Bilinear-Pairing-Based Remote User Authentication Schemes Using Smart Cards

Al-Sakib Khan Pathan and Choong Seon Hong
Department of Computer Engineering, Kyung Hee University
1 Seocheon, Giheung, Yongin 446701, South Korea
spathan@networking.khu.ac.kr, cshong@khu.ac.kr

ABSTRACT

This paper presents a detailed review of remote user authentication schemes with smart cards based on bilinear pairings. The first scheme regarding this was proposed by Manik et al. in 2006, which had been modified and improved later in several ways. Here, we analyze all the related proposed schemes, point out their weaknesses and flaws, and note down the sequence of improvements. After reviewing all the relevant schemes, we propose our scheme which is designed in such a way that it could ensure all sorts of facilities needed for remote user authentication procedure and could resist all types of known attacks. We present detailed validation, security, and performance analysis to prove the efficiency of our scheme.

Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: Security and protection

General Terms

Algorithm, Security, Verification.

Keywords

Remote, Authentication, Bilinear, Bilateral, Smart Card

1. INTRODUCTION

Access control of remote user is a method where the remote server confirms the validity of the user before giving him any opportunity to communicate with the server. This sort of communication is mainly needed for e-commerce, e-transactions, e-banking, etc. With the rapid development of communication technologies and wide-spread use of distributed networking, remote user authentication has become a critical issue. In many cases, it is also needed to make sure that the right entity is communicating with the user and thus user needs to verify the legitimacy of the remote server. To facilitate such communications, several works have been done. However, the first scheme based on bilinear pairings was proposed by Manik et al. [1]. This scheme was later modified, extended, or improved by several other researchers. In 2001, bilinear pairings like Weil and Tate pairing defined on elliptic curves were proved and they can now be applied to cryptography. The bilinear pairings are effective methods to reduce the complexity of the discrete logarithm problem in a finite field [2], [3]. Pairing provides a good setting for the bilinear Diffie–Hellman problem [4] and has been used to design several cryptosystems. The benefit of a bilinear pairing cryptosystem is that it remains the same security level but reduces the computation cost. In this paper, we restrict our focus only on the remote user authentication schemes based on bilinear pairings.

Major Contributions of this paper are as follows:

1. A detailed review of the remote user authentication schemes with smart cards which are based on bilinear pairings.
2. Pointing out the weaknesses of the proposed schemes.
3. Our proposed scheme with robust security and user-friendliness.

The rest of the paper is organized as follows; Section II presents the basic terms and preliminaries, Section III presents a detailed review of the proposed schemes based on bilinear pairings, Section IV presents our proposed scheme, Section V contains the security and performance analysis of our scheme, and finally Section VI concludes the paper delineating the findings from this work.

2. BASIC TERMS AND PRELIMINARIES

2.1 Bilinear Pairings

Let G_1 be an additive cyclic group of prime order q and G_2 be the multiplicative cycle group of the same order. In reality, G_1 is thought to be a group of points on an elliptical curve over Z_q^*, and G_2 is a subgroup of the multiplicative group of a finite field $Z_{q^k}^*$ for some $k \in Z_q^*$. Let P be a generator of G_1. A bilinear pairing is a map $e: G_1 \times G_1 \rightarrow G_2$ having three properties:

(i) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$, for all $P, Q \in G_1$ and $a, b \in Z_q^*$.

ICUIMC'09, January 15–16, 2009, Suwon, S. Korea.

(ii) Non-degenerate: $\forall P$ where P is not a generator, there exists $Q \in G_1$ such that, $e(P,Q) \neq 1$.

(iii) Computable: There exists an efficient algorithm to compute $e(P,Q)$ for all $P,Q \in G_1$.

2.2 Other Mathematical Backgrounds

Discrete Logarithm Problem (DLP): Given two elements $P,Q \in G_1$ find an integer $a \in Z_q^*$, such that $Q = aP$ whenever such an integer exists.

Computational Diffie-Hellman Problem (CDHP): Given (P, aP, bP) for any $a,b \in Z_q^*$, compute abP.

Decisional Diffie-Hellman Problem (DDHP): Given (P, aP, bP, cP) for any $a,b,c \in Z_q^*$, decide whether $c = ab \bmod q$.

Gap Diffie-Hellman (GDH) group: G_1 is a *GDH group* if there exists an efficient polynomial time algorithm which solves the DDHP in G_1 and there is no probabilistic polynomial time algorithm which solves the CDHP in G_1 with non negligible probability of success.

Bilinear Diffie-Hellman Problem (BDHP): Given (P, aP, bP, cP) for any $a,b,c \in Z_q^*$, compute $e(P,P)^{abc}$.

3. REVIEW OF THE SCHEMES BASED ON BILINEAR PAIRINGS

3.1 Manik et al.'s Novel Scheme

In 2006, Manik et al. [1] (will be termed as *basic scheme* throughout the rest of the paper) proposed a remote user authentication scheme using the properties of bilinear pairings. In their scheme, like other remote authentication schemes, the user sends login request to a remote server and if the login request is valid, the server allows access to it. Their scheme does not support multiple logged in users at the same time and has the password changing facility for the registered users without any assistance of the remote server. Here, we first present their scheme. There are mainly four phases in Manik et al.'s scheme:

Setup Phase. Suppose G_1 is an addictive cyclic group of order prime q, and G_2 is a multiplicative cyclic group of the same order. Suppose P is a generator of G_1, $e: G_1 \times G_1 \to G_2$ is a bilinear mapping and $H: \{0,1\}^* \to G_1$ is a cryptographic hash function. The remote system/server (RS) selects a secret key *s* and computes the public key as $Pub_{RS} = sP$. Remote server publishes the system parameters (G_1, G_2, *e*, *q*, *P*, Pub_{RS}, *H*) and *s* is kept as a secret.

Registration Phase. When the user U_i wants to register with the remote server, the following operations are done. This phase is performed over a secure channel:

Step 1. U_i submits his identity ID_i and an arbitrarily chosen password PW_i to RS.

Step 2. After receiving the request, RS computes,

$$Reg_{ID_i} = s \cdot H(ID_i) + H(PW_i)$$

Step 3. RS issues the smart card with the parameters (ID_i, Reg_{ID_i}, H(.)) over a secure channel to user U_i.

Login Phase. For logging in to the server, the user needs to attach the smart card with the input device (terminal) and must key in his identity and password, ID_i and PW_i. If the identity ID_i is same as the one that is stored in the smart card, the smart card performs the following steps:

Step 1. Computes $DID_i = T \cdot Reg_{ID_i}$ where T is the user system's timestamp.

Step 2. Computes $V_i = T \cdot H(PW_i)$.

Step 3. Sends the login request (ID_i, DID_i, V_i, *T*) to the RS over a public channel.

Authentication Phase. After receiving the login request message from the user, the RS does the following operations:

Step 1. Verifies the validity of the time interval between T′ and T, where T′ is the timestamp of receiving the login message. If $(T' - T) \leq \Delta T$, then RS goes to step 2 else rejects the login request. Here, ΔT denotes the allowed time interval for transmission delay. Step 2. Checks the condition, $e(DID_i - V_i, P) = e(H(ID_i), Pub_{RS})^T$. If it holds, RS accepts the login request, otherwise rejects it.

This method works because,

$$\begin{aligned} e(DID_i - V_i, P) &= e(T \cdot Reg_{ID_i} - V_i, P) \\ &= e(T(s \cdot H(ID_i) + H(PW_i)) - T \cdot H(PW_i), P) \\ &= e(s \cdot H(ID_i), P)^T \\ &= e(H(ID_i), sP)^T \\ &= e(H(ID_i), Pub_{RS})^T \end{aligned}$$

Password Change Phase. Whenever the user U_i wants to change his password, he has to go through this phase. For this:

Step 1. User first attaches his smart card to the input device and keys in his ID_i and PW_i. If the ID_i is same as the one that is stored in the card, the other steps are allowed otherwise, the password changing phase terminates here.

Step 2. U_i submits a new password PW_i^*.

Step 3. The smart card computes,

$$Reg^{*}_{ID_i} = Reg_{ID_i} - H(PW_i) + H(PW_i^{*})$$
$$= s \cdot H(ID_i) + H(PW_i^{*})$$

Step 4. Now, the new password becomes PW_i^{*} and the card replaces previously stored Reg_{ID_i} by newly computed $Reg^{*}_{ID_i}$.

3.2 Chou et al.'s Attack and Modification

Chou et al. [5] analyzed the *basic scheme* and devised an impersonation attack against it. This attack works as follows:

A1. An attacker captures the legal login request message (ID_i, DID_i, V_i, T) sent from a legitimate user U_i to RS as it is sent over a public channel. Then, computes,

$$DID_i - V_i = T \cdot Reg_{ID_i} - T \cdot H(PW_i)$$
$$= T \cdot (s \cdot H(ID_i) + H(PW_i)) - T \cdot H(PW_i))$$
$$= T \cdot s \cdot H(ID_i)$$

A2. The attacker chooses a random timestamp T_r and computes $T_r \cdot H(PW_a)$, where PW_a is the attackers selected password which is not confirmed by RS.

A3. Then the attacker computes its own DID_a and V_a:

$$DID_a = T_r \cdot (DID_i - V_i) + T_r \cdot T \cdot H(PW_a)$$
$$= T_r \cdot T \cdot s \cdot H(ID_i) + T_r \cdot T \cdot H(PW_a)$$
$$V_a = T_r \cdot T \cdot H(PW_a)$$

Then, it computes $T_r \cdot T = T_a$

$$DID_a - V_a = T_a \cdot Reg_{ID_i} - T_a \cdot H(PW_i)$$
$$= T_a \cdot (s \cdot H(ID_i) + H(PW_i)) - T_a \cdot H(PW_i))$$
$$= T_a \cdot s \cdot H(ID_i)$$

Step 4. At a later time T_a, when the attacker wants to launch the attack, it can use a forged message as, (ID_i, DID_a, V_a, T_a). Thus an impersonation attack could be launched.

To avoid this weakness in the *basic scheme*, Chou et al. suggested a modification in the verification phase that instead of checking $e(DID_i - V_i, P) = e(H(ID_i), Pub_{RS})^T$, the condition $e(DID_i, P) = e(T \cdot s \cdot H(ID_i) + V_i, P)$ should be checked.

3.3 Thulasi et al.'s Attacks

Thulasi et al. [6] analyzed both the *basic scheme* and its improved version proposed by Chou et al. [5]. Eventually they found that even Chou et al.'s scheme is also vulnerable. For this, the attacker computes $DID_a = DID_i + a'$ and $V_a = V_i + a'$ where $a' \in G_1$ so that the modified checking condition could be passed as follows:

$$e(DID_a, P) = e(DID_i + a', P)$$
$$= e(DID_i, P)e(a', P)$$
$$= e(T \cdot s \cdot H(ID_i) + V_i, P)e(a', P)$$
$$= e(T \cdot s \cdot H(ID_i) + V_i + a', P)$$
$$= e(T \cdot s \cdot H(ID_i) + V_a, P)$$

The authors also showed another forgery attack on the *basic scheme* as:

A1. The attacker taps the login request message from a legitimate user and gets the parameters, (ID_i, DID_i, V_i, T). As $DID_i = T \cdot Reg_{ID_i}$, where $T \in Z_q^{*}$ and $V_i = T \cdot H(PW_i)$, the attacker can compute T^{-1}, Reg_{ID_i}, and $H(PW_i)$ as follows:

$Reg_{ID_i} = T^{-1} DID_i = T^{-1} \cdot T \cdot Reg_{ID_i}$ and
$H(PW_i) = T^{-1} V_i = T^{-1} \cdot T \cdot H(PW_i)$

A2. Now the attacker can form a valid login request with (ID_i, DID_a, V_a, T_a) for timestamp T_a where, $DID_a = T_a \cdot Reg_{ID_i}$ and $V_a = T_a \cdot H(PW_i)$.

Thulasi et al. also pointed out that there is a weakness in password changing method in the *basic scheme* as before changing the password, the old password is not checked in the method. So, any adversary having the smart card and knowing the identity can change the secret information Reg_{ID_i}.

In spite of showing the weaknesses and flaws of the two previous schemes, they did not suggest any solution.

3.4 Jeon et al's Improvement

Jeon et al. [7] noted further weakness of the *basic scheme* that it is vulnerable to offline guessing attack and it does not guarantee bilateral verification, that is the user's legitimacy is only checked while the server spoofing might happen to the *basic scheme*.

***Weakness of Jeon et al.'s Scheme*.** They also commented on Thulasi et al.'s [6] findings that, the password changing phase in the *basic scheme* does not have any problem with password changing. However, their comment is not correct. This is because; in the *basic scheme*, in the password changing phase, only the identity (ID_i) of the user is checked. There is no suggested step to verify the old password to allow the user to change the password. So, if an adversary knows the identity of the legitimate

user and somehow gets the smart card, he can change the password without legitimate user's consent and can cause trouble.

[7] proposed an improved scheme with mutual verification so that both the user and server can verify each other's legitimacy but based on their wrong assumption they kept the password changing phase weak as it was in the *basic scheme*. In fact, they did not mention how the old password is to be checked in their scheme to make sure that the user of the smart card is a legitimate entity and he wants to change his own password.

3.5 Oh et al.'s Improved Scheme

Oh et al. [8] came up with another improved scheme with bilateral verification. Unfortunately they did the same mistake as [7] for the password changing phase. The fact is, there is no stored password table in the RS and there is no other mechanism suggested in their scheme to confirm that the legitimate user is using the smart card to change his password. Similar to Jeon et al.'s scheme, any adversary having the smart card (stolen or in some other way) and knowing the identity of the legitimate user can change the password which can cause trouble at least for some time to the user. Even if the user gets back his smart card, his own password wouldn't work. Neither he can change his password any more unless he re-registers with the RS or informs the authority regarding his problem. So, the weakness still remains in Oh et al.'s scheme.

3.6 Jia et al.'s Scheme Using Bilinear Pairings and Elliptic Curve Cryptosystem (ECC)

Jia et al. [9] took the *basic scheme* as a base and proposed a security enhanced authentication scheme. Their scheme has some verification mechanisms before changing the password and the operational method is made more complex. However, bilateral verification is absent in their proposed scheme. Moreover, Vo and Kim [10] showed a weakness of their scheme.

3.7 Vo and Kim's Security Enhanced Authentication Scheme over Jia et al.'s Scheme

Very recently, Vo and Kim [10] proposed a security enhanced scheme based on Jia et al.'s scheme. They showed that [9] cannot withstand a tricky impersonation attack. According to Vo and Kim, the login request message (ID_i, C_1, C_2, T_1) could be tapped and an evil user can forge a login message calculating the necessary parameters and with a new timestamp.

To overcome the flaw in Jia et al. [9] scheme, they proposed an improved scheme. Unfortunately, their scheme also suffers from the absence of bilateral verification mechanism. Server message might be spoofed by an insider or attacker as there is no way that the user can verify the communicating entity in the other end (i.e., RS).

3.8 Yang et al.'s Password-Based Access Control Scheme

Based on bilinear mapping, Yang et al. [11] proposed a new access control scheme using smart cards. Unfortunately, their scheme also has the weakness in password changing phase as it is in [1], [7], [8]. The same problem remains because of not checking the validity of the old password before accepting the new password and changing the secret information in the smart card. Moreover, their scheme does not have bilateral verification which is a considerable weakness.

4. OUR IMPROVED SCHEME

After analyzing all the proposed schemes, their weaknesses, attacks, and improvements, we devise our solution to surmount all the known flaws. Our major design goals are: (a) Minimum processing and transmission requirement. (b) Robust password changing phase so that the smart card can verify the legitimacy of the user before changing its content and allowing the user to change the password. (c) Bilateral verification, where not only the server but also the user can verify each other's legitimacy. (d) Resistance against all the known attacks against the schemes based on bilinear pairings. (e) No storage table in the RS containing any particular secret information for a particular user.

Our proposed scheme has mainly three phases. The setup phase is kept same as Manik et al.'s scheme that is: G_1 is an addictive cyclic group of order prime q, and G_2 is a multiplicative cyclic group of the same order. Suppose P is a generator of G_1, $e: G_1 \times G_1 \rightarrow G_2$ is a bilinear mapping and $H: \{0,1\}^* \rightarrow G_1$ is a cryptographic hash function. The remote system/server (RS) selects a secret key s and computes the public key as $Pub_{RS} = sP$. Remote server publishes the system parameters (G_1, G_2, *e*, *q*, *P*, Pub_{RS}, *H*) and *s* is kept as a secret.

Registration Phase. This phase takes place over a secure channel.

Step 1. When user U_i wants to register with the RS, he chooses an identity ID_i and a password PW_i and submits those to the RS over the secure channel.

Step 2. Remote server (RS) computes $Reg_{ID_i} = s \cdot H(ID_i)$ and $V_i = H(PW_i)$.

Step 3. RS issues the smart card to the user by storing ($ID_i, Reg_{ID_i}, V_i, H(\cdot)$) in the card's memory and assigns that to the user U_i.

Login Phase. For logging in to the server, the user needs to attach the smart card with the input device and must key in his identity and password, ID_i and PW_i. If the identity ID_i is same as the one that is stored in the smart card, the smart card performs the following steps:

Step 1. It first verifies whether the user of this smart card is legitimate or not by verifying the condition, $V_i = H(PW_i)$. If it fails, further steps are terminated.

Step 2. Then the smart card computes, $DID_i = T \cdot Reg_{ID_i}$ and $T_{enc} = Enc_{Pub_{RS}}(T)$, where $Enc_{Pub_{RS}}$ is the public key

encryption and *T* is the timestamp.

Step 3. Then the login request message, M=(ID_i , DID_i , T_{enc}) is sent to the RS over public channel.

Mutual Authentication Phase. Our authentication phase has two sub-phases. First the server verifies the legitimacy of the user and then the user verifies the legitimacy of the communicating server. This is done so that the user can be sure that further communications and transactions are done with a valid server.

User Authentication Phase.

Step 1. On receiving the login request M=(ID_i , DID_i , T_{enc}), the RS gets the value of *T* by decrypting T_{enc} with its private key *s*. Then, checks the condition $(T' - T) \le \Delta T$, where T' is the timestamp of receiving the login request message and ΔT is the allowed time interval for transmission delay of the login request message. If this condition fails, the request is rejected by RS otherwise it proceeds to the next step.

Step 2. RS checks the condition, $e(DID_i, P) = e(H(ID_i), Pub_{RS})^T$. If it holds the RS becomes sure of the legitimacy of the user. Otherwise, the process is terminated and access is denied.

Step 3. RS computes $C_R = H(T'' \cdot s \cdot H(ID_i))$, where T'' is the timestamp of the server. Then it sends back M´=(C_R , T'') to the user side.

Server Authentication Phase

Step 1. On receiving the message M´=(C_R , T''), first the user checks the time interval condition, $(T''' - T'') \le \Delta T$, where T''' is the user's message receiving timestamp. If it holds, the second step is performed otherwise, the process is terminated.

Step 2. Checks whether the condition $C_R = H(T'' \cdot Reg_{ID_i})$ holds or not. If the condition holds, the user side is confirmed that the server is legitimate.

Password Changing Phase.

Step1. When the user wants to change his password, he attaches the smart card with the input device and keys in ID_i and PW_i .

Step 2. Smart card verifies the identity ID_i and checks whether $H(PW_i)$ is equal to the stored value V_i or not. If the old password is not correct, this operation fails and the user is rejected to change the password, else the next step is executed.

Step 3. If smart card is sure that the carrier of the smart card is a person who has both the legitimate identity and password (as made sure in step 2), it allows the user to enter the new password PW_i^* .

Step 4. Smart card computes, $V_i^* = H(PW_i^*)$ and replaces previously stored V_i by V_i^*

5. VALIDATION AND SECURITY ANALYSIS OF OUR SCHEME

5.1 Correctness

For user authentication phase in step 2, the verification is done because:

$$e(DID_i, P) = e(T \cdot Reg_{ID_i}, P)$$

$$= e(T \cdot s \cdot H(ID_i), P)$$

$$= e(T \cdot H(ID_i), sP) \text{ [as } e(bP, Q) = e(P, bQ) \text{]}$$

$$= e(T \cdot H(ID_i), Pub_{RS}) \text{ [as } Pub_{RS} = sP \text{]}$$

$$= e(H(ID_i), Pub_{RS})^T$$

[as $e(aP, Q) = e(P, Q)^a$, bilinearity of *e*]

5.2 Security Analysis

Replay Attack. An adversary is able to tap the login request message, M=(ID_i , DID_i , T_{enc}) from a valid user. But, the timestamp *T* is encrypted. Moreover, as *s* is the secret kept in the server, finding *T* from T_{enc} is a discrete logarithm problem (DLP). So, replay attack is in no way possible in our scheme. Even if the adversary somehow knows the timestamp of sending the message, step 1 in user authentication phase prevents the replay attack.

Forgery Attack. From the login request message, at best an adversary can know the values of ID_i , DID_i , and T_{enc} . But, no useful information can be derived from any of these. *T* cannot be known because of the DLP problem. Only one value $DID_i = T \cdot Reg_{ID_i}$ does not reveal any important information as Reg_{ID_i} is a secret value stored in the smart card and *T* cannot be gained from the message.

Server Spoofing Attack. In the server authentication phase, the server sends $C_R = H(T'' \cdot s \cdot H(ID_i))$ and timestamp T'' to the user. From these values, it's not possible to obtain any important information or the RS cannot be impersonated. One interesting point is, to reduce the number of processings, $C'_R = T'' \cdot s \cdot H(ID_i)$ could be sent and in the user side the checking condition could be made as $C'_R = T'' \cdot Reg_{ID_i}$, that is extra hashing can be omitted. Even in such case, RS cannot be impersonated as the attacker cannot find *s* even after knowing T'' , ID_i and H. This is because; finding *s* from these values means solving a DLP problem.

Insider Attack. There is no need of storing any verification table in the server side. Also storing the passwords is not needed. Moreover, *s* is a secret value in RS. So, insider attack which in many password based schemes could be possible is not applicable for our scheme. In fact, our scheme's security depends on some secret information stored in the smart card, timestamp, and server

secret *s*. So, as a whole it is resistant against password table modification or any other similar type of insider attack.

Table 1. Computational Costs in Our Proposed Scheme

	User's Smart Card	Remote Server
Registration		1 scalar multiplication 2 hash operations
For User Authentication	1 scalar multiplication 1 hash operation 1 encryption operation	1 scalar addition 1 hash operation 2 pairing operations 1 exponent operation 1 decryption
For Server Authentication	1 scalar addition 1 scalar multiplication 0 or 1 hash operation (as mentioned in section 5.2)	2 scalar multiplications 1 or 2 hash operations (based on number of hashing used as the second hashing could be omitted if needed (section 5.2))

Other than these, in the step1 in login phase, by checking $V_i = H(PW_i)$ it is made sure that the legal user is using the smart card to make the login request. So, at each step a high level of security in maintained throughout our scheme.

5.3 Performance

Our scheme doesn't allow multiple logged in user with the same identity and password. Even if an adversary knows the identity and password of a legitimate user, he cannot do any harm unless he gets the smart card at hand. As some of the secret information are stored in the smart card, a user must have all the required items and values to access the server. In fact, if anybody knows and has everything required by legitimate user, he is allowed to access as a legitimate user. Any other attempt cannot succeed. Once the legitimate user removes the smart card from the input device, the session is expired and no attacker can input anything to access the server. Our scheme is also user-friendly as we allow free choosing of identity and password by the valid user. The user is even allowed to change his password whenever needed and before changing the password it is ensured that the legal user is involved in the password changing process. It should also be noted that in the password changing operation, there is no need to involve the RS rather the smart card itself can take care of it. As our scheme does not need any verification table in the server side, there is no need of extra memory or storage facility. During the login request, in our case, only three parameters are sent over the public channel which is pretty easy and the size of the login request message is less than other alternative solutions. Also we made sure that, our scheme provides better security and facilities than all other previously proposed schemes based on bilinear pairings. As this scheme prevents all sorts of known attacks and provides all the desired facilities, to the best of our knowledge this is the best solution proposed so far based on bilinear pairings.

Now, in Table 1 we show the computational costs of our proposed scheme. Compared to other proposed schemes, our scheme has fairly little amount of computational costs.

6. CONCLUSIONS

In this paper, we have presented an extended review of the remote user authentication schemes using smart cards which are based on bilinear pairings. We have shown that almost all of the proposed schemes so far have at least one weakness. After detailed analysis, we set our design goals and have proposed our solution which meets all the desired facilities for such authentication mechanisms. Our scheme provides robust security and is resistant against all the known attacks. Also it requires minimum amount of processing, memory, and transmission costs.

7. ACKNOWLEDGMENTS

This research was supported by the MKE under the ITRC support program supervised by the IITA"(IITA-2008-(C1090-0801-0016))". Dr. CS Hong is corresponding author.

8. REFERENCES

[1] Das, M. L., Saxena, A., Gulati, V. P., and Phatak, D. B. A Novel Remote User Authentication Scheme Using Bilinear Pairings, *Computers and Security*, Vol. 25, (2006) 184-189.

[2] Frey, G. and Rück, H.-G. A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves. *Mathematics of Computation*, Vol. 62, No. 206, (Apr. 1994), 865-874.

[3] Menezes, A. J., Okamoto, T., and Vanstone, S. A. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. on Inf. Theory*, Vol. 39, Issue 5, (1993), 1639-1646.

[4] Rhee, M.Y., *Internet Security: Cryptographic Principles, Algorithms and Protocols*, Wiley, 2003.

[5] Chou, J.-S., Chen, Y., and Lin, J.-Y. Improvement of Manik et al.'s remote user authentication scheme. available at, http://eprint.iacr.org/2005/450.pdf

[6] Thulasi, G., Das, M. L., Saxena, A. Cryptanalysis of recently proposed Remote User Authentication Schemes. available at, http://eprint.iacr.org/2006/028.pdf

[7] Jeon, J.-C., Kang, B.-H., Kim, S.-M., Lee, W.-S., and Yoo, K.-Y. An Improvement of Remote User Authentication Scheme Using Smart Cards. *LNCS 4325*, (2006), 416-423.

[8] Oh, J.-B., Jeon, J.-C., and Yoo, K.-Y. Further Improvement of Manik et al.'s Remote User Authentication Scheme Using Smart Cards. *ISPA 2006, LNCS 4331*, (2006), 57-64.

[9] Jia, Z., Zhang, Y., Shao, H., Lin, Y., and Wang, J. A Remote User Authentication Scheme Using Bilinear Pairings and ECC. *Proc. of ISDA'06*, Vol. 2, 2006, 1091-1094.

[10] Vo, D.-L. and Kim, K. Security Enhancement of a Remote User Authentication Scheme Using Bilinear Pairings and ECC. *Proceedings of NPC'07*, 2007, 144-147.

[11] Yang, C., Ma, W., Huang, B., and Wang, X. Password-Based Access Control Scheme with Remote User Authentication Using Smart Cards. *Proceedings of AINAW'07*, Volume 2, 2007, 448-452.