Scientia Iranica D (2019) 26(3), 1700–1713

SCIENTIA
IRANICA

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
http://scientiairanica.sharif.edu

An attribute-based anonymous broadcast encryption scheme with adaptive security in the standard model

R. Rabaninejad[a], M.H. Ameri[b], M. Delavar[b], and J. Mohajeri[b,*]

a. *Department of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran.*
b. *Electronics Research Institute, Sharif University of Technology, Tehran, Iran.*

Received 10 December 2016; received in revised form 4 September 2017; accepted 2 October 2017

KEYWORDS
Broadcast encryption;
Ciphertext-Policy Attribute-Based Encryption (CP-ABE);
Access structure;
Anonymity;
Provable security;
Attack.

Abstract. In broadcast encryption schemes, a distribution center broadcasts an encrypted message to a subset chosen from a universe of receivers, and only the intended users are able to decrypt the message. Most broadcast encryption schemes do not provide anonymity, and the identities of target receivers are sent in plaintext. However, in several applications, the authorized users' identities have the same sensitivity as the broadcasted messages. Yu, Ren, and Lou (YRL) [Yu, S., Ren, K., and Lou, W. "Attribute-based on-demand multicast group setup with membership anonymity", *Computer Networks*, **54**(3), pp. 377-386 (2010).] considered this issue and introduced an efficient anonymous attribute-based broadcast encryption scheme. This paper first proposed an attack on the YRL scheme, and showed that the unauthorized receivers could also decrypt the broadcasted message. Next, we proposed the Improved-YRL scheme and proved that it achieved anonymity and semantic security under adaptive corruptions in the chosen ciphertext setting. The proof is provided by the dual system encryption technique and is based on three complexity assumptions in composite order bilinear maps. The Improved-YRL scheme is a step forward in solving the long-standing problem of secure and low overhead anonymous broadcast encryption.

1. Introduction

Broadcast Encryption. The concept of Broadcast Encryption (BE) [1] is used when a sender wants to send a message to an arbitrary subset chosen from a universe of receivers via an insecure broadcast channel. In this scenario, the distribution center chooses an arbitrary subgroup of receivers, S, encrypts the message due to set S, and broadcasts the ciphertext through the channel. In a secure broadcast encryption scheme, only the legitimate receivers, which belong to set S, can decrypt the received message, while the unauthorized users obtain no information about the message even if they collude. The broadcast encryption schemes are helpful in several applications including TV subscription and electronic learning services [2] in which only the subscribed users who have made a payment to a certain channel or paid to a virtual course could be able to receive the service. Broadcast encryption schemes can also be used for providing access control in encrypted file systems where a file is encrypted so that only users who have access to the file can decrypt it. Copyrighted content protection and group key distribution are also some other potential applications of the broadcast encryption schemes [3].

Attribute-Based Broadcast Encryption. Since

[*] *Corresponding author.*
E-mail addresses: rabaninejad@ee.kntu.ac.ir. (R. Rabaninejad); ameri_mohammadhasan@ee.sharif.edu (M.H. Ameri); m.delavar@sharif.edu (M. Delavar); mohajer@sharif.edu. (J. Mohajeri)

doi: 10.24200/sci.2017.4517

the introduction of broadcast encryption in 1993 by Fiat and Naor [1], many broadcast encryption schemes have been proposed (see e.g., [4-11]). In these schemes, the broadcaster specifies the legitimate receivers individually, while, in real applications, broadcasters often address groups of receivers with the same characteristics. In these scenarios, especially when the number of receivers is large, identifying each individual receiver is impractical. By using Attribute-Based Broadcast Encryption (ABBE), a broadcaster can encrypt a message under a specified attribute policy, and only the receivers who own the intended attributes can decrypt the message. In other words, in an ABBE scheme, the target set of receivers S is specified by the attributes of its members stated as an access policy. Therefore, the broadcaster has the flexibility to encrypt the message, either with or without the identity information of each individual receiver. Several ABBE schemes have been proposed in the literature among which we can refer to [3,12-15].

Anonymous Broadcast Encryption. In the previous broadcast encryption schemes, the authorized receiver needs information about the intended set of receivers S in order to decrypt the ciphertext correctly. Therefore, set S must be transmitted as part of the ciphertext. Hence, all users including the authorized and unauthorized ones will be aware of the authorized set of receivers. This causes important privacy issues; for example, in group key distribution, everyone will know which users and how many of them are involved in a task. In addition, in applications like television broadcasting, the user who has paid a subscription to a certain channel will know who else has paid for that subscription and the user's privacy is violated. To solve this issue, Barth et al. [16] proposed the first anonymous broadcast encryption scheme. Their scheme protects receivers' identities; however, the number of receivers is leaked by the ciphertext length. In addition, the computation and communication overheads are linear in the number of users. Libert et al. [17] suggested another anonymous broadcast encryption scheme in the standard model with overhead linear in the number of receivers. Schemes [16] and [17] provide full anonymity, meaning that any user, whether he is in set S or not, is unable to obtain information about intended receivers. Outsider anonymity [18] is another definition that only guarantees the anonymity of intended receivers from the view of users outside of set S. However, users in S can still learn the identities of other legal receivers. Fazio and Perea [18] proposed an outsider-anonymous broadcast encryption scheme with sublinear overheads. The attribute-based anonymous multicast scheme presented by Yu et al. [3], which we call it the YRL scheme in this paper, suggests a stronger definition of full anonymity; the scheme not only hides the identities of receivers but also protects the number of intended users. In addition, communication and computation overheads are linear in the number of attributes and are independent of the number of receivers; therefore, the scheme provides high efficiency because of its attribute-based structure. The scheme relies on the notion of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and aims to solve the group key distribution problem. Therefore, instead of broadcasting a message M, a group key GK is emitted.

Our contributions. In this paper, we have made the following three main contributions:

- We propose an attack on the YRL scheme. This attack shows that all users, including the authorized and unauthorized ones, can decrypt the broadcasted message. Therefore, the YRL scheme is not secure and does not provide the main requirement of broadcast encryption schemes that only the authorized users should be able to decrypt the broadcasted message [19];
- We develop an enhanced scheme in composite order bilinear groups, called the improved-YRL scheme, which is secure against the proposed attack. We also prove the security of the improved-YRL scheme in the standard model using dual system encryption technique [20]. Our proof is based on the security model for adaptive CCA adversaries proposed in [17] which considers anonymity and indistinguishability in one security game, simultaneously;
- We demonstrate that the new scheme retains low overhead and high-performance property of the basic YRL scheme, which means that computation and communication overheads are linear in the number of attributes independent of the number of receivers.

Boneh et al. [6] proposed that "it is a long-standing open problem to build a low-overhead anonymous broadcast encryption system"; therefore, presenting the improved-YRL scheme as an anonymous broadcast encryption scheme with adaptive security and overhead proportional to the number of attributes is an effort toward solving this open problem.

The paper is organized as follows. Section 2 is dedicated to the background on bilinear groups and state the access policy used in the YRL scheme. Section 3 concerns the YRL scheme. Section 4 proposes the attack on the YRL scheme. In Sections 5 and 6, we describe our improved-YRL scheme and prove its security, respectively. Section 7 gives the performance evaluation; finally, Section 8 concludes the paper.

2. Preliminaries

2.1. Bilinear maps

The YRL scheme is based on bilinear maps. Let G and G_T be two multiplicative cyclic groups of prime order p. Let g be a generator of G and e be a bilinear map, $e : G \times G \rightarrow G_T$. The bilinear map e is a function with the following properties:

1. Bilinearity: for all $u, v \in G$ and $a, b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$;
2. Non-degeneracy: $e(g, g) \neq 1$, where 1 denotes the identity element of G_T;
3. Computability: There is an efficient algorithm to compute $e(u, v)$ for $u, v \in G$.

2.2. Composite order bilinear maps

Composite order bilinear maps were first introduced in [21]. Let $\mathcal{G}$ be a group generator algorithm. It takes as input a security parameter, λ, and outputs a tuple $(N = p_1p_2p_3, G, G_T, e)$, where p_1, p_2, p_3 are distinct prime numbers, G and G_T are multiplicative cyclic groups of composite order $N = p_1p_2p_3$ and $e : G \times \ G \rightarrow \ G_T$ is a composite order bilinear map. For each $p_i, i \in \{1, 2, 3\}$, let G_{p_i} be a subgroup of G of order p_i with a generator named as g_i. Each $T \in G$ can be represented as $T = X_1X_2X_3$ where $X_i \in G_{p_i}$ is referred to as the "G_{p_i} component of G". In addition, for all $x, y, z \in \{1, p_1, p_2, p_3\}$, G_{xyz} denotes a subgroup of order xyz in G. To generate a random element $r \in G_{p_i}$, one can set $r = g_i^\alpha$ where α is a random element in Z_{p_i}.

The main property of composite order bilinear maps is that subgroups $G_{p_1}, G_{p_2}, G_{p_3}$ are *orthogonal* under the bilinear map e, meaning that if $h \in G_{p_i}$ and $u \in G_{p_j}$ for $i \neq j$, then $e(h, u) = 1$. The other properties of composite order bilinear maps are the same as prime order bilinear maps described in Subsection 2.1.

2.3. Access policy

Herein, the access policy used in the YRL scheme is reviewed to specify the intended group of receivers [3]. Let n denote the total number of attributes. Each user is assigned an n-element string $\{Att_{i,b} \mid \forall i \in Z_n, b = 0$ or $1\}$ such that $Att_{i,0}$ and $Att_{i,1}$ show the negative and positive incidents of the i-th attribute, respectively. In other words, the binary sequence $X_{n-1}X_{n-2}...X_0$ can be used to demonstrate the attribute set of each user. In this sequence, the bit '0' implies that the user does not have the corresponding attribute, and the bit '1' shows that the user owns that attribute.

The access policy is demonstrated using AND logic. For example, $(Att_{3,1} \wedge Att_{1,0})$ or $X_3\bar{X}_1$ is used for showing the access policy for the users with the 3rd attribute and do not possess the 1st attribute. Here, X_2 is don't-care, i.e., for this access policy, it is not important what the value of X_2 is.

3. The YRL Scheme

The YRL scheme [3] relies on the notion of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) in which the intended subset of users, S, is specified with the access policy, T. In the YRL scheme, access policy T is not broadcasted along with the ciphertext, and the authorized users can decrypt the received messages without knowing the access structure. In this way, the scheme can provide the anonymity of users. The scheme consists of four algorithms: Setup, KeyGen, Encryption, and Decryption. The scheme is reviewed in the following.

Setup $(\lambda, X_{n-1}X_{n-2}...X_0) \rightarrow (PP, MK)$: The input parameters of the setup algorithm include security parameter, λ, and the attribute set of each user, $X_{n-1}X_{n-2}...X_0$, and the outputs include public parameters, PP, and Master Key, MK, generated as follows. Due to security parameter, λ, the algorithm chooses a group, G, of prime order, p, with generator, g. Each attribute in vector $X_{n-1}X_{n-2}...X_0$ is mapped to one of the members of G. Consider that attribute $Att_{i,b}$ is mapped to $h_{i,b}$ as a member of G. The Setup algorithm randomly selects $(a_i, b_i) \in Z_p$ and sets the values of $h_{i,0}$ and $h_{i,1}$ equal to g^{a_i} and g^{b_i}, respectively and sets $\gamma_i = a_i + b_i$. Finally, this algorithm outputs $PP = (p, g, G)$ as the public parameters and master key, $MK = (\alpha, \beta, \{a_i, b_i\}_{\forall i \in Z_n})$, where $\alpha, \beta \in_R Z_N$. MK is only held by the broadcaster.

KeyGen $(MK, X_{n-1}X_{n-2}...X_0) \rightarrow SK$: The input parameters of the KeyGen algorithm include master key, MK, and the attribute set of each user, $X_{n-1}X_{n-2}...X_0$, and its output is the tuple $(D, \hat{D}, \breve{D}, \{D_i\}_{\forall i \in Z_n})$ as the secret key of the user which is generated through Eq. (1):

$$SK = (D = g^{(\alpha+r)/\beta}, \hat{D} = g^r, \breve{D} = g^{\beta r},$$

$$\{D_i = h^r_{i,\bar{X}_i}\}_{\forall i \in Z_n}). \tag{1}$$

The parameters $(g, \alpha, \beta, h_{i,\bar{X}_i})$ are defined in the setup algorithm, above. In addition, r is a random element chosen from Z_p.

Encryption $(GK, T, MK) \rightarrow CT$: In order to distribute the group key GK, the broadcaster first encrypts GK using this algorithm. The input parameters of the Encryption algorithm include GK, access policy T, and master key, $MK = (\alpha, \beta, \{a_i, b_i\}_{\forall i \in Z_n})$; the output is the ciphertext $CT = (\tilde{C}, \breve{C}, \{\hat{C}_j\}_{j=0,1}, \{C_i\}_{\forall i \in Z_n})$. The first component of

CT is in the form of $\tilde{C} = (GK \parallel MAC).X$, where X is a blinding factor to hide the value $(GK \parallel MAC)$. The other three components of the ciphertext are used to construct X and obtain GK. In the last component, each C_i is generated corresponding to the i-th bit of the attribute set, $X_{n-1}X_{n-2}...X_0$, through steps 1 to 4:

1. The random values, $s_0, s_1, s_{n-1}, k_0, k_1 \in_R Z_p$, are chosen and δ is set equal to $\sum_{i=0}^{n-1} \gamma_i s_i$; where γ_i has been determined in the Setup algorithm.
2. C_i is equal to the tuple $(g^{s_i}, C_{i,0}, C_{i,1})$ where $C_{i,0}$ and $C_{i,1}$ are the members of G. If $X_i \in T$, the i-th attribute is an intended attribute in the access policy. Then a random value, $t_i \in_R Z_p$, is chosen, and the values $C_{i,X_i} = h_{i,X_i}^{s_i+t_i}$ and $C_{i,1-X_i} = h_{i,1-X_i}^{s_i}$ are computed. Otherwise, $C_{i,0}$ and $C_{i,1}$ are set equal to $h_{i,0}^{s_i}$ and $h_{i,1}^{s_i}$, respectively.
3. The broadcaster sets $g^{s'} = \prod_{i=0}^{n-1} C_{i,0}C_{i,1} = g^{\delta+x}$ where δ has been defined in step 1 and $x \in Z_p$ satisfies the equation $g^x = \prod_{\forall j, X_j \in T} h_{j,X_j}^{t_j}$. Then, the second term of the ciphertext is calculated as $\breve{C} = g^{\beta s'}$, and the values of $C_{i,0}$ and $C_{i,1}$ are updated as follows:

$$C_{i,0} = g^{k_0} C_{i,0}, \qquad C_{i,1} = g^{k_1} C_{i,1}. \tag{2}$$

4. Finally, the ciphertext is generated through Eq. (3):

$$\begin{aligned} CT =& (\tilde{C} = (GK \parallel MAC) e(g,g)^{\alpha s'}, \\ & \breve{C} = g^{\beta s'}, \{\hat{C}_j = g^{k_j/\beta}\}_{j=0,1}, \{C_i\}_{\forall i \in Z_n}), \end{aligned} \tag{3}$$

where $MAC = H(GK)$ ($H(.)$ is a cryptographic hash function).

Decryption $(CT, X_{n-1}X_{n-2}...X_0, SK) \rightarrow (GK \text{ or} \perp)$: Each authorized group member, GM, runs this algorithm to obtain the group key GK. The inputs of this algorithm include the ciphertext, the attribute set, $X_{n-1}X_{n-2}...X_0$, and the secret key of the GM. The output is GK or $\perp$ depending on whether the attribute set of the GM satisfies the access structure or not. The decryption procedure is as follows:

1. For $j = 0, 1, B_j = e(\hat{C}_j, \breve{D}) = e(g^{k_j/\beta}, g^{\beta r}) = e(g,g)^{rk_j}$ is calculated. Then, for each bit X_i of the user's attribute set, $X_{n-1}X_{n-2}...X_0$, the value F_i corresponding to X_i is computed using $C_i = (g^{s_i}, C_{i,0}, C_{i,1})$ through Eq. (4).

$$\begin{aligned} F_i &= e(D_i, g^{s_i}) e(C_{i,X_i}, \hat{D})/B_{X_i} \\ &= e(h_{i,\bar{X}_i}^{r}, g^{s_i}) e(g^{k_{X_i}} h_{i,X_i}^{s_i+t_i}, g^r)/B_{X_i} \\ &= e(g,g)^{r\gamma_i s_i} e(g, h_{i,X_i})^{rt_i}. \end{aligned} \tag{4}$$

In Eq. (4), if $X_i \in T$, $t_i \neq 0$; otherwise, $t_i = 0$.

2. F is computed by multiplying the values of F_i:

$$\begin{aligned} F &= \prod_{i=0}^{n-1} F_i = \prod_{i=0}^{n-1} e(g,g)^{r\gamma_i s_i} e(g, h_{i,X_i})^{rt_i} \\ &= e(g,g)^{r\delta} e(g,g)^{rx'}, \end{aligned} \tag{5}$$

where $x' \in Z_p$ is defined in Eq. (6):

$$g^{x'} = \prod_{i=0}^{n-1} h_{i,X_i}^{t_i}, (t_i = 0 \text{ if } X_i \notin T). \tag{6}$$

Therefore, according to the properties of bilinear maps, we will have:

$$\prod_{i=0}^{n-1} e(g, h_{i,X_i})^{rt_i} = e(g,g)^{rx'}. \tag{7}$$

Eq. (7) is used to calculate the value of F in Eq. (5).

If the GM's attributes satisfy the access policy, x' will be equal to x. Otherwise, the probability of x' being equal to x will be negligible (note that x has been defined in the Encryption algorithm as $g^x = \prod_{\forall j, X_j \in T} h_{j,X_j}^{t_j}$).

3. Finally, each group member of GM computes M' and recovers the Group Key, GK, using Eq. (8):

$$\begin{aligned} M' &= \frac{\tilde{C}}{e(\breve{C}, D)/F} = \frac{(GK \parallel MAC) e(g,g)^{\alpha s'}}{e(g,g)^{\alpha s' + rs'}/e(g,g)^{r(\delta + x')}} \\ &= (GK \parallel MAC) e(g,g)^{r(x'-x)}. \end{aligned} \tag{8}$$

Then, the user checks whether the hash value of the first part of M' is equal to its second part or not. If the user is a member of the target subset, x' will be equal to x; as a result, the hash value of the first part will be equal to the second part. Thus, the user obtains the correct GK. Otherwise, the user is unauthorized and cannot obtain GK.

As can be seen, in the YRL scheme, access structure, T, is not sent along with the ciphertext and the authorized users are able to decrypt the received message without knowing the access structure. As a result, the authorized user, after decryption, does not know which attributes or how many of them make the message accessible to him. In addition, he is not aware of the membership of other users in the subset or even the number of authorized users. Therefore, not only the unauthorized users but also the authorized ones are not able to obtain any information about the access structure, and the YRL scheme provides the anonymity property.

Yu et al. [3] also claimed that their introduced scheme was secure. It means that a user can obtain the correct GK iff his attributes satisfy the access policy. However, no proof is provided for neither anonymity nor security in their paper. In the next section, we propose an attack, which violates the security of the YRL scheme.

4. Attack on the YRL scheme

Herein, we demonstrate that the claim that a user can obtain GK iff he holds all the attributes required by the access policy, is not true; further to that, all of the users, including authorized and unauthorized ones, can decrypt the received message. Assume that user u with secret key SK_u has received a broadcasted ciphertext CT. As mentioned in Section 3, the secret key SK_u and the ciphertext CT are computed as follows:

$$\begin{aligned} SK_u = (D = g^{(\alpha+r)/\beta}, \hat{D} = g^{r}, \breve{D} \\ = g^{\beta r}, \{D_i = h^{r}_{i,\bar{X}_i}\}_{\forall i \in Z_n}), \end{aligned} \tag{9}$$

$$\begin{aligned} CT = (\tilde{C} = (GK \parallel MAC)e(g,g)^{\alpha s'}, \breve{C} \\ = g^{\beta s'}, \{\hat{C}_j = g^{k_j/\beta}\}_{j=0,1}, \{C_i\}_{\forall i \in Z_n}). \end{aligned} \tag{10}$$

Now, user u can decrypt the ciphertext using his secret key through the following procedure:

1. Computes:

$$e(D, \breve{C}) = e(g^{(\alpha+r)/\beta}, g^{\beta s'}) = e(g,g)^{(\alpha+r)s'}.$$

2. Calculates $e(g,g)^{rs'}$: As mentioned in the Decryption algorithm, s' satisfies the equation:

$$g^{s'} = \prod_{i=0}^{n-1} C_{i,0}C_{i,1},$$

however, after that, $C_{i,0}$ and $C_{i,1}$ were updated to new values, $C_{i,0} = g^{k_0}C_{i,0}$, and $C_{i,1} = g^{k_1}C_{i,1}$. Since $C_i = (g^{s_i}, C_{i,0}, C_{i,1})$, user u can compute $\prod_{i=0}^{n-1} C_{i,0}C_{i,1}$ by using C_is:

$$C_{i,0} = g^{k_0}C_{i,0}, \qquad C_{i,1} = g^{k_1}C_{i,1}$$

$$\prod_{i=0}^{n-1} C_{i,0}C_{i,1} = \prod_{i=0}^{n-1} g^{k_0}C_{i,0}.g^{k_1}C_{i,1} = g^{n(k_0+k_1)}g^{s'}. \tag{11}$$

Therefore:

$$\begin{aligned} e(g^{n(k_0+k_1)}g^{s'}, \hat{D}) &= e(g^{n(k_0+k_1)}g^{s'}, g^{r}) \\ &= e(g^{s'}, g^{r})e(g^{n(k_0+k_1)}, g^{r}) \\ &= e(g,g)^{rs'}e(g,g)^{rn(k_0+k_1)}. \end{aligned} \tag{12}$$

Furthermore, we have:

$$\begin{aligned} B_j = e(\hat{C}_j, \breve{D}) = e(g^{k_j/\beta}, g^{\beta r}) = e(g,g)^{rk_j}, \\ j = 0,1 \rightarrow B_0.B_1 = e(g,g)^{r(k_0+k_1)}. \end{aligned} \tag{13}$$

Thus, using Eqs. (12) and (13), the user u can obtain $e(g,g)^{rs'}$ as follows:

$$\frac{(12)}{(13)^n} = \frac{e(g,g)^{rs'}e(g,g)^{rn(k_0+k_1)}}{(e(g,g)^{r(k_0+k_1)})^n} = e(g,g)^{rs'}. \tag{14}$$

3. Computes $e(g,g)^{\alpha s'}$ by dividing the result of the first step of the attack, $e(g,g)^{(\alpha+r)s'}$, by the result of the second step, $e(g,g)^{rs'}$.

$$\frac{e(g,g)^{(\alpha+r)s'}}{e(g,g)^{rs'}} = e(g,g)^{\alpha s'}. \tag{15}$$

4. Finally, user u can obtain GK as follows:

$$\begin{aligned} \frac{\tilde{C}}{e(g,g)^{\alpha s'}} &= \frac{(GK \parallel MAC)e(g,g)^{\alpha s'}}{e(g,g)^{\alpha s'}} \\ &= (GK \parallel MAC). \end{aligned} \tag{16}$$

Therefore, this user, regardless of what his set of attributes is, can obtain GK. This shows that the YRL scheme is not secure and does not provide the main requirement of a broadcast encryption scheme that only the intended users should be able to decrypt the broadcasted message.

5. Improved-YRL scheme

In this section, we improve the YRL scheme in order to remove its weakness and make it secure against the proposed attack in Section 4. The update procedure of $C_{i,0}$ and $C_{i,1}$ is the vulnerability point of the YRL scheme. As mentioned before, for all $i \in Z_n$, the broadcaster uses fixed values k_0, and k_1 for updating $C_{i,0}$ and $C_{i,1}$ to new values $C_{i,0} = g^{k_0}C_{i,0}$, and $C_{i,1} = g^{k_1}C_{i,1}$. Therefore, $e(\prod_{i=0}^{n-1} C_{i,0}C_{i,1}, \hat{D})$ has a fixed term $e(g,g)^{rn(k_0+k_1)}$ which can be omitted using the term $\hat{C}_j$ in the ciphertext. As a result, $e(g,g)^{rs'}$ and $e(g,g)^{\alpha s'}$ are obtained which help the attacker to obtain GK. Hence, in order to fix this weakness, we randomize the update process and eliminate the third term in both the ciphertext and secret key. In addition, in order to propose a security proof, the Improved-YRL scheme is based on composite order bilinear maps. In what follows, the improved-YRL scheme is describe in detail:

Setup $(\lambda, X_{n-1}X_{n-2}...X_0) \rightarrow (PP, MK)$: The input parameters of the setup algorithm include security parameter, λ, and the attribute set of each user, $X_{n-1}X_{n-2}...X_0$, and the outputs are public parameters, PP, and master key, MK, which are generated as follows. Due to security parameter, λ, the algorithm selects a cyclic group, G, of composite order, $N = p_1p_2p_3$. Let G_{p_1}, G_{p_2}, and G_{p_3} be three subgroups of G with orders p_1, p_2, p_3 and generators g_1, g_2, g_3, respectively. Then, the same as before, each of attributes, $Att_{i,b}$, is mapped to $h_{i,b}$. $h_{i,0}$ and $h_{i,1}$ are set equal to $g_1^{a_i}R_{3,0}$ and $g_1^{b_i}R_{3,1}$, respectively, where a_i and b_i are randomly selected forms of Z_{p_1} and $\gamma_i = a_i + b_i$. The only difference here is that $h_{i,0}$ and $h_{i,1}$ have additional factors $R_{3,0}$ and $R_{3,1}$ which are randomly chosen from G_{p_3}. Hence $h_{i,b}$ is an element of subgroup $G_{p_1p_3}$. The algorithm outputs, $PP = (N = p_1p_2p_3, G)$, as the public parameters and master key, $MK = (\alpha, \beta, \{a_i, b_i\}_{\forall i \in Z_n})$, where $\alpha, \beta \in_R Z_N$ and MK is only held by the broadcaster as before.

KeyGen $(MK, X_{n-1}X_{n-2}...X_0) \rightarrow SK$: This algorithm takes the master key, MK, the attribute set of a user, $X_{n-1}X_{n-2}\ldots X_0$, chooses random $r \in_R Z_{p_1}$ and outputs $(D, \hat{D}, \{D_i\}_{\forall i \in Z_n})$ as the secret key of the user through Eq. (17):

$$SK = (D = g_1^{(\alpha+r)/\beta}, \hat{D} = g_1^r, \{D_i = h_{i,\bar{X}_i}^r\}_{\forall i \in Z_n}). \quad (17)$$

The parameters $(g_1, \alpha, \beta, h_{i,\bar{X}_i})$ are defined in the setup algorithm above. Therefore, it is the same as the YRL's KeyGen algorithm except that the third term $\breve{D} = g^{\beta r}$ is omitted and D_is are the members of $G_{p_1p_3}$.

Encryption $(GK, T, MK) \rightarrow CT$: As before, The inputs are group key, GK, access policy, T, and master key, MK; the output is the ciphertext CT. However, this algorithm has some differences with the YRL's Encryption algorithm. The first difference is the procedure for updating the values of $C_{i,0}$ and $C_{i,1}$, and the second one omit the term $\{\hat{C}_j = g^{k_j/\beta}\}_{j=0,1}$ from the ciphertext because the decryption successfully works without it. In addition, g is turned into g_1; therefore, the first term in C_i is $g_1^{s_i} \in G_{p_1}$; because h_{i,X_i} is an element of $G_{p_1p_3}$ as stated in the setup algorithm, $C_{i,0}$ and $C_{i,1}$ are elements of $G_{p_1p_3}$, too. Therefore, the ciphertext:

$$CT = (\tilde{C}, \breve{C}, \{C_i\}_{\forall i \in Z_n})$$

is generated as follows:

1. The values $s_0, s_1, s_{n-1} \in_R Z_{p_1}$ are randomly selected and δ is set equal to $\sum_{i=0}^{n-1} \gamma_i s_i$.
2. $\forall i \in Z_n : C_i = (g_1^{s_i}, C_{i,0}, C_{i,1})$, where $C_{i,0}$ and $C_{i,1}$ belong to the group $G_{p_1p_3}$. If $X_i \in T$ (the i-th attribute is an intended attribute in the access policy), then a random value $t_i \in_R Z_{p_1}$ is selected and the values of C_{i,X_i} and $C_{i,1-X_i}$ are set equal to $h_{i,X_i}^{s_i+t_i}$ and $h_{i,1-X_i}^{s_i}$, respectively. Otherwise, $C_{i,0} = h_{i,0}^{s_i}$ and $C_{i,1} = h_{i,1}^{s_i}$.
3. The term $g_1^{s'}$ is computed through Eq. (18):
$$\prod_{i=0}^{n-1} C_{i,0}C_{i,1} = g_1^{s'}R_3 = g_1^{\delta+x}R_3, \quad (18)$$
where R_3 is the product of all elements in G_{p_3}. In addition, $x \in Z_{p_1}$ is such that $g_1^x = \prod_{\forall j, X_j \in T} h_{j,X_j}^{t_j}$, where only G_{p_1} part of h_{j,X_j} is considered. Then the value of $\breve{C}$ is set equal to $(g_1^{s'}R_3)^\beta = g_1^{\beta s'}R_3' \in G_{p_1p_3}$. In other words, $\breve{C}$ has an extra R_3' term in comparison to the basic YRL scheme.
4. $C_{i,0}$ and $C_{i,1}$ are updated as follows: If $X_i \in T$, the random value $k_i \in Z_{p_1}$ is selected and:
$$C_{i,X_i} = C_{i,X_i}, C_{i,1-X_i} = g_1^{k_i}C_{i,1-X_i}.$$
In this way, only $C_{i,1-X_i}$, which is not intended in the access policy, is updated; C_{i,X_i}, which is intended in the access policy does not change. Otherwise, if $X_i \notin T$ or X_i is don't-care, $C_{i,0}$ and $C_{i,1}$ remain unchanged. For example, if we have an access policy with $n = 4$ and $T = \bar{X}_3X_2X_0$, only the values of $C_{0,0}$, $C_{2,0}$, and $C_{3,1}$ are updated and the other values do not change.
5. Finally, the ciphertext is computed as:
$$CT = (\tilde{C} = (GK \parallel MAC)e(g_1, g_1)^{\alpha s'},$$
$$\breve{C} = g_1^{\beta s'}R_3', \{C_i\}_{\forall i \in Z_n}),$$
where $e(g_1, g_1)^{\alpha s'}$ is computed as:
$$e(\prod_{i=0}^{n-1} C_{i,0}C_{i,1}, g_1^\alpha) = e(g_1^{s'}R_3, g_1^\alpha) = e(g_1, g_1)^{\alpha s'}.$$
The last equality holds due to the orthogonality property of composite order bilinear maps.

Decryption $(CT, X_{n-1}X_{n-2}...X_0, SK) \rightarrow (GK \text{ or} \perp)$: The inputs of this algorithm include the ciphertext, the attribute set $X_{n-1}X_{n-2}...X_0$, and the private key of a GM; in addition, its output is GK or $\perp$ depending on whether the GM's attribute set satisfies the access structure or not.

In this algorithm, the first step of the basic YRL's Decryption algorithm for calculating B_j is omitted. In addition, F_is are computed in a simpler way:

1. For each bit, X_i, of the GM's attribute set, $X_{n-1}X_{n-2}...X_0$, F_i is computed through Eq. (19):
$$\begin{aligned} F_i &= e(D_i, g_1^{s_i})e(C_{i,X_i}, \hat{D}) \\ &= e(h_{i,\bar{X}_i}^r, g_1^{s_i})e(g_1^{k_i}h_{i,X_i}^{s_i+t_i}, g_1^r) \\ &= e(g_1, g_1)^{r\gamma_i s_i}e(g_1, h_{i,X_i})^{rt_i}e(g_1, g_1)^{k_i r}. \quad (19) \end{aligned}$$

It can be easily verified that the elements of G_{p_3} are omitted due to the orthogonality property of composite order bilinear maps. In addition, values of parameters t_i and k_i change due to the following conditions:

- $X_i \in T, t_i \neq 0$ and $k_i = 0$;
- $\bar{X}_i \in T, t_i = 0$ and $k_i \neq 0$;
- $X_i \notin T, t_i = 0$ and $k_i = 0$.

2. The GM calculates F by multiplying the values of F_is obtained in the previous step:

$$F = \prod_{i=0}^{n-1} F_i = \prod_{i=0}^{n-1} e(g_1, g_1)^{r\gamma_i s_i} e(g_1, h_{i,X_i})^{rt_i} e(g_1, g_1)^{k_i r}$$

$$= e(g_1, g_1)^{r\delta} e(g_1, g_1)^{rx'} e(g_1, g_1)^{kr}. \tag{20}$$

x' satisfies Eq. (21) where only G_{p_1} part of h_{j,X_j} is considered:

$$g_1^{x'} = \prod_{i=0}^{n-1} h_{i,X_i}^{t_i}, (t_i = 0 \text{ if } X_i \notin T). \tag{21}$$

If the GM's attributes satisfy the access policy, then we will have:

- $x' = x$;
- $\forall i, k_i = 0 \rightarrow k = 0$.

Otherwise, the probability of $x' = x$ or $k = 0$ will be negligible.

3. Each user calculates M' corresponding to his attributes through Eq. (22) to obtain GK:

$$M' = \frac{\tilde{C}}{e(\breve{C}, D)/F}$$

$$= \frac{(GK \parallel MAC) e(g_1, g_1)^{\alpha s'}}{e(g_1, g_1)^{\alpha s' + rs'} / e(g_1, g_1)^{r(\delta + x' + k)}}$$

$$= (GK \parallel MAC) e(g_1, g_1)^{r(k + x' - x)}. \tag{22}$$

Then, each user verifies whether the hash value of the first part of M' is equal to its second part or not. If the user is a member of the target group, this equality will be obtained because $x = x'$ and $k = 0$. Otherwise, the user is unauthorized and cannot obtain the correct value of GK.

In the next section, we will analyze the security of the proposed scheme, and prove that it achieves both indistinguishability and anonymity in the standard model.

6. Security analysis

This section begins by explaining why the proposed attack in Section 4 would not succeed on the Improved-YRL construction. Then, in order to prove the security of the proposed scheme, we will formally define the exact security definition in Subsection 6.1. Next, we state the complexity assumptions in composite order bilinear groups and present the proof in Subsections 6.2 and 6.3, respectively.

Lemma 1. The Improved-YRL scheme is secure against the proposed attack in Section 4.

Proof. The attack process involves computing the equation of $\prod_{i=0}^{n-1} C_{i,0} C_{i,1} = g_1^{k_r}.g_1^{s'} R_3$, where k_r is the sum of all k_i values used for updating $C_{i,0}$ and $C_{i,1}$ in the update phase of the improved-YRL scheme. k_r is completely random and unpredictable because there is no term in the ciphertext containing information about it. Therefore, the attacker will not be able to obtain $e(g_1, g_1)^{rs'}$, and the blinding factor, $e(g_1, g_1)^{\alpha s'}$, and the attack will not work.

6.1. Security definitions

In this section, a model is defined for the anonymous broadcast encryption with CCA security against adaptive adversaries. This model is a modification of the security model defined in [17].

Definition 1. ANO-IND-CCA security game for a broadcast encryption scheme, BE, is as follows.

Setup. Challenger $\mathcal{C}$ runs Setup algorithm to generate master key MK.

Phase 1. Adversary $\mathcal{A}$ issues queries for secret keys corresponding to the set of attributes Att_1, $Att_2, \ldots, Att_{q'}$. Challenger $\mathcal{C}$ runs KeyGen algorithm and returns the corresponding secret keys $SK_1, SK_2, \ldots, SK_{q'}$ to $\mathcal{A}$. $\mathcal{A}$ can also make decryption queries (CT, Att_i), meaning that decryption of ciphertext CT for user i with attribute set Att_i, and challenger $\mathcal{C}$ will return the decrypted message or $\perp$ using Decryption algorithm.

Challenge. $\mathcal{A}$ submits two equal length group keys GK_0 and GK_1 and two access policies T_0^* and T_1^*. The submitted access policies T_0^* and T_1^* should be such that none of the queried attribute sets $Att_1, Att_2, \ldots, Att_{q'}$ in Phase 1 satisfy them. Then, the challenger chooses a random bit $b \in \{0, 1\}$ and encrypts GK_b under T_b^* using Encryption algorithm and returns CT^* to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues querying secret keys corresponding to the set of attributes $Att_{q'+1}$, $Att_{q'+2}, \ldots, Att_q$, none of which satisfies T_0^* or T_1^* and receives corresponding secret keys $SK_{q'+1}, SK_{q'+2}, \ldots, SK_q$. In addition, $\mathcal{A}$ continues making decryption queries (CT, Att_i) with the restriction that if $CT = CT^*$, then Att_i should not satisfy any of T_0^* or T_1^*.

Guess. $\mathcal{A}$ outputs b' as its guess for b and wins the game if $b = b'$. The advantage of $\mathcal{A}$ in this game is defined as $Adv_{\mathcal{A},BE}^{ANO-IND-CCA}(\lambda) = |Pr[b = b'] - 1/2|$.

Definition 2. A broadcast encryption scheme, BE, is said to be anonymous and indistinguishable against CCA adversaries or is ANO-IND-CCA secure if any PPT adaptive CCA adversary has at most a negligible advantage in the above security game.

6.2. Complexity assumptions

In what follows, we state three complexity assumptions in composite order bilinear groups which we will rely on to prove the security of the improved-YRL scheme. Rao and Dutta [22] closely followed [23] to show that Assumptions 1, 2, and 3 hold in the generic group model under the assumption that finding a non-trivial factor of N, where $N = p_1p_2p_3$, is hard.

Assumption 1. Let $\mathcal{G}$ be a group generator and $\vec{y} = (N = p_1p_2p_3, G, G_T, e) \leftarrow \mathcal{G}(\lambda)$. Choose g_1, and g_3 randomly from G_{p_1} and G_{p_3}, respectively. Then, for each PPT adversary $\mathcal{A}$ which is given $D = (\vec{y}, g_1, g_3)$, $\mathcal{A}$'s advantage of distinguishing $T_0 \in G$ from $T_1 \in G_{p_1p_3}$ is negligible, where T_0 and T_1 are randomly chosen from the corresponding groups. In other words, for any PPT algorithm $\mathcal{A}$, we have:

$$\begin{aligned} Adv_{\mathcal{A}}^1 &= |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]| \\ &\leq negl(\lambda), \end{aligned} \tag{23}$$

where $negl(.)$ is a negligible function.

Assumption 2. Let $\mathcal{G}$ be a group generator and $\vec{y} = (N = p_1p_2p_3, G, G_T, e) \leftarrow \mathcal{G}(\lambda)$. Choose random elements $g_1 \in G_{p_1}$, $g_3 \in G_{p_3}$, $X_1X_2X_3 \in G$, $Y_1Y_2 \in G_{p_1p_2}$. Then, for each PPT adversary $\mathcal{A}$ which is given $D = (\vec{y}, g_1, g_3, X_1X_2X_3, Y_1Y_2)$, $\mathcal{A}$'s advantage of distinguishing $T_0 \in G_{p_1}$ from $T_1 \in G_{p_1p_2}$, is negligible, where T_0 and T_1 are randomly chosen from the corresponding groups. In other words, for any PPT algorithm $\mathcal{A}$, we have:

$$\begin{aligned} Adv_{\mathcal{A}}^2 &= |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]| \\ &\leq negl(\lambda), \end{aligned} \tag{24}$$

where $negl(.)$ is a negligible function.

Assumption 3. Let $\mathcal{G}$ be a group generator and $\vec{y} = (N = p_1p_2p_3, G, G_T, e) \leftarrow \mathcal{G}(\lambda)$. Choose random elements $\alpha, \varsigma \in \mathbb{Z}_N, g_1 \in G_{p_1}, X_2, Y_2 \in G_{p_2}, X_3, Y_3 \in G_{p_3}$. Then, for each PPT adversary $\mathcal{A}$ which is given $D = (\vec{y}, g_1, g_1^{\alpha}X_2, X_3, g_1^{\varsigma}Y_2Y_3)$, $\mathcal{A}$'s advantage of distinguishing $T_0 = e(g_1, g_1)^{\alpha\varsigma}$ from T_1 as a random element in G_T is negligible. In other words, for any PPT algorithm $\mathcal{A}$, we have:

$$\begin{aligned} Adv_{\mathcal{A}}^3 &= |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]| \\ &\leq negl(\lambda), \end{aligned} \tag{25}$$

where $negl(.)$ is a negligible function.

6.3. Security proof

Here we use a technique called dual system encryption [20] to prove the security of improved-YRL scheme in the ANO-IND-CCA security game described in Subsection 6.1. In a dual system, ciphertexts and secret keys can be either normal or semi-functional. Semi-functional terms are not part of the real system; however, they are only used in the security proof. A normal secret key, can decrypt both normal and semi-functional ciphertexts; however, a semi-functional secret key can only decrypt normal ciphertexts. In other words, one would fail to decrypt a semi-functional ciphertext using a semi-functional secret key. The semi-functional ciphertexts and secret keys for improved-YRL are defined as below.

Semi-functional ciphertext. To obtain a semi-functional ciphertext, first, Encryption algorithm is run to obtain a normal ciphertext $CT = (\tilde{C}, \breve{C}, \{C_i\}_{\forall i \in Z_n})$, where $C_i = (g^{s_i}, C_{i,0}, C_{i,1})$ for all $i \in Z_n$. Then, the semi-functional ciphertext $CT' = (\tilde{C}', \breve{C}', \{C_i{}'\}_{\forall i \in Z_n})$ is computed as follows:

$$\tilde{C}' = \tilde{C}, \quad \breve{C}' = \breve{C} \times {g_2}^{\delta\beta}, \quad C_i{}' = C_i \times {g_2}^{\delta/n}, \tag{26}$$

where δ is chosen randomly from Z_N and in $C_i{}' = C_i \times {g_2}^{\delta/n}$, only the terms $C_{i,0}$, and $C_{i,1}$ are multiplied by ${g_2}^{\delta/n}$. In addition, n is the number of attributes, and β is a part of master key as stated before.

Semi-functional secret key. To compute a semi-functional secret key for a user with the attribute set $X_{n-1}X_{n-2}...X_0$, we first run the algorithm KeyGen to obtain a normal secret key, $SK = (D, \hat{D}, \{D_i\}_{\forall i \in Z_n})$, where $D_i = h_{i,\bar{x_i}}^r$ for all $i \in Z_n$. The semi-functional secret key $SK' = (D', \hat{D}', \{D_i{}'\}_{\forall i \in Z_n})$ is computed as follows:

$$D' = D \times {g_2}^{\gamma/\beta}, \quad \hat{D}' = \hat{D} \times {g_2}^{\gamma}, \quad D_i{}' = D_i \times {g_2}^{\gamma e_i}, \tag{27}$$

where γ is a random element in Z_N and $e_i = a_i$ if $h_{i,\bar{x_i}} = g_1^{a_i}X_3$ and $e_i = b_i$ if $h_{i,\bar{x_i}} = g_1^{b_i}X_3$.

Security is proved using a sequence of games, which are proven to be indistinguishable under assumptions given in Section 6.2. Considering q as the maximum number of secret key queries an adversary can make, the sequence of games is as follows:

- $Game_{ANO-IND-CCA}$: In this game as described in Section 6.1, all ciphertexts and secret keys are normal;
- $Game_0$: Herein, the challenge ciphertext is semi-functional; however, all secret keys are normal;
- $Game_k (1 \leq k \leq q)$: In this game, in addition to the challenge ciphertext, the first k queried secret keys are semi functional, and the rest of them are normal. Therefore, in $Game_q$, all the secret keys would be semi-functional;
- $Game_{final}$: This game is the same as $Game_q$ except that the ciphertext is randomized. Therefore, the challenge ciphertext is independent of the group keys and access policies given by the adversary in the challenge step.

The sequence of hybrid games in the proof are related as follows:

$$Game_{ANO-IND-CCA} \Leftrightarrow Game_0 \Leftrightarrow Game_1 ...$$

$$\Leftrightarrow Game_{q-1} \Leftrightarrow Game_q \Leftrightarrow Game_{final},$$

where the notation "$\Leftrightarrow$" means that the two games are computationally indistinguishable. Now, the above relations are shown through the following lemmas.

Lemma 2. Suppose that there exists a polynomial time algorithm $\mathcal{A}$ such that:

$$Adv_{\mathcal{A}}^{Game_{ANO-IND-CCA}} - Adv_{\mathcal{A}}^{Game_0} = \varepsilon.$$

Then, we can build a PPT algorithm $\mathcal{B}$ with advantage ε in breaking Assumption 1.

Proof. $\mathcal{B}$ is given $(\vec{y}, g_1, g_3, T)$. It will simulate $Game_{ANO-IND-CCA}$ or $Game_0$ for $\mathcal{A}$ depending on whether T is an element of G or it is an element of $G_{p_1p_3}$. We now describe how $\mathcal{B}$ interacts with $\mathcal{A}$ to break Assumption 1.

Setup. $\mathcal{B}$ chooses random elements $\alpha, \beta \in Z_N$, and for each element of the attribute set, it chooses a_i, and b_i randomly from Z_{p_1} and keeps master key, $MK = (\alpha, \beta, \{a_i, b_i\}_{\forall i \in Z_n})$.

Phase 1 and Phase 2. $\mathcal{B}$ generates normal secret keys $SK = (D, \hat{D}, \{D_i\}_{\forall i \in Z_n})$ in response to $\mathcal{A}$'s secret key queries using the KeyGen algorithm. This is possible since $\mathcal{B}$ possess master key, MK, and g_1, g_3. As mentioned in the ANO-IND-CCA security game, none of the attribute sets queried in secret key requests should satisfy the access policies given by $\mathcal{A}$ in the challenge phase. In addition, in response to $\mathcal{A}$'s decryption requests (CT, Att_i), $\mathcal{B}$ generates the secret key corresponding to Att_i, and decrypts CT using Decryption algorithm.

Challenge. $\mathcal{A}$ sends $\mathcal{B}$ two equal length group keys GK_0 and GK_1 and two access policies T_0^* and T_1^*. $\mathcal{B}$ chooses a random bit b and performs the normal encryption of GK_b under T_b^* to obtain $CT_b = (\tilde{C}, \breve{C}, \{C_i\}_{\forall i \in Z_n})$. Next, in order to generate the challenge ciphertext $CT^* = (\tilde{C}', \breve{C}', \{C_i{}'\}_{\forall i \in Z_n})$, $\mathcal{B}$ performs the following calculations:

$$\tilde{C}' = \tilde{C} \times e(g_1^{\alpha}, T),$$

$$\breve{C}' = \breve{C} \times T^{\beta},$$

$$C'_{i_{0,1}} = C_{i_{0,1}} \times T^{1/n}. \tag{28}$$

Guess. $\mathcal{A}$ outputs a bit b' as its guess of b and wins if $b = b'$. It can be seen that if $T \in G_{p_1p_3}$, CT^* is a properly distributed *normal ciphertext* and $Game_{ANO-IND-CCA}$ is simulated. Else, if $T \in G$, CT^* is a properly distributed *semi-functional ciphertext* and we have $Game_0$. In addition, if ϵ is a non-negligible function, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between two values of T and break Assumption 1.

Lemma 3. Suppose that there exists a polynomial time algorithm $\mathcal{A}$ such that:

$$Adv_{\mathcal{A}}^{Game_{k-1}} - Adv_{\mathcal{A}}^{Game_k} = \varepsilon,$$

where $1 \leq k \leq q$ and q is the maximum secret key queries that $\mathcal{A}$ can make. Then we can build a PPT algorithm $\mathcal{B}$ that has advantage ε in breaking Assumption 2.

Proof. $\mathcal{B}$ is given $(\vec{y}, g_1, g_3, X_1X_2X_3, Y_1Y_2, T)$. It will simulate $Game_{k-1}$ or $Game_k$ depending on whether T is an element of G_{p_1} or is an element of $G_{p_1p_2}$. We now describe how $\mathcal{B}$ interacts with $\mathcal{A}$ to break Assumption 2.

Setup. $\mathcal{B}$ chooses random elements $\alpha, \beta \in Z_N$; for each element of the attribute set, it chooses a_i, and b_i randomly from Z_{p_1} and keeps master key, $MK = (\alpha, \beta, \{a_i, b_i\}_{\forall i \in Z_n})$.

Phase 1 and Phase 2. Herein, $\mathcal{B}$ responds to secret keys queries from $\mathcal{A}$ in three ways depending on the query number. For the first $k-1$ queries, $\mathcal{B}$ generates semi-functional secret keys. For this purpose, $\mathcal{B}$ first computes a normal secret key $(D, \hat{D}, \{D_i\}_{\forall i \in Z_n})$ using the KeyGen algorithm. Then

$SK_j = (D', \hat{D}', \{D_i{}'\}_{\forall i \in Z_n})$ for $1 \leq j \leq k-1$ is computed as below:

$$r_j \in_R Z_N,$$

$$D' = D \times (Y_1 Y_2)^{r_j/\beta},$$

$$\hat{D}' = \hat{D} \times (Y_1 Y_2)^{r_j},$$

$$D_i{}' = D_i \times (Y_1 Y_2)^{r_j e_i}, \tag{29}$$

where e_i can take two values: $e_i = a_i$ if $h_{i,\bar{x}_i} = g_1^{a_i} X_3$ and $e_i = b_i$ if $h_{i,\bar{x}_i} = g_1^{b_i} X_3$.

For query k, $\mathcal{B}$ first generates a normal secret key $(D, \hat{D}, \{D_i\}_{\forall i \in Z_n})$ and sets SK_k as follows:

$$D' = D \times T^{1/\beta},$$

$$\hat{D}' = \hat{D} \times T,$$

$$D_i{}' = D_i \times T^{e_i}, \tag{30}$$

where e_i is as defined above. It can be seen that if $T \in G_{p_1}$, SK_k is a normal secret key; if $T \in G_{p_1 p_2}$, SK_k is a semi-functional secret key.

Finally, for SK_j, $k+1 \leq j \leq q$, $\mathcal{B}$ simply generates a normal secret key.

In addition, in response to $\mathcal{A}$'s decryption requests (CT, Att_i), $\mathcal{B}$ generates the normal secret key corresponding to Att_i, and decrypts CT using Decryption algorithm.

Challenge. $\mathcal{A}$ sends $\mathcal{B}$ two equal length group keys and two access policies T_0^* and T_1^*. $\mathcal{B}$ chooses a random bit b and performs the normal encryption of GK_b under T_b^* to obtain $CT_b = (\tilde{C}, \breve{C}, \{C_i\}_{\forall i \in Z_n})$. Next, in order to generate the semi-functional ciphertext $CT^* = (\tilde{C}', \breve{C}', \{C_i{}'\}_{\forall i \in Z_n})$, $\mathcal{B}$ performs the following calculations:

$$\tilde{C}' = \tilde{C} \times e(g_1^{\alpha}, (X_1 X_2 X_3)),$$

$$\breve{C}' = \breve{C} \times (X_1 X_2 X_3)^{\beta}, \tag{31}$$

$$C'_{i_{0,1}} = C_{i_{0,1}} \times (X_1 X_2 X_3)^{1/n}.$$

CT^* is sent back to $\mathcal{A}$.

Guess. $\mathcal{A}$ outputs a bit b' as its guess of b and wins if $b = b'$. It can be seen that if $T \in G_{p_1}$, $\mathcal{B}$ has interaction with $\mathcal{A}$ in $Game_{k-1}$. Else, if $T \in G_{p_1 p_2}$, $Game_k$ is simulated. So $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between two values of T and break Assumption 2.

Lemma 4. Suppose that there exists a polynomial time algorithm $\mathcal{A}$ such that:

$$Adv_{\mathcal{A}}^{Game_q} - Adv_{\mathcal{A}}^{Game_{final}} = \varepsilon.$$

Then, a PPT algorithm $\mathcal{B}$ can be built that has advantage ε in breaking Assumption 3.

Proof. $\mathcal{B}$ is given $(\vec{y}, g_1, g_1^{\alpha} X_2, X_3, g_1^{\varsigma} Y_2 Y_3, T)$. It will simulate $Game_q$ or $Game_{final}$ depending on whether $T = e(g_1, g_1)^{\alpha\varsigma}$ or it is a random element of G_T. We now describe how $\mathcal{B}$ interacts with $\mathcal{A}$ to break Assumption 3.

Setup. $\mathcal{B}$ chooses random $\beta \in Z_N$, and for each element of the attribute set, it chooses a_i, and b_i randomly from Z_{p_1} and keeps master key, $MK = (\beta, \{a_i, b_i\}_{\forall i \in Z_n})$. Herein, $\mathcal{B}$ cannot choose parameter α himself, because this parameter is given to him via the term $g_1^{\alpha} X_2$. However, it can be easily shown that normal ciphertext and secret keys can be generated using the term $g_1^{\alpha} X_2$, without having α directly.

Phase 1 and Phase 2. Herein, all the queried secret keys are semi functional. In response to $\mathcal{A}$'s secret key queries, $\mathcal{B}$ first generates a normal secret key $(D, \hat{D}, \{D_i\}_{\forall i \in Z_n})$ using the KeyGen algorithm, $g_1, X_3, g_1^{\alpha} X_2$, and MK. Then, semi-functional secret key $SK_j = (D', \hat{D}', \{D_i{}'\}_{\forall i \in Z_n})$ for $1 \leq j \leq q$ is computed below:

$$r_j \in_R Z_N,$$

$$D' = D \times (g_1^{\alpha} X_2)^{r_j/\beta},$$

$$\hat{D}' = \hat{D} \times (g_1^{\alpha} X_2)^{r_j},$$

$$D_i{}' = D_i \times (g_1^{\alpha} X_2)^{r_j e_i}, \tag{32}$$

where e_i is as defined in Lemma 3.

In addition, in response to $\mathcal{A}$'s decryption requests (CT, Att_i), $\mathcal{B}$ generates the normal secret key corresponding to Att_i, and decrypts CT using the Decryption algorithm.

Challenge. $\mathcal{A}$ sends $\mathcal{B}$ two equal length group keys and two access policies T_0^* and T_1^*. $\mathcal{B}$ chooses a random bit b and performs the normal encryption of GK_b under T_b^* to obtain:

$$CT_b = (\tilde{C}, \breve{C}, \{C_i\}_{\forall i \in Z_n}).$$

Next, the challenge ciphertext:

$$CT^* = (\tilde{C}', \breve{C}', \{C_i{}'\}_{\forall i \in Z_n}),$$

is generated below:

$$\tilde{C}' = \tilde{C} \times T,$$

$$\breve{C}' = \breve{C} \times (g_1^{\varsigma} Y_2 Y_3)^{\beta}, \qquad (33)$$

$$C_{i_{0,1}}{}' = C_{i_{0,1}} \times (g_1^{\varsigma} Y_2 Y_3)^{1/n},$$

CT^* is sent back to $\mathcal{A}$.

Guess. $\mathcal{A}$ outputs a bit b' as its guess of b and wins if $b = b'$. It can be seen that if $T = e(g_1, g_1)^{\alpha\varsigma}$, $\mathcal{B}$ has interaction with $\mathcal{A}$ in $Game_q$. Else, if $T \in_R G_T$, the challenge ciphertext is randomized and $Game_{final}$ is simulated. Therefore, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between two values of T and break Assumption 3.

Theorem 1. If Assumptions 1, 2, and 3 hold, then Improved-YRL is an anonymous adaptive CCA secure broadcast encryption scheme.

Proof. We have shown in the previous lemmas that $Game_{ANO-IND-CCA}$ is indistinguishable from $Game_{final}$. In $Game_{final}$, the adversary receives no information about b information theoretically and the chance of any adversary in guessing the true b is exactly 1/2. Therefore, this is true in $Game_{ANO-IND-CCA}$ and the adversary cannot guess which GK is encrypted and also cannot obtain any information about access structure from the ciphertext with a probability greater than 1/2. Therefore, the improved-YRL scheme has both indistinguishability and anonymity and the proof is completed. □

7. Performance evaluation

7.1. Overhead analysis

This subsection analyzes the performance of the proposed scheme. For this aim, the computation, communication, and storage overheads are calculated in terms of the total number of attributes in the network, which is denoted by n. Modular multiplications, exponentiations, and pairings over composite orders are denoted by $Mul.cmp$, $Exp.cmp$, and $Pair.cmp$, respectively.

7.1.1. Computation Overhead.

Herein, we investigate the computation load of the Setup, KeyGen and Encryption algorithms of improved-YRL executed by the broadcaster and Decryption algorithm executed by the receivers. Similar to the YRL's setup overhead [3], improved-YRL's setup has a term $2nExp.cmp$. In addition, as in the improved-YRL's setup, some elements of subgroup G_1 are multiplied by some elements of subgroup G_3, and then the extra term $2nMul.cmp$ is added. Therefore, the total overhead of the improved-YRL's setup is $2nExp.cmp + 2nMul.cmp$. In the KeyGen algorithm, the term $g^{\beta r}$ is eliminated in comparison with KeyGen algorithm of the YRL scheme. Therefore, the computation overhead of the KeyGen algorithm is $(n + 2)Exp.cmp$. In the Encryption algorithm, the terms $g^{\frac{k_0}{\beta}}$ and $g^{\frac{k_1}{\beta}}$ are eliminated. Therefore, in comparison with the Encryption algorithm of the YRL scheme, two *Exp* computations are omitted and the resulting computation overhead is reduced to $(3n + 1)Exp.cmp$. In the Decryption algorithm, there is no need to compute $\{B_j = e(g^{\frac{k_j}{\beta}}, g^{\beta r})\}_{j \in \{0,1\}}$; consequently, the total number of pairing computations is reduced by two units. Therefore, the total computation overhead of Decryption algorithm becomes $nMul.cmp + (n + 2)Pair.cmp$. These results are summarized in Table 1.

7.1.2. Communication Overhead.

The total communication overhead of the proposed scheme is $(n + 1)\log_2 p_1 + 2n \log_2(p_1 p_3) + \log_2 |G_T^{cmp}|$, as presented in Table 1.

7.1.3. Storage Overhead.

The main storage load for users comes from the secret key SK. The storage load of the YRL scheme is $(n + 3)\log_2 p$ [3]; in addition, as in the Improved-YRL scheme, $g^{\beta r}$ is omitted from the secret key, and each user needs $2\log_2 p_1 + n\log_2(p_1 p_3)$ bits to store his secret key. These results are illustrated in Table 1.

Table 1. A summary of overhead analysis of our proposed scheme.

Criteria		**Improved-YRL**
Computation overhead	**Setup**	$2nExp.cmp + 2nMul.cmp$
	KeyGen	$(n + 2)Exp.cmp$
	Encryption	$(3n + 1)Exp.cmp$
	Decryption	$nMul.cmp + (n + 2)Pair.cmp$
Communication overhead		$(n + 1)\log_2 p_1 + 2n\log_2(p_1 p_3) + \log_2(\mathbb{G}_T^{cmp}\|)$
Storage overhead		$2\log_2 p_1 + n\log_2(p_1 p_3)$

Table 2. A comparison of anonymous broadcast encryption schemes; n denotes the number of attributes and N the number of the intended receivers.

Criteria	**Scheme**			
	Improved-YRL	**YRL [3]**	**[16]**	**[17]**
Encryption time	$O(n)$	$O(n)$	$O(N)$	$O(N)$
Decryption time	$O(n)$	$O(n)$	$O(N)$	$O(N)$
Key size	$O(n)$	$O(n)$	$O(1)$	$O(N)$
Communication overhead	$O(n)$	$O(n)$	$O(N)$	$O(N)$
Storage overhead	$O(n)$	$O(n)$	$O(1)$	$O(N)$

7.2. Discussion and comparison

Table 2 compares the overheads of our scheme with the basic YRL scheme [3] and two other anonymous broadcast encryption schemes [16,17], as discussed in the Introduction Section. In this table, parameters n and N denote the number of the attributes and number of the intended receivers, respectively. As in Table 2, the computation, communication, and storage overheads of the improved-YRL scheme, the same as the basic YRL scheme, are linear in the number of attributes, and independent of the number of receivers. Therefore, improved-YRL not only enjoys enhanced and provable security in comparison to basic YRL [3] and resists the proposed attack in this paper, but also preserves low overhead and high-performance properties of the basic YRL scheme.

Furthermore, a comparison between the proposed scheme and two selected anonymous broadcast encryption schemes [16,17] demonstrates that our scheme is much more efficient. That is, as can be seen in Table 2, the computation (encryption and decryption time) and communication overheads of both [16] and [17] grow linearly with regard to the number of the intended receivers. In this case, if 1024 bit RSA algorithm for encryption is used, then the ciphertext size will be $1024N$, where N is the total number of receivers. This implies a huge computation and communication overhead in the large-scale systems. Since the attributes are usually shared by the unlimited number of group members, it can be seen that in the case of large-scale applications, the computation and communication overheads of the improved YRL scheme, which are linear in the number of attributes, can be well controlled. In fact, in systems with large-scale structures, the number of the required attributes in comparison to the total number of users could be significantly small. In addition to better efficiency, the proposed scheme suggests stronger anonymity, that is, our scheme not only hides the identities of receivers, but also protects the *number* of intended receivers. In [16] and [17], the number of intended receivers is not protected.

Besides, according to [3], in order to minimize the communication overhead in a limited bandwidth environment such as wireless networks, or to minimize the computation overhead in resource-constrained receivers, there are different implementations of bilinear pairings. Depending on the application, the appropriate implementation should be chosen for minimizing the communication/computation overhead in practice.

8. Conclusion and future work

In this study, we investigated an anonymous broadcast encryption scheme, called YRL scheme and showed its vulnerability. Our investigation demonstrated that all of the users in this scheme, including authorized and unauthorized ones, could decrypt the received message. Thus, the YRL scheme did not provide the main requirement of the broadcast encryption schemes.

Since the introduction of an anonymous, efficient and provably secure broadcast encryption scheme is one of the most important open problems in this field, the YRL scheme in composite order bilinear groups was improved, making it secure against the proposed attack. We also proved anonymity and semantic security of the improved-YRL scheme under adaptive corruptions in the chosen ciphertext setting.

The same as the basic YRL, the computation and communication overheads of the improved-YRL scheme, as illustrated in Table 1, are $O(n)$, where n is the number of attributes and independent of the number of receivers. Since the attributes are usually shared by the unlimited number of group members, the scheme is more efficient than the anonymous broadcast encryption schemes with overheads related to the number of receivers [17].

Furthermore, presenting a real-world application of the proposed scheme, e.g., secure and scalable e-health architectures and secure cloud storage systems, can be considered as an interesting future work.

Acknowledgement

This research is research is partially supported by Center of Excellence in Cryptography and Information Security, Sharif University of Technology.

References

1. Fiat, A. and Naor, M. "Broadcast encryption", in *Annual International Cryptology Conference*, pp. 480-491, Springer (1993).
2. Aljawarneh, S. "A web engineering security methodology for e-learning systems", *Network Security*, **2011**(3), pp. 12-15 (2011).
3. Yu, S., Ren, K., and Lou, W. "Attribute-based on-demand multicast group setup with membership anonymity", *Computer Networks*, **54**(3), pp. 377-386 (2010).
4. Boneh, D., Gentry, C., and Waters, B. "Collusion resistant broadcast encryption with short ciphertexts and private keys", in *Annual International Cryptology Conference*, pp. 258-275, Springer (2005).
5. Boneh, D. and Waters, B. "A fully collusion resistant broadcast, trace, and revoke system", in *Proceedings of the 13th ACM Conference on Computer and Communications Security,* pp. 211-220, ACM (2006).
6. Boneh, D., Waters, B., and Zhandry, M. "Low overhead broadcast encryption from multilinear maps", in *International Cryptology Conference*, pp. 206-223, Springer (2014).
7. Guo, D., Wen, Q., Li, W., Zhang, H., and Jin, Z. "Adaptively secure broadcast encryption with constant ciphertexts", *IEEE Transactions on Broadcasting*, **62**(3), pp. 709-715 (2016).
8. Wu, Q., Qin, B., Zhang, L., Domingo-Ferrer, J., Farràs, O., and Manjon, J.A. "Contributory broadcast encryption with efficient encryption and short ciphertexts", *IEEE Transactions on Computers*, **65**(2), pp. 466-479 (2016).
9. Yang, Y., Yang, S.-L., Wang, F.-H., and Sun, J. "Post-quantum secure public key broadcast encryption with keyword search", *Journal of Information Science & Engineering*, **33**(2), pp. 485-497 (2017).
10. Sun, M., Ge, C., Fang, L., and Wang, J. "A proxy broadcast re-encryption for cloud data sharing", *Multimedia Tools and Applications*, pp. 1-15 (2017).
11. Aljawarneh, S., Yassein, M.B., et al. "A resource-efficient encryption algorithm for multimedia big data", *Multimedia Tools and Applications*, **76**(21), pp. 1-22 (2017).
12. Lubicz, D. and Sirvent, T. "Attribute-based broadcast encryption scheme made efficient", in *International Conference on Cryptology in Africa*, pp. 325-342, Springer (2008).
13. Zhou, Z. and Huang, D. "On efficient ciphertext-policy aribute based encryption and broadcast encryption", in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pp. 753-755, ACM (2010).
14. Aljawarneh, S.A., Alawneh, A., and Jaradat, R. "Cloud security engineering: Early stages of sdlc", *Future Generation Computer Systems*, **74**, pp. 385-392 (2017).
15. Wesolowski, B. and Junod, P. "Ciphertext-policy attribute-based broadcast encryption with small keys", in *International Conference on Information Security and Cryptology*, pp. 53-68, Springer (2015).
16. Barth, A., Boneh, D., and Waters, B. "Privacy in encrypted content distribution using private broadcast encryption", in *International Conference on Financial Cryptography and Data Security*, pp. 52-64, Springer (2006).
17. Libert, B., Paterson, K.G., and Quaglia, E.A. "Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model", in *Public Key Cryptography-PKC 2012*, pp. 206-224, Springer (2012).
18. Fazio, N. and Perera, I.M. "Outsider-anonymous broadcast encryption with sublinear ciphertexts", in *International Workshop on Public Key Cryptography*, pp. 225-242, Springer (2012).
19. Rabaninejad, R., Delavar, M., Ameri, M.H., and Mohajeri, J. "On the security of YRL, an anonymous broadcast encryption scheme", in *Telecommunications, IST 2016, International Symposium on. IEEE* (2016).
20. Waters, B. "Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions", in *Advances in Cryptology-CRYPTO*, 2009, pp. 619-636, Springer (2009).
21. Boneh, D., Goh, E.-J., and Nissim, K. "Evaluating 2-dnf formulas on ciphertexts", in *Theory of Cryptography Conference*, pp. 325-341, Springer (2005).
22. Sreenivasa Rao, Y. and Dutta, R. "Fully secure bandwidth-efficient anonymous ciphertext-policy attribute-based encryption", *Security and Communication Networks*, **8**(18), pp. 4157-4176 (2015).
23. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., and Waters, B. "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption", in *Advances in Cryptology-EUROCRYPT* 2010, pp. 62-91, Springer (2010).

Biographies

Reyhaneh Rabaninejad is currently a third year PhD student at the Electrical Engineering Department at K. N. Toosi University of Technology, Tehran, Iran. She received both her BSc and MSc degrees in Electrical Engineering from Sharif University of Technology in 2012 and 2014, respectively. Her research interests include network security and applied cryptography. Her current research area focuses on secure cloud storage and search mechanisms on encrypted data.

Mohammad Hassan Ameri received his BS degree in Electrical Engineering from Shahid Bahonar University, Kerman, Iran in 2013 with the honor of the first ranked student among electrical engineering students of the same entrance, and his MS degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran in 2015. Currently, he is working as a researcher in Electronic Research Institute at Sharif University of Technology. His major research interests include cloud security, searchable encryption, provable security, information-theoretic security, network security, design and cryptanalysis of cryptographic protocols.

Mahshid Delavar received the BSc degree from the Amirkabir University of Technology, Tehran, Iran in 2006, the MSc degree from Islamic Azad University, South Tehran Branch, Tehran, Iran in 2009, and the PhD degree from Iran University of Science and Technology, Tehran, Iran in 2016, all in Electronics Engineering. Her current research interests include hardware security and cryptographic protocols.

Javad Mohajeri is an Assistant Professor in Electronics Research Institute and an Adjunct Assistant Professor at the Electrical Engineering Department at Sharif University of Technology, Tehran, Iran. His research interests include design and cryptanalysis of cryptographic algorithms, and protocols and data security. He is the author/co-author of about 100 research articles in refereed journals/conferences and is one of the founding members of Iranian Society of Cryptology.