International Journal of Innovative Computing, Information and Control
Volume **6**, Number **4**, April **2010**

pp. 1–10

A SECURE AND EFFICIENT ECC-BASED AKA PROTOCOL FOR WIRELESS MOBILE COMMUNICATIONS

Jung-Wen Lo[1,2], Cheng-Chi Lee[3], Min-Shiang Hwang[4] and Yen-Ping Chu[5]

[1]Department of Computer Science and Engineering
National Chung Hsing University
250 Kuo Kuang Road, Taichung, Taiwan 402, R.O.C.
[2]Department of Information Management
National Taichung Institute of Technology
129 Sec. 3, San-min Rd., Taichung, Taiwan 413, R.O.C.
asalo@ntit.edu.tw

[3]Department of Computer & Communication Engineering
Asia University
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung, Taiwan, R.O.C.
cclee@asia.edu.tw

[4]Department of Management Information Systems
National Chung Hsing University
250 Kuo Kuang Road, Taichung, Taiwan 402, R.O.C.
mshwang@nchu.edu.tw

[5]Department of Computer Science & Information Engineering
Tunghai University
181 Sec. 3, Taichung Harbor Rd., Taichung, Taiwan 407, R.O.C.
ypchu@thu.edu.tw

Received July 2009; revised December 2009

Abstract. *Considering the performance, encrypting the transmission message by the symmetric cryptosystem is the best choice if the session key generation problem is overcome. Diffie-Hellman key exchange protocol gives a new direction for making session key but is lack of the authentication property. The RSA-based public key cryptography makes the sufficient security for the key agreement but is not suitable for wireless mobile systems. In 2005, Sui et al. proposed an authenticated key agreement (AKA) protocol based on the elliptic curve cryptography (ECC) for wireless mobile communication. Lu et al. pointed out a secure defect in Sui et al.'s protocol and proposed an enhanced protocol. Later, Chang et al. stated that Lu's scheme is insecure and proposed an improved scheme in 2008. We found that Chang et al.'s scheme did not satisfy the mutual authentication, so a securer and more efficient protocol is proposed. The protocol can be applied not only in 3GPP2 specification but also in other wireless environments.*
Keywords: Authentication, Key agreement, Key exchange, Elliptic curve cryptography, Wireless communications.

1. **Introduction.** To avoid the leak of transmission messages, using the cryptosystem to encrypt the message is the best solution. As lots of malicious people snoop the network for the valuable data, therefore we should protect the transmission data over networks. Asymmetric cryptography provides a good protection but it is inefficient due to the computations of the exponentiation, especially for the mobile devices. On the contrary, the symmetric cryptosystem may be a good choice. However, it has the secret key management problem. We need a better solution for the mobile devices.

Diffie-Hellman proposed a concept of the key exchange in 1976 which allowed both participating parties to construct a common session key without leaking any session key information during transmission period

[4]Responsible for correspondence: Prof. Min-Shiang Hwang.

[1]. It is vulnerable for man-in-the-middle attack due to the lack of authentication [2-4]. To overcome this problem, lots of key exchange schemes were proposed [5-16]. Most of them are not suitable for mobile devices because of the heavy load of exponential computations. Neal Koblitz and Victor Miller individually proposed the public key cryptosystem based on elliptic curve cryptography (ECC) in 1985. The ECC can significantly reduce the computation and storage overhead. For example, ECC keys of 160 bits have almost the same level of security as RSA keys of 1024 bits.

Sui et al. proposed an improved authenticated key agreement protocol [17] based on the simple authenticated key agreement scheme [13] in 2005. Their scheme provided perfect forward secrecy but had an off-line password attack problem [18]. Lu et al. proposed an enhanced authenticated key agreement protocol for wireless mobile communication in 2007 [18]. Chang et al. pointed out that the Lu et al.'s scheme cannot resist against the parallel guessing attack in 2008 [19]. However, the Chang et al.'s scheme does not offer the mutual authentication property.

Both the electric power and computation ability in a wireless mobile device are very limited, therefore the desirable protocol for wireless mobile communication should bear lower computational load. We proposed a scheme which is securer and more efficient than above mentioned schemes and is suitable for the wireless environment.

The requirements for the key agreement protocol are shown as follows. Later, we will show that our protocol satisfies the all requirements.

Mutual authentication: The two communication parities can authenticate each other to make sure that they are communicating with the correct party.

Key verification: When a session key is set, both communication participants should verify the key.

Perfect forward secrecy: When one session key is compromised, the attacker cannot derive any previous session key.

Known-key security: The known-key security guarantees that the session key generated by every session is unique.

Key control security: The property of key control is to prevent either party from choosing the key value [20].

Session key security: The session key should be only known by both the communication participants.

The most contribution of the proposed protocol is that the protocol is suitable for the mobile environment and reaches all requirements of the key agreement protocol. Comparing with other protocols, the proposed protocol is the best choice for the mobile devices.

The remainder of this paper is organized as follows. The Sui et al.'s protocol, Lu et al.'s protocol and Chang et al.'s protocol are briefly introduced in Section 2. In Section 3, a detailed description of our protocol is presented. The discussions about the properties of key agreement and the security are stated in the Section 4 and the application to 3GPP2 is exhibited in Section 5. The last section is our conclusion.

2. Related Works.

In this section, we introduce the Sui et al.'s protocol [17], Lu et al.'s enhanced protocol [18] and Chang et al.'s improved protocol [19] separately. The notations used throughout this article are shown in Table 1.

2.1. Review of Sui et al.'s Protocol.

Herein, we briefly review the Sui et al.'s protocol which has four steps.

Step 1: A picks a random number $d_A \in [1, n-1]$ and sends $Q_A = (d_A + t)P$ to B.

Step 2: B picks a random number $d_B \in [1, n-1]$ and sends $\{Q_B = (d_B - t)P,\ tY = t(Q_A - tP)\}$ to A.

Step 3: If the received message tY equals to the td_AP, A sends the $tX = t(Q_B + tP)$ to B, and sets the session key $K_A = d_AX$.

Step 4: If the received message tX equals to the td_BP, B computes the session key $K_B = d_BY$.

Only A and B can obtain the session key $K_A = K_B = d_Ad_BP$. However, this scheme cannot resist against the off-line password guessing attack [17] and does not have the key verification property.

2.2. Review of Lu et al.'s Protocol.

To prevent the off-line password guessing attack, Lu et al.'s proposed an enhanced protocol based on Sui et al.'s protocol by using a one-way hash function. The steps of Lu et al.'s protocol are stated as follows.

Step 1: A picks a random number $d_A \in [1, n-1]$ and sends $\{Q_{A1} = (d_A + t)P,\ Q_{A2} = (d_A^2)P\}$ to B.

Table 1. The notations used in the article

Notations	Descriptions
A, B	The abbreviation/identity of the participator Alice (client) and Bob (server) individually
E	An elliptic curve defined over a finite field F_q with large group order
n	A large prime
P	A point in the elliptic curve with large order n
D	A uniformly distributed dictionary of size $\|D\|$
S	A low-entropy password shared between Alice and Bob, which is randomly chosen from D or chosen by the user
t	The value t is derived from the password S in a predetermined way, which is uniformly distributed in Z_n^*
H	A secure one-way hash function
$X \Rightarrow Y : \{Z\}$	The host X sends message Z to Y

Step 2: B picks two random numbers $d_{B1}, d_{B2} \in [1, n-1]$ and computes $Y = Q_{A1} - tP,\ Q_{B1} = d_{B1}P + d_{B2}Y,\ Q_{B2} = d_{B1}Y + d_{B2}Q_{A2}$. Then, B sends $\{H_B = H(A||B||Q_{A1}||Q_{B1}||Q_{B2}),\ Q_{B1}\}$ to A.

Step 3: A computes $X = d_A Q_{B1}$ and checks whether the equality of H_B and $H(A||B||Q_{A1}||Q_{B1}||X)$. If it holds, A sends $H_A = H(B||A||Q_{B1}||Q_{A1}||X)$ to B, and sets the session key $K_A = X$.

Step 4: If the received message H_A equals to the $H(B||A||Q_{B1}||Q_{A1}||Q_{B2})$, B sets the session key $K_B = Q_{B2}$.

Only A and B can have the session key $K_A = K_B = d_{B1}d_A P + d_{B2}d_A^2 P$.

2.3. **Review of Chang et al.'s Protocol.** Chang et al. proposed an improved protocol to avoid the parallel guessing attack. The steps are stated as follows.

Step 1: A picks a random number $d_A \in [1, n-1]$ and sends $\{Q_{A1} = (d_A + t)P,\ Q_{A2} = (d_A^2)P\}$ to B.

Step 2: B picks two random numbers $d_{B1}, d_{B2} \in [1, n-1]$ and computes $Y = Q_{A1} - tP,\ Q_{B1} = d_{B1}P + d_{B2}Y,\ Q_{B2} = d_{B1}Y + d_{B2}Q_{A2}$. Then, B sends $\{H_B = H(A||B||Q_{A1}||Q_{B1}||Y),\ Q_{B1}\}$ to A.

Step 3: A computes $X = d_A P$ and checks whether the equality of H_B and $H(A||B||Q_{A1}||Q_{B1}||X)$. If it holds, A sends $H_A = H(B||A||Q_{B1}||Q_{A1}||X)$ to B, and sets the session key $K_A = d_A Q_{B1}$.

Step 4: If the received message H_A equals to the $H(B||A||Q_{B1}||Q_{A1}||Q_{B2})$, B sets the session key $K_B = Q_{B2}$.

Only A and B can have the session key $K_A = d_A Q_{B1} = d_A(d_{B1}P + d_{B2}d_A P) = d_{B1}d_A P + d_{B2}d_A^2 P = K_B$. However, their scheme does not offer a mutual authentication property because B cannot authenticate A. We proposed a more efficient protocol with mutual authentication in Section 3.

3. **The Improved Protocol.** Our protocol is illustrated in Figure 1. The detail steps are described as follows.

Step 1: Alice (A) picks a random number $d_A \in [1, n-1]$ and computes $Q_A = (d_A + t)P$, where t is an integer which is predetermined by the corresponding password and P is a point in the elliptic curve. Then A sends out its identity A and Q_A to Bob (B).

Step 2: B also chooses a random number $d_B \in [1, n-1]$ and then computes $Q_B = (d_B - t)P,\ Y = Q_A - tP,\ K_B = d_B Y$ and $H_B = H(K_B||Y)$. Next, B sends the Q_B and H_B to A .

Step 3: A computes $X = Q_B + tP$ and then computes $K_A = d_A X$. Next, A verifies the equality of $H(K_A||d_A P)$ and H_B. If it does not hold, the protocol is terminated. Otherwise, A sends the $H_A = H(K_A||X)$ to B and sets the session key to K_A.

Step 4: When B receives the message, it checks the equality of $H(K_B||d_B P)$ and H_A. Only if the equality holds, B agrees the session key K_B. Otherwise, B terminates the protocol.

This scheme can resist against the off-line password guessing attack. The computational load is less than that of Lu et al.'s scheme because both participants only compute one Q_A/Q_B individually. Without any relations among the messages transmitted over the network, the parallel guessing attack will not succeed. Besides, the server side only needs to choose one random number d_B, so the used storage is also less than that of Lu et al.'s scheme and Chang et al.'s scheme. Definitely, our protocol is simpler and has a better security and performance for wireless environments.

Alice(A)		Bob(B)
Random d_A		Random d_B
$Q_A = (d_A + t)P$	$\xrightarrow{A, Q_A}$	$Q_B = (d_B - t)P$
		$Y = Q_A - tP$
	$\xleftarrow{Q_B, H_B}$	$K_B = d_B Y$
$X = Q_B + tP$		$H_B = H(K_B \| \| Y)$
$K_A = d_A X$		
$H(K_A \| \| d_A P) \stackrel{?}{=} H_B$		
$H_A = H(K_A \| \| X)$	$\xrightarrow{H_A}$	
		$H(K_B \| \| d_B P) \stackrel{?}{=} H_A$

FIGURE 1. Our Improved Protocol

4. **Discussion and Analysis.** In this section, we show that the protocol satisfies the requirements of the key agreement protocol and discuss the possible attack cases. First, we review some security terms for later using.

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

Definition 2. When given a value x for computing $H(x) = y$, it is hard to find a value z, $z \neq x$, which satisfies the equation $H(z) = H(x)$.

Definition 3. The elliptic curve discrete logarithm problem (ECDLP) defines as follows: Let E be an elliptic curve over a finite field F_q with a large group order. Suppose P is a point in E with large order n. Let Q be a point in E and $Q = xP$ where $0 \leq x < n$. It is hard to find out the exact x even if Q and P are given.

4.1. **Requirements of Key Agreement.** We shows that our protocol satisfies the requirements of a key agreement protocol, such as mutual authentication, known-key security, and perfect forward secrecy, etc.

Mutual authentication

The two parities can authenticate each other to make sure that they communicate with the right party. Bob obtains the $Y = d_A P$ from the message Q_A and replies the hash message $H(K_B||Y)$ in order to prevent the leak of Y. Alice authenticates Bob by the replied parameter Y which embeds in a random number d_A chosen by Alice previously. Bob also authenticate Alice by the message $H(K_A||X)$ where X has Bob's random number d_B.

Key verification

When a session key is set, it should be verified by both participants. In our protocol, Alice can verify the validity of session key K_B from the message $H(K_B||Y)$. She computes her session key K_A and checks the equality of $H(K_A||d_A P)$ and $H(K_B||Y)$. She accepts the key when the result is hold. Bob also can confirm the session key from the message $H(K_A||X)$ by using the same method as Alice does.

Perfect forward secrecy

When one session key is compromised, any attacker cannot derive any previous session key. Because the number d_A and d_B are randomly picked in every session, we can guarantee an unique session key for each session. All session keys are not relevant, so we do not have any perfect forward security problem.

Known-key security

The known-key security should guarantee that the session key produced by every session is unique. Because Alice and Bob choose random numbers, d_A and d_B, for every session individually, each session has an unique session key $d_A d_B P$.

Key control security

The property of key control is to prevent either party from choosing the key value [20]. In our protocol, the session key, $K_A = K_B = d_A d_B P$, is generated by both participants with the random numbers d_A and

d_B which are chosen by Alice and Bob separately. Therefore, no one can determinate the session key by itself.

Session key security

The session key should be only known by the communicated parties. In our protocol, the session key information is hidden in the hash function. An attacker only can eavesdrop Q_A, Q_B, $H(K_B||Y)$ and $H(K_A||X)$ from the message flows. According to the Definition 2, it is very hard to find out two different numbers which can have the same hash values, especially the session key concatenates with another values, X or Y. The attacker also can not derive the session key from the Q_A and Q_B owing to the ECDLP.

4.2. **Security Analysis.** In this subsection, we discuss the security of our protocol. The protocol can resist against the well-known attacks.

Password guessing attack

A password guessing attack occurs when an unauthorized user tries to guess the password. The unauthorized user can on-line login to a computer repeatedly or off-line to guess from the messages he eavesdrops from the network. We usually limit the number of incorrect login to avoid the on-line password guessing attack. In our scheme, the attacker only can eavesdrop Q_A, Q_B, $H(K_B||Y)$ and $H(K_A||X)$. The password is hidden in Q_A and Q_B, and it is well protected by ECDLP. Therefore, both the on-line and off-line password guessing attacks fail.

Man-in-the-middle attack

The man-in-the-middle attack is an active eavesdropping in which the attacker makes connections between the two entities and relays modified messages between them. The legal entities believe that they are communicating directly to each other, but the entire conversation is controlled by the attacker. In the proposed scheme, the authentication information is hidden in the message flows and is well protected by the ECDLP and the hash function. Without the common shared data t, the attacker can not play the man-in-the-middle attack successfully.

Unknown-key share attack

An entity A finishes the key agreement protocol with an entity B together and believes that the session key is shared with the entity B, but B believes that the key is shared with another entity who is not A [2]. This attack happens when entity A and entity B do not mutually authenticate. In our protocol, Alice authenticates Bob with the $H(K_B||Y)$ and Bob authenticates Alice with $H(K_A||X)$, therefore the attack is withstood.

Parallel guessing attack

When a server fails to deal with the requests carefully, an attacker can guess some useful data from the messages transmitted from server. It is happened in three-pass/four-pass protocols [21]. For example, an attacker obtains the messages $\{Q_{A1},\ Q_{A2},\ H_B,\ Q_{B1},\ H_A\}$ from an transaction in Lu et al.'s scheme. He guesses a X' and checks if the equality of H_B and $H(A||B||Q_{A1}||Q_{B1}||X')$ holds or not. If it holds, the attacker guesses the correct value of X. All transmitted messages are well designed in our protocol so the attack will fail.

Replay attack

The replay attack is an active attack which the valid data is transmitted repeatedly for some special purpose. The goal of the shared information is to authenticate each other. Only the server and client share the same common information, but others have no knowledge about it. If an attacker eavesdrops the communication and replays the old Q_A, server (Bob) will response a new $Q'_B = (d'_B - t)P$, $K'_B = d_A d'_B P$ and $Y' = Y = d_A P$. Next, the attacker needs to compute $X' = Q'_B + tP$ before having the session key $K'_A = d_A X'$. However, the K'_A cannot be obtained because the attacker can not derive the X' without the correct t. If the attacker replies the old $H(K_A||X)$, the server will detect it due to $H(d_A d'_B P||d'_B P) \neq H(d_A d_B P||d_B P)$. The replay attack therefore cannot succeed.

Impersonation attack

The attacker impersonates one of the communication entities to obtain the password or key. If the attacker pretends to be Alice(A), she guesses a t' and send the $Q'_A = (d'_A + t')P$ to Bob. Bob computes $Y = (d'_A + t' - t)P$ and $K_B = d'_A d_B P + d_B(t' - t)P$, and sends $\{Q_B = (d_B - t)P,\ H(K_B||Y)\}$ to the attacker. The attacker computes $X' = (d_B + t' - t)P$ and $K'_A = d'_A d_B P + d'_A(t' - t)P$. Then, the attacker checks the equality of $H(K'_A||d'_A P)$ and $H(K_B||Y)$. If it holds, the t' is the same as t, otherwise the attacker picks up another t' to try again. This attack is a little similar to the on-line password guessing, so it is easy to detect.

If the attacker pretends to be Bob(B), he only has incorrect $Q'_B = (d_B - t')P$, $Y' = Q_A - t'P$ and $K'_B = d_B Y'$ because he does not have any knowledge about t. Alice can easily detect it when she verifies the hash value $H(K'_B || Y')$ with $H(K_A || d_A P)$. Therefore, this attack (Reflection attack) is not working.

4.3. **Comparisons.** The comparisons of performance, property and security among the Sui et al.'s protocol[17], Lu et al.'s protocol[18], Chang et al.'s protocol[19] and our protocol are shown in Table 2.

TABLE 2. Comparisons of Sui et al.'s protocol[17], Lu et al.'s protocol[18], Chang et al.'s protocol[19] and our protocol

	Sui[17] A / B	Lu[18] A / B	Chang[19] A / B	Our protocol A / B
Scalar multiplication	4 / 4	3 / 5	3 / 5	3 / 3
Point addition	1 / 1	− / 3	− / 3	1 / 1
Hash operation	− / −	2 / 2	2 / 2	2 / 2
Key length	160-bit	160-bit	160-bit	160-bit
Mutual authentication	Yes	Yes	No	Yes
Key verification	No	Yes	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes	Yes
Known-key security	Yes	Yes	Yes	Yes
Key control security	Yes	Yes	Yes	Yes
Session key security	Yes	No	Yes	Yes
Off-line password attack	Yes	No	No	No
Man-in-the-middle attack	No	No	No	No
Unknown-key share attack	No	No	Yes	No
Parallel guessing attack	No	Yes	No	No
Replay attack	No	No	No	No
Impersonation attack	No	No	No	No

The comparisons of the performance are shown in the upper part of Table 2. In Sui et al.'s protocol, Alice (A) and Bob (B) both have to compute four scalar multiplications and one point addition operations. In Lu et al.'s and Chang et al.'s protocols, A executes three scalar multiplications, two hash operations, and B executes five scalar multiplications, three point additions and two hash operations. In our protocol, three scalar multiplications, one point addition and two hash operations are needed. The Lu et al.'s and Chang et al.'s protocols may result in the denial-of-service (DoS) attack because the computation load of the server is heavier than that of the client. The 160-bit key is used in three protocols in order to have enough security strength in an elliptic curve system.

The comparisons of requirements of key agreement are listed in the middle of Table 2. Sui et al.'s protocol is lack of the key verification property. Lu et al.'s protocol does not offer the property of session key security. Chang et al.'s protocol lacks of the mutual authentication ability.

The comparisons of security attacks are stated in the last part of Table 2. In addition, Sui et al.'s protocol is suffering from the off-line password guessing attack, Lu et al.'s protocol cannot resist against the parallel guessing attack and Chang et al.'s protocol cannot resist against the unknown-key share attack.

From the results of Table 2, it is obvious that our protocol is better in efficiency and satisfies all of the properties of key agreement as well as it can also resist against the off-line password guessing and parallel guessing attacks. Therefore, our protocol is more secure and suitable for wireless mobile communication.

5. **Application to 3GPP2 Mobile Networks.**

Herein, we use the 3GPP2 environment as the example to show that the proposed protocol can be applied in the wireless networks. The A-Key is the master key of 3GPP2 mobile networks and there are several proposed protocols for the key generation and distribution. The OTASP (Over the Air Service Provisioning) is the preferred protocol by 3GPP2 [22-23]. The protocol includes five participators: mobile subscriber (MS), mobile switching center (MSC), over-the-air service provisioning function (OTAF), home location register (HLR), authentication center (AC). The symbols used in the protocol are listed in Table 3.

The improved protocol can be easily implemented in 3GPP2 specifications. The protocol is stated in Table 4. We assume that a password is stored in the mobile unit when it is distributed to an MS. The

TABLE 3. The symbols used in the A-Key distribution protocol

Symbol	Meaning
ACTCODE	Action Code
AKEYPV	A Key Protocol Version parameter
SRVIND	Service Indicator parameter
OTASPREQ	OTASP Request
SMDPP	SMS Delivery Point To Point
SMS BearerData	Containing an OTASP data message
ACK	Acknowledging a message

TABLE 4. The improved A-Key distribution protocol

Steps	Participators	Transmission message or action
Step 1	OTAF⇒HLR:	$OTASPREQ[ACTCODE, AKEYPV]$
Step 2	HLR⇒AC:	$OTASPREQ[ACTCODE, AKEYPV]$
	AC:	$Compute\ Q_A$
Step 3	AC⇒HLR:	$OTASPREQ[AKEYPV, P, E, n, Q_A]$
Step 4	HLR⇒OTAF:	$OTASPREQ[AKEYPV, P, E, n, Q_A]$
Step 5	OTAF⇒MSC:	$SMDPP[SMS\_BearerDate(AKEYPV, P, E, n, Q_A), SRVIND]$
Step 6	MSC⇒MS:	$AKEYPV, P, E, n, Q_A$
	MS:	Compute Q_B, H_B and A-KEY
Step 7	MS⇒MSC:	Q_B, H_B
Step 8	MSC⇒OTAF:	$SMDPP[SMS\_BearerDate(Q_B, H_B)]$
Step 9	OTAF⇒HLR:	$OTASPREQ[Q_B, H_B, ACTCODE]$
Step 10	HLR⇒AC:	$OTASPREQ[Q_B, H_B, ACTCODE]$
	AC:	Compute X, H_A and A-KEY
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:	$SMDPP[SMD\_BearerData(ACK)]$

password can be chosen by the user and is only known by the MS as well as the AC of the home network. We also assume that MS has the ability to compute ECC point addition and point multiplication.

The main idea of the protocol runs as follows. The AC computes the $Q_A(= (d_A + t)P)$ in Step 2 and transmits Q_A to MS. When receiving Q_A in Step 6, MS computes $Q_B(= (d_B - t)P)$, $H_B(= H(K_B||Y))$ and set the A-Key $(= d_BY)$ where d_B is a random number and $Y = Q_A - tP$. In Step 10, AC computes $X(= Q_B + tP)$ and verifies the validity of the H_B. When the verification is successful, AC computes the A-Key $(= d_AX)$ and $H_A(= H(K_A||X))$. Later, MS can verify the validity of the H_A in Step 14.

The original A-Key distribution protocol is based on the basic Diffie-Hellman key exchange mechanism and has the man-in-the-middle attack problem [17]. Sui et al.'s A-Key protocol is more secure than the original A-Key protocol in current 3GPP2, but still has an off-line password guessing attack problem. Lu et al.'s protocol overcomes the problem but makes the protocol more complex. Fortunately, our improved A-Key protocol is not only securer than Sui et al.'s protocol but also more efficient than Lu et al.'s protocol, therefore it is a better choice.

6. **Conclusion.** In a key agreement protocol, using a common password to authenticate each other is simple and costless, especially in a wireless environment. Comparing with other public key cryptography, using the elliptic curve cryptography can reduce the computational overhead. We have shown that our protocol satisfies all requirements of key agreement and can resist against the well-known attacks in Section 4. With the comparisons of Sui et al.'s protocol, Lu et al.'s protocol, Chang et al.'s protocol and ours, our protocol has better efficiency and enough security. Moreover, we have shown that our protocol can enhance the security of A-key distribution protocol for current mobile cellular systems.

Acknowledgement. This work is partially supported by the Taiwan Information Security Center (TWISC) and the National Science Council under the grants NSC95-2218-E-001-001, NSC95-2218-E-011-015 and NSC97-2218-E-468-010. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Whitfield Diffie and M. Hellman, "New directions in cryptology," *IEEE Transations on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[2] Whitfield Diffie, Paul C. van Oorschot, and Michael J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107–125, 1992.

[3] Min-Shiang Hwang, Jung-Wen Lo, and Chia-Hsin Liu, "Enhanced of key agreement protocols resistant to a denial-of-service attack," *Fundamenta Informaticae*, vol. 61, no. 4, pp. 389–398, 2004.

[4] Cheng-Chi Lee, Min-Shiang Hwang, and Li-Hua Li, "A new key authentication scheme based on discrete logarithms," *Applied Mathematics and Computation archive*, vol. 139, no. 2-3, pp. 343–349, 2003.

[5] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," pp. 72–84, Proceedings of 1992 IEEE Computer Society Conference on Research in Security and Privacy, 1992.

[6] Chin-Chen Chang, Jung-San Lee, and Jui-Yi Kuo, "Time-bound Based Authentication Scheme for Multi-server Architecture," *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 11, pp. 2987–2996, 2008.

[7] H. Chen, Q. Ding L. Ding, and X. Dong, "Experimental Study on Secure Communication of Different Scroll Chaotic Systems with Identical Structure," *ICIC Express Letters*, vol. 2, no. 2, pp. 201–206, 2008.

[8] Hui-Feng Huang and Wei-Chen Wei, "A new efficient and complete remote user authentication protocol with smart card," *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 11, pp. 2803–2808, 2008.

[9] D. P. Jablon, "Strong password-only authenticated key exchange," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 5, pp. 5–26, 1996.

[10] Wen-Shenq Juang and Jing-Lin Wu, "Efficient user authentication and key agreement with user privacy protection," *International Journal of Network Security*, vol. 7, no. 1, pp. 120–129, 2008.

[11] Eric Jui-Lin Lu and Min-Shiang Hwang, "An improvement of a simple authenticated key agreement algorithm," *Pakistan Journal of Applied Sciences*, vol. 2, no. 1, pp. 64–65, 2002.

[12] Philip MacKenzie, Sarvar Patel, and Ram Swaminathan, "Password-authenticated key exchange based on RSA," in *Proceedings of ASIACRYPT 2000*, Lecture Notes in Computer Science 1976, pp. 599–613, Japan, 2000.

[13] D. Seo and P. Sweeney, "Simple authenticated key agreement algorithm," *IEE Electronics Letters*, vol. 35, no. 13, pp. 1073–1074, 1999.

[14] Shengbao Wang and Zhenfu Cao and Haiyong Bao, "Efficient Certificateless Authentication and Key Agreement (CL-AK) for Grid Computing," *International Journal of Network Security*, vol. 7, no. 3, pp. 342–347, 2008.

[15] K. Yamada, K. Kimura, H. Yuki, and K. Yoshida, "The home network system by mutual complement of wireless and wired communications," *ICIC Express Letters*, vol. 2, no. 1, pp. 73–79, 2008.

[16] Liu Yongliang and Wen Gao and Hongxun Yao and Xinghua Yu, "Elliptic Curve Cryptography Based Wireless Authentication Protocol," *International Journal of Network Security*, vol. 5, no. 3, pp. 327–337, 2007.

[17] Ai-Fen Sui, Lucas C.K. Hui, S.M. Yiu, K.P. Chow, W.W. Tsang, C.F. Chong, K.H. Pun, and H.W. Chan, "An improved authenticated key agreement protocol with perfect forward secrecy for wireless mobile communication," *2005 IEEE Wireless Communications and Networking Conference*, vol. 4, pp. 2088–2093, 2005.

[18] Rongxing Lu, Zhenfu Cao, and Haojin Zhu, "An enhanced authenticated key agreement protocol for wireless mobile communication," *Computer Standards & Interfaces*, vol. 29, pp. 647–652, 2007.

[19] Chin-Chen Chang and Shih-Chang Chang, "An improved authentication key agreement protocol based on elliptic curve for wireless mobile networks," in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 1375–1378, 2008.

[20] C.J. Mitchell, M. Ward, and P. Wilson, "Key control in key agreement protocols," *Electronics Letters*, vol. 34, no. 10, pp. 980–981, 1998.

[21] Taekyoung Kwon, "Practical authenticated key agreement using passwords," Lecture Notes in Computer Science 3225, pp. 1–12, 2004.
[22] *3GPP2 N.S0011 v1.0, OTASP and OTAPA*. available at: http://www.3gpp2.org, 1999.
[23] *3GPP2 C.S0016-C v1.0, Over-the-Air Service Provisioning of Mobile Stations in Spread Spectrum Standards*. available at: http://www.3gpp2.org, 2004.