

Cryptanalysis of Bohio et al.'s ID-Based Broadcast Signcryption (IBBSC) Scheme for Wireless Ad-hoc Networks

S. Sharmila Deva Selvi

sharmila@cse.iitm.ernet.in

S. Sree Vivek*

svivek@cse.iitm.ernet.in

Naga Naresh Karuturi†

nnaresh@cse.iitm.ernet.in

Ragavendran Gopalakrishnan

ragav@cse.iitm.ernet.in

Pandu Rangan Chandrasekaran*

prangan@iitm.ac.in

*Indian Institute of Technology Madras
Theoretical Computer Science Laboratory
Department of Computer Science and Engineering
Chennai, India*

Abstract

Broadcast signcryption enables the broadcaster to simultaneously encrypt and sign the content meant for a specific set of users in a single logical step. It provides a very efficient solution to the dual problem of achieving confidentiality and authentication during content distribution. Among other alternatives, ID-based schemes are arguably the best suited for its implementation in wireless ad-hoc networks because of the unique advantage that they provide - any unique, publicly available parameter of a user can be his public key, which eliminates the need for a complex public key infrastructure. In 2004, Bohio et al. [4] proposed an ID-based broadcast signcryption (IBBSC) scheme which achieves constant ciphertext size. They claim that their scheme provides both message authentication and confidentiality, but do not give formal proofs. In this paper, we demonstrate how a legitimate user of the scheme can forge a valid signcrypted ciphertext, as if generated by the broadcaster. Moreover, we show that their scheme is not IND-CCA secure. Following this, we propose a fix for Bohio et al.'s scheme, and formally prove its security under the strongest existing security models for broadcast signcryption (IND-CCA2 and EUF-CMA). While fixing the scheme, we also improve its efficiency by reducing the ciphertext size to two elements compared to three in [4].

*Work supported by Project No. CSE/05-06/076/DITX/CPAN on Protocols for Secure Communication and Computation sponsored by Department of Information Technology, Government of India

†Work supported by Project No. CSE/05-06/075/MICO/CPAN on Foundation Research in Cryptography sponsored by Microsoft Research India

1. Introduction

With the advent of mobile and portable devices such as cell phones and PDAs used in wireless networks, accessing multimedia content through these devices in the wireless network is becoming increasingly popular. On the other hand, a wireless network is much easier to eavesdrop than a wired network. Therefore, the need to securely deliver multimedia content to the user over a wireless network is becoming more important and critical. Furthermore, wireless communication is a good way to broadcast messages to many users in one go. In such applications, a central authority needs to deliver encrypted data to a large number of recipients in such a way that only a privileged subset of users can decrypt it. A broadcasting news channel may face this problem, for example, when a large number of people subscribe to a daily exclusive news feature. This is exactly the kind of problem that broadcast encryption attempts to efficiently solve. On top of this, especially in the current digital era, junk content or spam is a major turn off in almost every Internet application. If all the users who subscribe to the news feed receive meaningless noise or unwanted content, then the broadcaster is going to lose his subscribers. This results in the additional requirement that subscribers must have source authentication with respect to their broadcaster.

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. The efficiency of a broadcast signcryption scheme is mainly measured by three parameters - length of transmission messages, storage cost, and computational overhead at a user device. All these param-

ters are extremely important to mobile devices as they have limited memory and computational power as compared to a personal computer, and from the broadcaster's point of view, wireless bandwidth is an extremely costly resource. While several alternatives exist in implementing broadcast signcryption schemes, ID-based schemes are the best suited because of their unique advantage — any unique, publicly available parameter of a user can be his public key, which eliminates the need for a complex public key infrastructure.

Our Contribution. We give the general framework of an IBBSC scheme and define the formal security models for confidentiality and authentication for IBBSC (which we call IND-IBBSC-CCA2 and EUF-IBBSC-CMA respectively). In 2004, Bohio et al. [4] proposed an authenticated broadcasting scheme for wireless ad-hoc networks which is an IBBSC scheme, and claim that it provides message authentication and confidentiality though they prove neither property formally. In this paper, we demonstrate that any privileged user can forge the signcryption of any (possibly malicious) message as if the broadcaster signcrypted it. Moreover, we also show that their scheme is not IND-CCA secure (from the point of view of confidentiality). That is, during the IND-CCA game, when one of two messages of the adversary's choice is signcrypted and given back, he can easily find out which message was signcrypted. Following this, we propose an improvement to Bohio's ID-based broadcast signcryption (IBBSC) scheme to fix these security leaks, and formally prove its security (confidentiality and unforgeability) under the strongest existing security models for broadcast signcryption (IND-CCA2 and EUF-CMA respectively). We note that by fixing Bohio et al.'s IBBSC scheme, we do not hurt the efficiency of their scheme. In fact, we improve it by reducing the size of signcrypted ciphertext from three elements down to just two.

Organization. The rest of this paper is organized as follows. In Section 2, we review the underlying cryptographic concepts that are involved, like ID-based cryptography, signcryption, broadcast encryption, bilinear pairings and related computational problems, the general framework of ID-based broadcast signcryption (IBBSC) schemes and the formal security models for IBBSC. Next, in Section 3, we review the ID-based authenticated broadcast encryption scheme of Bohio et al. [4]. We present our attacks on this scheme in Section 4. Following this, in Section 5, we lay out the details of our improved IBBSC scheme, following the general framework of IBBSC schemes. We present the formal proofs of correctness, unforgeability and confidentiality of our improved scheme in Sections 6, 7 and 8 respectively. The proofs are presented in the strongest existing security models for IBBSC. In Section 9, we discuss the efficiency of our scheme. Finally, in Section 10, we conclude the discussion.

2. Preliminaries

2.1. Identity-based Cryptography

The concept of an Identity-based (ID-based) cryptosystem was introduced by Shamir in 1984 [26]. The distinguishing characteristic of *ID-based cryptography* is the ability to use any string as a public key. In particular, this string maybe the email address, telephone number, or any publicly available parameter of a user that is unique to him. The corresponding private key can only be derived by a trusted Private Key Generator (PKG) who keeps a master secret which is involved in deriving the private keys. Several early schemes that were proposed were unsatisfactory in different aspects. The first practical ID-based encryption (IBE) scheme was introduced by Boneh and Franklin in 2001 [8]. Since 2001, several schemes have been introduced [11, 27, 10, 7, 6, 5, 15].

There are several advantages offered by ID-based cryptography. If there are only a finite number of users, after all users have been issued with keys, the master secret key can be destroyed, because in the basic ID-based cryptosystem, keys once issued are always valid. Also, as public keys are derived from identities, IBE eliminates the need for a public key distribution infrastructure (PKI). The authenticity of the public keys is guaranteed implicitly as long as the transport of the private keys to the corresponding user is kept secure.

2.2. Signcryption

To avoid forgery and ensure confidentiality of the contents of a letter, for centuries it has been a common practice for the sender of the letter to sign his name on it and then seal it in an envelope, before handing it over to a deliverer. Public key cryptography now makes it possible for people who have never met before to communicate with one another in a secure and authenticated way over an open and insecure network such as the Internet. In doing so, this same two-step approach has been followed. Namely, before a message is sent out, the sender of the message would sign it using a digital signature scheme, and then encrypt the message (and the signature) using a private key encryption algorithm under a randomly chosen message encryption key. The random message encryption key would then be encrypted using the recipient's public key. This traditional two-step approach is called *signature-then-encryption*.

Signature generation and encryption consume machine cycles, and also introduce 'expanded' bits to an original message. Symmetrically, a comparable amount of computation time is generally required for signature verification and decryption. Hence, the cost of a cryptographic operation on a message is typically measured in the message expansion rate and the computational time invested by both the

sender and the recipient. With the standard *signature-then-encryption* approach, the cost for delivering a message in a secure and authenticated way is essentially the sum of the cost for digital signature and that for encryption.

In 1997 [29], Yuliang Zheng presented these concerns and raised an important question as to whether it is possible to transfer a message of arbitrary length in a *secure* and *authenticated* way with an expense less than that required by the *signature-then-encryption* approach. Zheng answered his question by proposing *signcryption* which simultaneously fulfills both the functions of digital signature and public key encryption in a logically single step, and with a cost *significantly* smaller than that required by *signature-then-encryption*. He defines a *signcryption scheme* as consisting of a pair of (polynomial time) algorithms (S, U) , where S is called the *signcryption algorithm* and U is called the *unsigncryption algorithm* (also most commonly known as *designcryption algorithm*). S in general is probabilistic, while U is most likely to be deterministic. (S, U) satisfy the following conditions.

1. **Unique Unsigncryptability.** Given a message m of arbitrary length, the algorithm S signcrypts m and outputs a signcrypted text c . On input c , the algorithm U unsigncrypts c and recovers the original message unambiguously.
2. **Security.** (S, U) fulfill, simultaneously, the properties of a secure encryption scheme and those of a secure digital signature scheme. These properties mainly include *confidentiality* of message contents, *unforgeability*, and *non-repudiation*.
3. **Efficiency.** The computational cost, which includes the computational time involved both in signcryption and unsigncryption, and the communication overhead or added redundant bits, of the scheme is smaller than that required by the best currently known *signature-then-encryption* scheme with comparable parameters.

Zheng's discovery went on to revolutionize the cryptographic research community and in a short span of a decade, *signcryption* has become an exploding research area. Since 1997, several efficient signcryption schemes have been proposed [2, 30, 14, 25, 22, 18, 28, 21]. The first example of formal security proof in a formal security model was published in 2002 [1]. However, none of these schemes were ID-based. Malone-Lee [20] proposed the first method that achieved ID-based signcryption. Libert and Quisquater [19] pointed out that [20] is not semantically secure because the signature of the message is visible in the signcrypted message. In [3], Barreto et al. constructed the most efficient ID-based signcryption scheme to date.

2.3. Broadcast Encryption

Amos Fiat and Moni Naor, in 1993 [13], analyzed the problem of a center broadcasting a message (e.g., a key to decipher a video clip) to a *dynamically changing* privileged subset of the users in such a way that non-members of the privileged class cannot learn the message, and proposed a solution which results in efficiency in both measures — transmission length and storage at the users end, without compromising the computational efficiency involved in carrying out the scheme. Their framework is called *broadcast encryption*. Apart from the normal security requirements of a two-party cryptosystem where there is one sender and one receiver, an additional property that is desired from any secure broadcast encryption scheme is *collusion resistance*. This means that even if all the non-privileged users collude in an attempt to learn the plaintext, they should not be able to do so.

Since its introduction by Fiat and Naor [13], the problem received significant attention, and many of its variants have been studied; many broadcast encryption systems have been proposed [23, 17, 16, 9, 12]. The best known fully collusion resistant systems are the schemes of Boneh, Gentry and Waters [9] which achieve $O(\sqrt{n})$ -size ciphertexts and public key; or, constant size ciphertexts, $O(n)$ -size public key and constant size private keys.

2.4. Bilinear Pairing

Let \mathbb{G}_1 be an additive cyclic group generated by P , with prime order q , and \mathbb{G}_2 be a multiplicative cyclic group of the same order q . A bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties.

- **Bilinearity.** For all $P, Q, R \in \mathbb{G}_1$,

$$\begin{aligned} - \hat{e}(P + Q, R) &= \hat{e}(P, R)\hat{e}(Q, R) \\ - \hat{e}(P, Q + R) &= \hat{e}(P, Q)\hat{e}(P, R) \\ - \hat{e}(aP, bQ) &= \hat{e}(P, Q)^{ab} \end{aligned}$$

- **Non-Degeneracy.** There exist $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq I_{\mathbb{G}_2}$, where $I_{\mathbb{G}_2}$ is the identity element of \mathbb{G}_2 .
- **Computability.** There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

2.5. Computational Assumptions

In this section, we review the computational assumptions related to bilinear maps that are relevant to the protocol we discuss.

2.5.1 Bilinear Diffie-Hellman Problem (BDHP)

Given $(P, aP, bP, cP) \in \mathbb{G}_1^4$ for unknown $a, b, c \in \mathbb{Z}_q^*$, the BDH problem in \mathbb{G}_1 is to compute $\hat{e}(P, P)^{abc}$.

Definition. The advantage of any probabilistic polynomial time algorithm \mathcal{A} in solving the BDH problem in \mathbb{G}_1 is defined as

$$Adv_{\mathcal{A}}^{BDH} = Pr[\mathcal{A}(P, aP, bP, cP) = \hat{e}(P, P)^{abc} \mid a, b, c \in \mathbb{Z}_q^*]$$

The *BDH Assumption* is that, for any probabilistic polynomial time algorithm \mathcal{A} , the advantage $Adv_{\mathcal{A}}^{BDH}$ is negligibly small.

2.5.2 Decisional Bilinear Diffie-Hellman Problem (DBDHP)

Given $(P, aP, bP, cP, \alpha) \in \mathbb{G}_1^4 \times \mathbb{G}_2$ for unknown $a, b, c \in \mathbb{Z}_q^*$, the DBDH problem in \mathbb{G}_1 is to decide if $\alpha = \hat{e}(P, P)^{abc}$.

Definition. The advantage of any probabilistic polynomial time algorithm \mathcal{A} in solving the DBDH problem in \mathbb{G}_1 is defined as

$$Adv_{\mathcal{A}}^{DBDH} = |Pr[\mathcal{A}(P, aP, bP, cP, \hat{e}(P, P)^{abc}) = 1] - Pr[\mathcal{A}(P, aP, bP, cP, \alpha) = 1]|$$

The *DBDH Assumption* is that, for any probabilistic polynomial time algorithm \mathcal{A} , the advantage $Adv_{\mathcal{A}}^{DBDH}$ is negligibly small.

2.5.3 Computational Diffie-Hellman Problem (CDHP)

Given $(P, aP, bP) \in \mathbb{G}_1^3$ for unknown $a, b \in \mathbb{Z}_q^*$, the CDH problem in \mathbb{G}_1 is to compute abP .

Definition. The advantage of any probabilistic polynomial time algorithm \mathcal{A} in solving the CDH problem in \mathbb{G}_1 is defined as

$$Adv_{\mathcal{A}}^{CDH} = Pr[\mathcal{A}(P, aP, bP) = abP \mid a, b \in \mathbb{Z}_q^*]$$

The *CDH Assumption* is that, for any probabilistic polynomial time algorithm \mathcal{A} , the advantage $Adv_{\mathcal{A}}^{CDH}$ is negligibly small.

2.5.4 Inverse - Computational Diffie-Hellman Problem (Inverse - CDH)

Given $(P, aP) \in \mathbb{G}_1^2$ for unknown $a \in \mathbb{Z}_q^*$, the Inverse-CDH problem in \mathbb{G}_1 is to compute $\frac{1}{a}P$.

Definition. The advantage of any probabilistic polynomial time algorithm \mathcal{A} in solving the Inverse-CDH problem in \mathbb{G}_1 is defined as

$$Adv_{\mathcal{A}}^{Inverse-CDH} = Pr\left[\mathcal{A}(P, aP) = \frac{1}{a}P \mid a \in \mathbb{Z}_q^*\right]$$

The *Inverse-CDH Assumption* is that, for any probabilistic polynomial time algorithm \mathcal{A} , the advantage $Adv_{\mathcal{A}}^{Inverse-CDH}$ is negligibly small.

2.6. Framework of ID-based Broadcast Signcryption (IBBSC)

We describe below, the framework for IBBSC that we use to describe Bohio et al.'s scheme and our fix. It is not a general framework but a tailored one to fit the present context and purpose. It consists of the following probabilistic polynomial time algorithms.

1. **Setup**(k). Given a security parameter k , the Private Key Generator (PKG) generates the public parameters $params$ and master secret key msk of the system.
2. **Keygen**(ID_A). Given an identity ID_A , the PKG, using the public parameters $params$ and the master secret key msk , computes the corresponding private key S_A and transmits it to A in a secure way.
3. **Signcrypt**(m, ID_A, \hat{S}_A). To send a message m to legal users, the broadcaster A with identity ID_A and secret value \hat{S}_A runs this algorithm to obtain the signcrypted ciphertext σ .
4. **Designcrypt**(σ, \hat{S}_A). A user who has the common secret value \hat{S}_A , on receiving the signcrypted ciphertext σ from broadcaster A , runs this algorithm to obtain either the plain text m or \perp according as whether σ was a valid signcryption by A or not.

For consistency, if $\sigma = \text{Signcrypt}(m, ID_A, \hat{S}_A)$, then $m = \text{Designcrypt}(\sigma, \hat{S}_A)$.

2.7. Security Model for ID-based Broadcast Signcryption

The two 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 signatures (IND-CCA2 and EUF-CMA respectively) to IBBSC below (keeping the framework that we discussed in the previous section in mind).

2.7.1 Indistinguishability under Adaptive Chosen Ciphertext Attack for IBBSC (IND-IBBSC-CCA2)

An ID-based broadcast signcryption scheme is semantically secure against adaptive chosen ciphertext attack (IND-IBBSC-CCA2) if no probabilistic polynomial time adversary \mathcal{A} has a non-negligible advantage in the following game.

1. The challenger \mathcal{C} runs $Setup(k)$ and sends the system public parameters $params$ to the adversary \mathcal{A} .
2. \mathcal{A} makes polynomially bounded number of queries to the following oracles.
 - (a) **Keygen Oracle** — \mathcal{A} produces an identity ID and queries for the secret key of ID . The *Keygen Oracle* returns S_{ID} to \mathcal{A} .
 - (b) **Signcrypt Oracle** — \mathcal{A} produces a message m , broadcaster identity ID_A . \mathcal{C} returns to \mathcal{A} , the signcrypted ciphertext as $\sigma = Signcrypt(m, ID_A, \hat{S}_A)$, where \hat{S}_A is the secret value.
 - (c) **Designcrypt Oracle** — \mathcal{A} produces a broadcaster identity ID_A , receiver identity ID_i and a signcryption σ . \mathcal{C} returns the result of $Designcrypt(\sigma, ID_A, ID_i, \hat{S}_A)$ to \mathcal{A} , where \hat{S}_A is the common secret value.
3. \mathcal{A} produces two messages m_0 and m_1 of equal length from the message space \mathcal{M} , and the target broadcaster identity ID_A . The adversary is not provided with the common secret value which is used for designcryption by all legal users. The challenger \mathcal{C} flips a coin, sampling a bit $b \leftarrow \{0, 1\}$ and obtains the challenge signcrypted ciphertext by running $Signcrypt(m_b, ID_A, \hat{S}_A)$, which is returned to \mathcal{A} .
4. \mathcal{A} is allowed to make polynomially bounded number of new queries as in Step 2 with the restrictions that it should not query the *Designcrypt Oracle* for the designcryption of σ^* .
5. Finally, \mathcal{A} outputs a bit b' and wins the game if $b' = b$. The advantage of the adversary in winning this game is defined as

$$Adv_{\mathcal{A}}^{IND-IBBSC-CCA2} = \left| Pr[b' = b] - \frac{1}{2} \right|$$

We mention that this model of security takes into account collusion resistance too, because we provide the adversary with the secret keys of every user of the system except the ones he attacks.

2.7.2 Existential Unforgeability under Adaptive Chosen Message Attack for IBBSC (EUF-IBBSC-CMA)

An ID-based broadcast signcryption scheme is existentially unforgeable under adaptive chosen message attack (EUF-IBBSC-CMA) if no probabilistic polynomial time adversary \mathcal{A} has a non-negligible advantage in the following game.

1. The challenger \mathcal{C} runs $Setup(k)$ and sends the system public parameters $params$ to the adversary \mathcal{A} .
2. This step is identical to that in the indistinguishability game of the previous section.
3. \mathcal{A} produces a signcrypted ciphertext σ for the target broadcaster ID_A and wins the game if the secret value of broadcaster ID_A was not queried and \perp is not returned by $Designcrypt(\sigma, ID_i, \hat{S}_A)$ for any legal user ID_i who has \hat{S}_A and σ is not the output of a previous query to the *Signcrypt Oracle*. Under these conditions, the advantage of the adversary in winning this game is defined as $Adv_{\mathcal{A}}^{EUF-IBBSC-CMA} = Pr \left[Designcrypt(\sigma, ID_i, \hat{S}_A) \neq \perp \right]$

We mention that this model of security takes into account collusion resistance too, because we allow the adversary to query for the secret keys of any entity.

3. Overview of IBBSC Scheme of Bohio et al.

Bohio et al.'s IBBSC scheme [4] is an unconventional scheme, in the following sense. Neither does the signcryption involve the secret key of the broadcaster, nor does the designcryption involve the secret key of the receiver. Both signcryption and designcryption involve some other common secret, which is established beforehand using the secret keys. Once this common secret (called the broadcast parameter below) is established, the secret keys are not used at all. But this has the advantage of providing constant ciphertext size. The scheme consists of the three algorithms *INITIALIZE* (which includes *Setup* and *Keygen*), *SIGNCRYPT* and *DESIGNCRYPT*, which we describe below.

INITIALIZE — The steps in the setup phase are given below.

1. The security parameter of the scheme is k . The trusted authority chooses two groups \mathbb{G}_1 and \mathbb{G}_2 of prime order q where $|q| = k$, a generator P of \mathbb{G}_1 , a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and hash functions $H_0 : \{0, 1\}^{k_2} \rightarrow \{0, 1\}^{k_1}$, $H_1 : \{0, 1\}^{k_0} \rightarrow \mathbb{G}_1$, $H_2 : \{0, 1\}^{k_2} \rightarrow \{0, 1\}^{k_3}$ and $H_3 : \{0, 1\}^{k_3} \rightarrow \mathbb{Z}_q^*$,

where k_0, k_1, k_2 and k_3 are the number of bits required to represent an identity, a \mathbb{G}_1 element, a \mathbb{G}_2 element, and a message respectively. The master private key is $s \in_R \mathbb{Z}_q^*$ and the master public key is $P_{pub} = sP$. The public parameters of this scheme are $(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_0, H_1, H_2, H_3)$.

2. The public key of a broadcaster B with identity ID_B is $Q_B = H_1(ID_B)$ and the corresponding private key is $S_B = sQ_B$.
3. The public key of a user i with identity ID_i is $Q_i = H_1(ID_i)$ and the corresponding private key is $S_i = sQ_i$.
4. The steps followed by broadcaster B are as follows.
 - (a) Store a precomputed value $\omega_B = \hat{e}(Q_B, P)$ to be used during signcryption.
 - (b) Select randomly a value $x_B \in \mathbb{Z}_q^*$ to be the *broadcast secret* and compute the *broadcast parameter* as $x_B Q_B$. When a subscriber (user) i joins the broadcaster, send the broadcast parameter, encrypted with the key $H_0(\hat{e}(S_B, Q_i))$ using the one time pad. The user can compute the key as $H_0(\hat{e}(S_i, Q_B))$ and recover the broadcast parameter. Note that no secure channel need be used during user joining.

SIGNCRYPT — In order to signcrypt a message m , the broadcaster B will do the following.

1. Compute $h = H_3(m)$.
2. Choose $r \in_R \mathbb{Z}_q^*$ and compute the session key as $d = H_2(\omega_B^{(r+h)})$.
3. Compute the ciphertext $c = m \oplus d$.
4. Compute $U = rP$ and $V = x_B^{-1}(r+h)P$.
5. Broadcast (c, U, V) to all the users.

DESIGNCRYPT — For the designcryption of the message, the authorized receivers (those provided with the broadcast parameter $x_B Q_B$) will do the following.

1. Compute the key d' by performing the following computations.

- (a) Compute the quantity ω' as

$$\begin{aligned} \omega' &= \hat{e}(x_B Q_B, V) \\ &= \hat{e}(x_B Q_B, x_B^{-1}(r+h)P) \\ &= \hat{e}(Q_B, P)^{(r+h)} \\ &= \omega_B^{(r+h)} \end{aligned}$$

- (b) $d' = H_2(\omega')$

2. The message m is then decrypted as $m = c \oplus d'$.
3. The authentication is provided by computing $h' = H_3(m)$ and verifying whether $\hat{e}(Q_B, U + h'P) \stackrel{?}{=} \omega'$.

4. Attacks on IBBS Scheme of Bohio et al.

Bohio et al. claimed that their scheme provides both confidentiality and unforgeability (authentication), but they do not give any formal proof to support their claims. We show in this section the following two attacks.

4.1. Attack on Authentication

Here, we demonstrate that their scheme is universally forgeable — any legitimate user can generate a valid ciphertext for any message m^* as if it were generated by the broadcaster. For a ciphertext to be valid, it should satisfy $\hat{e}(Q_B, U + h'P) = \omega'$ where ω' is obtained using V . Also, the message hash h' and U are independent, and hence it is easy to play with the parameters to mount a forgeability attack. We describe how this attack proceeds in this section.

To forge the ciphertext of B on a message m^* of his choice, a legitimate user simply does the following.

1. Compute the hash of the message $h^* = H_3(m^*)$.
2. Choose a value $r^* \in_R \mathbb{Z}_q^*$ and compute the following.
 - (a) $V^* = r^* Q_B$
 - (b) $U^* = r^* x_B Q_B - h^* P$
 - (c) $\omega^* = \hat{e}(Q_B, x_B Q_B)^{r^*}$
 - (d) $d^* = H_2(\omega^*)$
 - (e) $c^* = m^* \oplus d^*$
3. The malicious user then broadcasts the values (c^*, U^*, V^*) .

Now, we prove below that (c^*, U^*, V^*) is indeed a valid signcryption from broadcaster B on the message m^* .

The **DESIGNCRYPT** algorithm will do the following.

1. The key computation would proceed as follows.

- (a) The quantity ω' would be computed as

$$\begin{aligned} \omega' &= \hat{e}(x_B Q_B, V^*) \\ &= \hat{e}(x_B Q_B, r^* Q_B) \\ &= \hat{e}(x_B Q_B, Q_B)^{r^*} \end{aligned}$$

- (b) $d' = H_2(\omega')$

2. The message m' is then decrypted as $m' = c^* \oplus d'$.
3. The authentication is provided by computing $h' = H_3(m')$ and verifying whether $\hat{e}(Q_B, U^* + h'P) \stackrel{?}{=} \omega'$. This is satisfied because,

$$\begin{aligned}\hat{e}(Q_B, U^* + h'P) &= \hat{e}(Q_B, r^*x_BQ_B - h'P + h'P) \\ &= \hat{e}(Q_B, r^*x_BQ_B) \\ &= \omega'\end{aligned}$$

From this, it is clear that a malicious user can forge the signcryption of his broadcaster on any message of his choice.

4.2. Attack on Confidentiality

Here, we demonstrate that Bohio et al.'s scheme is not IND-CCA secure. Specifically, in the IND-CCA game, during the challenge phase, when the adversary gives two messages m_0 and m_1 of his choice to the challenger and the challenger randomly signcrypts one of them and returns it, the adversary can find out whether the challenge ciphertext is that of m_0 or m_1 . For an adversary to distinguish which message was encrypted, it is enough if he can find out which symmetric key was used for encryption. Since ω can be computed easily as $\hat{e}(Q_B, U + hP)$ where h is either $H_3(m_0)$ or $H_3(m_1)$. The attack proceeds as follows.

On receiving the challenge signcrypted ciphertext (c, U, V) , the adversary does the following.

1. Compute $h_0 = H_3(m_0)$ and $h_1 = H_3(m_1)$.
2. Compute the quantities $\omega_0 = \hat{e}(U + h_0P, Q_B)$ and $\omega_1 = \hat{e}(U + h_1P, Q_B)$.
3. Check if $c \stackrel{?}{=} H_2(\omega_0) \oplus m_0$ then return $b = 0$, otherwise, if $c \stackrel{?}{=} H_2(\omega_1) \oplus m_1$ then return $b = 1$, otherwise abort.

The adversary is sure to succeed, because of the inherent construction of the scheme — the verification method reveals the key used for encrypting the message. Hence the scheme is not IND-CCA secure.

5. Our Fix for Bohio et al.'s Scheme

We now describe the fix for Bohio's scheme. The *INITIALIZE* algorithm is the same as that of Bohio et al.'s (described in Section 3) except for a modification in the hash functions H_2 and H_3 as follows.

$$H_2 : \{0, 1\}^{k_2} \rightarrow \{0, 1\}^{k_1+k_3}$$

$$H_3 : \{0, 1\}^{k_3} \times \{0, 1\}^{k_1} \rightarrow \mathbb{Z}_q^*$$

The *SIGNCRYPT* and *DESIGNCRYPT* algorithms are described below.

SIGNCRYPT — In order to signcrypt a message m , broadcaster B will do the following.

1. Choose $r \in_R \mathbb{Z}_q^*$ and compute $U = rP$.
2. Compute $h = H_3(m, U)$.
3. Compute the session key as $d = H_2(\omega_B^{(r+h)})$.
4. Compute the ciphertext $c = (m||U) \oplus d$.
5. Compute $V = x_B^{-1}(r + h)P$.
6. Broadcast (c, V) to all the users.

DESIGNCRYPT — For the designcryption of the message, the authorized receivers (those provided with the broadcast parameter x_BQ_B) will do the following.

1. Compute the key d by performing the following computations.
 - (a) Compute the quantity ω' as
$$\begin{aligned}\omega' &= \hat{e}(x_BQ_B, V) \\ &= \hat{e}(x_BQ_B, x_B^{-1}(r + h)P) \\ &= \hat{e}(Q_B, P)^{(r+h)} \\ &= \omega_B^{(r+h)}\end{aligned}$$
 - (b) $d' = H_2(\omega')$
2. The message m is retrieved from $m||U$ after decrypting c as $m||U = c \oplus d'$.
3. The authentication is provided by computing $h' = H_3(m, U)$ and verifying whether $\hat{e}(Q_B, U + h'P) \stackrel{?}{=} \omega'$.

6. Correctness of our IBBS Scheme

In this section, we proceed to prove that our proposed scheme is indeed consistent and correct. If $\sigma = (c, V)$ is a valid signcrypted ciphertext from broadcaster B to his privileged subscribers then the *DESIGNCRYPT* algorithm will do the following.

1. Compute the quantity ω' as
$$\begin{aligned}\omega' &= \hat{e}(x_BQ_B, V) \\ &= \hat{e}(x_BQ_B, x_B^{-1}(r + h)P) \\ &= \hat{e}(Q_B, P)^{(r+h)}\end{aligned}$$

2. Compute $d' = H_2(\omega')$.
3. Retrieve the message m from $m\|U$ by decrypting $m\|U = c \oplus d'$.
4. Compute $h = H_3(m, U)$.
5. The check, $\hat{e}(Q_B, U + hP) \stackrel{?}{=} \omega'$, succeeds because,

$$\begin{aligned}
\hat{e}(Q_B, U + hP) &= \hat{e}(Q_B, rP + hP) \\
&= \hat{e}(Q_B, (r + h)P) \\
&= \hat{e}(Q_B, P)^{(r+h)} \\
&= \omega'
\end{aligned}$$

7. Proof of Unforgeability of our IBBSC Scheme

Theorem 7.1 *Our ID-based broadcast signcryption scheme is secure against any EUF-IBBSC-CMA adversary \mathcal{A} under the random oracle model if Inverse-CDHP is hard in \mathbb{G}_1 .*

Proof. The challenger \mathcal{C} receives an instance (P, aP) of the Inverse-CDH problem. His goal is to determine the value of $\frac{1}{a}P$. Suppose there exists an EUF-IBBSC-CMA adversary \mathcal{A} for our improved scheme. We show that \mathcal{C} can use \mathcal{A} to solve the Inverse-CDH problem. \mathcal{C} will set the random oracles¹ \mathcal{O}_{H_1} , \mathcal{O}_{H_2} , \mathcal{O}_{H_3} , $\mathcal{O}_{KeyExtract}$, $\mathcal{O}_{Signcrypt}$ and $\mathcal{O}_{Designcrypt}$. The answers to the oracles \mathcal{O}_{H_1} , \mathcal{O}_{H_2} , and \mathcal{O}_{H_3} are randomly selected; therefore, to maintain consistency, \mathcal{C} will maintain three lists $L_1 = \langle ID, x_{ID}, S_{ID}, Q_{ID} \rangle$, $L_2 = \langle \omega, h_2 \rangle$, and $L_3 = \langle m, U, h_3 \rangle$. The reasons for and meanings of the elements of these lists will become clear during the discussion of the corresponding oracles. We assume that \mathcal{A} will ask for $H_1(ID)$ before ID is used in any key extraction, signcryption and designcryption queries. First, the adversary \mathcal{A} outputs the identity of the broadcaster whose signcryption he claims to forge. Without loss of generality, let it be ID_B . The challenger \mathcal{C} gives \mathcal{A} the system parameters $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3 \rangle$, where $P_{pub} = sP$ for some $s \in_R \mathbb{Z}_q^*$. The broadcaster secret $x_B Q_B$ is set as aQ_B ². During the query stage, \mathcal{A} can query the following oracles.

Oracle $\mathcal{O}_{H_1}(ID)$. \mathcal{C} checks if there exists a tuple $(ID, \hat{x}_{ID}, S_{ID}, Q_{ID})$ in L_1 . If such a tuple exists, \mathcal{C} answers with Q_{ID} . Otherwise, \mathcal{C} does the following.

¹ $x_B Q_B$ is given as part of the private information on using *KeyExtract* oracle. So there is no need for separate oracle for H_0 . $x_B Q_B$ is known to all legal users subscribed to broadcaster ID_B , hence it is also given to \mathcal{A} .

²Note that $Q_B = x_{ID_B} P$. Therefore aQ_B can be computed as $x_{ID_B} aP$.

1. Choose a new $x_{ID} \in_R \mathbb{Z}_q^*$, and set $Q_{ID} = x_{ID} P$, $S_{ID} = x_{ID} sP$.
2. Add $(ID, x_{ID}, S_{ID}, Q_{ID})$ to the list L_1 and return Q_{ID} .

Oracle $\mathcal{O}_{H_2}(\omega)$. \mathcal{C} checks if there exists a tuple (ω, h_2) in L_2 . If such a tuple exists, \mathcal{C} returns h_2 . Otherwise, \mathcal{C} chooses a new $h_2 \in_R \{0, 1\}^{k_1+k_3}$, adds the tuple (ω, h_2) to L_2 and returns h_2 .

Oracle $\mathcal{O}_{H_3}(m, U)$. \mathcal{C} checks if there exists a tuple (m, U, h_3) in L_3 . If such a tuple exists, \mathcal{C} returns h_3 . Otherwise, \mathcal{C} chooses a new $h_3 \in_R \mathbb{Z}_q^*$, adds the tuple (m, U, h_3) to L_3 and returns h_3 .

Oracle $\mathcal{O}_{KeyExtract}(ID)$. If L_1 does not contain an entry for ID , return \perp . Otherwise, \mathcal{C} recovers the tuple $(ID, x_{ID}, S_{ID}, Q_{ID})$ from L_1 and returns (S_{ID}, aQ_B) .

Oracle $\mathcal{O}_{Signcrypt}(m, ID_B)$. On receiving this query, \mathcal{C} checks if there is an entry for ID_B in L_1 not, then \mathcal{C} aborts. Otherwise, \mathcal{C} retrieves the tuple (ID_B, x_B, S_B, Q_B) from L_1 . It chooses $r \in \mathbb{Z}_q^*$ and new $h_2 \in \{0, 1\}^{k_1+k_3}$, $h_3 \in \mathbb{Z}_q^*$ randomly and does the following.

1. Compute $\omega = \hat{e}(aP, Q_B)^r$ and add (ω, h_2) to L_2 .
2. Compute $U = raP - h_3P$ and add (m, U, h_3) to L_3 .
3. Compute $c = m\|U \oplus h_2$.
4. Set $V = rP$.
5. Broadcast the signcrypted ciphertext $\sigma = (c, V)$.

Oracle $\mathcal{O}_{Designcrypt}(\sigma)$. On receiving the signcryption $\sigma = (c, V)$, \mathcal{C} executes the *DESIGNCRYPT* algorithm on σ in the normal way and returns what the designcryption algorithm returns.

Eventually \mathcal{A} outputs a forged signcryption $\sigma^* = (y^*, V^*)$ on some message m^* from the broadcaster B to his subscribers. Challenger \mathcal{C} executes the *DESIGNCRYPT* algorithm on σ^* . If σ^* is a valid signcryption from the broadcaster B to his subscribers, i.e., a message m^* is returned by the decryption algorithm, then \mathcal{C} applies the oracle replay technique³ to produce two valid signcryptions $\sigma' = (c', V')$ and $\sigma'' = (c'', V'')$ on some arbitrary message m from the broadcaster B to all his subscribers. Now we can apply standard arguments for the outputs of the forking lemma since both V' and V'' are valid signatures for the same

³We use the oracle replay technique as described and employed by Pointcheval et al. in [24].

message m and same random tape of the adversary. Finally, \mathcal{C} obtains the solution to the Inverse-CDH instance as $(h'_3 - h''_3)^{-1}(V' - V'')$.⁴ We have,

$$\begin{aligned} (h'_3 - h''_3)^{-1}(V' - V'') &= (h'_3 - h''_3)^{-1}(h'_3 - h''_3) \frac{1}{a} P \\ &= \frac{1}{a} P \end{aligned}$$

So, we can see that the challenger \mathcal{C} has the same advantage in solving the Inverse-CDH problem as the adversary \mathcal{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 Inverse-CDH problem with non-negligible advantage. Since this is not possible, it is infeasible for an adversary to forge a valid signcryption. Hence, our proposed scheme is secure against any EUF-IBBSC-CMA attack. \square

8. Proof of Confidentiality of our IBBSC Scheme

Theorem 8.1 *Our improved ID-based broadcast signcryption scheme is secure against any IND-IBBSC-CCA2 adversary \mathcal{A} under the random oracle model if DBDHP is hard in \mathbb{G}_1 .*

Proof. The challenger \mathcal{C} receives (P, aP, bP, cP, α) , an instance of the DBDH problem. His goal is to determine if $\hat{e}(P, P)^{abc} \stackrel{?}{=} \alpha$. Suppose there exists an IND-IBBSC-CCA2 adversary \mathcal{A} for our improved scheme. We show that \mathcal{C} can use \mathcal{A} to solve the DBDH problem. \mathcal{C} will set the random oracles $\mathcal{O}_{H_1}, \mathcal{O}_{H_2}, \mathcal{O}_{H_3}, \mathcal{O}_{KeyExtract}, \mathcal{O}_{Signcrypt}$ and $\mathcal{O}_{Designcrypt}$. The answers to the oracles $\mathcal{O}_{H_1}, \mathcal{O}_{H_2}$, and \mathcal{O}_{H_3} are randomly selected; therefore, to maintain consistency, \mathcal{C} will maintain five lists $L_0 = \langle \omega, h_0 \rangle$, $L_1 = \langle ID, x_{ID}, S_{ID}, Q_{ID} \rangle$, $L_2 = \langle \omega, h_2 \rangle$, $L_3 = \langle m, U, h_3 \rangle$, and $L_s = \langle \sigma, m \rangle$. The reasons for and meanings of the elements of these lists will become clear during the discussion of the corresponding oracles. We assume that \mathcal{A} will ask for $H_1(ID)$ before ID is used in any key extraction, signcryption and designcryption queries. First, the adversary \mathcal{A} outputs the identity of the broadcaster whose subscribers he plans to attack. Without loss of generality, let it be ID_B . The challenger \mathcal{C} gives \mathcal{A} the system parameters $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3 \rangle$, where $P_{pub} = sP$ for some $s \in_R \mathbb{Z}_q^*$. The broadcaster secret $x_B Q_B$ is set as aQ_B .⁵ During the query stage, \mathcal{A} can query the following oracles.

⁴Note that, h'_3 and h''_3 can be obtained from L_3 as \mathcal{O}_{H_3} must be used during designcryption with the input (m, U) .

⁵Note that $Q_B = bP$. Therefore $aQ_B = abP$ which is also unknown to \mathcal{C} .

Oracle $\mathcal{O}_{H_0}(\omega)$. \mathcal{C} checks if there exists a tuple (ω, h_0) in L_0 . If such a tuple exists, \mathcal{C} returns h_0 . Otherwise, \mathcal{C} chooses a new $h_0 \in_R \{0, 1\}^{k_1}$, adds the tuple (ω, h_0) to L_0 and returns h_0 .

Oracle $\mathcal{O}_{H_1}(ID)$. \mathcal{C} checks if there exists a tuple $(ID, x_{ID}, S_{ID}, Q_{ID})$ in L_1 . If such a tuple exists, \mathcal{C} answers with Q_{ID} . Otherwise, \mathcal{C} does the following. If $ID = ID_B$, then set $Q_{ID_B} = bP$, add $(ID_B, \perp, S_{ID_B}, Q_{ID_B})$ to L_1 and return Q_{ID_B} . Else, do the following.

1. Choose a new $x_{ID} \in_R \mathbb{Z}_q^*$, and set $Q_{ID} = x_{ID}P$, $S_{ID} = x_{ID}sP$.
2. Add $(ID, x_{ID}, S_{ID}, Q_{ID})$ to the list L_1 and return Q_{ID} .

Oracle $\mathcal{O}_{H_2}(\omega)$. \mathcal{C} checks if there exists a tuple (ω, h_2) in L_2 . If such a tuple exists, \mathcal{C} returns h_2 . Otherwise, \mathcal{C} chooses a new $h_2 \in_R \{0, 1\}^{k_1+k_3}$, adds the tuple (ω, h_2) to L_2 and returns h_2 .

Oracle $\mathcal{O}_{H_3}(m, U)$. \mathcal{C} checks if there exists a tuple (m, U, h_3) in L_3 . If such a tuple exists, \mathcal{C} returns h_3 . Otherwise, \mathcal{C} chooses a new $h_3 \in_R \mathbb{Z}_q^*$, adds the tuple (m, U, h_3) to L_3 and returns h_3 .

Oracle $\mathcal{O}_{KeyExtract}(ID)$. If L_1 does not contain an entry for ID , return \perp . Otherwise, \mathcal{C} recovers the tuple $(ID, x_{ID}, S_{ID}, Q_{ID})$ from L_1 and returns S_{ID} .

Oracle $\mathcal{O}_{Signcrypt}(m, ID_B)$. On receiving this query, \mathcal{C} checks if there is an entry for ID_B in L_1 not, then \mathcal{C} aborts. Otherwise, \mathcal{C} retrieves the tuple (ID_B, x_B, S_B, Q_B) from L_1 . It chooses $r \in \mathbb{Z}_q^*$ and new $h_2 \in \{0, 1\}^{k_1+k_3}$, $h_3 \in \mathbb{Z}_q^*$ randomly and does the following.

1. Compute $\omega = \hat{e}(aP, Q_B)^r$ and add (ω, h_2) to L_2 .
2. Compute $U = raP - h_3P$ and add (m, U, h_3) to L_3 .
3. Compute $c = m \| U \oplus h_2$.
4. Set $V = rP$.
5. Add (σ, m) to L_s and broadcast the signcrypted ciphertext $\sigma = (c, V)$.

Oracle $\mathcal{O}_{Designcrypt}(\sigma)$. Note that the challenger does not know the broadcast parameter $x_B Q_B = abP$ and hence cannot do a normal designcryption. So, he does the following (assume $\sigma = (c, V)$).

1. Traverse through the list L_s . If there exists an entry (σ, m) , then return m . Otherwise, continue to the next step.
2. Traverse through the list L_3 . For each entry (m, U, h_3) , do the following.

- (a) Compute $h_2 = (m||U) \oplus c$.
 - (b) Traverse through the list L_2 . If there exists an entry of the form (ω, h_2) , then continue to the next step. Else, continue traversing the list L_3 (Go back to Step 2).
 - (c) Check if $\omega \stackrel{?}{=} \hat{e}(Q_B + U + h_3P)$. If so, then return m . Else, continue scanning the list L_2 for matching entries.
3. If the list L_3 is exhausted without any match, then return \perp .

After this first query stage, \mathcal{A} outputs two plaintext messages m_0 and m_1 of equal length. Now, \mathcal{C} chooses a random bit $b \in \{0, 1\}$, retrieves the tuple (ID_B, x_B, S_B, Q_B) from L_1 , and signcrypts the message m_b as follows.

1. Choose a new random number $h_3 \in \mathbb{Z}_q^*$
2. Compute $U = cP$ and $V = cP + h_3P$
3. Compute $\omega = \alpha \cdot \hat{e}(aP, bP)^{h_3}$
4. Compute $h_2 = H_2(\omega)$ using his own oracle as $\mathcal{O}_{H_2}(\omega)$.
5. Compute $y = (m||U) \oplus h_2$ and add $\langle m, U, h_3 \rangle$ to L_2 .

\mathcal{C} returns $\sigma = \langle y, V \rangle$ as the challenge signcryptured ciphertext.

\mathcal{A} is now allowed to perform at most q_s queries (which may include queries to any of the oracles) as above. However, it cannot query the designcrypton oracle with the challenge signcrypton. Finally the adversary outputs the guess b' of the bit b . If the adversary has to win the game with non-negligible advantage, then he must have decrypted the challenge signcryptured ciphertext for which, he should have queried \mathcal{O}_{H_2} with $\hat{e}(x_B Q_B, V)$ which is nothing but $\hat{e}(abP, (c + h_3)P) = \hat{e}(P, P)^{abc+abh_3}$. Since L_2 records the queries of the adversary to \mathcal{O}_{H_2} in the form of $\langle \omega, h_2 \rangle$, within q_s queries, we can decide if $\alpha \stackrel{?}{=} \hat{e}(P, P)^{abc}$ by checking if $\omega = \alpha \cdot \hat{e}(aP, h_3bP)$ every time the adversary queries \mathcal{O}_{H_2} . The challenger aborts the adversary as soon as the DBDH problem is solved.

So, if there exists a non-trivial adversary who can defeat the signcrypton by learning something about the encrypted message, that means there exists an algorithm to solve the DBDH problem with non-negligible advantage. Since this is not possible, no adversary can defeat the signcrypton this way. Hence, our proposed scheme is secure against any IND-IBBSC-CCA2 attack. \square

9. Efficiency of our IBBSC Scheme

In this section, we discuss the efficiency of our improved IBBSC scheme. The major parameters involved are the computation costs for *signcrypton* and *designcrypton* operations, the communication cost and the storage at the user's end. For computational cost, we consider the number of pairing computations performed, as they are the costliest operations involved. Our improved IBBSC scheme performs no pairing operation during *Signcrypt* (except during setup of ω_B) and two pairing operations per user for *Designcrypt*, which is the same as that of Bohio et al.'s scheme. For the communication cost, we have to broadcast only a tuple of two elements compared to three in Bohio et al.'s scheme. Coming to storage cost, we consider the storage at the broadcaster's end and storage at the user's end. The storage cost for the broadcaster and a user is only $O(1)$ as they do not have to store anything other than their secret key and precomputed secret. Thus, our improved scheme does not compromise any of the efficiency properties of Bohio et al.'s scheme. The added advantage here is that the size of signcryptured ciphertext has been reduced to two elements.

10. Conclusion

In this paper, we have considered the problem of secure and authenticated content distribution over large networks, especially wireless networks, which, on one hand, are increasingly becoming popular choices for the modern civilization, what with the advent of mobile and portable devices such as cell phones and PDAs, and on the other hand, are much easier to eavesdrop than wired networks. IBBSC schemes provide the best solution to this problem. First, we have demonstrated a break of Bohio et al.'s IBBSC scheme, both in terms of authentication and confidentiality. Following this, we have proposed a fixed version of the scheme to prevent forgeability and we have also 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 fixed Bohio et al.'s IBBSC scheme, we have not compromised the performance of their scheme. In fact, we reduce the size of the signcryptured ciphertext to two elements.

Future Work. While Bohio et al.'s scheme (and hence our fixed version) offers the indisputable advantage of constant size ciphertext, it suffers from a major drawback. Suppose a user is to be revoked or a user leaves the group. Then the whole scheme comes to a standstill. Because now the broadcast parameter has to be changed and sent to every remaining user of the group. So, it becomes important for the research community to investigate for conventional IBBSC schemes that also achieve constant size ciphertexts.

References

- [1] J. Baek, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption. In *PKC 2002: Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 80–98. Springer-Verlag, 2002.
- [2] F. Bao and R. H. Deng. A signcryption scheme with signature directly verifiable by public key. In *PKC '98: Proceedings of the 1st International Workshop on Practice and Theory in Public Key Cryptography*, volume 1431 of *Lecture Notes in Computer Science*, pages 55–59. Springer-Verlag, 1998.
- [3] P. S. L. M. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *Advances in Cryptology - ASIACRYPT 2005: Proceedings of the 11th International Conference on the Theory and Application of Cryptology and Information Security*, volume 3788 of *Lecture Notes in Computer Science*, pages 515–532. Springer-Verlag, 2005.
- [4] M. J. Bohio and A. Miri. An authenticated broadcasting scheme for wireless ad hoc network. In *2nd Annual Conference on Communication Networks and Services Research (CNSR)*, pages 69–74, 2004.
- [5] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2004: Proceedings of the 23rd International Conference on the Theory and Application of Cryptographic Techniques*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer-Verlag, 2004.
- [6] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology - CRYPTO 2004: Proceedings of the 24th Annual International Cryptology Conference*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer-Verlag, 2004.
- [7] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology - EUROCRYPT 2005: Proceedings of the 24th International Conference on the Theory and Application of Cryptographic Techniques*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer-Verlag, 2005.
- [8] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 2001: Proceedings of the 21st Annual International Cryptology Conference*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, 2001.
- [9] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Advances in Cryptology - CRYPTO 2005: Proceedings of the 25th Annual International Cryptology Conference*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer-Verlag, 2005.
- [10] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2004: Proceedings of the 23rd International Conference on the Theory and Application of Cryptographic Techniques*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer-Verlag, 2004.
- [11] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA 2001: Proceedings of the 8th International Conference in Cryptography and Coding*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer-Verlag, 2001.
- [12] C. Delerablée, P. Paillier, and D. Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In *PAIRING 2007: Proceedings of the 1st International Conference on Pairing-Based Cryptography*, volume 4575 of *Lecture Notes in Computer Science*, pages 39–59. Springer-Verlag, 2007.
- [13] A. Fiat and M. Naor. Broadcast encryption. In *Advances in Cryptology - CRYPTO '93: Proceedings of the 13th Annual International Cryptology Conference*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer-Verlag, 1994.
- [14] C. Gamage, J. Leiwo, and Y. Zheng. Encrypted message authentication by firewalls. In *PKC 1999: Proceedings of the 2nd International Workshop on Practice and Theory in Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 69–81. Springer-Verlag, 1999.
- [15] C. Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2006: Proceedings of the 25th International Conference on the Theory and Application of Cryptographic Techniques*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer-Verlag, 2006.
- [16] M. T. Goodrich, J. Z. Sun, and R. Tamassia. Efficient tree-based revocation in groups of low-state devices. In *Advances in Cryptology - CRYPTO 2004: Proceedings of the 24th Annual International Cryptology Conference*, volume 3152 of *Lecture Notes in Computer Science*, pages 511–527. Springer-Verlag, 2004.
- [17] D. Halevy and A. Shamir. The lsd broadcast encryption scheme. In *Advances in Cryptology - CRYPTO 2002: Proceedings of the 22nd Annual International Cryptology Conference*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60. Springer-Verlag, 2002.
- [18] H. Y. Jung, K. S. Chang, D. H. Lee, and J. I. Lim. Signcryption schemes with forward secrecy. In *WISA 2001: Proceedings of the 2nd International Workshop on Information Security Applications*, pages 403–475, 2001.
- [19] B. Libert and J.-J. Quisquater. A new identity based signcryption scheme from pairings. *Proceedings of the IEEE Information Theory Workshop*, pages 155–158, 2003.
- [20] J. Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002.
- [21] J. Malone-Lee and W. Mao. Two birds one stone: Signcryption using rsa. In *Topics in Cryptology - CT-RSA 2003: Proceedings of the Cryptographers' Track at the RSA Conference 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 211–225. Springer-Verlag, 2003.
- [22] Y. Mu and V. Varadharajan. Distributed signcryption. In *Progress in Cryptology - INDOCRYPT 2000: Proceedings of the 1st International Conference in Cryptology in India*,

- volume 1977 of *Lecture Notes in Computer Science*, pages 155–164. Springer-Verlag, 2000.
- [23] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Advances in Cryptology - CRYPTO 2001: Proceedings of the 21st Annual International Cryptology Conference*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer-Verlag, 2001.
- [24] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
- [25] M. Seo and K. Kim. Electronic funds transfer protocol using domain-verifiable signcryption scheme. In *ICISC '99: Proceedings of the 2nd International Conference on Information Security and Cryptology*, volume 1787 of *Lecture Notes in Computer Science*, pages 269–277. Springer-Verlag, 1999.
- [26] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84: Proceedings of the 4th Annual International Cryptology Conference*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.
- [27] B. Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2005: Proceedings of the 24th International Conference on the Theory and Application of Cryptographic Techniques*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer-Verlag, 2005.
- [28] D. H. Yum and P. J. Lee. New signcryption schemes based on kcdsa. In *ICISC 2001: Proceedings of the 4th International Conference on Information Security and Cryptology*, volume 2285 of *Lecture Notes in Computer Science*, pages 305–317. Springer-Verlag, 2001.
- [29] Y. Zheng. Digital signcryption or how to achieve $\text{cost}(\text{signature} \ \& \ \text{encryption}) \ll \text{cost}(\text{signature}) + \text{cost}(\text{encryption})$. In *Advances in Cryptology - CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 1997.
- [30] Y. Zheng and H. Imai. How to construct efficient signcryption schemes on elliptic curves. *Information Processing Letters*, 68(5):227–233, December 1998.