社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2008－CSEC－42 (41)
2008 / 7 / 25

否認可能なリング認証に基づくグループ鍵生成法

タニグチ エリオット† 山本 博資†

† 東京大学大学院新領域創成科学研究科 〒277-8561 千葉県柏市柏の葉 5-1-5
E-mail: †taniguchi@it.k.u-tokyo.ac.jp, ††Hirosuke@ieee.org

あらまし 2002 年に M. Noar は否認可能なリング認証方式を提案した．本稿ではその手法をグループ鍵生成に拡張した 3 つの方式を提案する．どの方式においても，Alice は，Alice が自由に選択したグループに対して秘密鍵共有の要求を出すことができ，グループ内の任意でかつ匿名である Bob の認証の下でグループ鍵を安全に生成できる．認証した Bob 以外の他のグループメンバーも，安全性を確認して秘密鍵を共有することができる．

キーワード グループ鍵共有法，リング認証，否認可能な認証

Ring Authenticated Group Key Establishment with Member Repudiation

Elliot T. TANIGUCHI† and Hirosuke YAMAMOTO†

† Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba, 277-8561, Japan
E-mail: †taniguchi@it.k.u-tokyo.ac.jp, ††Hirosuke@ieee.org

Abstract In this paper, we first extend the Deniable Ring Authentication Protocol proposed by M. Naor to include key establishment. Then, two new secure ring authenticated group key establishment protocols with member repudiation will be proposed where some initiator Alice can generate a secret key between Alice and all members of a group chosen by Alice under the authentication of some anonymous Bob in the group. All of these protocols allow all members of the group to listen and communicate on the channel created by the key request initiator Alice and Bob.

Keywords Group Key Establishment, Ring Authentication, Deniable Authentication

1. Introduction

There are many instances when one or both parties wish to remain anonymous due to the nature of the discussion or the perceived risks being associated with the discussion. On the other hand, blindly speaking with completely arbitrary members has an adverse affect on the reliability of the information or feedback. A middle ground is required to initiate a discussion with several knowledgeable members of the field of interest. Assuming the initiator also belongs to this group, all members of this party have assurances that even though the identity of the other participant is hidden, the other party will have valuable information to contribute.

In addition, there are situations where it is desirable to have secure group channel created rather than a secure point-to-point channel. For instance, a secure group channel is efficient in transmitting information from the sender to each member of the group rather than relying on other users to relay the information. Also, the open nature of the communication allows for better collaboration within the group and the monitoring of other group members.

In this paper, we present three versions of an anonymous initiator protocol for group key establishment. This protocol imposes no restrictions on the key establishment requester Alice. In fact, Alice does not even need to have a public-private key pair although access to the public keys of other users is required. This scenario is particularly attractive when Alice prefers to remain completely anonymous or when Alice has not yet established a public-private key pair.

We will refer to this situation as the "Anonymous Counterparty Negotiation" case. Here, some anonymous seller wishes to negotiate the sale of a large block of common stock to a select list of potential buyers. Alice wishes to only negotiate with her set of potential buyers and prefers to retain deniability. This helps Alice prevent a potential backlash of other majority stock holders who may see their stock price valua-

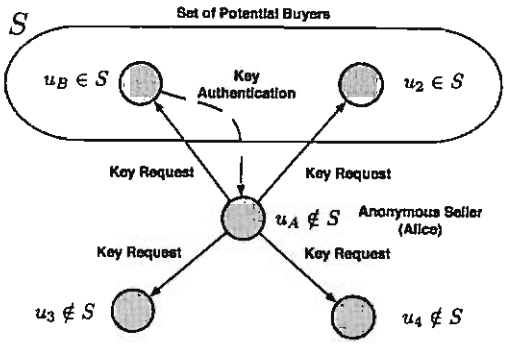

Figure 1 "Anonymous Counterparty Negotiation" Case

tions drop. Some interested buyer Bob within Alice's buyer list wishes to purchase the stock but may wish to negotiate the terms and condition. In addition, Bob would also prefer to keep his identity hidden to ensure fairness and deniability. The main goal here is to protect both the seller's identity and the buyer's identity yet establish a symmetric session key for repeated use. This is also useful in cases where if the seller decides to also include himself in the recipient user set S, other users in user set S may guess the seller's identity. The communication model is shown in Fig.1.

The first scheme called Naive Group Key Establishment (section 3.1) is applies Naor's deniable ring authentication scheme [8] directly to the group key establishment problem. The second scheme (section 3.2) uses a general one-way hash function which has the effect of reducing the number of non-malleable public-key encryptions of the first scheme. The final scheme (section 3.3) is based on the discrete logarithm and has similar computational requirements of the second scheme with the added benefit that the symmetric key contributed by Bob is hidden from non-members even if Alice's proposed session key is weak.

2. Related Work

In 1997, R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky studied Deniable Encryption [1] where a sender Alice can convince some third party Eve that the ciphertext was generated from a different message M' and seed ρ'. More specifically, after Alice sends ciphertext $C = \text{Encrypt}_{ek}(M,\rho)^{(\text{note1})}$, Alice can still find some other "plausible" message M' and seed ρ' such that $C = \text{Encrypt}_{ek}(M',\rho')$. Thus, Alice retains deniability to any third party Eve of the encrypted message M.

In 2001, R.L. Rivest, A. Shamir, and Y. Tauman proposed a ring signature scheme [10] which only reveals an arbitrary set of possible signers. This is similar to groups signatures except that no interaction between signers is required, no group managers are need, and non-revocation procedures are needed. For typical group signatures, the group manager is required to generate the secret keys for each group member in addition the (public) group verification key. In addition, group signature revocation schemes are rather inefficient and information about the revoked group members must also be transmitted for revocation.

M. Naor later published a Deniable Ring Authentication Protocol [8] in 2002 which allows any arbitrary user Alice to interactively authenticate a message M by some anonymous user in an arbitrary user set S. The only assumption used is that all users in user set S must have a public-private key pair of some algorithm immune against chosen ciphertext in the post-processing mode (CCA2) public key encryption scheme. The main goal of the above paper is to allow some verifier Alice to authenticate a message while allowing the anonymous prover Bob of set S to later repudiate the authentication to Alice. Thus, Bob retains deniability even after authenticating the message to Alice. In other words, the authentication performed by Bob is non-transferable to a third-party even if the secret keys of all members of set S are revealed. W. Susilo and Y. Mu [12] [13] later proposed a non-interactive deniable ring authentication scheme using Chameleon Hash functions [2] [7].

2.1 Non-Malleable Public Key Encryption

In this paper, we will assume that there exists a non-malleable public key encryption / decryption algorithm immune against chosen ciphertext in the post-processing mode (CCA2). Under this assumption, the encryption algorithm must be probabilistic and a random number generator will be required. This means that for every encryption of the same message (with different random seed), the ciphertext appears random.

KeyGen(λ). The key generation algorithm computes a random public (encryption) key ek and secret (decryption) key dk pair using some security parameter λ.

$$(ek, dk) \leftarrow \text{KeyGen}(\lambda) \qquad (1)$$

Encrypt(ek, M, ρ). The encryption algorithm encrypts a message M using encryption key ek and random seed $\rho \in \{0,1\}^l$. The ciphertext of message M and seed ρ is denoted by C_M^ρ.

$$C_M^\rho \leftarrow \text{Encrypt}_{ek}(M,\rho) \qquad (2)$$

Decrypt(dk, C_M^ρ). The decryption algorithm decrypts the ciphertext C_M^ρ using *only* the secret key dk. If the secret

(note1) : $\text{Encrypt}_{ek}(M,\rho)$ is a probabilistic public key encryption algorithm with encryption key ek, message M, and random seed ρ.

(decryption) key dk is incompatible with the ciphertext C_M^ρ, then "reject" or $\perp$ will be output. Otherwise, the original message will be output.

$$M \leftarrow \text{Decrypt}_{dk}(C_M^\rho) \tag{3}$$

The probabilistic nature of non-malleable PKE will be used in Naor's deniable ring authentication scheme (section 2.2) and each of our proposed group key establishment (section 3.1 - 3.3). The reason is that although the random seed ρ is used for encryption, it is not required for decryption and the random seed ρ can also be used for message commitment. For instance, if we require that the encryption scheme be binding, then for a carefully chosen encryption key ek each ciphertext C_M^ρ should have a unique message M and some ρ such that $C_M^\rho = \text{Encrypt}_{ek}(M, \rho)$. Thus, the terms random seed and commitment value will be used interchangeably throughout the rest of this paper. An efficient implementation of the above protocol is the Cramer-Shoup Cryptosystem [3] based on the Diffie-Hellman decision problem.

2.2 Deniable Ring Authentication

In this subsection, we will introduce the deniable ring authentication scheme proposed by M. Naor [8] which enables the authentication of messages with user repudiation. This protocol is an interactive protocol between a single verifier and multiple "anonymous" provers. This scheme is constructed using only CCA2 secure public key encryption schemes and makes use of the ciphertext randomization seeds as commitment values.

The basic Deniable Ring Authentication scheme will be summarized below. For any arbitrary user set $S = \{u_1, u_2, ..., u_n\}$, Alice u_A will attempt to authenticate message M with some anonymous prover Bob $u_B \in S$ by following the protocol below.

Step 1: Alice's Verification Request. Alice will randomly choose $r \in \{0,1\}^{l_1}$ and $\rho_i \in \{0,1\}^{l_2}$. Then, Alice will broadcast the following ciphertexts.

$$\left\{c_i \leftarrow \text{Encrypt}_{ek_i}(M \parallel r, \rho_i)\right\}_{u_i \in S} \tag{4}$$

Step 2: Bob's Commitment. Bob will decrypt some ciphertext c_B which he has the corresponding decryption key dk and compute M and r.

$$M \parallel r \leftarrow \text{Decrypt}_{dk_B}(c_B) \tag{5}$$

Bob will randomly choose $r_i \in \{0,1\}^{l_1}$ such that $r = \oplus_{i=1}^n r_i$ and $\sigma_i \in \{0,1\}^{l_2}$. Bob will compute the send the following commitment values.

$$\left\{d_i \leftarrow \text{Encrypt}_{ek_i}(r_i, \sigma_i)\right\}_{u_i \in S} \tag{6}$$

Step 3: Alice Opens Commitment. Alice will open r and $\rho_1, ..., \rho_n$ to show that all the ciphertexts $\{c_i\}$ contain the same plaintext M and r.

Step 4: Bob Verifies Commitment. Bob verifies that all the initial ciphertexts $\{c_i\}$ were generated properly. This can be done by checking the following for all i.

$$\forall i, c_i \stackrel{?}{=} \text{Encrypt}_{ek_i}(M \parallel r, \rho_i) \tag{7}$$

If a single ciphertext c_{i^*} fails, STOP. Otherwise, Bob will broadcast his commitment values $r_1, ..., r_n, \sigma_1, ..., \sigma_n$.

Step 5: Alice Verifies Commitment. Alice will verify the following for all $i \in \{1, ..., n\}$ proving that Bob knew r before Step 3.

$$r \stackrel{?}{=} \oplus_{j=1}^n r_j \tag{8}$$

$$\forall i, d_i \stackrel{?}{=} \text{Encrypt}_{ek_i}(r_i, \sigma_i) \tag{9}$$

3. Proposed Schemes

3.1 Naive Group Key Establishment

In this subsection, we will propose a naive scheme which allows a group of users to anonymously establish a group key with member repudiation using Naor's Deniable Ring Authentication Scheme described earlier. To retain deniability, each user $u_i \in S$ cannot rely on the authentication performed by another user $u_j \in S$. Thus, the protocol must allow for *multiple* users to authenticate the channel request. As a result, we define Bob_i to be the the i-th user $u_i \in S$ to participate in the group key establishment. The protocol explained here ensures that Alice's identity remains entirely anonymous. In fact, Alice is not even required to have a permanent public-private key pair (ek, dk). Thus, Alice's could be any member who has access to the public keys of all users in user set S. In addition, an untraceable session identity will also be generated for each participating member.

Step 1: Alice's Key Establishment Request. Alice u_A first chooses a user set S and generates a (temporary) public-private key pair (ek_A, dk_A) using security parameter λ.

$$(ek_A, dk_A) \leftarrow \text{KeyGen}(\lambda) \tag{10}$$

After choosing random string r and random symmetric session key K_A, Alice broadcasts the following.

$$\left\{c_j \leftarrow \text{Encrypt}_{ek_j}(r \parallel ek_A \parallel K_A \parallel ID_A, \rho_j)\right\}_{u_j \in S} \tag{11}$$

Step 2: Bob_i's Commitment. All users $u_j \in S$ will compute r, ek_A, K_A, ID_A using their secret decryption key

dk_j.

$$r^{(j)} \parallel ek_A^{(j)} \parallel K_A^{(j)} \parallel ID_A^{(j)} \leftarrow \text{Decrypt}_{dk_j}(c_j) \quad (12)$$

Interested user Bob_i ($u_{B_i} \in S$) (or set of users $\{\text{Bob}_{i_1}, ..., \text{Bob}_{i_m}\}$) will randomly choose $\{r_j^{(i)}\}$ such that $r^{(i)} = \oplus_{j=1}^{n+1} r_j^{(i)}$ and securely broadcasts the following commitment on r to all users $u \in S^+ \equiv S \bigcup \{u_A\}$.

$$\left\{ d_j^{(i)} \leftarrow \text{Encrypt}_{ek_j}(r_j^{(i)} \parallel K_B^{(i)} \parallel ID_i, \sigma_j^{(i)}) \right\}_{u_j \in S^+} \quad (13)$$

Step 3: Alice Opens Commitment. Alice sends the commitment values $\vec{\rho} = (\rho_1, ..., \rho_n)$ using the secure group channel key K_A for all users $u \in S^+$.

$$C_A \leftarrow E_{K_A}(\vec{\rho}) \quad (14)$$

Step 4: Bob_i Verifies Alice's Commitment. Bob_i u_{B_i} will decode the commitment values $\{\rho_i\}$ and verify that the initial key establishment request $\{c_j\}$ was generated properly. Specifically, Bob_i will check the following.

$$\forall j, c_j \stackrel{?}{=} \text{Encrypt}_{ek_j}(r^{(i)} \parallel ek_A^{(i)} \parallel K_A^{(i)} \parallel ID_A^{(i)}, \rho_j) \quad (15)$$

If a single ciphertext c_j- fails, STOP. Otherwise, Bob will securely send the commitment values $\vec{r}^{(i)} = (r_1^{(i)}, ..., r_n^{(i)})$ and $\vec{\sigma}^{(i)} = (\sigma_1^{(i)}, ..., \sigma_n^{(i)})$ to all users $u \in S^+$.

$$C_B^{(i)} \leftarrow E_{K_A}(\vec{r}^{(i)} \parallel \vec{\sigma}^{(i)}) \quad (16)$$

Step 5: Alice Verifies Bob_i's Commitment. Alice will decode the ciphertexts $C_B^{(i)}$ and $d_A^{(i)}$.

$$\vec{r} \parallel \vec{\sigma} \leftarrow D_{K_A}(C_B^{(i)}) \quad (17)$$

$$r_A^{(i)} \parallel K_B^{(i)} \parallel ID_i \leftarrow \text{Decrypt}_{ek_A}(d_A^{(i)}) \quad (18)$$

Then, Alice will verify the following for all j.

$$r \stackrel{?}{=} \oplus_{j=1}^{n+1} r_j \quad (19)$$

$$\forall j, d_j^{(i)} \stackrel{?}{=} \text{Encrypt}_{ek_j}(r_j^{(i)} \parallel K_B^{(i)} \parallel ID_j, \sigma_j) \quad (20)$$

Suppose $d_j^{(i^*)}$ passes for all j, then Alice can compute the authenticated group key $K_G \leftarrow \mathcal{H}(K_A \parallel K_B^{(i_1^*)} \parallel \cdots \parallel K_B^{(i_m^*)})$ where $\mathcal{H}$ is a one-way hash function and $(i_1^*, ..., i_m^*) = \arg_{i^*} \text{Sort}\{ID_{i^*}\}$.

Step 6: Bob_j Verifies Bob_i's Commitment. Bob_j will decode the ciphertexts $C_B^{(i)}$ and $d_A^{(i)}$

$$\vec{r} \parallel \vec{\sigma} \leftarrow D_{K_A}(C_B^{(i)}) \quad (21)$$

$$r_A^{(i)} \parallel K_B^{(i)} \parallel ID_i \leftarrow \text{Decrypt}_{ek_A}(d_A^{(i)}) \quad (22)$$

Then, Bob_j will verify the following for all l.

$$r^{(j)} \stackrel{?}{=} \oplus_{l=1}^{n+1} r_l \quad (23)$$

$$\forall l, d_l^{(i)} \stackrel{?}{=} \text{Encrypt}_{ek_l}(r_l^{(i)} \parallel K_B^{(i)} \parallel ID_i, \sigma_l) \quad (24)$$

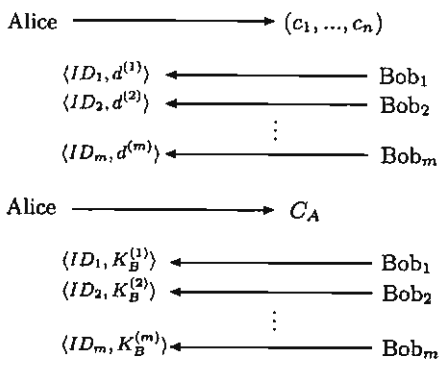

Figure 2 Hash Function Based Key Exchange

Suppose $d_l^{(i^*)}$ passes for all l, then Bob_j can compute the authenticated group key $K_G \leftarrow \mathcal{H}(K_A \parallel K_B^{(i_1^*)} \parallel \cdots \parallel K_B^{(i_m^*)})$ where $\mathcal{H}$ is a one-way hash function and $(i_1^*, ..., i_m^*) = \arg_{i^*} \text{Sort}\{ID_{i^*}\}$.

3.2 Hash Function Based Scheme

In this subsection, we will present a hash function based group key establishment protocol which allows multiple users $u_{B_i} \in S$ to efficiently generate an authenticated group key and untraceable session identity. A diagram illustrating the information exchange Alice and Bob is shown in Fig. 2.

Step 1: Alice's Key Establishment Request. Alice u_A first chooses a user set S, random string r, symmetric session key K_A, and session identity ID_A.

$$\left\{ c_j \leftarrow \text{Encrypt}_{ek_j}(r \parallel K_A \parallel ID_A, \rho_j) \right\}_{u_j \in S} \quad (25)$$

Step 2: Bob_i's Commitment. All users $u_j \in S$ will compute $r \parallel K_A \parallel ID_A$ using their secret decryption key dk_j.

$$r^{(j)} \parallel K_A^{(j)} \parallel ID_A^{(j)} \leftarrow \text{Decrypt}_{dk_j}(c_j) \quad (26)$$

Interested user Bob_i ($u_{B_i} \in S$) (or set of users $\{\text{Bob}_{i_1}, ..., \text{Bob}_{i_m}\}$) will randomly choose a session identity ID_i and random symmetric key K_B. Bob_i will then broadcast the following using a one-way hash function $\mathcal{H}$ and commitment value $r^{(i)}$ decoded earlier.

$$\langle ID_i, d^{(i)} \rangle \quad (27)$$

where,

$$d^{(i)} \leftarrow \mathcal{H}(r^{(i)} \parallel K_B^{(i)} \parallel ID_i) \quad (28)$$

Step 3: Alice Opens Commitment. Alice sends the commitment values $\vec{\rho} = (\rho_1, ..., \rho_n)$ using the secure group channel key K_A for all users $u \in S^+$.

$$C_A \leftarrow E_{K_A}(\vec{\rho}) \quad (29)$$

Step 4: Bob$_i$ Verifies Alice's Commitment. Bob$_i$ will decode Alice's commitment values $\{\rho_j\}$ and verify that the initial group key establishment request $\{c_j\}$ was generated properly. Specifically, Bob$_i$ will check the following.

$$\forall j, c_j \stackrel{?}{=} \text{Encrypt}_{ek_j}(r^{(i)} \parallel K_A^{(i)} \parallel ID_A^{(i)}, \rho_j) \tag{30}$$

If a single ciphertext c_{j^*} fails, STOP. Otherwise, Bob$_i$ will send his commitment values $\langle ID_i, K_B^{(i)} \rangle$ to all users $u \in S^+$ using symmetric key K_A.

Step 5: Alice Verifies Bob$_i$'s Commitment. Alice will verify the following for all i.

$$d^{(i)} \stackrel{?}{=} \mathcal{H}(r^{(i)} \parallel K_B^{(i)} \parallel ID_i) \tag{31}$$

Suppose $d^{(i^*)}$ passes, then Alice can compute the authenticated group key K_G where $\mathcal{H}$ is a one-way hash function and $(i_1^*, ..., i_m^*) = \arg_{i^*} \text{Sort}\{ID_{i^*}\}$.

$$K_G \leftarrow \mathcal{H}(K_A \parallel K_B^{(i_1^*)} \parallel K_B^{(i_2^*)} \parallel \cdots \parallel K_B^{(i_m^*)}) \tag{32}$$

Step 6: Bob$_j$ Verifies Bob$_i$'s Commitment. Bob$_j$ will verify the following for all $i \neq j$.

$$d^{(i)} \stackrel{?}{=} \mathcal{H}(r^{(i)} \parallel K_B^{(i)} \parallel ID_i) \tag{33}$$

Suppose $d^{(i^*)}$ passes, then Bob$_j$ can compute the authenticated group key K_G where $\mathcal{H}$ is a one-way hash function and $(i_1^*, ..., i_m^*) = \arg_{i^*} \text{Sort}\{ID_{i^*}\}$.

$$K_G \leftarrow \mathcal{H}(K_A \parallel K_B^{(j_1^*)} \parallel K_B^{(j_2^*)} \parallel \cdots \parallel K_B^{(j_m^*)}) \tag{34}$$

3.3 Discrete Log Based Scheme

In this subsection, we will present a discrete logarithm based group key establishment protocol which allows multiple users $u_{B_i} \in S$ to efficiently generate an authenticated group key and an untraceable session identity. A diagram illustrating the information exchange between Alice and Bob is shown in Fig. 3.

Step 1: Alice's Key Establishment Request. Alice u_A first chooses a user set S, random string r, symmetric session key K_A, and session identity ID_A.

$$\left\{c_j \leftarrow \text{Encrypt}_{ek_j}(r \parallel K_A \parallel ID_A, \rho_j)\right\}_{u_j \in S} \tag{35}$$

Step 2: Bob$_i$'s Commitment. All users $u_j \in S$ will compute $r \parallel K_A \parallel ID_A$ using their secret decryption key dk_j.

$$r^{(j)} \parallel K_A^{(j)} \parallel ID_A^{(j)} \leftarrow \text{Decrypt}_{dk_j}(c_j) \tag{36}$$

Interested user Bob$_i$ ($u_{B_i} \in S$) (or set of users $\{\text{Bob}_{i_1}, ..., \text{Bob}_{i_m}\}$) will randomly choose a session identity ID_i, seed σ_i, and symmetric key $K_B^{(i)}$. Bob$_i$ will then broadcast the following triple $\langle F_K^{(i)}, F_\sigma^{(i)}, F_{\text{ID}}^{(i)} \rangle$.

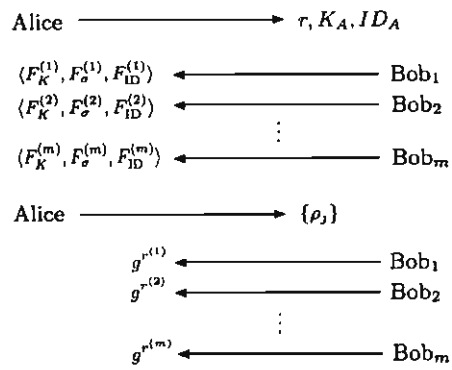

Figure 3 Discrete Log Based Key Exchange

$$F_K^{(i)} \leftarrow K_B^{(i)} \cdot g^{r^{(i)}} \tag{37}$$

$$F_\sigma^{(i)} \leftarrow \sigma_i \cdot g^{r^{(i)}} \tag{38}$$

$$F_{\text{ID}}^{(i)} \leftarrow ID_i \cdot g^{r^{(i)}} \tag{39}$$

Step 3: Alice Opens Commitment. Alice sends the commitment values $\vec{\rho} = (\rho_1, ..., \rho_n)$ using the secure group channel key K_A for all users $u \in S^+$.

$$C_A \leftarrow E_{K_A}(\vec{\rho}) \tag{40}$$

Step 4: Bob$_i$ Verifies Alice's Commitment. Bob$_i$ will decode Alice's commitment values $\{\rho_j\}$ and verify that the initial group key establishment request $\{c_j\}$ was generated properly. Specifically, Bob$_i$ will check the following.

$$\forall j, c_j \stackrel{?}{=} \text{Encrypt}_{ek_j}(r^{(i)} \parallel K_A^{(i)} \parallel ID_A^{(i)}, \rho_j) \tag{41}$$

If a single ciphertext c_{j^*} fails, STOP. Otherwise, Bob$_i$ will reveal his commitment value $\langle g^{\sigma_i} \rangle$.

Step 5: Alice Verifies Bob$_i$'s Commitment. Alice will verify that $g^{\sigma_i'} \stackrel{?}{=} g^{\sigma_i}$ where σ_i' can be computed as follows.

$$\sigma_i' = \frac{F_\sigma^{(i)}}{g^r} \tag{42}$$

Suppose $g^{\sigma_{i^*}}$ passes, Alice will immediately compute Bob$_{i^*}$'s key $K_B^{(i^*)}$ and identity ID_{i^*}.

$$K_B^{(i^*)} \leftarrow \frac{F_K^{(i^*)}}{g^r} \tag{43}$$

$$ID_{i^*} \leftarrow \frac{F_{\text{ID}}^{(i^*)}}{g^r} \tag{44}$$

Finally, Alice can compute the authenticated group key below where $(i_1^*, ..., i_m^*) = \arg_{i^*} \text{Sort}\{ID_{i^*}\}$.

$$K_G \leftarrow \mathcal{H}(K_A \parallel K_B^{(i_1^*)} \parallel K_B^{(i_2^*)} \parallel \cdots \parallel K_B^{(i_m^*)}) \tag{45}$$

Step 6: Bob$_j$ Verifies Bob$_i$'s Commitment. Bob$_j$ will verify that $g^{\sigma_i'} \stackrel{?}{=} g^{\sigma_i}$ where σ_i' can be computed as follows.

$$\sigma_i' = \frac{F_\sigma^{(i)}}{g^{r(j)}} \tag{46}$$

Suppose $g^{\sigma_{i^*}}$ passes, Bob_j will immediately compute Bob_{i^*}'s key $K_B^{(i^*)}$ and identity ID_{i^*}.

$$K_B^{(i^*)} \leftarrow \frac{F_K^{(i^*)}}{g^{r(j)}} \tag{47}$$

$$ID_{i^*} \leftarrow \frac{F_{\text{ID}(i^*)}}{g^{r(j)}} \tag{48}$$

Finally, Bob_j can compute the authenticated group key below where $(i_1^*, ..., i_m^*) = \arg_{i^*} \text{Sort}\{ID_{i^*}\}$.

$$K_G \leftarrow \mathcal{H}(K_A \parallel K_B^{(i_1^*)} \parallel K_B^{(i_2^*)} \parallel \cdots \parallel K_B^{(i_m^*)}) \tag{49}$$

4. Performance of Proposed Schemes

4.1 Computational Complexity

The computational complexity for the three proposed schemes can be computed and is compared in Table 1 where $|S| = n$ and $|\{\text{interested Bob}_j\}| = m$. The Naive Scheme has a rather large inefficiency due to number of public key encryption computations. The Hash and Discrete Log Schemes reduce this computation to $\Theta(n)$. Another interesting fact is that for both schemes, passive users who wish to simply compute the group key without actually authenticating the channel must perform the same exact number of computations as Bob_i assuming the cost of symmetric encryption and decryption are identical.

4.2 Security Analysis of the Group Key Establishment

In this subsection we will show at the three proposed schemes have the following properties: member exclusivity, group key randomization, zero-knowledge proof of membership, ID binding, and repudiation by each member. Short proofs of the above properties will be explained for all three of the proposed schemes.

Member Exclusivity. Suppose there is a user Eve ($u_E \notin S$) which would like to complete the group key establishment protocol (see Fig. 4). Since $u_E \notin S$, u_E cannot decrypt any of the ciphertexts $\{c_i\}$ and has no knowledge of random string r. Although Eve may still return a commitment to $r' \neq r$ in step 3, with overwhelming high probability Eve will not be able to prove her knowledge of r in step 5 and be exposed as a non-member. In addition, her key K_E will not be included in the group key computation so Alice will have no influence on the group key K_G.

Group Key Randomization. Alice or any other user cannot influence the generation of the final symmetric group key K_G. The reason is that since all interested users $u_i \in S$ must commitment to their random session key $K_B^{(i)}$ before

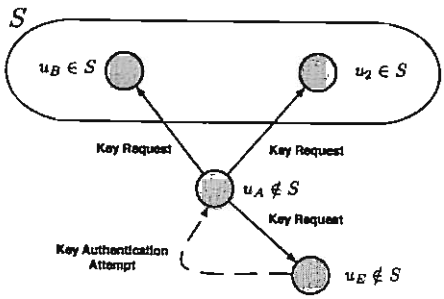

Figure 4 Attempted Attack by Eve

all other users reveal their random session keys. Thus, each user must choose a random session key independently of all other users. The first two schemes - naive scheme and the hash based schemes - heavily rely on the

Zero-Knowledge Proof of Membership. The random string r chosen by Alice is never revealed to any user $u \notin S$. All participating users $u_{B_i} \in S$, only reveal commitment to the knowledge of r but not r itself. This allows Alice and all other participating users $u_{B_i} \in S$ to verify that all other participating users u_{B_j} are also within user set S without revealing the random string to users outside of user set S.

ID Binding. The unique session identity ID_i chosen by user $u_i \in S$ is bounded to the key $K_B^{(i)}$ and zero-knowledge proof of r. The naive scheme binds the session identity using non-malleable encryption. In the hash based scheme, the session identity is bounded using a one-way hash function. In the discrete log scheme, the session identity is bounded using $g^{r^{(i)}}$.

Repudiation. For all users $u_{B_i} \in S$ who respond to Alice's group key establishment request will retain deniability. The reason is that the proof of membership is based on the same random string r which was verified that each user $u_j \in S$ received the same $r^{(j)}$, or $\forall j, r = r^{(j)}$.

4.3 Anonymous Linkable ID Generation

In each of the proposed schemes, temporary session identities were run-time computed and included during the group key establishment. In this subsection, we will further discuss how each user can use this identities to help distinguish users apart. This is particularly important for distinguishing users in a multi-channel environment. For instance, all users prefer to have a temporary identity, which cannot be traced to their original identity, that is linkable for the duration of the group channel key (see Fig. 5).

Assuming public system parameters (g, p), Alice initial creates a random identity $ID_A = g^a \mod p$ where a is Alice's secret key. Similarly, Bob_i create a random identity $ID_i = g^{\alpha_i} \mod p$ where α_i is Bob_i's secret key. Thus, since

Table 1 Computational Complexity of the Naive, Hash, and Discrete Log Schemes, respectively. The table lists the following computations: 1) SK: Symmetric key encryption / decryption, 2) PK-Enc: Non-malleable public key encyption, 3) PK-Dec: Non-malleable public key decryption, 4) Hash: One-way hash function, and 5) Exp: Exponentiation.

	Naive Scheme				Hash Scheme				Discrete Log Scheme				
	SK	PK-Enc	PK-Dec	Hash	SK	PK-Enc	PK-Dec	Hash	SK	PK-Enc	PK-Dec	Hash	Exp
Alice	$\Theta(1)$	$\Theta(mn)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	$\emptyset$	$\Theta(m)$	$\Theta(1)$	$\Theta(n)$	$\emptyset$	$\Theta(1)$	$\Theta(m)$
Bob_i	$\Theta(1)$	$\Theta(mn)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	$\Theta(1)$	$\Theta(m)$	$\emptyset$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(m)$
Passive User	$\Theta(1)$	$\Theta(mn)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	$\Theta(1)$	$\Theta(m)$	$\emptyset$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(m)$

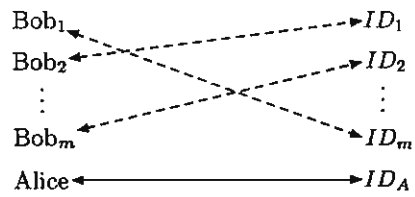

Figure 5 ID-Public Key Anonymous Linkability

each user $u \in S \bigcup \{u_A\}$ can use any signature scheme which is based on the discrete logarithm problem - ElGamal Signature Scheme [6] and Schnorr Signature Scheme [11]. In fact, any common public key can be used as the identity ID_i.

5. Conclusion

In this paper, we presented three authenticated group key establishment schemes with user repudiation for anonymous initiators. For each of these schemes, a user deniable symmetric group key K_G is created that allows any member in user set S to communicate securely to Alice (channel initiator). All users $u \in S$ can communicate anonymously to all users in set S but must actively authenticate the channel to ensure proper execution of the group key generation.

Unfortunately, there are a few inefficiencies in the three proposed scheme. First, the proposed scheme does not prevent any user from submitting multiple group key establishment commitments. It is possible to have more authenticated user identities ID_i than the size of the user set S^+. Thus, our proposed protocols do not force a one-to-one mapping from group members to user ID's. Second, the scheme does not prevent a non-member Eve $u_E \notin S$ from interfering with the key establishment. Eve may send many spurious commitments to increase the computational complexity of the group key establishment protocol by increasing parameter m (see Table 1). Finally, if the user identities ID_i are Diffie-Hellman based then there is a possibility that two users can create a private symmetric key between two users violating the open communication of the group key establishment.

References

[1] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable Encryption," *Springer-Verlag*, LNCS 1294 (Crypto '97), pp. 90–104, 1997.

[2] D. Catalano, R. Gennaro, N. Howgrave–Grahm, and P.Q. Nguyen, "Paillier's Cryptosystem Revisited," ACM CCS 2001, 2001.

[3] Ronald Cramer and Victor Shoup, "A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack," *Springer-Verlag*, LNCS 1462 (Crypto '98), pp.13–25, 1998.

[4] D. Dolev, D. Dwork, and M. Noar, "Non-Malleable Cryptography," 23rd Annual ACM Symposium on Theory of Computing, pp. 542–552, 1991.

[5] D. Dolev, D. Dwork, and M. Noar, "Non-Malleable Cryptography," Manuscript (full length version of STOC '91 paper), 2000.

[6] Taher Elgamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory, vol. IT-31, no. 4, pp. 469–472, July 1985.

[7] H. Krawczyk and T. Rabin, "Chameleon Signatures," Proc. Symposium on Network and Distributed Systems Security, pp.143–154, Feb 2000.

[8] Moni Naor, "Deniable Ring Authentication," *Springer-Verlag*, LNCS 2442 (Crypto '02), pp.481–498, 2002.

[9] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," 22nd Annual ACM Symposium on Theory of Computing, pp. 427–437, 1990.

[10] R. Rivest, A. Shamir, Y. Tauman, "How to Leak a Secret," *Springer-Verlag*, LNCS 2248 (ASIACRYPT '01), pp. 552–565, 2001.

[11] C. P. Schnorr, "Efficient Signature Generation by Smart Cards," Journal of Cryptography, vol. 4, no. 3, pp. 161–174, 1991.

[12] W. Susilo and Y. Mu, "Non-interactive Deniable Ring Authentication," *Springer-Verlag*, LNCS 2971 (ICISC '03), pp. 386–401, 2004.

[13] W. Susilo and Y. Mu, "Deniable Ring Authentication Revisited," *Springer-Verlag*, LNCS 3089 (ACNS '04), pp. 149–163, 2004.