ELSEVIER

Computer Communications 22 (1999) 1502–1511

computer communications

Anonymous channel and authentication in wireless communications

W.-S. Juang, C.-L. Lei*, C.-Y. Chang

Department of Electrical Engineering, Rm. 343, National Taiwan University, Taipei, Taiwan

Received 19 August 1998; received in revised form 1 June 1999; accepted 1 June 1999

Abstract

In this paper, we propose a scheme for providing anonymous channel service in wireless communications. By this service, many interesting applications, such as electronic elections, anonymous group discussions, with user identification confidential can be easily realized. No one can trace a sender's identification and no one but the authority centre can distinguish an anonymous message from a normal message when a user uses the anonymous channel. The user anonymity in our scheme is neither based on any trusted authority nor on the cooperation of all potential senders. Our scheme can be easily applied to existing wireless systems, such as GSM and CDPD, without changing their underlying structures. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Anonymous channel; Authentication; Untraceable e-mail systems; Electronic elections; Anonymous group discussions; Privacy and security

1. Introduction

Many applications, such as electronic voting schemes [1–3], anonymous group discussions, can be easily realized using anonymous channels [4,5]. In wireline networks, several anonymous channel protocols [4–6] have been proposed. The *mix-net* approach is used in Refs. [4,6] to realize a sender untraceable e-mail system. In the *mix-net* approach, the encrypted messages are sent to a mix agent who will disarrange all received messages, hold the encrypted messages for some random time and send them to the next agent. Finally, the last agent will send the encrypted messages to their destinations. The basic assumption of the *mix-net* approach is that at least one mix agent is honest. Pfitzmann [7] shows several attacks on the anonymous channels proposed in Ref. [6]. The *dc-net* method based on the Dining Cryptographers Problem is used in Ref. [5] to achieve a sender untraceable e-mail system which is unconditionally or cryptographically secure depending on whether it is based on one-time keys or on keys generated by public key distribution systems or pseudo-random number generators. In a *dc-net* scheme without a trusted authority, every pair of potential senders must share a secret key. To send an anonymous message, all potential senders must transmit the message bit by bit. In the *i*th bit transmission, each potential sender outputs the sum (modulo two) of all the *i*th bits of the keys he shares. If a sender wishes to transmit the bit '1', he inverts his output. Since every secret bit contributes exactly twice, if only one participant transmits the bit '1', the total sum (modulo two) of all participants' outputs must be one. If the message includes some redundancy information, anyone can detect collision of messages. When the sender detects a collision, he can retransmit his message after a period of time. In the *dc-net* method, it does not need any trusted mix-agent, but all potential senders must participate in the mail system when someone is delivering a message.

The most important and popular wireless systems are cellular systems, such as AMPS and GSM [8]. The first generation cellular systems, such as AMPS, are primarily aimed at voice communications. There is a growing need for wireless communication systems to provide data services, such as e-mail, fax, etc. for mobile units. The Cellular Digital Packet Data (CDPD) system [9] is designed to provide data services in an overlap to AMPS. It is designed to make use of cellular channels that are not being used for voice traffic. Another cellular system, the global systems for mobile communication systems (GSMs) used in European and some Asian countries, is designed to provide secure digital services such as user authentication, traffic confidential and key distribution.

In wireless communications, due to the lack of association between a user and a particular location, it makes it easier for an illegal user to attempt fraudulent acquisition of service. Thus, user authentication and the anonymous channel service must be addressed simultaneously. For

* Corresponding author. Tel.: + 886-2-2363-5251; fax: + 886-2-2363-8247.

E-mail address: lei@cc.ee.ntu.edu.tw (C.-L. Lei)

PII: S0140-3664(99)00108-5

accounting purpose, before the service provider provides an anonymous channel service to a user, user authentication must be considered in advance. In wireless channels, user anonymity and user authentication have rarely been addressed simultaneously. Many schemes [10–16] have considered user identification confidential against outsiders but not against the service provider.

In wireless communications, it is easier to realize anonymous channels due to roaming, dynamic channel assignment and broadcasting [8,9]. In this paper, we propose an efficient anonymous channel in wireless environments, such that, it can be easily applied to the existing wireless systems. In our scheme, no one can trace a sender's identification and no one but the *home service* domain can distinguish an anonymous message from a normal message when a user uses the anonymous channel. The user anonymity in our scheme is neither based on any trusted authority like in Ref. [4] nor on the cooperation of all potential senders [5]. In the downlink channels (base station to mobile station), our scheme uses a secret key cryptosystem to encrypt the transmission message but in the uplink channels (mobile station to base station), it uses a public key cryptosystem to preserve the message privacy. The reason for our scheme to use public key encipher functions instead of secret key encipher functions in the uplink channel is to preserve the privacy of the subscriber's identification.

The remainder of the paper is organized as follows: In Section 2, we describe a high-level system architecture for GSM and CDPD-like wireless communication systems. In Section 3, we briefly describe blind signatures and low-cost public key message encryption algorithms used in our protocol. Based on the high-level system architecture, we propose our anonymous channel and authentication scheme in Section 4. The security issues of this scheme are examined in Section 5. Then we discuss the implementation issues in Section 6. Finally, a concluding remark is given in Section 7.

2. System architecture

In this section, we describe a high-level system architecture for GSM and CDPD-like wireless communication systems. In this architecture, a mobile station (MS) will communicate with a base station (BS), which comprising the radio equipment and small switch functions. The BS links the mobile service switching centre (MSC) with the MSs. The MSC, which performs the switching functions for MSs and allocates radio resources, can connect to other MSCs or the existing wireline networks, such as PSTN, B-ISDN, etc. via wireline networks. The MSC also connects to a dedicated authentication centre (AUC) which performs the authentication for each call attempt made by an MS.

BSs, MSCs and the AUC collectively form a service domain (SD). For simplicity, we will treat MSCs and the AUC as an integrated logical entity. Any data manipulation in an SD must be done in the logical entity, and all BSs will not keep any important secret data. The functions of BSs are just data receiving and transmission. Any user who plans to acquire a wireless service has to register himself with an SD and becomes a subscriber to this SD. The SD that a user registered with is referred to as his *home* SD (HSD), and other SDs that the user visits are his *visiting* SDs (VSDs). The SDs do not have to trust each other, so they do not have to share any private information of their own subscribers. In each SD, the network topology of BSs and the MSC is a star network. Each BS only connects to its MSC. An MS can communicate with the current MSC via the nearest BS by radio. The MSCs can communicate with other MSCs, existing B-ISDN or Internet by the wireline networks.

3. Blind signature and low-cost public key message encryption

3.1. Secure blind signature schemes

The concept of blind signatures was proposed by Chaum [17]. It is an interactive protocol that involves two kinds of participants, a signer and a set of requesters. It allows a requester to obtain signatures on messages he provides to the signer without revealing these messages. A distinguishing property required by a typical blind signature scheme [1,17–19] is the so-called "unlinkability", which ensures that a requester can prevent the signer from deriving the exact correspondence between the actual signing process performed by the signer and the signature which is later made public. The blind signatures can realize the secure electronic payment systems [20,21] protecting customers' anonymity, and the secure voting schemes [1–3] preserving voters' privacy. In a distributed environment, every signed blind message can be thought as a fixed amount of electronic money in secure electronic payment systems, or as a ticket in applications such as secret voting schemes. The security of the blind signature schemes proposed in Refs. [1,17] is based on the hardness of factorization [22] and that of the schemes proposed in Refs. [18,19] is based on the hardness of the computing discrete logarithm [23].

Any secure and efficient blind signature scheme can be applied to our proposed scheme. For simplicity, we adopt the RSA blind signature scheme [17] as an example. The RSA blind signature scheme is illustrated as follows. Let m be a message to be signed, s be the signature of m and $x \equiv_n y$ denote $x = y \bmod n$:

1. The requester sends to the signer a message $m' \equiv_n m\beta^e$, (e, n) is the public key of the signer and β is a random number chosen by the requester such that $\gcd(\beta, n) = 1$.
2. Upon receiving the message m', the signer generates its signature $s' \equiv_n (m')^d$ with his secret key d. Then he sends the message s' back to the requester.
3. Upon receiving the message s', the requester can obtain

signature s for m by computing $s \equiv_n s'\beta^{-1} \equiv_n (m\beta^e)^d\beta^{-1} \equiv_n m^d$.

The signer cannot derive m from m' since m' is transformed by the unknown random number β. In contrast the requester, knowing the value β, can compute the signature s of the message m from the message s'. To verify the signature s, one simply computes $m \equiv_n s^e \equiv_n m^{ed}$ and checks if m has some redundancy information. If m has no proper redundancy, a secure public one-way hashing function f can be applied to m for preventing the multiplicative attack. To verify the signature $s \equiv_n f(m)^d$ on m without redundancy, one must send m along with s to the verifier. The verifier can check if $f(m) \equiv_n s^e$.

3.2. Low-cost public key encryption algorithm

For achieving the low-cost computations in mobile units, the subscriber encrypts his messages by modified RSA encryption schemes [24,25]. For simplicity, we use Rabin's scheme [24] in our protocol to encrypt subscriber's messages. Rabin's scheme is illustrated as follows. Every user i randomly chooses his secret key (p_i,q_i), where p_i and q_i are two large strong primes, such that $p_i \equiv_4 q_i \equiv_4 3$, and publishes his public key n_i, where $n_i = p_iq_i$. For sending a secret message m to user i, anyone can encrypt the message by computing $c \equiv_{n_i} m^2$ and sending the ciphertext c to user i. With the information of the secret key (p_i,q_i), user i can efficiently decrypt the ciphertext as $m \equiv_{n_i} \sqrt{c}$. Rabin proved that computing m given c and n_i is as difficult as factoring n_i. Although Rabin's encryption function is not one-to-one (it is four-to-one), if we add some redundancy information to the message, the user with the secret key can decrypt the ciphertext and choose the correct plaintext. Since Rabin's scheme only needs one modulo multiplication to encrypt a message, it is especially suitable for mobile units with low-computation capability.

4. The proposed scheme

In this section, an authentication scheme for an anonymous channel is presented. A typical session of the scheme involves a subscriber, his HSD and the VSD from where the subscriber requests the service. The communication between the subscriber and his VSD is via wireless communications. The VSD can communicate with the HSD via a high-speed wireline network. The scheme consists of three protocols: the ticket issuing protocol, the authentication protocol and the ticket renewal protocol. In our scheme, if a subscriber plans to send an anonymous message, he first requests a blind ticket from his HSD using the ticket issuing protocol. Then he can use the ticket in the authentication protocol. If the lifetime of the ticket expires, the subscriber can revive the lifetime of the ticket via the ticket renewal protocol. For accounting purpose, the HSD keeps a ticket database to check if the requested ticket is out of money or expires.

The underlying assumptions of these protocols are that:

(a) there exists a secure blind signature scheme [1,17–19];
(b) there exists a secure asymmetric cryptosystem [22,24,25];
(c) there exist a secure symmetric cryptosystem [26,27] and a secure one-way hash function [28,29]; and
(d) no one can derive the origin of any message in the underlying mobile communication systems [8,9].

In our protocol, for simplicity, we use the RSA blind signature scheme as an example to generate blind tickets in the ticket issuing protocol. Any secure and efficient blind signature scheme can apply to our scheme. For achieving the low-cost computations in mobile units, the subscriber encrypts his messages by modified RSA encryption schemes [24,25].

In symmetric cryptosystems, if the secret keys are not known, it requires a great deal of processing in order to derive the plaintext from a ciphertext. For example, The well-known differential attack on DES [30] requires 2^{47} operations for a "chosen plaintext" attack.

Due to the roaming, dynamic channel assignment and broadcasting features of mobile communications, if a subscriber broadcasts a message without describing his identification in the uplink channel and the entropy of the potential subscribers' identifications are greater than zero, no one can determine anything about the correspondence between the message and the subscriber. If the anonymous channel service is requested by very few subscribers in a short period of time, the identification of a subscriber can still be tracked. We assume that there are many subscribers randomly requesting anonymous channel services in some period of time.

Let S denote a subscriber, V denote the current VSD of S, H denote the HSD of S and $X \rightarrow Y : Z$ denote that a sender X sends a message Z to a receiver Y. Also, let K_{vh} be the secret key shared by H and V, HID be H's identification number, $\{m\}_{e_r}$ denote the ciphertext of m encrypted using Rabin's public key e_r, $(m)_k$ denote the ciphertext of m encrypted using the secret key k of some secure symmetric cryptosystem and "$\cdot$" denote the conventional string concatenation operator. Let f be a secret one-way function known only by H and h be a public one-way function. H has RSA keys n_h, e_h and d_h, where n_h and e_h are the public keys and d_h is the corresponding secret key, Rabin's public key, e_r, and the corresponding secret keys p_r and q_r, where p_r and q_r are two large strong primes, such that $p_r \equiv_4 q_r \equiv_4 3$, and $e_r = p_r * q_r$. Let ID_i be a unique identification of subscriber i. Upon registration, every subscriber i shares a secret key $f(Key_i)$, where Key_i is a unique public number for subscriber i, with his HSD and keeps $(ID_i, f(Key_i), e_r, n_h, e_h, h(\,))$ in his handset (mobile unit).

4.1. The ticket issuing protocol

Before S can send an anonymous message via the wireless channel, he must purchase a blind ticket from H. This ticket will be used as the authentication ticket and the hash value of the ticket will be used as the secret key shared with H when S uses the anonymous channel. The protocol is as follows:

Step 1 : $S \rightarrow V : HID, N_1, \{ID_i, \Psi, Cert_i, T_1\}_{e_r}$
Step 2 : $V \rightarrow H : \{ID_i, \Psi, Cert_i, T_1\}_{e_r}, N_2$
Step 3 : $H \rightarrow V : (\Gamma, N_2)K_{vh}$
Step 4 : $V \rightarrow S : N_1, T$

In step 1, S sends his HID, a nonce N_1, his ID_i, a blind message Ψ, his authentication information $Cert_i$ and a time-stamp T_1 to V, where

$$\Psi = \beta^{e_h}(Tkt) \bmod n_h \qquad (\gcd(\beta, n_h) = 1,\ 1 < \beta < n_h,$$

$$\text{and } Cert_i = (T_1, \gamma)_{f(Key_i)}), \tag{1}$$

where γ is a random number for preventing the guessing attack since the entropy of the time stamp T_1 may be small and β is the blind factor of the blind signature. The time-stamp T_1 will be used for accounting check such that any malicious person cannot replay the message $\{ID_i, \Psi, Cert_i, T_1\}_{e_r}$ to fool H. If a nonce N' is used instead of the time stamp T_1, H must keep all old nonces for preventing the replay attack. The blind ticket information $Tkt = RD{\cdot}\delta{\cdot}lifetime$ contains a redundancy string RD, the beginning of a ticket lifetime and a random number δ for increasing the entropy of the message Tkt. All messages, except HID and N_1, will be encrypted by the H's Rabin's public key for privacy. In step 2, V simply passes a nonce N_2 and the received encrypted message to H.

Upon receiving the message in step 2, H first decrypts the message and then checks if S's identification is valid by verifying if T_1 is in the content of the certificate $Cert_i$ and T_1 has not been presented before. If yes, H signs the blind ticket by computing

$$\Gamma = (\Psi)^{d_h} \bmod n_h \tag{2}$$

and deducts a fixed amount of money from S's account. Then he sends the signed ticket $(\Gamma, N_2)_{K_{vh}}$ back to V. For simplicity we will take the size of $|n_h|$ to be 512 bits in our discussion. For verifying the signature signed by H, we can define a valid signature space as

$$\Re = \{RD{\cdot}x{\cdot}y | RD = 0^{256}, x \in \{0,1\}^{224}, |y| = 32, y + T \geq Current \geq y\}, \tag{3}$$

where $Current$ is the current time when the verifier receives the ticket, T the length of the duration that the ticket is valid, x a random number to increase the entropy of $\Re$ for preventing the guessing attack and y the beginning of a ticket lifetime.

Upon receiving the message in step 3, V checks if the nonce N_2 is in the encrypted message. If yes, V then simply broadcasts the received blind ticket Γ and N_1 to S via the wireless channel. The nonce N_1 will be used as the indicator of the blind ticket Γ so that S can seize Γ from the downlink channel. Upon receiving the blind ticket, S can obtain the real ticket by computing

$$K_{sh} = \beta^{-1}\Gamma \bmod n_h = (Tkt)^{d_h} \bmod n_h \tag{4}$$

and verify the validity of the ticket by checking if $(K_{sh})^{e_h} \bmod n_h = Tkt$.

4.2. The authentication protocol

After receiving the ticket from V, S can use it as an authentication ticket when he requests an anonymous channel service. When the first anonymous call is made, H will assign a pseudo-account (PA), which contains a pseudo-account ID and the volume of this account, to this ticket. This ticket can be used until the volume of its associated PA becomes empty or the ticket expires. The following protocol is the ith anonymous call with respect to this ticket:

Step 1 : $S \rightarrow V : HID, N_3, \{K_{sh}, r_i\}_{e_r}$
Step 2 : $V \rightarrow H : \{K_{sh}, r_i\}_{e_r}, N_4$
Step 3 : $H \rightarrow V : (K_i, r_i, PA, lifetime, N_4)_{K_{vh}}$
Step 4 : $V \rightarrow S : N_3, (I_i, r_i)_{K_i}$

In step 1, S sends his HID, a nonce N_3 and the encrypted message $\{K_{sh}, r_i\}_{e_r}$ to V. The encrypted message includes the authentication ticket K_{sh} and the ith random challenge r_i. The challenge r_i is used for computing the ith session key K_i and checking freshness. In step 2, V sends the received message $\{K_{sh}, r_i\}_{e_r}$ and a nonce N_4 to H.

Upon receiving the message in step 2, H first decrypts the message, and then checks if $(K_{sh})^{e_h} \equiv_{n_h} ((Tkt)^{d_h})^{e_h} \equiv_{n_h} Tkt \in \Re$, where the valid ticket domain $\Re$ is defined in (3) and r_i has not been presented before. H rejects the ticket if it is not valid. If the ticket expires, H rejects it. If it is valid and has not been presented before, H assigns a new PA to this ticket. Otherwise, H retrieves the PA corresponding to this ticket from the ticket database. Each entry of the ticket database contains the volume of each pseudo-account, PA, the ticket, Tkt, and the expired Tkt' if it exists. If the volume of PA is not empty, then he computes the session key $K_i = h(K_{sh}{\cdot}r_i)$, and sends the message $(K_i, r_i, PA, lifetime, N_4)_{K_{vh}}$ back to V.

Upon receiving the message in step 3, V decrypts the message and checks if the nonce N_4 is in it for freshness checking. If yes, V generates a pseudo-identification number I_i for this call and encrypts I_i and the challenge r_i with the session key K_i. Then he sends the message $N_3(I_i, r_i)_{K_i}$ back to S. The nonce N_3 will be used as the indicator of this call response so that S can seize the message $(I_i, r_i)_{K_i}$ from the downlink channel.

After receiving the encrypted message, S then obtains I_i by the session key K_i, which can be computed by $K_i =$

$h(K_{sh} \cdot r_i)$, and verifies the freshness of the message from the challenge r_i. Then he can use the pseudo-identification number I_i and the session key K_i to send anonymous messages until the volume of *PA* is empty. After the subscriber completes the anonymous message transmission, *V* can send (PA, r_i, $Cost_i$) to *H* via a secure channel for deducting the cost $Cost_i$, where $Cost_i$ is the cost of this anonymous call.

4.3. The ticket renewal protocol

If the lifetime of the requested ticket expires, *S* can ask *H* to revalidate the ticket lifetime.[1] The protocol is similar to the ticket issuing protocol except the authentication message $ID_i, Cert_i$ is replaced by the expired ticket K_{sh}. Note that *S* does not have to send another stamp T_2 to *H*. The reason is that if any malicious person replays the encrypted message to fool *H*, he can neither derive the expired ticket nor any new ticket without knowing the expired ticket since the expired ticket is encrypted by H's public key e_r and the new ticket is encrypted by the key $h(K_{sh})$ which can be computed only by *S* or *H*, and *H* will not deduct any money from *S*. If any malicious person replay the message $HID, N_5, \{K_{sh}, Tkt'\}_{e_r}$ to fool *H*, *H* can find that K_{sh}, Tkt' is in the ticket database and just ignore it. The protocol is described as follows:

Step 1 : $S \rightarrow V : HID, N_5, \{K_{sh}, Tkt'\}_{e_r}$
Step 2 : $V \rightarrow H : \{K_{sh}, Tkt'\}_{e_r}, N_6$
Step 3 : $H \rightarrow V : ((K'_{sh})_{h(K_{sh})}, N_6)_{K_{vh}}$
Step 4 : $V \rightarrow S : N_5, (K'_{sh})_{h(K_{sh})}$

In step 1, *S* sends his *HID*, a nonce N_5 and the encrypted message $\{K_{sh}, Tkt'\}_{e_r}$ to *V*. The encrypted message includes the expired ticket K_{sh} and the new ticket contents $Tkt' = RD \cdot \delta' \cdot lifetime'$, where $lifetime'$ is the new ticket lifetime and δ' is another random number. In step 2, *V* simply passes the received encrypted message and a nonce N_6 to *H*.

Upon receiving the message in step 2, *H* first decrypts the message, and then computes $(K_{sh})^{e_h} \bmod n_h = Tkt$ and checks if the redundancy information *RD* is in the message *Tkt*. If yes and the expired ticket has been used, he signs the new ticket $K'_{sh} = (Tkt')^{d_h} \bmod n_h$, and carries over the expired ticket to the new ticket in the ticket database. If yes and the expired ticket has not been used, he also signs the new ticket K'_{sh} and adds a new entry containing the expired ticket and the new ticket to the ticket database. Then he sends the encrypted message $((K'_{sh})_{h(K_{sh})}, N_6)_{K_{vh}}$ back to *V*.

Upon receiving the message in step 3, *V* checks if N_6 is in the encrypted message. If yes, he broadcasts the message $N_5, (K'_{sh})_{h(K_{sh})}$ via the wireless channel. The nonce N_5 will be used as the indicator of this call response so that *S* can seize the encrypted ticket $(K'_{sh})_{h(K_{sh})}$ from the downlink channel. Upon receiving the encrypted message, *S* can obtain the new ticket K'_{sh} by decrypting the message $(K'_{sh})_{h(K_{sh})}$.

[1] For manipulating the database of valid tickets, the service provider can provide serveral values of the durations that the ticket is valid according to the response time the subscriber can tolerate.

5. Security considerations

5.1. Protocol verification

To model all aspects of a cryptographic protocol by a particular formal method is extremely difficult, and thus it is unlikely that any formal method will be able to detect or prevent all types of protocol flaws. The best we can hope for is that it will be able to guarantee that the protocol is correct under a certain well-defined set of assumptions. Among the many formal methods [31–34] for cryptographic protocol analysis, methods based on communicating state machine models and methods based on logic of knowledge and belief are usually adopted.

The approach which uses the logic of knowledge and belief to analyse protocols is to use modal logic similar to those that have been developed for the analysis of the evolution of knowledge and belief in distributed systems. The best-known and most influential logic was that developed by Burrows, Abadi and Needham, commonly known as BAN logic [32]. Since BAN logic does not attempt to model knowledge, it cannot be used to prove results of secrecy; it can only reason about authentication. A major drawback of BAN logic is the lack of ability to recognize if a bit string is a meaningful message [31,33]. In Ref. [33], it has been showed that the protocol proposed in Ref. [35] has a security flaw that the intruder can convince a participant that a nonkey is a key. This attack can be prevented by forcing the encrypted message to be formatted. Basically, there are two kinds of approaches to increase the effectiveness of BAN logic [31,33]. One of them is to increase the scope of BAN logic itself. The extended version [34] of BAN logic includes rules for reasoning about message recognizability that makes it possible to reason about a principal's ability to recognize if a bit string is a meaningful message. However, this extended version of BAN logic is quite complex (containing over 50 rules). Since the encrypted messages in our authentication protocol are formatted, our authentication protocol is free from the above attack. For simplicity, we only present proof of the functionality of our authentication protocol using BAN logic. In this subsection, we will verify if the functions of the authentication protocol proposed in Section 4 are correct by BAN logic.

In BAN logic, there are several sorts of objects: principals, encryption keys and statements. In our protocol, the symbol *S*, *V*, and *H* denote specific principals; K_{vh} and $h(K_{sh})$ denote the secret keys; N_1,N_2 and N_3 denote specific statements. The symbols *P*, *Q* and *R* range over principals; *N* ranges over statements; *K* ranges over encryption keys. The only prepositional connective is conjunction, denoted by a comma. The notation $P \overset{K}{\leftrightarrow} Q$ denotes principal *P* shares a

secret key K with principal Q, $S \overset{X}{\Leftrightarrow} H$ denotes the formula X is a secret known only to S and H. $\overset{e_r H}{\longrightarrow}$ denotes e_r is the public key of H and $\{N\}_K$ denotes the ciphertext of the message N encrypted under the key K.

In our protocol, the goal of the authentication is to establish a shared session key K_i between S and V. Thus the authentication is complete between S and V if there is a K_i such that

S believes $S \overset{K_i}{\leftrightarrow} V$ and V believes $S \overset{K_i}{\leftrightarrow} V$.

A strong authentication may add the following two statements:

S believes V believes $S \overset{K_i}{\leftrightarrow} V$ and V believes S believes $S \overset{K_i}{\leftrightarrow} V$.

Although V does not know the real identity of S, he can charge the communication cost to the pseudo-account PA of S. In our protocol, the message is presented in an informal notation designed to suggest what the concrete implementation would be like. In the BAN logic analysis, the informal notation must be transformed to an idealized form. The idealized messages correspond quite closely to the messages described in the proposed protocol. We transform our protocol into the idealized form as follows:

Message 1 : $S \rightarrow V : \{S \overset{K_{sh}}{\Leftrightarrow} H, r_i\} e_r$
Message 2 : $V \rightarrow H : \{S \overset{K_{sh}}{\Leftrightarrow} H, r_i\} e_r$
Message 3 : $H \rightarrow V : \{r_i, S \overset{K_i}{\leftrightarrow} V, N_4\} K_{vh}$
Message 4 : $V \rightarrow S : \{r_i, S \overset{K_i}{\leftrightarrow} V\} K_i$

For example, message 4 $(I_i, r_i)K_i$ of the authentication protocol can be transformed to idealized message $\{r_i, S \overset{K_i}{\leftrightarrow} V\}_{K_i}$, which tells S, that K_i is a key to communicate with V. It is obvious that the initial assumptions must invariably be made to guarantee the success of each protocol. Generally, the assumptions state what keys are initially shared between the principals, which principals have generated fresh nonces and which principals are trusted in certain ways. The detail description of BAN logic can be found in Ref. [32]. Upon receiving message 2, H decrypts the received message $\{S \overset{K_{sh}}{\Leftrightarrow} H, r_i\} e_r$. Since K_{sh} can only be computed by H or S and will be used as the secret key shared between S and H, we assume when H sees $S \overset{K_{sh}}{\Leftrightarrow} H$ and r_i then

H believes $S \overset{K_i = h(K_{sh} \cdot r_i)}{\leftrightarrow} H$.

To analyze our protocol, we first give the following assumptions:

(a) S believes H controls $S \overset{K_i}{\leftrightarrow} V$;
(b) V believes H controls $S \overset{K_i}{\leftrightarrow} V$;
(c) H believes $V \overset{K_i}{\leftrightarrow} H$;
(d) V believes $V \overset{K_{vh}}{\leftrightarrow} H$;
(e) S believes $S \overset{K_i}{\leftrightarrow} H$;
(f) V believes fresh (N_4);
(g) S believes fresh (r_i);
(h) S believes $\overset{e_r}{\rightarrow} H$; and
(i) H believes $S \overset{K_i}{\leftrightarrow} H$.

Now we analyze the function of the authentication protocol proposed in Section 4 as follows.

From Message 4, assumption (e) and the message-meaning rule, we derive

S believes H said $(r_i, S \overset{K_i}{\leftrightarrow} V)$.

From assumption (g), we can apply the nonce-verification rule to it and obtain

S believes H believes $(r_i, S \overset{K_i}{\leftrightarrow} V)$.

From assumption (a), we can apply the jurisdiction rule to it and derive

S believes $S \overset{K_i}{\leftrightarrow} V$. (5)

From Message 3, assumption (d) and the message-meaning rule, we can derive

V believes H said $(S \overset{K_i}{\leftrightarrow} V, N_4)$.

From assumption (f), we can apply the nonce-verification rule to it and obtain

V believes H believes $(N_4, S \overset{K_i}{\leftrightarrow} V)$.

From assumption (b), we can apply the jurisdiction rule to it and derive

V believes $S \overset{K_i}{\leftrightarrow} V$. (6)

From Message 4, (5) and the message-meaning rule, we can obtain

S believes V said $(r_i, S \overset{K_i}{\leftrightarrow} V)$.

From assumption (g), we can apply nonce-verification rule to it can obtain

S believes V believes $S \overset{K_i}{\leftrightarrow} V$. (7)

From statements (5)–(7), we can know that our protocol has the following three properties:

S believes $S \overset{K_i}{\leftrightarrow} V$,

V believes $S \overset{K_i}{\leftrightarrow} V$ and S believes V believes $S \overset{K}{\leftrightarrow} V$.

The property "V believes S believes $S \overset{K}{\leftrightarrow} V$" can be made when S transmits the message $(r_i + 1)$ encrypted by the session key K_i to V in the following anonymous session.

5.2. Untraceability and accountability

The most important feature of our proposed protocol is the untraceability property. Moreover, the subscriber and the HSD must authenticate each other. We now show that our proposed scheme satisfies the above properties.

Based on the technique of blind signatures, we first show

that no one can derive the subscriber's identification when he uses the anonymous channel.

There are two possible ways that the identification of a subscriber may be deduced by his HSD: (1) The HSD and the VSD cooperate to derive the link between the plaintext message $(ID_i, \Psi, Cert_i, T_1)$ which is sent to V in step 1 of the ticket issuing protocol and the plaintext message (K_{sh}, r_i), which is sent to V in step 1 of the authentication protocol; (2) Acquire the identification of S when he sends the message $\{K_{sh}, r_i\}_{e_r}$ to V in step 1 of the authentication protocol or sends any message to V in the subsequent communication.

To derive the link between the string $(ID_i, \Psi, Cert_i, T_1)$ and (K_{sh}, r_i), is computationally infeasible since it clearly contradicts assumption (a) mentioned in Section 4. To acquire the identification of S when he sends the message $\{K_{sh}, r_i\}_{e_r}$ to V or sends any message to V in subsequent communication is impossible since it contradicts assumption (d) mentioned in Section 4.

Thus, we claim that the provided channel is untraceable.

In our protocol, the accounting of using wireless channel is achieved by the *PA*s. When a subscriber requests a blind ticket from the HSD, the HSD will withdraw a fixed amount of money from the subscriber's account. Upon receiving a ticket from an anonymous subscriber, the HSD will assign a *PA* to this ticket. All costs of using the wireless channel will be deducted from its *PA* until the volume of *PA* is empty. If some subscriber plans to use the anonymous channel without paying the cost, he must forge a legal ticket $(Tkt'')^{d_h} \bmod n_h$ or impersonate a legal subscriber i in the ticket issuing protocol. However, this contradicts assumptions (a) or (c) mentioned in Section 4. From the above, we claim that the provided channel is accountable.

For manipulating the database of *PA*, a ticket can only be used during its lifetime. If the lifetime of the ticket expires, the subscriber can renew the ticket by the ticket renewal protocol.

5.3. The low exponent protocol failures

Due to the mobile communication characteristics, a wireless network requires a public cryptosystem that offers low computational cost. Therefore in our protocol the HSDs will publish the modified RSA's encryption key $e = 2$ with different modulo n [24,25]. Although there are two kinds of the well-known low exponent protocol failures in Refs. [36–38], we show why our protocol can withstand these kinds of protocol failures.

5.3.1. The messages are encrypted with the same modulo

In our protocol, every subscriber only has to encrypt his private message with his HSD's Rabin's public key e_r. A cryptoanalyst can only eavesdrop equations as follows:

$$c_1 = (ID_i \cdot \Psi \cdot Cert_i \cdot T_1)^2 \bmod e_r, \tag{8}$$

$$c_2 = (K_{sh} \cdot r_i)^2 \bmod e_r \tag{9}$$

and

$$c_3 = \{K_{sh} \cdot Tkt')^2 \bmod e_r \tag{10}$$

with the same modulo e_r.

In our scheme, 8 contain at least three unknown independent variables δ, r_i and δ'. A cryptoanalyst can only eavesdrop a new equation containing another unknown independent variable r_{i+1} or δ'' in the $(i+1)$th authentication protocol or another ticket renewal protocol. Thus, our scheme is free from the attack proposed in Ref. [38].

Since these variables δ, r_{i+1}, r_i, δ' and δ'' are independent random variables, it is intractable to find public equations between them. Any attack in Ref. [36] cannot be used in our scheme.

We claim that to solve Eqs. (8)–(10) we have to break Rabin's scheme. It had been shown that breaking Rabin's scheme [24] is equivalent to solving the factorization problem. Therefore, our protocol is against this kind of low exponent attack.

5.3.2. The messages are encrypted with the different modulos

The protocol failure mentioned in Refs. [37,38] is as follows. In a distributed environment, assume that user i has his own RSA public keys $e_i = 3$ and n_i. Suppose that a user plans to send a secure message M to users j, k and l. The ciphertexts are $C_j = M^3 \bmod n_j$, $C_k = M^3 \bmod n_k$ and $C_l = M^3 \bmod n_l$.

If n_j, n_k and n_l are relatively prime, we can compute $M^3 \bmod (n_j n_k n_l)$ from C_j, C_k and C_l by the Chinese remainder theorem. Since $M^3 < n_j n_k n_l$, M can be recovered. If n_j, n_k and n_l are not relatively prime, then these composite numbers can be factored.

In our protocol, although a user may has several handsets, each handset (subscriber) can only register with a unique SD. All sensitive messages transmitted in the wireless uplink channel must be encrypted with his HSD's Rabin's public key e_r. So any cryptoanalyst can only get equations with the same modulo e_r which makes our protocol free from this kind of low exponent attack.

6. Discussion

6.1. Transparency to the VSD in the authentication process

We can think that the anonymous channels [4,5] are another advanced services, such that any subscriber needs to pay additional cost for using them. If the mail system supports the anonymous channel and the usual channel simultaneously, a subscriber can choose either type of these two services without disclosing any information of what the service is. Considering the authentication protocol of Section 4, the authentication message $\{K_{sh}, r_j\}_{e_r}$ of jth

Table 1
The functionality and complexity of related wireless authentication protocols

	Untraceability			Session key protection	Non-trusted authority	Mod multiplies
	Outsider	VSD	HSD			
GSM [8]	Yes	No	No	No	No	0
Beller [11]	Yes	No	No	Yes	No	2
Samfat [16]	Yes	Yes	No	No	No	2
Lin [15]	Yes	Yes	No	Yes	No	1
Our scheme	Yes	Yes	Yes	Yes	Yes	1

anonymous call can be easily replaced as the other authentication message $\{ID_i, Cert_{i,j}, T_j\}_{e_r}$, where $Cert_{i,j} = (T_j, \gamma_j)_{f(Key_i)}$, T_j is a timestamp and γ_j is a random number. Thus, no one except the HSD knows if the request is an anonymous message or a regular message. This implies that our proposed protocol can be easily embedded in the existing wireless authentication protocol without effecting the underlying structure of the VSD. The only thing that has to be done is that the service provider (the HSD) needs to distinguish the proposed authentication protocol and other existing protocols.

6.2. Implementation considerations

Different from Kerberos [39] where the ticket lifetime is chosen by the key server, the ticket lifetime is chosen by the subscriber in our protocol. This approach allows the subscriber to hide the ticket lifetime information when he purchases a ticket. The service provider can provide several values of the duration that the ticket is valid according to the response time the subscriber can tolerate, e.g. if there are three kinds of durations: one, three and five years. If the subscriber chooses the duration time as five years, he can use this ticket longer but has a longer waiting time for the service provider to check the validation of this ticket since for checking the validity of the ticket it has to search a much larger ticket database.

We assume that a portable unit contains a low-power microcontroller in order to perform the various tasks associates data manipulation and user interface. A typical 8-bit microcontroller dissipates 75–150 MW when operating at 6 MHz. Beller et al. [11] implemented modular multiplication on a typical microcontroller, such that, the implementation completes a single 512-bit modulo multiplication in 180 ms. The network server (HSD) can be done using a special-purpose processor. Dussé and Kaliski [40] have published an algorithm and claimed performance results that correspond to performing a single 512-bit modular multiplications in around 145 μs on a general-purpose Digital Signal Processor (DSP). A single such processor could perform the decryption of the modified RSA cryptosystems [24,25] for 10–20 calls/s. Assume three calls per user per hour, thus the processor can support 12 000–24 000 customers.

When $e_h = 3$, our protocol requires a precomputation on the order of 200 modular multiplications (20 s on an 8-bit microcontroller) in the portable unit for the ticket issuing protocol because the subscriber must compute the inverse of β to extract the real ticket from the blind ticket. The ticket issuing protocol can be performed in advance, and the ticket can be preserved for future authentication. For the ticket renewal protocol, it only needs three modular multiplications: one for encrypting the old ticket message and two for verifying the new ticket when receiving the ticket from the HSD.

6.3. Comparison of protocols

We summarize the functionality and complexity of related wireless authentication protocols in Table 1. The most important feature of our protocol is its untraceability. Generally, the adversaries can be classified into “outsiders” and “insiders”. An “outsider” is one who can only ascertain what can be intercepted via radio waves, while an “insider” is one who can obtain information by theft, conspiracy or computer system intrusion. An “insider” can be either a VSD or the HSD. Generally, the HSD will keep some secret information of the subscriber, such as the subscriber’s secret key. The only secret information shared between the subscriber and the VSD is the session key for some session call. From Table 1, we know that no one can derive a sender’s identification in our protocol. Lin et al.’s scheme and Samfat et al.’s scheme can hide a sender’s identification from outsiders and the VSD but not the HSD. For GSM and Beller et al.’s scheme, the VSD and the HSD both know a sender’s identification when the sender sends messages. Beller et al. proposed a session key protection scheme that if some subscriber’s secret key has been compromised, the old session keys will not been compromised. In our authentication protocol, the ith session key K_i is the hash value of the blind ticket K_{sh} and the ith challenge value r_i. All the challenge values are encrypted by the HSD’s Rabin’s public key. If the HSD’s Rabin’s secret key has not been compromised, and even if the intruder knows the subscriber’s secret key, he still can not know the conversation of the ith session. In our authentication protocol, the number of multiplications is 1 for the subscriber by using the Rabin’s encryption function. The number of multiplications in GSM is zero

Table 2
The comparison of related anonymous channels

	Mix-net [4]	Dc-net [5]	Our scheme
Free from trusted authorities	No	Yes	Yes
Sender's independence	Yes	No	Yes
Acountability	Yes	No	Yes
Communication cost	$(t+1)m$ (bits)	nm (bits)	$5m$ (bits)
Computation cost (each sender)	$\lceil m/512 \rceil (t+1)r$	$m(n-1)$	$y + \lceil m/64 \rceil x$
No. of secret keys (each sender)	0	$n-1$	1
No. of public keys (each sender)	$t+1$	0	1
Network infrastructure	Wireline	Wireline	Wireless
Transmission mode	Multicast	Broadcast	Broadcast

since it only uses the one-way hashing functions to implement the authentication protocol.

Assume that n is the number of potential anonymous senders in the dc-net method, t the number of mix-agents in the mix-net approach and x the number of bit operations of a DES encryption, r the number of bit operations of an RSA encryption, with a 512-bit modulo, and y the computation cost of the subscriber in our authentication protocol, which includes one modulo multiplication and one DES decryption (for simplicity, we assume the bit length of I_i and r_i is less than 64). Table 2 illustrates the comparison of related anonymous channels. Mix-net needs at least a trustworthy mix-agent to preserve the sender's identification privacy. A dc-net needs all possible senders to cooperate when someone sends an anonymous message. Our protocol needs neither a trustworthy agent nor all potential senders cooperating during an anonymous message transmission. User authentication is not considered in the dc-net method since all potential senders need to cooperate when someone is delivering an anonymous message and all potential senders is the candidate of the current sender. If the anonymous message is m bits, the communication complexity of our protocol is $5m$ bits which include two messages between MS and BS, two messages between BS and MSC and one message between MSC and MSC. In the dc-net method, it need nm bits to send an m bits anonymous message which is very impractical in a large network. In Ref. [40], a fast RSA implementation on the DSP56000 achieves 11.6 Kbits/s for 512-bit exponentiation with the Chinese remainder theorem and the DES implementation with the optional DES chip runs at 3.8 Mbits/s in the CBC mode. In the above implementations, the ratio of r/x is about $(3.8 \times 10^6 \times 8)/(11.6 \times 10^3) = 2.62 \times 10^3$. Typically, under a modulus n, the computation time for a modular exponentiation operation is about $O(|n|)$ times that of a modular multiplication where $|n|$ denotes the bit length of n. By the repeated squaring and multiplication method, a modular exponentiation operation is about $3|n|/2$ times that of a modular multiplication. By this method, the ratio of y/r is about $((2/3n) + (1/2.62 \times 10^3)) = ((2/3 \times 512) + (1/2.62 \times 10^3)) \approx 1.34 \times 10^{-3}$. The computation cost for the subscriber in our scheme is only $y + \lceil m/64 \rceil x$ bit operations. But the computation cost of the mix-net method [4] for transmitting m bits anonymous message is $\lceil m/512 \rceil (t+1)r$ bit operations. In the dc-net method, since each potential sender must compute the sum (modulo two) of all the ith bits of the keys he shares, the computation cost for transmitting m bits anonymous message is $m(n-1)$. In the dc-net method, each sender must share a secret key with every other potential sender. But in our scheme, every subscriber only needs to share a secret key with his HSD. In the mix-net method [4], every sender has to keep all mix-agents' public keys and the receiver's public key. In our approach, the sender only has to keep H's public key. It is more suitable for the dc-net method to use the broadcast transmission mode to transmit messages since all interested users need to compute the sum of all potential senders' outputs. But in the mix-net methods [4], since the sender only needs to transmit the anonymous message to the first trusted agent, it does not need broadcast a channel to transmit anonymous messages.

7. Conclusion

In this paper, we propose an efficient scheme for providing an anonymous channel service in wireless communications. By this service, many interesting applications with user identification confidential can be easily realized. Our scheme can be easily applied to the existing mobile communication systems, such as GSM, CDPD, without affecting the underlying structure of VSDs. The user anonymity in our scheme is neither based on any trusted authority nor on the cooperation of all potential senders. Nobody can trace the identification of any subscriber when he uses the anonymous channel. Further, no one but the HSD authority can distinguish anonymous messages from normal messages.

References

[1] C. Fan, C. Lei, A multi-recastable ticket scheme for electronic elections, Advances in Cryptology: Proc. AisaCrypt'96, Lecture Notes in Computer Science, 1163, Springer, Berlin, 1996, pp. 116–124.
[2] W. Juang, C. Lei, A collision free secret ballot protocol for computerized general elections, Computers and Security 15 (4) (1996) 339–348.
[3] W. Juang, C. Lei, A secure and practical electronic voting scheme for

real world environments, IEICE Trans. Fundamentals E80-A (1) (1997) 64–71.
[4] D. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Commun. ACM 24 (2) (1981) 84–88.
[5] D. Chaum, The dining cryptographers problem: unconditional sender and recipient untraceability, J. Cryptology 1 (1988) 65–75.
[6] C. Park, K. Itoh, K. Kurosawa, Efficient anonymous channel and all/nothing election scheme, Advances in Cryptology: Proc. EuroCrypt'93, Lecture Notes in Computer Science, 765, Springer, Berlin, 1993, pp. 248–259.
[7] B. Pfitzmann, Breaking an efficient anonymous channel, Advances in Cryptology: Proc. EuroCrypt'94, Lecture Notes in Computer Science, 950, Springer, Berlin, 1995, pp. 332–340.
[8] ETSI: GSM recommendations: GSM 01.02-12.21, February, 1993, Release 92.
[9] P.J.B King, Cellular Digital Packet Data System Specification: Release 1.1, CDPD Forum Inc., January 1995.
[10] N. Asokan, Anonymity in a Mobile Computing Environment, in: Proc. Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994.
[11] M.J. Beller, L. Chang, Y. Yacobi, Privacy and authentication on a portable communication system, IEEE J. Selected Areas Commun. 11 (6) (1993) 821–829.
[12] M. Beller, Y. Yacobi, Fully-fledged two-way public key authentication and key agreement for low-cost terminals, Electronic Letters 29 (11) (1993) 999–1001.
[13] U. Carlsen, Optimal privacy and authentication on a portable communications system, Operating Systems Rev. 28 (3) (1994) 16–23.
[14] L. Jianwei, W. Yumin, A user authentication protocol for digital mobile communication network, in: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 1995, pp. 608–612.
[15] H. Lin, L. Harn, Authentication in wireless communications, IEEE Global Commun. (1993) 550–554.
[16] D. Samfat, R. Molva, N. Asokan, Untraceability in mobile networks, in: Proc. First ACM Int. Conf. on Mobile Computing and Networking, November 1995, pp. 26–36.
[17] D. Chaum, Blind signatures systems, Advances in Cryptology: Proc. Crypt'83, Plenum Press, New York, 1993, pp. 153.
[18] J.L. Camenisch, J.M. Pivereau, M.A. Stadler, Blind signatures based on the discrete logarithm problem, Advances in Cryptology: Proc. EuroCrypt'94, Lecture Notes in Computer Science, 950, Springer, Berlin, 1995, pp. 428–432.
[19] W. Juang, C. Lei, Blind threshold signatures based on discrete logarithm, Proc. Second Asian Computing Science Conf. on Programming, Concurrency and Parallelism, Networking and Security, Lecture Notes in Computer Science, 1179, Springer, Berlin, 1996, pp. 172–181.
[20] D. Chaum, Privacy protected payments: unconditional payer and/or payee untraceability, Smartcard 2000, North Holland, Amsterdam, 1988.
[21] N. Ferguson, Single term off-line coins, Advances in Cryptology: Proc. EuroCrypt'93, Lecture Notes in Computer Science, 765, Springer, Berlin, 1993, pp. 318–328.
[22] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Commun. ACM (21) (1978) 120–126.
[23] S. Pohlig, M.E. Hellman, An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, IEEE Trans. Inform. Theory IT-24 (1978) 106–110.
[24] M.O. Rabin, Digitalized signatures and public key functions as intractable as factorization, MIT Laboratory of Computer Sciences, TR 212, January 1979.
[25] H.C. Williams, A modification of RSA public-key encryption, IEEE Trans. Inform. Theory IT-26 (6) (1980) 726–729.
[26] S. Muyaguchi, The FEAL cipher family, Advances in Cryptology: Proc. Crypt'90, Lecture Notes in Computer Science, 537, Springer, Berlin, 1990, pp. 727–738.
[27] M.F. Smid, D.K. Branstd, The data encryption standard: past and future, Proc. IEEE 76 (5) (1988) 550–559.
[28] R.L. Rivest, The MD5 message-digest algorithm, RFC 1321, Internet Activities Board, Internet Privacy Task Force, 1992.
[29] NIST FIPS PUB 180, Secure hash standard, National Institute of Standards and Technology, US Department of Commerce, DRAFT, 1993.
[30] E. Biham, A. Shamir, Differential cryptanalysis of DES-like cryptosystems, Advances in Cryptology: Proc. Crypt'90, Lecture Notes in Computer Science, 537, Springer, Berlin, 1990, pp. 2–21.
[31] C. Boyd, W. Mao, On a limitation of BAN logic, Advances in Cryptology: Proc. EuroCrypt'93, Lecture Notes in Computer Science, 765, Springer, Berlin, 1994, pp. 240–247.
[32] M. Burrows, M. Abadi, R. Needham, A logic of authentication, ACM Trans. Computer Systems 8 (1) (1990) 18–36.
[33] C.A. Meadows, Formal verification of cryptographic protocols: a survey, Advances in Cryptology: Proc. AsiaCrypt'94, Lecture Notes in Computer Science, 917, Springer, Berlin, 1995, pp. 135–150.
[34] L. Gong, R. Needham, R. Yahalom, Reasoning about belief in cryptographic protocols, in: IEEE Computer Society Symposium in Security and Privacy, 1994, pp. 234–248.
[35] A. Aziz, W. Diffie, Privacy and authentication for wireless local area networks, IEEE Personal Commun. 1 (1) (1994) 25–31.
[36] D. Coppersmith, M. Franklin, J. Patarin, M. Reiter, Low-exponent RSA with related messages, Advances in Cryptology: Proc. EuroCrypt'96, Lecture Notes in Computer Science, 1070, Springer, Berlin, 1996, pp. 1–9.
[37] J. Hastad, On using RSA with low exponent in a public key network, Advances in Cryptology: Proc. Crypt'85, Lecture Notes in Computer Science, 218, Springer, Berlin, 1985, pp. 403–408.
[38] J.H. Moore, Protocol failures in cryptosystems, Proc. IEEE 76 (5) (1988) 594–601.
[39] W. Stallings, Network and Internetwork Security, Prentice-Hall, Englewood Cliffs, NJ, 1995, pp. 315–333.
[40] S.R. Dusse, B.S. Kaliski Jr, A cryptographic library for the Motorola DSP56000, Advances in Cryptology: Proc. EuroCrypt'90, Lecture Notes in Computer Science, 473, Springer, Berlin, 1991, pp. 230–244.