SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks. 2009; **2**:427–444
Published online 30 December 2008 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.78

A survey of anonymity in wireless communication systems

Hui Chen[1], Yang Xiao[2]*,†, Xiaoyan Hong[2], Fei Hu[3] and Jiang (Linda) Xie[4]

[1]*Department of Mathematics and Computer Science, Virginia State University, P.O. Box 9068, Petersburg, VA 23806, U.S.A.*
[2]*Department of Computer Science, The University of Alabama, 101 Houser Hall, P.O. Box 870290, Tuscaloosa, AL 35487-0290, U.S.A.*
[3]*Department of Electrical and Computer Engineering, The University of Alabama, 317 Houser Hall, P.O. Box 870286, Tuscaloosa, AL 35487-0290, U.S.A.*
[4]*Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223-0001, U.S.A.*

Summary

Anonymity is an important security aspect of wireless communications and has continuously attracted significant attention. Implementing anonymity of mobile users not only protects their privacy but also reduces the chances of attacks based on impersonation; therefore security can be improved. Untraceability is a related issue to anonymity. If a user is traceable, its hidden identity can be revealed through profiling the activities associated to a user. In this paper, we conduct a survey on anonymity issues of wireless communication systems. We first discuss general issues of anonymity in wireless communication systems. Then we survey some protocols in the literature, which are designed for wireless mobile systems as well as wireless *ad hoc* networks. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: anonymity; untraceability; temporary identity; *ad hoc* networks; authentication; routing; wireless networks

1. Introduction

Wireless communications systems have kept gaining momentum in innovation and deployments. As a result, we have seen proliferation of wireless applications. Due to the openness of wireless media, security has become an important issue in wireless systems. In this paper, we discuss user anonymity and untraceability in wireless communications systems.

It is quite often that a wireless user disconnects or connects to a system voluntarily. A user authentication process is often required when the user connects to a wireless network. A mobile user may roam

*Correspondence to: Yang Xiao, Department of Computer Science, The University of Alabama, 101 Houser Hall, P.O. Box 870290, Tuscaloosa, AL 35487-0290, U.S.A.
†E-mail: yangxiao@ieee.org

off its home network and avail other network's services. The visiting network often needs the user's credential to authorize its use of the services. In wireless *ad hoc* networks, nodes often act as routers, needing routing information. In the above scenarios, while availing network services, many applications and services require to maintain users' anonymity. To maintain a user's anonymity, two categories of information need to be protected [1]. They are (1) movements and locations of network users and (2) activities of the network users, i.e., messages sent from or to the user. The former is often referred to as location anonymity (or privacy protection) and the latter data origin/destination anonymity (privacy protection) [1].

Anonymity has two following meaningful impacts [1]. First, implementing effective anonymity of a network user reduces security breaches under various attacks. Many attacks are launched by means of impersonation. To keep a network user's identity anonymous prevents an unintended party from associating its identity to the messages sent to or from the network user, or participating in the user's network sessions which the unintended party is not supposed to be in. In other words, it prevents the unintended party from impersonating the network user. Second, implementing effective anonymity of a network user prevents unintended parties from invading the user's privacy.

Anonymity is an effective mechanism to protect a user's privacy and also complies with the principle of least information [1]. Many researches aim at providing anonymous communication channels and deterring attacks on the channels [2,33]. Practical anonymity services such as Tor [3] have been deployed, and have protected privacy and deter censorship for many users. Emerging of wireless networks has posed additional challenges to anonymity, such as stated in Reference [4].

In this paper, we survey a number of recent proposed authentication protocols preserving anonymity for wireless mobile networks and routing protocols preserving anonymity for wireless *ad hoc* networks.

The rest of the paper is organized as follows. Section 2 describes requirements and constraints. Section 3 provides an overview of wireless authentication protocols. We introduce some attacks and protocol analysis in Section 4. Section 5 surveys wireless authentication protocols preserving user anonymity and untraceability, and provides some discussions such as pointing out some weaknesses. Section 6 discusses those in wireless *ad hoc* networks. Finally, we conclude this paper in Section 7.

2. Requirements and Constraints of Anonymity and Untraceability

In this section, we discuss general requirements and constraints for applying anonymity and untraceability of wireless users.

2.1. Mobility of Wireless Users

There are two kinds of wireless users, stationary and mobile users. Stationary users avail network services in their home network and authenticate with their home location registrars. When mobile users move out of their home networks, they avail network services through visiting networks and authenticate with the visiting location registrars, which may relay the authentication request to their corresponding home location registrars. The defense for tempering anonymity and untraceability of a wireless user must be placed at the authentication protocol for both stationary and mobile users. In fact, a stationary user can be regarded as a special mobile user who has never stepped out of its home network. An authentication protocol for mobile users can be easily modified and applied to stationary users as in Reference [5]. Without lose of generality, we will only study authentication protocols for mobile users in this paper.

In a wireless mobile network, the participants of an authentication protocol are home network, the mobile user, and the visiting network. Note that we use home network and home location registrar interchangeably. To assume anonymity of the mobile user is to protect the mobile user's identity from other participants, such as the visiting network.

2.2. *Ad Hoc* Networks

In an *ad hoc* network, some nodes act as routers. Even though the identity of a mobile user can be hidden from eavesdroppers and all the routing nodes in authentication protocols, the location of a network node and the relationship of the node with other nodes could be revealed by examining the routing information. Then the identity of the node could be compromised. Therefore, the routing protocol used by the *ad hoc* network must provide certain protection such as limiting the topological knowledge of the

DOI: 10.1002/sec

network to routing nodes to ensure the anonymity of wireless nodes.

2.3. Anonymity and Untraceability

The anonymity and the untraceability of a user, though two concepts, are related. For example, we may assign an alias to a wireless user. The user's true identity is not directly revealed. Since the alias is not changing, one may find out the whole activity of a single user. This may help eavesdroppers to profile network users and furthermore may use it to find out who the user is. Therefore, it is important to make the user's identity untraceable.

Five levels of untraceability are defined in Reference [6] according to (1) whose identity should be made untraceable and (2) to whom those identity should be made untraceable. The candidate identities that could be made untraceable are those of mobile users, home network, and visiting network. Those identities could be made untraceable to eavesdroppers, visiting network, legitimate network entities (e.g., other authorized third party, router nodes in the *ad hoc* networks), visiting network, and home network of the mobile user. We summarize the five levels of untraceability [6] as follows:

- In untraceability level 1, only the identity of the mobile user is made hidden to only eavesdroppers. A common strategy to implement untraceability level is to assign different temporary alias to a mobile user at different time. A long-term alias is not recommended because the true identity may be revealed by analyzing the activities associated to a long-term alias. Furthermore, as long as a long-term alias for a user is maintained, its true identity is no longer important since the long-term alias is capable of tracing the user.
- In untraceability level 2, the identity of the mobile user is hidden from not only the eavesdroppers but also legitimate network entities and the visiting network.
- In untraceability level 3, besides the identify of the mobile user, the identity of the home network is hidden from eavesdroppers and the legitimate network entities except the visiting network. This will pose a challenge for finding out the user's identity by inference. The following example shows the importance to hide user's home network. Suppose that an eavesdropper somehow knows that user Alice is the only user who is currently in network B. Assume that Alice's home network is not hidden. If Alice avails network B's service by authenticating with her home network A, the eavesdropper can easily infer that it is Alice who is using network B's service even though her identity is hidden from her disclosed home network.
- In addition to all the conditions in untraceability level 3, the identity of a mobile user's home network is hidden from the visiting network in untraceability level 4. If there is no solvency issue between home network and visiting network, this level of untraceability can be applied.
- Untraceability level 5 provides the most protection, and in this level, the identity of the mobile user is even hidden to its home network. That is to say, 'prefect' privacy is provided and no one except the user itself knows the user's identity.

As suggested in Reference [6], which level of untraceability is required depends on many factors, such as solvency of mobile users, accounting, billing, and intrusion detection. We believe that for implementation of such untraceability levels, besides effective algorithms and protocols, security and privacy policies, agreements, and laws also play important roles.

2.4. Constraints

These constraints are general to protocol design for preserving user's anonymity and untraceability in wireless networks [5,7].

- Mobile terminals usually have low computational power. Protocols requiring intensive computation on mobile terminals are not ideal. Public-key cryptography usually needs more computation than secret (symmetric)-key cryptography. Therefore, extra care should be taken when applying public-key cryptography.
- Wireless channel usually have lower bandwidth and high error rate. Protocols should be designed toward reducing message rounds and message sizes.

2.5. Common Solutions for Providing Anonymity

Attackers can violate location privacy [8,9] including using (1) domains visits, (2) physical geographical location visits, (3) motion traces, etc. Correlated

DOI: 10.1002/sec

information can also be used such as time-based analysis—from same stream/user and spatial—across different locations to violate location and data origin/destination privacy. Furthermore, attackers can use some other indirect metrics to track an identity, e.g., keystroke dynamics [10], TCP timestamps [11], clock skews [12], PHY signal patterns [4], etc. Finger printing techniques [13] using the above metrics can be used by attackers. There are two kinds of identifiers: implicit identifiers and explicit identifiers. Examples of implicit identifiers for WiFi include (1) vendor specific channel scan behaviors, (2) default requests to home network through SSID (a service set identifier), (3) broadcast packet sizes, (4) and other non-standard 802.11 header fields. Communication relationship can also be derived for (1) sender /receiver and (2) different sessions.

Common solutions for providing anonymity include (1) using encryption, e.g., Wi-Fi protected access (WPA) in WiFi, (2) frequently changing pseudonyms, (3) providing silent periods, (4) using broadcast packets, (5) dispersing messages to different routes, (6) dispersing locations of users, etc. There are some anonymous routing research problems: (1) overlay of MIXes, i.e., onion routing [2,14], (2) phantom routing [15], (3) traffic obfuscations, etc.

Finally, anonymity should come from all layers: the PHY layer, the MAC layer, the routing layer, and the upper layers (such as the TCP/UDP layer, the IP layer, the application layer), as well as protocols and algorithms in these layers.

3. Overview of Wireless Authentication Protocols

Authentication is a procedure by which an entity establishes a claimed property to another entity. Authentication protocols are notoriously error-prone. Evidently, the design of authentication protocols often demonstrates attack-fix-attack cycles [16]. A survey of authentication protocols for wired networks can be found in Reference [17], which is pertinent to but more general than wireless authentication protocols surveyed herewith.

A wireless authentication protocols is a well-defined procedure for users to provide their claimed identities and prove they own the identities to the service providers, such as their home networks, visiting networks, or both. During the process, an encryption/description key agreement will be usually reached.

Wireless authentication protocol usually involves at least three parties, namely, a mobile user, a visiting network, and a home network. Figure 1 shows a general and abstract view of a wireless authentication protocol. The wireless authentication protocol is usually more complex than wired counterparts. This is because a mobile user may roam off its home network, and it must communicate to its home network *via* the visiting network. Therefore, an ideal wireless authentication protocol could have up to three mutual authentication processes, (1) the mobile user and the visiting network authenticates to each other; (2) the visiting network and the home network authenticates to each other; (3) the mobile user and its home network authenticates to each other. Since the visiting network may not know the mobile user, the mutual authentication between the mobile user and the visiting network needs the assistance of the home network. Since the mobile user cannot directly communicate with the home network, the mutual authentication between the mobile user and the home network needs the visiting network to forward messages between them.

To assume anonymity and untraceability of a mobile user, the identity has to be hidden to some network entities, such as the visiting network. Obviously, authentication process and user's privacy protection are two conflicting requirements. Furthermore, a mobile user usually lacks of computational power and network bandwidth. Thus, wireless authentication protocols cannot be too complex; otherwise, the demand for computational power and bandwidth will make the protocols infeasible.

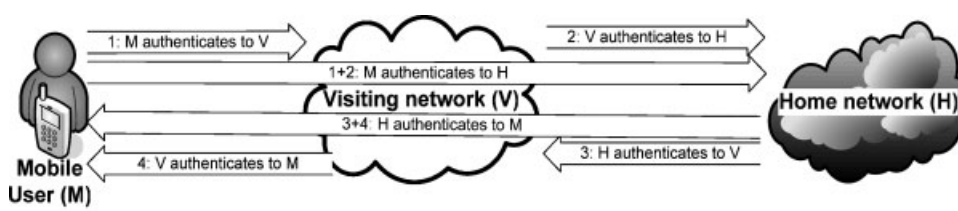

Fig. 1. Wireless authentication protocols.

DOI: 10.1002/sec

These constraints make the design of the protocol preserving anonymity and untraceability a greater challenge.

In general, an authentication protocols can be divided into protocols based on secret-key cryptography and those based on public-key cryptography. Due to multiple authentications illustrated in Figure 1, a wireless authentication protocol is often a hybrid between secret-key authentication protocols and public-key authentication protocols.

3.1. Secret-key Cryptography Based Authentication Protocols

It requires that the mobile user and home network shares a long-term secret key, which should be delivered through pre-established secure channel. Let us look at an example that Alice wants to authenticate herself to Bob. In this case, Alice must perform a cryptographic process using the shared key between them on a piece of data. Bob verifies if the correct cryptographic process has been performed. Thus, Bob indirectly tells if Alice possesses the shared key. In this process, Alice must announce her identity. However, an authentication protocol preserving anonymity must not announce an authenticatee's identity. It is suggested [1] that a pseudo identity (PID) can be created at a user's home domain, and should be changed frequently. The key issue is to indirectly verify the authenticatee's knowledge on the shared key without knowing its identity directly. Though this is not an easy task, the secret-key cryptography based authentication protocols are preferred by wireless networks because they are usually less demanding on processor cycles.

3.2. Public-key Cryptography Based Authentication Protocol

Some wireless authentication protocols are based on public-key cryptography. Generally, there are three different authentication frameworks based on public-key cryptography, i.e., directory-based public-key authentication protocols, identity-based authentication protocols [31], and self-certified authentication protocols [16,18].

As described in Reference [16], the difference of these three frameworks can be summarized as follows. Denote (S, P) be a pair of secret (S) and public (P) keys. A public-key authentication framework has a certification (or trusted) authority (CA), which generates the key pair with a guarantee (G) that links P to a user's identity (I). A user sends its digital signature to the authenticator, i.e., the party that will authenticate it. In a directory-based public-key authentication protocol, the signature is on pair (I, P), which along with G are made public. The authenticator first verifies G using CA's public key, and then authenticates the user by using P. In a directory-based public-key authentication protocol, a tree-like hierarchical public-key certification infrastructure needs to be maintained, which incurs a non-trivial level of complexity and cost [16]. In an identity-based authentication frame, public key P is I itself. The user signs the public key, its identity, using a private key (S). The signature is public. The authenticator has to verify the genuineness of the signature using public key $P = I$. In another word, the guarantee (G) is actually the secret key (S) itself. In a self-certified authentication protocol, the guarantee (G) is the public key (P). The user chooses the matching secret key (S), i.e., a user is defined by triple (S, P, I), where P uniquely identifies a user as well. Therefore, the protocol is said to be self-certified. A few self-certified protocols have been proposed [5].

Public-key cryptography based authentication protocols are natural to hide users' identities because an authenticatee can always use the public key to encrypt its identity [1]. However, public-key cryptography based authentication protocols are generally computational more expensive. Mobile devices are usually resource poor. Thus, public-key cryptography is often used in the authentication procedures between home networks and visiting networks. Between mobile users and its home network or visiting networks, secret-key cryptography is often used.

3.3. Anonymity and Untraceability

The mobile user's anonymity is usually achieved by using a temporary identity (TID) instead of using the mobile user's real identity in the authentication protocol [5,7,19,20]. TID is sometimes named differently in different protocols. In Reference [5], the PID of a mobile user is used. In Reference [6], the alias instead of a mobile user's identity is adopted. Many authentication protocols differ in that how TID is chosen.

Using TID is not sufficient to ensure anonymity of a mobile user if untraceability is not ensured. Therefore, TID has to be constantly changing; otherwise, a TID's activity profile can be established and lead to revealing

DOI: 10.1002/sec

the user's identity. Furthermore, there should not be a long-term key visible to network entities from which the identity of the mobile user is hidden; otherwise, profiling the activity of a fixed key could lead to the violation of the anonymity. The visible long-term key can also be exploited by an intruder [5]. Therefore, wireless authentication protocols often employ the mechanism to keep visible keys fresh.

3.4. Session Key Freshness and Backward and Forward Secrecy

It is important to make a session key fresh, which makes attack on the security system more difficult. In general, a session key will be discarded when it is used for one or a few times, and a new session key shall be generated. Therefore, an authentication protocol often has two phases, authentication phase and session key renewal phase.

In the session key renewal phase, the property of backward and forward secrecy [19] needs to be maintained. Backward secrecy implies that by using a comprised session key the intruder cannot find out any preceding session keys while forward secrecy implies that intruder cannot find out any future session keys. This pair of properties are sometimes overlooked in the session key renewal phase and are not adequately analyzed, see e.g., Reference [20].

4. Attacks and Protocol Analysis

The authentication protocols design often shows attack-fix-attack cycles. Wireless authentication protocols preserving anonymity and untraceability have never been exceptions due to its complexity. Through the research, a list of common attacks and their corresponding solutions have been unearthed. These attacks [5–7][16,19–24] include but not limit to parallel session attack, reflection attack, interleaving attack, attack due to type flaw, attack due to name omission, attack due to misuse of cryptographic service, side channel attack, timing analysis attack, implementation dependent attack, binding attack, encapsulation attack, misplaced trust in server attack, chosen-plaintext attack, cipher-text-only attack, confabulation attack, dictionary attack, guessing attack, colluding attack, and passive attack. These attacks and solutions have greatly shaped the current authentication protocol design. In the following, we will list a few common attacks which are often analyzed along with their proposed wireless authentication protocols. Note that we do not intent to be exhaustive because the list is still growing and many attacks are overlapping in nature.

4.1. Message Replay Attack

It is sometimes referred as message replaying attack, or simply replaying attack. In this attack, Malice records an old message and uses it to establish communications with either Alice or Bob. If a protocol cannot distinguish an old message and a fresh message, the protocol can be compromised because the old message is legitimate messages generated by the participating parties. Many protocol proposals have used this attack to examine the proposed protocols [5,20,23,24].

To counter this attack, a general solution is use a nonce which varies at each session, the nonce is usually a time stamp. Evidently, the time stamp nonce is used in all protocols we have reviewed, such as References [5,7,19,20,22–24]. Though the message can be replayed; however, the nonce has been altered when a participant receives the message again. Therefore, the key used to encrypt the message has been altered and the participating party cannot decrypt the message properly. The methodology is usually referred as 'self-encryption' [5]. It requires a distributed clock to use time stamp nonce. The distributed clock is arguably difficult to obtain. The nonce can be a random number as well. *Via* the random number nonce and encryption process on the nonce, an extra exchange between the two authenticating principals can be used to overcome the freshness deficiency, thus defeats the message replay attack without the requirement of maintaining distributed clock [25].

4.2. Forgery Attack

A forgery attack is an example of the Man-in-the-Middle attack [16]. It is sometimes referred as a fraud attack as in Reference [5]. In this attack, an intruder intercepts a legitimate message and alters it before sending it to the intended party of the original message. If the message is an acknowledgement, the attacker can then impersonate the message sender of the original message.

To defeat this attack, all protocol participants need to provide data-origin authentication service on both directions of message exchanges [16], see e.g., References [5,19]. This can be achieved by digital signature schemes or message authentication codes.

DOI: 10.1002/sec

4.3. Exhaustive Search Attack

The exhaustive search attack is sometimes called the brutal force attack. For example, it can be done by searching the entire key space or search the entire space for a random number used as nonce. Many public-key based authentication protocols rely on the property that the mathematical problem to acquire a private key is computational intractable. However, if the key space is too small, it is feasible to find the key through exhaustive search attack. Therefore, in the authentication protocols, whenever a random number is used, the space of the random number has to be large enough, see e.g., [5,20].

4.4. Passive Attack

Transmissions in wireless open media permits attacks to be performed passively against the *ad hoc* networks [26–28]. The behavior of such attacks is very different from other related security problems such as network disruption and 'denial-of-service' attacks. The passive attackers will avoid such aggressive schemes, so to be as 'invisible' as possible, until it traces, locates, and then physically destroys the targeted nodes. They can also try to be protocol compliant, instead of altering, inserting, dropping messages, so that they are harder to be detected before potential devastating physical attacks are launched. The passive attackers eavesdropping wireless transmissions, store and analyze the data to obtain or infer useful contents, identities, relationships of identities, and locations in the network. Such attacks can be very harmful, because when they can be detected, the damage may have already made. The solutions used in anonymous *ad hoc* routing protocols typically hide node identities and relationships on a routing path via sending them in a self-encrypted and decrypted layer of the network. [26,27].

5. Survey of Wireless Authentication Protocols Preserving User Anonymity and Untraceability

In this section, we will survey a few wireless authentication protocols. We name each protocol using the authors' last names. Throughout the section, we use the notations given in Table I.

5.1. Zhu–Ma Authentication Protocol

Zhu and Ma proposed a directory-based authentication protocol in Reference [20]. Although the protocol is a public-key authentication protocol, only secret-key cryptography is used by mobile users. In the protocol, a directory server, i.e., a certification authority (CA), issues X.509-like [29] certificates Cert_H and Cert_V to a mobile user's home network (H) and visiting network (V), respectively. Cert_H takes the following form, $\text{Cert}_H = \text{ID}_H||\text{PK}_H||T_H||\text{LF}_H||E_{\text{SK}_{\text{CA}}}(\text{ID}_H||\text{PK}_H||T_H||\text{LF}_H)$, where ID_H, PK_H, T_H, and LF_H are the identity, the public key of the home network, the issuing time, and the life time of the certificate, respectively. $E_{\text{SK}_{\text{CA}}}(\text{ID}_H||\text{PK}_H||T_H||\text{LF}_H)$ is actually a signature of the certificate. The signature is signed by the CA's private key SK_{CA}. The genuineness of the

Table I. Notations.

M	mobile user
H	home network
V	visiting network
CA	certification authority
ID_x	principal x's identity
Cert_x	x's public key certificate issued by CA.
$h(\cdot)$	one-way hash function
T_x	time stamp generated at principal x.
N_x, N'_x	large random numbers generated at principal x.
K_{xy}	key shared between principals x and y.
SK_x	private key owned by x
PK_x	public key owned by x
$E_{K_{xy}}$, E_{PK_x}, and E_{SK_x}	cryptographic procedure (such as encryption) using key K_{xy} or PK_x, respectively. Whether the procedure is a public-key based or secret-key based one depends on the notation of keys.
$\|\|$	message concatenation
TID_M	temporary identifier or pseudo identifier of mobile user M.

DOI: 10.1002/sec

signature can be verified by the corresponding public key of the CA. Cert_V has the similar form. The home network issues a smart card to mobile user M. The smart card contains hash function $h(\cdot)$, ID_H, and $r = h(N_H \,||\, \text{ID}_H) \oplus h(N_H \,||\, \text{ID}_M) \oplus \text{ID}_H \oplus \text{ID}_M$. The home network delivers $\text{PW}_M = h(N_H \,||\, \text{ID}_M)$ to M through pre-established secure channel. The delivery of PW_M is outside the scope of the protocol. M calculates a temporary identifier by $\text{TID}_M = r \oplus \text{PW}_M = h(N_H \,||\, \text{ID}_H) \oplus h(N_H \,||\, \text{ID}_M) \oplus \text{ID}_H \oplus \text{ID}_M \oplus h(N_H \,||\, \text{ID}_M) = h(N_H||\, \text{ID}_H) \oplus \text{ID}_H \oplus \text{ID}_M$. The protocol has two phases. The first phase is an authentication process and the second phase is a key refreshment process.

5.1.1. Authentication process

The authentication process is depicted in Figure 2.

In the authentication process, M chooses two nonces, N_M and T_M, where N_M is a random number and T_M is the time stamp of the message. $K_{MH} = h(T_M \oplus \text{PW}_M) = h(T_M \oplus\ h(\text{ID}_M \,||\, N_H))$ is a temporary secret key. V chooses two nonces, N_V and T_V, where N_V is a random number and T_V is the time stamp of the message. H also chooses two nonces, N'_H and T_H, where N'_H is a random number and T_H is the time stamp of the message. In steps 2 and 3, the signatures of the two messages are signed by V and H's private keys, $\text{S}K_\text{V}$ and $\text{S}K_\text{H}$, respectively. H and V can verify the integrity of the messages using their corresponding public keys.

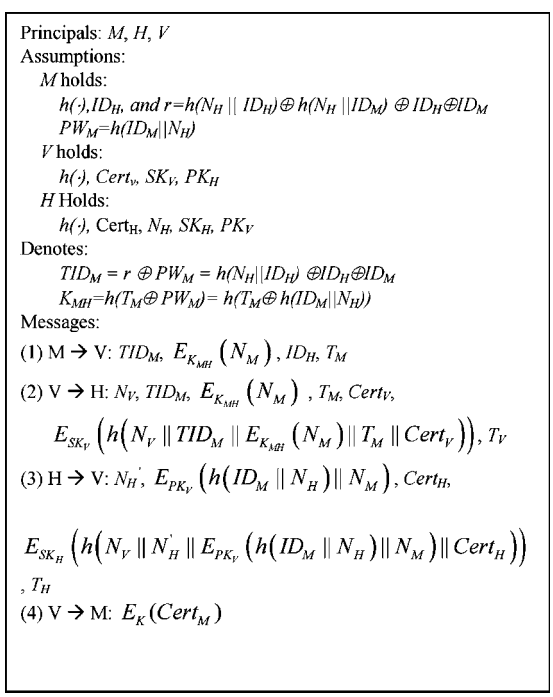

Fig. 2. Authentication process of Zhu–Ma's authentication protocol

In step 1, N_M is encrypted by secret key K_{MH}. $E_{K_{MH}}(N_M)$ constitutes a challenge that M presents to H. It is first sent to V and in turn forwarded to H by V.

The purpose of step 2 is (1) to authenticate V to H and (2) to authenticate M to H. Upon receiving the message, H decides if Cert_V is valid by the corresponding public key issued by CA. If Cert_V is valid, H obtains M's true identity by performing the following calculation on the received TID_M, $\text{ID}_M = h(N_H \,||\, \text{ID}_H) \oplus \text{TID}_M \oplus \text{ID}_H$. H can determine if M is a legal user through its temporary identity TID_M because TID_M actually contains the information known only by M and H. Key K_{MH} can be obtained by $K_{MH} = h(T_M \oplus h(N_H \,||\, \text{ID}_M))$.

The purposes of step 3 are for H: (1) to inform V that M is successfully authenticated with H, (2) to authenticate H with V. The message is signed by H's private key. V verifies Cert_H to determine the identity of H. In the end, V issues a temporary certificate to M. The certificate is encrypted by secret key $K = h(\text{ID}_M \,||\, N_H) \oplus N_M$, where N_M is what H responses with M's challenge and $h(\text{ID}_M \,||\, N_H)$ can be obtained by decrypting $E_{KU_V}(h(\text{ID}_M)||N_M)$ using V's private key.

5.1.2. Session Key Renewal Process

The session key is renewed at each session. For this purpose, M choose a different nonce x_i for each session. For the very first session key, the nonce is N_M. The secret key used to encrypt the temporary certificate is the session key for the next session. The ith session key takes the form of $k_i = h(\text{ID}_M) \oplus x_{i-1}$. Nevertheless, at the ith session, M sends a message of the following form to V, $\{\text{Cert}_M,\ E_{k_i}\{x_i||\text{Cert}_M||\text{Others}\}\}$. V first checks if Cert_M is valid. Then it computes k_i to encrypt the second part of the message. Cert_M in the second part of the message is intended to be used to verify the integrity of the message. V then saves x_i for generating next session key.

5.1.3. Discussion

In the protocol, the mobile user's identity is hidden from both the visiting network and the eavesdroppers. However, the mobile user's TID will

DOI: 10.1002/sec

not change. Nevertheless, the protocol does not have any untraceability. One may discover the mobile user's true identity by profiling a TID's activity.Furthermore, as pointed in Reference [19], this protocol also has security flaws. The fix of this protocol is given in Reference [19] and is summarized in the next subsection.

5.2. Lee–Hwang–Liao's Security Enhanced Mutual Authentication Protocol

As suggested in Reference [19], Zhu–Ma's authentication protocol has three security flaws.

1. First, Zhu–Ma's protocol does not achieve perfect forward secrecy [20]. In Zhu–Ma's protocol, if the attacker somehow obtains a session key k_i and the attacker intercepts message $\{\text{Cert}_M\|, E_{k_i}\{x_i||\text{Cert}_M||\text{Others}\}\}$ sent during session key renewal phase, the attacker can use the key to decrypt the message and the attacker can obtains the nonce x_i. The attacker can then computes the future session keys by plugging the nonce into formula $h(\text{ID}_M) \oplus x_{i-1}$.
2. Second, Zhu–Ma's protocol does not counter against a forgery attack. In the authentication phase of Zhu–Ma's protocol, an attacker can intercept the message sent at step 1, i.e., $\{\text{TID}_M, E_{K_{MH}}(N'_M), \text{ID}_H, T_M\}$. The attacker can then modify the message to $\{\text{TID}_M, E_{L'}(N'_M), \text{ID}_H, T_M\}$ where N'_M and L' are two random numbers chosen by the attacker. Then H will be deceived and authenticates the attacker because H can derive M's true identity from the received TID_M.
3. Third, Lee, Hwang, and Liao [19] continuous argue that Zhu–Ma's protocol does not achieve mutual authentication. They argue that if an attacker intercepts $E_K(\text{Cert}_M)$ and alters it to $E_{K'}(\text{Cert}'_M)$, where Cert'_M and K' are two random numbers chosen by the attacker, M will receive a wrong temporary certificate.

An enhanced authentication is then proposed in Reference [19] to address these security flaws. The authentication process of the enhanced protocol is shown in Figure 3.

The enhancements are highlighted as follows. In steps 1 and 2, $E_{K_{MH}}(N_M)$ is replaced by $E_{K_{MH}}(h(\text{ID}_M)||N_M||N'_M)$. In step 3, $h(\text{ID}_M||N_M)||N'_M$ is replaced by $h(\text{ID}_M||N_M||N'_M)$. In step 4, $E_K(\text{Cert}_M)$ is replaced by $E_K\{\text{Cert}_M \,||\, h(N_M \,||\, N'_M)\}$.

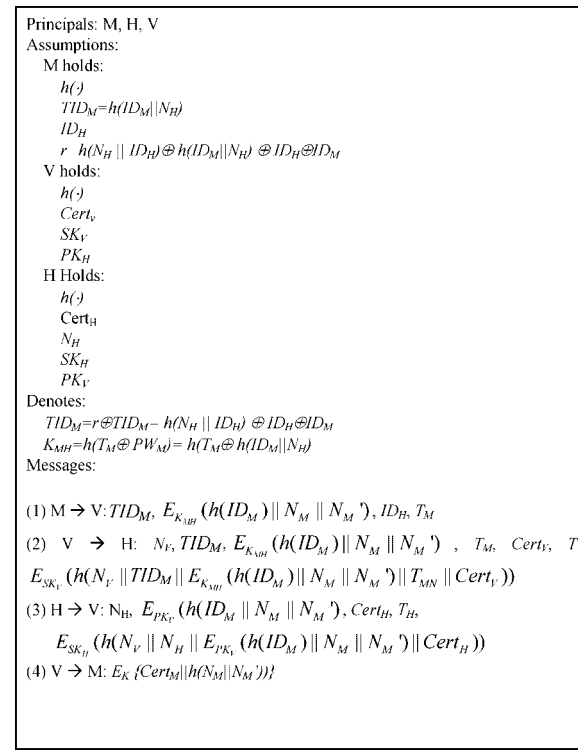

Fig. 3. Authentication process of Lee–Hwang–Liao's authentication protocol.

In Zhu–Ma's protocol, a user's TID is not signed in any of the messages. When an attacker impersonates TID, V and H cannot discover it. In the enhanced protocol, $E_{K_{MH}}(h(\text{ID}_M)||N_M||N'_M)$ can be regarded as a signature of TID_M. In step 2, when the message arrives at H, H derives ID_M and $h(\text{ID}_M)$ by applying $\text{ID}_M = h(N \,||\, \text{ID}_H) \oplus \text{TID}_M \oplus \text{ID}_H$ where N and ID_H are the knowledge of H and TID_M is received in the message. H then obtains $h(\text{ID}_M)$ from the message by decrypting $E_{K_{MH}}(h(\text{ID}_M)||N_M||N'_M)$, where $K_{MH} = h(T_M \oplus \text{PW}_M) = h(T_M \oplus h(\text{ID}_M \,||\, N_H)$. If one attacker impersonates TID by altering message $\{\text{TID}_M, E_{K_{MH}}(h(\text{ID}_M)||N_M||N'_M), \text{ID}_H, T_M\}$ to $\{\text{TID}_M, E_{K'_{MH}}(h(\text{ID}_M)'||N'_M||N''_M)\}$. Then the calculated two versions of $h(\text{ID}_M)$ will not match. Therefore, the enhanced protocol defeats the forgery attack.

In Zhu–Ma's protocol, M challenges both V and H with a nonce. In the message for V to deliver the temporary certificate to M, the response of the challenge is in secret key $K = h(\text{ID}_M) \oplus N_M$. However, the message does not provide sufficient information for M to verity the response, i.e., in case K is forged, M cannot verify the validity of

DOI: 10.1002/sec

Cert_M. The fix to this is to add extra information to the message as illustrated in Figure 3. M can use the added information to validate Cert_M. To do this, M only needs to compare the decrypted version of $h(N_M \,||\, N'_M)$ with its local version. This revision then completes the mutual authentication process between M and V and that between M and H.

In the session key renewal phase as shown in Figure 4, an attacker will not be able to calculate future session keys if the attacker somehow obtains a session key k_i. This is because the session is now in the form of the attacker $k_i = h(\text{ID}_M \,||\, x) \oplus x_{i-1}$ and x is unknown to the attacker.

5.2.1. Discussion

In the fix protocol, the authors argue that they can make the protocol more secure by calculating session $k_i = h(\text{ID}_M \,||\, x) \oplus x_{i-1}$ for both M and V. However, $h(\text{ID}_M \,||\, x)$ is unknown to V. Nevertheless, how V can obtain $h(\text{ID}_M \,||\, x)$ remains an issue.

5.3. Jiang–Lin–Shen–Shi's Mutual Authentication and Key Exchange Protocols

Jiang, Lin, Shen, and Shi proposed two authentication protocols in Reference [5]. Among these two protocols, one is based on secrete-key cryptography and the other is based on public-key cryptography (a self-certified authentication protocol). We denote them as JLSS1 and JLSS2, respectively.

5.3.1. JLSS1

JLSS1 is divided into two sub-protocols (phases), a mutual authentication protocol (MAP) and a one-time session key renewal protocol (SKRP). JLSS1 is based on the *secret-key cryptography* and there must be a *pre-established secure channel* between the home network and its mobile users.

The goal of the MAP is to authenticate the mobile user to the visiting network (V) through the home network (H) while keeping its identity hidden from the visiting network. To achieve this goal, the home network assigns a pseudo identity (PID_M) to the mobile user (M) based on its true identity (ID_H), a large random number (N_M), and the identity of the home network (ID_H) as follows:

$$\text{PID}_M = h(N_M || \text{ID}_H) \oplus \text{ID}_M \oplus \text{ID}_H \qquad (1)$$

(1) M → V: $Cert_M E_{k_i}(N_{M,i} \,||\, Cert_M \,||\, Others)$

Fig. 4. Session key renewal process of Lee–Hwang–Liao's authentication protocol.

PID_M is delivered to M through the pre-established secure channel.

K_{MH} and K_{VH} are the secret keys between M and H, V and H, respectively. K_{MH} is a long-term key and calculated as $K_{MH} = f(\text{ID}_H)$, where $f(\cdot)$ is a one-way function and is the common knowledge of both M and H. K_{VH} is a long-term key between the home network and the visiting network, and it must be the common knowledge between V and H. This can be done is a pre-arranged manner such as face-to-face arrangement or delivered through pre-established secure channel between V and H. Then the authentication process runs as Figure 5, where r_M and r_V are two random numbers generated by M and V, respectively, and t_V is the time stamp when the message is generated by V. Note that according to Reference [26] H recovers the identity of the mobile user (IDM) by computing

$$\text{ID}_M = \text{PID}_M \oplus h(N_M || \text{ID}_H) \oplus \text{ID}_H \qquad (2)$$

in step 3. Once ID_M is discovered, $K_{MH} = f(\text{ID}_M)$ can be computed. Then H knows both K_{MH} and K_{VH} and the decryption and encryption in step 3 can be carried on. If t_V suggests that the message is too old, the authentication will be terminated to counter the replay attack. In the final step, K_{auth} is the initial session key for further communication. K_{auth} is known by V because it is actually computed as follows:

$$K_{\text{auth}} = r_M \oplus r_V \qquad (3)$$

and r_M and r_V are known to V at this moment.

The MAP only hides the identity from the visiting network and the eavesdroppers. However, one can track a PID of a mobile user and profile its activity, and then the true identity may be revealed. The SKRP renews the session key and the PID of a mobile user as Figure 2.

In Figure 6, K_{i-1} is the session key generated at the $(i-1)th$ protocol run. $\text{PID}_{M,i}$ is the pseudo identity of the mobile user used in the ith protocol run. K_{i-1} is

DOI: 10.1002/sec

Principals: M, H, V
Assumptions:
M holds:
$h(\cdot)$
PID_M
ID_M
$E_{K_{MH}}$
V holds:
$h(\cdot)$
$E_{K_{VH}}$
H Holds:
$h(\cdot)$
$E_{K_{MH}}$
$E_{K_{VH}}$
Denotes:
$PID_M = h(N_M||ID_H) \oplus ID_M \oplus ID_H$
Messages:
(1) M → V: ID_H, PID_M, $E_{K_{MH}}(r_M \| K_{MH})$
(2) V→ H: PID_M, $E_{K_{VH}}(r_v \| t_v \| E_{K_{MH}}(r_M \| K_{MH}))$
(3) V←H: $E_{K_{VH}}(r_v \| r_M \| h(ID_M))$, $E_{K_{MH}}(r_M \| r_v \| ID_v)$
(4): M←V: $E_{K_{MH}}(r_M \| r_v \| ID_v)$
(5): M→V: $E_{K_{auth}}(K_{auth})$

Fig. 5. The MAP of the JLSS1 protocol.

(1) M → V: ID_V, $PID_{M,i}$, $E_{K_{i-1}}(r_{M,i} \| K_{i-1})$
(2) M ← V: $E_{K_{i-1}}(r_{M,i} \| r_{v,i} \| ID_H)$
(3) M → V: $E_{K_i}(K_i)$

Fig. 6. The SKRP of the JLSS1 protocol.

computed as follows:

$$K_{i-1} = r_{M,i} \oplus r_{V,i} \tag{4}$$

K_{auth} in the MAP is the initial value, i.e., $K_{\text{auth}} = K_0$. $\text{PID}_{M,i}$ is computed as follows:

$$\text{PID}_{M,i} = h(\text{ID}_M) \oplus r_{M,i} \tag{5}$$

Note that $r_{M,i}$ and $r_{V,i}$ are two random numbers generated at M and V, respectively. Cleary K_{i-1} and $\text{PID}_{M,i}$ vary at each session. As shown in Figure 6, V obtains $r_{M,i}$ by computing

$$r_{M,i} = \text{PID}_{M,i} \oplus h(\text{ID}_M) \tag{6}$$

5.3.2. JLSS2

In JLSS2, both M and V register with H, which is regarded as a temporary trusted third party (TTP) for both M and V. H will serve as a witness once the registrations are successful, and M shall directly negotiate with V without accessing its home network. The scheme that the mobile user's home network is used as a temporary TTP is a self-certified scheme [5,18].

The self-certified scheme is based on public-key cryptography. In fact, the JLSS2 uses both public- and secret-key cryptography for M to authenticate with H to avail V's service, while the JLSS1 uses only secret cryptography.

The JLSS2 is also divided into two sub-protocols, the MAP and the SKRP. We start with the MAP of the JLSS2.

H chooses two large secret safe prime numbers p and q such that $p-1=2p'$, $q-1=2q'$, and p' and q' are also primes. It then computes $n=pq$. A public exponent e is chosen such that e *is coprime to* $\phi(n)=(p-1)(q-1)$, and a secret exponent d is chosen such that $ed=1 \bmod \phi(n)$. H also picks a large integer number $u<p'q'$. It then computes element $g \in Z_u^*$ where Z_u^* is the multiplicative group of order u. f is a public one-way function that will output positive integers less than p' and q'. H publishes (g, u, f, n) as public key material and keep r as secret key material. Any other components are discarded.

Suppose that r_M and r_V are large random numbers generated by M and V, respectively. Then two numbers y_M and y_V are chosen such that $y_M = g^{r_M} \bmod(n)$ and $y_V = g^{r_V} \bmod(n)$. Denote two witnesses issued by H to M and V respectively as w_M and w_V. They are computed as follows:

$$w_M = \left((y_M \oplus I_M)^{f(\text{ID}_M)^{-1}}\right) \bmod(n) \tag{7}$$

$$w_V = \left((y_V \oplus I_V)^{f(\text{ID}_V)^{-1}}\right) \bmod(n) \tag{8}$$

where I_{M} and I_V are information related to their corresponding identity such as address, telephone number, etc. Denote M's temporary identity as TID_M, which is calculated by $\text{TID}_M = E_{K_{MH}}(g^{r_M} \oplus ID_M)$. Note that the witnesses are the guarantee issued by the

DOI: 10.1002/sec

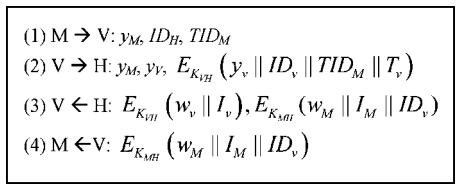

Fig. 7. The MAP of the JLSS2 protocol

home network to bind the identity of the mobile user to the public key.

Then the MAP sub-protocol runs as Figure 7.

Note in Figure 7, the mobile user's temporary identity is calculated as $\mathrm{TID}_M = E_{K_{MH}}(g^{r_M} \oplus \mathrm{ID}_M)$, where $K_{MH} = (\mathrm{PK}_H)^{r_M}$ is the shared key between M and H and PK_H is the public key. Though the protocol is a self-certified public-key protocol, it uses secret-key cryptography as well. H will deliver shared keys K_{MH} and K_{VH} to M and V, respectively. In step 2, H obtains TID_M by decrypting the last part of the message. The user's identity can be computed, i.e., $\mathrm{ID}_M = E^{-1}_{K_{MH}}\left(E_{K_{MH}}(g^{r_M} \oplus \mathrm{ID}_M)\right) \oplus g^{r_M}$. H then calculates two witnesses as described before and send them to V as indicated in step 3. V then finishes authenticating with H if y_V equals to $\left((w_V)^{f(I_V)} \bmod (n)\right) \oplus I_V$. In the last step, M authenticates with V if y_M equals to $\left((w_M)^{f(I_M)} \bmod (n)\right) \oplus I_M$.

The SKRP sub-protocol is described in Figure 8.

For V, the session key is computed using the material it receives from M. M first generates a number $t_M \in Z^*_u$. Then it calculates $y_M = ((w_M)^{f(I_M)} \bmod (n)) \oplus I_M$ and $K_V = (y_M)^{t_V}\left(g^{t_M}\right)^{r_V}$. Then the session key is $K_{MV} = h(K_V)$. For M, the session key is computed using the material it receives from V. V first generates a number $t_V \in Z^*_u$. Then it calculates $y_V = ((w_V)^{f(I_V)} \bmod (n)) \oplus I_V$ and $K_M = (y_V)^{t_M}\left(g^{t_V}\right)^{r_M}$. Then the session key is $K_{MV} = h(K_M)$, where $h(K_M) = h(g^{r_V t_M + r_M t_V} \bmod (n)) = h(K_V)$. This sub-protocol yields different session key for each renewal.

5.3.3. Discussion

In JLSS1, the protocol suggests that V computes the session key K_i based on Equation (4). For this purpose, V needs to obtain $r_{M,i}$ by computing Equation (5). Therefore, V needs to find out matching $h(\mathrm{ID}_M)$ considering that V has many roaming users. It is a difficult task unless V keeps track of all $\mathrm{PID}_{M,i}$ and $h(\mathrm{ID}_M)$.

The SKRP of JLSS1 and JLSS2 can only make the mobile user's activity untraceable to eavesdroppers, i.e., to achieve untraceability level 1.

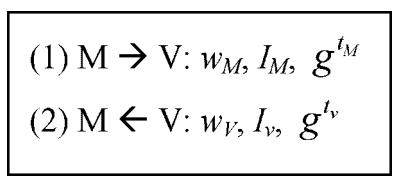

Fig. 8. The MAP of the JLSS2 protocol. t_M and t_V where $t_M \neq t_V$ denotes two elements of Z^*_u.

5.4. Go–Kim's Authentication Protocol Preserving User Anonymity

Go and Kim proposed an authentication protocol in Reference [22]. The protocol is based on public-key cryptography. It uses digital signature of messages and a Diffie–Hellman key exchange. A TTP issues public key certificate to each protocol participants. The protocol is illustrated in Figure 9.

1. In step 1, M chooses a random number r_M. M constructs a $\mathrm{TID}_M = (E_{K_{MH}} h(\mathrm{ID}_M) \oplus g^{r_M})$ to hide user's true identity. Denote H's secret key as SK_{H}, which is a long-term key. K_{MH} is calculated as $K_{MH} = g^{SK_H \cdot r_M}$.
2. In step 2, r_V is the random number chosen by V. In the message, the signature of the message is signed by V's secret key SK_V. T_V is the time stamp of the message. Cert_V is the public key certificate issued by the TTP. It serves the purpose for authenticating V to H and later to M in step 4. Step 2 finishes two authentications. (1) H computes $K_{MH} = g^{SK_H \cdot r_M}$, obtains $h(\mathrm{ID}_M) \oplus g^{r_M}$ from decrypting TID_M, then looks up ID_M by using $h(\mathrm{ID}_M) = (h(\mathrm{ID}_M) \oplus g^{r_M}) \oplus g^{r_M}$. This finishes the authentication of M to H. (2) H uses Cert_V to authenticate V to H.

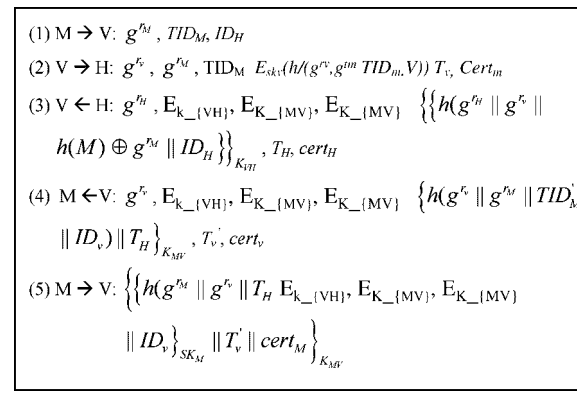

Fig. 9. Go–Kim's authentication protocol.

DOI: 10.1002/sec

3. In step 3, r_H and T_H are a random number and the time stamp of the message, respectively. H needs to (1) authenticate H to V to finish the mutual authentication between V and H; (2) pass sufficient information to V such that V can pass them to M to finish authentication of H to M. K_{VH} takes the form of $K_{VH} = h1(g^{r_M r_V} || g^{SK_V r_M})$ where SK_V is V's secret key. Upon receiving the message, V computes the message signature and verifies Cert_H. H can then be authenticated.
4. In step 4, r_V and T'_V are a random number and a time stamp of the message, respectively. V calculates a new TID_M for M, i.e., $\text{TID}'_M = h(g^{r_M r_V} || h(\text{ID}_M))$. Cert_V is sent to M to authenticates V to M.
5. In step 5, Cert_M is sent to the V to finish the mutual authentication between M to V and M to H. The session key is calculated as $K'_{MV} = h_2(g^{r_M r_V}, g^{SK_V r_M})$ where $h_2(\cdot)$ is a hash function.

5.4.1. Discussion

There is a weakness in the protocol. In step 4, the message does not carry any information that this message is from H. Therefore, the mutual authentication between M and H is not complete. If V and an eavesdropper are both attackers, the eavesdropper can play the role of H and all the messages sent from M will be compromised. To fix the protocol, one may simply add H's certificate to the message in step 4.

5.5. Park's Authentication Protocol based on Secret Key Certification and Error-Correction Code

Park proposed a one-way authentication protocol with anonymity and untraceability [7]. The protocol is on the secret key certification and error-correction codes.

The protocol requires an authentication server (AS). The AS sends a mobile user a secret certificate through a pre-established secure channel. The certification uniquely identifies a mobile user at the AS. The authentication protocol becomes a process that the AS verifies whether a mobile user possesses the certificate.

The protocol can be described in two phases as suggested in Reference [7].

5.5.1. Secret Certificate Initialization and Distribution

This phase happens through a pre-established secure channel or in a face-to-face manner. It happens only once as a setup process. Pseudo random number generator $h(\cdot)$ generates a $2n$-bit random number when receiving a n-bit input. The output random number is divided into two halves. The left and right halves are denoted as $h_L(x)$ and $h_R(x)$, respectively.

The mobile user generates authentication tokens $x_i$and k_i as follows:

$$x_{i-1} = h_L(x_i)$$
$$k_i = h_R(x_i)$$

where $i = s, s-1,\ldots,1$. Then the mobile user sends x_0 to the AS. The AS calculates a secret certificate $m = (E_{K_{AS}}(\text{ID}_M || x_0)$, where K_{AS} is a secret key only known to the AS.

Though ID_M is concealed in the certificate, we cannot directly use it. This is because the certificate is a fixed value. The protocol loses untraceability if directly using it. To ensure untraceability, the certificate is translated in the following manner.

The AS chooses a linear error-correction code [30] of length N, dimension K, and minimum distance D, i.e., (N, K, D), which has an efficient encoding algorithm. The AS then chooses a secret K-by-N generator matrix $\mathbf{G}$ together with a secret $(N-K)$-by-N parity-check matrix $\mathbf{H}$. Then the AS encodes the secret certificate m into a N-bit codeword by $\mathbf{c} = \mathbf{m} \cdot \mathbf{G}$, where m is regarded as a K-tuple (i.e., a vector of length K). The AS only keeps G, H, and K_{AS} and sends $\mathbf{c}$ to M.

5.5.2. One-way authentication process

In this process, M sends $r_i = c + e^{(i)}$ to the AS. $e^{(i)}$ is a N-bit error vector used in the ith session. As suggested in Reference [7], if the Humming weight of $e^{(i)}$ is less than or equal to $(D-1)/2$, r_i can be decoded into $\mathbf{c}$ with knowledge of $\mathbf{H}$. Once the AS receives r_i, the AS decodes r_i to obtain ID_M and x_0. M can then be authenticated.

Since r_i is different at each session and eavesdroppers do not possess it, untraceability can be ensured. In fact, the purpose of the error vector is for ensuring untraceability.

This protocol is unique because no TID is used and there is only one message in the authentication process.

DOI: 10.1002/sec

Table II. Comparisons.

Protocol	# of SKCS	# of PKCS	# of $h(.)$	# of $e^{(.)}$	# of messages
Zhu–Ma protocol	2	0	0	0	2
Lee–Hwang–Liao protocol	2	0	2	0	2
Jiang–Lin–Shen–Shi protocol I	3	0	0	0	3
Jiang–Lin–Shen–Shi protocol II	1	0	0	2	2
Go–Kim protocol	2	0	1	2	3

Though as argued in Reference [7], this one-way authentication protocol can be easily extended to MAP, the actual MAP is not given.

However, Dominguez argues in Reference [21] that the anonymity, the untraceability, and the security of the authentication protocol defined in Reference [7] are not guaranteed.

As pointed in Reference [21], the protocol has at least three weaknesses regarding to the anonymity, untraceability, and the data security.

- First, it is possible for a legitimate user to trace the mobile user. Suppose the authentication messages, $r_i = c + e^{(i)}$ and $r_j = c + e^{(j)}$ are intercepted at session i and j by an eavesdropper. Then the hamming weight of $r_i + r_j = c + e^{(i)} + c + e^{(j)}$ is less than $2t + 1$. If the eavesdropper is a legitimate user, the eavesdropper knows t. Then it is possible for the eavesdropper to tie two sessions together by examining the property of the hamming distance.
- Second, the protocol does not illustrate the network model being used. If a mobile roams to a visiting network, the visiting network may need to forward the authentication message to the AS. That is to say, some system entities may be able to know certificate **c**. The owner of **c** can always be traced since the hamming distance $c + r_i = c + c + e^{(i)}$ is less than t.
- Third, the protocol is built upon a linear system. The secret generator matrix held by the AS may be recovered by a large number of encoded certificates. In Reference [21], an algorithm is sketched to recover the generator polynomial. Therefore, the identity of a mobile user could be recovered and the session key can be revealed.
- Furthermore, the protocol hides important details about the network model [21]. For example, where is the AS? Can a mobile user directly talk to the AS? Therefore, to make Park's protocol feasible, many details need to be further explored.

5.6. Summary

Table II shows a comparison of surveyed protocols. We assume the visiting and home domains have extensive computational power, and the only limitation is the computational power at mobile clients. The comparison shows the computational cost at mobile devices. A SKCS means an encryption/decryption operation in secret-key cryptography system and a PKCS indicates an encryption/decryption operation in public-key cryptography system. An $h(.)$ denotes a call to one-way hash function and an $e^{(.)}$ stands for an exponential operation. We also count the number of messages sent to/from the mobile clients.

6. Anonymity in Routing Protocols for Wireless *Ad Hoc* Networks

Since wireless *ad hoc* networks are often deployed in hostile environment such as battlefield, security is very important to wireless *ad hoc* networks. Wireless authentication protocols discussed in the above section can make the data secure and the identities of the source and the destination nodes anonymous to eavesdroppers and other nodes. However, in wireless *ad hoc* network, each network node sometimes plays a role of router. A compromised node not only poses a danger to the data it forwards, but may also reveal the locations of the source node and the destination node. Nevertheless, the security of the network and the safety of the source and the destination nodes can be compromised. Therefore, it will be beneficial to the network security if the following anonymity is enforced: a routing node does not have sufficient knowledge of other network entities' identities, their locations, and relationships among them. However, this is a challenging problem because the very role of the router is to route data to the appropriate destination. Anonymous routing protocols are designed to tackle the challenge.

DOI: 10.1002/sec

6.1. Secure Distributed Anonymous Routing Protocol

In Reference [26], a routing protocol, called secure distributed anonymous routing protocol (SDAR), for achieving anonymity in wireless *ad hoc* networks was proposed. This protocol requires a trusted certificate authority outside the *ad hoc* network, which issues public and private key to wireless nodes inside the network.

A network node's community is the set of nodes that are within 1-hop distance. The node itself is called the central node. Each node broadcasts its public key through simple HELLO message. The central node assigns trust values for each node in the community. A node that is not willing to cooperate for routing and data delivering in the manner that is required by the protocol will be regarded as a tempering node and has low trust value, which is ranging from 0 to 1. All the nodes in the community are divided into three classes according to two thresholds. The central node generates two sets of keys for the community, i.e., high trust level community key (HCK) and medium trust level community key (MCK). The class with high trust level gets both HCK and MCK while the class with medium trust level gets only MCK. The HCK and MCK are encrypted using the receiving node's public key when they are sent.

The SDAR has three phases, (1) path discovery phase; (2) path reverse phase; and (3) data transfer phase.

The goal of the path discovery phase is to establish a routing path through intermediate wireless nodes. However, none of the intermediate nodes knows the identity of the source node (S) and its knowledge about the network topology is limited to 1-hop distance.

In the path discovery phase, source node (S) sends a path discovery message to all its 1-hop neighbors. The message has the following format:

$$\text{Type}||\text{TRUST\_REQ}||\text{TPK}||E_{\text{PK}_R}(\text{ID}_R||K_S||\text{PL}_S)||P_S||$$
$$E_{K_S}(\text{ID}_S||\text{PK}_S||\text{TPK}||\text{TSK}||\text{SN}_{\text{Session\_ID}}||\text{Sign}_S(M_S))$$

where

$$M_S = \text{TYPE}||\text{TRUST\_REQ}||E||\text{TSK}||\text{ID}_R||K_S||\text{ID}_S||\text{PK}_S||\text{SN}_{\text{Session\_ID}_S}||\text{PL}_S||P_S$$

Note the following two important issues:

1. PK_R is the public key of the receiver node (R). One way to obtain it is through a certificate authority. $E_{\text{PK}_R}(\text{ID}_R||K_S||\text{PL}_S)$ is the encrypted version of the identity of source node, the symmetric key generated by the source node, and the length of the message. Note that the message is encrypted using key PK_R. Supposedly, the message can only be decrypted by the receiver (R). Therefore, *the identity of the source node is hidden from any other nodes except the receiver node*.
2. In message, TPK is the temporary public key generated by the source node for the path discovery session. Its corresponding private key is TSK. *Note* that nobody knows TSK except S.

Besides, also note the following issues:

1. K_S is the secret key generated by the source node. It is used to generate a signature of the message, which is used to prevent the replay attack.
2. Padding P_S is generated by the source node () to prevent the message size attack.
3. TYPE and TRUST_REQ are used to define the appropriate trust level. TRUST_REQ can be high, medium, and low. As we have discussed before, a route between the source and the receiver can only consist of nodes that has the same or higher trust level. Then badly behaved nodes will not be in the routing path.

Once receiving the message, the intermediate node adds $E_{\text{TPK}}(\text{ID}_i||K_i||\text{SN}_{\text{Session\_ID}_i}||\text{Sign}_{\text{ID}_i}(M_{\text{ID}_i}))$ to the received message and sends the complete message to its 1-hop neighbors. Supposed the message has reached the mth node, the complete message will have the following format:

$$\text{Type}||\text{TRUST\_REQ}||\text{TPK}||E_{\text{PK}_R}(\text{ID}_R||K_S||\text{PL}_S)||P_S||$$
$$E_{K_S}(\text{ID}_S||\text{PK}_S||\text{TPK}||\text{TSK}||\text{SN}_{\text{Session\_ID}}||\text{Sign}_S(M_S))||$$
$$E_{\text{TPK}}(\text{ID}_1||K_1||\text{SN}_{\text{Session\_ID}_1}||\text{Sign}_{\text{ID}_1}(M_{\text{ID}_1}))$$
$$\ldots$$
$$E_{\text{TPK}}(\text{ID}_i||K_i||\text{SN}_{\text{Session\_ID}_i}||\text{Sign}_{\text{ID}_i}(M_{\text{ID}_i}))$$
$$\ldots$$
$$E_{\text{TPK}}(\text{ID}_m||K_m||\text{SN}_{\text{Session\_ID}_m}||\text{Sign}_{\text{ID}_m}(M_{\text{ID}_m}))$$

Note that we denote the identity of the ith node as ID_i and the message becomes self-explanatory.

The receiver possesses the secret key to decrypt $E_{\text{PK}_R}(\text{ID}_R||K_S||PL_S)$. The message triggers the path reverse phase when the receiver receives the message. Note that the anonymity of the routing path is hidden

DOI: 10.1002/sec

to all the intermediate nodes except the receiver. This is because nobody else has secret key K_S except the source node and the receiver node. Without K_S, nobody else can decrypt the following part in the message:

$$E_{K_S}(\mathrm{ID}_S||\mathrm{PK}_S||\mathrm{TPK}||\mathrm{TSK}||\mathrm{SN}_{\mathrm{Session\_ID}}||\mathrm{Sign}_S(M_S))$$

which contains the key (TSK) to decrypt the parts of message that can construct a route, i.e.,

$$E_{\mathrm{TPK}}(\mathrm{ID}_1||K_1||\mathrm{SN}_{\mathrm{Session\_ID}_1}||\mathrm{Sign}_{\mathrm{ID}_1}(M_{\mathrm{ID}_1}))$$
$$\ldots$$
$$E_{\mathrm{TPK}}(\mathrm{ID}_i||K_i||\mathrm{SN}_{\mathrm{Session\_ID}_i}||\mathrm{Sign}_{\mathrm{ID}_i}(M_{\mathrm{ID}_i}))$$
$$\ldots$$
$$E_{\mathrm{TPK}}(\mathrm{ID}_m||K_m||\mathrm{SN}_{\mathrm{Session\_ID}_m}||\mathrm{Sign}_{\mathrm{ID}_m}(M_{\mathrm{ID}_m}))$$

The goal of the path reverse phase is to inform the source node that a path has been discovered and the data transfer phase can start. However, during the path reverse phase, neither the anonymity of the source node nor the route should be violated. Note that in the path discovery phase, each node generates a secret key (K_i) and put in the part that concatenated to the message. The receiver uses the secret key TSK to decrypt the part to find out not only the previous node in the route but also the secret key issued by the node. It then uses the key to encrypt all the secret keys (K_i) issued along the route and all the relevant information. The message that arrives at the *i*th node along the route has the following format:

$$\mathrm{Type}||\ E_{K_i}||(E_{K_{i-1}}(E_{K_{i-2}}\ldots(E_{K_2}(E_{K_1}(E_{K_s}(\mathrm{SN}_{\mathrm{Session\_ID}_1}||K_1||$$
$$\mathrm{SN}_{\mathrm{Session\_ID}_2}||K_2\ldots SN_{\mathrm{Session\_ID}_{i-2}}||K_{i-2}||\mathrm{SN}_{\mathrm{Session\_ID}_{i-1}}||K_{i-1}$$
$$\mathrm{SN}_{\mathrm{Session\_ID}_i}||K_i||\mathrm{SN}_{\mathrm{Session\_ID}_R}||\mathrm{PL}_R||P_R)....)...)...)...)...)$$

Note that when the message moves back from the receiver to the source, the message is decrypted by using the secret key issued by each node it passes. Then the message is comprehensive to the nodes in the route within 1-hop range. The anonymity of the routing information is ensured.

When the message arrives at the source, the data transfer phase shall begin.

6.2. Anonymous and Untraceable *Ad Hoc* Routing

Untraceability of an *ad hoc* network can be critical because the consequences often lead to locate and identify users who participate in the communications of interests. An anonymous and untraceable on-demand *ad hoc* routing protocol has been proposed in Reference [27], namely, *Anonymous On-Demand Routing protocol (ANODR)*. ANODR is designed to countermeasure passive attackers. It consists of an anonymous route discovery phase and then the anonymous data forwarding phase. In both phases, node identities are not directly used. Thus, each node does not know its immediate upstream and downstream nodes. Instead, the node only knows the physical presence of neighboring *ad hoc* nodes. This is achieved by establishing an anonymous virtual circuit through a special anonymous signaling procedure and using it for data delivery.

ANODR protocol uses the assumption of anonymous global trapdoor, i.e., the source has an established secret with the desired destinations. The global trapdoor is an encryption of a well-known tag message that can only be decrypted by the destination. Once the destination receives the flooded route request (RREQ) packet, it decrypts the global trapdoor and sees the well-known tag.

The anonymous signaling procedure is implemented with the route discovery of an on-demand routing protocol. The source creates the inner core of an 'onion' in the RREQ packet together with the anonymous global trapdoor of the destination. It then initiates the search for the destination by flooding the packet through the network. When the RREQ packet is flooded from the source to the destination, each RREQ forwarding node adds a self-aware layer to the onion.

Given a node N_i along a path, to form a layer to the onion, it encrypts a received onion with an arbitrary symmetric key K_i. Key K_i is kept to node N_i itself, because only K_i itself uses the key for decryption later. Thus the onion_i that the $\mathrm{N_i}$ broadcast will be:

$$\mathrm{onion}_i = E_{K_i}(E_{K_{i-1}}(E_{K_{i-2}}(\ldots(\mathrm{core})\ldots)))$$

The 'core' is a random nonce. Let $\mathrm{GLOBAL}_{\mathrm{trap}}$ denotes the anonymous global trapdoor of the destination. The RREQ is in the following format. PK_i is a one-time temporary public key for other nodes to encrypt the returning route reply (RREP) message. The node N_i will use its private key to decrypt. Each node will record PK_{i-1} from the upstream node and override

DOI: 10.1002/sec

the field with its own one-time public key. The use of the one-time public key PK_i will be clear in RREP message.

$$\langle \text{RREQ, SEQ\#, GLOBAL}_{\text{trap}}, \text{onion}_{\text{i}}, \text{PK}_{\text{i}}) \rangle$$

Eventually the destination receives the RREQ with the multi-layer onion. The destination broadcasts a RREP packet with the onion. Only the right upstream node that produced the outmost layer of the onion is able to decrypt it and mark itself en route. Thus when node N_i receives an RREP message that contains onion_i, it decrypts it and uses the inner layer of the onion, i.e., onion_{i-1}, to broadcast the RREP to the next upstream. Eventually the RREP traces the onion layers and is forwarded back to the source. The onion_{i-1} and the RREP message format are given below:

$$\text{onion}_{i-1} = E_{K_{i-1}}(E_{K_{i-2}}(\ldots(\text{core})\ldots)$$

$$\langle \text{RREP}, E_{\text{PK}_{i-1}}(S_i), E_{S_i}(\text{PROOF}_{\text{dest}}, \text{onion}_{i-1}) \rangle$$

where, key S_i a secret key that node N_i wants to share with the upstream node N_{i-1}. These two consecutive RREP forwarders N_{i-1} and N_i will produce one-time packet content in order to allow traffic mixing among neighborhood. $\text{PROOF}_{\text{dest}}$ is the cryptographic structure which shows that the destination successfully opened the global trapdoor $\text{GLOBAL}_{\text{trap}}$. After all, without revealing any identifiers, nodes are able to use the embedded encryption and decryption operations in the layered structure of the onion to establish themselves on the routing path for the data forwarding.

The session key S_i is also served as the route pseudonym N_i, or say, the identifier of the anonymous virtual circuit (anonymous circuit identifier (ACI)) for the link of N_{i-1} and N_i. Each node records the incoming route pseudonym together with the outgoing route pseudonym and inserts the pseudonym pair to the route table (ACI table). The anonymous virtual circuit is established when the source receives the RREP with route discovery session information confirmed.

In anonymous data forwarding, the route pseudonym N_i shared by the two ends of a link N_{i-1} and N_i is used in data packets transmitted by node N_{i-1}:

$$\langle \text{DATA}, N_i, E_{S_i}(\text{payload}) \rangle$$

Nodes hearing the packet must look up the route pseudonym N_i in their ACI tables. A node discards the packet if the route pseudonym in the packet does not match any incoming ACI in its table. Otherwise, it changes the packet's route pseudonym field to the matched outgoing ACI, then acts as the current forwarder and local broadcast the modified packet. The procedure is then repeated until the data packet arrives at the destination.

ANODR also suggests more sophisticated design options that uses S_i as the secret seed to generate cryptographically strong pseudorandom sequences and use the ith in the sequence as the route pseudonym ACI for the ith data packet. The ACI table updates itself for each sequence items. Such design ensures stronger untraceability. In the presence of network intruders, ANODR can limit the information leaking to only intruded nodes.

7. Conclusion

In this paper, we discussed an important issue in wireless network, user anonymity. Untraceability, though a separate concept, is closely knitted with anonymity. To preserve anonymity and untraceability, proper authentication protocols have to be designed with common constraints of wireless networks, i.e., low computational power at mobile terminals, lower network bandwidth, and high channel error rate. Many wireless authentication protocols have been proposed using either secret-key cryptography, public-key cryptography, or both. Most of the protocols only preserve anonymity of wireless users from eavesdroppers and visiting networks. Their untraceability level often reaches level 1, i.e., the activity of the mobile users are only untraceable to the eavesdroppers.

In wireless *ad hoc* networks, many nodes play roles of routers. The routing information can release locations of wireless nodes and their relationship, from which, an identity or the location of a wireless node can be compromised, even if a proper authentication protocol is in place. Some *ad hoc* network routing protocols have been proposed to limit the network topology knowledge of routing nodes. Nevertheless, the anonymity and the untraceability of wireless nodes are improved.

We have surveyed a few authentication protocols and *ad hoc* routing protocols, preserving anonymity and untraceability of wireless users. However, the design of such protocols is very challenging. Through our survey, we often see the attack-fix-attack cycle during the design. In other words, the conquest

DOI: 10.1002/sec

for searching good wireless authentication protocols preserving user anonymity and untraceability is still on.

Acknowledgements

This work was supported in part by the National Science Foundation (NSF) under grants CNS-0716211, CNS-0737325, and CCF-0829827.

References

1. Asokan N. Anonymity in a mobile computing environment. *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications.* 1994.
2. Danezis G, Diaz C. A Survey of anonymous communication channels. *Microsoft Research Technical Report MSR-TR-2008-35*, January 2008.
3. McCoy D, Bauer K, Grunwald D, Kohno T, Sicker D. Shining light in dark places: understanding the Tor network. *Proceedings of the 8th International Symposium on Privacy Enhancing Technologies (PETS 2008)*, Leuven, Belgium, July 2008; 63–76.
4. Fu X, Zhu Y, Graham B, Bettati R, Zhao W. On flow marking attacks in wireless anonymous communication networks. *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2005; 493–503.
5. Jiang Y, Lin C, Shen X, Shi M. Mutual authentication and key exchange protocols for roaming services in wireless mobile networks. *IEEE Transactions on Wireless Communications* 2006; **5**(9): 2569–2577.
6. Samfat D, Molva R, Asokan N. Untraceability in mobile networks. *Proceedings of the 1st Annual International Conference on Mobile Computing and Networking (MobiCom'95)*, November 1995; 26–36.
7. Park C-S. Authentication protocol providing user anonymity and untraceability in wireless mobile communication systems. *Computer Networks* 2004; **44**(2): 267–273.
8. Gedik B, Liu L. Location privacy in mobile systems: a personalized anonymization model. *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, Columbus, Ohio, USA, June 2005; 620–629.
9. Schilit B, Hong J, Gruteser M. Wireless location privacy protection. *Computer* 2003; **36**(12): 135–137.
10. Yampolskiy RV, Govindaraju V. Behavioural biometrics: a survey and classification. *International Journal of Biometrics* 2008; **1**(1): 81–113.
11. Pang R, Allman M, Paxson V, Lee J. The devil and packet trace anonymization. *ACM SIGCOM Computer Communication Review* 2006; **36**(1): 29–38.
12. Murdoch SJ. Hot or not: revealing hidden services by their clock skew. *Proceedings of 13th ACM Conference on Computer and Communications Security*, October 2006; 27–36.
13. Pang J, Greenstein B, Gummadi R, Seshan S, Wetherall D. 802.11 user fingerprinting. *Proceedings of 13th ACM International Conference on Mobile Computing and Networking*, Montreal, Quebec, Canada, 2007; 99–110.
14. Shmatikov V, Wang M. Measuring relationship anonymity in mix networks. *Proceedings of the Workshop on Privacy in the Electronic Society*, October 2006; 59–62.
15. Ozturk C, Zhang Y, Trappe W. Source-location privacy in energy-constrained sensor network routing. *Proceedings of 2nd ACM workshop on Security of Ad Hoc and Sensor Networks*, 2004; 88–93.
16. Mao W. Modern Cryptography: Theory and Practice. Prentice-Hall: Upper Saddle River, New Jersey, 2004.
17. Clark J, Jacob J. A survey of authentication protocol literature: version 1.0. November 1997; Available online at: http://www.cs.york.ac.uk/jac/papers/drareview.ps.gz or http://www.csl.sri.com/users/millen/capsl/library.html
18. Girault M. Self-certified public keys. In Advance in Cryptology--Proceeding of Eurocrypt' 91, Lecture Notes in computer Science, Vol. 547, No. 3 Davies DW. (ed.). Springer-Verlag: Berlin, Germany, 1991; 491–497.
19. Lee C-C, Hwang M-S, Liao I-E. Security enhancement on a new authentication scheme with anonymity for wireless environments. *IEEE Transactions on Industrial Electronics* 2006; **53**(5): 1683–1687.
20. Zhu J, Ma J. A new authentication scheme with anonymity for wireless environments. *IEEE Transactions on Consumer Electronics* 2004; **50**(1): 231–235.
21. Dominguez AP. Cryptanalysis of Park's authentication protocol in wireless mobile communication systems. *International Journal of Network Security* 2006; **3**(3): 279–282.
22. Go J, Kim K. Wireless authentication protocols preserving user anonymity. *Proceedings of the 2001 Symposium on Cryptography and Information Security*, Oiso, Japan, Vol. 1, 23–26 January 2001; 159–164.
23. Wan Z, Bao F, Deng RH, Ananda AL. Security analysis on a conference scheme for mobile communications. *IEEE Transactions on Wireless Communications* 2006; **5**(6): 1238–1240.
24. Yi X, Siew K, Tan CH. A secure and efficient conference scheme for mobile communications. *IEEE Transactions on Vehicular Technology* 2003; **52**(4): 784–793.
25. Needham RM, Schroeder MD. Authentication revisited. *Operating Systems Review* 1987; **21**(7): 7.
26. Boukerche A, Khatib El, Xu K, Lorba L. A novel solution for achieving anonymity in wireless *ad hoc* routing protocol. *Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks table of contents*, October 2004; 30–38.
27. Kong J, Hong X, Gerla M. An identity-free and on demand routing scheme against anonymity threats in mobile ad-hoc networks. *IEEE Transaction on Mobile Computing* 2007; **6**(8): 888–902.
28. Hong X, Kong J, Gerla M. Mobility changes anonymity: new passive threats in mobile ad hoc networks. *Wireless Communications & Mobile Computing (WCMC)* 2006; **6**(3): .281–293.
29. ITU-T. Rec. X.509 (revised) the Directory–Authentication Framework, International Telecommunication Union, Geneva, Switzerland, (equivalent to ISO/IEC 9594-8:1995), 1993.
30. Berlekamp MJ, McEliece RJ, van Tilborg HCA. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory* 1978; **24**(3): 384–386.
31. Saeednia S, Safavi-Naini R. A New identity-based key exchange protocol minimizing computation and communication. *Proceedings of the Information Security Workshop (ISW' 97), LNCS*, 1997; 328–334.

DOI: 10.1002/sec