An Efficient Identity-based Broadcast Signcryption Scheme

Ming Luo
School of Software, Nanchang University, Nanchang 330047, P. R. China
Email: lmhappy21@163.com

Chunhua Zou and Jianfeng Xu
School of Software, Nanchang University, Nanchang 330047, P. R. China
Email: zch168@tom.com, jianfeng_x@ncu.edu.cn

***Abstract*—Broadcast signcryption, which enables the broadcaster to simultaneously encrypt and sign the content meant for a specific set of users in a single logical step, provides the most efficient solution to this dual problem of confidentiality and authentication. Recently, several identity-based broadcast signcryption (IBBSC) schemes have been proposed. However, we find almost all IBBSC schemes that have been proposed until now do not satisfy register secrecy and forward secrecy. Following this, we propose a new IBBSC scheme and formally prove its security under the random oracle model for broadcast signcryption (IND-CCA2 and EUF-CMA) While we propose a secure IBBSC scheme, we do not compromise the performance. The proposed scheme only requires two pairing operations for end user devices with limited computing capability.**

***Index Terms*—identity-based cryptography; signcryption; broadcast signcryption; forward secrecy; random oracle**

I. INTRODUCTION

The concept of public key signcryption schemes was proposed by Zheng in 1997 [1]. The purpose of this kind of primitive is to perform encryption and signature in a single logical step in order to obtain confidentiality, integrity, authentication and non-repudiation more efficiently than the sign-then-encrypt approach. The drawback of this latter solution is to expand the final ciphertext size and increase the sender and receiver's computing time. In Zheng's approach, the public key of a signer is essentially a random string selected from a given set. Therefore, it is infeasible to prove that a party is indeed the sender for a given signcryption message. This problem can be solved via a certificate which provides an unforgeable and trusted link between the public key and the identity of the signer by the CA's signature. And there is a hierarchical framework that is called a public key infrastructure (PKI) to issue and manage certificates. However, this system requires a large amount of computing time and storage when the number of users increases rapidly. To simplify key management procedures of conventional PKIs, Shamir [2] introduced the concept of Identity-Based Cryptography (IBC) in 1984, but a satisfying identity based encryption scheme (IBE) only appeared in 2001. It was designed by Boneh and Franklin [3] and cleverly uses bilinear maps (the Weil or Tate pairing) over supersingular elliptic curves. Subsequently, several ID-based signcryption schemes were proposed [4,5,6]. The main practical benefit of IBC is in greatly reducing the need for, and reliance on, the public key certificates.

Broadcast signcryption, which enables the broadcaster to simultaneously encrypt and sign the content meant for a specific set of users in a single logical step, provides the most efficient solution to this dual problem of confidentiality and authentication. In 2004, Bohio *et al.* [7] proposed an authenticated broadcasting scheme for wireless ad-hoc networks and Mu *et al.* [8] proposed Identity-based authenticated broadcast encryption and distributed authenticated encryption, which achieve the same security goals as broadcast signcryption and hence, their schemes are also a broadcast signcryption scheme. The term broadcast signcryption was coined much later by Fagen Li et al. [9], but Zhang and Geng [10] showed that their scheme cannot be against outside attack and inside attack. In [11,12,13], Selvi *et al.* separately pointed out that in [7,8,9] schemes are insecure and they proposed secure schemes. For the scheme [11], every subscriber obtains the same session key from the broadcaster, if one of the subscribers wants to unregister from the broadcaster, the broadcaster needs to re-change and re-distribute the session key for the other subscribers. Recently, [14] and [15] proposed an efficient identity-based broadcast signcryption scheme respectively, their schemes have constant size ciphertext for the broadcaster, but the broadcaster needs to transmit the identity information of designated receivers, and the public key generated and delivered by the PKG is of size linear in the maximal value of the set of receivers, for the receiver ID_i the cost of multiplication and exponentiation operations depends on number of receivers, that computation overheads is too high for a user, thus their schemes do not apply to the end user with limited computing capability. Moreover, the schemes all above

Corresponding author: Ming Luo (lmhappy21@163.com).
Ming Luo is supported by the the National Natural Science Foundation of China under grant no. 61070139 and the Science and Technology Foundation of the Education Department of Jiangxi Province under grant no. GJJ11039.

doi:10.4304/jsw.7.2.366-373

do not satisfy the register secrecy and forward secrecy security attributes.

In this paper, we propose an identity-based broadcast signcryption scheme, and formally prove its security (confidentiality and unforgeability) under the strongest existing security models for broadcast signcryption (IND-CCA2 and EUF-CMA respectively). Compared with other broadcast signcryption schemes regarding the security and computation overheads, we believe that our scheme is more efficient and more suitable for broadcast system devices with low computational capabilities. Our scheme has the following merits: (1) one subscriber can securely register and unregister from a broadcaster without affecting the other subscribers; (2) the scheme satisfies the forward secrecy attribute.

The remainder of this paper is organized as follows. The preliminaries for bilinear pairings and security definitions are given in the next section. The formal models of identity-based broadcast signcryption are described in Section 3. Section 4 describes a concrete identity-based broadcast signcryption scheme. The security analysis and discussions of the proposed scheme are presented in Section 5. In Section 6, the performance comparison among the proposed scheme and the recently proposed schemes is presented. Section 7 gives our conclusion and the future work of this research.

II. Preliminaries

In this section, the mathematical preliminaries required to understand the identity-based broadcast signcryption scheme presented in the section IV are introduced. Using the notation of the first encryption scheme using bilinear pairings proposed by Boneh & Franklin [3], let G_1 be an additive group of prime order q and G_2 be a multiplicative group of the same order q. Assume the existence of a map $\hat{e}$ from $G_1 \times G_1$ to G_2. Typically, G_1 will be a subgroup of the group of points on an elliptic curve over a finite field, G_2 will be a subgroup of the multiplicative group of a related finite field and the map $\hat{e}$ will be derived from either the Weil or Tate pairing on the elliptic curve. The mapping $\hat{e}$ must be efficiently computable and has the following properties.

1) Bilinearity: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all P, $Q \in G_1$, $a, b \in Z_q^*$
2) Non-degeneracy: There exists P and $Q \in G_1$ such that $\hat{e}(P, Q) \neq 1_{G2}$
3) Computability: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in G_1$

The security of our scheme described here relies on the hardness of the following problems.

Definition 1*:* Elliptic Curve Discrete Logarithm Problem (ECDLP): Given a group G_1 and two elements P, $Q \in G_1$, the ECDLP in G_1 is to compute x given $(P, Q = xP)$.

Definition 2*:* Bilinear Inverse Diffie-Hellman Problem (BIDHP): Given two groups G_1 and G_2 of the same prime order q, a bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ and a generator P of G_1, the BIDHP in $(G_1, G_2, \hat{e})$ is to compute $\hat{e}(P, P)^{ab^{-1}}$ given (P, aP, bP).

Definition 3*:* Inverse Computational Diffie-Hellman Problem (Inv-CDHP) in G_1*:* Given (P, aP), where $a \in Z_q^*$, compute $a^{-1}P$.

III. Formal Models of Identity-based Broadcast Signcryption

In the section, we present the generic model and security model of identity-based broadcast signcryption. In our models, there are three types of entities: broadcaster; users who want to subscribe or register to the broadcaster; trusted authority called the Private Key Generator (PKG).

A. Generic Model

The model of identity-based broadcast signcryption consists of the following four algorithms:

Setup: On input of a security parameter k the PKG uses this algorithm to produce its master public/private key pair (P_{pub}, s). It also outputs some *params* which are the global public parameters for the system.

Extract: On input of an identity ID_U, system's public parameters *params* and PKG's corresponding private key s, the PKG uses this algorithm to compute public and private key pair (Q_U, D_U) corresponding to ID_U.

Signcrypt： To send a message m to t users with identities $(ID_1, ID_2, ..., ID_t)$, the broadcaster B with identity ID_B and private key D_B uses this algorithm with input $(m, D_B, \Phi = \{ID_1, ID_2, ..., ID_t\})$ to compute and produce a ciphertext σ.

UnSigncrypt: When a user with identity ID_A and private key D_A receives the signcrypted ciphertext σ from his broadcaster B, he uses this algorithm with input (σ, D_A) to obtain either the plain text m or $\perp$ according as whether σ was a valid signcryption from broadcaster B or not.

The above algorithms have the following consistency requirement. If s =signcrypt$(m, D_B, \Phi = \{ID_1, ID_2, ..., ID_t\})$,then we must have m=Unsigncrypt(σ, D_A) for $ID_A \in \{ID_1, ID_2, ..., ID_t\}$.

B. Security Model

The two basic security properties that are desired out of any IBBSC scheme are message confidentiality and unforgeability. We formally extend the existing strongest security notions for encryption and digital signature (IND-CCA2 and EUF-CMA respectively) to IBBSC below.

Definition 4 (Message Confidentiality). *An identity-based broadcast signcryption scheme(IBBSC) is said to have the indistinguishability against adaptive chosen ciphertext attacks property (IND-IBBSC-CCA2) if no polynomially bounded adversary A has a non-negligible advantage in the following game.*

Game

– **Initial:** The challenger C runs the Setup algorithm with a security parameter k and sends the system parameters to the adversary A.

– **Phase1:** A performs a polynomially bounded number of the following queries (A can present its requests adaptively – every request may depend on the answer to the previous ones):
- **Extract query:** C runs the algorithm Extract and returns D_U to A.
- **Signcrypt query:** A produces a message m, broadcaster identity ID_B and a list of receiver identities $\{ID_1, ID_2, ..., ID_t\}$. C computes σ = signcrypt(m, D_B, $\Phi = \{ID_1, ID_2, ..., ID_t\}$) and sends σ to A.
- **UnSigncrypt query:** A produces a broadcaster identity ID_B, a receiver identity ID_A and a ciphertext σ. C computes Unsigncrypt(σ, D_A) and sends the result to A. The result returned is $\perp$ if σ is an invalid signcrypted ciphertext from broadcaster.

At the end of Phase1 A chooses two plaintexts (m_0, m_1), an arbitrary broadcaster identity ID_B and the set of identities of the receivers of that broadcaster $\Phi = \{ID_1, ID_2, ..., ID_t\}$, on which he wishes to be challenged. He cannot have made an Extract query on the broadcaster identity ID_B in the first stage.

– **Challenge:** The challenger takes a bit $b \in_R \{0,1\}$ and computes σ = signcrypt(m_b, D_B, $\Phi = \{ID_1, ID_2, ..., ID_t\}$) which is sent to A.

– **Phase 2:** A continues to probe the challenger with the same type of queries that it made in Phase 1. It is not allowed to obtain the private key of broadcaster identity ID_B and it is not allowed to make an UnSigncrypt query for σ.

– **Response:** A returns a bit b'. We say that the adversary wins if $b' = b$.

Definition 5 (Signature Unforgeability). *An identity-based broadcast signcryption scheme(IBBSC) is said to have the existential unforgeability against adaptive chosen messages attacks (EUF-IBBSC-CMA) if no polynomially bounded adversary A has a non-negligible advantage in the following game.*

Game

– **Initial:** The challenger C runs the Setup algorithm with a security parameter k and sends the system parameters and PKG's public key to the adversary A.

– **Probing:** A performs a polynomially bounded number of queries just like in the Definition 4.

– **Forge:** A outputs a forgery $(s^*, \{ID_1, ID_2, ..., ID_t\}, ID_B)$ from an arbitrary broadcaster ID_B that is not produced by the Signcrypt oracle, we say A wins the game if the result of Unsigncrypt(σ, D_A) is not $\perp$, where D_A corresponds to $ID_A \in \{ID_1, ID_2, ..., ID_t\}$.

IV. Proposed Scheme

Our scheme consists of the following concrete algorithms:

Setup: Suppose G_1 is an additive cyclic group of prime order q, and G_2 is a multiplicative cyclic group of the same order. Suppose P is a generator of G_1. There exists a bilinear pairing map $\hat{e}$ from $G_1 \times G_1$ to G_2 and cryptographic hash functions H_1: $\{0,1\}^n \rightarrow Z_q^*$, H_2: $G_2 \times (G_1)^2 \rightarrow \{0,1\}^n$, H_3: $G_2 \rightarrow Z_q^*$, H_4: $(G_1)^3 \times \{0,1\}^n \rightarrow Z_q^*$ and H_5: $G_1 \rightarrow \{0,1\}^n$. A PKG selects a random number $s \in Z_q^*$ as the private key and computes the public key $P_{pub} = sP$. The public parameters of the system are $<G_1, G_2, \hat{e}, q, P, P_{pub}, H_1, H_2, H_3, H_4, H_5>$. This phase is executed only once.

Extract: A communication entity U submits his identity ID_U to the PKG. The PKG first computes $a_U = H_1(ID_U)$, then computes $Q_U = (a_U + s)P$ and $D_U = (a_U + s)^{-1}P$ as the public key and private key of the entity U respectively.

Suppose that broadcaster B_j gets his public and private key pair ($Q_{B\text{-}j}$, $D_{B\text{-}j}$), and user i gets his public and private key pair (Q_i, D_i).

Register: This phase is executed whenever a user A with identity ID_A wants to subscribe or register to a broadcaster B with identity ID_B. This phase is further divided into user authentication and broadcaster recordation phases.

1. In the user authentication phase, user A follows the steps below.
   (a) Choose one random nonce $r \in Z_q^*$, compute $R_0 = rQ_B$ and $R_1 = rD_A$
   (b) Compute $u = \hat{e}(P,P)^r$
   (c) Compute $z = H_2(u, R_0, Q_A) \oplus R_1$
   (d) Send (z, R_0) to the broadcaster B
2. In the broadcaster recordation phase, broadcaster B follows the steps below.
   (a) Compute $u' = \hat{e}(D_B, R_0)$
   (b) Recover $R_1 = z \oplus H_2(u', R_0, Q_A)$
   (c) Check if $u' \stackrel{?}{=} \hat{e}(R_1, Q_A)$
   (d) If the check succeeds, set $z_i = H(u', R_0, Q_A)$, in which H is a key derivation function, add the entry (ID_A, z_i) to his subscriber list L_B and update his subscriber polynomial as $f_B(z) = f_B(z) \cdot (z - z_i)$. For all broadcasters B, $f_B(z)$ is initially set to 1.

Unregister: When a user A with identity ID_A is unsubscribing or unregistering from a broadcaster B with identity ID_B, he generates an unsubscribe request message m_0 and uses his precomputed secret $z_i = H(u, R_0, Q_A)$ to computes $y_1 = z_i \oplus m_0$, then sends y_1 to the broadcaster. The broadcaster will look up (ID_A, z_i) in his subscriber list L_B and retrieve $m_0 = z_i \oplus y_1$. If m_0 is correct, the broadcaster updates his subscriber polynomial as $f_B(z) = f_B(z) \cdot (z - z_i)^{-1}$. He also removes the entry (ID_A, z_i) from the list L_B.

Signcrypt: When broadcaster B wants to send a message m to his subscribers 1, 2, . . ., t, he does the following.

1. Choose one random nonce $k \in Z_q^*$ and compute $P_1 = kc_1P$, $P_2 = kc_2P$, . . ., $P_t = kc_tP$, where c_i is the coefficient of z^i in $f_B(z)$.
2. Compute the following.
   (a) Compute $R = kQ_B$
   (b) Compute $l = \hat{e}(Q_B, Q_A)^k = \hat{e}(Q_A, R)$
   (c) Compute $U = H_3(l)P$
   (d) Compute $P_0 = kc_0P + U$

(e) Compute $h=H_4(Q_B, P_0, R, m)$
(f) Compute $V=k^{-1}(hP+D_B)$
(g) Compute $y=(m||R||V)\oplus H_5(U)$

3. Broadcast the signcryption message $s =(ID_B, y, P_0 P_1, \ldots, P_t)$.

Unsigncrypt: When receiving $s =(ID_B, y, P_0 P_1, \ldots, P_t)$, he does the following.

1. Compute $U = \sum_{j=0}^{t} z_i^j P_j$
2. Recover $m||R||V= y \oplus H_5(U)$
3. Compute $l' = \hat{e}(R, Q_A)$
4. Check if $U' \stackrel{?}{=} H_3(l')P$
5. Compute $h=H_4(Q_B, P_0, R, m)$
6. Check if $\hat{e}(R,V) \stackrel{?}{=} \hat{e}(hP, Q_B)\hat{e}(P,P)$
7. If the check succeeds, return m. Else, return $\perp$.

V. Security Analysis

Based on the Elliptic Curve Discrete Logarithm Problem, Bilinear Inverse Diffie-Hellman Problem and Inverse Computational Diffie-Hellman Problem in the random oracle model, we show that the proposed scheme offers message confidentiality, signature non-repudiation, register secrecy and forward secrecy security attributes.

A. Basic security

Theorem 1. In the random oracle model, our identity-based broadcast signcryption scheme is secure against any IND-IBBSC-CCA2 adversary A if BIDHP is hard.

Proof. Let P be the generator of G_1.We assume the challenger C receives a random instance (P, aP, bP) of the Bilinear Inverse Diffie-Hellman Problem. His goal is to compute $\hat{e}(P,P)^{ab^{-1}}$. C will run A as a subroutine and act as A's challenger in the *IND-IBBSC-CCA*2 game *of Definition* 4. To maintain consistency between queries made by A, C keeps the following lists: L_i for i = 1, 2, 3, 4, 5 of data for query/response pairs to random oracle H_i; L_s of signcryptions generated by the simulator; and L_d of some of the queries made by A to the unsigncrypt oracle. At the beginning of the game, the adversary A outputs a list $\Phi = \{ID_1, ID_2, \ldots, ID_t\}$ of the users whom he proposes to attack, and the identity ID_B of the broadcaster who signcrypts the message to these users. Then, C gives A the system parameters.

***$H_1(ID_i)$* queries:** C searches an element (ID_i, h_i, w, Q_i, q_i) in the list L_1. If such an element is found, C answers $h_i =H_1(ID_i)$, otherwise, he does the following.

1. If $ID_i= ID_B$, C sets $w=\perp$ and $Q_i = bP$
2. If $ID_i \neq ID_B$, C chooses a random number $w\in Z_q^*$ and sets $Q_i = wP$
3. If ID_i is an identity of a broadcaster, add the tuple $(ID_i, h_i, w, Q_i, \{1\})$ to L_1 and answers h_i
4. If ID_i is not an identity of a broadcaster, C add the tuple $(ID_i, h_i, w, Q_i, \emptyset)$ to L_1 and answers h_i

If ID_i is an identity of a broadcaster, we use q_i to denote the set of coefficients of the subscriber polynomial (which is initially just the constant term 1). Otherwise, if it is a user's ID, we use it to store the set of $(z_i, ID_{B\text{-}j})$ values (where z_i is the precomputed secret of the user ID_i and $ID_{B\text{-}j}$ is the broadcaster to whom, when registering, (z, R_0) was sent by the user).

***$H_2(u, Q_i, R_0)$* queries:** C searches an element (u, Q_i, R_0, h_2) in the list L_2. If such an element is found, C answers h_2, otherwise he answers A by a random number $h_2\in \{0,1\}^n$ and puts the (u, Q_i, R_0, h_2) into L_2.

***$H_3(l)$* queries:** C checks if there exists (l, h_3) in L_3. If such an element is found, C answers h_3, otherwise he answers A by a random binary sequence $h_3\in Z_q^*$ and puts the (l, h_3) into L_3.

***$H_4(Q_{B\text{-}j}, P_0, R, m)$* queries:** C checks if there exists $(Q_{B\text{-}j}, P_0, R, m, h_4)$ in L_4. If such an element is found, C answers h_4, otherwise he answers A by a random binary sequence $h_4\in Z_q^*$ and puts the $(Q_{B\text{-}j}, P_0, R, m, h_4)$ into L_4.

***$H_5(U)$* queries:** C checks if there exists (U, h_5) in L_5. If such an element is found, C answers h_5, otherwise he answers A by a random binary sequence $h_5\in \{0,1\}^n$ and puts the (U, h_5) into L_5.

Extract(ID_i) queries: On a corruption query ID_i, we assume that $H_1(ID_i)$ query for ID_i has been asked. If $ID_i=ID_B$, then C fails and stops. Otherwise, C will check the list L_1 and return $D_i=w^{-1}P$ to A.

Register*(ID_i, $ID_{B\text{-}j}$, z, R_0)* queries: If L_1 does not contain an entry for ID_i or $ID_{B\text{-}j}$, then abort. Otherwise, consider the following two cases.

Case 1: $ID_i \notin \Phi$. C obtains the private key $D_{B\text{-}j}$ corresponding to the broadcaster by running the Extract query. Then C computes $u'= \hat{e}(D_{B\text{-}j},R_0)$, recovers $R_1= z\oplus H_2(u', R_0, Q_A)$ and checks if $u' \stackrel{?}{=} \hat{e}(R_1, Q_A)$. If not, then abort. Otherwise, he sets $z_i= H(u',Q_i,R_0)$, updates the tuple (ID_i, h_i, w, Q_i, q_i) in L_1 by setting $q_i = q_i \cup \{(z_i, ID_{B\text{-}j})\}$, retrieves the tuple $(ID_{B\text{-}j}, h_i, w, Q_{B\text{-}j}, q_{B\text{-}j})$ from L_1, where $q_{B\text{-}j} = \{c_0, c_1, \ldots, c_t\}$, constructs the new subscriber polynomial as $f_{B-j}(z) = (\sum_{l=0}^{t} c_l z^l)\cdot(z - z_i)$, let the set of new coefficients be $q'_{B\text{-}j} = \{c'_0, c'_1, \ldots, c'_t\}$, finally updates this tuple in L_1 by replacing $q_{B\text{-}j}$ with $q'_{B\text{-}j}$.

Case 2: $ID_i\in \Phi$. The adversary should not be allowed to register the member because then he'll trivially have all the information he needs to designcrypt the signcryption. So, the oracle ignores the last one parameter. It instead retrieves the tuples (ID_i, h_i, w, Q_i, q_i) and $(ID_{B\text{-}j}, h_i, w, Q_{B\text{-}j}, q_{B\text{-}j})$ from L_1 and, takes $R_0= R_0^* =aP$ and executes the following steps.

1. Updates the tuple (ID_i, h_i, w, Q_i, q_i) in L_1 by setting $q_i = q_i \cup \{(z_i, ID_{B\text{-}j})\}$, where $z_i= H(\Psi ,Q_i,R_0)$ (Ψ is C candidate for the BIDHP)
2. Perform Step 2 exactly as in the previous case.

Unregister*(ID_i, $ID_{B\text{-}j}$)* queries: If L_1 does not contain an entry for ID_i or $ID_{B\text{-}j}$, then abort. Otherwise, C does the following.

1. Obtain z_i from the tuple (ID_i, h_i, w, Q_i, q_i) in L_1, retrieve $m_0= z_i\oplus y_1$, check if m_0 is correct. If not, then abort.

2. Update the tuple (ID_i, h_i, w, Q_i, q_i) in L_1 by setting $q_i = q_i - \{(z_i, ID_{B-j})\}$. Temporarily store the value z_i for use in the next step.
3. Retrieve the tuple $(ID_{B-j}, h_i, w, Q_{B-j}, q_{B-j})$ from L_1, where $q_{B-j} = \{c_0, c_1, ..., c_t\}$. Construct the new subscriber polynomial as $f_{B-j}(z) = (\sum_{l=0}^{t} c_l z^l) \cdot (z - z_i)^{-1}$. Let the set of new coefficients be $q'_{B-j} = \{c'_0, c'_1, ..., c'_{t-1}\}$. Update this tuple in L_1 by replacing q_{B-j} with q'_{B-j}.

Signcrypt(m, D_{B-j}, $\Phi = \{ID_1, ID_2, ..., ID_t\}$) queries: We will assume that A makes the queries $H_1(ID_{B-j})$ before it makes a Signcrypt query for a plaintext m. We have the following two cases to consider.

Case 1: $ID_{B-j} \neq ID_B$. C checks if there is an entry for ID_{B-j} in L_1 and if the set q_{B-j} is not singleton. If one or both of these conditions are not satisfied, then C aborts. Otherwise, C retrieves the tuple $(ID_{B-j}, h_i, w, Q_{B-j}, q_{B-j})$ from L_1, where $q_{B-j} = \{c_0, c_1, ..., c_t\}$, obtains the private key D_{B-j} corresponding to the broadcaster by running the Extract query, and answers the query by a call to signcrypt(m, D_B, $\Phi = \{ID_1, ID_2, ..., ID_t\}$).

Case 2: $ID_{B-j} = ID_B$. C first retrieves the tuple $(ID_{B-j}, h_i, w, Q_{B-j}, q_{B-j})$ as in the case 1, where $q_{B-j} = \{c_0, c_1, ..., c_t\}$. Then C chooses $h, k \in Z_q^*$, computes $P_1 = kc_1P$, $P_2 = kc_2P, \ldots, P_t = kc_tP$, $R = kP$, $l = \hat{e}(R, Q_A)$, $U = H_3(l)P$, $P_0 = kc_0P + U$, $V = k^{-1}(hbP + P)$ and $y = (m||R||V) \oplus H_5(U)$. Finally, C returns $s = (ID_{B-j}, y, P_0\ P_1, \ldots, P_t)$ as the answer.

Unsigncrypt(σ, D_i) queries: On receiving this query, C checks if there are entries for ID_{B-j} and ID_i in L_1 and there is a tuple of the form $(z_i, ID_{B-j}) \in q_i$. If one or more of these conditions are not satisfied, then C returns $\perp$. Otherwise, C executes Unsigncrypt(σ, D_i) in the normal way and returns what the unsigncrypt algorithm returns.

After the first stage, A outputs messages (m_0, m_1) and broadcaster identity ID_B^*. If $ID_B^* \neq ID_B$, C aborts. Obviously, at this point, all the subscribers of the broadcaster B must be in Φ. This is because, in our scheme, the broadcaster always signcrypts messages for all his subscribers. Now, C chooses a random bit $b = b'$, and executes signcrypt(m_b, D_B, $\Phi = \{ID_1, ID_2, ..., ID_t\}$) as the case 2 of the signcrypt queries and returns what the signcryption algorithm returns as the challenge signcryption.

A then performs a second series of queries which is treated in the same way as the first one. At the end of the simulation, he produces a bit b' for which he believes that the challenge signcryption is the signcryption of $m_{b'}$ from ID_B to its subscribers. At this moment, if $b = b'$, C then answers 1 the answer to the BIDHP. Otherwise, it outputs 0. Since the adversary is denied access to the Unsigncrypt oracle with the challenge signcryption, he can recognize which message was signcrypted by seeing the signcryption alone, only if he has computed U, for which he must have computed the value of z_i for some user $ID_i \in \Phi$ who subscribes to the broadcaster B. This means, the z_i that he computes must be the same as the z_i that was used in the construction of the subscriber polynomial. We have,

$$\begin{aligned} z_i &= H(\Psi, Q_i, R_0) \\ &= H(\hat{e}(R_0^*, D_B), Q_i, R_0) \\ &= H(\hat{e}(aP, b^{-1}P), Q_i, R_0) \\ &= H(\hat{e}(P, P)^{ab^{-1}}, Q_i, R_0) \end{aligned}$$

So, if there exists a non-trivial adversary A who can defeat the signcryption by learning something about the encrypted message, that means there exists an algorithm to solve the BIDHP with non-negligible advantage. Since this is not possible, no adversary can defeat the signcryption this way. Hence, our proposed scheme is secure against any *IND-IBBSC-CCA*2 *adversary A* attack.

Theorem 2 (Unforgeability). In the random oracle model, our identity-based broadcast signcryption scheme is secure against any EUF-IBBSC-CMA adversary A if Inv-CDHP is hard in G_1.

Proof. Let P be the generator of G_1. We assume the distinguisher C receives a random instance (P, aP) of Inverse Computational Diffie-Hellman Problem. His goal is to compute $a^{-1}P$. C will run A as a subroutine and act as A's challenger in the *EUF-IBBSC-CMA* game *of Definition* 5. To maintain consistency between queries made by A, C keeps the following lists: L_i for $i = 1, 2, 3, 4, 5$ of data for query/response pairs to random oracle H_i; L_s of signcryptions generated by the simulator; and L_d of some of the queries made by A to the unsigncrypt oracle.

$H_1(ID_i)$ queries: C searches an element (ID_i, h_i, w, Q_i, q_i) in the list L_1. If such an element is found, C answers $h_i = H_1(ID_i)$, otherwise, he does the following.

1. C chooses a random number $w \in Z_q^*$ and sets $Q_i = wP$
2. If ID_i is an identity of a broadcaster, add the tuple $(ID_i, h_i, w, Q_i, \{1\})$ to L_1 and answers h_i
3. If ID_i is not an identity of a broadcaster, C add the tuple $(ID_i, h_i, w, Q_i, \varnothing)$ to L_1 and answers h_i

If ID_i is an identity of a broadcaster, we use q_i to denote the set of coefficients of the subscriber polynomial (which is initially just the constant term 1). Otherwise, if it is a user's ID, we use it to store the set of (z_i, ID_{B-j}) values (where z_i is the precomputed secret of the user ID_i and ID_{B-j} is the broadcaster to whom, when registering, (z, R_0) was sent by the user).

Extract(ID_i) queries: On a corruption query ID_i, we assume that $H_1(ID_i)$ query for ID_i has been asked. C will check the list L_1 and return $D_i = w^{-1}P$ to A.

Reister queries: If L_1 does not contain an entry for ID_i or ID_{B-j}, then abort. Otherwise, C obtains the private key D_{B-j} corresponding to the broadcaster B by running the Extract query. Then C computes $u' = \hat{e}(D_{B-j}, R_0)$, recovers $R_1 = z \oplus H_2(u', R_0, Q_A)$ and checks if $u' \stackrel{?}{=} \hat{e}(R_1, Q_A)$. If not, then abort. Otherwise, he sets $z_i = H(u', Q_i, R_0)$, updates the tuple (ID_i, h_i, w, Q_i, q_i) in L_1 by setting $q_i = q_i \cup \{(z_i, ID_{B-}$

$_{j})\}$, retrieves the tuple $(ID_{B\text{-}j}, h_i, w, Q_{B\text{-}j}, q_{B-j})$ from L_1, where $q_{B-j} = \{c_0, c_1, ..., c_t\}$, constructs the new subscriber polynomial as $f_{B-j}(z) = (\sum_{l=0}^{t} c_l z^l) \cdot (z - z_i)$, let the set of new coefficients be $q'_{B\text{-}j} = \{c'_0, c'_1, ..., c'_t\}$, finally updates this tuple in L_1 by replacing q_{B-j} with $q'_{B\text{-}j}$.

Signcrypt(m, $D_{B\text{-}j}$, $\Phi = \{ID_1, ID_2, ..., ID_t\}$) queries: We will assume that A makes the queries $H_1(ID_{B\text{-}j})$ before it makes a Signcrypt query for a plaintext m. C first checks if there is an entry for $ID_{B\text{-}j}$ in L_1 and if the set q_{B-j} is not singleton. If one or both of these conditions are not satisfied, then C aborts. Otherwise, C retrieves the tuple $(ID_{B\text{-}j}, h_i, w, Q_{B\text{-}j}, q_{B-j})$ from L_1, where $q_{B-j} = \{c_0, c_1, ..., c_t\}$, obtains the private key $D_{B\text{-}j}$ corresponding to the broadcaster by running the Extract query, and answers the query by a call to signcrypt(m, $D_{B\text{-}j}$, $\Phi = \{ID_1, ID_2, ..., ID_t\}$).

***H_2*, *H_3*, *H_4*, *H_5*, Unregister, Unsigncrypt queries:** these queries as in the proof of the Theorem 1.

At last, A chooses outputs a valid forgery $s^*_B = (ID^*_B, y^*, P^*_0, P^*_1, ..., P^*_t)$ on some message m^* from the broadcaster B to all his subscribers. C retrieves the entry corresponding to ID_B in L_1 and uses one of the tuples of (ID_i, h_i, w, Q_i, q_i), say $(ID_A, h_A, w, Q_A, q_i = (z_A, ID_B))$ to execute Unsigncrypt(s^*_B, D_A). If s^*_B is a valid signcryption from the broadcaster B to his subscribers, that is, a message m^* is returned by the unsigncrypt algorithm, then C applies the oracle replay technique to produce two valid signcryptions $s'_B = (ID_B, y', P'_0, P'_1, ..., P'_t)$ and $s''_B = (ID_B, y'', P''_0, P''_1, ..., P''_t)$ on some message m from the broadcaster B to all his subscribers(where $(P'_0, P'_1, ..., P'_t) = (P''_0, P''_1, ..., P''_t)$, and sets $P'_0 = c_0aP + U$, $P'_i = c_iaP$). C unsigncrypts s'_B and s''_B to obtain the signatures $V' = k^{-1}(h'P + D_B)$ and $V'' = k^{-1}(h''P + D_B)$. Now we can apply standard arguments for the outputs of the forking lemma since both V' and V'' are valid signatures for the same message m and same random tape of the adversary. Finally, C obtains the solution to the Inv-CDHP instance as $(h' - h'')^{-1}(V' - V'')$. We have

$$(h' - h'')^{-1}(V' - V'') = (h' - h'')^{-1}(h' - h'')k^{-1}P = k^{-1}P = a^{-1}P$$

So, we can see that the challenger C has the same advantage in solving the Inv-CDHP as the adversary A has in forging a valid signcryption. So, if there exists an adversary who can forge a valid signcryption with non-negligible advantage, that means there exists an algorithm to solve the Inv-CDHP with non-negligible advantage. Since this is not possible, no adversary can forge a valid signcryption with non-negligible advantage. Hence, our proposed scheme is secure against any EUF-IBBSC-CMA adversary A attack.

B. Further security considerations

In this section we will heuristically argue that the identity-based broadcast signcryption scheme satisfies the following security properties.

1. Register Secrecy: Suppose the adversary A wants to deceive broadcaster B into thinking Alice with identity ID_A and private key D_A wants to subscribe to her service, he can not forge the correct $u = \hat{e}(P,P)^r = \hat{e}(D_B, R_0)$ and $R_1 = rD_A$ to satisfy $u \stackrel{?}{=} \hat{e}(R_1, Q_A)$ without private key D_A of Alice. Given $(P, Q_U = aP)$, it is hard to compute $D_A = a^{-1}P$ under the assumption of Inv-CDHP. Thus in addition broadcaster B and Alice, no one can forge the the authentication value (z, R_0) and thus B can authenticate Alice by verifying the value (z, R_0). So, the adversary A cannot subscribe to any services for Alice without Alice's permission. But in [12] the adversary A can deceive broadcaster B into thinking Alice wants to subscribe to her service, since he can chooses a random number $r_i \neq y_i \in Z_q^*$ to generate valid register messages $(R = r_iQ_i, T = r_iQ_B)$ which satisfies $\hat{e}(R, Q_B) = \hat{e}(T, Q_i)$.

2. Forward Secrecy: In our scheme, compromise of previously established session key $z_{i\text{-}0}$ does not affect the secrecy of the later established session key $z_{i\text{-}1}$. Further, suppose the adversary knows the private key D_A of the subscriber Alice does not affect the secrecy of the signcryption on some message m. For the adversary A, he needs to know $z_{i\text{-}1} = H(\hat{e}(P,P)^r, R_0, Q_A)$ to obtain the message m, but he can't obtain the r to compute $\hat{e}(P,P)^r$. Given $(Q_B, R_0 = rQ_B)$, it is hard to compute r under the assumption of ECDLP. Hence, our scheme provides the forward secrecy. But in [12] if the private key S_i of the subscriber is compromised by an attacker, then the attacker can obtain the message m by computing the secret $x_i = H_2(a_i)$, where $a_i = \hat{e}(T, S_i)$; in [13] if the private key S_j of the subscriber is compromised by an attacker, then the attacker can obtain the message m by computing the secret $w' = \hat{e}(X', s_iP) = \hat{e}(\sum_{l=0}^{t} x_j^l P_l, s_iP)$, where $x_j = H_0(S_j)$.

VI. Protocol Comparison

In this section, we compare the efficiency of our scheme with other schemes appearing in the literature in Table 1 regarding the security and computation overheads not including precomputation overheads required by different phases including registration phase, signcryption phase and unsigncryption phase.

We use the following notations to analyze the computational complexity for our scheme and some existing previous schemes.

- t_a is the time for addition of two elements in the additive group $< G_1, + >$.
- t_m is the time for point scalar multiplication on the additive group $< G_1, + >$.
- t_g is the time for $x \in Z_q^*$ times multiplication in the multiplicative group $< G_2, \times >$.
- t_e is the time for bilinear pairing operation.
- t is the number of subscribers.

TABLE I.
A COMPARISON OF EFFICIENCY

Scheme	Registration		Signcryption	Unsigncryption	RA	FS
-	User	Broadcaster	Broadcaster	User	-	-
Selvi's scheme[12]	$2t_m$	$3t_e$	$(t+4)t_m+2t_a+t_g$	$2t_m+2t_e+t_a$	N	N
Selvi's scheme[13]	-	-	$(t+3)t_m+t_a+t_g$	$t_m+3t_e+t_a$	N	N
Our scheme	$3t_m$	$2t_e$	$(t+4)t_m+2t_a+t_g$	t_m+2t_e	Y	Y

- Y and N denote that the property holds and does not hold in the scheme respectively.

As we all know, a bilinear pairing operation is very time-consuming than other operations [3]. Table 1 summarizes the performance result of the proposed scheme in terms of the computational costs for the registration phase, sigcryption phase and unsigcryption phase, respectively.

As shown in the Table 1, our scheme is more efficient than that of [12,13] in terms of the security and computational costs of sigcryption phase and unsigcryption phase, and our scheme only requires two bilinear pairing operations in registration phase. Hence, consider the broadcast system devices with limited computing capability and communication security it may be that our identity-based broadcast signcryption scheme is more applicable.

Ⅶ. CONCLUSIONS

Broadcast encryption is useful for distributing digital contents to Internet users over a broadcast channel. It allows a center to deliver the encrypted data to a large set of users so that only a particular subset of privileged users can decrypt it. However, we find almost all IBBSC schemes that have been proposed until now do not satisfy register secrecy and forward secrecy. Following this, we have proposed a fixed version of the scheme to achieve register secrecy and forward secrecy attributes, also we have proven its IND-CCA2 and EUF-CMA security formally in the random oracle model. These are the strongest security notions for message confidentiality and authentication respectively. While we have proposed a secure IBBSC scheme, we have not compromised the performance. In fact, the proposed scheme only requires two pairing operations for end user devices with limited computing capability.

In the future, we will consider the influences of the number of privileged users on the size of the signcryption and the computation overheads on the broadcaster side. Another aspect of future work is further reducing the number of pairing computations during designcryption on the subscriber side. Besides, it is unrealistic to assume that a single trusted authority will be responsible for issuing secret keys to members of a large-scale network. Therefore, we will consider multiple domains environment where a subscriber can register to another domain broadcaster.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China under grant no. 61070139 and the Science and Technology Foundation of the Education Department of Jiangxi Province under grant no. GJJ11039.

REFERENCES

[1] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) (cost (signature) + cost (encryption))", in: *Proceedings of Cryptology-CRYPTO'97*, California, USA, pp. 165-179, 1997.
[2] A. Shamir, Identity-based cryptosystems and signature schemes, in: *Proceedings of the Cryptology - CRYPTO'84*, California, USA, pp. 47–53, 1984.
[3] D. Boneh, M. Franklin, "Identity based encryption from the Weil pairing", in: *Proceedings of the Crypto 2001*, California, USA, pp. 213- 229, 2001.
[4] G. Yu, X.X. Ma, Y. Shen, et al, "Provable secure identity based generalized signcryption scheme", *Theoretical Computer Science*, vol. 411, no. 40, pp. 3614-3624, 2010.
[5] B. Zhang, Q. L. Xu, "An ID-Based Anonymous Signcryption Scheme for Multiple Receivers Secure in the Standard Model," in: *Proceedings of Computer Science and Information Technology*, Chengdu, China, pp. 15-27, 2010.
[6] Z. P. Jin, Q. Y. Wen, and H. Z. Du, "An improved semantically-secure identity-based signcryption scheme in the standard model," *Computers & Electrical Engineering*, Vol. 36, pp. 545-552, 2010.
[7] M. J. Bohio, A. Miri, "An authenticated broadcasting scheme for wireless ad hoc network", in: *Proceedings of the 2nd Annual Conference on Communication Networks and Services Research (CNSR)*, Fredericton, N.B., Canada, pp. 69–74, 2004.
[8] Y. Mu, W. Susilo, Y. X. Lin, C. Ruan, "Identity-based authenticated broadcast encryption and distributed authenticated encryption", In: *Proceedings of the 9th Asian Computing Science Conference*, Chiang Mai, Thailand, pp. 169-181, 2004.
[9] F. G. Li, X. J. Xin, Y. P. Hu, "Identity-based broadcast signcryption", *Computer Standards and Interfaces*, vol. 30, no.1, pp. 89-94, 2008.
[10] J. H. Zhang, Q. Geng, "Comment on an ID-based Broadcast Signcryption Scheme", in: *Proceedings of the International Conference on Networking and Digital Society*, Guiyang,China, pp.37-40, 2009.
[11] S. S. D. Selvi, S. S. Vivek, N. N. Karuturi, "Cryptanalysis of Bohio et al.'s ID-Based Broadcast Signcryption (IBBSC)

Scheme forWireless Ad-hoc Networks”, In: *Proceedings of the PST 2008*, Fredericton, N.B., Canada, pp. 109-120, 2008.

[12] S. S. D. Selvi, S. S. Vivek, R. Gopalakrishnan, et al, “Provably Secure ID-based Broadcast Signcryption (IBBSC) Scheme”, in: *Cryptology ePrint Archive*, Report 2008/225, pp. 1-24, 2008.

[13] S. S. D. Selvi, S. S. Vivek, R. Gopalakrishnan, et al, “Cryptanalysis of Mu et al.'s and Li et al.'s Schemes and a Provably Secure ID-Based Broadcast Signcryption (IBBSC) Scheme”, in: *Proceedings of the WISA 2008*, Jeju Island, Korea, pp. 115-129, 2009.

[14] D. T. Hien, T. N. Tien, T. T. T. Hien, “An Efficient Identity-based Broadcast Signcryption Scheme,” in: *Proceedings of the KSE 2010*, Hanoi, Vietnam, pp. 209-216, 2010.

[15] I. T. Kim, S. O. Hwang, “An Efficient Identity-Based Broadcast Signcryption Scheme for Wireless Sensor Networks,” in: *Proceedings of the ISWPC 2011*, Hong Kong, China, pp. 1-6, 2011.

Ming Luo was born in 1983. He received the B.E. and Ph.D degree from Northeastern University, Shenyang, China in 2004 and 2010, respectively. Now he is a lecturer in the School of Software, Nanchang University, Nanchang, China. He has won lots of scholarships in China and was supported by the National Natural Science Foundation of China under grant no. 60602061, 60803131 and 61070139, the National High-Tech Research and Development Plan of China under grant no. 2006AA01Z413 and the Science and Technology Foundation of the Education Department of Jiangxi Province under grant no. GJJ11039. He has published more than twenty papers. His research interests are information security, networks security and cryptography.

Chunhua Zou was born in 1972. He received his Ph.D degree from Nanchang University China in 2007. Now he is a associate professor in the School of Software, Nanchang University, Nanchang, China. He has won lots of scholarships in China and participated in many computer software projects and published more than thirty papers in the computer area. His researches include network security and management, software engineering and embedded intelligent control system.

Jianfeng Xu was born in 1973. He received his M.S. degree from Nanchang University China in 2006. Now he is a associate professor in the School of Software, Nanchang University, Nanchang, China. He has won lots of scholarships in China and participated in many computer software projects and published more than thirty papers in the computer area. His researches include computer networking, information security, next generation network, IPv6 technology and cryptography.