Available online at www.sciencedirect.com
ScienceDirect
Computer Standards & Interfaces 29 (2007) 647–652
COMPUTER STANDARDS & INTERFACES
www.elsevier.com/locate/csi

An enhanced authenticated key agreement protocol for wireless mobile communication

Rongxing Lu [a,*], Zhenfu Cao [a], Haojin Zhu [b]

[a] Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, Peoples Republic of China
[b] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Received 6 November 2005; accepted 1 April 2007
Available online 19 April 2007

Abstract

With the rapid progress of wireless mobile communication, the authenticated key agreement protocol has attracted an increasing amount of attention. However, due to the limitations of bandwidth and storage of the mobile devices, most of the existing authenticated key agreement protocols are not suitable for wireless mobile communication. Quite recently, Sui et al. have presented an efficient authenticated key agreement protocol based on elliptic curves cryptography and included their protocol in 3GPP2 specifications to improve the security of A-Key distribution. However, in this paper, we show that Sui et al.'s protocol can't resist the off-line password guessing attack, and therefore present an enhanced authenticated key agreement protocol. At the same time, we also consider including our enhanced protocol in 3GPP2 specifications.

Keywords: Authenticated key agreement; Off-line password guessing attack; Wireless mobile communication; 3GPP2

1. Introduction

Over the last few years, fueled by the explosive growth of the wireless mobile communication technologies, many full-functional mobile devices have provided many new opportunities for a broad range of our communities. Based upon the current situation, the further widespread deployment of wireless mobile networks will depend on whether secure networking can be achieved. However, as we know, many existing protocols for wireless mobile networks didn't pay much attention to the fundamental issues of security: privacy and authentication [14]. The purpose of privacy is to protect sensitive messages against eavesdropping and modification, while authentication is to prevent unauthorized and forgery in wireless mobile services [7]. Therefore, in order to achieve these two security goals simultaneously, the authenticated key agreement (AKA) protocol has been introduced [1,2].

An authenticated key agreement protocol is an essential primitive in secure communication, by which a pair of users that communicate over a public unreliable channel can authenticate each other and generate a secure session key to guarantee the later communications' privacy and data integrity. Up to now, many excellent AKA protocols have been put forth. However, due to the mobile devices typical with constraints on available power consumption, bandwidth and storage limitation, most of these existing AKA protocols [1–4] are not suitable for the wireless mobile communications.

Quite recently, based on the simple authenticated key agreement algorithm (SAKA) due to Seo and Sweeney [16], Sui et al. [15] have developed an improved elliptic curve authenticated key agreement (ECAKA) protocol for wireless mobile communication, which not only eliminates the disadvantages of SAKA, but provides identity authentication, key validation and perfect forward secrecy. At the same time, since it is based on the elliptic curve cryptography (ECC) [10,12], Sui et al.'s protocol [15] becomes more efficient[1] in

* Corresponding author.
E-mail addresses: rxlu@cs.sjtu.edu.cn (R. Lu), cao-zf@cs.sjtu.edu.cn (Z. Cao), h9zhu@bbcr.uwaterloo.ca (H. Zhu).

[1] For example, to achieve the same security strength, when using the point compression technique, ECC keys of about 160 bits are about equivalent to RSA or DSA keys of about 1024 bits.

0920-5489/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2007.04.002

Alice (A)		Bob (B)
$d_A \in [1, n-1]$		$d_B \in [1, n-1]$
$Q_A = (d_A + t)P$	$\xrightarrow{Q_A}$	
		$Q_B = (d_B - t)P$
	$\xleftarrow{Q_B, td_AP}$	$Y = Q_A - tP,\ td_AP$
$X = Q_B + tP,\ td_BP$	$\xrightarrow{td_BP}$	
$K_A = d_AX = d_Ad_BP$		$K_B = d_BY = d_Ad_BP$

Fig. 1. Sui et al.'s ECAKA protocol.

terms of computation and storage overhead than other AKA protocols [1–4]. Therefore, Sui et al.'s protocol is very suitable for wireless mobile communications. In [15], Sui et al. have included their protocol in the current 3GPP2 (3rd Generation Partnership Project 2) specifications [8,9] for OTASP (Over the Air Service Provisioning) to improve the security of A-Key (Authentication Key) distribution [8], since the current OTASP proposal is apt to the man-in-the-middle attack.

In this paper, however, we will show that Sui et al.'s protocol [15] is not secure as they claimed. Like some other password-based AKA protocols [5,11], Sui et al.'s protocol also can't resist the off-line password guessing attack. Therefore, to remedy the security loopholes existing in Sui et al.'s protocol, we present an enhanced authenticated key agreement protocol for wireless mobile communications and also include our enhanced protocol in 3GPP2 specifications to improve the security of A-Key distribution protocol.

The rest of this paper is organized as follows. In Section 2, some notations used throughout this paper are first introduced. Then, we review Sui et al.'s ECAKA protocol in Section 3, and point out the off-line password guessing attack exists in Sui et al.'s protocol in Section 4. Later, we present our enhanced ECAKA protocol and make some analysis and comparisons in Sections 5 and 6, respectively. And then we also consider including our enhanced protocol in 3GPP2 to improve the A-Key distribution protocol in Section 7. Finally, we draw our conclusions in Section 8.

2. Notations

Before the review of Sui et al.'s ECAKA protocol [15], we first introduce common notations used throughout this paper as follows.

- Alice (A), Bob (B): two communication users;
- $\mathbb{E}$: an elliptic curve defined over a finite field $\mathbb{F}q$ with large group order;
- n: a secure large prime;
- P: a point in $\mathbb{E}$ with large order n;
- $\mathcal{D}$: a uniformly distributed dictionary of size $|\mathcal{D}|$;
- S: a low-entropy password shared between Alice and Bob, which is randomly chosen from $\mathcal{D}$;
- t: the value t is derived from the password S in a predetermined way, which is uniformly distributed in $\mathbb{Z}_n^*$;
- H: a secure one-way hash function [13].

3. Review of Sui et al.'s ECAKA protocol

In this section, we will review Sui et al.'s protocol [15], which, as shown in Fig. 1, is composed of four steps.

- Step 1: A first chooses a random number $d_A \in [1, n-1]$, computes and sends $Q_A = (d_A + t)\,P$ to B.
- Step 2: B also chooses a random number $d_B \in [1, n-1]$, computes $Q_B = (d_b - t)P$, $Y = Q_A - tP = d_AP$ and td_AP. Then, B sends $(Q_B,\ td_AP)$ to A and computes the session key $K_B = d_BY = d_Ad_BP$.
- Step 3: A computes and checks td_AP. If it is correct, A computes $X = Q_B + tP$ and td_BP. Then, A sends td_BP to B for checking. At the same time, A also computes the session key $K_A = d_AX = d_Ad_BP$.
- Step 4: B checks td_BP, and informs A the checking result.

Clearly, the completeness of Sui et al.'s protocol [15] can be proved by the procedure of the protocol. Since both td_AP and td_BP can be verified and thus the common session key d_Ad_BP can be computed only by A and B.

4. Off-line password attack on Sui et al.'s protocol

Off-line password guessing attack succeeds when there is information in communications, which can be used to verify the correctness of the guessed passwords. In [15], Sui et al. claimed that their protocol can resist the off-line password guessing attack. However, in this section, we will show that the off-line password guessing attack, not as they claimed, is still effective in Sui et al.'s protocol.

In Sui et al.'s protocol, since all transcripts are transmitted over an open network, a benign (passive) adversary, can easily obtain a valid information pair $(Q_A,\ td_AP)$ such that $Q_A = (d_A + t)\,P$ for some $t \in \mathbb{Z}_n^*$. On the other hand, since t is derived from the password $S \in \mathcal{D}$ in a predetermined way, the adversary can guess a password $S^\star$ from $\mathcal{D}$ and derive the corresponding $t^\star$, then verify it by checking $t^*\,(Q_A - t^*P) = td_AP$. If it does hold, the adversary has guessed the correct secret password $S^\star = S$. Otherwise, the adversary repeatedly guesses a new password $S^\star$ from $\mathcal{D}$ until $t^*(Q_A - t^*P) = td_AP$ holds.

```
Off − linePasswordAttack − I(Q_A, td_AP, D)
  for i := 0 to |D|
    S* ← D;   t* ← S*   [ predetermined way]
    if t*(Q_A − t*P) = td_AP
    then return S*
```

Therefore, the above guessing attack is not a brute fore attack, and Sui et al.'s protocol is still vulnerable to such an off-line password guessing attack.

On the other hand, if the adversary is an active attacker, then Sui et al.'s protocol will suffer from another off-line password guessing attack. First, the adversary chooses a random number $d_A \in [1, n-1]$, and sends $Q_A = d_A P$ to B. Then, he will obtain a returned value $t(d_A P - tP)$ from B. With these, he can guess a password $S^\star$ from $\mathcal{D}$ and derive the corresponding $t^\star$, then verify it by checking $t^\star(d_A P - t^\star P) = t(d_A P - tP)$.If it holds, the adversary has guessed the correct secret password $S^\star = S$. Otherwise, the adversary repeatedly guesses a new password $S^\star$ from $\mathcal{D}$ until $t^\star(d_A P - t^\star P) = t(d_A P - tP)$ holds.

Off − line Password Attack − II($\mathcal{D}$)
  choose $d_A \in [1, n-1]$, send it to B
  receive the value $t(d_A P - tP)$
  for $i := 0$ to $|\mathcal{D}|$
    $S^\star \leftarrow \mathcal{D}$; $t^\star \leftarrow S^\star$ [predetermined way]
    if $t^\star(d_A P - t^\star P) = t(d_A P - tP)$
    then return $S^\star$

5. Our enhanced ECAKA protocol

To avoid the off-line password guessing attack, in this section, we present an enhanced ECAKA protocol. Our enhanced ECAKA protocol, as shown in Fig. 2, will also consist of four steps.

- Step 1: A first chooses a random number $d_A \in [1, n-1]$, computes $Q_{A1} = (d_A + t)P$, $Q_{A2} = d_A^2 \cdot P$ and then sends (Q_{A1}, Q_{A2}) to B.
- Step 2: B also chooses two random numbers $d_{B1}, d_{B2} \in [1, n-1]$, computes $Y = Q_A - tP = d_A P$, $Q_{B1} = d_{B1}P + d_{B2}Y$ and $Q_{B2} = d_{B1}Y + d_{B2}Q_{A2}$. Then, B sends $(H_B = H(A\|B\|Q_{A1}\|Q_{B1}\|Q_{B2}), Q_{B1})$ to A, where H denotes a secure one-way hash function and denotes the concatenation.
- Step 3: A computes $X = d_A Q_{B1} = d_{B1} d_A P + d_{B2} d_A^2 P$ and checks whether the equality $H(A\|B\|Q_{A1}\|Q_{B1}\|X) = H_B$ hold or not. If it does hold, A computes and sends $H_A = H(B\|A\|Q_{B1}\|Q_{A1}\|X)$ to B, and sets the session key $K_A = X$.
- Step 4: B checks the equality $H(B\|A\|Q_{B1}\|Q_{A1}\|Q_{B2}) = H_A$. If it does hold, B sets the session key $K_B = Q_{B2}$.

In the end, only A and B can obtain the common session key $K_A = K_B = d_{B1} d_A P + d_{B2} d_A^2 P$. Therefore, the correctness of our enhanced ECAKA protocol follows.

6. Security analysis and comparisons

In this section, we examine the security of our proposed enhanced ECAKA protocol in terms of the following security properties, replay attack, reflection attack, known-key security, perfect forward secrecy and off-line password guessing attack. First of all, we will review some security terms needed for security analysis.

Definition 1. A secure hash function, H: $x \to y$, is one-way, if given x, it is easy to compute $H(x) = y$; however, given y, it is hard to compute $H^{-1}(y) = x$.

Definition 2. The elliptic curve discrete logarithm problem is as follows: given a point $Q = xP$, where $0 \le x \le n-1$, it is hard to determine such an x.

Definition 3. The elliptic curve computational Diffie–Hellman problem is as follows: given random (P, aP, bP), for $0 \le a, b \le n-1$, it is hard to compute abP.

- Replay attack: In the replay attack, an intruder impersonates a legal user by replaying the user's transmitting contents. Here, the intruder may intercepts (Q_{A1}, Q_{A2}) from A and uses them to impersonate A. However, without knowing the password S, it is infeasible for the intruder to compute a correct H_A, due to the elliptic curve discrete logarithm problem. Similarly, the intruder also can't impersonate B, since A can detect such an attack by checking H_B. Therefore, our enhanced protocol is secure against replay attacks.

Alice (A)		Bob (B)
$d_A \in [1, n-1]$		$d_{B1}, d_{B2} \in [1, n-1]$
$Q_{A1} = (d_A + t)P$		
$Q_{A2} = d_A^2 \cdot P$	$\xrightarrow{Q_{A1}, Q_{A2}}$	
		$Y = Q_{A1} - tP = d_A P$
		$Q_{B1} = d_{B1}P + d_{B2}Y$
	$\xleftarrow{H_B = H(A\|B\|Q_{A1}\|Q_{B1}\|Q_{B2}), Q_{B1}}$	$Q_{B2} = d_{B1}Y + d_{B2}Q_{A2}$
$X = d_A Q_{B1}$		
$H(A\|B\|Q_{A1}\|Q_{B1}\|X) \stackrel{?}{=} H_B$		
$K_A = X$	$\xrightarrow{H_A = H(B\|A\|Q_{B1}\|Q_{A1}\|X)}$	
		$H(B\|A\|Q_{B1}\|Q_{A1}\|Q_{B2}) \stackrel{?}{=} H_A$
		$K_B = Q_{B2}$
$K_A = K_B = X = Q_{B_2} = d_{B1} d_A P + d_{B2} d_A^2 P$		

Fig. 2. Our enhanced ECAKA protocol.

Table 1
Comparisons of Sui et al.'s protocol and our enhanced protocol

	Sui et al.'s protocol		Our enhanced protocol	
	A	*B*	*A*	*B*
Scalar multiplication	5	5	3	5
Point addition	1	1	–	3
Hash operation	–	–	2	2
Key length		160-bit		160-bit
Replay attack		Yes		Yes
Reflection attack		Yes		Yes
Known-key security		Yes		Yes
Perfect forward secrecy		Yes		Yes
Off-line password attack		Yes		No

- Reflection attack: In the reflection attack, an intruder attempts to impersonate the receiver by sending back message the same as the ones from the sender. However, in our enhanced protocol, by validating the hash values H_A and H_B, this attack can be detected. As a result, this attack cannot follow.
- Known-key security: A protocol is known-key security if it still achieves its goal in the face of an intruder who has learned some previous session keys. In view of the randomness of d_A, d_{B1}, d_{B2} in our enhanced protocol, session keys in different key agreement are independent of each other, and then the knowledge of previous session keys does not help an intruder to derive any future session key. Hence, our enhanced protocol has the property of known-key security.
- Perfect forward secrecy: The compromise of a session key $K_A=K_B=d_{B1}d_AP+d_{B2}d_A^2P$ requires the knowledge of the corresponding short-term secret keys d_A, d_{B1}, d_{B2} in our enhanced protocol. Suppose an intruder knows the password S and the corresponding t, when he wants to learn the previous session keys, he still faces the elliptic curve computational Diffee–Hellman problem. Therefore, the property of perfect forward secrecy is satisfied.
- Off-line password guessing attack: Different from Sui et al.'s protocol [15], our enhanced protocol can resist the off-line password guessing attack.
- If an intruder is benign, all he receives by wiretapping are Q_{A1}, Q_{A2}, Q_{B1}, H_A and H_B. Since the one-wayness of the hash function, the relation for helping guess the password in Section 4 is not available to the intruder. Therefore, it is hard to guess the correct password. On the other hand, even though in an extreme case that the intruder knows the corresponding session key $d_{B1}d_AP+d_{B2}d_A^2P$, from $Q_{A1}=(d_A+t)P$, $Q_{A2}=d_A^2\cdot P$ and $Q_{B1}=d_{B1}P+d_{B2}d_AP$, it is still hard to guess the password, due to the hardness of elliptic curve computational Diffie–Hellman problem.
- If an intruder is malicious, he may choose a random number d_A in $[1,n-1]$ and sends $Q_{A1}=d_AP$, $Q_{A2}=d_A^2P$ to B, then he will receive $Q_{B1}=d_{B1}P+d_{B2}(d_A-t)P$ and the hash value H_B from B according to the protocol. However, due to the randomness and independence of d_{B1} and d_{B2}, when the intruder guess a password $S^\star$, he can derive the corresponding $t^\star$ in a predetermined way, but can't validate it. Therefore, the off-line password guessing attack doesn't follow.

In the end, we will use Table 1 to compare our enhanced ECAKA protocol with Sui et al.'s protocol [15].

In Sui et al.'s protocol, Alice (A) and Bob (B) both have to execute 5 scalar multiplication and 1 point addition operations. While in our enhanced protocol, 3 scalar multiplication, 2 hash operations are needed for Alice (A) and 5 scalar multiplication, 3 point addition and 2 hash operations are required for Bob (B). On the other hand, to achieve the enough security strength, 160-bit key is assumed to be used in both protocols. In addition,

	MS ⟷ MSC ⟷ OTAF ⟷ HLR ⟷ AC	
Step 1:	OTAF ⟶ HLR:	OTASPREQ[ACTCODE,AKEYPV]
Step 2:	HLR ⟶ AC:	OTASPREQ[ACTCODE,AKEYPV]
	AC:	compute Q_{A1}, Q_{A2}
Step 3:	AC ⟶ HLR:	OTASPREQ[AKEYPV, P, E, n, Q_{A1}, Q_{A2}]
Step 4:	HLR ⟶ OTAF:	OTASPREQ[AKEYPV, P, E, n, Q_{A1}, Q_{A2}]
Step 5:	OTAF ⟶ MSC:	SMDPP[SMS_BearerData(AKEYPV, P, E, n, Q_{A1}, Q_{A2}), SRVIND]
Step 6:	MSC ⟶ MS:	AKEYPV, P, E, n, Q_{A1}, Q_{A2}
	MS:	compute Q_{B1}, Q_{B2}, H_B, A-KEY = Q_{B2}
Step 7:	MS ⟶ MSC:	Q_{B1}, H_B
Step 8:	MSC ⟶ OTAF:	SMDPP[SMS_BearerData(Q_{B1}, H_B)]
Step 9:	OTAF ⟶ HLR:	OTASPREQ[Q_{B1}, H_B, ACTCODE]
Step 10:	HLR ⟶ AC:	OTASPREQ[Q_{B1}, H_B, ACTCODE]
	AC:	compute X, H_A, A-KEY = X
Step 11:	AC ⟶ HLR:	OTASPREQ[H_A]
Step 12:	HLR ⟶ OTAF:	OTASPREQ[H_A]
Step 13:	OTAF ⟶ MSC:	SMDPP[SMS_BearerData(H_A), SRVIND]
Step 14:	MSC ⟶ MS:	H_A
Step 15:	MS ⟶ MSC:	ACK
Step 16:	MSC ⟶ OTAF:	SMDPP[SMS_BearerData(ACK)]

Fig. 3. A-Key distribution using our enhanced ECAKA protocol.

from Table 1, we can also see that our enhance protocol can resist the off-line password guessing attack. Therefore, our enhance protocol is more secure than Sui et al.'s protocol and suitable for wireless mobile communication.

7. Application to 3GPP2 mobile networks

In 3GPP2 mobile networks, A-Key is the master key of 3GPP2 networks and is usually used to generate the sub keys for voice, data and signaling privacy [8]. In current 3GPP2 specifications, to exchange the A-Key, the OTASP [8,9] proposal is using the basic Diffie–Hellman key exchange protocol [6] between the mobile user and its home network. However, since the basic Diffie–Hellman key exchange doesn't provide the authentication, this method easily suffers from the man-in-the-middle attack.

Therefore, in [15], Sui et al. suggested including their protocol in the current 3GPP2 specifications to improve the A-Key distribution. Clearly, A-Key distribution using Sui et al.'s protocol can largely enhance the security. However, since Sui et al.'s protocol can't resist the off-line password guessing attack in essence, the security of A-Key distribution using Sui et al.'s protocol still can't be well guaranteed.

Fortunately, our enhanced protocol avoids the security loopholes in Sui et al.'s protocol [15]. Therefore, we can include our enhanced ECAKA protocol in 3GPP2 specifications [8,9] to thoroughly improve the security of A-Key distribution. In the following, we first introduce some used symbols in 3GPP2 specifications, then use Fig. 3 to show our improved A-Key protocol. Here we should note that, when we include our enhanced protocol in 3GPP2 specifications, a short password is assumed to be shared by the mobile subscriber and the authentication center of its home network in advance.

- MS: Mobile Subscriber;
- MSC: Mobile Switching center;
- OTAF: Over-the-Air Service Provisioning Function;
- HLR: Home Location Register;
- AC: Authentication Center;
- ACTCODE: ActionCode;
- AKEYPV: A Key Protocol Version parameter;
- SRVIND: ServiceIndicator parameter;
- OTASPREQ: OTASPRequest;
- SMDPP; SMSDeliveryPointToPoint;
- SMS BearerData: Containing an OTASP data message;
- ACK: Acknowledging a message;

Since our enhanced protocol can resist the off-line password guessing attack, it is obvious to see that the A-Key distribution using our enhanced ECAKA protocol is more secure than that using Sui et al.'s protocol [15].

8. Conclusions

In this paper, we first reviewed Sui et al.'s ECAKA protocol [15], and pointed out that their protocol can't resist the off-line password guessing attack. Then, to remedy the security loopholes existing in Sui et al.'s protocol, we presented an enhanced ECAKA protocol, analyzed its security and compared it with Sui et al.'s protocol. Our result shows that our enhanced protocol not only keeps the advantages of Sui et al.'s protocol, but also improves the insecurity of the latter. In the end, we also considered including our enhanced protocol in 3GPP2 specifications [8,9] to improve the security of A-Key distribution protocol.

Acknowledgments

The authors would like to thank the anonymous referees for their suggestions which significantly improved this paper. This work was supported by the National Natural Science Foundation of China under Grant Nos. 60572155 and 60673079, the National High Technology Development Program of China under Grant No. 2006AA01Z424, and the National Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20060248008.

References

[1] S. Bellovin, M. Merritt, Encrypted Key Exchange: Password-based Protocols Secure Against Dictionary Attacks, Proc. of the Symposium on Security and Privacy, 1992, pp. 72–84.
[2] S. Bellovin, M. Merritt, Augmented Encrypted Key Exchange: Password-based Protocols Secure Against Dictionary Attacks and Password File Compromise, Proc. of the 1st CCS, ACM Press, 1993, pp. 244–250.
[3] V. Boyko, P. MacKenzie, S. Patel, Provably secure password authenticated key exchange using Diffie–Hellman, Advances in Cryptology Eurocrypt '00, Lecture Notes in Computer Science, vol. 1807, Springer, Berlin, 1985, pp. 156–171.
[4] M. Bellare, D. Pointcheval, P. Rogaway, Authenticated key exchange secure against dictionary attacks, Advances in Cryptology Eurocrypt '00, Lecture Notes in Computer Science, vol. 1807, Springer, Berlin, 1985, pp. 139–155.
[5] K.-K.R. Choo, C. Boyd, Y. Hitchcock, The importance of proofs of security for key establishment protocols: formal analysis of Jan-Chen, Yang-Shen-Shieh, Kim-Huh-Hwang-Lee, Lin-Sun-Hwang, and Yeh-Sun protocols, Computer Communications 29 (15) (2006) 2788–2797.
[6] W. Diffie, M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory IT-22 (1976) 644–654.
[7] R. Hwang, F. Su, A new efficient authentication protocol for mobile networks, Computer Standards and Interfaces 28 (2) (2005) 241–252.
[8] 3GPP2 N.S001 v1.0, OTASP and OTAPA, available at: http://www.3gpp2.org Jan. 1999.
[9] 3GPP2 C.S0016-B v1.0, Over-the-Air Service Provisioning of mobile stations in spread spectrum standards, available at: http://www.3gpp2.org Oct. 2002.
[10] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation 48 (177) (1987) 203–209.
[11] R. Lu, Z. Cao, Off-line password guessing attack on an efficient key agreement protocol for secure authentication, International Journal of Network Security 3 (1) (2006) 35–38.
[12] V.S. Miller, Use of elliptic curves in cryptography, Advances in Cryptology - Crypto 85, Lecture Notes in Computer Science, vol. 128, Springer, Berlin, 1985, pp. 417–426.
[13] B. Schneier, Applied Cryptography, 2nd ed., John Wiley, New York, 1996.
[14] S. Shieh, F. Ho, Y. Huang, An efficient authentication protocol for mobile networks, Journal of Information Science and Engineering 15 (1999) 505–520.
[15] A. Sui, L. Hui, S. Yiu, K. Chow, W. Tsang, C. Chong, K. Pun, H. Chan, An improved authenticated key agreement protocol with perfect forward secrecy for wireless mobile communication, IEEE Wireless and Communications and Networking Conference (WCNC 2005), 2005, pp. 2088–2093.
[16] D. Seo, P. Sweeney, Simple authenticated key agreement algorithm, Electronics Letters 35 (13) (1999) 1073–1074.

Rongxing Lu received the B.Sc. and M.Sc. degrees in computer science from the Tongji University, Shanghai, China, in 2000 and 2003, respectively. In 2006, he received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China. Currently, he is a Post-doctoral fellow at the University of Waterloo, Waterloo, Canada. His current research interests include wireless network security and cryptography. Now, he is also a guest member of Trusted Digital Technology Laboratory of Shanghai Jiao Tong University.

Zhenfu Cao received his B.S. degree in computer science and technology from Harbin Institute of Technology, China, in 1983, and his Ph.D. degree in mathematics from the same university. Currently, he is a professor and a doctoral supervisor at the Department of Computer Science and Engineering of Shanghai Jiao Tong University. His main research areas are number theory, modern cryptography, theory and technology of information security etc. Now, he is the director of Trusted Digital Technology Laboratory of Shanghai Jiao Tong University.

Haojin Zhu received the B.Sc. degree in computer science from Wuhan University, Wuhan, China, in 2002 and the M.Sc. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2005. He is currently working toward a Ph.D. degree in the electrical and computer engineering at the University of Waterloo, Waterloo, Canada. His current research interests include wireless network security and applied cryptography. Now, he is also a guest member of Trusted Digital Technology Laboratory of Shanghai Jiao Tong University.