1-1-2014

Anonymous broadcast encryption with an untrusted gateway

Hui Cui
University of Wollongong, hc892@uowmail.edu.au

Yi Mu
University of Wollongong, ymu@uow.edu.au

Man Ho Au
University of Wollongong, aau@uow.edu.au

Anonymous broadcast encryption with an untrusted gateway

Abstract

We propose a verifiable and anonymous broadcast encryption scheme, where an 'untrusted' gateway can verify incoming communication flows to ensure only the intended anonymous receivers in the target domain can receive them. This scenario is interesting while the privacy of receivers should be considered. The difficulty in this setting is how to achieve both confidentiality of the message and anonymity of receivers during the gateway verification. To achieve this goal, we introduce a new notion of encrypted identity search, which allows the gateway blindly verifies the incoming traffic. Our scheme captures security properties: confidentiality and anonymity against dishonest gateway, corrupted receivers and collusion attacks. We present a concrete construction of gateway-based verifiable and anonymous broadcast encryption system from bilinear pairings, and give its security reduction under the computational assumptions related to bilinear pairings.

Keywords

encryption, untrusted, broadcast, gateway, anonymous

Publication Details

Cui, H., Mu, Y. & Au, M. (2014). Anonymous broadcast encryption with an untrusted gateway. International Journal of Security and Networks, 9 (1), 20-29.

Anonymous broadcast encryption with an untrusted gateway

Hui Cui*, Yi Mu and Man Ho Au

School of Computer Science and Software Engineering,
University of Wollongong,
Wollongong, NSW2522, Australia
E-mail: hc892@uowmail.edu.au
E-mail: ymu@uow.edu.au
E-mail: aau@uow.edu.au
*Corresponding author

Abstract: We propose a verifiable and anonymous broadcast encryption scheme, where an 'untrusted' gateway can verify incoming communication flows to ensure only the intended (anonymous) receivers in the target domain can receive them. This scenario is interesting while the privacy of receivers should be considered. The difficulty in this setting is how to achieve both confidentiality of the message and anonymity of receivers during the gateway verification. To achieve this goal, we introduce a new notion of encrypted identity search, which allows the gateway blindly verifies the incoming traffic. Our scheme captures security properties: confidentiality and anonymity against dishonest gateway, corrupted receivers and collusion attacks. We present a concrete construction of gateway-based verifiable and anonymous broadcast encryption system from bilinear pairings, and give its security reduction under the computational assumptions related to bilinear pairings.

Keywords: anonymity; verifiable encryption; broadcast encryption.

Reference to this paper should be made as follows: Cui, H., Mu, Y. and Au, M.H. (2014) 'Anonymous broadcast encryption with an untrusted gateway', *Int. J. Security and Networks*, Vol. 9, No. 1, pp.20–29.

Biographical notes: Hui Cui is a PhD student from the Faculty of Engineering and Information Sciences at University of Wollongong. Her academic supervisors are Professor Yi Mu and Dr. Man Ho Au. She received both her Bachelor (2008) and Master (2011) degrees in the Faculty of Computer Science, China University of Mining and Technology.

Yi Mu received his PhD from the Australian National University in 1994. Prior to joining University of Wollongong, he was a Senior Lecturer in the Department of Computing, Macquarie University. He also worked in Department of Computing and IT, University of Western Sydney as a Lecturer. He has been with the University of Wollongong since 2003. His current research interest includes cryptography, network security, access control, and computer security. He also previously worked in the areas of quantum cryptography, quantum computers, atomic computations, and quantum optics. He has published over 270 research papers.

Man Ho Au obtained his PhD from the Faculty of Informatics at University of Wollongong in the year 2009. His academic supervisors are Dr. Willy Susilo and Dr. Yi Mu. He is currently an associate lecturer at UoW. His research interests include public key cryptography and network security. In particular, he is interested in privacy-preserving cryptographic systems including anonymous credential systems and electronic cash systems.

1 Introduction

Considering the following scenario. The gateway of an organisation rejects any inbound traffic unless a message is indeed for at least one member in the organisation. In this scenario, the gateway might not be regarded as honest in terms of whom the intended receiver(s) would be. It could be a case that an agent who works for the organisation sends an inbound encrypted message to the organisation, where only the agent knows the receivers. In this scenario, a sender outside the organisation should be able to make sure that a message for target receivers inside the organisation will not be rejected by the gateway while maintaining the message confidentiality and receiver anonymity. To ensure the maximal security, we should also consider that the gateway could potentially collude with the corrupted receivers, who will leak all the personal information related to their identities in order to compromise other users.

In order to define a model for the above scenario, we propose a framework in which one can design a protocol

among three parties (the sender, the gateway and the receivers) to protect both the message content and the privacy of receivers. We apply a technique, where each member in the organisation sends a 'trapdoor' related to its private key and identifier, to the gateway through a public channel, so that the gateway can use this piece of information to verify whether the receivers belong to the organisation. The trapdoor does not disclose any personal information, as it is encrypted; however, the gateway can blindly verify the identifier embedded in the trapdoor. We call the method as public key encryption and identity search (PKEIS). In order to achieve this feature we need an anonymous encryption scheme, addressing the issue of receiver privacy. Specifically speaking, a designated receiver can only be sure that he is one of receivers of the message and learns nothing about other receivers of the message. On the other hand, it is computationally hard for the gateway to distinguish the relationship between a public key and a trapdoor by combining the ciphertext and trapdoors together without the private keys of receivers.

The most challenging aspect of a gateway-based verifiable and anonymous broadcast scheme is preventing collusion. Unlike conventional broadcast encryption, collusion attacks in our model are much more complicated:

- the gateway may collude with corrupted receivers, who provide their personal information (such as private key, identity token, trapdoor), to obtain the message content
- the gateway may collude with the corrupted receivers to guess the identities of privileged receivers
- the gateway may collude with corrupted receivers to discern the relations between the ciphertexts and the trapdoors involved.

For example, suppose $R_1, \ldots, R_t$ are some privileged receivers, and $R'_1, \ldots, R'_t$ are some non-privileged receivers. In our model, the gateway, with the help of receivers $S \subset \{R_1, \ldots, R_t\}$, should not be able to confirm who else can decrypt the ciphertext; meanwhile, with the help of receivers $S \subset \{R'_1, \ldots, R'_t\}$, should not be able to decrypt the ciphertext or obtain the information about the receivers involved in the ciphertext.

We require the following security notions which capture dishonest behaviours of the sender, the gateway and the receivers, respectively.

- The confidentiality of the message (or semantic security of the message). Confidentiality here means that the gateway cannot obtain any information about the plaintext from the ciphertext, even if it colludes with the non-privileged receivers.
- The anonymity of the receivers. Anonymity here means that the gateway cannot know who will be the privileged receivers from the ciphertext, even if it colludes with the corrupted receivers, which also implies that the gateway cannot distinguish the relations between the ciphertexts and the trapdoors with the information it obtains from the corrupted receivers.

Assume that the collusion is among t_1 non-privileged receivers and t_2 privileged receivers. With the above security notions, breaking into the gateway would neither help t_1 non-privileged receivers to gain any information about the message and identity information of the privileged receivers, nor t_2 privileged receivers to obtain the identity information of other members who receive the same message.

A related but not closely related work is verifiable encryption. Verifiable encryption has the property that the validity of ciphertext can be verified without knowledge of private key, and it has been used for fair exchange (Asokan et al., 1998; Bao, 1998), key escrow schemes (Poupard and Stern, 2000), signature sharing schemes (Franklin and Reiter, 1995) and publicly verifiable secret sharing (Stadler, 1996). The concept of verifiable encryption was first introduced by Stadler (1996) with the cut-and-choose methodology in the context of publicly verifiable sharing schemes in 1996. Then, Asokan et al. (1998) proposed a more general form of verifiable encryption with perfect separability for the purpose of fair exchange of signatures in 1998. Bao (1998) gave a verifiable encryption scheme without using the cut-and-choose methodology, but it failed to provide semantic security (Goldwasser and Micali, 1984). Camenisch and Lysyanskaya (2001) proposed an anonymous verifiable encryption scheme which did not use the cut-and-choose methodology in 2001, but the prover needs to know the private key of the receiver. Camenisch and Shoup (2003) introduced a verifiable encryption system that provides chosen ciphertext security and avoids inefficient cut-and-choose proofs in 2003; however, it requires to use Paillier encryption function (Paillier, 1999).

Broadcast encryption (BE) is designed to address the problem of broadcasting a message to an arbitrary subset S from a universe of receivers U who are listening on a broadcast channel. Since it was introduced by Fiat and Naor (1993) in 1993, various BE schemes have been proposed (Boneh et al., 2005; Delerabl et al., 2007; Delerablée, 2007; Dodis and Fazio, 2002; Fiat and Naor, 1993; Gentry and Waters, 2009) from different aspects such as strength of security notions, public and private key storage requirements, ciphertext length, computational costs, and so on. Concerning making ciphertexts to be as short as possible, schemes in Boneh et al. (2005) and Gentry and Waters (2009) are close to optimal. With regard to the privacy of receivers, (Barth et al., 2006) first considered privacy of users in broadcast encryption under the context of encryption systems, Then Libert et al. (2011) gave a generalised and unified security definition for anonymous broadcast encryption.

Private information retrieval (PIR) protocols introduced by Chor et al. (1995) allow users to retrieve some data items or search some data items from a public database without revealing to the database administrator which items they retrieve or search, but the data should be public. Boneh et al. (2004) proposed public key encryption with keyword search (PEKS) to enable one, say Alice, to provide a key to the gateway such that the gateway can test whether the word 'urgent' (or other words) is a keyword in the e-mail without learning anything else about the e-mail, but it needs a secure channel between Alice and the e-mail server. In order to solve this problem, Baek et al. (2008) provided a secure channel

free public key encryption with keyword search (SCF-PEKS) scheme.

Motivated by the above work, we propose a cryptographic system: public key encryption and identity search to solve our problem, such that the gateway could check whether the identities tokens (not the data items or keywords) used in the broadcast encryption are from the data stored by the gateway. We introduce a new primitive called gateway-based verifiable and anonymous broadcast encryption which would make the gateway allow a sender to transmit some messages to multiple receivers remaining perfectly confidential and anonymous. Specifically, this primitive provides confidentiality of the plaintext and anonymity of privileged receivers while preventing collusion attacks between the gateway and the corrupted receivers.

In order to provide a secure, verifiable and anonymous broadcast encryption scheme, we propose a notion called public key encryption and identity search (PKEIS), with which we can design a secure framework for our problem. In this construction, there is a third party authority centre (AC) to generate a public key and private key pair for the gateway and assign every potential receiver a secret identity token, whilst the receivers generate their own public and private key pairs. More specifically speaking, when Alice wants to send a message to Bob, John and so on, she encrypts her message with PKEIS algorithm to generate the ciphertext composed of two parts: encryption of the message and verification of receivers' identity tokens. The gateway checks whether the message is for the members in the organisation with the verification part while the privileged receivers decrypt the ciphertext to gain the plaintext with the encryption part.

Our construction for gateway-based verifiable and anonymous broadcast encryption protects the confidentiality of message content and privacy of privileged receivers according to a formal model which we specify. It provides the additional property: collusion resistance, which means that the gateway cannot learn any information about the message if it colludes with t_1 non-privileged receivers; besides, it cannot learn any information about the privacy of privileged receivers if it colludes with t_2 privileged receivers. Briefly speaking, the gateway learns nothing from the ciphertext even if it colludes with corrupted receivers.

The remainder of this paper is organised as follows. In Section 2, we describe the framework and security model of gateway-based verifiable and anonymous broadcast encryption. In Section 3, we review the conception of bilinear pairings and the related complexity assumptions. In Section 4, we provide an efficient instantiation according to the formal framework of gateway-based verifiable and anonymous broadcast encryption, as well as its security proof based on the BDH complexity assumptions and an application. In Section 5, we conclude the contributions in this paper and leave some open problems.

2 Preliminaries

In this section, we first present a formal framework of our primitive: gateway-based verifiable and anonymous broadcast encryption. We describe the adversary model and games between an adversary and a challenger.

2.1 Definition

In gateway-based verifiable and anonymous broadcast encryption, three parties are involved: sender, receivers, and gateway. Every receiver generates a trapdoor according to its identity token and sends it to the gateway. The sender generates and sends encrypted message which we call 'PKEIS ciphertext'. The gateway receives PKEIS ciphertext and performs verification according to the trapdoors received from the receivers. If the ciphertext passes the verification, the gateway broadcasts the ciphertext; otherwise, it rejects the ciphertext.

In our framework, every receiver R_i generates its own public and private key pair (pk_{R_i}, sk_{R_i}). An authority centre (AC) provides an identity token ID_i for every potential receiver R_i and a public and private key pair (pk_G, sk_G) for gateway $\mathcal{G}$. Every receiver R_i generates a trapdoor T_i on identity token ID_i with private key sk_{R_i} and sends this trapdoor to gateway $\mathcal{G}$. Gateway $\mathcal{G}$ maintains a trapdoor list L_T, whenever it receives a trapdoor from a receiver R_i, it adds this trapdoor to L_T. With the private key sk_G and the trapdoor list L_T, gateway $\mathcal{G}$ can verify the ciphertext and determine whether to broadcast it or not.

A gateway-based verifiable and anonymous broadcast encryption scheme is specified by seven randomised algorithms: Setup, Make-GKey, Make-RKey, Encrypt, Verify and Decrypt.

- Setup(k): Taking a security parameter k as input, this algorithm outputs a common parameter $params$.
- Make-GKey($params$): Taking the public parameter $params$ as input, this algorithm outputs the public and private key pair (pk_G, sk_G) for gateway $\mathcal{G}$.
- Make-RKey($params$): Taking the public parameter $params$ as input, this algorithm outputs the public and private key pair (pk_{R_i}, sk_{R_i}) for receiver R_i.
- Trapdoor($params$, sk_{R_i}, ID_i): Taking the public parameter $params$, the secret key sk_{R_i} and the identity token ID_i of the receiver R_i as input, this algorithm outputs the trapdoor T_i for ID_i.
  Receiver R_i sends its trapdoor T_i to gateway $\mathcal{G}$ through a public broadcast channel. Once gateway $\mathcal{G}$ receives a trapdoor T_i, it will add T_i to the trapdoor list L_T which is initially empty.
- Encrypt($params$, M, pk_G, PK, ID_i): Taking the public parameter $params$, the message M, the public key pk_G of gateway $\mathcal{G}$, the public keys $pk_{R_1}, \ldots, pk_{R_t}$ and the corresponding identity tokens $ID_1, \ldots, ID_t$ of receivers $R_1, \ldots, R_t$ as input, this algorithm outputs the ciphertext C.
- Verify($params$, sk_G, L_T, C): Taking the public parameter $params$, the private key sk_G of gateway $\mathcal{G}$, the trapdoor list L_T for all the receivers, and the

ciphertext C as input, this algorithm outputs C in case of success or $\perp$ in case of failure.

- Decrypt($params$, sk_{R_i}, T_i, C): Taking the public parameter $params$, the private key sk_{R_i} and the trapdoor T_i of the receiver R_i, and the ciphertext C as input, this algorithm outputs the message M in case of being a member of privileged receivers or $\perp$ in case of being a member of non-privileged receivers.

Let S be the recipient set, we require that our system is correct, meaning that for all $R_i \in S$, if $params \leftarrow$ Setup(k), (pk_G, sk_G) $\leftarrow$ Make-GKey($params$), (pk_{R_i}, sk_{R_i}) $\leftarrow$ Make-RKey($params$), $T_i \leftarrow$ Trapdoor($params$, sk_{R_i}, ID_i), $C \leftarrow$ Encrypt($params$, M, pk_G, pk_{R_i}, ID_i), and $C \leftarrow$ Verify($params$, sk_G, L_T, C), then $M =$ Decrypt($params$, sk_{R_i}, T_i, C).

2.2 Security model

The goal of a gateway-based verifiable and anonymous broadcast encryption scheme is to send a message to multiple receivers under the verification of the gateway while keeping the confidentiality of the message content and the anonymity of the receivers. In a gateway-based verifiable and anonymous broadcast encryption protocol, which involves three parties: a sender, a gateway and the potential receivers, collusion resistance have to be dealt with:

- a collusion of the corrupted receivers and the gateway may help the adversary to recover the message from the ciphertext
- a collusion of the corrupted receivers and the gateway may help the adversary to identify the identity tokens of the privileged receivers
- a collusion of the corrupted receivers and the gateway may help the adversary to discern the trapdoors involved in the ciphertext.

In our security model, collusion will be modelled by Corrupt queries, which will provide all the secret data of the gateway and the corrupted receivers to the adversary. Then, from all these information, the adversary will be allowed to do anything it wants to transfer some information.

As a consequence, we provide the adversary with two Corrupt oracles in the security model assuming that the adversary has known the public keys of receivers in the system: a Gateway_Corrupt oracle that outputs the trapdoor list stored by the gateway, and a Receiver_Corrupt oracle that outputs the related information (including public key, private key, identity token, trapdoor) of corrupted receiver R_i. We denote all the public keys of potential receivers by PK, and the corruption list of corrupted receivers by L_{CR}.

- Gateway_Corrupt(PK) $\rightarrow L_T$, takes the public key set PK as input, this algorithm outputs all the trapdoors in L_T.
- Receiver_Corrupt(pk_{R_i}) $\rightarrow$ (sk_{R_i}, ID_i, T_i), takes the public key pk_{R_i} as input, this algorithm outputs the corresponding private key, identity token, and trapdoor in a 3-tuple (sk_{R_i}, ID_i, T_i).

Confidentiality. We define a chosen plaintext attack for gateway-based verifiable and anonymous broadcast encryption to ensure the confidentiality of the message content. More precisely, confidentiality is defined using a game between an adversary algorithm $\mathcal{A}$ and a challenger algorithm $\mathcal{B}$ that algorithm $\mathcal{A}$ cannot distinguish a ciphertext intended for one message from a ciphertext intended for another message.

- *Initialisation.* Algorithm $\mathcal{B}$ runs Setup to obtain the public parameter $params$. Then, algorithm $\mathcal{B}$ generates the public and private key pair (pk_G, sk_G) for gateway $\mathcal{G}$. Algorithm $\mathcal{B}$ gives the public parameter $params$, the public and private key pair (pk_G, sk_G) to algorithm $\mathcal{A}$. For every receiver R_i where $i = 1, \ldots, n$, algorithm $\mathcal{B}$ generates a public and private key pair (pk_{R_i}, sk_{R_i}). Algorithm $\mathcal{B}$ gives their public keys pk_{R_i} to algorithm $\mathcal{A}$ while keeping their private keys sk_{R_i} secret.
- *Query Phase 1.* Algorithm $\mathcal{A}$ adaptively issues Receiver_Corrupt query on pk_{R_i}. Algorithm $\mathcal{B}$ forwards the corresponding private key, identity token, and trapdoor in a 3-tuple (sk_{R_i}, ID_i, T_i) to algorithm $\mathcal{A}$.
- *Query Phase 2.* Algorithm $\mathcal{A}$ issues Gateway_Corrupt query on $PK = \{pk_{R_1}, \ldots, pk_{R_n}\}$. Algorithm $\mathcal{B}$ forwards the trapdoor list $L_T = \{T_1, \ldots, T_n\}$ to algorithm $\mathcal{A}$.
  Note that algorithm $\mathcal{A}$ knows nothing about the relations between the public key pk_{R_i} and the trapdoor T_i except those in the list L_{CR}.
- *Challenge.* When algorithm $\mathcal{A}$ decides that Phase 2 is over, it outputs two messages M_0^*, M_1^*, a public key set $PK^* \subset PK$ on which it wishes to be challenged. The only constraint is that $pk_{R_i} \in PK^*$ did not appear in Phase 1. To generate the challenge ciphertext, algorithm $\mathcal{B}$ chooses a random bit $\gamma \in \{0, 1\}$, and runs Encrypt on M_γ^* to obtain the ciphertext C^*. It sends C^* as the ciphertext to algorithm $\mathcal{A}$.
- *Query Phase 3.* $\mathcal{A}$ continues to adaptively issue Receiver_Corrupt query on pk_{R_i} of which $pk_{R_i} \notin PK^*$, as in Phase 1.
- *Guess.* $\mathcal{A}$ outputs its guess $\gamma' \in \{0, 1\}$ for γ and wins the game if $\gamma = \gamma'$.

We refer to such an adversary algorithm $\mathcal{A}$ as an IND-CPA adversary. We define the advantage of the adversary algorithm $\mathcal{A}$ in attacking a gateway-based verifiable and anonymous broadcast encryption scheme $\varepsilon =$ (Setup, Make-GKey, Make-RKey, Trapdoor, Encrypt, Verify, Decrypt) as

$$\text{Adv}_{\varepsilon,\mathcal{A}} = |\Pr[\gamma = \gamma'] - 1/2|$$

The probability is over the random bits used by the challenger and the adversary.

Definition 1: We say that a gateway-based verifiable and anonymous broadcast encryption scheme ε is (t, q_T, ϵ)-IND-CPA secure if for any IND-CPA adversary algorithm $\mathcal{A}$ that runs in time t, makes at most q_T Receiver_Corrupt queries, we have that $\text{Adv}_{\varepsilon,\mathcal{A}} < \epsilon$.

There is another stronger version of security, the chosen ciphertext security, where the adversary is not only allowed to issue adaptive Receiver_Corrupt queries, but also allowed to issue decryption queries.

Definition 2: We say that a gateway-based verifiable and anonymous broadcast encryption scheme ε is (t, q_T, ϵ)-IND-CPA secure if ε is $(t, q_T, 0, \epsilon)$-IND-CCA2 secure.

Anonymity. We define the following game to ensure that the adversary cannot distinguish a ciphertext intended for one recipient set from a ciphertext intended for another recipient set. More precisely, receiver anonymity is defined using a game between an adversary algorithm $\mathcal{A}$ and a challenger algorithm $\mathcal{B}$.

- *Initialisation.* Algorithm $\mathcal{B}$ runs Setup to obtain the public parameter $params$. Then, algorithm $\mathcal{B}$ generates the public and private key pair (pk_G, sk_G) for gateway $\mathcal{G}$. Algorithm $\mathcal{B}$ gives the public parameter $params$, the public and private key pair (pk_G, sk_G) to algorithm $\mathcal{A}$. For every receiver R_i where $i = 1, \ldots, n$, algorithm $\mathcal{B}$ generates an identity token ID_i, a public and private key pair $(ID_i, pk_{R_i}, sk_{R_i})$ as well as the trapdoor T_i. Algorithm $\mathcal{B}$ gives their public keys (pk_{R_i}) to algorithm $\mathcal{A}$.
  item *Query Phase 1.* Algorithm $\mathcal{A}$ adaptively issues Receiver_Corrupt query on pk_{R_i}. Algorithm $\mathcal{B}$ forwards the corresponding private key, identity token, and trapdoor in a 3-tuple (sk_{R_i}, ID_i, T_i) to algorithm $\mathcal{A}$.
- *Query Phase 2.* Algorithm $\mathcal{A}$ issues Gateway_Corrupt query on $PK = \{pk_{R_1}, \ldots, pk_{R_n}\}$. Algorithm $\mathcal{B}$ forwards the trapdoor list $L_T = \{T_1, \ldots, T_n\}$ to algorithm $\mathcal{A}$.
  Note that algorithm $\mathcal{A}$ knows nothing about the relations between the public key pk_{R_i} and the trapdoor T_i except those in the list L_{CR}.
- *Challenge.* When algorithm $\mathcal{A}$ decides that Phase 2 is over, it outputs a messages M^*, and two public key sets $PK_0^*, PK_1^* \subset PK$ on which it wishes to be challenged where PK_0^*, PK_1^* are of equal size l. The only constraint is that none of the $pk_{R_i} \in PK_0^* \cup PK_1^*$ appeared in Phase 1. To generate the challenge ciphertext, algorithm $\mathcal{B}$ retrieves the corresponding identity token sets $S_0^*, S_1^* \subset \{ID_1, \ldots, ID_n\}$ of equal size l, and them chooses a random bit $\gamma \in \{0, 1\}$, and runs Encrypt on PK_γ^*, S_γ^* to obtain the ciphertext C^*. It sends C^* as the ciphertext to algorithm $\mathcal{A}$.
- *Query Phase 3.* $\mathcal{A}$ continues to adaptively issue Receiver_Corrupt query on pk_{R_i}, as in Phase 1.
- *Guess.* $\mathcal{A}$ outputs its guess $\gamma' \in \{0, 1\}$ for γ and wins the game if $\gamma = \gamma'$.

We refer to such an adversary algorithm $\mathcal{A}$ as an ANON-IND-CPA adversary. We define the advantage of the adversary algorithm $\mathcal{A}$ in attacking a gateway-based verifiable and anonymous broadcast encryption scheme $\varepsilon =$ (Setup, Make-GKey, Make-RKey, Trapdoor, Encrypt, Verify, Decrypt) as

$$\text{Adv}_{\varepsilon,\mathcal{A}} = |\Pr[\gamma = \gamma'] - 1/2| \,.$$

The probability is over the random bits used by the challenger and the adversary.

Definition 3: We say that a gateway-based verifiable and anonymous broadcast encryption scheme ε is (t, q_T, ϵ)-ANON-IND-CPA secure if for any ANON-IND-CPA adversary algorithm $\mathcal{A}$ that runs in time t, makes q_T Receiver_Corrupte query, we have that $\text{Adv}_{\varepsilon,\mathcal{A}} < \epsilon$.

There is another stronger version of security, the chosen ciphertext security, where the adversary is not only allowed to issue adaptive Receiver_Corrupt queries, but also allowed to issue decryption queries.

Definition 4: We say that a gateway-based verifiable and anonymous broadcast encryption scheme ε is $(t, q_T, 0, \epsilon)$-ANON-IND-CCA2 secure if ε is $(t, q_T, 0, \epsilon)$-ANON-IND-CCA2 secure.

3 Bilinear pairings and complexity assumptions

In this section, we review the definitions of bilinear pairings and bilinear pairing groups, and the computational assumptions related to bilinear pairings that are relevant to the protocol we discuss in this paper.

3.1 Notations

If n is a positive integer, we use $[n]$ to denote the set $\{1, \ldots, n\}$.

3.2 Bilinear pairings

Let G and $\hat{G}$ be two multiplicative cyclic groups of prime order q. Let g be a generator of G and $\hat{g}$ be a generator of $\hat{G}$, we define $\hat{e} : G \times \hat{G} \to G_T$ to be a bilinear map if it has the following properties (**?**):

- *Bilinear*: for all $g \in G$, $\hat{g} \in \hat{G}$ and $a, b \in Z$, we have $\hat{e}(g^a, \hat{g}^b) = \hat{e}(g, \hat{g})^{ab}$.
- *Non-degenerate*: $\hat{e}(g, \hat{g}) \neq 1$.

We say that $(G, \hat{G})$ is a bilinear group if the group action in $(G, \hat{G})$ can be computed efficiently and there exists a group G_T and an efficiently computable bilinear map $\hat{e} : G \times \hat{G} \to G_T$ as above.

3.3 Complexity assumptions

Computational DL. We say that an algorithm $\mathcal{A}$ has advantage $\text{Adv}_{\mathcal{A}}^{\text{DL}} = \epsilon$ in solving the computational discrete log (DL) problem in G if

$$\Pr[\mathcal{A}(g, g^a) = a] \geq \epsilon$$

where the probability is over the random choice of generators g of G, the random choice of exponent a in Z_p^*, and the random bits used by algorithm $\mathcal{A}$.

Computational DL Oracle $\mathcal{O}_g^{DL}$. This oracle is defined over a cyclic group $\mathbb{G} = \langle g \rangle$. On input a value $Y \in \mathbb{G}$, this oracle outputs $a \in \mathbb{Z}_{|\mathbb{G}|}$ such that $Y = g^a$.

We make use of the generalisation of the Bilinear Diffie-Hellman assumptions in Boneh and Boyen (2011), which includes a class of assumptions appearing with various pairing-based schemes.

Computational BDH. We say that an algorithm $\mathcal{A}$ has advantage $\text{Adv}_{\mathcal{A}}^{\text{BDH}} = \epsilon$ in solving the computational BDH problem in $(G, \hat{G})$ if

$$\Pr[\mathcal{A}(g, g^a, g^b, \hat{g}, \hat{g}^a, \hat{g}^c) = \hat{e}(g, \hat{g})^{abc}] \geq \epsilon$$

where the probability is over the random choice of generators g of G and $\hat{g}$ of $\hat{G}$, the random choice of exponents a, b, c in Z_p^*, and the random bits used by algorithm $\mathcal{A}$.

Decisional BDH. We say that an algorithm $\mathcal{A}$ that outputs a bit $\gamma \in \{0, 1\}$ has advantage $\text{Adv}_{\mathcal{A}}^{\text{D-BDH}} = \epsilon$ in solving the Decisional Bilinear Diffie-Hellman (BDH) problem in $(G, \hat{G})$ if

$$\big|\Pr[\mathcal{A}(g, g^a, g^b, \hat{g}, \hat{g}^a, \hat{g}^c, \hat{e}(g, \hat{g})^{abc}) = 0] - \Pr[\mathcal{A}(g, g^a, g^b, \hat{g}, \hat{g}^a, \hat{g}^c, Z) = 0]\big| \geq \epsilon$$

where the probability is over the random choice of generators g of G and $\hat{g}$ of $\hat{G}$, the random choice of exponents a, b, c in Z_p^*, the random choice of $Z \in G_T$, and the random bits used by algorithm $\mathcal{A}$.

We also define a variant version of computational BDH assumption and decisional BDH assumption, respectively.

Computational X-BDH. We say that an algorithm $\mathcal{A}$ has advantage $\text{Adv}_{\mathcal{A}}^{\text{XBDH}} = \epsilon$ in solving the computational X-BDH problem in $(G, \hat{G})$ if

$$\Pr[\mathcal{A}(g, g^a, \{g^{b_i}\}_{i\in[n]}, \hat{g}, \hat{g}^a, \hat{g}^c, \mathcal{O}_g^{\mathsf{DL}}) = \{\hat{e}(g, \hat{g})^{ab_kc}\}_{k\in\mathcal{L}}, \mathcal{L}\}] \geq \epsilon$$

where $\mathcal{L} \subset [n]$ and all inputs to $\mathcal{O}_g^{\mathsf{DL}}$ belongs to the set $\{g^{b_i}\}_{i\in[n]\setminus\mathcal{L}}$. The probability is over the random choice of generators g of G and $\hat{g}$ of $\hat{G}$, the random choice of exponents $a, \{b_i\}_{i\in[n]}, c$ in Z_p^*, and the random bits used by algorithm $\mathcal{A}$.

Decisional X-BDH. We say that an algorithm $\mathcal{A}$ outputs a bit $\gamma \in \{0, 1\}$ has advantage $\text{Adv}_{\mathcal{A}}^{\text{D-XBDH}} = \epsilon$ in solving the Decisional X-BDH problem in $(G, \hat{G})$ if

$$(\mathsf{st}, \mathcal{L}) \leftarrow \mathcal{A}(g, g^a, \{g^{b_i}\}_{i\in[n]}, \hat{g}, \hat{g}^a, \hat{g}^c, \mathcal{O}_g^{\mathsf{DL}}),$$
$$\big|\Pr[\mathcal{A}(\mathsf{st}, \mathcal{O}_g^{\mathsf{DL}}, \{\hat{e}(g, \hat{g})^{ab_kc}\}_{k\in\mathcal{L}}) = 0] - \Pr[\mathcal{A}(\mathsf{st}, \mathcal{O}_g^{\mathsf{DL}}, \{Z_k\}_{k\in\mathcal{L}}) = 0]\big| \geq \epsilon$$

where $\mathcal{L} \subset [n]$, and all inputs to $\mathcal{O}_g^{\mathsf{DL}}$ belongs to the set $\{g^{b_i}\}_{i\in[n]\setminus\mathcal{L}}$. The probability is over the random choice of generators g of G and $\hat{g}$ of $\hat{G}$, the random choice of exponents $a, \{b_i\}_{i\in[n]}, c$ in Z_p^*, the random choice of $\{Z_k\}_{i=1}^{l} \in G_T$, and the random bits used by algorithm $\mathcal{A}$.

In the decisional BDH assumption, we refer to the distribution over $G^3 \times \hat{G}^3 \times G_T$ of the 7-tuple in the true instance (on the left) as $\mathcal{P}_{DBDH}$, and in the false instance (on the right) as $\mathcal{R}_{DBDH}$.

Similarly, in the decisional X-BDH assumption, we refer to the distribution of $(g,\ g^a,\ \{g^{b_i}\}_{i\in[n]},\ \hat{g},\ \hat{g}^a,\ \hat{g}^c,\ \{\hat{e}(g, \hat{g})^{ab_kc}\}_{k\in\mathcal{L}})$ over $G^{n+2} \times \hat{G}^3 \times G_T^{|\mathcal{L}|}$ as $\mathcal{P}_{XBDH}$, and the distribution on the right as $\mathcal{R}_{XBDH}$.

Definition 5: We say that the (t, ϵ)-Decisional BDH assumption holds in $(G, \hat{G})$ if no t-time algorithm has advantage at least ϵ in solving the Decisional BDH problem in $(G, \hat{G})$.

Similarly, we say that the (t, ϵ)-Decisional X-BDH assumption holds in $(G, \hat{G})$ if no t-time algorithm has advantage at least ϵ in solving the Decisional X-BDH problem in $(G, \hat{G})$.

4 Proposed scheme

In this section, we give a secure gateway-based verifiable and anonymous broadcast encryption scheme. We also provide a security proof for this scheme in the random oracle model.

4.1 Description

Suppose that there are a gateway $\mathcal{G}$ and a recipient set $S = \{R_1, \ldots, R_n\}$ in the system. The AC assigns an identity token ID_i for receiver R_i where $i = 1, \ldots, n$, and a public and private key pair $(pk_G,\ sk_G)$ for gateway $\mathcal{G}$. Receiver R_i generates its public and private key pair $(pk_{R_i},\ sk_{R_i})$, it publishes the public key pk_{R_i} while keeping the private key sk_{R_i} secret. After that, receiver R_i computes its trapdoor T_i corresponding to identity ID_i with private key sk_{R_i}, and sends T_i to gateway $\mathcal{G}$. Every time gateway $\mathcal{G}$ receives a trapdoor T_i, it will add T_i to the trapdoor list L_T. When gateway $\mathcal{G}$ receives an outside ciphertext, it checks whether this message is for some inside receivers ($R_i \in S$) with its private key sk_G and the trapdoor list L_T. If so, gateway $\mathcal{G}$ broadcasts this ciphertext; otherwise, it rejects and outputs $\perp$.

Our gateway-based verifiable and anonymous broadcast encryption scheme consists of the following seven algorithms.

- Setup(k): This algorithm takes a security parameter k as input. It chooses two groups $G, \hat{G}$ of prime order $q \geq 2^k$. It constructs a bilinear pairing $\hat{e} : G \times \hat{G} \rightarrow G_T$. It defines a hash function $H_1 : \{0,1\}^* \rightarrow \hat{G}$. It outputs $params = (G, \hat{G}, G_T, q, \hat{e}, g, \hat{g}, H_1)$ as the public parameter where $g \in G$.
- Make-GKey($params$): This algorithm takes the public parameter $params$ as input. It chooses $x \in Z_p^*$ uniformly at random and computes $X = g^x$. It outputs the public and private key pair $(pk_G, sk_G) = (X, x)$ for gateway $\mathcal{G}$.
- Make-RKey($params$): This algorithm takes the public parameter $params$ as input. For $i = 1, \ldots, n$, it chooses $y_i \in Z_p^*$ uniformly at random and computes $Y_i = g^{y_i}$. It outputs the public and private key pair $(pk_{R_i}, sk_{R_i}) = (Y_i, y_i)$ for receiver R_i.
- Trapdoor($params$, sk_{R_i}, ID_i): This algorithm takes the public parameter $params$, and the secret key sk_{R_i} and the identity token ID_i of receiver R_i as input. It computes $T_i = H_1(ID_i)^{y_i}$ where $i = 1, \ldots, n$. It outputs T_i as the trapdoor for receiver R_i.
- Encrypt($params$, M, pk_G, $\{pk_{R_i}, ID_i\}_{i \in \mathcal{L}}$): This algorithm takes the public parameter $params$, the message $M \in G_T$, the public key pk_G of the gateway, and the set of recipients' public key and identity token $\{Y_i, ID_i\}_{i \in \mathcal{L}}$ as input. It chooses $r \in Z_p^*$ and computes $\{C_{1,i} = M \cdot \hat{e}(Y_i, \hat{g})^r\}_{i \in \mathcal{L}}$, $C_2 = g^r$, $\{C_{3,i} = \hat{e}(X, \hat{g})^r \cdot \hat{e}(Y_i, H_1(ID_i))^r\}_{i \in \mathcal{L}}$. It outputs $C = (\{C_{1,i}\}_{i \in \mathcal{L}}, C_2, \{C_{3,i}\}_{i \in \mathcal{L}})$ as the ciphertext.
- Verify($params$, sk_G, L_T, C): This algorithm takes the public parameter $params$, the secret key sk_G of gateway G and the trapdoor list L_T and a ciphertext C as input. It parses the ciphertext as $C_{1,i}$, C_2, $C_{3,i}$ for $i = 1$ to K. Next, it checks whether there exists a trapdoor $T \in L_T$ such that $\hat{e}(C_2, \hat{g}^x \cdot T) = C_{3,i}$ for all $i = 1$ to K. If all the equation holds, it outputs the ciphertext C.
- Decrypt($params$, sk_{R_i}, T_i, C): This algorithm takes the public parameter $params$, the secret key sk_{R_i} and the trapdoor T_i of receiver R_i, and the ciphertext C as input. If receiver R_i is a privileged receiver, it outputs $M = C_{1,i} \cdot \hat{e}(C_2, \hat{g}^{-y_i})$. Otherwise, it outputs a failure symbol $\perp$.

Efficiency. Our scheme achieves $O(1)$-size public keys and $O(n)$-size ciphertexts and constant size private keys. Note that the ciphertext is linear in the size of S, and not in the maximal number of decryption keys that can be distributed. Besides, because $\hat{e}(Y_i, \hat{g})$, $\hat{e}(X, \hat{g})$ and $\hat{e}(Y_i, H_1(ID_i))$ can be pre-computed, pairing computations are greatly reduced. In our scheme, encryption needs no pairing computation and $2 \cdot t + 2$ exponentiation computation while decryption needs one pairing computation and one exponentiation computation.

4.2 Security analysis

We present the security reduction of our gateway-based verifiable and anonymous broadcast encryption scheme by showing that it is secure under the games defined in Section 2.

In our scheme, public keys of receiver R_i and gateway $\mathcal{G}$ will be publicised to all the involved parties, while identity token ID_i of receiver R_i generated by the AC is known to receiver R_i and the outside sender, private key sk_{R_i} of receiver R_i generated by receiver R_i is only known to receiver R_i, private key sk_G of gateway $\mathcal{G}$ generated by the AC is known to gateway $\mathcal{G}$, and trapdoor T_i of receiver R_i generated by receiver R_i is known to gateway $\mathcal{G}$ and receiver R_i. Note that only receiver R_i knows the relations between its public key pk_{R_i}, private key sk_{R_i} and trapdoor T_i.

Theorem 1: *The above scheme is confidential assuming that the Decisional* (t, ϵ) *X-BDH assumption holds in* $(G, \hat{G})$.

Proof: Suppose there exists a (t, ϵ)-algorithm $\mathcal{A}$ against the confidentiality of our gateway-based verifiable and anonymous broadcast scheme. Algorithm $\mathcal{A}$ may get the help from gateway $\mathcal{G}$ and q_T corrupted receivers. The former will give its public and private key pair (pk_G, sk_G) and answer the Gateway_Corrupt query on the public key sets $PK = \{pk_{R_i}\}_{i \in \mathcal{L}}$, and the latter will answer the Receiver_Corrupt query on pk_{R_i}.

Thus, we construct an algorithm $\mathcal{B}$ that solves the Decisional (t, ϵ) X-BDH problem. Specifically, algorithm $\mathcal{B}$ is given $(g, g^a, \{g^{b_i}\}_{i \in [n]}, \hat{g}, \hat{g}^a, \hat{g}^c, \mathcal{O}_{DL})$ and the definition of G, $\hat{G}$, G_T together with the bilinear map $\hat{e}$ as the problem instance. Recalled that at some point algorithm $\mathcal{B}$ have to output a set $\mathcal{L}^* \subset [n]$ and receives a set $\{Z_k\}_{k \in \mathcal{L}^*}$. We defer the description of this step later. Based on the problem instance, algorithm $\mathcal{B}$ creates the system parameter as follows.
Initialisation. The system parameter is generated base on the problem instance. Algorithm $\mathcal{B}$ sets $\hat{g}' = \hat{g}^c$, and

- outputs $params = (G, \hat{G}, G_T, q, \hat{e}, g, \hat{g}', H_1)$ as the public parameter, where H_1 is a random oracle controlled by algorithm $\mathcal{B}$.
- generates n identity tokens $ID_1, \ldots, ID_n$, and sets a corruption list L_{CR} which is initially empty.
- chooses $x \in Z_p^*$ uniformly at random and computes $X = g^x$. It outputs $(pk_G, sk_G) = (X, x)$ as the public and private key pair of gateway $\mathcal{G}$.
- outputs $PK = \{g^{b_1}, \ldots, g^{b_n}\} = \{Y_1, \ldots, Y_n\}$ as the public keys of n potential receivers $R_1, \ldots, R_n$.

Phase 1. Algorithm $\mathcal{A}$ queries pk_{R_i} to the Receiver_Corrupt oracle, algorithm $\mathcal{B}$

- issues a query to oracle $\mathcal{O}_g^{\mathsf{DL}}$ on input Y_i and obtains the value b_i such that $Y_i = g^{b_i}$.
- computes the trapdoor $T_i = H_1(ID_i)^{b_i}$.

- outputs the corresponding private key, identity token and trapdoor in a 3-tuple $(sk_{R_i}, ID_i, T_i) = (b_i, ID_i, H_1(ID_i)^{b_i})$ as the answer.
- updates the corruption list L_{CR} with $(pk_{R_i}, sk_{R_i}, ID_i, T_i)$.

Phase 2. Algorithm $\mathcal{A}$ queries $PK = \{pk_{R_1}, \ldots, pk_{R_n}\}$ to the Gateway_Corrupt oracle. Algorithm $\mathcal{B}$

- chooses $r_i \in Z_p^*$, and computes $T_i = \hat{g}'^{r_i}$.
- outputs all the trapdoors $\{T_1, \ldots, T_n\}$ to algorithm $\mathcal{A}$.

Note that algorithm $\mathcal{B}$ has implicitly assumed $H_1(ID_i) = \hat{g}'^{r_i/b_i}$. If algorithm $\mathcal{A}$ query ID_i without issuing Receiver_Corrupt query on Y_i, algorithm $\mathcal{B}$ aborts. This happens with negligible probability since the value ID_i is hidden from algorithm $\mathcal{A}$ if the corrupt query on Y_i is not issued. Otherwise, algorithm $\mathcal{B}$ returns $\hat{g}'^{r_i/b_i}$ as the hash value of ID_i. Thus, simulation is perfect in the random oracle model.
Challenge. Algorithm $\mathcal{A}$ outputs two messages $M_0^*, M_1^* \in G_T$, and a public key set $PK^* \subset PK$ where PK^* is of size l, with the restriction that $pk_{R_i} \in PK^*$ does not exist in the list L_{CR}. For notational convenience, let $\mathcal{L}^*$ be an index set such that $\{Y_i\}_{i\in\mathcal{L}} = PK^*$. At this stage algorithm $\mathcal{B}$ submits $\mathcal{L}^*$ and receives $\{Z_k\}_{k\in\mathcal{L}^*}$. Algorithm $\mathcal{B}$'s task is to distinguish if $Z_k = \hat{e}(g,\hat{g})^{ab_kc}$ for all $k \in \mathcal{L}^*$. To create the challenge ciphertext with this problem instance, algorithm $\mathcal{B}$

- selects a random bit $\gamma \in \{0,1\}$, set $C_2^* = g^a$, and computes, for all $i \in \mathcal{L}^*$,

$$C_{1,i}^* = M_\gamma^* \cdot Z_i, \quad C_{3,i}^* = \hat{e}(g^a, \hat{g}^c)^x \cdot \hat{e}(g^a, \hat{g}^c)^{r_i}.$$

- responds with the challenge ciphertext $C^* = (C_{1,i}^*, C_2^*, C_{3,i}^*)$.

For all $k \in \mathcal{L}^*$, if $Z_k = \hat{e}(g,\hat{g})^{ab_kc}$, we have

$$\begin{aligned} C_{1,k}^* &= M_\gamma^* \cdot Z_k = M_\gamma^* \cdot \hat{e}(g^{b_k}, \hat{g}^c)^a \\ &= M_\gamma^* \cdot \hat{e}(Y_i, \hat{g}')^a, \\ C_{3,k}^* &= \hat{e}(g^a, \hat{g}^c)^x \cdot \hat{e}(g^a, \hat{g}^c)^{r_k} \\ &= \hat{e}(g^x, \hat{g}')^a \cdot \hat{e}(g^{b_k}, (\hat{g}^c)^{r_k/b_k})^a \\ &= \hat{e}(X, \hat{g}')^a \cdot \hat{e}(Y_k, \hat{g}'^{r_k})^a \\ &= \hat{e}(X, \hat{g}')^a \cdot \hat{e}(Y_k, H_1(ID_k))^a. \end{aligned}$$

Hence, when $Z_k = \hat{e}(g,\hat{g})^{ab_kc}$, meaning that algorithm $\mathcal{B}$'s input is sampled from $\mathcal{P}_{XBDH}$, then C^* is a valid encryption of M_γ^* under the public key set PK_γ^* chosen by algorithm $\mathcal{A}$. On the other hand, when Z_k is uniform and independent in G_T, meaning that algorithm $\mathcal{B}$'s input is sampled from $\mathcal{R}_{XBDH}$, then C^* is independent of γ in the view of Algorithm $\mathcal{A}$.

Phase 3. Algorithm $\mathcal{A}$ continues to adaptively query pk_{R_i} to Receiver_Corrupt oracle with the restriction that $pk_{R_i} \notin PK^*$. Algorithm $\mathcal{B}$ responds as in Phase 1.

Guess. Finally, algorithm $\mathcal{A}$ outputs a guess $\gamma' \in \{0,1\}$. If $\gamma = \gamma'$, algorithm $\mathcal{B}$ outputs 1 meaning it wins the game. Otherwise, it outputs 0.

If $Z_k = \hat{e}(g,\hat{g})^{ab_kc}$, the simulation is perfect and algorithm $\mathcal{A}$ must satisfy $|\Pr[\gamma = \gamma']| = 1/2 + \epsilon$. On the other hand, if $Z_k \in_R \mathbb{G}_T$, the challenge ciphertext C^* contains no information on γ and thus $\Pr[\gamma = \gamma'] = 1/2$. The overall probability that algorithm $\mathcal{B}$ solves the X-DBDH problem correctly is thus $1/2 + \epsilon/2$.

This completes the proof of Theorem 1.

Theorem 2: *The above scheme is anonymous assuming that the (t,ϵ) Decisional X-BDH assumption holds in $(G, \hat{G})$.*

Proof: Suppose there exists a $(t,\ \epsilon)$-algorithm $\mathcal{A}$ against the confidentiality of our gateway-based verifiable and anonymous broadcast scheme. Algorithm $\mathcal{A}$ may get the help from gateway $\mathcal{G}$ and q_T corrupted receivers. The former will give its public and private key pair $(pk_G,\ sk_G)$ and answer the Gateway_Corrupt query on the public key sets $PK = \{pk_{R_i}\}_{i\in\mathcal{L}}$, and the latter will answer the Receiver_Corrupt query on pk_{R_i}.

Thus, we construct an algorithm $\mathcal{B}$ that solves the Decisional (t,ϵ) X-BDH problem. Specifically, algorithm $\mathcal{B}$ is given $(g, g^a, \{g^{b_i}\}_{i\in[n]}, \hat{g}, \hat{g}^a, \hat{g}^c, \mathcal{O}_{DL})$ and the definition of G, $\hat{G}$, G_T together with the bilinear map $\hat{e}$ as the problem instance. Recalled that at some point algorithm $\mathcal{B}$ have to output a set $\mathcal{L}^* \subset [n]$ and receives a set $\{Z_k\}_{k\in\mathcal{L}^*}$. We defer the description of this step later. Based on the problem instance, $\mathcal{B}$ creates the system parameter as follows.

Initialisation. The same in Theorem 1.

Phase 1. The same in Theorem 1.

Phase 2. The same in Theorem 1.

Challenge. Algorithm $\mathcal{A}$ outputs a message $M^* \in G_T$, and two public key sets $PK_0^*, PK_1^* \subset PK$ with the restriction that $pk_{R_i} \in PK_0^* \cup PK_1^*$ does not exist in the list L_{CR}, where PK_0^*, PK_1^* are of the same size l. For notational convenience, let $\mathcal{L}^*$ be an index set such that $\{Y_i\}_{i\in\mathcal{L}} = PK_0^* \cup PK_1^*$. At this stage algorithm $\mathcal{B}$ submits $\mathcal{L}^*$ and receives $\{Z_k\}_{k\in\mathcal{L}^*}$. Algorithm $\mathcal{B}$'s task is to distinguish if $Z_k = \hat{e}(g,\hat{g})^{ab_kc}$ for all $k \in \mathcal{L}^*$. To create the challenge ciphertext with this problem instance, algorithm $\mathcal{B}$

- chooses two identity token sets $S_0^*, S_1^* \subset \{ID_1, \ldots, ID_n\}$ of equal size l
- selects a random bit $\gamma \in \{0,1\}$, sets $C_2^* = g_1 = g^a$, and computes, for all $i \in \mathcal{L}^*$

$$C_{1,i}^* = M^* \cdot Z_i, \quad C_{3,i}^* = \hat{e}(g^a, \hat{g}^c)^x \cdot \hat{e}(g^a, \hat{g}^c)^{r_i}$$

- responds with the challenge ciphertext $C^* = (C_{1,i}^*, C_2^*, C_{3,i}^*)$.

For all $k \in \mathcal{L}^*$, if $Z_k = \hat{e}(g,\hat{g})^{ab_kc}$, we have

$$\begin{aligned} C_{1,k}^* &= M^* \cdot Z_k = M^* \cdot \hat{e}(g^{b_k}, \hat{g}^c)^a \\ &= M^* \cdot \hat{e}(Y_i, \hat{g}')^a, \\ C_{3,k}^* &= \hat{e}(g^a, \hat{g}^c)^x \cdot \hat{e}(g^a, \hat{g}^c)^{r_k} \\ &= \hat{e}(g^x, \hat{g}')^a \cdot \hat{e}(g^{b_k}, (\hat{g}^c)^{r_k/b_k})^a \end{aligned}$$

$$= \hat{e}(X, \hat{g}')^a \cdot \hat{e}(Y_k, \hat{g}'^{r_k})^a$$
$$= \hat{e}(X, \hat{g}')^a \cdot \hat{e}(Y_k, H_1(ID_k))^a.$$

where $ID_i \in S^*_\gamma$ and $Y_i \in PK^*_\gamma$.

Hence, when $Z_k = \hat{e}(g, \hat{g})^{ab_kc}$, meaning that algorithm $\mathcal{B}$'s input is sampled from $\mathcal{P}_{XBDH}$, then C^* is a valid encryption of M^* under the public key set PK^*_γ chosen by algorithm $\mathcal{A}$. On the other hand, when Z_k is uniform and independent in G_T, meaning that algorithm $\mathcal{B}$'s input is sampled from $\mathcal{R}_{XBDH}$, then C^* is independent of γ in the view of algorithm $\mathcal{A}$.

Phase 3. Algorithm $\mathcal{A}$ continues to adaptively query pk_{R_i} to Receiver_Corrupt oracle with the restriction that $pk_{R_i} \notin PK^*_0 \cup PK^*_1$. Algorithm $\mathcal{B}$ responds as in Phase 1.

Guess. Finally, algorithm $\mathcal{A}$ outputs a guess $\gamma' \in \{0, 1\}$. If $\gamma = \gamma'$, algorithm $\mathcal{B}$ outputs 1 meaning it wins the game. Otherwise, it outputs 0.

If $Z_k = \hat{e}(g, \hat{g})^{ab_kc}$, the simulation is perfect and algorithm $\mathcal{A}$ must satisfy $|\Pr[\gamma = \gamma'] = 1/2 + \epsilon$. On the other hand, if $Z_k \in_R \mathbb{G}_T$, the challenge ciphertext C^* contains no information on γ and thus $\Pr[\gamma = \gamma'] = 1/2$. The overall probability that algorithm $\mathcal{B}$ solves the X-DBDH problem correctly is thus $1/2 + \epsilon/2$.

This completes the proof of Theorem 2.

IND-CCA2 Secure Scheme. The result of Boneh and Katz (2005) can be applied to the above IND-CPA secure scheme, thus we can obtain a new scheme that is provably IND-CCA2 secure in random oracle by just making $\text{MAC}_i = H(ID_i \parallel M \parallel C_{3,i})$ as the addition to the ciphertext where ID_i is the identity token in the privileged receiver set. The only problem here is that the size of MAC_i is linear to the size of privileged receivers.

5 Conclusions and open problems

In this paper, we introduce a new primitive called gateway-based verifiable and anonymous broadcast encryption, which is a combination of verifiable encryption and anonymous broadcast encryption but we authorise an un-trusted gateway to verify whether the privileged receivers of an inbound ciphertext belong to the organisation without leaking the information of plaintext and privileged receivers. In order to solve such a problem, we propose a new notion: public key encryption and identity search (PKEIS), which is enlightened from private information retrieval (PIR) (Chor et al., 1995) and public key encryption with keyword search (PEKS) (Boneh et al., 2004). In PKEIS, the gateway stores all the trapdoors generated by receivers, and when it receives an inbound ciphertext, it decides to broadcast it or not by verifying this ciphertext with the stored trapdoors and its private key. Our proposed security model for a gateway-based verifiable and anonymous broadcast encryption system achieves good security, through which the gateway can complete verification without learning anything about the plaintext and the personal information of receivers even if it colludes with corrupted receivers. We also provide a concrete implementation of this new notion based on bilinear pairing and prove its security in the random oracle model. Although our construction is based on bilinear pairings, it is very efficient because encryption needs no pairing computation while decryption only needs one pairing computation.

In our proposed scheme, the size of ciphertext is linear to the number n of privileged receivers as $2 \cdot n + 1$. Hence, we leave as an open problem the question of building a gateway-based verifiable and anonymous broadcast encryption system which is CPA or CCA2 secure with shorter size or constant size of the ciphertext in random oracle model or standard model.

References

Asokan, N., Shoup, V. and Waidner, M. (1998) 'Optimistic fair exchange of digital signatures (extended abstract)', *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, Springer, pp.591–606.

Baek, J., Safavi-Naini, R. and Susilo, W. (2008) 'Public key encryption with keyword search revisited', *ICCSA (1)*, volume 5072 of *Lecture Notes in Computer Science*, Springer, pp.1249–1259.

Bao, F. (1998) 'An efficient verifiable encryption scheme for encryption of discrete logarithms', In *CARDIS*, volume 1820 of *Lecture Notes in Computer Science*, Springer, pp.213–220.

Barth, A., Boneh, D. and Waters, B. (2006) 'Privacy in encrypted content distribution using private broadcast encryption', *Financial Cryptography*, volume 4107 of *Lecture Notes in Computer Science*, Springer, pp.52–64.

Boneh, D. and Boyen, X. (2011) 'Efficient selective identity-based encryption without random oracles', *J. Cryptology*, Vol. 24, No. 4, pp.659–693.

Boneh, D., Crescenzo, G.D., Ostrovsky, R. and Persiano, G. (2004) 'Public key encryption with keyword search', In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, Springer, pp.506–522.

Boneh, D. and Franklin, M. (2001) 'Identity-based encryption from the weil pairing', *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, Springer-Verlag, pp.213–219.

Boneh, D., Gentry, C. and Waters, B. (2005) 'Collusion resistant broadcast encryption with short ciphertexts and private keys', *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, Springer, pp.258–275.

Boneh, D. and Katz, J. (2005) 'Improved efficiency for cca-secure cryptosystems built using identity-based encryption', *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, Springer, pp.87–103.

Camenisch, J. and Lysyanskaya, A. (2001) 'An efficient system for non-transferable anonymous credentials with optional anonymity revocation', *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, Springer, pp.93–118.

Camenisch, J. and Shoup, V. (2003) 'Practical verifiable encryption and decryption of discrete logarithms', *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, Springer, pp.126–144.

Chor, B., Goldreich, O., Kushilevitz, E. and Sudan, M. (1995) 'Private information retrieval', *FOCS*, IEEE Computer Science 1995, pp.41–50.

Delerablée, C. (2007) 'Identity-based broadcast encryption with constant size ciphertexts and private keys', *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, Springer, pp.200–215.

Delerablée, C., Paillier, P. and Pointcheval, D. (2007) 'Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys', *Pairing*, volume 4575 of *Lecture Notes in Computer Science*, Springer, pp.39–59.

Dodis, Y. and Fazio, N. (2003) 'Public key broadcast encryption for stateless receivers', *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, Springer, pp.61–80.

Fiat, A. and Naor, M. (1994) 'Broadcast encryption', *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, Springer, pp.480–491.

Franklin, M.K. and Reiter, M.K. (1995) 'Verifiable signature sharing', *EUROCRYPT*, volume 921 of *Lecture Notes in Computer Science*, Springer, pp.50–63.

Gentry, C. and Waters, B. (2009) 'Adaptive security in broadcast encryption systems (with short ciphertexts)', *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pp.171–188.

Goldwasser, S. and Micali, S. (1984) 'Probabilistic encryption', *J. Comput. Syst. Sci.*, Vol. 28, No. 2, pp.270–299.

Joux, A. (2000) 'A one round protocol for tripartite diffie-hellman', *ANTS*, volume 1838 of *Lecture Notes in Computer Science*, Springer-Verlag, pp.385–394.

Joux, A. and Nguyen, K. (2003) 'Separating decision diffie-hellman from computational diffie-hellman in cryptographic groups', *J. Cryptology*, VoL. 16, No. 4, pp.239–247.

Libert, B., Paterson, K.G. and Quaglia, E.A. (2011) 'Anonymous broadcast encryption', *IACR Cryptology ePrint Archive*, Vol. 2011, p.476.

Paillier, P. (1999) 'Public-key cryptosystems based on composite degree residuosity classes', *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, Vol. 1592, Springer, pp.223–238.

Poupard, G. and Stern, J. (2000) 'Fair encryption of rsa keys', *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, Springer, pp.172–189.

Stadler, M. (1996) 'Publicly verifiable secret sharing', *EUROCRYPT*, volume 1070 *Lecture Notes in Computer Science*, Springer, pp.190–199.