Round-Optimal Deniable Ring Authentication in the Presence of Big Brother

Rafael Dowsley[1], Goichiro Hanaoka[2], Hideki Imai[2], Anderson C. A. Nascimento[1]

[1] Department of Electrical Engineering, University of Brasília.
Campus Universitário Darcy Ribeiro,Brasília, CEP: 70910-900, Brazil,
E-mail: rafaeldowsley@redes.unb.br, andclay@ene.unb.br
[2] National Institute of Advanced Industrial Science and Technology (AIST)
1-18-13, Sotokanda, Chyioda-ku, 101-0021, Tokyo, Japan
E-mail: hanaoka-goichiro@aist.go.jp, h-imai@aist.go.jp

Abstract. In this work we propose a Deniable Ring Authentication scheme secure against a powerful Big Brother type of adversary and yielding an optimal number of communication rounds. Our scheme is based on an infra-structure assumption: the existence of verifiable Broadcast Encryption. Particularly, our solution can be instantiated by using the Broadcast Encryption protocol of Boneh, Gentry and Waters (CRYPTO 2005), resulting in a Deniable Ring Authentication protocol with constant message size.

Keywords: Deniable Ring Authentication, Broadcast Encryption, Big Brother.

1 Introduction

Digital Signatures [5] play a very important role in modern cryptography and are used in applications ranging from e-commerce to contract signing and secure email communication. In signature schemes one usually desires non-repudiation of messages: the receiver of a signed message can convince anyone that the sender actually signed that message. However, non-repudiation can be highly undesirable in many applications, for example, consider the situation when the receiver is paying for the authentication as in the case of software distribution. Deniable Authentication [9] has been proposed to cope with the cases where non-repudiation is a problem. It is an authentication protocol that convinces a receiver of the authenticity of a given message but does not allow he/she to prove this authenticity to other parties.

In [18], Naor combined Deniable Authentication and Ring Signatures [22] to obtain Deniable Ring Authentication in which it is possible to convince any receiver (a.k.a. verifier) that a sender (a.k.a. prover) that is member of some *ad hoc* subset of the parties (a ring) is authenticating a message M without revealing the identity of the prover and in such way that the verifier cannot convince other parties of this authentication. In the same paper, Naor provided a Deniable Ring Authentication protocol that assumes the existence of a Public Key Infrastructure, has four rounds of communication (since deniable authentication is stronger than ZKIP, it requires at least four rounds [12]) and has message size proportional to the number of ring's members. Naor also considered a different scenario where the protocol security must be proven against a powerful adversary which knows all the secret keys (called a Big Brother). This properly models attacks against Deniable Ring Authentication protocols based on Identity Based Encryption infrastructure [23, 2] and on Broadcast Encryption [10, 3], where a center provides keys to the users. In this stricter scenario, Naor obtained a secure protocol based on the existence of Identity Based Encryption infrastructure and Broadcast Encryption. The resulting protocol had six rounds of communication in total.

1.1 Motivation

We obtain a practical deniable ring authentication schemes secure against a Big Brother adversary yielding

- Optimal communication rounds (4 rounds).
- Constant message size.

Our solution assumes the existence of a Broadcast Encryption Infrastructure with one additional requirement: verifiability. This property can be found in the protocol of [3], for example. By assuming this infra-structure assumption, we present a Deniable Ring Authentication protocol that is secure in the presence of Big Brother and has four rounds of communication (round-optimal). When instantiated with the particular protocol proposed in [3], the resulting scheme has constant message size (i.e., it does not dependent on the number of ring's members). Thus assuming a Broadcast Encryption Infrastructure instead of a Public Key Infrastructure, we obtain a protocol that has the same four rounds as Naor's protocol for Public Key Infrastructure, but has constant message size instead of message size linear in the number of ring's members. In comparison to Naor's protocol that is secure in the presence of Big Brother, our protocol saves two rounds of communication.

1.2 Background

Deniable Authentication Deniable authentication is a stronger notion of zero knowledge interactive proof system in which the transcript of the interaction cannot be an evidence for enforcing non-repudiation of the sender. Note that in the security proof of ZKIPs we construct a simulator which can create the same transcript without using the witness, but this does not immediately imply deniability.

In Naor's paper "Deniable Ring Authentication" [18], he extends this notion to the context of Rivest, Shamir, Tauman's ring signature framework [22]. Ring signature is a very similar notion to Group signature except for:

- There exists no authority who can violate the user's anonymity.
- It should be setup free, i.e. we can use only existing infrastructures (e.g. PKI) which are used for common purposes, e.g. normal encryption or authentication.

Ring authentication is an interactive version of ring signatures.

Broadcast Encryption Broadcast Encryption (BE) is an encryption scheme in which the messages have multiple recipients. The first non-trivial solution was present in [10] by Fiat and Naor. Naor et al. [19] obtained a more efficient scheme to broadcast encrypted messages for large subsets of the system parties (i.e., only a small fraction of the users can be revoked). Other schemes of BE for large sets were proposed in [20, 14, 6, 7, 13]. Boneh, Gentry and Waters [3] constructed the first fully collusion resistant BE protocol that has constant size for ciphertext and decryption keys.

1.3 Outline of the Paper

In section 2 we define Deniable Ring Authentication, Broadcast Encryption and the computational assumptions used in this paper. In section 3 we present Naor's Deniable Ring Authentication protocol. We present our new protocol in section 4. Section 5 describes an efficient implementation of our protocol using Boneh-Gentry-Waters' Broadcast Encryption protocol. The conclusions are in section 6.

2 Preliminaries

In this section we introduce the definitions of Deniable Ring Authentication and Broadcast Encryption. We also present the computational assumptions used in the scheme of section 5. We closely follow the lines of [18] in the description of Deniable Ring Authentication and the lines of [3] in the description of Broadcast Encryption and computational assumptions.

2.1 Deniable Ring Authentication

In the Deniable Ring Authentication model, we assume that the set of possible provers (each having an unique id $i \in \{1, 2, \ldots, n\}$) has access to some infrastructure, either Public Key Infrastructure or Broadcast Encryption. A ring $\mathcal{S}$ is any subset of $\{1, 2, \ldots, n\}$.

An honest authenticator $\mathcal{P} \in \mathcal{S}$ executes an interactive protocol with a verifier $\mathcal{V}$ to authenticate a message M. We do not require that the verifier be part of the PKI/BE in question, we only require that the verifier and the prover know the public keys of all members of $\mathcal{S}$. We assume that messages are routed anonymously and that the message M to be authenticated is known previously by the prover and the verifier. The adversary $\mathcal{A}$ controls some parties of the system and knows the secret keys of all players (i.e., we assume the Big Brother model). The Deniable Ring Authentication should be complete and secure as defined below.

Let $\mathrm{Output}(\mathcal{S}, \mathcal{P}, \mathcal{V}, M)$ denote the result of executing the Deniable Ring Authentication protocol between the verifier $\mathcal{V}$ and the prover $\mathcal{P}$ that tries to prove that some member of the ring $\mathcal{S}$ is authenticating the message M. Similarly, let $\mathrm{View}(\mathcal{S}, \mathcal{P}, \mathcal{V}, M)$ denote the transcript of such execution.

Definition 1 (Completeness). *A Deniable Ring Authentication protocol is complete if for any valid ring $\mathcal{S}$, any honest prover $\mathcal{P} \in \mathcal{S}$, any honest verifier $\mathcal{V}$, and any message M, we have that* $\mathrm{Output}(\mathcal{S}, \mathcal{P}, \mathcal{V}, M) = \mathrm{accept}$ *with overwhelming probability.*

A Deniable Ring Authentication protocol is secure if it satisfy three requirements described below: Soundness (Existential Unforgeability), Source Hiding and Zero-Knowledge (Deniability).

To an adversary $\mathcal{A}$ trying to forge a message we associate the following game. $\mathcal{A}$ initially knows the identities and the public keys of all possible provers. It also chooses a target ring $\mathcal{S}$ (which we call *honest provers*), and is given all private keys of $\{1, ..., n\} \backslash \mathcal{S}$.

Query Phase: $\mathcal{A}$ adaptively chooses messages $M_1, M_2, \ldots$, rings $\mathcal{S}_1, \mathcal{S}_2, \ldots$ and honest provers $\mathcal{P}_1, \mathcal{P}_2, \ldots$ such that $\mathcal{P}_i \in \mathcal{S}_i$. The honest prover $\mathcal{P}_i$ executes the protocol authenticating the message M_i as been sent by some member of $\mathcal{S}_i$ ($\mathcal{A}$ controls the verifiers in these protocol executions).

Output Phase: $\mathcal{A}$ playing the role of the prover $\mathcal{P}$ chooses a message M, and executes the authentication protocol with an honest verifier $\mathcal{V}$. $\mathcal{A}$ wins if $\text{Output}(\mathcal{S}, \mathcal{P}, \mathcal{V}, M) = \text{accept}$, $(\mathcal{S}, M) \notin \{\mathcal{S}_i, M_i\}_{i=1,2,\ldots}$.

Definition 2 (Soundness - Existential Unforgeability). *A Deniable Ring Authentication protocol meets the Soundness requirement if for any probabilistic polynomial time adversary $\mathcal{A}$, we have that its winning probability in the previous game is negligible.*

To an adversary $\mathcal{A}$ trying to discover the identity of the prover we associate the following game. $\mathcal{A}$ initially knows the identities and the keys (both public and private) of all possible provers.

Challenge Phase: $\mathcal{A}$ (that is given *all* secrets in the system and plays the role of the verifier $\mathcal{V}$) chooses a ring $\mathcal{S}$, two honest provers $\mathcal{P}_0, \mathcal{P}_1 \in \mathcal{S}$ and a message M and sends this information to the challenger. The challenger randomly chooses $b \in \{0, 1\}$ and executes the authentication protocol with $\mathcal{P}_b$ as the prover.

Output Phase: $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$. $\mathcal{A}$ wins the game if $b' = b$.

Definition 3 (Source Hiding:). *A Deniable Ring Authentication protocol is Source Hiding, if for all probabilistic polynomial time adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the game above is negligibly close to $\frac{1}{2}$.*

Definition 4 (Zero-Knowledge - Deniability). *Consider an adversary $\mathcal{A}$ that initially knows the identities and the keys (both public and private) of all possible provers. A protocol meets the Zero-Knowledge requirement if for any transcript $\text{View}(\mathcal{S}, \mathcal{P}, \mathcal{V}, M)$ generated by a protocol execution in which a member $\mathcal{P}$ of $\mathcal{S}$ acted as the prover and authenticated the message M to $\mathcal{V}$, there exists a polynomial-time simulator $\mathcal{Z}$ that knowing only $\mathcal{S}, \mathcal{V}, M$ and the public keys generates an indistinguishable transcript (to every one but the sender).*

2.2 Broadcast Encryption

We present the definitions of our main tool for obtaining our result: Broadcast Encryption. We define Broadcast Encryption as a key encapsulation mechanism (the key generated by this protocol can be used in a One-time Symmetric Key Encryption protocol to encrypt the message M [4]). It is constituted of three algorithms:

Setup: Takes as input the number of parties n and outputs the public key PK and the private keys $d_1, \ldots, d_k$ (one for each party).

Encrypt: Takes as input the public key PK, a set $\mathcal{S} \subseteq \{1, \ldots, n\}$ of receivers of the broadcast and local randomness *coin*. It outputs a header H and a key of a symmetric encryption scheme K.
The key K is then used in a symmetric key encryption scheme to encrypt the message M obtaining a ciphertext L. The message broadcasted to the users is $C = (\mathcal{S}, H, L)$. We will denote the result of executing this algorithm by $C = \text{Enc}_{PK,\mathcal{S},coin}(M)$.

Decrypt: Takes as input a public key PK, an user id $i \in \{1, \ldots, n\}$, the private key d_i and a ciphertext C (constituted of a header H, a set $\mathcal{S} \subseteq \{1, \ldots, n\}$ and a ciphertext L of a symmetric encryption scheme). If $i \in \mathcal{S}$, it outputs a key of a symmetric encryption scheme K.
The key K is then used to decrypt L in the symmetric key encryption scheme obtaining the message M.

For all $\mathcal{S} \subseteq \{1, \ldots, n\}$ and all $i \in \mathcal{S}$, if the public and private keys were correctly generated by the Setup algorithm and the ciphertext was generated following the procedures of the Encrypt algorithm, then the output obtained by i executing correctly the Decrypt algorithm must be M with overwhelming probability.

We define security against a static adversary that selects the parties that it will attack before the execution of the Setup procedure. Security of a broadcast encryption scheme is defined as a game between a challenger and an adversary $\mathcal{A}$ who chooses some subset of the parties to attack and controls all the other parties. The game proceeds in the sequence below.

Initialization: $\mathcal{A}$ outputs a set $\mathcal{S}^* \subseteq \{1, \ldots, n\}$ of the parties that it will attack.

Setup: The challenger runs the setup algorithm of the scheme and obtains the public key and the privates keys. It sends to $\mathcal{A}$ the keys of the parties that $\mathcal{A}$ controls (i.e., all the parties that are not members of $\mathcal{S}^*$).

Query Phase 1: $\mathcal{A}$ adaptively sends decryption queries to the challenger. Each decryption query consists of triple $(i, \mathcal{S}, H)$ such that $\mathcal{S} \subseteq \mathcal{S}^*$ and $i \in \mathcal{S}$. The challenger executes the decryption procedure using the private key of party i and sends the output (i.e., the symmetric key) to $\mathcal{A}$.

Challenge: For the set $\mathcal{S}^*$, the challenger using the Encrypt algorithm generates a header H^* and a key K^* of the symmetric encryption scheme. It chooses randomly $b \in \{0, 1\}$, sets $K_b = K$ and chooses randomly a key K_{1-b}. It sends (H^*, K_0, K_1) to $\mathcal{A}$.

Query Phase 2: $\mathcal{A}$ adaptively sends decryption queries to the challenger. Each decryption query consists of triple $(i, \mathcal{S}, H)$ such that $\mathcal{S} \subseteq \mathcal{S}^*$, $i \in \mathcal{S}$ and $H \neq H^*$. The challenger executes the decryption procedure using the private key of party i and sends the output (i.e., the symmetric key) to $\mathcal{A}$.

Output: $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$. $\mathcal{A}$ wins the game if $b' = b$.

If the adversary is adaptive, we have to modify the above game as follows: (1) there is no Initialization phase, (2) the adversary can corrupt the parties adaptively, (3) $\mathcal{A}$ only fix the set of honest parties that it will attack, $\mathcal{S}^*$, in the Challenge phase.

We now define what it means for a broadcast encryption to be CCA2 secure [21].

Definition 5 (CCA2 Security). *The Broadcast Encryption protocol is CCA2 secure, if for all probabilistic polynomial time adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the game is negligibly close to $\frac{1}{2}$.*

2.3 Verifiability in Broadcast Encryption Schemes

Now we explain the definition of verifiability for broadcast encryption schemes that we consider in this paper. Verifiability is a property that allows the valid receivers to check that each recipient of the broadcasted encrypted message received the same message (i.e., it must be possible to verify the equality of the messages that each recipient decrypts). The definition is identical to that proposed by Hanaoka and Kurosawa in a recent paper [15]. There are two types of verifiability: public and private.

We say that a BE scheme is publicly verifiable if each valid receiver of the broadcasted message can verify without using its decryption key that the message received by each receiver is the same one.

For public verifiability, we define the advantage of an adversary $\mathcal{A}$ as

$$\begin{aligned}\mathrm{AdvVfy}_{\mathcal{A}} = \Pr[&\exists i, j \in \mathcal{S}, \mathrm{Decrypt}(PK, i, d_i, C) \neq \mathrm{Decrypt}(PK, j, d_j, C) |\\ &((d_1, \ldots, d_n), PK) \leftarrow \mathrm{Setup}(n); C \leftarrow \mathcal{A}((d_1, \ldots, d_n), PK)]\end{aligned}$$

Definition 6. *A Broadcast Encryption scheme is publicly verifiable if for all probabilistic polynomial time adversary $\mathcal{A}$,* $\mathrm{AdvVfy}_{\mathcal{A}}$ *is negligible.*

2.4 The Bilinear Diffie-Hellman Exponent Assumption

We use the same the notation as [16, 17, 2, 3] for bilinear maps and bilinear map groups. Let $\mathbb{G}$ and $\mathbb{G}_1$ be two (multiplicative) cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$. A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ with the following properties

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

A group $\mathbb{G}$ is bilinear if the group operation in $\mathbb{G}$ can be computed efficiently and there exists a group $\mathbb{G}_1$ and an efficiently computable bilinear map as described above.

We will use the computational assumption known as Bilinear Diffie-Hellman Exponent (BDHE) assumption [1, 3]. Let $\mathbb{G}$ be a bilinear group of prime order p (p is a security parameter), let g and h be random generators in $\mathbb{G}$ and let α be random in $\mathbb{Z}_p^*$. The decision l-BDHE states that given the vector

$$y_{g,\alpha,l} = (g^{(\alpha)}, g^{(\alpha^2)}, \ldots, g^{(\alpha^l)}, g^{(\alpha^{l+2})}, \ldots, g^{(\alpha^{2l})}) \in \mathbb{G}^{2l-1},$$

no probabilistic polynomial time algorithm has non-negligible advantage, in the security parameter p, in distinguishing the inputs $(g, h, y_{g,\alpha,l}, e(g^{(\alpha^{l+1})}, h))$ and $(g, h, y_{g,\alpha,l}, T)$, where T is a random element of $\mathbb{G}_1$. The advantage is computed over the random choice of g, h, α, T and the random bits used by the algorithm.

We will use henceforth the notation g_i to denote $g^{(\alpha^i)}$.

3 Previous Work: Naor's Scheme

In this section, as an important previous work, we review Naor's Deniable Ring Authentication protocol.

3.1 Naor's Idea

Naor started his scheme [18] from Dwork, Naor, and Sahai's authentication scheme [9] (which is an extension of Dolev, Dwork, Naor's scheme [8]). Dwork-Naor-Sahai scheme (with a single sender) is as follows:

Let (dk, PK) be a PKE key-pair of the prover. We assume that this PKE scheme is CCA2 [21]. Let M be the message to be authenticated. We assume that M is already known to both prover and verifier. We denote by $\mathrm{Enc}_{PK,coin}(M)$ the result of executing the PKE's encryption algorithm with message M, public key PK and local randomness $coin$. The authentication protocol for proving possession of dk is carried out as

Protocol 1 (Deniable Authentication)

1. $\mathcal{V}$ sends $C = \mathrm{Enc}_{PK,coin}(M||R)$ to $\mathcal{P}$ where R is a random number.
2. $\mathcal{P}$ decrypts C (using its private key dk) to obtain R and sends $C' = \mathrm{Enc}_{PK,coin'}(R)$ to $\mathcal{V}$.
3. $\mathcal{V}$ sends R and *coin* to $\mathcal{P}$. $\mathcal{P}$ re-encrypts $M||R$ using the same *coin* and verifies whether $\mathrm{Enc}_{PK,coin}(M||R) = C$ holds or not.
4. $\mathcal{P}$ sends R and $coin'$ to $\mathcal{V}$. $\mathcal{V}$ re-encrypts R using the same $coin'$ and verifies whether $\mathrm{Enc}_{PK,coin'}(R) = C'$ holds or not.

The above protocol is provably deniable and existentially unforgeable (for single-user setting).

In Sec. 5 of Naor's paper, he presents a ring authentication version of the above protocol. The essential idea of Naor's protocol is the following. For each member of $\mathcal{S}$, we run a independent parallel copy of the above protocol using the same R. However, there is a delicate point which has to be carefully handled in order to guarantee source hiding, and therefore, Naor also fixes this issue by splitting R into $R = R_1 + \ldots + R_n$ and encrypt them separately in Step 2. For a set $\mathcal{S}$ such that the members public keys are $PK_1, \ldots, PK_n$, the protocol is executed as follows

Protocol 2 (Naor's Deniable Ring Authentication Scheme)

1. $\mathcal{V}$ generates a random R and sends $(C_1, \ldots, C_n) = (\mathrm{Enc}_{PK_1,coin_1}(M||R), \ldots, \mathrm{Enc}_{PK_n,coin_n}(M||R))$ to $\mathcal{P}$.
2. $\mathcal{P}$ extracts R from C_i (using its secret key dk_i), chooses random $R_1, \ldots, R_n$ such that $R = R_1 + \ldots + R_n$ and sends $(C'_1, \ldots, C'_n) = (\mathrm{Enc}_{PK_1,coin'_1}(R_1), \ldots, \mathrm{Enc}_{PK_n,coin'_n}(R_n))$ to $\mathcal{V}$.
3. $\mathcal{V}$ sends R and $(coin_1, \ldots, coin_n)$ to $\mathcal{P}$. $\mathcal{P}$ verifies if the ciphertexts from Step 1 were properly formed.
4. $\mathcal{P}$ sends $(R_1, \ldots, R_n)$ and $(coin'_1, \ldots, coin'_n)$ to $\mathcal{V}$. $\mathcal{V}$ verifies if the ciphertexts from Step 2 were properly formed and if $R = R_1 + \ldots + R_n$.

The main issue of this scheme is that the communication complexity of the above scheme is linear in n, and this is considered not very efficient.

3.2 Modified Naor's Scheme from Broadcast Encryption

In Sec. 7 of the same paper, Naor addresses an interesting and efficient variant of the above-mentioned scheme by using Broadcast Encryption (BE). In this variant, we assume that there exists a dedicated infrastructure of BE (for contents distribution or something like that). If you want to prove that you are a member of a specific subset of the set of all users, you can use the BE system by replacing it with the PKE scheme in the above protocol. Then, we can have a deniable ring authentication with "setup-free" property since we already have a BE infrastructure (which would be commonly established in our real life).

However, in a strict sense, the above simple modification is not sufficient. Namely, in a BE system, there exists the *center* who knows all users' secrets, and he can violate any user's anonymity.

Here, we omit the concrete method for revealing anonymity. But, anyway, if one knows all users' secrets, he (i.e. *center*) can easily reveal it by using invalid ciphertexts at Step 1 of the protocol. More specifically (for the two parties case), if it (invalidly) consists of

$$(C1, C2) = (\mathrm{Enc}_{PK_1,coin_1}(M||R), \mathrm{Enc}_{PK_2,coin_2}(M||R'))$$

in Step 1 and if the returned message from $\mathcal{P}$ at Step 2 consists of

$$(C1', C2') = (\mathrm{Enc}_{PK_1, coin'_1}(R_1), \mathrm{Enc}_{PK_2, coin'_2}(R_2)),$$

such that $R_1 + R_2 = R'$, then $\mathcal{V}$ can immediately know that $\mathcal{P}$ has $dk2$.

Therefore, Naor further modifies the scheme for protecting against the above attack. The final scheme is presented in Sec. 7 on Naor's paper and has two extra rounds of communication in comparison to the protocol using PKE.

From the above results, we see that if we use the standard PKE infrastructure, transmission data size of the resulting ring authentication protocol becomes linear in the size of the ring (but its round complexity is optimal, i.e. four rounds), and if we use BE infrastructure, the round complexity becomes not optimal, i.e. six rounds (but transmission data can be shorter than that from the standard PKE infrastructure). Hence, a deniable ring authentication scheme (with setup-free property) which yields both constant transmission data size and optimal round complexity has not been known.

4 Our Scheme

In this section we introduce our deniable ring authentication protocol that is based on a broadcast encryption scheme.

4.1 Discussion: Essential Problem of the Naive Scheme

Here, we discuss the essential problem in the above faulty scheme (see section 3.2). The main point is that the verifier $\mathcal{V}$ can reveal $\mathcal{P}$'s anonymity by encrypting two different random numbers R and R' in the first step of the protocol and using the fact that it knows the private keys of all parties in order to discover which party encrypted the message sent to the verifier in the second step. So the verifier can violate the anonymity taking advantage of its ability of sending different messages to the members of the ring.

Naor solved this problem using a non-malleable commitment with respect to the encryptions of the first step, this way he protects the protocol against everyone (see [18] for details). But this approach adds two rounds of communication to the protocol.

We follow a different approach and use a broadcast encryption protocol that is verifiable to construct our deniable ring authentication protocol. In a verifiable broadcast encryption protocol it is possible to the prover to check if the verifier sent the same message to all recipients of the first step message, and so the attack above does not work any more.

4.2 Our Scheme

Our idea is very simple. We just use a verifiable BE system in the above protocol instead of an ordinary one in order to assure that the verifier sends the same message to all members of the ring in the first step of the protocol. Despite the simplicity of this idea, it solves the problem of the above faulty scheme since it forces the verifier to send the same message to all members of the ring. Interestingly, the Boneh-Gentry-Waters (BGW) BE system [3], which is considered as the "basic" BE scheme, originally has verifiability, and therefore, it is not very unnatural to assume a verifiable BE infrastructure.

Our protocol is similar to protocol 1, but it uses BE to the members of the ring instead of using public key encryption with prover's keys. Letting $\mathrm{Enc}_{PK,\mathcal{S},coin}(M)$ denote encryption of plaintext M for users $\mathcal{S}$ under public key PK of the underlying BE with local randomness *coin*, for any ring $\mathcal{S}$ such that the prover $\mathcal{P} \in \mathcal{S}$ (where $\mathcal{S}$ is a subset of all users), $\mathcal{P}$ can prove that he is member of $\mathcal{S}$ as follows:

Protocol 3 (Our Deniable Ring Authentication Protocol)

1. $\mathcal{V}$ sends $C = \mathrm{Enc}_{PK,\mathcal{S},coin}(M||R)$ to $\mathcal{P}$ where R is a random number.
2. $\mathcal{P}$ verifies if all the receivers of the broadcasted encrypted messages received the same message (using the verifiability of the broadcast encryption scheme) and stops the protocol if C is invalid. $\mathcal{P}$ decrypts C to obtain R and sends $C' = \mathrm{Enc}_{PK,\mathcal{S},coin'}(R)$ to $\mathcal{V}$.
3. $\mathcal{V}$ sends R and *coin* to $\mathcal{P}$. $\mathcal{P}$ re-encrypts $M||R$ using the same *coin* and checks whether $\mathrm{Enc}_{PK,\mathcal{S},coin}(M||R) = C$ holds or not.
4. $\mathcal{P}$ sends R and $coin'$ to $\mathcal{V}$. $\mathcal{V}$ re-encrypts R using the same $coin'$ and checks whether $\mathrm{Enc}_{PK,\mathcal{S},coin'}(R) = C'$ holds or not.

Since the deniability requirement implies that the deniable ring authentication protocols should be zero-knowledge, these protocols are stronger than ZKIP. Therefore these protocols requires at least four rounds, because ZKIP is impossible with three rounds [12]. Hence, the above protocol is round optimal.

4.3 Security of the Protocol

We now proof the security of the above protocol following the definitions of security described in section 2 and assuming that the Broadcast Encryption protocol used is CCA2 secure and verifiable and that the One-time Symmetric Key Encryption is CCA2 secure. I.e., we argue that the protocol meets the four requirements described previously: completeness, soundness, source hiding and deniability.

Theorem 1. *Assume that the Broadcast Encryption scheme is verifiable and CCA2 secure, and that the One-time Symmetric Key Encryption scheme is CCA2 secure. Then the Deniable Ring Authentication protocol presented above is secure according to the definitions of Sec. 2.*

We briefly sketch the proof of security of our scheme. Due to space constraints, we leave the complete proof to a full version of this paper.

Completeness: The Broadcast Encryption and the One-time Symmetric Key Encryption schemes used within our protocol must be correct in the sense that if the parties follow the procedures of the protocol, then the original message is decrypted correctly with overwhelming probability in the random choices of the procedure. Therefore the completeness requirement follows easily from this property of the Broadcast Encryption and the One-time Symmetric Key Encryption schemes, since the valid prover decrypts correctly the message $M||R$ and learns the correct R in first step with overwhelming probability, and from the fact that the rest of the execution for honest parties is correct if the first step is.

Soundness - Existential Unforgeability: As the Key Encapsulation Mechanism (i.e., the Broadcast Encryption) and the One-time Symmetric Key Encryption schemes are CCA2 secure, it follows that the Hybrid Encryption scheme is also CCA2 secure [4]. So the Hybrid Encryption scheme is non-malleable. The soundness of our protocol follows from the fact that the ciphertext $C' = \mathrm{Enc}_{PK,\mathcal{S},coin'}(R)$, which the prover sends in the second step of the protocol, is a non-malleable commitment to the random value R that the verifier sends in the first step. The access to the authentication oracle essentially means that the can make decryption queries in the Hybrid scheme. The adversary cannot take advantage of its access to the authentication oracle in order to forge a message with non-negligible probability, since this would imply in a non-negligible advantage against the Hybrid Encryption scheme contradicting the assumption that the Broadcast Encryption and One-time Symmetric Key Encryption schemes are CCA2 secure.

Source Hiding: The verifiability property of the Broadcast Encryption protocol guarantees that the actual prover can check if all possible provers in the current ring (i.e., all the receivers of the broadcasted encrypted message) received the same message in the first step of the protocol. Due to the verifiability of the Broadcast Encryption scheme, this tests fails only with negligible probability. Therefore the actual prover has the guarantee that all possible honest provers in the ring would encrypt the same value R (and so send indistinguishable messages) in the second step of the protocol with overwhelming probability, and so the source hiding property is satisfied by our protocol.

Zero-Knowledge - Deniability: We run the simulator with the prover $\mathcal{P}$ encrypting a random value R' in the second step of the protocol. If the verifier $\mathcal{V}$ opens R in the third step, the simulator rewind to just after step 1 and run the protocol again with the prover encrypting the correct value R in the second step.

To deal with verifiers that do not open the random value in third step, we use the fact that the Broadcast Encryption and the One-time Symmetric Key Encryption used are CCA2 secure, so the message in the second step is a secure commitment to R' and the verifier cannot learn non-negligible information about R'.

As our scheme meets the security requirements for deniable ring authentication schemes described in section 2, it is a secure deniable ring authentication scheme.

5 Efficient Implementation from BGW Protocol

5.1 BGW Protocol

Here, we review an efficient CCA2 secure variant of BGW protocol [3] which is secure under the Bilinear Diffie-Hellman Exponent (BDHE) assumption, which is due to Hanaoka and Kurosawa [15]. As mentioned in [3], the BGW can also be modified using a signature scheme and a collision resistant hash function to become CCA2 secure.

Let $\mathbb{G}$ and $\mathbb{G}_1$ be multiplicative cyclic groups with prime order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be a bilinear mapping such that for all $a, b \in \mathbb{Z}$ and $u, v \in \mathbb{G}$, we have that $e(u^a, v^b) = e(u, v)^{ab}$ and for a generator $g \in \mathbb{G}$ we have that $e(g, g) \neq 1$.

The CCA2 secure variant of BGW protocol is as follows:

Setup: Choose $\ell \in \mathbb{N}$ such that ${}_{2\ell}C_\ell \geq p$. Let $\mathbb{G}$ be a bilinear group with prime order p. Pick a random generator $g \in \mathbb{G}$ and random $\alpha \in \mathbb{Z}_p$. Compute $g_i = g^{(\alpha^i)} \in \mathbb{G}$ for $i = 1, 2, ..., n + 2\ell, n + 2\ell + 2, ..., 2(n + 2\ell)$. Pick a injective mapping $\mathsf{INJ} : \mathbb{G} \to \mathcal{P}$, where $\mathcal{P}$ is the set of all $\Delta\mathcal{S} \subseteq \{n + 1, ..., n + 2\ell\}$ with $|\Delta\mathcal{S}| = \ell$. Pick a random $\gamma \in \mathbb{Z}_p$ and set $v = g^\gamma \in \mathbb{G}$. Set $Z = e(g_{n+2\ell+1}, g)$ where $g_{n+2\ell+1} = g^{\alpha^{n+2\ell+1}}$. The public key is $PK = (g, g_1, ..., g_{n+2\ell}, g_{n+2\ell+2}, ..., g_{2(n+2\ell)}, v, Z, \mathsf{INJ})$, and the decryption keys for user $i \in \{1, ..., n\}$ is set as $d_i = g_i^\gamma \in \mathbb{G}$. Output $(d_1, ..., d_n, PK)$.

Encrypt: Pick a random $t \in \mathbb{Z}_p$, and set $K = Z^t \in \mathbb{G}_1$. Compute $\Delta\mathcal{S} = \mathsf{INJ}(g^t)$, and output (ψ, K) where $\psi = (g^t, (v \cdot \prod_{j \in \mathcal{S} \cup \Delta\mathcal{S}} g_{n+2\ell+1-j})^t) \in \mathbb{G}^2$.

Decrypt: Letting $\psi = (C_0, C_1)$, compute $\Delta\mathcal{S} = f(C_0)$, and check whether $e(g, C_1) \overset{?}{=} e(v \cdot \prod_{j \in \mathcal{S} \cup \Delta\mathcal{S}} g_{n+2\ell+1-j}, C_0)$, and if not, output $\perp$. Otherwise, output $K = e(g_i, C_1)/e(d_i \cdot \prod_{j \in \mathcal{S} \cup \Delta\mathcal{S} \setminus \{i\}} g_{n+2\ell+1-j+i}, C_0)$.

The security of the above scheme is addressed as follows:

Theorem 2. *Let $\mathbb{G}$ be a bilinear group with prime order p, and* INJ *be an injective mapping. Then, for any positive integers n, the above scheme is CCA2 secure under the BDHE assumption on $\mathbb{G}$ such that ${}_{2\ell}C_\ell \geq p$.*

As explained in [15] (see the full version of the paper), BGW protocol can be slightly modified to add verifiability. To add verifiability to their protocol, we only have to check in the beginning of the decryption procedure if

$$e(g, C_1) \overset{?}{=} e(v \cdot \prod_{j \in \mathcal{S} \cup \Delta\mathcal{S}} g_{n+2\ell+1-j}, C_0)$$

and output an error symbol if they are not equal.

By applying this scheme to our generic construction in the previous section, we have the *first* deniable ring authentication with constant transmission data size and optimal round complexity.

5.2 Implementing Our Protocol from BGW

Assuming that the parties have access to a Broadcast Encryption infrastructure which is based on (CCA2 secure variant of) BGW protocol, one can carry out Deniable Ring Authentication protocol as follows:

1. $\mathcal{V}$ picks a random $t \in \mathbb{Z}_p$ and computes the header H and the symmetric key K. $\mathcal{V}$ uses an One-Time Symmetric Key Encryption protocol with the key K and randomness Z to encrypt $M||R$ (where R is a random number) and obtain a ciphertext L. It sends $C = (\mathcal{S}, H, L)$ to $\mathcal{P}$.
2. $\mathcal{P}$ verifies if $e(g, C_1) \overset{?}{=} e(v \cdot \prod_{j \in \mathcal{S} \cup \Delta\mathcal{S}} g_{n+2\ell+1-j}, C_0)$ and stops the protocol if it is not equal. $\mathcal{P}$ decrypts H to obtain the symmetric key K, and then uses K to decrypt $M||R$ and obtain R. $\mathcal{P}$ picks a random $t' \in \mathbb{Z}_p$ and computes the header H' and the symmetric key K'. $\mathcal{P}$ encrypts R uses an One-Time Symmetric Key Encryption protocol with the key K' and randomness Z' to encrypt R and obtain a ciphertext L'. It sends $C' = (\mathcal{S}, H', L')$ to $\mathcal{V}$.

3. $\mathcal{V}$ sends R, t and Z to $\mathcal{P}$. $\mathcal{P}$ re-encrypts $M||R$ using the same t and Z. It checks whether the result is equal to $C = (\mathcal{S}, H, L)$ or not, and stops if it is not.
4. $\mathcal{P}$ sends R, t' and Z' to $\mathcal{V}$. $\mathcal{V}$ re-encrypts R using the same t' and Z'. It checks whether the result is equal to $C' = (\mathcal{S}, H', L')$ or not, and stops if it is not.

Theorem 3. *The above protocol is a secure Deniable Ring Authentication protocol under the BDHE assumption on $\mathbb{G}$ in the sense of the definitions of Sec. 2.*

It should be noticed that the above protocol requires only constant transmission data size (which is independent of the size of the ring) and four rounds (which is optimal).

5.3 A Drawback of this Scheme

There is one unsolved issue in our scheme. Namely, its security can be proven against only static adversaries since the BGW scheme has only static security. More specifically, before the setup phase, the adversary has to first commit to a subset $\mathcal{S}^*$ for which it wants to compromise the soundness property (the goal of adversary is to pretend to be a member of $\mathcal{S}^*$ without using no valid decryption keys). Obviously, this security notion is weaker than the adaptive adversarial model in which the adversary can adaptively choose $\mathcal{S}^*$ after the setup phase.

One possible solution would be to prove the verifiability of some adaptively secure BE system with constant ciphertext size. One possible candidate is the scheme recently proposed by Gentry and Waters [11].

6 Conclusion

We have constructed a practical deniable ring authentication schemes which has optimal communication rounds and constant message size. To the best of our knowledge, this is the first solution meeting those properties against a Big Brother like adversary. In our scheme we assumed the existence of a verifiable Broadcast Encryption protocol. Our solution can be implemented using the Boneh-Gentry-Waters' protocol [3]. Since their protocol has been proven secure only against adversaries that selects the set of parties that it wants to attack prior to the setup phase of the broadcast encryption scheme, this implementation of our scheme can be proven secure only against this static type of adversary.

One open problem is to prove the verifiability of some practical broadcast encryption protocol that has constant message size and that has been proven secure against adversaries that adaptively selects the participants that it wants to attack. Such protocol can be used with our construction to obtain a deniable ring authentication protocol secure against adversaries that adaptively corrupts the parties.

We also suggest as a future research direction to investigate further uses of broadcast encryption with the extra verifiability property.

References

1. D. Boneh, X. Boyen, E. Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. EUROCRYPT 2005, pp. 440–456.
2. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. CRYPTO 2001, LNCS 2139, Springer, 2001, pp. 213-229.
3. D. Boneh, C. Gentry, B. Waters. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. CRYPTO 2005, pp. 258–275.
4. R. Cramer, V. Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. SIAM Journal of Computing 33:167-226, 2003.
5. W. Diffie, M.E. Hellman. New Directions in Cryptography. IEEE Trans. on Info. Theory, IT-22 (Nov. 1976), pp. 644-654.
6. Y. Dodis and N. Fazio. Public Key Broadcast Encryption for Stateless Receivers. In Proceedings of the Digital Rights Management Workshop 2002, volume 2696 of LNCS, pages 6180. Springer, 2002.
7. Y. Dodis and N. Fazio. Public Key Broadcast Encryption Secure Against Adaptive Chosen Ciphertext Attack. In Workshop on Public Key Cryptography (PKC), 2003.
8. D. Dolev, C. Dwork, M. Naor. Non-malleable Cryptography. SIAM J. Comput. 30(2), pp. 391–437 (2000).
9. C. Dwork, M. Naor, A. Sahai. Concurrent Zero-Knowledge. STOC 1998, pp. 409–418.
10. A. Fiat, M. Naor: Broadcast Encryption. CRYPTO 1993, pp. 480–491.
11. C. Gentry, B. Waters. Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). EUROCRYPT 2009: 171-188.
12. O. Goldreich, Y. Oren. Definitions and Properties of Zero-Knowledge Proof Systems. J. Cryptology 7(1), pp. 1–32 (1994).
13. M. T. Goodrich, J. Z. Sun, R. Tamassia. Efficient Tree-based Revocation in Groups of Low-state Devices. CRYPTO 2004, volume 2204 of LNCS, 2004.
14. D. Halevy and A. Shamir. The LSD Broadcast Encryption Scheme. CRYPTO 2002, pp. 47-60.
15. G. Hanaoka, K. Kurosawa. Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption. ASIACRYPT 2008, pp. 308–325. Full version available at http://eprint.iacr.org/2008/211.
16. A. Joux. A One Round Protocol for Tripartite Diffie-Hellman. ANTS 2000: 385-394.
17. A. Joux, K. Nguyen. Separating Decision Diffie-Hellman from Computational Diffie-Hellman in Cryptographic Groups. J. Cryptology 16(4): 239-247 (2003).
18. M. Naor. Deniable Ring Authentication. CRYPTO 2002, pp. 481–498.
19. D. Naor, M. Naor, J. Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. CRYPTO 2001: pp. 41–62.
20. M. Naor, B. Pinkas. Efficient Trace and Revoke Schemes. Financial Cryptography 2000: pp. 1–20.
21. C. Rackoff, D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. CRYPTO 1991, pp. 433–444.
22. R. L. Rivest, A. Shamir, Y. Tauman. How to Leak a Secret. ASIACRYPT 2001, pp. 552–565.
23. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. CRYPTO84, LNCS 196, Springer, 1985, pp. 47-53.