EFFICIENT AND SECURE DYNAMIC ID-BASED REMOTE USER AUTHENTICATION SCHEME WITH SESSION KEY AGREEMENT FOR MULTI-SERVER ENVIRONMENT

Rafael Martínez-Peláez[1, 2], Francisco Rico-Novella[1], Cristina Satizábal[3] and Jacek Pomykała[4]

[1]Telematic Engineering Department, Technical University of Catalonia
{rafaelm, f.rico}@entel.upc.edu

[2]Licenciatura en Informatica, Universidad de la Sierra Sur
rpelaez@unsis.edu.mx

[3]Division de Estudios de Postgrado, Universidad de la Sierra Sur
isatizabal@unsis.edu.mx

[4]Faculty of Mathematics, Informatics and Mechanics, Warsaw University
pomykala@mimuw.edu.pl

ABSTRACT

In 2007, Liao-Wang proposed a dynamic ID-based remote user authentication scheme for multi-server environment using smart cards. However, Hsiang-Shih demonstrated that Liao-Wang's scheme has security flaws. Moreover, Hsiang-Shih proposed an improvement scheme which resolves the security flaws of Liao-Wang scheme. In this paper, we propose an enhanced remote user authentication scheme which maintains the merits of Hsiang-Shih's scheme. Performance evaluation demonstrated the efficiency of the proposed scheme over related works. Security analysis proved that the proposed scheme is secure against known attacks.

KEYWORDS

Low-cost Cryptography, Multi-server, Mutual Authentication, Secure Communication, Smart-card

1. INTRODUCTION

Since the beginning of Internet, remote user authentication has been one of the major issues in network security. A traditional remote user authentication scheme permits identifying legal users, among those who are not, to get them access to resources. In order to improve network security, many schemes provide mutual verification. Mutual verification is the process in where each participant, in a communication, can verify the identity of each other. Other property desirable in a remote user authentication scheme is the key agreement between parties.

Lamport [1] introduced the concept of hash chain to authenticate remote user over open networks. Due to the low computational cost of a one-way hash function [2], [3], Lamport's scheme is feasible for practical implementation with lightweight devices. However, Lamport's scheme requires that the server maintains a verification table, making it vulnerable to steal information. Hwang et al. [4] proposed a server authentication scheme without verification table in 1990. Since Lamport's scheme and Hwang et al.'s scheme, several remote user authentication schemes with smart cards have been proposed [5], [6], [7], [8], [9], [10], [11].

DOI : 10.5121/ijnsa.2010.2409

In this paper, we focus on dynamic ID-based remote user authentication schemes using smart cards. The first remote user authentication scheme which introduced the concept of dynamic ID-based was proposed by Das et al. [12] in 2004. The scheme is based on one-way hash function and the server does not maintain a verification table. This scheme prevents that an attacker or an observer can know the identity of each user. However, Das et al.'s scheme is not suitable for multi-server environment. Afterwards, Liao-Wang [13] proposed a dynamic ID-based remote user authentication scheme for multi-server environment. However, Liao-Wang's scheme is vulnerable to the following common attacks: insider, masquerade, server spoofing and registration centre spoofing. Hsiang-Shih [14] proposed an improvement of Liao-Wang's scheme which allows mutual authentication, establishes a session key and prevents common attacks.

Although, the scheme proposed by Hsiang-Shih is more secure than Liao-Wang, this scheme has the following drawback: each user must known the identification of the server in order to create the login request message. In this case, if the user does not known the identification of the server he cannot create the login request message, making this scheme less practical for real implementation in a multi-server environment.

In this paper, we propose a new dynamic ID-based remote user authentication scheme based on Hsian-Shih's scheme. In contrast to the original scheme, the proposed scheme offers the following advantages: 1) it is easy-of-use - users can create the login request message without known the identification of each server; 2) more efficient in terms of computational cost; 3) and provides strong security.

The rest of this paper is organized as follows. In section 2, we review Hsian-Shih's scheme. Section 3 shows the details of the proposed scheme. In section 4, we carry out the security analysis of the proposed scheme. In section 5, we compare the proposed scheme with the related schemes. Conclusions are given in section 6.

2. RELATED WORKS

In this section, we review the scheme proposed by Hsiang-Shih which form the basis of our proposal. The notations used in this paper are summarized in Table 1.

Table 1. Notations.

Notation	Meaning
U	The user
S	The server
RC	The registration centre
ID	The identification of U
CID	The dynamic ID of U
SID	The identification of S
$H()$	A secure one-way hash function
$SK_{U\text{-}S}$	Session key between U and S
r, x, y, z	The secret keys maintained by RC
$\oplus$	Exclusive-or operation
$\|\|$	String concatenation operation

In general, dynamic ID-based remote user authentication schemes using smart card are defined by three phases: Registration, Login, and Mutual Verification and Session Key Agreement. In the registration phase, each user obtains the needed parameters in the scheme. In the login

phase, each user uses her smart card to initialize a communication with a remote server. In the mutual authentication and session key agreement phase, the participants (user and server) carry out the authentication process and establish a session key for securing the transmitted data. In addition, each scheme has a password change phase that enables users to update their passwords freely without establish a communication with *RC*.

2.1. Hsiang-Shih's Scheme

Hsiang-Shih's [14] scheme is an improvement of the scheme proposed by Liao-Wang [13]. It features are: 1) modification of the former protocol to provide mutual verification and establish a session key; and 2) improve security.

Registration phase. In this phase, *U* obtains the needed parameters to be part of the scheme. The process is as follows: *U* chooses and keys *ID*, *PW* and secret number b, and computes $h(b \oplus PW)$. Then, *U* sends *ID* and $h(b \oplus PW)$ to *RC* through a secure channel. Upon receiving the registration request message, *RC* computes the security parameters (from step 4 to 9) to make her member of the scheme. In the step 10, *RC* stores *V*, *B*, *H*, *R*, *h*() in *U*'s smart card. Finally, *U* enters *b* into her smart card. Table 2 shows the protocol.

Table 2. Registration protocol.

1. U:	ID, PW, b
2. U:	h(b ⊕ PW)
3. U → RC:	ID, h(b ⊕ PW)
4. RC:	T = h(ID || x)
5. RC:	V = T ⊕ h(ID || h(b ⊕ PW))
6. RC:	A = h(h(b ⊕ PW) || r) ⊕ h(x ⊕ r)
7. RC:	B = A ⊕ h(b ⊕ PW)
8. RC:	R = h(h(b ⊕ PW) || r)
9. RC:	H = h(T)
10. RC → U:	V, B, H, R, h()
11. U:	V, B, H, R, h(), b

Login phase. In this phase, *U* creates the login request message as follows: Firstly, *U* keys her *ID* and *PW*, and the identification of *S*. Then, her smart card recovers *T'* from *V* and computes *H**. The smart card checks whether or not *H** and *H* are equal. If the verification is correct, the identity of *U* is assured and the smart card generates a nonce *N*. The nonce *N* is used to compute the security parameters (*A*, *CID*, *P*, *Q*, *D*, *C*). Finally, *U* sends the login request message (*CID*, *P*, *Q*, *D*, *C*, *N*) to *S*. Table 3 shows the protocol.

Table 3. Login protocol.

1. U:	ID, PW, SID
2. U:	T' = V ⊕ h(ID || h(b ⊕ PW))
3. U:	H* = h(T')
4. U:	H* ?= H
5. U:	N
6. U:	A' = B ⊕ h(b ⊕ PW)
7. U:	CID = h(b ⊕ PW) ⊕ h(T' || A' || N)
8. U:	P = T' ⊕ h(A' || N || SID)
9. U:	Q = h(B || A' || N)

10. U:	$D = R \oplus SID \oplus N$
11. U:	$C = h(A' \,||\, N+1 \,||\, SID)$
12. U → S:	CID, P, Q, D, C, N

Mutual verification and session key agreement phase. In this phase, *S*, *RC* and *U* perform the following process:
Upon receiving the login request message, *S* generates a nonce N_1 and computes *M*. Then, *S* sends the user's verification message to *RC*. Afterward, *RC* recovers N_1' from *M* and computes *R**, *A** and *C** (steps 5, 6 and 7). *RC* checks whether or not *C** and *C* are equal. If they are equal, it generates a nonce N_2 and computes C_1 and C_2. In step 12, *RC* sends the acknowledgement message to *S*. *S* verifies the authenticity of *RC* through steps 13 and 14. If the verification process is correct, *S* computes and sends the challenge request message to *U*. *U* computes M_2* and compares it with M_2. If they are equal, the identity of *S* is assured. *C* computes and sends the challenge response message to *S*. *S* computes and compares M_3* with M_3. If the verification is correct, the identity of *U* is assured. Finally, *U* and *S* computes the session key SK_{U-B}. Table 4 shows the protocol.

Table 4. Mutual verification and session key agreement protocol.

1. S:	N_1
2. S:	$M = h(SID \,||\, y) \oplus N_1$
3. S → RC:	M, SID, D, C, N
4. RC:	$N_1' = h(SID \,||\, y) \oplus M$
5. RC:	$R^* = D \oplus SID \oplus N$
6. RC:	$A^* = R^* \oplus h(x \oplus r)$
7. RC:	$C^* = h(A^* \,||\, N+1 \,||\, SID)$
8. RC:	$C^* \stackrel{?}{=} C$
9. RC:	N_2
10. RC:	$C_1 = h(N_1 \,||\, h(SID \,||\, y) \,||\, N_2)$
11. RC:	$C_2 = A \oplus h(h(SID \,||\, y) \oplus N_1$
12. RC → S:	C_1, C_2, N_2
13. S:	$C_1^* = h(N_1 \,||\, h(SID \,||\, y) \,||\, N_2)$
14. S:	$C_1^* \stackrel{?}{=} C_1$
15. S:	$A' = C_2 \oplus h(h(SID \,||\, y) \oplus N_1$
16. S:	$T' = P \oplus h(A' \,||\, N \,||\, SID)$
17. S:	$H(b \oplus PW)' = CID \oplus h(T' \,||\, A' \,||\, N)$
18. S:	$B^* = A' \oplus h(b \oplus PW)'$
19. S:	$Q^* = h(B^* \,||\, A' \,||\, N)$
20. S:	$Q^* \stackrel{?}{=} Q$
21. S:	N_3
22. S:	$M_2 = h(B^* \,||\, N \,||\, A' \,||\, SID)$
23. S → U:	M_2, N_3
24. U:	$M_2^* = h(B^* \,||\, N \,||\, A' \,||\, SID)$
25. U:	$M_2^* \stackrel{?}{=} M_2$
26. U:	$M_3 = h(B \,||\, N_3 \,||\, A \,||\, SID)$
27. U → S:	M_3
28. S:	$M_3^* = h(B \,||\, N_3 \,||\, A \,||\, SID)$
29. S:	$M_3^* \stackrel{?}{=} M_3$
30. U:	$SK_{U-B} = h(B \,||\, A \,||\, N \,||\, N_3 \,||\, SID)$
31. S:	$SK_{U-B} = h(B \,||\, A \,||\, N \,||\, N_3 \,||\, SID)$

3. PROPOSED SCHEME

In this section, we propose an improvement on Hsiang-Shih's scheme which keeps the merits of the original scheme and is easy-of-use.

3.1. Registration phase

In this phase, when *U* wants to be a legal participant in the scheme, she must submit her identity *ID* to *RC*. In our scheme, *U* must apply in person at a Registration Centre office in where each new user must present identification document, birth certificate and other personal documents. New users must choose and submit her *ID* to *RC*. *RC* generates randomly a password *PW* and secret value *b*, and computes the security parameters for each new user. The process is as follows:

Firstly, *U* chooses and submits her *ID* to *RC*. Upon receiving the identification of *U*, *RC* generates randomly two values for each user, *PW* and *b*. Then, *RC* computes the security parameters (from step 4 to 10) using user's identification *ID*, secret keys (*x* and *z*), and user's password *PW* and secret value *b*. Finally, *RC* stores some secure parameters (*V*, *B*, *R*, *H*, *b*, *h*()) in the *U*'s smart card and sends this smart card to this user via secure channel. Table 5 describes the operations carried out by *U* and *RC*.

Table 5. Our proposed registration protocol.

1. U:	ID
2. U → RC:	ID
3. RC:	PW, b
4. RC:	T = h(ID || x || b)
5. RC:	V = T ⊕ h(ID || PW)
6. RC:	A = h(h(ID || PW) || z)
7. RC:	B = A ⊕ h(ID || PW || b)
8. RC:	R = h(h(PW || b) || x)
9. RC:	H = h(T)
10. RC:	I = H ⊕ R
11. RC → U:	V, B, H, I, b, h()

In addition, each server must obtain its security parameter $h(SID \| y)$ via a secure channel.

3.2. Login phase

In this phase, when *U* wants to login the server *S*, she must insert her smart card and key her *ID* and *PW*. Her smart card must verify the identity of *U* before creates and sends the login request message to *S*. The process is as follows:

In the first step, *U* keys her *ID* and *PW*. Then, her smart card recovers *T'* from *V*. The smart card computes *H** and checks whether or not is equal to *H*. If the verification is correct, the legality of *U* is assured and the smart card computes the needed parameters (from step 5 to 12) to create the login request message. Finally, *U*'s smart card sends the login request message ($h(PW \| b)$, *Q*, *D*, *CID*, *P*) to *S*. Table 6 shows the operations carried out by *U*.

Table 6. Our proposed login protocol.

1. U:	ID, PW
2. U:	$T' = V \oplus h(ID \| PW)$
3. U:	$H^* = h(T')$
4. U:	$H^* ?= H$
5. U:	$R' = H^* \oplus I$
6. U:	$A' = B \oplus h(ID \| PW \| b)$
7. U:	N
8. U:	$Q = R' \oplus N$
9. U:	$D = R' \oplus h(ID \| PW) \oplus N$
10. U:	$C = h(A' \| R' \| N)$
11. U:	$CID = h(PW \| b \| N) \oplus C$
12. U:	$P = h(C \| h(PW \| b \| N) \| Q)$
13. U → S:	$h(PW \| b), Q, D, CID, P$

Note that *U* does not need to know server's identity *SID*.

3.3. Mutual verification and session key agreement phase

In this phase, when *S* receives the login request message from *U*, *S* asks *RC* about the legality of *U*. *RC* verifies the legality of *S* and provides security information, which can be used to carry out the authentication of *U*, to *S*. Afterward, *S* and *U* complete the mutual authentication process. Finally, *U* and *S* establish a session key for protecting transmitted data. Table 7 shows the operations carried out by *S*, *RC* and *U*.

Table 7. Our proposed mutual verification and session key agreement protocol.

1. S:	N_1
2. S:	$M = h(h(SID \| y) \| N_1)$
3. S:	$O = M \oplus h(PW \| b)$
4. S → RC:	SID, N_1, O, Q, D
5. RC:	$h(SID \| y)^*$
6. RC:	$M^* = h(h(SID \| y)^* \| N_1)$
7. RC:	$M^* ?= M$
8. RC:	$h(PW \| b)' = M^* \oplus O$
9. RC:	$R^* = h(h(PW \| b)' \| x)$
10. RC:	$N' = R \oplus Q$
11. RC:	$h(ID \| PW)' = R^* \oplus D \oplus N'$
12. RC:	$A^* = h(h(ID \| PW)' \| z)$
13. RC:	$C^* = h(A^* \| R^* \| N')$
14. RC:	N_2
15. RC:	$M_1 = C^* \oplus h(h(SID \| y) \| N_2)$
16. RC:	$C_1 = A^* \oplus h(h(SID \| y) \| N')$
17. RC:	$C_2 = h(h(SID \| y) \| N') \oplus h(h(SID \| y) \| N_2) \oplus N'$
18. RC → S:	N_2, M_1, C_1, C_2
19. S:	$h(h(SID \| y) \| N_2)^*$
20. S:	$C' = M_1 \oplus h(h(SID \| y) \| N_2)^*$
21. S:	$h(PW \| b \| N)' = CID \oplus C'$
22. S:	$P^* = h(C' \| h(PW \| b \| N)' \| Q)$

23. S:	$P^* \stackrel{?}{=} P$
24. S:	N_3
25. S:	$M_2 = C' \oplus N_3$
26. S:	$M_3 = h(h(h(SID \| y) \| N_2)^* \| N_3)$
27. S → U:	C_1, C_2, M_2, M_3
28. U:	$h(h(SID \| y) \| N')' = A \oplus C_1$
29. U:	$h(h(SID \| y) \| N_2)' = h(h(SID \| y) \| N')' \oplus C_2 \oplus N$
30. U:	$N_3' = C' \oplus M_2$
31. U:	$M_3^* = h(h(h(SID \| y) \| N_2)' \| N_3')$
32. U:	$M_3^* \stackrel{?}{=} M_3$
33. U:	$M_4 = h(C \| h(h(SID \| y) \| N_2)' \| N_3')$
34. U → S:	M_4
35. S:	$M_4^* = h(C' \| h(h(SID \| y) \| N_2)^* \| N_3)$
36. S:	$M_4^* \stackrel{?}{=} M_4$
37. U:	$SK_{U-S} = h(h(PW \| b \| N) \| h(h(SID \| y) \| N_2)')$
38. S:	$SK_{U-S} = h(h(PW \| b \| N)' \| h(h(SID \| y) \| N_2))$

Upon receiving the login request message, *S* generates a nonce N_1 and computes *M* and *O*. *S* sends the user's verification request message (*SID*, N_1, *O*, *Q*, *D*) to *RC*. After receiving the message from *S*, *RC* computes the security parameter of *S* ($h(SID \| y)^*$). Then, *RC* verifies if *S* is a legal server comparing M^* with *M* (step 7). If the identity of *S* is assured, *RC* recovers $h(PW \| b)'$ and computes R^* using its secret key *x*. Afterward, *RC* recovers *N'* and $h(ID \| PW)'$ from *Q* and *D*, respectively. *RC* computes A^* using its secret key *z* and $h(ID \| PW)'$. *RC* has all the security parameters to compute C^*. Finally, *RC* computes and sends the user's verification response message (N_2, M_1, C_1, C_2) to *S*.

After receiving the message from *RC*, *S* verifies the identity of *RC* computing steps 19 to 23. If the verification is correct, *S* generates a nonce N_3 and computes a challenge (M_2, M_3). Finally, *S* sends the login response message (C_1, C_2, M_2, M_3) to *U*.

Upon receiving the login response message, *U* computes the challenge response by using C_1 and C_2. By means of C_1 and C_2, *U* can recover two security parameters computed by *RC*. Then, *U* recovers N_3 generated by *S* and computes M_3^*. Afterward, *U* checks whether or not M_3^* is equal to M_3. If there are equal, the identity of *S* is assured. Finally, *U* computes and sends the challenge-response message (M_4) to *S*.

Upon receiving the challenge-response message, *S* computes and checks whether or not M_4^* and M_4 are equal. If there are equal, the identity of *U* is assured.

In steps 37 and 38, *U* and *S* computes the session key SK_{U-S}.

3.4. Password phase

In this phase, *U* can update her password whenever she wants. The process is as follows: *U* inserts her smart card and keys her *ID* and *PW* and request to change her *PW*. Then, *U*'s smart card checks the validity of *U* comparing H^* with *H*. If they are equal, *U* can key her new password PW_{new}. *U*'s smart card computes $V^* = T \oplus h(ID \| PW_{new})$ and replaces *V* with V^*.

4. SECURITY ANALYSIS

In this section, we describe the security of the proposed scheme. It demonstrates that our scheme can resistance known attacks to provide strong security.

Leak of password. If a malicious user obtains the *U*'s smart card, she cannot recover *U*'s identification *ID* and password *PW* by using *V*, *B*, *H*, *I* and *b* or other combination of them.

Masquerade registration centre. If a malicious user wants to impersonate the registration centre, she needs to know *y*, *x*, and *z* for computing *C*, M_1, C_1 and C_2.

Masquerade server attack. If a malicious user attempts to impersonate the server *S*, she must be able to forge a valid challenge (M_2, M_3). However, this attempt will fail, because it is infeasible to compute $M_3 = h(h(SID \| y) \| N_2) \| N_3)$ without the knowledge of $h(SID \| y)$. Note also that there is no way to compute M_3, even if *SID* and N_1 are known by the malicious user.

Masquerade user attack. If a malicious user tries to masquerade as legal user, she must be able to forge a valid login request message ($h(PW \| b)$, *Q*, *D*, *CID*, *P*). However, it is impossible to compute $CID = h(PW \| b \| N) \oplus C$ or $P = h(C \| h(PW \| b \| N) \| Q)$ without the knowledge of *A*, *R*, $h(ID \| PW)$ and the user's nonce *N*. Although, the malicious user can access all the secure parameters stored in the smart card [15], [16] (*V*, *B*, *H*, *I*, and *b*), she cannot obtain security parameters $h(ID \| PW)$, *A* and *R* from *V*, *B*, *H*, *I*, *b*, *Q*, *D*, *CID*, *P* or any other combinations of them.

Replay attack. The proposed scheme uses nonce to withstand replay attack. Nonces *N*, N_1, N_2 and N_3 are generated independently and their values differ among sessions. For that reason, malicious users cannot get access to the system by using previous messages.

Secure password change. In the proposed scheme, *U*'s smart card verifies the correctness of the actual password *PW* by comparing *H** with the stored *H*. If the verification process is correct, *U*'s smart card accepts the new password PW_{new}.

Secret key guessing attack. A malicious user can try to extract the registration centre keys x, y and z from $A = h(h(ID \| PW) \| z)$, $B = A \oplus h(ID \| PW \| z)$ and $R = h(h(PW \| b) \| x)$. However, this attempt will fail because it is computationally infeasible to invert a one-way hash function.

Stolen secret keys. A malicious user can try to extract the *RC*'s secret keys *x* and *z* from *A* and *R*. However, she cannot obtain these parameters because it is computationally infeasible to invert a one-way hash function h().

Stolen verification table. Because the servers and the registration centre do not store and maintain any verification table, the proposed scheme is secure against this attack.

5. PERFORMANCE COMPARISON

In this section, we compare the computational cost, communication cost and storage capacity of the proposed scheme with two other multi-server authentication schemes. The performance comparison is summarized in Table 8.

Due to the limited computational power of smart cards, the scheme must take computational cost evaluation into consideration. In order to carry out the computational cost evaluation, we use the following notation T_h as the execution time for one-way hash function. Because

exclusive-or operation requires very low execution time, it is usually neglected considering its computational cost.

Computational cost. The computational cost is defined as the total time of various operations executed in each step. From Table 8, we can observe that our scheme is more efficient than Liao-Wang's scheme and Hsiang-Shih's scheme. In our scheme, users perform 6 one-way hash functions in the login phase. On the other hand, in the scheme proposed by Hsiang-Shih users perform 7 one-way hash functions. The totally number of one-way hash functions carried out by each user in our scheme is 9 while users in the scheme proposed by Hsiang-Shih performed 11 one-way hash functions. The computational cost required by our scheme and Liao-Wang's scheme is the same from users' point of view. Moreover, servers performed fewer operations in our scheme than in previous works.

Communication cost. Table 8 shows the communication cost required by each scheme. In the login phase, our scheme has a better performance than Hsiang-Shih's scheme. Although our scheme requires more communication cost than Hsiang-Shih's scheme in the mutual verification and session key agreement phase, the cost is very low for the current network technologies.

Storage capacity. We evaluate the storage capacity required by our scheme. We assume that the output size of a one-way hash function, random numbers and secret keys are 160-bit, and identification, password and nonce are 32-bit length; so the memory needed in the user's smart card is 800(5*160) bits, the server requires 160(1*160) bits to store its secret parameter and the registration centre requires 480(3*160).

Table 8. Performance comparison between our scheme and related schemes.

		[13]	[14]	**Ours**
Computational cost in the registration phase	*U*	-	$1T_h$	-
	S	-	-	-
	RC	$5T_h$	$6T_h$	$6T_h$
Computational cost in the login phase	*U*	$6T_h$	$7T_h$	$6T_h$
	S	-	-	-
	RC	-	-	-
Computational cost in the mutual verification and session key agreement phase	*U*	$3T_h$	$3T_h$	$3T_h$
	S	$7T_h$	$8T_h$	$6T_h$
	RC	-	$5T_h$	$7T_h$
Communication cost in the login phase	*U*	512bits	832bits	800bits
	S	-	-	-
	RC	-	-	-
Communication cost in the mutual verification and session key agreement phase	*U*	160bits	160bits	160bits
	S	192bits	736bits	1312bits
	RC	-	352bits	512bits
Storage capacity	*U*	640bits	800bits	800bits

6. CONCLUSIONS

We have analyzed the scheme proposed by Hsiang and Shih from the following point of views: 1) security; and 2) efficiency. Their scheme prevents the known attacks, making it secure. On the other hand, their scheme requires high computational cost, as we explained in Table 8. Moreover, the scheme is not easy-of-use because each user must know the server's identity which is not comfortable for users. Hence, we have proposed a new dynamic ID-based remote user authentication scheme using smart cards for multi-server environment. We have shown that

our scheme is more efficient, in terms of computational cost, and at the same time provides strong security. The proposed scheme prevents the known attacks, keeping the merits of Hsiang and Shih's scheme. Furthermore, the proposed scheme provides the following properties: establish a session key, mutual verification, easy-of-use, without verification table and based on one-way hash function.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish public funded projects ARES (CONSOLIDERINGENIO-2010 CSD2007-00004) and ITACA (TSI2006-13409-C02-02), and graduate scholarship from CONACYT (Mexico).

REFERENCES

[1] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, pp. 770-772, 1981.

[2] R. Rivest, "RFC 1321 - the MD5 message-disgest algorithm," IETF Working Group 1992.

[3] NIST, "Secure Hash Standard (SHA), FIPS PUB 180-1," National Institute of Standards and Technology 1995.

[4] T. Hwang, Y. Chen, and C. S. Laih, "Non-interactive password authentication without password tables," presented at IEEE Region 10 Conference on Computer and Communication System, 1990.

[5] C. C. Chang and T. C. Wu, "Remote password authentication with smart cards," *IEE Proceedings-E*, vol. 138, pp. 165-168, 1991.

[6] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "An Efficient and practical solution to remote authentication: smart card," *Computers & Security*, vol. 21, pp. 372-375, 2002.

[7] C. C. Lee, L. H. Li, and M. S. Hwang, "A remote user authentication scheme using hash functions," *ACM SIGOPS Operating Systems Review*, vol. 36, pp. 23-29, 2002.

[8] W. C. Ku and S. M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 204-207, 2004.

[9] B. Wang and Z. Q. Li, "A forward-secure user authentication scheme with smart cards," *International Journal of Network Security*, vol. 3, pp. 116-119, 2006.

[10] M. A. Ahmed, D. R. Lakshmi, and S. A. Sattar, "Cryptanalysis of a more efficient and secure dynamic ID-based remote user authentication scheme," *International Journal of Network Security & Its Applications*, vol. 1, pp. 32-37, 2009.

[11] R. Martínez-Peláez, F. Rico-Novella, C. Satizabal, and J. Pomykala, "Strong remote user authentication scheme using smart cards," presented at Eighth International Network Conference, 2010.

[12] M. L. Das, A. Saxena, and V. P. Gulati, "A Dynamic ID-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 629-631, 2004.

[13] Y.-P. Liao and S.-S. Wang, "A Secure Dynamic ID Based Remote User Authentication Scheme for Multi-server environment," *Computer Standards & Interfaces*, vol. Available online, 2007.

[14] H. C. Hsiang and W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, pp. 1118-1123, 2009.

[15] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology - Crypto'99*, vol. LNCS 1666, 1999, pp. 388-397.

[16] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, pp. 541-552, 2002.

Authors

Rafael Martínez-Peláez received his degree in Computational System Engineering from the Universidad del Valle de Mexico in 2004. Currently, he is working on a PhD in Telematic Engineering at the Technical University of Catalonia (Spain). Presently, he works in the Licenciatura en Informatica of Universidad de la Sierra Sur and collaborates with the Tecnologias de Informacion y Comunicaciones group. His main areas of interest are electronic and mobile payment systems, and applications of smart card and biometrics with security protocols design.

Francisco Rico-Novella received his degree in Telecommunication Engineering and his PhD from the Technical University of Catalonia (Spain) in 1989 and 1995, respectively. Presently, he works in the Department of Telematic Engineering with the Telematics Service group. His current research interests include network security and electronic commerce.

Cristina Satizábal received her degree in Electronic and Telecommunications Engineering from Cauca University (Colombia) in 2000 and her PhD in Telematic Engineering from the Technical University of Catalonia (Spain) in 2007. Currently, she is part of the División de Estudios de Postgrado of Universidad de la Sierra Sur (Mexico). Her current research interests include network security and e-government.

Jacek Pomykała received his Master degree in 1981 and PhD degree in 1986 from the Warsaw University. Presently, he works in the Faculty of Mathematics Informatics and Mechanics of Warsaw University, and Computer Science Department of Warsaw Management Academy. His research interests include number theory, cryptology, computational complexity, and computer systems security.