An Attribute-Based Anonymous Broadcast Encryption Scheme with Adaptive Security in the Standard Model

Reyhaneh Rabaninejad, Mohammad Hassan Ameri, Mahshid Delavar, Javad Mohajeri

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) considered this issue and introduced an efficient anonymous attribute-based broadcast encryption scheme. In this paper, we first propose an attack on the YRL scheme and show that the unauthorized receivers can also decrypt the broadcasted message. Next, we propose the Improved-YRL scheme and prove that it achieves anonymity and semantic security under adaptive corruptions in the chosen ciphertext setting. The proof is provided using 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.

Index Terms


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 the set $S$ and broadcasts the ciphertext through the channel. In a secure broadcast encryption scheme, only the legitimate receivers which belong to the 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 paid 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 the introduction of broadcast encryption in 1993 by Fiat and Naor [1], lots of 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. 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 which among them 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, the 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. Also, 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 protect receivers’ identities but number of receivers is leaked by the ciphertext length. Also, 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 which means that any user, whether he is in the set $S$ or not, is unable to obtain information about intended receivers. Outsider anonymity [18] is another definition which only guarantees the anonymity of intended receivers from the view of users outside of the set $S$. However, users in $S$ can still learn the identities of other legal receivers. Fazio et al. in [18] proposed an outsider-anonymous broadcast encryption scheme with sublinear overheads. The attribute-based anonymous multicast scheme presented by Yu, Ren and Lou [3], which we call it the YRL scheme in this paper, suggests stronger definition of full anonymity; the scheme not only hides the identities of receivers, but also it protects the number of intended users. Also, communication and computation overheads are linear in the number of attributes and independent of the number of receivers; so the scheme provides a high efficiency thank to its attribute-based structure. The scheme relies on the notion of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and aims solving the group key distribution problem. So, instead of broadcasting a message $M$, a group key $GK$ is emitted.

**Our contributions.** In this paper, we have 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 computation and communication overheads are linear in the number of attributes and independent of the number of receivers.

Boneh et al. have stated in [6] that “it is a long-standing open problem to build a low-overhead anonymous broadcast encryption system”; so 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. In Section 2, we give background on bilinear groups and state the access policy used in the YRL scheme. In Section 3, we present 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, and 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 $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.

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 $G$ be a group generator algorithm. It takes as input a security parameter, $\lambda$, and outputs a tuple, $(N = p_1 p_2 p_3, G, G_T, e)$, where $p_1, p_2, p_3$ are distinct prime numbers, $G$ and $G_T$ are multiplicative cyclic groups of composite order $N = p_1 p_2 p_3$ 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 $g_{p_i}$. Each $T \in G$ can be represented as $T = X_1 X_2 X_3$ where $X_i \in G_{p_i}$ is referred to as the "$G_{p_i}$ component of $G$". Also, 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 $r = g_{p_i}^\alpha$ where $\alpha$ is a random element in $\mathbb{Z}_{p_i}$.

The main property of composite order bilinear maps is that the 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

Here we review the access policy used in the YRL scheme to specify the intended group of receivers [3]. Let \( n \) denote the total number of attributes. Each user is assigned an \( n \)-element string \( \{ \text{Att}_{i,b} | \forall i \in \mathbb{Z}_n, b = 0 \text{ or } 1 \} \) such that \( \text{Att}_{i,0} \) and \( \text{Att}_{i,1} \) show the negative and positive incident of \( i \)-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, \( (\text{Att}_{3,1} \land \text{Att}_{1,0}) \) or \( X_3X_1 \) is used for showing the access policy for the users which have 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, the 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. In the following, we will review the scheme.

**Setup** \((\lambda, X_{n-1}X_{n-2}...X_0) \rightarrow (PP, MK)\): The input parameters of the setup algorithm are the security parameter \( \lambda \) and the attribute set of each user \( X_{n-1}X_{n-2}...X_0 \), and the outputs are public parameters \( PP \) and the Master Key \( MK \) which are generated as follows. Due to the security parameter \( \lambda \), the algorithm chooses a group \( G \) of prime order \( p \) with generator \( g \). Each attribute in the vector \( X_{n-1}X_{n-2}...X_0 \) is mapped to one of the members of \( G \). Consider the attribute \( \text{Att}_{i,b} \) is mapped to \( h_{i,b} \) as a member of \( G \). The Setup algorithm randomly selects \((a_i, b_i) \in \mathbb{Z}_p \) 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 the Master Key \( MK = (\alpha, \beta, \{a_i, b_i\}_{\forall i \in \mathbb{Z}_n}) \), where \( \alpha, \beta \in RZ_N \). \( MK \) 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 are the master key \( MK \) and the attribute set of each user, \( X_{n-1}X_{n-2}...X_0 \), and its output is the tuple \( (D_i, \hat{D}, \hat{D}^r, \{D_i\}_{\forall i \in \mathbb{Z}_n}) \) as the secret key of the user which is generated through Relation 1.

\[
SK = (D = g^{(\alpha+r)/\beta}, \hat{D} = g^r, \hat{D}^r = g^{3r}, \{D_i = h_{i,\hat{X}_i}^r\}_{\forall i \in \mathbb{Z}_n})
\]  

(1)

The parameters \((g, \alpha, \beta, \hat{h}_{i,\hat{X}_i})\) are defined in the setup algorithm, above. Also, \( r \) is a random element chosen from \( \mathbb{Z}_p \).

**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 are \( GK \), the access policy \( T \), and the master key \( MK = (\alpha, \beta, \{a_i, b_i\}_{\forall i \in \mathbb{Z}_n}) \) and the output is the ciphertext \( CT = (\hat{C}, \hat{C}, \{\hat{C}_j\}_{j=0,1}, \{C_i\}_{\forall i \in \mathbb{Z}_n}) \). The first component of \( CT \) is of the form \( \hat{C} = (GK \parallel MAC)X \), 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, \( s_0, s_1, ..., s_{n-1}, k_0, k_1 \in R \) are chosen and \( \delta \) is set equal to \( \sum_{i=0}^{n-1} \gamma_i s_i \); where \( \gamma_i \) 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 \( t_i \in R \) 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 \( g^\psi = \prod_{i=0}^{n-1} C_{i,0}C_{i,1} = g^{\delta+x} \) where \( \delta \) has been defined in step 1 and \( x \in Z_p \) satisfies the equation \( g^\psi = \prod_{\forall j \in X_i} h_{j,1}^{x_j} \). Then the second term of the ciphertext is calculated as \( \tilde{C} = g^{\beta \psi} \) and the values of \( C_{i,0} \) and \( C_{i,1} \) are updated as follows:

\[
C_{i,0} = g^{k_0}C_{i,0}, C_{i,1} = g^{k_1}C_{i,1}
\]  

(2)

4) Finally, the ciphertext is generated through Relation 3.

\[
CT = (\tilde{C} = (GK || MAC)e(g, g)^{\alpha s_i}, \tilde{C} = g^{\beta \psi}, \{\tilde{C}_j = g^{k_j/\beta}\}_{j=0,1}, \{C_i\}_{\forall i \in Z_n})
\]

(3)

Where \( MAC = H(GK) \) (\( H(.) \) is a cryptographic hash function).

**Decryption** \((CT, X_{n-1}X_{n-2}...X_0, SK) \rightarrow (GK or \perp)\): Each authorized group member, GM, runs this algorithm to obtain the group key \( GK \). The inputs of this algorithm are 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 \( j = 0, 1 \), \( B_j = e(\tilde{C}_j, \tilde{D}) = e(g^{k_j/\beta}, g^{\beta \psi}) = e(g, g)^{r k_j} \) 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 Equation 4.

\[
F_i = e(D_i, g^{s_i})e(C_{i,X_i}, \tilde{D})/B_{X_i}
\]

\[
= e(h_{i,X_i}^{t_i}, g^{s_i})e(g^{k_i}h_{i,X_i}^{s_i+t_i}, g^\gamma)/B_{X_i}
\]

\[
= e(g, g)^{r \gamma s_i}e(g, h_{i,X_i})^{r t_i}
\]

(4)

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

2) \( F \) is computed by multiplying the values of \( F_i \):

\[
F = \prod_{i=0}^{n-1} F_i = \prod_{i=0}^{n-1} e(g, g)^{r \gamma s_i}e(g, h_{i,X_i})^{r t_i} = e(g, g)^{r d}e(g, g)^{r x'}
\]

(5)

where \( x' \in Z_p \) is defined in Equation 6:

\[
ge^{x'} = \prod_{i=0}^{n-1} h_{i,X_i}^{t_i}, (t_i = 0 \iff X_i \notin T)
\]

(6)

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

\[
\prod_{i=0}^{n-1} e(g, h_{i,X_i})^{r t_i} = e(g, g)^{r x'}
\]

(7)

Equation 7 is used in calculating the value of \( F \) in (5).
If the GM’s attributes satisfies 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 \( g^x = \prod_{j \in T} h_{j,x_j} \)).

3) Finally, each group member GM computes \( M' \) and recovers the group key \( GK \) using (8).

\[
M' = \frac{\bar{C}}{e(\bar{C}, D)/F} = \frac{\langle GK \| MAC \rangle e(g, g)^{x' x}}{e(g, g)^{x' x} / e(g, g)^r (x' - x)} = \langle GK \| MAC \rangle e(g, g)^r (x' - x)
\]

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 \( x \), and 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, the 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. Also, he is not aware of the membership of other users in the subset or even the number of authorized users. So, 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 claim that their introduced scheme is 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

Here, 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 and all of the users, including authorized and unauthorized ones, can decrypt the received message. Assume a 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:

\[
SK_u = \langle D = g^{(x+r)/\beta}, \bar{D} = g^r, \{D_i = h_{i,x_i} \} \rangle \quad (9)
\]

\[
CT = \langle \bar{C} = \langle GK \| MAC \rangle, \{\bar{C}_j = g^{k_j/\beta} \} \rangle \quad (10)
\]

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

1) Computes \( e(D, \bar{C}) = e(g^{(x+r)/\beta}, g^{r s'}) = e(g, g)^{t(x+r)s'} \).

2) Calculates \( e(g, g)^{t x'} \):

As mentioned in the Decryption algorithm, \( s' \) satisfies the equation, \( g^{s'} = \prod_{i=0}^{n-1} C_{i,0} C_{i,1} \); but after that \( C_{i,0} \) and \( C_{i,1} \) were updated to new values, \( C_{i,0} = g^{h_{i,0}} C_{i,0}, C_{i,1} = g^{h_{i,1}} C_{i,1} \). Since \( C_{i} = (g^{x_i}, C_{i,0}, C_{i,1}) \), user \( u \) can compute \( \prod_{i=0}^{n-1} C_{i,0} C_{i,1} \) by using \( C_i \):

\[
C_{i,0} = g^{h_{i,0}} C_{i,0}, C_{i,1} = g^{h_{i,1}} C_{i,1}
\]

\[
\prod_{i=0}^{n-1} C_{i,0} C_{i,1} = \prod_{i=0}^{n-1} g^{h_{i,0}} C_{i,0} \cdot g^{h_{i,1}} C_{i,1} = g^{n(k_0 + k_1)} g^{s'}
\]

September 26, 2017 DRAFT
So,

\[ e(g^{n(k_0+k_1)}g^{s'}, \hat{D}) = e(g^{n(k_0+k_1)}g^{s'}, g^\gamma) = e(g^{s'}, g^\gamma)e(g^{n(k_0+k_1)}, g^\gamma) = e(g,g)^{rs'}e(g,g)^{rn(k_0+k_1)} \]  

(12)

Furthermore, we have:

\[ B_j = e(C_j, \hat{D}) = e(g^{k_j/2}, g^{\beta}) = e(g,g)^{rk_j}, j = 0, 1 \rightarrow B_0, B_1 = e(g,g)^{rk_0+k_1} \]  

(13)

Thus, using Equations 12 and 13, user \( u \) can obtain \( e(g,g)^{rs'} \) as follows:

\[ \frac{(12)}{(13)^n} = \frac{e(g,g)^{rs'}e(g,g)^{rn(k_0+k_1)}}{(e(g,g)^{c(k_0+k_1)})^n} = e(g,g)^{rs'} \]  

(14)

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

\[ \frac{e(g,g)^{(\alpha + \gamma)s'}}{e(g,g)^{rs'}} = e(g,g)^{\alpha s'} \]  

(15)

4) Finally, user \( u \) can obtain \( GK \) as follows:

\[ \frac{\hat{C}}{e(g,g)^{\alpha s'}} = \frac{(GK \parallel MAC)e(g,g)^{\alpha s'}}{e(g,g)^{\alpha s'}} = (GK \parallel MAC) \]  

(16)

So this user, regardless of what his attributes set 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 Improving-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 \( k_0, k_1 \) for updating \( C_{i,0} \) and \( C_{i,1} \) to new values \( C_{i,0} = g^{k_0}C_{i,0}, C_{i,1} = g^{k_1}C_{i,1} \). So \( e(\prod_{i=0}^{n-1} C_{i,0}C_{i,1}, \hat{D}) \) has a fixed term \( e(g,g) \) which can be omitted using the term \( \hat{C} \) in the ciphertext. As a result, \( e(g,g)^{rs'} \) and \( e(g,g)^{\alpha s'} \) are obtained which helps the attacker to obtain \( GK \).

Hence, in order to fix this weakness, we randomize the update process and eliminate the third term in both ciphertext and secret key. Also, in order to propose a security proof, the Improved-YRL scheme is based on composite order bilinear maps. In what follows, we describe the Improved-YRL scheme in detail:

Setup \((\lambda, X_{n-1}X_{n-2}...X_0) \rightarrow (PP, MK)\): The input parameters of the setup algorithm are the security parameter \( \lambda \) and the attribute set of each user \( X_{n-1}X_{n-2}...X_0 \), and the outputs are public parameters \( PP \) and the Master Key \( MK \) which are generated as follows. Due to the security parameter \( \lambda \), 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 the 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 form \( Z_{p_1} \) and \( \gamma_i = a_i + b_i \). The only difference here is that \( h_{i,0} \) and \( h_{i,1} \) have an additional factor \( R_{3,0} \) and \( R_{3,1} \) which are randomly chosen from \( G_{p_2} \). Hence \( h_{i,b} \) is an element of the subgroup \( G_{p_1p_3} \). The algorithm outputs \( PP = (N = p_1p_2p_3, G) \) as the public parameters and the 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, \(X_{n-1}X_{n-2}...X_0\), 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 Relation 17.

\[
SK = (D = g_1^{(\alpha+r)/\beta}, \hat{D} = g_1^r, \{D_i = h_{i,X_i}^*\}_{\forall i \in Z_n})
\]  

(17)

The parameters \((g_1, \alpha, \beta, h_{i,X_i}^*)\) are defined in the setup algorithm above. So it is the same as the YRL’s KeyGen algorithm except that the third term \(\hat{D} = g_1^\beta r\) is omitted and \(D\)’s are members of \(G_{p_1,p_3}\).

Encryption \((GK, T, MK) \rightarrow CT\): As before, The inputs are the group key \(GK\), the access policy \(T\), and the master key \(MK\) and the output is the ciphertext \(CT\). But, this algorithm has some differences with the YRL’s Encryption algorithm; The first difference is the procedure of updating the values of \(C_{i,0}\) and \(C_{i,1}\) and the second one is omitting the term \({\hat{C}_j = g_{k_j^*}^{(\alpha+1)/\beta}}\) \(\forall j \in \mathbb{N}_0\) from the ciphertext because the decryption successfully works without it. Also \(g_1\) is turned into \(g_1\), so the first term in \(C_i\) is \(g_1^{\alpha_i} \in G_{p_1}\) and because \(h_{i,X_i}\) is an element of \(G_{p_1,p_3}\) as stated in the setup algorithm, \(C_{i,0}\) and \(C_{i,1}\) are elements of \(G_{p_1,p_3}\), too.

So the ciphertext \(CT = (\hat{C}, \check{C}, \{C_i\}_{\forall i \in Z_n})\) is generated as below:

1) The values, \(s_0, s_1, s_{n-1} \in R\) \(Z_{p_1}\), are randomly selected and \(\delta\) 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 \(G_{p_1,p_3}\). 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 \(g_1^{s_i}\) is computed through Equation 18:

\[
\prod_{i=0}^{n-1} C_{i,0} C_{i,1} = g_1^{s_i} R_3 = g_1^{\delta+x} R_3
\]

(18)

Where \(R_3\) is the product of all elements in \(G_{p_3}\). Also, \(x \in Z_{p_1}\) is such that \(g_1^x = \prod_{j \in X_j \in T} h_{j}^{s_i} X_j\), where only \(G_{p_1}\) part of \(h_{j}^{s_i} X_j\) is considered. Then the value of \(\hat{C}\) is set equal to \((g_1^{s_i} R_3)^{\beta} = g_1^{\beta s_i} R_3 \in G_{p_1,p_3}\). In other words, \(\hat{C}\) has an extra \(R_3^\beta\) 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 \(C_{i,X_i} = C_{i,0} C_{i,1} = h_{i,X_i}^{s_i} k_i C_{i,1-X_i}\). In this way, only \(C_{i,1-X_i}\), which is not intended in the access policy is updated and \(C_{i,X_i}\), 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 \(T = \{X_3, X_2, X_0\}\), 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 \(CT = (\check{C} = (GK \Downarrow MAC) e(g_1, g_1)^{\alpha s_i}, \hat{C} = g_1^{\beta s_i} R_3, \{C_i\}_{\forall i \in Z_n})\), where \(e(g_1, g_1)^{\alpha s_i}\) is computed as \(e(\prod_{i=0}^{n-1} C_{i,0} C_{i,1}, g_1^\beta) = e(g_1^{s_i} R_3, g_1^\beta) = e(g_1, g_1)^{\alpha s_i}\). The last equality holds due to the orthogonality property of composite order bilinear maps.

Decryption \((CT, X_{n-1}X_{n-2}...X_0, SK) \rightarrow (GKor \perp)\): The inputs of this algorithm are the ciphertext, the attribute set \(X_{n-1}X_{n-2}...X_0\), and the private key of a GM and 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. Also, $F_i$s 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 Equation 19:

$$F_i = e(D_i, g_1^{s_i})e(C_i, X_i, \hat{D})$$

$$= e(h_i^{r}, g_1^{s_i})e(g_1^{k_1s_i+t_i}, g_1^{r_i})$$

$$= e(g_1, g_1)^{r\gamma_i}s_i e(g_1, h_i^{s_i}, X_i)^{rt_i}e(g_1, g_1)^{k_1r} \tag{19}$$

It can be easily verified that the elements of $G_{p_i}$ are omitted due to the orthogonality property of composite order bilinear maps. Also, values of the 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 $F_i$s obtained in the previous step:

$$F = \prod_{i=0}^{n-1} F_i = \prod_{i=0}^{n-1} e(g_1, g_1)^{r\gamma_i}s_i e(g_1, h_i^{s_i}, X_i)^{rt_i}e(g_1, g_1)^{k_1r}$$

$$= e(g_1, g_1)^{r\delta_t}s_i x'' e(g_1, g_1)^{r\bar{t}}e(g_1, g_1)^{k_1r} \tag{20}$$

where $x'$ satisfies the Equation 21 where only $G_{p_i}$ part of $h_j, X_j$ is considered:

$$g_{1}^{x'} = \prod_{i=0}^{n-1} h_i^{s_i}(t_i = 0 \text{ if } X_i \notin T) \tag{21}$$

If the GM’s attributes satisfies 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 Equation 22 to obtain $GK$:

$$M' = \frac{C}{e(C, D)/F} = \frac{e(g_1, g_1)^{s_i}x''}{e(g_1, g_1)^{s_i + r\gamma_i}} = (GK \parallel MAC)e(g_1, g_1)^{r(k + k' - x)} \tag{22}$$

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 security of the proposed scheme and prove that it achieves both indistinguishability and anonymity in the standard model.

### 6 Security Analysis

We begin 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.** The Improved-YRL scheme is secure against the proposed attack in Section 4.

**Proof.** The attack process involves computing the equation \( \prod_{i=0}^{n-1} C_{i,0} C_{i,1} = g_1^{k_r} g_1^{r_i} R_3 \), where \( k_r \) is sum of all \( k_i \) values that is 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

Here we define a model for 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.** The challenger \( C \) runs Setup algorithm to generate the master key \( MK \).

**Phase1.** Adversary \( A \) issues queries for secret keys corresponding to the set of attributes \( Att_1, Att_2, \ldots, Att_{q'} \). Challenger \( C \) runs KeyGen algorithm and returns the corresponding secret keys \( SK_1, SK_2, \ldots, SK_{q'} \) to \( A \). \( A \) can also make decryption queries \( (CT, Att_i) \), which means decryption of the ciphertext \( CT \) for user \( i \) with attribute set \( Att_i \), and the challenger \( C \) will return the decrypted message or \( \bot \) using Decryption algorithm.

**Challenge.** \( A \) 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 \( A \).

**Phase2.** \( A \) continues querying secret keys corresponding to the set of attributes \( Att_{q'+1}, Att_{q'+2}, \ldots, Att_q \) which none of them satisfies \( T_0^* \) or \( T_1^* \) and receives corresponding secret keys \( SK_{q'+1}, SK_{q'+2}, \ldots, SK_q \). \( A \) also 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.** \( A \) outputs \( b' \) as its guess for \( b \) and wins the game if \( b = b' \). The advantage of \( A \) in this game is defined as \( \text{Adv}_{A,BE}^{ANO-IND-CCA}(\lambda) = |\Pr[b = b'] - 1/2| \).

**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 security of the Improved-YRL scheme. Rao et al. in [22] closely followed [23] to show that Assumptions 1, 2, 3 hold in the generic group model under the assumption that finding a non-trivial factor of \( N \), where \( N = p_1 p_2 p_3 \), is hard.

**Assumption 1.** Let \( G \) be a group generator and \( \hat{g} = (N = p_1 p_2 p_3, G, G_T, e) \leftarrow G(\lambda) \). Choose \( g_1, g_3 \) randomly from
$G_{p_1}$ and $G_{p_2}$ respectively. Then for each PPT adversary $\mathcal{A}$ which is given $D = (\bar{y}, g_1, g_3)$, $\mathcal{A}$’s advantage to distinguish $T_0 \in G$ from $T_1 \in G_{p_1,p_2}$, 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:

$$Adv^1_{\mathcal{A}} = |Pr[\mathcal{A}(D,T_0) = 1] - Pr[\mathcal{A}(D,T_1) = 1]| \leq \text{negl}(\lambda)$$

Where negl(.) is a negligible function.

**Assumption 2.** Let $\mathcal{G}$ be a group generator and $\bar{y} = (N = p_1p_2p_3, G, G_T, e) \leftarrow \mathcal{G}(\lambda)$. Choose random elements $g_1 \in G_{p_1}, g_3 \in G_{p_2}, X_1, X_2, X_3 \in G, Y_1, Y_2 \in G_{p_1,p_2}$. Then for each PPT adversary $\mathcal{A}$ which is given $D = (\bar{y}, g_1, g_3, X_1, X_2, X_3, Y_1Y_2)$, $\mathcal{A}$’s advantage to distinguish $T_0 \in G_{p_1}$ from $T_1 \in G_{p_1,p_2}$, 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:

$$Adv^2_{\mathcal{A}} = |Pr[\mathcal{A}(D,T_0) = 1] - Pr[\mathcal{A}(D,T_1) = 1]| \leq \text{negl}(\lambda)$$

Where negl(.) is a negligible function.

**Assumption 3.** Let $\mathcal{G}$ be a group generator and $\bar{y} = (N = p_1p_2p_3, G, G_T, e) \leftarrow \mathcal{G}(\lambda)$. 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 $D = (\bar{y}, g_1, X_2, X_3, g_1^\varsigma Y_2Y_3)$, $\mathcal{A}$’s advantage to distinguish $T_0 = e(g_1, g_1)^\alpha$ from $T_1$ which is a random element in $G_T$, is negligible. In other words, for any PPT algorithm $\mathcal{A}$, we have:

$$Adv^3_{\mathcal{A}} = |Pr[\mathcal{A}(D,T_0) = 1] - Pr[\mathcal{A}(D,T_1) = 1]| \leq \text{negl}(\lambda)$$

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, but they are only used in the security proof. A normal secret key, can decrypt both normal and semi-functional ciphertexts, but a semi-functional secret key can only decrypt normal ciphertexts. In other words, one would fail to decrypt a semi-functional ciphertext using a normal secret key.

**Semi-functional ciphertext.** To obtain a semi-functional ciphertext, we first run Encryption algorithm to obtain a normal ciphertext $CT = (\tilde{C}, \tilde{C'}, \{C_i\}_{\forall i \in \mathbb{Z}_n})$, where $C_i = (g^{a_i}, C_{i,0}, C_{i,1})$ for all $i \in \mathbb{Z}_n$. Then the semi-functional ciphertext $CT' = (\tilde{C'}, \tilde{C'}, \{C_i'\}_{\forall i \in \mathbb{Z}_n})$ is computed as below

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

where $\delta$ is chosen randomly from $\mathbb{Z}_N$ and in $C_i' = C_i \times g_2^{\delta/n}$ only the terms $C_{i,0}, C_{i,1}$ are multiplied by $g_2^{\delta/n}$. Also, $n$ is the number of attributes and $\beta$ 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 \( SK = (D, \hat{D}, \{D_i\}_{\forall i \in \mathbb{Z}_n}) \), where \( D_i = h_i^{x_i} \) for all \( i \in \mathbb{Z}_n \). The semi-functional secret key \( SK' = (D', \hat{D'}, \{D'_i\}_{\forall i \in \mathbb{Z}_n}) \) is computed as below

\[
D' = D \times g_2^\gamma, \quad \hat{D}' = \hat{D} \times g_2^\gamma, \quad D'_i = D_i \times g_2^{e_i} \tag{27}
\]

Where \( \gamma \) is a random element in \( \mathbb{Z}_N \) and \( e_i = a_i \) if \( h_i^{x_i} = g_1^{a_i} X_3 \) and \( e_i = b_i \) if \( h_i^{x_i} = g_1^{b_i} X_3 \).

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 are as follows:

- \( Game_{ANO-IND-CCA} \): In this game which was described in Section 6.1, all ciphertexts and secret keys are normal.
- \( Game_0 \): Here the challenge ciphertext is semi-functional, but 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. So in \( Game_k \), all the secret keys would be semi-functional.
- \( Game_{final} \): This game is the same as \( Game_q \) except that the ciphertext is randomized. So 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:

\[
Game_{ANO-IND-CCA} \Leftrightarrow Game_0 \Leftrightarrow Game_1 \Leftrightarrow \ldots \Leftrightarrow Game_{q-1} \Leftrightarrow Game_q \Leftrightarrow Game_{final}.
\]

Where the notation “ \( \Leftrightarrow \) “ means that the two games are computationally indistinguishable. Now, we will show the above relations through the following lemmas.

**Lemma 1.** Suppose that there exists a polynomial time algorithm \( \mathcal{A} \) such that \( Adv^ {Game_{ANO-IND-CCA}}_\mathcal{A} - Adv^ {Game_0}_\mathcal{A} = \varepsilon \). Then we can build a PPT algorithm \( \mathcal{B} \) with advantage \( \varepsilon \) in breaking Assumption 1.

**Proof.** \( \mathcal{B} \) is given \((\hat{g}, g_1, g_3, T)\). 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 \( \alpha, \beta \in \mathbb{Z}_N \), and for each element of the attribute set, it chooses \( a_i, b_i \) randomly from \( \mathbb{Z}_{p_i} \) and keeps the master key \( MK = (\alpha, \beta, \{a_i, b_i\}_{\forall i \in \mathbb{Z}_n}) \).

**Phase 1 and Phase 2.** \( \mathcal{B} \) generates normal secret keys \( SK = (D, \hat{D}, \{D_i\}_{\forall i \in \mathbb{Z}_n}) \) in response to \( \mathcal{A} \)’s secret key queries using the KeyGen algorithm. This is possible since \( \mathcal{B} \) possess the master key \( MK \) and \( g_1, g_3 \). As we mentioned in the ANO-IND-CCA security game, non of the attribute sets queried in secret key requests, should satisfy the access policies given by \( \mathcal{A} \) in the challenge phase. Also, 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 \( CT_b = (\hat{C}, \hat{C}, \{C_i\}_{\forall i \in \mathbb{Z}_n}) \). Next, in order to generate the challenge ciphertext \( CT^* = (\hat{C}', \hat{C}', \{C'_i\}_{\forall i \in \mathbb{Z}_n}) \), \( \mathcal{B} \) performs the following calculations:

\[
\hat{C}' = \hat{C} \times T \quad \hat{C}' = \hat{C} \times T^b \\
C'_{i_0,1} = C'_{i_0,1} \times T^{1/n} \tag{28}
\]

**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 $Game_0$. So, if $\epsilon$ is a non-negligible function, $B$ can use the output of $A$ to distinguish between two values of $T$ and break Assumption 1.

Lemma 2. Suppose that there exists a polynomial time algorithm $A$ such that $Adv^r_A^\text{Game}_{k-1} - Adv^r_A^\text{Game}_k = \epsilon$, where $1 \leq k \leq q$ and $q$ is the maximum secret key queries that $A$ can make. Then we can build a PPT algorithm $B$ that has advantage $\epsilon$ in breaking Assumption 2.

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

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

Phase1 and Phase2. Here $B$ responds to secret keys queries from $A$ in three ways depending on the query number. For the first $k - 1$ queries, $B$ generates semi-functional secret keys. For this purpose, $B$ first computes a normal secret key $(D, \tilde{D}, \{D_i\}_{\forall i \in Z_n})$ using the KeyGen algorithm. Then $SK_j = (D', \tilde{D}', \{D_i'\}_{\forall i \in Z_n})$ for $1 \leq j \leq k - 1$ is computed as below:

$$r_j \in \mathbb{R}Z_N$$
$$D' = D \times (Y_1Y_2)^{r_j/\beta}$$
$$\tilde{D}' = \tilde{D} \times (Y_1Y_2)^{r_j}$$
$$D_i' = D_i \times (Y_1Y_2)^{r_j e_i}$$

(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 the query $k$, $B$ first generates normal secret key $(D, \tilde{D}, \{D_i\}_{\forall i \in Z_n})$ and sets $SK_k$ as below:

$$D' = D \times T^{1/\beta}$$
$$\tilde{D}' = \tilde{D} \times T$$
$$D_i' = D_i \times T^{e_i}$$

(30)

Where $e_i$ is as defined above. It can be seen that if $T \in G_{p_1}$, the $SK_k$ is a normal secret key and if $T \in G_{p_1p_2}$, $SK_k$ is a semi-functional secret key.

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

Also, in response to $A$’s decryption requests $(CT, \text{Att}_i)$, $B$ generates the normal secret key corresponding to $\text{Att}_i$, and decrypts $CT$ using Decryption algorithm.

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

$$\tilde{C}' = \tilde{C} \times (g_1^{a_i}(X_1X_2X_3))$$
$$\tilde{C}' = \tilde{C} \times (X_1X_2X_3)^\beta$$
$$C'_{i_0,1} = C'_{i_0,1} \times (X_1X_2X_3)^{1/n}$$

(31)
CT* is sent back to A.

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

**Lemma 3.** Suppose that there exists a polynomial time algorithm A such that $Adv_A^{\text{Game}_q} - Adv_A^{\text{Game}_{\text{final}}} = \varepsilon$. Then we can build a PPT algorithm B that has advantage $\varepsilon$ in breaking Assumption 3.

**Proof.** B is given $(\tilde{g}, g_1, g_1^a X_2, X_3, g_1^2 Y_2 Y_3, T)$. It will simulate $\text{Game}_q$ or $\text{Game}_{\text{final}}$ depending on whether $T = e(g_1, g_1)^{\alpha \varsigma}$ or it is a random element of $G_T$. We now describe how B interacts with A to break Assumption 3.

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

**Phase1 and Phase2.** Here all the queried secret keys are semi-functional. In response to A’s secret key queries, B first generates normal secret key $(D, \tilde{D}, \{D_i\}_{i \in Z_n})$ using the KeyGen algorithm, $g_1, X_3, g_1^a X_2$ and MK. Then semi-functional secret key $SK_i = (D', \tilde{D}', \{D_i'\}_{i \in Z_n})$ for $1 \leq j \leq q$ is computed as below:

\[
\begin{align*}
    r_j & \in R_Z N \\
    D' &= D \times (g_1^a X_2)^{r_j / \beta} \\
    \tilde{D}' &= \tilde{D} \times (g_1^a X_2)^{r_j} \\
    D_i' &= D_i \times (g_1^a X_2)^{r_j e_i},
\end{align*}
\]

Where $e_i$ is as defined in Lemma 2.

Also, in response to A’s decryption requests $(CT, \text{Att}_i)$, B generates the normal secret key corresponding to $\text{Att}_i$, and decrypts CT using the Decryption algorithm.

**Challenge.** A sends B two equal length group keys and two access policies $T_i$ and $T_i'$. B chooses a random bit b and performs the normal encryption of $GK_b$ under $T_b^*$ to obtain $CT_b = (\tilde{C}, \tilde{C}', \{C_i\}_{i \in Z_n})$. Next, the challenge ciphertext $CT^* = (\tilde{C}', \tilde{C}', \{C_i\}_{i \in Z_n})$ is generated as below:

\[
\begin{align*}
    \tilde{C}' &= \tilde{C} \times T \\
    \tilde{C}' &= \tilde{C} \times (g_1^2 Y_2 Y_3)^{\beta} \\
    C_{i_{0,1}}' &= C_{i_{0,1}} \times (g_1^2 Y_2 Y_3)^{1/n},
\end{align*}
\]

$CT^*$ is sent back to A.

**Guess.** A outputs a bit b' as its guess of b and wins if b = b'. It can be seen that if $T = e(g_1, g_1)^{\alpha \varsigma}$, B has interacted with A in Game$_q$. Else if $T \in R G_T$, the challenge ciphertext is randomized and $\text{Game}_{\text{final}}$ is simulated. So B can use the output of 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 previous lemmas that the Game\textsubscript{ANO−IND−CCA} is indistinguishable from Game\textsubscript{final}. In Game\textsubscript{final}, the adversary receives no information about \(b\) information theoretically and the chance of any adversary in guessing the true \(b\) is exactly \(1/2\). So this is true in Game\textsubscript{ANO−IND−CCA} and the adversary can not guess which \(GK\) is encrypted and also can not obtain any information about access structure from the ciphertext with a probability greater than \(1/2\). So the Improved-YRL scheme has both indistinguishability and anonymity and the proof is completed.

\(\square\)

7 Performance Evaluation

7.1 Overhead Analysis

This subsection analyzes the performance of the proposed scheme. For this aim, we calculate the computation, communication and storage overheads in terms of the total number of attributes in the network which is denoted by \(n\). We denote modular multiplications, exponentiations and pairings over composite orders by \(\text{Mul.cmp}\), \(\text{Exp.cmp}\) and \(\text{Pair.cmp}\) respectively.

**Computation Overhead.** Here 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 \(2n\text{Exp.cmp}\). Also, as in the Improved-YRL’s setup, some elements of subgroup \(G_1\) are multiplied with some elements of subgroup \(G_3\), the extra term \(2n\text{Mul.cmp}\) is added. So the total overhead of the Improved-YRL’s setup is \(2n\text{Exp.cmp} + 2n\text{Mul.cmp}\). 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 \((n + 2)\text{Exp.cmp}\). In the Encryption algorithm, the terms \(g^{k_0\beta}\) and \(g^{k_1\beta}\) are eliminated. Therefore, in comparison with the Encryption algorithm of the YRL scheme, two \(\text{Exp}\) computations are omitted and the resulting computation overhead is reduced to \((3n + 1)\text{Exp.cmp}\). In the Decryption algorithm, there is no need to compute \(\{B_j = e(g^{\frac{g^{k_j\beta}}{g^{\beta r}}}, g^{\beta r})\}_{j \in \{0, 1\}}\) and consequently, the total number of pairing computations is reduced by two units. Therefore, the total computation overhead of Decryption algorithm becomes \(n\text{Mul.cmp} + (n + 2)\text{Pair.cmp}\). These results are summarized in Table 1.

**Communication Overhead.** The total communication overhead of the proposed scheme is \((n+1) \log_2 p_1 + 2n \log_2(p_1p_3) + \log_2 |G_T^{\text{comp}}|\) as presented in Table 1.

**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], and as in the Improve-YRL scheme, \(g^{\beta r}\) is omitted from the secret key, each user needs \(2 \log_2 p_1 + n \log_2(p_1p_3)\) bits to store his secret key. These results are illustrated in Table 1.

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], which were discussed in the Introduction Section. In this table, the parameters \(n\) and \(N\) denote number of the attributes and number of the intended receivers, respectively. As can be seen 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 against the proposed attack in this paper, but also it preserves low overhead and high performance property of the basic YRL scheme.

Furthermore, comparison between our scheme and two selected anonymous broadcast encryption schemes [16] and [17], demonstrate 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 number of the intended receivers. In this case, if we use 1024 bit RSA algorithm for encryption, then the ciphertext size would be $1024N$ where $N$ is the total number of receivers. This implies to a huge computation and communication overhead in the large-scale systems. Since the attributes are usually shared by unlimited number of group members, therefore 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 required attributes in compare to the total number of users could be significantly small. Also in addition to better efficiency, our proposed scheme suggests stronger anonymity; that is, our scheme not only hides the identities of receivers, but also it 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 paper, 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, can decrypt the received message. Thus, the YRL scheme does not provide the main requirement of the broadcast encryption schemes.

Since introducing an anonymous, efficient and provably secure broadcast encryption scheme is one of the most important open problems in this field, we improved the YRL scheme in composite order bilinear groups and made 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 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, for example secure and scalable e-health architectures and secure cloud storage systems, can be considered as an interesting future work.

References

17


### TABLE 1
The summary of overhead analysis for our proposed scheme.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Improved-YRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation Overhead</td>
<td></td>
</tr>
<tr>
<td>Setup</td>
<td>$2n \exp.cmp + 2n \mul.cmp$</td>
</tr>
<tr>
<td>KeyGen</td>
<td>$(n + 2) \exp.cmp$</td>
</tr>
<tr>
<td>Encryption</td>
<td>$(3n + 1) \exp.cmp$</td>
</tr>
<tr>
<td>Decryption</td>
<td>$n \mul.cmp + (n + 2) \pair.cmp$</td>
</tr>
<tr>
<td>Communication Overhead</td>
<td>$(n + 1) \log_2 p_1 + 2n \log_2 (p_1 p_3) + \log_2 (</td>
</tr>
<tr>
<td>Storage Overhead</td>
<td>$2 \log_2 p_1 + n \log_2 (p_1 p_3)$</td>
</tr>
</tbody>
</table>

### TABLE 2
Comparison among anonymous broadcast encryption schemes. In this table, $n$ denotes the number of attributes and $N$ denotes the number of the intended receivers.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Improved-YRL</th>
<th>YRL [3]</th>
<th>[16]</th>
<th>[17]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption time</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(N)$</td>
<td>$O(N)$</td>
</tr>
<tr>
<td>Decryption time</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(N)$</td>
<td>$O(N)$</td>
</tr>
<tr>
<td>Key size</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(1)$</td>
<td>$O(N)$</td>
</tr>
<tr>
<td>Communication Overhead</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(N)$</td>
<td>$O(N)$</td>
</tr>
<tr>
<td>Storage Overhead</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(1)$</td>
<td>$O(N)$</td>
</tr>
</tbody>
</table>

Reyhaneh Rabaninejad is currently a third year Ph.D student in 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 B.S. degree in Electrical Engineering from Shahid Bahonar University, Kerman, Iran, in 2013 with the honor of first ranked student among electrical engineering students of the same entrance, and his M.S. 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 B.Sc. degree from the Amirkabir University of Technology, Tehran, Iran, in 2006, the M.Sc. degree from Islamic Azad University, South Tehran Branch, Tehran, Iran, in 2009 and the Ph.D. 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 adjunct assistant professor in 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 90 research articles in refereed journals/conferences and is one the founding members of Iranian Society of Cryptology.