Our scheme
Paring number	$n+2$	$5n+2$	$3n+3$
Ciphertext size	$2\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$
Security level	CPA	CCA2	CCA2
Using random oracle model?	Yes	Yes	No (in standard model)

That is, $C^* = (C_1^*, C_2^*, C_3^*, C_4^*)$ is the correct encryption of $m_{b'}$ under pk_i^*, $i = 1, \ldots, n$, if $T = e(g, g)^{abc}$. Otherwise, C^* is the encryption of a random element (we have required that there exists at least one element α_i^* with $\alpha_i^* = 0$). It is obvious that all elements given to $\mathcal{A}$ have the correct distribution and the view of $\mathcal{A}$ in the simulation is identical to the view in the real attack if $\mathcal{B}$ does not abort.

The left thing is to compute the probability that $\mathcal{B}$ does not abort. Denote by q_{sk} the number of private key generation query $\mathcal{O}_{sk}$ requested by $\mathcal{A}$. Then the probability that $\mathcal{B}$ does not abort during the query $\mathcal{O}_{sk}$ is $\gamma^{q_{sk}}$. On the other hand, the probability that $\mathcal{B}$ does not abort during the phase of **CHOICE AND CHALLENGE** is at least $1-\gamma$. So with the probability at least $f(\gamma) := \gamma^{q_{sk}}(1-\gamma)$, $\mathcal{B}$ solve the DBDH problem. The function $f(\gamma)$ has a maximal value

$$\frac{1}{q_{sk}}\left(1-\frac{1}{q_{sk}+1}\right)^{q_{sk}+1}$$

when $\gamma = 1 - 1/(q_{sk}+1)$. Consequently, we get

$$\epsilon' \geqslant \frac{\epsilon}{q_{sk}}\left(1-\frac{1}{q_{sk}+1}\right)^{q_{sk}+1}$$

and for large q_{sk}, $\epsilon' \geqslant \epsilon/\exp(1)q_{sk}$. This concludes our proof for Theorem 1. □

3.3. *Performance Analysis*

In Chai *et al.* (2007) compared their two MDE schemes, denoted by MDE_{Ch1} and MDE_{Ch2}, with the split-then-encrypt encryption schemes and showed their MDE schemes are more efficient both in computation cost and ciphertext size. In this section, we compare our proposed MDE scheme with MDE_{Ch1} and MDE_{Ch2} based on computation cost, ciphertext size and the level of security – see Table 1, where $|\mathbb{G}_1|$ and $|\mathbb{G}_T|$ denote the bit length of elements in groups $\mathbb{G}_1$ and $\mathbb{G}_T$ respectively, and n denotes the number of decrypters.

As illustrated in Table 1, when the number of the decrypters increases, the number of pairing computations in MDECh2 is higher than in our scheme. This is mainly due to the `Decrypt` algorithm where four pairing operations for each decrypter are required to verify the validation of the ciphertext in MDE_{Ch2}, whilst our scheme only requires two pairing operations. More importantly, our proposed scheme achieves a higher level of security as shown in Table 1.

4. Conclusion

Multi-decrypter encryption (MDE) is one multi-user cryptographic scheme that provides an efficient way of ensuring a message can only be read by n designated parties. In this paper, we proposed an efficient MDE scheme in which the ciphertext contains only four bilinear group elements regardless of the number of receivers involved. Our scheme is not only efficient both in terms of computation cost and ciphertext size, but also achieves CCA2-secure level under the decisional bilinear Diffie–Hellman assumption in the standard model. To the best of our knowledge, this is the most efficient and secure MDE scheme in the literature.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61321064, 61103222, 61103209, 61202465, and 11061130539; the Shanghai Natural Science Foundation under Grant No. 11ZR1411200; the Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20110076120016; and the Open Foundation of State Key Laboratory of Networking and Switching Technology of China (No. SKLNST-2009-1-13).

References

Baek, J., Zheng, Y. (2004). Identity-based threshold decryption. In: *Proceedings of PKC'04, Lecture Notes in Computer Science*, pp. 262–276.

Bellare, M., Boldyreva, A., Micali, S. (2000). Public-key encryption in a multi-user setting: security proofs and improvements. In: *Proceedings of EUROCRYPT'00, Lecture Notes in Computer Science*, pp. 259–274.

Bellare, M., Boldyreva, A., Staddon, J. (2003). Randomness re-use in multi-recipient encryption schemeas. In: *Proceedings of PKC'03, Lecture Notes in Computer Science*, pp. 85–99.

Bresson, E., Stern, J., Szydlo, M. (2002). Threshold ring signatures and applications to ad-hoc groups. In: *Proceedings of CRYPTO'02, Lecture Notes in Computer Science*, pp. 465–480.

Canetti, R., Goldreich, O., Halevi, S. (1998) The random oracle methodology, revisited (preliminary version). In: *Proceedings of STOC'98*, pp. 209–218.

Chai, Z., Cao, Z., Zhou, Y. (2007). Efficient id-based multi-decrypter encryption with short ciphertexts. *Journal of Computer Science and Technology*, 22(1), 103–108.

Choo, K.-K. R. (2007). A proof of revised Yahalom protocol in the Bellare and Rogaway (1993) mode. *The Computer Journal*, 50(5), 591–601.

Coron, J.-S. (2008). On the exact security of full domain hash. In: *Proceedings of CRYPTO'00, Lecture Notes in Computer Science*, pp. 229–235.

Dötzer, F. (2005). Privacy issues in vehicular ad hoc networks. In: *Proceedings of PET'05, Lecture Notes in Computer Science*, pp. 197–209.

Hwang, Y.H., Lee, P.J. (2007). Public key encryption with conjunctive keyword search and its extension to a multi-user system. In: *Proceedings of Pairing'07, Lecture Notes in Computer Science*, pp. 2–22.

Kurosawa, K. (2002). Multi-recipient public-key encryption with shortened ciphertext. In: *Proceedings of PKC'02, Lecture Notes in Computer Science*, pp. 48–63.

Papadimitratos, P. (2005). Securing ad hoc networks. In: *Proceedings of SPC'05, Lecture Notes in Computer Science*, pp. 46–47.

Qin, L., Cao, Z., Dong, X. (2008). Multi-receiver identity-based encryption in multiple PKG environment. In: *Proceedings of GLOBECOM'08*, pp. 1862–1866.

Ren, Y., Gu, D., Wang, S., Zhang, X. (2010). New fuzzy identity-based encryption in the standard model. *Informatica*, 21(3), 393–407.

Sakalauskas, E., Mihalkovich, A. (2014). New asymmetric cipher of non-commuting cryptography class based on matrix power function. *Informatica*, 25(2), 283–298.

Selvi, S.S.D., Vivek, S.S., Srinivasan, R., Rangan, C.P. (2009). An efficient identity-based signcryption scheme for multiple receivers. In: *Proceedings of IWSEC'09, Lecture Notes in Computer Science*, pp. 71–88.

Shoup, V., Gennaro, R. (2002). Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology*, 15(2), 75–96.

Smart, N.P. (2004). Efficient key encapsulation to multiple parties. In: *Proceedings of SCN'04, Lecture Notes in Computer Science*, pp. 208–219.

Tang, Q., Choo, K.-K.R. (2006). Secure password-based authenticated group key agreement for data-sharing peer-to-peer networks. In: *Proceedings of ACNS'06, Lecture Notes in Computer Science*, pp. 162–177.

Tsai, T.-T., Tseng, Y.-M., Wu, T.-Y. (2012). A fully secure revocable id-based encryption in the standard model. *Informatica*, 23(3), 487–505.

Tsai, T.-T., Tseng, Y.-M., Huang, S.-S. (2014). Efficient strongly unforgeable id-based signature without random oracles. *Informatica*, 25(3), 505–521.

Wang, S., Cao, Z., Cheng, Z., Choo, K.-K.R. (2009). Perfect forward secure identity-based authenticated key agreement protocol in the escrow mode. *Science in China Series F: Information Sciences*, 52(8), 1358–1370.

Wang, X.A., Yang, X., Zhang, M., Yu, Y. (2012). Cryptanalysis of a fuzzy identity based encryption scheme in the standard model. *Informatica*, 23(3), 299–314.

S. Wang received his PhD degree from Shanghai Jiao Tong University in 2008 and is currently an associate professor in School of Information Science and Engineering, Hangzhou Normal University. His research interests focus on applied cryptography and network information security.

P. Zeng received the BS and MS degrees in pure mathematics from Jiangxi Normal University, Jiangxi, and East China Normal University, Shanghai, China, in 2000 and 2003, respectively, and the PhD degree in computer science and technology from Shanghai Jiao Tong University, China, in 2009. He is currently an associate professor with the Software Engineering Institute, East China Normal University, Shanghai, China. His current research interests include applied cryptography, network information security, and coding theory.

K.-K.R. Choo received his PhD degree from Queensland University of Technology in 2006. He is currently a Fulbright Scholar and Senior Lecturer at the University of South Australia. He has (co)authored a number of publications including a book published in Springer's "Advances in Information Security" book series, a book published by Elsevier (Forewords written by Australiaą́s Chief Defence Scientist and Chair of the Electronic Evidence Specialist Advisory Group, Senior Managers of Australian and New Zealand Forensic Laboratories), and six Australian Government Australian Institute of Criminology (AIC) refereed monographs. He has been an invited speaker for a number of events (e.g. 2011 UNODC-ITU Asia–Pacific Regional Workshop on Fighting Cybercrime), and a Keynote/Plenary Speaker at ECPAT Taiwan's 2008 Conference on Criminal Problems and Intervention Strategy, 2010 International Conference on Applied Linguistics, 2011 Economic Crime Asia Conference, 2014 International Conference on Applied Linguistics & Language Teaching, and Anti-Phishing Working Group (APWG)'s 2014 Counter-eCrime Operations Summit (CeCOS VIII); and Invited Lecturer at the Bangladesh Institute of International and Strategic Studies. In 2009, he was named one of 10 Emerging Leaders in the Innovation category of The Weekend Australian Magazine/Microsoft's Next 100 series. He is also the recipient of several awards including the 2010 Australian Capital Territory (ACT) Pearcey Award for "Taking a risk and making a difference in the development of the Australian ICT industry", 2008 Australia Day Achievement Medallion in recognition of his dedication and contribution to the AIC, and through it to the public service of the nation, British Computer Society's Wilkes Award for the best paper published in the 2007 volume of the Computer Journal, and the Best Student Paper Award by the 2005 Australasian Conference on Information Security and Privacy.

H. Wang received the PhD degree in computer science and technology from Shanghai Jiao Tong University, China, in 2013. Her current research interests include applied cryptography, network security, and RFID security.

CCA2 saugos lygio daugelio vartotojų šifravimo schema nenaudojanti „juodosios dėžės" (Random Oracles) modelio

Shengbao WANG, Peng ZENG, Kim-Kwang Raymond CHOO, Hongbing WANG

Daugelio vartotojų šifravimo schemoje (MDE) siuntėjas užšifruoja pranešimą naudodamas n gavėjų viešuosius raktus, o šifrograma gali būti iššifruota tik derinant visų n gavėjų iššifravimo raktus. Straipsnyje pasiūlyta efektyvi skaičiavimų sudėtingumo ir šifrogramos dydžio prasmėmis MDE schema bei įrodyta jos CCA2 lygio sauga tuo atveju, kai galioja bitiesinė Diffi–Hellman'o prielaida.