The Computer Journal Advance Access published August 23, 2011

doi:10.1093/comjnl/bxr081

Fully Deniable Message Authentication Protocols Preserving Confidentiality

LEIN HARN[1], CHIA-YIN LEE[2], CHANGLU LIN[3] AND CHIN-CHEN CHANG[2,4,*]

[1]*Department of Computer Science Electrical Engineering, University of Missouri-Kansas City, Kansas City, MO 64110, USA*
[2]*Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 62102, Taiwan*
[3]*Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fujian 350007, P. R. China*
[4]*Department of Information Engineering and Computer Science, Feng Chia University, No. 100, Wenhwa Rd., Xitun Dist., Taichung 40724, Taiwan*
[*]*Corresponding author: ccc@cs.ccu.edu.tw*

Although the objective of secure communication can be achieved by using cryptographic tools, the undeniability that results from cryptographic properties may create a potential threat to the sender of the message. Unfortunately, most existing deniable protocols only provide 1-out-of-2 deniability. When both parties (the sender and the receiver) are allowed to deny generating the message, a dispute might occur between these two parties. The 1-out-of-2 deniable protocol can result in an unfair resolution of the dispute. Therefore, we propose a new model of deniability, called 1-out-of-∞ deniability, that can provide full deniability. The 1-out-of-∞ deniability protocol allows the originator of the message to deny that he or she generated the message, since there are an infinite number of possible message generators; at the same time, all transmitted messages can be protected and authenticated between the sender and the intended receiver. Our design can be implemented by using any public-key cryptography technique. We also analyze the correctness of the proposed protocols based on logical rules, and two practical examples are given to illustrate our design.

Keywords: deniability; authentication: confidentiality; privacy

Received 9 March 2011; revised 11 June 2011
Handling editor: Chin-Chen Chang

1. INTRODUCTION

Security is one of the most important services in various network communications. In most secure communications, the following two basic security properties are commonly considered.

1.1. Message confidentiality

For the sender, message confidentiality ensures that the messages can be read only by the intended receiver.

1.2. Message authentication

For the receiver, message authentication ensures that the message was sent by a specific sender and that the message was not altered en route.

To achieve these two security properties, a mutual authentication protocol must be used, and authenticated session keys should be shared between the communicating parties [1–5]. For message authentication and confidentiality, a keyed message authentication code (MAC) [6–8] can be used to verify the source of the transmitted message, and the session key can be used for encryption/decryption. Thus, before exchanging communication messages, a key establishment protocol is used to construct the session keys for the communication participants. The key establishment protocol must provide confidentiality and authentication for session keys.

As we know, the Diffie–Hellman key exchange protocol (DH-key exchange) [9] is one of the most commonly used key agreement protocols. In DH-key exchange, the session key is determined by exchanging one-time public keys of two communication parties. Since the public key itself does not provide any authentication, an additional digital signature is attached to the public key to provide authentication.

Although the objective of message authentication can be achieved by using the digital signature, there is still a potential threat to the sender of the message in that a digital signature

can be verified by any third party. Therefore, the receiver can pass the received message and the corresponding digital signature to any third party or post them on a public directory without the consent of the sender. Since the digital signature provides non-repudiation, the sender cannot deny sending the corresponding message. However, the property of undeniability is not desired in most personal communication applications, e.g. e-mail, instant messaging and electronic voting. Therefore, deniable authentication was proposed to overcome this privacy threat.

There are two types of deniability: *plausible deniability* and *full deniability* [10]. For plausible deniability, the message sender only can deny the transmission of a particular message. However, the sender is unable to deny the fact that he/she has communicated with the other participant. On the other hand, full deniability allows the message sender to totally deny that he/she has communicated with the other participant. In this article, we explain how our proposed protocols can achieve full deniability.

In 1998, Dwork *et al.* [11] first introduced the concept of deniable authentication protocols. Subsequently, many studies have been conducted and published in the literature [12–16] proposing various approaches for enhancing the security and efficiency of the protocols. On the other hand, designated verifier signatures [17, 18] also can provide deniable authentication services. In 2008, Harn and Ren [19] proposed a new deniable authentication protocol for electronic mail (e-mail) applications based on public-key cryptography. This protocol allows the message sender and message receiver the flexibility of using any public-key algorithms, such as the RSA cryptosystem [20] and the ElGamal cryptosystem [21]. In these solutions, deniability can be achieved because both the sender and the receiver have the ability to generate the transmitted message. Since the receiver knows that he or she did not generate the message, the receiver knows for certain that the sender generated the message. In addition, since any third party cannot know who generated the original message, the sender can deny generating the message. We call this type of deniability as 1-out-of-2 deniability because there is one user who originated the message out of the two users who could have generated the message. All deniable authentication protocols [11–19] proposed so far belong to this type of deniability.

The deniability is a unique property to protect the privacy of either the sender or the receiver in a secure communication. Most deniable communication protocols are designed to protect the privacy of the sender. If the transmitted messages between the sender and the receiver can be generated by both entities, the sender can deny the generation of the messages when the receiver passes the messages to a third party. In 2010, Yao and Zhao [22] proposed a new type of deniability, called *forward deniability*, to protect the privacy of the receiver. The forward deniability allows the receiver to deny the recorded messages that do not come from the sender, and the receiver has never talked to the sender, actually does not even know the sender. Yao and Zhao have constructed an Internet key exchange (IKE) protocol with forward and concurrent deniability. However, their deniable protocol belongs to the type of 1-out-of-2 deniability. In this article, the deniability is used to protect sender's privacy. We do not consider the forward deniability.

Unfortunately, the 1-out-of-2 deniable protocol has one potential problem. When both parties are allowed to deny generating the message, a dispute might occur between these two parties. This can be an issue because, when a dispute occurs between two parties, the general public often makes a subjective judgment against the party who has made prior mistakes, such as a criminal record or a bad credit history. Thus, the 1-out-of-2 deniable protocol can result in an unfair resolution of the dispute.

One way to improve the fairness of deniability is to increase the number of possible message generators from 2 to n, where n is a large positive integer. As we know, the ring signature [23, 24] can provide anonymity for the message signer. In a ring signature scheme, the message signer selects n ring members, including herself/himself, who could have possibly signed the message. The real signer can generate the ring signature by using her/his private key and the other $(n-1)$ ring members' public keys without their assistance or even awareness. However, the generated ring signature can convince any verifier that the ring signature indeed was signed by one of the ring members when the real signer's identity is fully anonymous to the verifier. Thus, the ring signature can provide 1-out-of-n deniability. One of the problems of the ring signature is that the computational complexity of generating and verifying a ring signature is proportional to the number of ring members. In addition, we need to point out that there is one major difference between the ring signature and the deniable authentication protocol. In the ring signature, the receiver cannot identify who the real message signer is, but, in the deniable authentication protocol, the receiver can authenticate the sender of the message.

To be truly fair, full deniability should be referred to as 1-out-of-∞ deniability, meaning that there are an infinite number of possible message generators. The computational complexity of a practical, fully deniable protocol should be as simple as a normal message authentication protocol. In this article, we propose two fully deniable communication protocols with message confidentiality and authentication. These proposed protocols are computationally efficient, can provide full deniability for the message originator and ensure the confidentiality and authentication of the message.

The main contributions of this article are summarized as follows:

(i) A new model of deniability, called 1-out-of-∞ deniability, that can provide full deniability is proposed. A fully deniable protocol allows the originator of the message to deny generating the message since the number of potential generators is infinite. In addition, messages can be protected and authenticated between the sender and the intended receiver.

(ii) Analyses of the deniability of the following security standards are provided: Pretty Good Privacy (PGP) [25], Secure/Multi-purpose Internet Mail Extensions (S/MIME) [26], Secure Socket Layer (SSL) protocol [27] and IKE [28, 29].
(iii) The proposed design can be implemented by using any public-key cryptography.
(iv) The proposed design is illustrated by two practical examples.
(v) Analyses of the correctness of the proposed protocols are conducted using logical rules.

The rest of this article is organized as follows. In Section 2, we present our analysis of the deniability of some well-known security standards. In Section 3, we describe our design concept and two design examples that provide confidentiality and authentication. In Section 4, we provide the analysis of the proposed protocols. Our conclusions are presented in Section 5.

2. ANALYSIS OF SOME WELL-KNOWN SECURITY STANDARDS

First, we define the three types of deniability as follows:

DEFINITION 1 (1-out-of-2 deniability). *The real message generator can deny generation of the message because there are two users who could have generated the message.*

DEFINITION 2 (1-out-of-n deniability). *The real message generator can deny generation of the message because there are n users who could have generated the message.*

DEFINITION 3 (1-out-of-∞ deniability). *The real message generator can deny generation of the message because any user could have generated the message.*

2.1. PGP and S/MIME

In recent years, e-mail has been one of the most important and widely used network applications. It has been used in communications between individuals, business organizations and governmental agencies around the world. E-mail is one of the most popular, non-interactive, communication applications in network environments. PGP and S/MIME are two well-known and useful secure e-mail solutions. Both solutions use a combination of conventional, symmetric-key (or secret-key) techniques and modern, asymmetric-key (or public-key) techniques to provide message confidentiality and message authentication.

In PGP and S/MIME applications, each user is assumed to have two pairs of public and private keys selected for long-term use. One pair of keys is used for message encryption, and the other pair is used for the digital signature. It is assumed that the public keys of all communication partners already have been stored securely in each user's public-key ring.

Both PGP and S/MIME use a digital envelope to provide message confidentiality. A digital envelope is a technique used by the sender to transmit the message in such a way that only the intended receiver can read the content of the message. First, the sender selects a session key randomly and uses this session key to encrypt the message. Then, the sender uses any public-key encryption algorithm to encrypt this session key by using the receiver's public key. After receiving the encrypted message, the receiver uses her/his private key to decrypt the message and obtain the session key. Then, the receiver uses the session key to decrypt the ciphertext. This approach for achieving message confidentiality provides 1-out-of-∞ deniability, since anyone can be the generator of the digital envelope.

Both PGP and S/MIME use a digital signature to provide message authentication. The message sender uses his or her private signing key to generate a digital signature on the message digest. The digital signature is attached along with the message, and both are sent to the receiver. The receiver can use the sender's public key to verify the digital signature. Since the digital signature is an evidence of non-repudiation, this approach for providing message authentication has no deniability at all.

2.2. SSL key exchange

SSL is an interactive protocol that provides confidentiality and data integrity for communications over TCP/IP networks. SSL has become a widespread security technology that is used in client–server applications, such as web browsing, Internet commerce and voice-over-IP (VoIP). The SSL protocol supports three kinds of DH-key exchange modes, i.e. two authenticated modes and one unauthenticated mode. DH-key exchange allows the client and the server to establish a common secret key by exchanging public information over an insecure channel. The general goal of the key exchange process in SSL is to establish a pre-master secret known only to the two participants. The pre-master secret will then be used to derive keys for message confidentiality and MAC keys for message authentication. In unauthenticated (anonymous) SSL mode, the pre-master secret is determined by the short-term DH public keys exchanged between the client and the server. Since the short-term DH public keys are unauthenticated, this protocol can provide 1-out-of-∞ deniability. However, anonymous DH-key exchange might suffer from the man-in-the-middle attack. In authenticated mode, the pre-master secret is determined either by fixed DH public keys with digital certificates or by short-term DH public keys signed by signatures (also called ephemeral DH), which are exchanged between the client and the server. In the fixed DH public keys with digital certificates, both participants know the common pre-master secret; so this protocol can provide 1-out-of-2 deniability. The main problem of this protocol is that the pre-master secret is never changed. This feature increases the risk of exposing the pre-master secret. In the short-term DH public keys signed with digital signatures,

the pre-master secret is different dynamically. However, since each participant signs the short-term DH public key, this protocol provides no deniability.

In SSL, there is another key exchange based on the RSA scheme. In RSA key exchange, a digital certificate for the server's public key must be made available. The client selects a pre-master secret randomly and then encrypts this pre-master secret with the server's RSA public key to create a digital envelope. Since the digital envelope can only be decrypted by the server's corresponding private key, this method protects the confidentiality of the pre-master secret. However, there is no authentication for the sender of the digital envelope. This key exchange method provides 1-out-of-∞ deniability.

2.3. Internet key exchange

IKE is the protocol used to set up a security association in the Internet Protocol Security (IPSec) [30] protocol suite. IKE uses the DH-key exchange to set up a shared secret, from which cryptographic keys are derived. Public-key techniques are used to mutually authenticate the communicating parties; alternatively, a pre-shared key can be used for this purpose. To allow for a variety of exchange methods, the IKE protocol includes defined modes for the phases. Here, we focus our analysis on key exchanges in the main mode.

In the pre-shared key method, the sender and the receiver have shared a secret key during the initialization. Then, these two parties exchange random nonces. Their common secret is calculated using a keyed-hash function of nonces and the pre-shared secret. Since the pre-shared secret is known by both entities, this key exchange method can achieve confidentiality and authentication. In addition, this method can provide 1-out-of-2 deniability.

In the revised public-key method, each party generates a one-time DH public key and encrypts this key under a one-time secret key to produce c_1. The one-time secret key is encrypted using the other party's public key to create a digital envelope as c_2. The pair (c_1, c_2) is sent to the other party. After receiving (c_1, c_2), the digital envelope c_2 can be opened with the corresponding private key, and, then, the one-time DH public key can be obtained. These two one-time DH public keys are combined to generate the common secret between the two parties. The digital envelope enables the sender and a specified receiver to share a secret. Since the sender can be any user, the digital envelope provides 1-out-of-∞ deniability. By using the digital envelope in both communication directions, the sender and the receiver can share a common secret. Thus, this method can achieve confidentiality and authentication with 1-out-of-∞ deniability.

In the digital signature method, the sender and the receiver must produce public-key certificates to verify the digital signatures. The digital signatures of all messages that are exchanged are used for authentication. Then, these two participants exchange nonces and one-time DH public keys. Finally, the common secret between these two participants can be calculated by a keyed-hash function of nonces and the one-time DH public keys. The use of the digital signature in this method allows confidentiality and authentication, but this method does not provide the property of deniability.

The analytical results of security protocols are summarized in Table 1.

3. PROPOSED DESIGN

3.1. Design concept

When the source of a message must be assured, the message sender can compute the MAC of the transmitted message

TABLE 1. Security properties of security protocols.

Protocols	Methods	Confidentiality	Authentication	Deniability
Ring signature [23, 24]	Digital signature	No	Yes[a]	1-out-of-n
PGP [25] or S/MIME [26]	Digital envelope	Yes	No	1-out-of-∞
	Digital signature	No	Yes	No
SSL [27]	Anonymous DH-key exchange	Yes	No	1-out-of-∞
	Fixed DH-key exchange	Yes	Yes	1-out-of-2
	Ephemeral DH-key exchange	Yes	Yes	No
	RSA cryptosystem	Yes	No	1-out-of-∞
IKE [28]	Pre-shared secret key	Yes	Yes	1-out-of-2
	Revised public key	Yes	Yes	1-out-of-∞
	Digital signature	Yes	Yes	No
Proposed protocols	DH-key exchange	Yes	Yes	1-out-of-∞
	RSA cryptosystem	Yes	Yes	1-out-of-∞

[a]The ring signature can be authenticated since the ring signature is generated by one of n possible ring members, but the receiver cannot identify who is the real signer of the message.

m under a one-time secret key k, such as $c = \mathrm{MAC}_k(m)$, and then send the pair $\{m, c\}$ to the receiver. Upon receiving $\{m, c\}$, the receiver can compute the MAC of message m as $c' = \mathrm{MAC}_k(m)$ and then determine whether $c = c'$. If the result is valid, the message m is authenticated; otherwise, the message is not authenticated. On the other hand, if the content of a communication requires the protection of its confidentiality, a common session key SK is required between the sender and the receiver. The confidentiality of the message can be ensured if the sender uses any symmetric encryption algorithm E to encrypt the message m, such as $c = E_{\mathrm{SK}}(m)$, where $E_{\mathrm{SK}}(m)$ refers to the encryption of message m using session key SK. Then, the ciphertext c is sent to the receiver, who can use session key SK to decrypt ciphertext c.

Commonly, two methods are used to distribute the common session key SK between the sender and the receiver, i.e. the symmetric-key solution and the asymmetric-key solution. When the symmetric-key solution is used, a secret key sk must be pre-shared between the sender and the receiver. Then, one participant selects the session key SK, encrypts it under the pre-shared secret key sk and then sends the ciphertext to the other participant. Since both communication parties know the pre-shared secret key sk, the symmetric-key solution can provide only 1-out-of-2 deniability.

In any public-key cryptosystem, the common session key SK can be constructed by the shared secret between the communicating parties. There are two usual approaches that can be used to share a secret between two parties, i.e. the DH-key exchange method and public-key based encryption. In the DH-key exchange method, the shared secret can be determined by exchanging short-term public keys between two parties. In public-key encryption, a participant is responsible for selecting the secret and then encrypting it under the other participant's long-term public key to create a digital envelope. Note that only the receiver can open the digital envelope with the corresponding private key.

In our design, full deniability is achieved by the fact that the transmitted ciphertext can be generated by any user. However, the security of the one-time key can only be shared between the sender and the receiver. In addition to the two contradictory objectives of deniability and security, we also need to consider the property of authentication.

3.1.1. Message Authentication with full deniability

For message authentication, the receiver of the message wants to make sure that only the specific sender can share the one-time secret key. Using the digital envelope technique, the receiver can select a one-time secret key and then encrypt the key in a digital envelope by using the sender's authenticated, long-term public key. Thus, only the sender can open the digital envelope by using the corresponding long-term private key. This solution can achieve message authentication, and, at the same time, it provides 1-out-of-∞ deniability since any user can generate the digital envelope. Using the DH-key exchange method, the receiver can compute a short-term DH public key and send this one-time public key to the sender. Then, the sender and receiver can share a one-time key based on the receiver's short-term DH key and the sender's long-term DH public key with digital certificate. This solution can achieve message authentication and, at the same time, it provides 1-out-of-∞ deniability since any user can generate the short-term public key.

3.1.2. Message confidentiality with full deniability

For message confidentiality, the sender of the message wants to make sure that only the intended receiver can share the message. The sender can encrypt the message by using a one-time secret key. Using the digital envelope technique, the sender can encrypt the one-time secret key in a digital envelope by using the receiver's authenticated long-term public key. Thus, the digital envelope can only be opened by the intended receiver by using the corresponding long-term private key. Actually, only the intended receiver can obtain the session key and use this secret key to decrypt the message. This solution can achieve confidentiality, and, at the same time, it provides full deniability since any user can generate the digital envelope. Using the DH-key exchange solution, the sender can compute a short-term DH public key and send this short-term public key to the intended receiver. Then, the sender and the receiver can share a one-time secret key based on the sender's short-term DH public key and the receiver's long-term DH public key with a digital certificate. This solution can also achieve confidentiality, and, at the same time, it provides full deniability since any user can generate the short-term DH public key.

3.2. Examples

3.2.1. Fully deniable protocol with authentication and confidentiality based on DH-key exchange

In SSL protocols, there are three algorithms that use the DH-key exchange with confidentiality, i.e. one with full deniability but no authentication, one with 1-out-of-2 deniability and authentication, and one with no deniability and authentication. Here, we propose a protocol with full deniability that also has message authentication and confidentiality.

We assume that Alice and Bob want to communicate with each other. Let the term $(x_{\mathrm{A}}, y_{\mathrm{A}})$ be Alice's pair of long-term private/public keys and the term $(x_{\mathrm{B}}, y_{\mathrm{B}})$ be Bob's pair of the long-term private/public keys, where $y_{\mathrm{A}} = g^{x_{\mathrm{A}}} (\mathrm{mod} p)$, $y_{\mathrm{B}} = g^{x_{\mathrm{B}}} (\mathrm{mod} p)$ and p is a public prime number. The terms $\mathrm{Cert}(y_{\mathrm{A}})$ and $\mathrm{Cert}(y_{\mathrm{B}})$ are the digital certificates of public keys y_{A} and y_{B}, respectively.

The communication between Alice and Bob, shown in Fig. 1, includes the following processes:

(i) Bob randomly selects a short-term private key k_1 and computes the corresponding short-term public key as

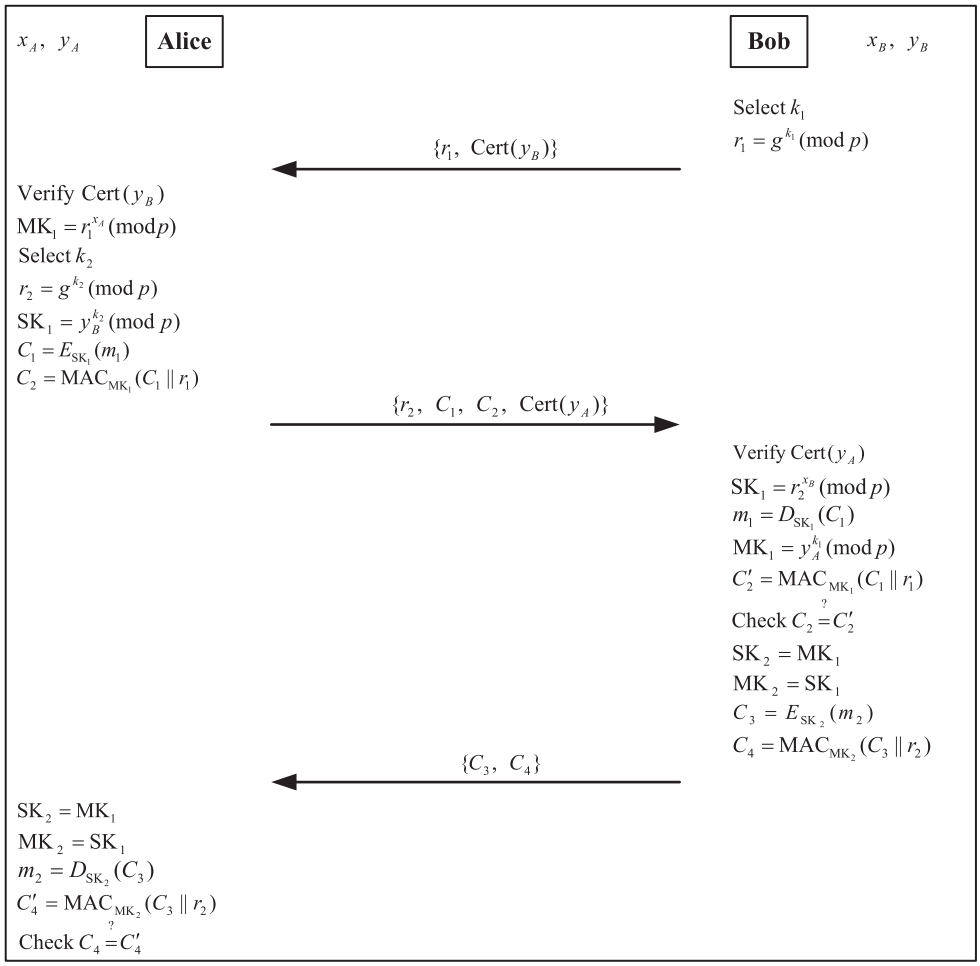

FIGURE 1. Fully deniable protocol with confidentiality and authentication based on DH-key exchange.

$r_1 = g^{k_1} (\text{mod} p)$. Then, he sends r_1 and the digital certificate $\text{Cert}(y_\text{B})$ of his long-term public key y_B to Alice.

(ii) Upon receiving $\{r_1, \text{Cert}(y_\text{B})\}$, Alice verifies the certificate $\text{Cert}(y_\text{B})$ and then obtains Bob's long-term public key y_B. Next, she selects a random k_2 as her short-term private key and then computes the short-term public key $r_2 = g^{k_2} (\text{mod}\, p)$.

(iii) Alice uses her short-term private key k_2 and Bob's long-term public key y_B to compute the session key SK_1 as $\text{SK}_1 = y_\text{B}^{k_2} (\text{mod} p)$, and then uses SK_1 to encrypt the message m_1 as $C_1 = E_{\text{SK}_1}(m_1)$.

(iv) Alice uses her long-term private key x_A to compute the one-time key MK_1 as $\text{MK}_1 = r_1^{x_\text{A}} (\text{mod} p)$ and then uses MK_1 to compute the MAC as $C_2 = \text{MAC}_{\text{MK}_1}(C_1 || r_1)$, where $||$ denotes the concatenation of messages. Afterward, Alice sends $\{r_2, C_1, C_2\}$ and the digital certificate $\text{Cert}(y_\text{A})$ of her long-term public key y_A to Bob.

(v) Upon receiving $\{r_2, C_1, C_2, \text{Cert}(y_\text{A})\}$, Bob verifies the certificate $\text{Cert}(y_\text{A})$ and then obtains Alice's long-term public key y_A.

(vi) According to DH-key exchange, Bob can use r_2 and his long-term private key x_B to compute the session key SK_1 as $\text{SK}_1 = r_2^{x_\text{B}} (\text{mod}\, p)$. Then, he can decrypt C_1 to obtain m_1 as $m_1 = D_{\text{SK}_1}(C_1)$.

(vii) Based on DH-key exchange, Bob can use his short-term private key k_1 and Alice's long-term public key y_A to compute the one-time key $\text{MK}_1 = y_\text{A}^{k_1} (\text{mod} p)$ and then uses MK_1 and r_1 to compute the MAC as $C_2' = \text{MAC}_{\text{MK}_1}(C_1 || r_1)$. If $C_2 = C_2'$, the message m_1 sent by Alice is authenticated; otherwise, Bob terminates the communication with Alice.

(viii) Bob sets the one-time keys SK_2 as $\text{SK}_2 = \text{MK}_1$ and MK_2 as $\text{MK}_2 = \text{SK}_1$. Next, he uses SK_2 to encrypt the message m_2 as $C_3 = E_{\text{SK}_2}(m_2)$ and then uses MK_2 and r_2 to compute the MAC as $C_4 = \text{MAC}_{\text{MK}_2}(C_3 || r_2)$. Afterward, Bob sends $\{C_3, C_4\}$ to Alice.

(ix) Upon receiving $\{C_3, C_4\}$, Alice sets the one-time keys SK_2 as $SK_2 = MK_1$ and MK_2 as $MK_2 = SK_1$. Then, she can decrypt C_3 to obtain the message m_2 as $m_2 = D_{SK_2}(C_3)$. Afterward, she uses MK_2 to compute the MAC as $C_4' = MAC_{MK_2}(C_3||r_2)$. If $C_4 = C_4'$, the message m_2 sent by Bob is authenticated; otherwise, Alice terminates the communication with Bob.

If the earlier-mentioned processes can be executed successfully, both message authentication and confidentiality between Alice and Bob can be ensured.

3.2.2. *Fully deniable protocol with confidentiality and authentication based on RSA cryptosystem*

In the SSL protocol, there is one algorithm based on RSA digital envelope, but this method does not provide authentication. The digital envelope is one useful method used to provide message confidentiality. Below, we show that digital envelopes can also be used to provide both message confidentiality and authentication with full deniability.

We assume that Alice and Bob want to communicate with each other. Let $y_A = (e_A, n_A)$ be the long-term RSA public key and d_A be Alice's long-term RSA private key, such that $e_A \cdot d_A \equiv 1(\text{mod}\varphi(n_A))$, where $\varphi(n_A) = (p_A - 1)(q_A - 1)$ and p_A and q_A are two large prime numbers selected by Alice. $\text{Cert}(y_A)$ denotes the digital certificate of Alice's long-term public key y_A. Similarly, let $y_B = (e_B, n_B)$ be the long-term RSA public key and d_B be Bob's long-term RSA private key, where $e_B \cdot d_B \equiv 1(\text{ mod } \varphi(n_B))$, and $\text{Cert}(y_B)$ denotes the digital certificate of Bob's long-term public key y_B.

The communication between Alice and Bob, shown in Fig. 2, includes the following processes:

(i) Alice sends Bob the digital certificate $\text{Cert}(y_A)$ of her long-term public key y_A.
(ii) After receiving $\text{Cert}(y_A)$, Bob verifies the certificate and obtains Alice's long-term public key $y_A = (e_A, n_A)$. Then, Bob randomly selects a one-time secret key k_1 and computes the digital envelope of k_1 as $r_1 = k_1^{e_A}(\text{mod}n_A)$. Afterward, Bob sends Alice the digital envelope r_1 and the digital certificate $\text{Cert}(yB)$ of his long-term public key y_B.
(iii) Upon receiving $\{r_1, \text{Cert}(y_B)\}$ from Bob, Alice verifies the certificate and obtains Bob's long-term public key $y_B = (e_B, n_B)$. Then, she randomly selects a one-time secret key k_2 and uses k_2 to encrypt the message m_1 as $C_1 = E_{k_2}(m_1)$.
(iv) Alice uses her long-term private key d_A to obtain k_1 as $k_1 = r_1^{d_A}(\text{mod}n_A)$. Then, she uses k_1 to compute the MAC as $C_2 = \text{MAC}_{k_1}(C_1||k_1)$.

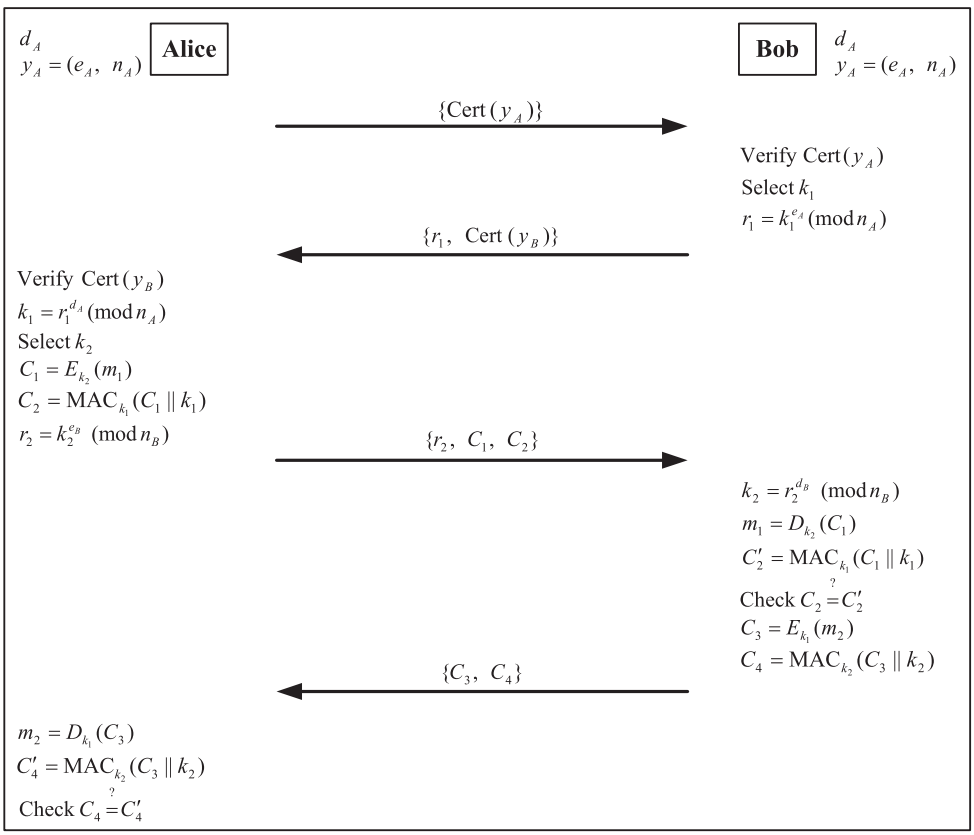

FIGURE 2. Fully deniable protocol with confidentiality and authentication based on RSA cryptosystem.

(v) Alice computes the digital envelope of k_2 as $r_2 = k_2^{e_B} (\mathrm{mod} n_B)$ and then sends r_2, C_1 and C_2 to Bob.

(vi) Upon receiving $\{r_2, C_1, C_2\}$ from Alice, Bob can obtain the secret key k_2 as $k_2 = r_2^{d_B} (\mathrm{mod}\, n_B)$. Then, he uses k_2 to decrypt the ciphertext C_1 to obtain the message m_1.

(vii) Bob uses the secret key k_1 to compute $C_2' = \mathrm{MAC}_{k_1}(C_1||k_1)$. If $C_2 = C_2'$, the message m_1 sent by Alice is authenticated; otherwise, Bob terminates the communication with Alice.

(viii) Bob uses the secret key k_1 to encrypt the message m_2 as $C_3 = E_{k_1}(m_2)$ and uses the secret key k_2 to compute the MAC as $C_4 = \mathrm{MAC}_{k_2}(C_3||k_2)$. Then, he sends the parameters C_3 and C_4 to Alice.

(ix) Upon receiving $\{C_3, C_4\}$ from Bob, Alice can use the secret key k_1 to decrypt the ciphertext C_3 and obtain the message m_2. Afterward, she uses the secret key k_2 to compute $C_4' = \mathrm{MAC}_{k_2}(C_3||k_2)$. If $C_4 = C_4'$, the message m_2 sent by Bob is authenticated; otherwise, Alice terminates the communication with Bob.

4. SECURITY ANALYSIS

In our design examples, both protocols employ the same authenticated encryption scheme to provide message authentication and message confidentiality. In an authenticated encryption scheme, a keyed MAC is used to verify the source of the transmitted message and a symmetric keyed encryption is used to protect the content of the transmitted message. The security analysis of three composition methods of MAC and encryption, namely *Encrypt-and-MAC, MAC-then-encrypt* and *Encrypt-then-MAC*, has been addressed by Bellare and Namprempre [31] in 2000. The method of *Encrypt-then-MAC* has been chosen in our design. Based on the results of [31], *Encrypt-then-MAC* is secure under the assumption that the given symmetric encryption scheme is secure against chosen-plaintext attack and the given MAC is unforgeable under chosen-message attack. For more information on the security analysis, interest readers can refer to the literature [30].

In 1990, Burrows *et al.* [32] proposed useful logical rules to prove the validity of authentication protocols. We use the model (BAN logic) proposed by Burrows *et al.* to analyze the correctness of our authentication protocols. In the appendix, we show that our protocols can achieve the features of message confidentiality and authentication, as well as provide full deniability for the sender of the message.

5. CONCLUSIONS

In this article, we propose a new concept of full deniability, called 1-out-of-∞ deniability. With 1-out-of-∞ deniability, when the sender sends an authenticated and encrypted message to the receiver, the sender can deny transmitting this message since anyone else could have generated the transmitted message. We analyze some well-known security protocols and discuss their deniability properties. In addition, we provide two design examples of full deniability and use the BAN logic model to analyze the correctness of the proposed protocols.

FUNDING

This research was supported by National Science Council of Taiwan (NSC 97-2221-E-035-039-MY3) and Natural Science Foundation of Fujian Province (No. 2011J05147).

REFERENCES

[1] Wu, T.Y. and Tseng, Y.M. (2010) An ID-based mutual authentication and key exchange protocol for low-power mobile devices. *Comput. J.*, **53**, 1062–1070.

[2] Lee, P.P.C., Lui, J.C.S. and Yau, D.K.Y. (2006) Distributed collaborative key agreement and authentication protocols for dynamic peer groups. *IEEE/ACM Trans. Netw.*, **14**, 263–276.

[3] Huang, C.M. and Li, J.W. (2007) Efficient and provably secure IP multimedia subsystem authentication for UMTS. *Comput. J.*, **50**, 739–757.

[4] Upmanyu, M., Namboodiri, A.M., Srinathan, K. and Jawahar, C.V. (2010) Blind authentication: a secure crypto-biometric verification protocol. *IEEE Trans. Inf. Forensics Sec.*, **5**, 255–268.

[5] Fan, C.I., Ho, P.H. and Hsu, R.H. (2010) Provably secure nested one-time secret mechanisms for fast mutual authentication and key exchange in mobile communications. *IEEE/ACM Trans. Netw.*, **18**, 996–1009.

[6] Bellare, M., Canetti, R. and Krawczyk, H. (1996) Keying Hash Functions for Message Authentication. *Proc. CRYPTO '96*, Santa Barbara, CA, USA, August 18–22, Lecture Notes in Computer Science, Vol. 1109, pp. 1–15. Springer, Berlin.

[7] Krawczyk, H., Bellare, M. and Canetti, R. (1997) *HMAC: Keyed-hashing for Message Authentication*. Internet Engineering Task Force (IETF), RFC 2104.

[8] Ge, R., Arce, G.R. and Di Crescenzo, G. (2006) Approximate message authentication codes for N-ary alphabets. *IEEE Trans. Inf. Forensics Sec.*, **1**, 56–67.

[9] Diffie W. and Hellman, M.E. (1976) New directions in cryptography. *IEEE Trans. Inf. Theory*, **IT-22**, 644–654.

[10] Kaufman, C., Perlman, R. and Speciner, M. (2002) *Network Security* (2nd edn). Prentice Hall PTR, Upper Saddle River, NJ.

[11] Dwork, C., Naor, M. and Sahai, A. (1998) Concurrent zero-knowledge. *Proc. 30th ACM Symp. on Theory of Computi*ng, Dallas, TX, USA, May 23–26, pp. 409–418. ACM.

[12] Deng, X. Lee, C.H. and Zhu, H. (2001) Deniable authentication protocols. *IEE Proc.-Comput. Digit. Tech.*, **148**, 101–104.

[13] Fan, L., Xu, C.X. and Li, J.H. (2002) Deniable authentication protocol based on Diffie–Hellman algorithm. *Electron. Lett.*, **38**, 705–706.

[14] Lu, R. and Cao, Z. (2005) Non-interactive deniable authentication protocol based on factoring. *Comput. Stand. Interfaces*, **27**, 401–405.

[15] Wang, B. and Song, Z. X. (2009) A non-interactive deniable authentication scheme based on designated verifier proofs. *Inf. Sci.*, **179**, 858–865.

[16] Youn, T.Y., Lee, C. and Park, Y.H. (2011) An efficient non-interactive deniable authentication scheme based on trapdoor commitment schemes. *Comput. Commun.*, **34**, 353–357.

[17] Saeednia, S., Kremer, S. and Markowitch, O. (2003) An Efficient Strong Designated Verifier Signature Scheme. *Proc. ICISC '03*, Seoul, Korea, November 27–28, Lecture Notes in Computer Science, Vol. 2971, pp. 40–54. Springer, Berlin.

[18] Kang, B., Boyd, C. and Dawson, E. (2009) A novel identity-based strong designated verifier signature scheme. *J. Syst. Softw.*, **82**, 270–273.

[19] Harn, L. and Ren, J. (2008) Design of fully deniable authentication service for E-mail applications. *IEEE Commun. Lett.*, **12**, 210–221.

[20] Rivest, R., Shamir, A. and Adleman, L. (1978) A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, **21**, 120–126.

[21] ElGamal, T.A. (1985) A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory*, **31**, 469–472.

[22] Yao, A.C.C. and Zhao, Y. (2010) Deniable Internet Key Exchange. *Proc. ACNS'10*, Beijing, China, June 22–25, Lecture Notes in Computer Science, Vol. 6123, pp. 329–348. Springer, Berlin.

[23] Rivest, R.L., Shamir, A. and Tauman, Y. (2001) How to Leak a Secret. *Proc. ASIACRYPT '01*, Gold Coast, Australia, December 9–13, Lecture Notes in Computer Science, Vol. 2248, pp. 552–565. Springer, Berlin.

[24] Ren, J. and Harn, L. (2008) Generalized ring signatures. *IEEE Trans. Dependable Secure Comput.*, **5**, 155–163.

[25] Garfinkel, S. (1994) *PGP: Pretty Good Privacy*. O'Reilly Media, Inc., Sebastopol, CA.

[26] Ramsdell, B. (2004) *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification.* Internet Engineering Task Force (IETF), RFC 3851. Sendmail, Inc.

[27] Freier, A.O., Karlton, P. and Kocher, P.C. (1996) *The SSL Protocol Version 3.0*. Netscape Communications Corp., VA. http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt.

[28] Harkins, D. and Carrel, D. (1998) *The Internet Key Exchange (IKE)*. Internet Engineering Task Force (IETF), RFC 2409. Microsoft, Inc.

[29] Kaufman C. (2005) *Internet Key Exchange (IKEv2) Protocol*. Internet Engineering Task Force (IETF), RFC 4306. Microsoft Inc.

[30] Kent, S. and Seo, K. (2005) *Security Architecture for the Internet Protocol*. Internet Engineering Task Force (IETF), RFC 4301. Microsoft Inc.

[31] Bellare, M. and Namprempre, C. (2000) Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. *Proc.ASIACRYPT '00*, Kyoto, Japan, December 3–7, Lecture Notes in Computer Science, Vol. 1976, pp. 531–545. Springer, Berlin.

[32] Burrows, M., Abadi, M. and Needham, R. (1990) A logic of authentication. *ACM Trans. Comput. Syst.*, **8**, 18–36.

APPENDIX

A.1. Authentication proof based on BAN logic

According to the analytical procedures of BAN logic, each round of the protocol must be transformed into an idealized form. First, we describe some notations of BAN logic in Table 2.

A.1.1. Authentication proof for the proposed protocol illustrated in Fig. 1

We use the rules of BAN logic to transform our first protocol, illustrated in Fig. 1, into the idealized form. The first message $M1_1$ of the protocol is omitted, since the message just includes a nonce and the certificate of the public key k_1. This message does not provide any of the logical properties of BAN logic. We describe the other messages in idealized form as follows:

$$M1_2.\ \text{Alice} \rightarrow \text{Bob} : \{m_1, r_1\}_{\text{SK}_1}, < \text{Alice} \overset{\text{SK}_1}{\leftrightarrow} \text{Bob}, r_1 >_{\text{MK}_1}, \{\text{Alice} \overset{\text{MK}_1}{\Leftrightarrow} \text{Bob}, r_1\}_{y_A^{-1}},$$

$$M1_3.\ \text{Bob} \rightarrow \text{Alice} : \{m_2, r_2\}_{\text{SK}_2}, < \text{Alice} \overset{\text{SK}_2}{\leftrightarrow} \text{Bob}, r_2 >_{\text{MK}_2}, \{\text{Bob} \overset{\text{MK}_2}{\Leftrightarrow} \text{Alice}, r_2\}_{y_B^{-1}}.$$

To analyze our first authentication protocol, we made some assumptions without loss of generality, as follows:

$A1_1$. Bob **believes fresh** r_1.

$A1_2$. Alice **believes fresh** r_2.

$A1_3$. Bob **believes** $\overset{y_A}{\mapsto}$ Alice.

$A1_4$. Bob **believes** (Alice **controls** Alice $\overset{\text{MK}_1}{\Leftrightarrow}$ Bob).

$A1_5$. Bob **believes** (Alice **controls** Alice $\overset{\text{SK}_1}{\leftrightarrow}$ Bob).

$A1_6$. Bob **believes** (Alice **controls** m_1).

$A1_7$. Alice **believes** $\overset{y_B}{\mapsto}$ Bob.

TABLE 2. Some notations of ban logic.

Notations	Descriptions
(X, Y)	The formula X or the formula Y is one part of the formula (X, Y)
$< X >_S$	The formula X combined with a secret S
$\{X\}_K$	The formula X encrypted by key K
$\overset{K}{\mapsto} U$	K is a public key of entity U. The corresponding private key K^{-1} will never be discovered by anyone except U
$U \overset{S}{\Leftrightarrow} V$	The secret formula S is known only to U and V. Only U and V can use S to prove their identities to each other
$U \overset{K}{\leftrightarrow} V$	P and Q may use the shared key K to communicate. Note that K will never be discovered by anyone except P and Q

$A1_8$. Alice **believes** (Bob **controls** Alice $\overset{MK_2}{\Leftrightarrow}$ Bob).

$A1_9$. Alice **believes** (Bob **controls** Alice $\overset{SK_2}{\leftrightarrow}$ Bob).

$A1_{10}$. Alice **believes** (Bob **controls** m_2).

The assumptions $A1_1$ and $A1_2$ are basic assumptions of BAN logic. We analyze the idealized form of the proposed protocol using the earlier-mentioned assumptions and the rules of BAN logic. We show the processes of the proof as follows:

PROPOSITION A.1. *Bob believes that the one-time secret MK_1 is shared with Alice.*

Proof. By $M1_2$, we apply the rule of BAN logic to break conjunctions and produce

Bob **sees** $\{m_1, r_1\}_{SK_1}$, (Statement 1)

Bob **sees** $<$ Alice $\overset{SK_1}{\leftrightarrow}$ Bob, $r_1 >_{MK_1}$, (Statement 2)

and

Bob **sees** $\{\text{Alice} \overset{MK_1}{\Leftrightarrow} \text{Bob}, r_1\}_{y_A^{-1}}$. (Statement 3)

By $A1_3$ and Statement 3, we apply the *message-meaning* rule to derive

Bob **believes** Alice **said** (Alice $\overset{MK_1}{\Leftrightarrow}$ Bob, r_1). (Statement 4)

By $A1_1$ and Statement 4, we apply the *nonce-verification* rule to derive

Bob **believes** Alice **believes** (Alice $\overset{MK_1}{\Leftrightarrow}$ Bob, r_1). (Statement 5)

By Statement 5, we break the conjunction to obtain

Bob **believes** Alice **believes** (Alice $\overset{MK_1}{\Leftrightarrow}$ Bob). (Statement 6)

By $A1_4$ and Statement 6, we apply the *jurisdiction* rule to obtain

Bob **believes** (Alice $\overset{MK_1}{\Leftrightarrow}$ Bob). (Statement 7) □

PROPOSITION A.2. *Bob believes that the real sender of message m_1 is Alice.*

Proof. By Statement 2 and Statement 7, we apply the *message-meaning* rule to derive

Bob **believes** Alice **said** (Alice $\overset{SK_1}{\leftrightarrow}$ Bob, r_1). (Statement 8)

By $A1_1$ and Statement 8, we apply the *nonce-verification* rule to derive

Bob **believes** Alice **believes** (Alice $\overset{SK_1}{\leftrightarrow}$ Bob, r_1). (Statement 9)

By Statement 9, we break the conjunction to obtain

Bob **believes** Alice **believes** (Alice $\overset{SK_1}{\leftrightarrow}$ Bob). (Statement 10)

By $A1_5$ and Statement 10, we apply the *jurisdiction* rule to obtain

Bob **believes** (Alice $\overset{SK_1}{\leftrightarrow}$ Bob). (Statement 11)

By Statement 1 and Statement 11, we apply the *message-meaning* rule to derive

Bob **believes** Alice **said** (m_1, r_1). (Statement 12)

By $A1_1$ and Statement 12, we apply the *nonce-verification* rule to derive

Bob **believes** Alice **believes** (m_1, r_1). (Statement 13)

By Statement 13, we break the conjunction to obtain

Bob **believes** Alice **believes** (m_1). (Statement 14)

By $A1_6$ and Statement 14, we apply the *jurisdiction* rule to obtain

Bob **believes** m_1. (Statement 15) □

PROPOSITION A.3. *Alice believes that the one-time secret MK_2 is shared with Bob.*

Proof. For $M1_3$, we apply the rule of BAN logic to break conjunctions and produce

Alice **sees** $\{m_2, r_2\}_{SK_2}$, (Statement 16)

Alice **sees** $<$ Alice $\overset{SK_2}{\leftrightarrow}$ Bob, $r_2 >_{MK_2}$, (Statement 17)

and

Alice **sees** $\{\text{Alice} \overset{MK_2}{\Leftrightarrow} \text{Bob}, r_2\}_{y_B^{-1}}$. (Statement 18)

By $A1_7$ and Statement 18, we apply the *message-meaning* rule to derive

Alice **believes** Bob **said** (Alice $\overset{MK_2}{\Leftrightarrow}$ Bob, r_2). (Statement 19)

By $A1_2$ and Statement 19, we apply the *nonce-verification* rule to derive

Alice **believes** Bob **believes** (Alice $\overset{MK_2}{\Leftrightarrow}$ Bob, r_2). (Statement 20)

By Statement 20, we break the conjunction to obtain

Alice **believes** Bob **believes** (Alice $\overset{MK_2}{\Leftrightarrow}$ Bob). (Statement 21)

By $A1_8$ and Statement 21, we apply the *jurisdiction* rule to obtain

Alice **believes** (Alice $\overset{MK_2}{\Leftrightarrow}$ Bob). (Statement 22) □

PROPOSITION A.4. *Alice believes that the real sender of message m_2 is Bob.*

Proof. By Statement 17 and Statement 22, we apply the *message-meaning* rule to derive

Alice **believes** Bob **said** $<$ Alice $\overset{SK_2}{\leftrightarrow}$ Bob, $r_2 >$. (Statement 23)

By $A1_2$ and Statement 23, we apply the *nonce-verification* rule to derive

Alice **believes** Bob **believes** $<$ Alice $\overset{SK_2}{\leftrightarrow}$ Bob, $r_2 >$. (Statement 24)

By Statement 24, we break the conjunction to obtain

Alice **believes** Bob **believes** (Alice $\overset{SK_2}{\leftrightarrow}$ Bob). (Statement 25)

By $A1_9$ and Statement 25, we apply the *jurisdiction* rule to obtain

Alice **believes** (Alice $\overset{SK_2}{\leftrightarrow}$ Bob). (Statement 26)

By Statement 16 and Statement 26, we apply the *message-meaning* rule to derive

Alice **believes** Bob **said** (m_2, r_2). (Statement 27)

By $A1_2$ and Statement 27, we apply the *nonce-verification* rule to derive

Alice **believes** Bob **believes** (m_2, r_2). (Statement 28)

By Statement 28, we break the conjunction to obtain

Alice **believes** Bob **believes** (m_2). (Statement 29)

By $A1_{10}$ and Statement 29, we apply the *jurisdiction* rule to obtain

Alice **believes** m_2. (Statement 30) □

From the above analysis, we prove the mutual authentication of the message between Alice and Bob.

A.1.2. Authentication proof for the proposed protocol illustrated in Fig. 2

Similarly, we transform our second protocol, illustrated in Fig. 2, into the idealized form according to the rules of BAN logic. The first message $M2_1$ and the second message $M2_2$ of this protocol are omitted, since these two messages do not provide any of the logical properties of BAN logic. We describe the other messages in idealized form as follows:

$M2_3$. Alice $\rightarrow$ Bob : $\{m_1, k_1\}_{k_2}$, $<$ Alice $\overset{k_2}{\leftrightarrow}$ Bob, $k_1 >_{k_1}$,

$M2_4$. Bob $\rightarrow$ Alice : $\{m_2, k_2\}_{k_1}$, $<$ Alice $\overset{k_1}{\leftrightarrow}$ Bob, $k_2 >_{k_2}$.

To analyze our second authentication protocol, we make some assumptions without loss of generality as follows:

$A2_1$. Bob **believes fresh** k_1.

$A2_2$. Alice **believes fresh** k_2.

$A2_3$. Bob **believes** (Alice $\overset{k_1}{\leftrightarrow}$ Bob).

$A2_4$. Alice **believes** (Alice $\overset{k_2}{\leftrightarrow}$ Bob).

$A2_5$. Bob **believes** (Alice **controls** m_1).

$A2_6$. Bob **believes** (Alice **controls** k_2).

$A2_7$. Alice **believes** (Bob **controls** m_2).

$A2_8$. Alice **believes** (Bob **controls** k_1).

PROPOSITION A.5. *Bob believes that the one-time key k_2 is shared with Alice.*

Proof. By $M2_3$, we apply the rule of BAN logic to break conjunctions and produce

Bob **sees** $\{m_1, k_1\}_{k_2}$ (Statement 31)

and

Bob **sees** $<$ Alice $\overset{k_2}{\leftrightarrow}$ Bob, $k_1 >_{k_1}$. (Statement 32)

By $A2_3$ and Statement 32, we apply the *message-meaning* rule to derive

Bob **believes** Alice **said** (Alice $\overset{k_2}{\leftrightarrow}$ Bob, k_1). (Statement 33)

By $A2_1$ and Statement 33, we apply the *nonce-verification* rule to derive

Bob **believes** Alice **believes** (Alice $\overset{k_2}{\leftrightarrow}$ Bob, k_1). (Statement 34)

By Statement 34, we break the conjunction to obtain

Bob **believes** Alice **believes** (Alice $\overset{k_2}{\leftrightarrow}$ Bob). (Statement 35)

By $A2_6$ and Statement 35, we apply the *jurisdiction* rule to obtain

Bob **believes** (Alice $\overset{k_2}{\leftrightarrow}$ Bob). (Statement 36) □

PROPOSITION A.6. *Bob believes that the real sender of message m_1 is Alice.*

Proof. By Statement 31 and Statement 36, we apply the *message-meaning* rule to derive

Bob **believes** Alice **said** (m_1, k_1). (Statement 37)

By $A2_1$ and Statement 37, we apply the *nonce-verification* rule to derive

Bob **believes** Alice **believes** (m_1, k_1). (Statement 38)

By Statement 38, we break the conjunction to obtain

Bob **believes** Alice **believes** (m_1). (Statement 39)

By $A2_5$ and Statement 39, we apply the *jurisdiction* rule to obtain

Bob **believes** m_1. (Statement 40) □

PROPOSITION A.7. *Alice believes that the one-time key k_1 is shared with Bob.*

Proof. By $M2_4$, we apply the rule of BAN logic to break conjunctions and produce

Alice **sees** $\{m_2, k_2\}_{k_1}$ (Statement 41)

and

Alice **sees** $<$ Alice $\overset{k_1}{\leftrightarrow}$ Bob, $k_2 >_{k_2}$. (Statement 42)

By $A2_4$ and Statement 42, we apply the *message-meaning* rule to derive

Alice **believes** Bob **said** (Alice $\overset{k_1}{\leftrightarrow}$ Bob, k_2). (Statement 43)

By $A2_2$ and Statement 43, we apply the *nonce-verification* rule to derive

Alice **believes** Bob **believes** (Alice $\overset{k_1}{\leftrightarrow}$ Bob, k_2). (Statement 44)

By Statement 44, we break the conjunction to obtain

Alice **believes** Bob **believes** (Alice $\overset{k_1}{\leftrightarrow}$ Bob). (Statement 45)

By $A2_8$ and Statement 45, we apply the *jurisdiction* rule to obtain

Alice **believes** (Alice $\overset{k_1}{\leftrightarrow}$ Bob). (Statement 46) □

PROPOSITION A.8. *Alice believes that the real sender of message m_2 is Bob.*

Proof. By Statement 41 and Statement 46, we apply the *message-meaning* rule to derive

Alice **believes** Bob **said** (m_2, k_2). (Statement 47)

By $A2_2$ and Statement 47, we apply the *nonce-verification* rule to derive

Alice **believes** Bob **believes** (m_2, k_2). (Statement 48)

By Statement 48, we break the conjunction to obtain

Alice **believes** Bob **believes** (m_2). (Statement 49)

By $A2_7$ and Statement 49, we apply the *jurisdiction* rule to obtain

Alice **believes** (m_2). (Statement 50) □

From the earlier-mentioned analysis, we prove the mutual authentication of the message between Alice and Bob.

A.2. Message confidentiality

In the first protocol, we know that the sender encrypts the message m_1 (or m_2) in such a way that only the intended receiver, who knows the long-term private key x_B (or x_A), can decrypt the

ciphertext C_1 (or C_3). Message confidentiality can be achieved in the earlier-mentioned communication.

In the second protocol, only Bob knows the correct long-term private key x_B to open the digital envelope r_1 created by Alice. Similarly, only Alice knows the correct long-term private key x_A to open the digital envelope r_2 created by Bob. Thus, message confidentiality can be achieved between Alice and Bob.

A.3. Deniability

PROPOSITION A.9. *The proposed protocol as illustrated in Fig. 1 achieves the property of deniability.*

Proof. We prove that all transcripts transmitted between Alice and Bob could be simulated by anyone else as follows.

Transcript Simulation: To simulate the transcripts between Alice and Bob, anyone else can choose two random number $\alpha, \beta \in Z_p^*$ and compute following terms:

$$\begin{cases} r_2 = g^{\beta} (\mathrm{mod}\, p) \\ \mathrm{SK}_1 = y_B^{\beta} (\mathrm{mod}\, p) \\ C_1 = E_{\mathrm{SK}_1}(m_1) \\ \mathrm{MK}_1 = y_A^{\alpha} (\mathrm{mod}\, p) \\ C_2 = \mathrm{MAC}_{\mathrm{MK}_1}(C_1 || r_1) \end{cases}$$

or

$$\begin{cases} r_1 = g^{\alpha} (\mathrm{mod}\, p) \\ \mathrm{SK}_2 = y_A^{\alpha} (\mathrm{mod}\, p) \\ C_3 = E_{\mathrm{SK}_2}(m_2) \\ \mathrm{MK}_2 = y_B^{\beta} (\mathrm{mod}\, p) \\ C_4 = \mathrm{MAC}_{\mathrm{MK}_2}(C_3 || r_2) \end{cases}.$$

Actually, the transcripts (r_2, C_1, C_2) (or (r_1, C_3, C_4)) in simulation are indistinguishable from those of Alice (or Bob). Therefore, Bob (or Alice) is not able to prove to a third party that the transcripts were produced by Alice (or Bob).

According to the above simulation, the proposed protocol can achieve full deniability. □

Similarly, we can prove that proposed second protocol as illustrated in Fig. 2 also achieve full deniability, since anyone can claim to be the creator of the digital envelope r_1 (or r_2) and knows the one-time secret key k_1 (or k_2).

REMARK. In order to provide 1-out-of-∞ deniability with confidentiality and authentication, the key exchange method of IKE, revised public-key method (Section 2), which employs techniques of both DH-key exchange and digital envelope. But this hybrid technique increases the computational overhead. In our design examples, we showed that using either the DH-key exchange method or the digital envelope can provide 1-out-of-∞ deniability with confidentiality and authentication. (See Table 1.)