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

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

Abstract

In broadcast encryption schemes, a distribution center broadcasts an encrypted message to a subset S 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 has the same sensitivity as the message itself. YRL, is an anonymous attribute-based broadcast encryption scheme with linear computation, communication and storage overheads in the number of attributes. In this paper, we first propose an attack on the YRL scheme and show that unfortunately 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

Broadcast Encryption, Ciphertext-Policy Attribute-Based Encryption (CP-ABE), Access Structure, Anonymity, Provable Security, Attack.

1 Introduction

The concept of Broadcast Encryption (BE) 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

- *R. Rabaninejad is with the with the Department of Electrical Engineering, K. N. Toosi University of Technology, Tehran. Iran. E-mail: rabaninejad@gmail.com*
- *M. H. Ameri is with the Electronics Research Institute of Sharif University of Technology, Tehran, Iran. E-mail: ameri_mohammadhasan@ee.sharif.edu*
- *M. Delavar is with the Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran. E-mail:mdelavar@iust.ac.ir*
- *J. Mohajeri is with the Electronics Research Institute of Sharif University of Technology, Tehran, Iran. E-mail: mohajer@sharif.edu*

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. BE schemes are helpful in several applications including TV subscription services, access control in encrypted file systems, copyrighted content protection and group key distribution.

Since the introduction of BE in 1993 by Fiat and Naor [1], lots of schemes have been proposed (see e.g., [2]–[8]). 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 massage 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 [9]–[12].

In the traditional BE schemes, the authorized receiver, in order to decrypt the ciphertext correctly, needs information about the intended set of receivers, S. 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. [13], proposed the first anonymous BE 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. [14] suggested another anonymous BE scheme in the standard model with overhead linear in the number of receivers. Schemes [13] and [14] 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 [15], is another definition which only guarantees the anonymity of intended receivers from the view of users outside of the set S. But, users in S, can still learn the identities of other legal receivers. Fazio et al. in [15], proposed an outsider-anonymous BE with sublinear overheads. Attribute-based anonymous multicast scheme presented by Yu et al. [10], which we call it 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 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 the 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 [16].

- We develop an enhanced scheme in composite order bilinear groups, called Improved-YRL, which is secure against the proposed attack. We also prove the security of Improved-YRL in the standard model using dual system encryption technique [17]. Our proof is based on the security model for adaptive CCA adversaries proposed in [14] which considers anonymity and indistinguishability in one security game, simultaneously.
- We demonstrate that the new scheme retains low performance overhead 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 [8] that "it is a long-standing open problem to build a low-overhead anonymous broadcast encryption system"; so presenting Improved-YRL as an anonymous BE 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. 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 $a, b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
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

The notion of composite order maps was first introduced in [18]. Let $\mathcal{G}$ be an algorithm called group generator. 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". 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 $r = g_i^{\alpha}$ where α is a random element in 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 [10]. Let n denote the total number of attributes. Each user is assigned an n-element string, $\{Att_{i,b} \mid \forall i \in Z_n, b = 0 \text{ or } 1\}$, such that $Att_{i,0}$ and $Att_{i,1}$ 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 $'0'$ imply that the user does not have the corresponding attribute and the bit $'1'$ show 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 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

YRL [10] which aims solving the group key distribution problem is a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) scheme in which the intended subset of users, S, is specified with the access policy, T. YRL, also provides anonymity, i.e., unlike some CP-ABE schemes, in which the access policy T is clearly broadcasted along with the ciphertext, in the YRL scheme, the access policy T is not sent and the authorized users can decrypt the received messages without knowing the access structure. The scheme consists of four algorithms: Setup, KeyGen, Encryption and Decryption. In the following, we will describe in detail how the scheme works.

Setup. This algorithm chooses a group G of order p with generator g. Each attribute is mapped to one of the members of G. Consider $h_{i,b}$ as the member of G which the attribute $Att_{i,b}$ is mapped to. 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 the 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. This algorithm takes the master key MK and the attribute set of each user, $X_{n-1}X_{n-2}...X_0$, as input and outputs the secret key of the user through Relation 1.

$$SK = (D = g^{(\alpha+r)/\beta}, \hat{D} = g^r, \breve{D} = g^{\beta r}, \{D_i = h_{i,\bar{X}_i}^r\}_{\forall i \in Z_n}) \tag{1}$$

Where $r \in_R Z_p$.

Encryption. In order to distribute the group key, GK, the broadcaster first encrypts GK using this algorithm. It takes GK, the access policy, T, and MK and generates a ciphertext $CT = (\tilde{C}, \breve{C}, \{\hat{C}_j\}_{j=0,1}, \{C_i\}_{\forall i \in Z_n})$. The first term of CT is of the form $\tilde{C} = (GK \parallel MAC).X$, where X is a blinding factor to hide the value of $(GK \parallel MAC)$. The other three terms in the ciphertext are used to construct X and obtain GK. In the last term, each C_i is generated corresponding to the i-th bit of the attribute set, $X_{n-1}X_{n-2}...X_0$ through the following procedure:

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 was 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 $C_{i,X_i} = h_{i,X_i}^{s_i+t_i}$ 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 was 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:

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

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

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

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

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

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

where $x' \in Z_p$ is defined in Equation 6:

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

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

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

Equation 7 is used in calculating 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 was defined in encryption algorithm as $g^x = \prod_{\forall j, X_j \in T} h_{j,X_j}^{t_j}$).

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

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

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, 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. [10] also claim that the scheme is secure, meaning that a user can obtain the correct GK iff his attributes satisfy the access policy. But, 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 as follows:

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

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

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, $g^{s'} = \prod_{i=0}^{n-1} C_{i,0}C_{i,1}$; but after that $C_{i,0}$ and $C_{i,1}$ were updated to new values, $C_{i,0} = g^{k_0}C_{i,0}, C_{i,1} = g^{k_1}C_{i,1}$. 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:

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

   So,

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

Furthermore, we have:

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

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

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

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:

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

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 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 YRL. 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 $e(g,g)^{rn(k_0+k_1)}$ 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 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, Improved-YRL is based on composite order bilinear maps. In what follows, we describe the Improved-YRL scheme in detail:

Setup. This 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_3}. Hence $h_{i,b}$ is an element of the subgroup $G_{p_1p_3}$. The algorithm outputs 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. This algorithm takes the master key MK, the attribute set of a user, $X_{n-1}X_{n-2}\dots X_0$, chooses random $r \in_R Z_{p_1}$ and outputs the user's private key using Equation 17:

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

So 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 members of $G_{p_1p_3}$.

Encryption. 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 $\{\hat{C}_j = g^{k_j/\beta}\}_{j=0,1}$ from the ciphertext because the decryption successfully works without it. Also g is turned into g_1, so the first term in C_i is $g_1^{s_i} \in G_{p_1}$ and 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.

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

1) The values, $s_0, s_1, ..., s_{n-1} \in_R Z_{p_1}$ are randomly selected and 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 Equation 18:

$$\prod_{i=0}^{n-1} C_{i,0}C_{i,1} = g_1^{s'} R_3 = g_1^{\delta+x} R_3 \tag{18}$$

Where R_3 is the product of all elements in G_{p_3}. Also, $x \in Z_{p_1}$ is such that $g_1^x = \prod_{\forall j, X_j \in T} h_{j,X_j}^{t_j}$, 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.
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 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 $T = \bar{X}_3 X_2 X_0$, 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 $CT = (\tilde{C} = (GK \parallel MAC)e(g_1, g_1)^{\alpha s'}, \breve{C} = g_1^{\beta s'} R_3', \{C_i\}_{\forall i \in Z_n})$, 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. 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_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 Equation 19:

$$\begin{aligned} F_i &= e(D_i, g_1^{s_i}) e(C_{i,X_i}, \hat{D}) \\ &= e(h_{i,\bar{X}_i}^r, g_1^{s_i}) e(g_1^{k_i} h_{i,X_i}^{s_i+t_i}, g_1^r) \\ &= e(g_1, g_1)^{r\gamma_i s_i} e(g_1, h_{i,X_i})^{rt_i} e(g_1, g_1)^{k_i r} \end{aligned} \tag{19}$$

It can be easily verified that the elements of G_{p_3} 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_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 the Equation 21 where only G_{p_1} part of h_{j,X_j} is considered:

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

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:

$$M' = \frac{\tilde{C}}{e(\breve{C}, D)/F} = \frac{(GK \parallel MAC) e(g_1, g_1)^{\alpha s'}}{e(g_1, g_1)^{\alpha s' + rs'} / e(g_1, g_1)^{r(\delta + x' + k)}} = (GK \parallel MAC) e(g_1, g_1)^{r(k + x' - x)} \tag{22}$$

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 security of the proposed scheme and prove that it achieves both inditiguishability 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 $\prod_{i=0}^{n-1} C_{i,0} C_{i,1} = g_1^{k_r} . g_1^{s'} R_3$, 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 Definition

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 [14].

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

Setup. The challenger $\mathcal{C}$ runs Setup algorithm to generate the master key MK.

Phase1. 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 sectret keys $SK_1, SK_2, \ldots, SK_{q'}$ to $\mathcal{A}$. $\mathcal{A}$ 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 $\mathcal{C}$ 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 $\mathcal{A}$.

Phase2. $\mathcal{A}$ 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 sectret keys $SK_{q'+1}, SK_{q'+2}, \ldots, SK_q$. $\mathcal{A}$ 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. $\mathcal{A}$ 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 security of the Improved-YRL scheme. Rao et al. in [19] closely followed [20] 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 $N = p_1p_2p_3$, 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, 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 to distinguish $T_0 \in G$ 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:

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

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 to distinguish $T_0 \in G_{p_1}$ 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:

$$Adv^2_{\mathcal{A}} = |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]| \leq negl(\lambda) \tag{24}$$

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 to distinguish $T_0 = e(g_1, g_1)^{\alpha\varsigma}$ from T_1 which is a random element in G_T, is negligible. In other words, for any PPT algorithm $\mathcal{A}$, we have:

$$Adv^3_{\mathcal{A}} = |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]| \leq negl(\lambda) \tag{25}$$

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

6.3 Security Proof

Here we use a technique called dual system encryption [17] to prove 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 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, we first run Encryption algorithm 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 below

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

where δ is chosen randomly from Z_N and in $C_i' = C_i \times {g_2}^{\delta/n}$ only the terms $C_{i,0}, C_{i,1}$ are multiplied by ${g_2}^{\delta/n}$. Also, 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 $D_i = h^r_{i,\bar{x_i}}$ 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 below

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

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 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 $Game_q$, all the secret keys would be semi-functional.

$Game_{final}$: This game is the same as $Game_q$ except that the ciphertext is randomized. 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:

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

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 $Adv_{\mathcal{A}}^{Game_{ANO-IND-CCA}} - Adv_{\mathcal{A}}^{Game_0} = \varepsilon$. 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, 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. $\mathcal{B}$ 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 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 $CT_b = (\tilde{C}, \breve{C}, \{C_i\}_{\forall i \in Z_n})$. 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:

$$\begin{aligned} \tilde{C}' &= \tilde{C} \times e(g_1^{\alpha}, T) \\ \breve{C}' &= \breve{C} \times T^{\beta} \\ C'_{i_{0,1}} &= C_{i_{0,1}} \times T^{1/n} \end{aligned} \tag{28}$$

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 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 2. 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 it 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 $\alpha, \beta \in Z_N$, 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 $\mathcal{B}$ 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:

$$\begin{aligned} & r_j \in_R Z_N \\ & D' = D \times (Y_1 Y_2)^{r_j/\beta} \\ & \hat{D}' = \hat{D} \times (Y_1 Y_2)^{r_j} \\ & D_i{}' = D_i \times (Y_1 Y_2)^{r_j e_i} \end{aligned} \tag{29}$$

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

$$\begin{aligned} & D' = D \times T^{1/\beta} \\ & \hat{D}' = \hat{D} \times T \\ & D_i{}' = D_i \times T^{e_i} \end{aligned} \tag{30}$$

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 $T \in G_{p_1 p_2}$, 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.

Also, 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:

$$\begin{aligned} & \tilde{C}' = \tilde{C} \times e(g_1^{\alpha}, (X_1 X_2 X_3)) \\ & \breve{C}' = \breve{C} \times (X_1 X_2 X_3)^{\beta} \\ & C'_{i_{0,1}} = C_{i_{0,1}} \times (X_1 X_2 X_3)^{1/n} \end{aligned} \tag{31}$$

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 interacted 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 3. Suppose that there exists a polynomial time algorithm $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{Game_q} - Adv_{\mathcal{A}}^{Game_{final}} = \varepsilon$. Then we can build a PPT algorithm $\mathcal{B}$ 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, b_i randomly from Z_{p_1} and keeps the master key $MK = (\beta, \{a_i, b_i\}_{\forall i \in Z_n})$. Here, $\mathcal{B}$ can not choose the parameter α 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 α directly.

Phase1 and Phase2. Here all the queried secret keys are semi-functional. In response to $\mathcal{A}$'s secret key queries, $\mathcal{B}$ first generates 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 as below:

$$\begin{aligned} & r_j \in_R Z_N \\ & D' = D \times (g_1^{\alpha} X_2)^{r_j/\beta} \\ & \hat{D}' = \hat{D} \times (g_1^{\alpha} X_2)^{r_j} \\ & D_i{}' = D_i \times (g_1^{\alpha} X_2)^{r_j e_i} \end{aligned} \tag{32}$$

Where e_i is as defined in Lemma 2.

Also, 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 as below:

$$\begin{aligned} & \tilde{C}' = \tilde{C} \times T \\ & \breve{C}' = \breve{C} \times (g_1^{\varsigma} Y_2 Y_3)^{\beta} \\ & C_{i_{0,1}}{}' = C_{i_{0,1}} \times (g_1^{\varsigma} Y_2 Y_3)^{1/n} \end{aligned} \tag{33}$$

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 interacted with $\mathcal{A}$ in $Game_q$. Else if $T \in_R G_T$, the challenge ciphertext is randomized and $Game_{final}$ is simulated. So $\mathcal{B}$ 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 previous lemmas that the $Game_{ANO-IND-CCA}$ 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 $1/2$. So this is true in $Game_{ANO-IND-CCA}$ 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 probablity greater than $1/2$. So the Improved-YRL scheme has both indistinguishability and anonymity and the proof is completed. □

7 Performance Evaluation

This section analyzes the performance of the proposed scheme. For this aim, we compute the computation, communication and storage overheads in terms of the total number of attributes in the network which is denoted by n. These results are achieved due to the YRL scheme's overheads presented in [10]. We denote modular multiplications, exponentiations and pairings over prime and composite orders by Mul, $Mul.cmp$, Exp, $Exp.cmp$, $Pair$ and $Pair.cmp$ respectively.

7.1 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 [10], Improved-YRL's setup has a term $2nExp.cmp$; 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 $2nMul.cmp$ is added. So the total overhead of the Improved-YRL's setup is $2nExp.cmp + 2nMul.cmp$.

In the KeyGen algorithm, there is no need to compute $g^{r\beta}$ in comparison with KeyGen algorithm of YRL. 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 YRL, 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 $\{B_j = e(g^{\frac{k_j}{\beta}}, g^{r\beta})\}_{j\in\{0,1\}}$ and consequently, the total number of pairing computations is reduced by two units. Therefore, the total computation overhead of Decryption algorithm becomes $nMul.cmp + (n+2)Pair.cmp$.

These results are summarized in Table 1.

7.2 Communication Overhead

The total communication overhead of the proposed scheme is $(n+1)\log_2 p_1 + 2n\log_2(p_1p_3) + \log_2 |G_T^{cmp}|$ as presented in Table 1

7.3 Storage Overhead

The main storage load for users comes from the secret key SK. The storage load of the YRL scheme is $(n+3)\log_2 p$ [10], 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.

8 Conclusion

In this paper, we investigated an anonymous broadcast encryption scheme called YRL 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, it 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

TABLE 1
Comparison between the proposed scheme and the basic YRL scheme [10]. Note that, according to [21], to achieve 192-bit security level we should select $\log_2 p_i = \log_2 p = 192$.

Criteria		Improved-YRL	YRL [10]
Computation Overhead	Setup	$2nExp.cmp + 2nMul.cmp$	$2nExp$
	KeyGen	$(n+2)Exp.cmp$	$(n+3)Exp$
	Encryption	$(3n+1)Exp.cmp$	$(3n+3)Exp$
	Decryption	$nMul.cmp + (n+2)Pair.cmp$	$nMul + (n+4)Pair$
Communication Overhead		$(n+1)\log_2 p_1 + 2n\log_2(p_1p_3) + \log_2(\lvert\mathbb{G}_T^{cmp}\rvert)$	$(3n+3)\log_2 p + \log_2(\lvert\mathbb{G}_T\rvert)$
Storage Overhead		$2\log_2 p_1 + n\log_2(p_1p_3)$	$(n+3)\log_2 p$

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 BE schemes with overheads related to the number of receivers.

References

[1] A. Fiat and M. Naor, "Broadcast encryption," in *Annual International Cryptology Conference*. Springer, 1993, pp. 480–491.

[2] J. A. Garay, J. Staddon, and A. Wool, "Long-lived broadcast encryption," in *Annual International Cryptology Conference*. Springer, 2000, pp. 333–352.

[3] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Annual International Cryptology Conference*. Springer, 2001, pp. 41–62.

[4] D. Halevy and A. Shamir, "The lsd broadcast encryption scheme," in *Annual International Cryptology Conference*. Springer, 2002, pp. 47–60.

[5] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Annual International Cryptology Conference*. Springer, 2005, pp. 258–275.

[6] D. Boneh and B. Waters, "A fully collusion resistant broadcast, trace, and revoke system," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 211–220.

[7] C. Delerablée, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in *International Conference on Pairing-Based Cryptography*. Springer, 2007, pp. 39–59.

[8] D. Boneh, B. Waters, and M. Zhandry, "Low overhead broadcast encryption from multilinear maps," in *International Cryptology Conference*. Springer, 2014, pp. 206–223.

[9] D. Lubicz and T. Sirvent, "Attribute-based broadcast encryption scheme made efficient," in *International Conference on Cryptology in Africa*. Springer, 2008, pp. 325–342.

[10] S. Yu, K. Ren, and W. Lou, "Attribute-based on-demand multicast group setup with membership anonymity," *Computer Networks*, vol. 54, no. 3, pp. 377–386, 2010.

[11] Z. Zhou and D. Huang, "On efficient ciphertext-policy attribute based encryption and broadcast encryption," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 753–755.

[12] B. Wesolowski and P. Junod, "Ciphertext-policy attribute-based broadcast encryption with small keys," in *International Conference on Information Security and Cryptology*. Springer, 2015, pp. 53–68.

[13] A. Barth, D. Boneh, and B. Waters, "Privacy in encrypted content distribution using private broadcast encryption," in *International Conference on Financial Cryptography and Data Security*. Springer, 2006, pp. 52–64.

[14] B. Libert, K. G. Paterson, and E. A. Quaglia, "Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model," in *Public Key Cryptography–PKC 2012*. Springer, 2012, pp. 206–224.

[15] N. Fazio and I. M. Perera, "Outsider-anonymous broadcast encryption with sublinear ciphertexts," in *International Workshop on Public Key Cryptography*. Springer, 2012, pp. 225–242.

[16] R. Rabaninejad, M. Delavar, M. H. Ameri, and J. Mohajeri, "On the security of yrl, an anonymous broadcast encryption scheme," in *Telecommunications, IST 2016, International Symposium on*. IEEE, 2016.

[17] B. Waters, "Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions," in *Advances in Cryptology-CRYPTO 2009*. Springer, 2009, pp. 619–636.

[18] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of Cryptography Conference*. Springer, 2005, pp. 325–341.

[19] Y. Sreenivasa Rao and R. Dutta, "Fully secure bandwidth-efficient anonymous ciphertext-policy attribute-based encryption," *Security and Communication Networks*, vol. 8, no. 18, pp. 4157–4176, 2015.

[20] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology–EUROCRYPT 2010*. Springer, 2010, pp. 62–91.

[21] C. Hu, R. Yang, P. Liu, Z. Yu, Y. Zhou, and Q. Xu, "Public-key encryption with keyword search secure against continual memory attacks," *Security and Communication Networks*, 2016.