

Robust and Efficient Authenticated Key Agreement Scheme for Low-cost RFID Tags

Wen-Shenq Juang¹, Hung-Yi Chang¹, Sian-Teng Chen²
Hui-Chin Tseng¹ and Yi-Chun Yeh¹

Department of Information Management¹
National Kaohsiung First University of Science and Technology
Kaohsiung, Taiwan
[wsjuang, leorean]@ccms.nkfust.edu.tw

Department of Information Management²
Shih Hsin University
Taipei, Taiwan

Abstract. For providing secure network environments, many authentication schemes have been proposed for preventing various kinds of attacks. Among them, many authentication schemes were proposed for the low-computation devices, like the smart cards and the RFID tags. Since the computation capacity of these devices is low, only low cost operations can be used. In this paper, we propose an efficient and robust authenticated key agreement scheme for the low-cost RFID tags. By our proposed scheme, eligible users can get the provided services using RFID tags securely and efficiently, and the service providers can authenticate users securely and efficiently. Also, even if an attacker can get the RFID tag and then gets the data stored in the tag's memory, she/he cannot use this stored information to derive the transmitted encrypted messages that are sent before she/he got the tag. Thus, our proposed scheme is robust and can provide forward-secrecy.

Keywords: RFID security, mutual authentication, key agreement, forward secrecy, denial of service attack, robustness, privacy protection.

1 Introduction

When people want to get services over the Internet, they usually care the problems about network security. This is a significant concern in the network. Many people and companies are concerned with the sensitive business data, important person information, which is stored in computers. If the above data is exposed by the attacker, the adversary may counterfeit the identity of a legal person to login into servers providing controlled services. It will lead to a large damage for a legal person.

If a security problem had been found, many security mechanisms may be proposed to prevent this problem. The authentication protocol is one of the mechanisms for preventing an illegal user to use the network services. Since Lamport [14] proposed

a password authentication scheme in 1981 to achieve the user authentication, many schemes have been proposed [1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 13, 16, 20, 22]. These proposed schemes can solve some problems of the Lamport's scheme and can prevent some of the adversary's attacks [9, 10, 11, 13, 16, 20].

In 2007, Le *et al.* [15] proposed a forward-secure RFID authentication and key exchange scheme. In their paper, they claimed that their scheme can provide forward-secrecy in the RFID system, and they claimed their scheme only needs low computation operation by using the pseudo-random function.

In this paper, we propose a new RFID scheme that not only satisfies all the benefits of Le *et al.*'s scheme but also that the computation cost is lower than Le *et al.*'s scheme.

The paper is organized as follows. In Section 2, we review the related works. In Section 3, we propose our proposed scheme. In Section 4, the security analysis of our proposed scheme is given. In Section 5, we make a comparison among our proposed scheme and the related schemes. In Section 6, we have a discussion. Finally, we make a conclusion in Section 7.

2 Related Works

In this section, we will review the related RFID authentication schemes. In 2007, Le *et al.* [15] proposed a forward-secure RFID authentication and key exchange scheme for RFID devices. In this section, we review Le *et al.*'s scheme, which consists of two protocols, O-FRAP and O-FRAKE. We describe their scheme as follows.

In Le *et al.*'s proposed scheme, it needs some trusted setup and a protected server database. The trusted setup is done in a physically secure environment. Each tag and the server both need to store a fresh, unique key triple (r_i, k^a, k^b) , which is randomly generated. The value r_i is a one-time-use anonym for the tag and is used for optimistic key-retrieval. The values k^a and k^b both are the tag's authentication key. k^a is used in the authentication protocol and updated after each successful authentication. k^b is a secondary key, that is used in the key-exchange protocol and also will be re-computed after the key-exchange protocol.

There is a key triple stored in the tag's non-volatile memory and in the server's database D . The form in the server's database D is $\langle i, previous_i, current_i \rangle$. At the setup phase, $previous_i$ is (\perp, \perp, \perp) , and $current_i$ is (r_i, k^a, k^b) . In Le *et al.*'s scheme, the server must keep a pair of key triples for each tag to preserve consistency since keys may be modified in the presence of active adversaries. Since the server computes the updated key triple before the tag can update the key triple, an adversary could tamper with the communication channel and try to prevent the tag from computing the updated key. For solving this problem, during an authentication phase, the server can detect if the tag is using $previous_i$ or $current_i$. If the tag uses $current_i$, then the server will replace $previous_i$ with $current_i$ and store the newly computed value into the $current_i$. If the tag uses $previous_i$ instead, then

$previous_i$ is preserved and $current_i$ is replaced with newly computed value. This operation is denoted $D.update(i)$ in Le *et al.*'s scheme.

2.1 O-FRAP

Le *et al.*'s first protocol O-FRAP is an optimistic forward-secure RFID authentication protocol. In this protocol, r_{sys} and r_{tag} are values, which is generated randomly by the server and the tag. This approach can preserve the anonymity of the session and prevent the replay attack. The value r_{tag} is generated for optimistic identification of the tag. The value k_{tag}^a is the tag's current key and will be updated by the server after the authentication protocol.

After being initialized by the server, the tag uses the pseudo-random function F to compute four values v_1, v_2, v_3, v_4 .

In O-FRAP, v_1 is used for updating the pseudo-random value r_{tag} ; v_2 is used for authenticating of the tag; v_3 is used for authenticating the server; v_4 is used for updating k_{tag}^a . In Le *et al.*'s O-FRAP protocol, the four values computed by the server by applying the pseudo-random function F to $(k_{tag}^a, r_{tag} \parallel r_{sys})$ are denoted as $v_1^*, v_2^*, v_3^*, v_4^*$. These values correspond to the non-starred values when the adversary is passive.

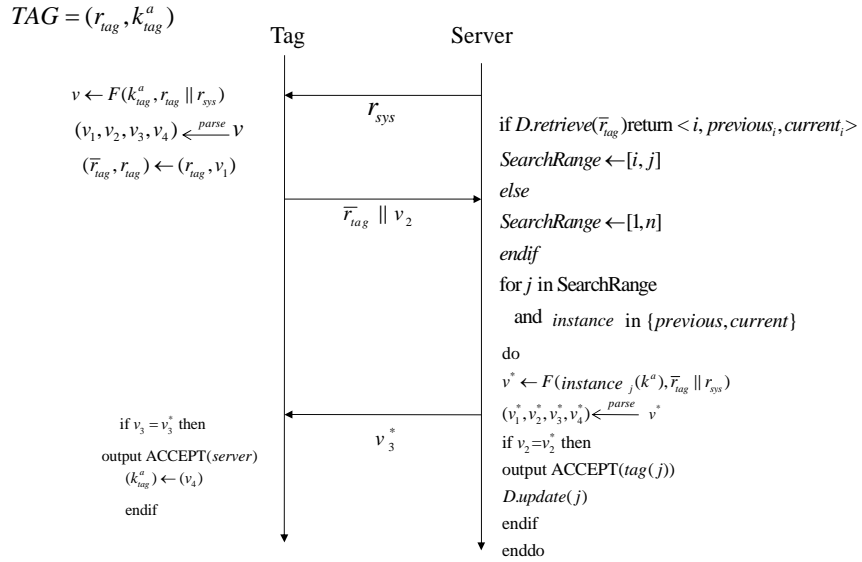


Figure 1. Le *et al.*'s O-FRAP protocol

Note that after each authentication, the tag key k_{tag}^a will be updated. This approach can give a strong separation between sessions. In particular, if a tag is attacked successfully by an adversary, the attacker cannot derive the relationship between sessions using the transcripts of earlier sessions. Le *et al.*'s protocol O-FRAP is shown in Figure 1.

2.2 O-FRAKE

Le *et al.* proposed another protocol O-FRAKE, that is an optimistic forward-secure RFID authenticated key exchange (AKE) protocol. This protocol is essentially the same as O-FRAP excluding that it uses five random numbers v_1, v_2, v_3, v_4, v_5 , which are generated by the pseudo-random function F . For securing the communication channel between the server and the tag, the protocol will output the value k_{tag}^b , which is an agreed session key for securing the subsequent communication. Le *et al.*'s protocol O-FRAKE is shown in Figure 2.

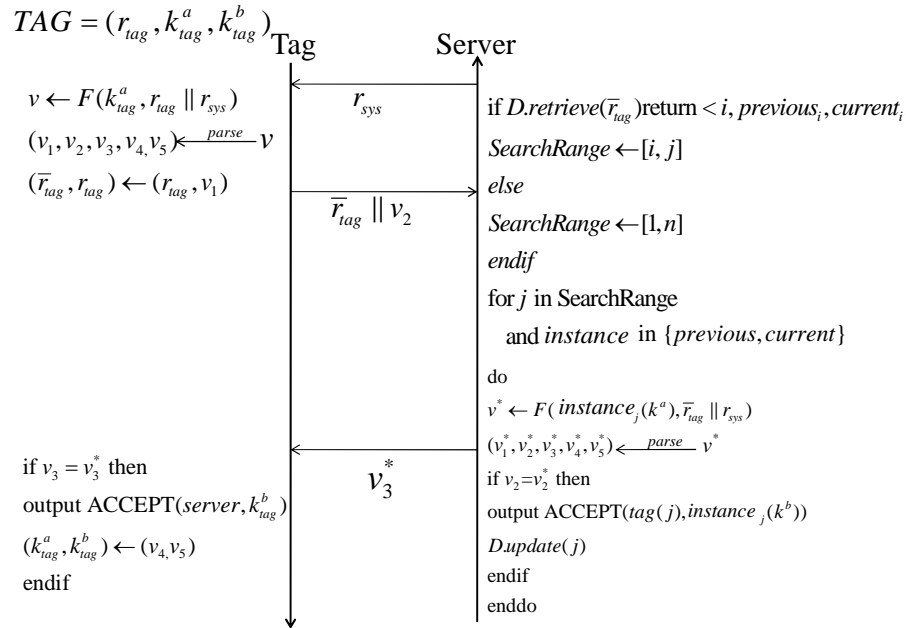


Figure 2. Le *et al.*'s O-FRAKE protocol

3 Our proposed scheme

In this section, we demonstrate our proposed RFID scheme. Our proposed scheme is robust and efficient by only using lightweight operations, e.g., the exclusive-or operation or the addition operation, in the tag.

In our proposed RFID scheme, there are some values stored securely in the RFID tag in advance. In the tag, the memory stores $\{ID_E, S_k, C_k\}$, where $ID_E = E_x(ID_k)$, ID_k is equal to $h(ID_{k-1} \| R_k)$ that is the tag's identity for the (k) th authentication protocol, x is a secret key kept secret and maintained by the server, and $S_k = h(x, R_k)$ is a shared key between this RFID tag and server. R_k is a random value that is chosen by the server in the authentication protocol and will be re-computed after the success of the authentication protocol. C_k is a verification value used in the authentication protocol. In our proposed RFID scheme, the server needs to store R_k and the R_{k-1} in its verification table for verification.

When a tag needs to authenticate a server for the k th time login, the tag will compute a randomly value N_1 , and then send $\{ID_E, C_k, S_k \oplus N_1\}$ to the server.

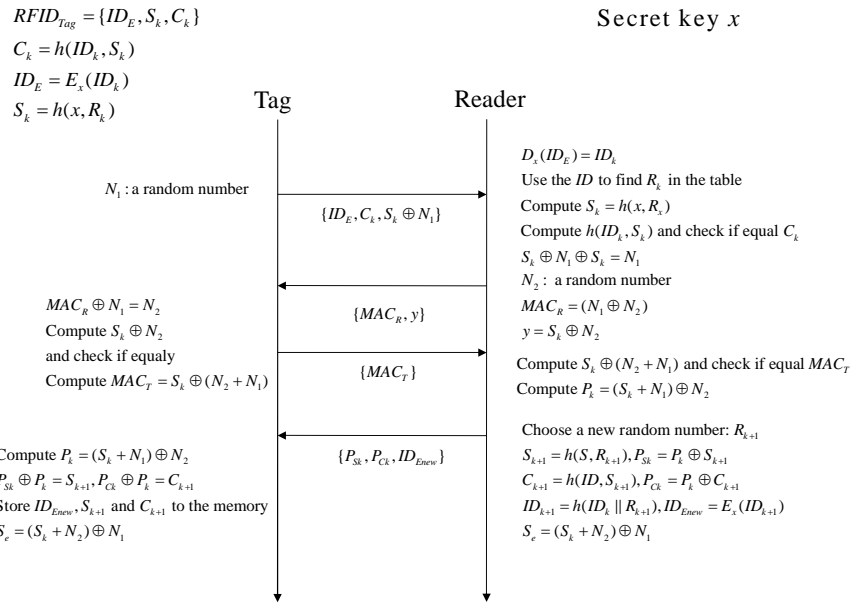
After the server receiving this message $\{ID_E, C_k, S_k \oplus N_1\}$, the server uses the secret key x to decrypt ID_E to obtain the tag's identity ID_k . It then uses the identity to search R_k from the verification table. After finding it, the server will compute $S_k = h(x, R_k)$, compute $h(ID_k, S_k)$, and check if it is equal to C_k . If it not equal to C_k , the server will retrieve the R_{k-1} from the table, compute $S_k = h(x, R_{k-1})$, create $h(ID_k, S_k)$, and check if it is equal to C_k . If it is not equal to C_k , the server will cancel this authentication protocol. Otherwise, the server computes $S_k \oplus N_1 \oplus S_k$ to get the nonce N_1 and generates the server's nonce N_2 . The server will compute $MAC_R = (N_1 \oplus N_2)$ and $y = S_k \oplus N_2$, then sends MAC_R and y to the tag.

When the tag getting the data successfully, the tag computes $MAC_R \oplus N_1$ to obtain the nonce N_2 and compute $S_k \oplus N_2$ to check if it is equal to y . If it is not equal to y , the tag will suspend this authentication protocol. Otherwise the tag computes $MAC_T = S_k \oplus (N_2 + N_1)$ and sends it to the server.

When receiving MAC_T from the tag, the server then computes $S_k \oplus (N_2 + N_1)$ and checks if it is equal to MAC_T . If not, the server stops this protocol. Otherwise the server computes $P_k = (S_k + N_1) \oplus N_2$ and chooses a new random number R_{k+1} , where R_{k+1} is a randomly value used for creating the $(k+1)$ th's shared key and verification value. After computing the values, the server computes $S_{k+1} = h(x, R_{k+1})$, $C_{k+1} = h(ID_k, S_{k+1})$ and $ID_{k+1} = h(ID_k \| R_{k+1})$. Then it sends $P_{S_k} = P_k \oplus S_{k+1}$, $P_{C_k} = P_k \oplus C_{k+1}$ and $ID_{E_{new}} = E_x(ID_{k+1})$ back to the tag. Finally,

the server stores R_{k+1} and ID_{k+1} in its verification table. If the session key agreement is needed, the server can compute a session key $S_e = (S_k + N_2) \oplus N_1$.

Finally, when the tag receiving $P_{Sk} = P_k \oplus S_{k+1}$, $P_{Ck} = P_k \oplus C_{k+1}$ and $ID_{E_{new}} = E_x(ID_{k+1})$ from the server, the tag will compute $P_{Sk} \oplus P_k = S_{k+1}$ and $P_{Ck} \oplus P_k = C_{k+1}$ to get S_{k+1} and C_{k+1} . Then it stores the values into the memory for the use in the next authentication protocol. If the session key agreement is needed, the tag can compute a session key $S_e = (S_k + N_2) \oplus N_1$. The robust RFID authentication protocol is described in Figure 3.



server share the common session key S_e . We can say that the authentication protocol is complete for mutual authentication between A and B [8, 23] if there is an S_e such that A believes $A \xleftrightarrow{S_e} B$ and B believes $A \xleftrightarrow{S_e} B$ for that transaction. When a scheme can deduce the following statement [8, 23]: A believes B believes $A \xleftrightarrow{S_e} B$ and B believes A believes $A \xleftrightarrow{S_e} B$, we can say that the scheme satisfies strong mutual authentication.

In Step 2 of our proposed RFID scheme, after A receiving the message $\{MAC_R, y\}$ from B , it will compute $MAC_R \oplus N_1 = N_2$ and $S_k \oplus N_2$ and check if $S_k \oplus N_2$ is equal to y . After finishing this step, A can compute the session key $S_e = (S_k + N_2) \oplus N_1$ and will believe $A \xleftrightarrow{S_e} B$. Since the nonce N_1 is selected by A , A believes that N_1 is fresh and can only be decrypted by B using the shared secret key S_k , and only B can use R_k to compute $S_k = h(x, R_k)$. Then A believes B believes $A \xleftrightarrow{S_e} B$.

In Step 3 of our proposed scheme, after B receiving the message $\{MAC_T\}$ from A , B first computes $S_k \oplus (N_2 + N_1)$ and checks if it is equal to $\{MAC_T\}$. If yes, B can compute the session key $S_e = (S_k + N_2) \oplus N_1$ and then believes $A \xleftrightarrow{S_e} B$. Since the nonce N_2 is chosen by B , B believes that the nonce N_2 is fresh. On receiving the authenticator $\{MAC_T\}$ from A , B can verify N_2 is embedded in $\{MAC_T\}$ by A and then B believes A believes $A \xleftrightarrow{S_e} B$.

4.2 Preventing the replay attack

When an attacker tries to camouflage an RFID by resending the messages transmitted between the tag and the server, we call that is the replay attack [21]. In our RFID scheme, we use the nonces to prevent the replay attack. In our RFID scheme, the tag computes a nonce N_1 in Step 1, and sends N_1 to the server. The second nonce N_2 is selected by the server, and also sent to the tag. In the scheme, the tag and the server will send back the corresponding two responses and verify that they are fresh by checking the nonces.

4.3 Forward-secrecy

If an attacker can get the tag, he may obtain the secret data that is stored in the tag's memory. However, even if the attacker can do that, she/he also cannot use the secret data to derive the transmitted encrypted messages that are sent before the attacker got the tag. If a scheme can prevent this kind of attack, we say that the scheme can provide forward-secrecy.

In our proposed RFID scheme, the shared key S_k is a one-time key that will be changed after a successful authentication. Therefore, in our RFID scheme, although

the attacker can obtain the tag's secret data, he also cannot use the share key S_k to derive the transmitted encrypted messages that are sent before the attacker gets the tag. So, our proposed RFID scheme can provide forward-secrecy.

4.4 Preventing the denial of service attack

In our proposed scheme, the server stores R_k and R_{k-1} in the table, where R_k is a random value chosen by the server in the authentication protocol and will be re-computed after the success of the authentication protocol and the server can use R_k to compute the share in the authentication protocol. In the authentication protocol, the server will store the current value R_k and old value R_{k-1} in the table. If an attacker uses the denial of service attack to attack our scheme, even if the attacker cuts the message successful and the tag's shared key is elder than the server, the tag also can authenticate the server by using the old shared key since the server has stored the old value R_{k-1} in the table.

5 Performance consideration

We will show that the communication and computation cost of our proposed scheme and the related scheme in this section. Beside that, we also will demonstrate the capability comparisons among our proposed scheme and related schemes.

5.1 Low communication and computation cost

We assume that the block size of secure symmetric cryptosystems is 128 bits [17, 19] and the output size of secure one-way hashing functions [18, 24] is 128 bits.

In our proposed RIFD scheme, only the exclusive-or operation, the random number generation function and the addition operation are used in the tags. In our proposed RFID scheme, the tag's computation cost is one random number generation operation, seven exclusive-or operations and two addition operations. The server's computation cost is one symmetric decrypt operation, two random number generation operations, three one-way hash operations, seven exclusive-or operations, and two addition operations.

In Henrici and Muller's RFID scheme [7], the tag's computation cost is three one-way hash operations. The server's computation cost is three one-way hash operations.

In the Le *et al*'s protocol [15], a pseudo-random number generation function F is used. The cost of the function F is near to a one-way hash function. Therefore, in the Le *et al*'s scheme, the tag needs five one-way hash operations. The server needs five one-way hash operations.

In Weis *et al*'s RFID scheme [21], the tag's computation cost is two one-way hash operations. The server's computation cost is N one-way hash operations, where N is the number of tags.

In our RFID scheme and the Le *et al*'s scheme, if the system needs to agree a session key, both schemes need some extra computation cost. In our proposed scheme, the tag and the server both need one extra x-or operation and one extra addition operation. In the Le *et al*'s scheme, the tag and the server both need one extra one-way hash operation. The efficiency comparison between our RFID scheme and related schemes is shown in Table 1.

Table 1. Efficiency comparison among our RFID scheme and related schemes

	E1	E2	E3
Our RFID scheme	128 bits	7 Xor + 2 Add	2 Sym + 4 Hash + 7 Xor + 2 Add
Henrici and Muller [7]	128 bits	3 Hash	3 Hash
Le <i>et al.</i> [15]	128 bits	5 Hash	5 Hash
Weis <i>et al.</i> [21]	128 bits	2 Hash	N Hash

E1: Key length needed; E2: Computation cost for a tag;
E3: Computation cost for a server; Hash: Hashing operation;
Sym: Symmetric encryption or decryption; Xor: Exclusive-or operation;
Add: Addition operation; N: Number of tags

5.2 Non-duplicability

If an adversary can duplicate a tag perfectly, the adversary may try to use the same tag information to pass another authentication. In our proposed RFID scheme, the stored secret data is a one-time key. After the success of an authentication protocol, the shared key will be changed by the server. Therefore, if an adversary copies a tag, there is only one tag can be used by the adversary since the tag's secret key will be changed, and the duplicable tag's shared key is different with the server.

5.3 Non-duplicability

In our RDIF scheme, we use two nonces N_1 and N_2 to prevent the replay attack. No logical time clocks are needed in our scheme.

5.4 Anonymity

The tag's identity ID_k in our RFID scheme is included in ID_E , which is sent to the server and encrypted by using the secret key x . Despite of the tag, only the server can decrypt ID_E and get ID_k . In our proposed scheme, since the tag's identity

ID_k will be re-computed after the authentication, the attacker cannot recognize the tag from the identity. Therefore, our proposed RFID scheme can provide anonymity.

5.5 Session key agreement

In our RFID scheme, the tag and the server both can agree a session key $S_e = (S_k + N_2) \oplus N_1$ after the authentication protocol if it is needed.

The capability comparison among our RFID scheme and related schemes is shown in Table 2.

Table 2. Capability comparisons among our RFID scheme and related schemes

	C1	C2	C3	C4	C5	C6	C7	C8	C9
Our RFID scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Henrici and Muller [7]	No	No	Yes	Yes	No	Yes	Yes	No	No
Le <i>et al.</i> [15]	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Weis <i>et al.</i> [21]	No	No	No	No	Yes	No	Yes	Yes	No

C1: Low communication and computation cost; C2: Mutual authentication;
 C3: Preventing the replay attack; C4: Forward-secrecy;
 C5: Preventing the denial of service attack; C6: Non-duplicability;
 C7: No time-synchronization problem; C8: Anonymity; C9: Session Key agreement

6 Discussion

We will discuss our proposed schemes for more detail considerations in this section. In Section 3, we describe that how to generate a session key and to check if the shared key is legal in our proposed RFID scheme. In some RFID systems, the session key is not necessary. In this kind of systems, the authentication protocol only needs to authenticate between the RFID tag and the server. In our proposed RFID scheme, the session key generation can be an option function for the server and the tag.

If there is a secure communication between them after the authentication, it must generate the session key in our scheme. The system must pay the additional computation cost. In our proposed RFID scheme, it is one exclusive-or operation and one additive operation for the tag and the server.

In our proposed RFID scheme, we can prevent the denial of service attack. If the system has generated the session key in the protocol, the system can prevent the

denial of service attack more easily. After the authentication protocol is successful and the tag and the server have generated the session key each other, the server can use the session key to encrypt the message to ask the tag if it has changed the newest shared key. This approach can help the system to prevent the denial of service attack.

7 Conclusions

In this paper, we have proposed a low-cost authentication and key agreement RFID protocol. The proposed scheme can resist from well-known attacks and provide many nice capabilities. In our proposed RFID scheme, we can provide forward-secrecy. Our proposed RFID scheme in the tag only uses the low cost exclusive-or operation and addition operation. Our proposed RFID authentication protocol also can provide identity protection to protect the identity of users or tags.

Acknowledgments. This work was supported in part by the National Science Council of the Republic of China under the Grant NSC 97-2221-E-327-036 and NSC 98-2221-E-327-026.

References

1. A. Awasthi and S. Lal, "A Remote User Authentication Scheme Using Smart Cards with Forward Secrecy," *IEEE Transactions on Consumer Electronics*, Vol. 49, No. 4, pp. 1246-1248, 2003.
2. A. Awasthi and S. Lal, "An Enhanced Remote User Authentication Scheme Using Smart Cards," *IEEE Transactions on Consumer Electronics*, Vol. 50, No 2, pp. 583-586, 2004.
3. S. Bellare and M. Merritt, "Augmented Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," *Research in Security and Privacy, Proceedings IEEE Computer Society Symposium*, pp. 72-84, 1992.
4. Y. Chang and C. Chang, "Authentication Schemes with No Verification Table," *Applied Mathematics and Computation*, Vol. 167, No 2, pp. 820-832, 2005.
5. H. Chien, J. Jan and Y. Tseng, "An Efficient and Practical Solution to Remote Authentication: Smart Card," *Computers & Security*, Vol. 21, No. 4, pp. 372-375, 2002.
6. C. Fan, Y. Chan and Z. Zhang, "Robust Remote Authentication Scheme with Smart Cards," *Computer & Security*, Vol. 24, No. 8, pp. 619-628, 2005.
7. D. Henrici and P. Muller, "Hash-based Enhancement of Location Privacy for Radio Frequency Identification Devices Using Varying Identifiers," *In Proc. of PERSEC'04*, pp. 149-153. *IEEE Computer Society*, 2004.
8. W. Juang, "Efficient Password Authenticated Key Agreement Using Smart Card," *Computer & Security*, Vol. 23, No 1, pp. 167-173, 2004.
9. W. Ku, H. Lee and C. Chen, "Reflection Attack on a Generalized Key Agreement and Password Authentication Protocol," *IEICE Transactions on Communications*, Vol. E87-B, No. 5, pp. 1386-1388, 2004.
10. W. Ku and S. Chen, "Weaknesses and Improvements of an Efficient Password Based Remote User Authentication Scheme Using Smart Cards," *IEEE Transactions on Consumer Electronics*, Vol. 50, No. 1, pp. 204-207, 2004.

- 11.W. Ku and S. Chang, "Impersonation Attack on a Dynamic ID-Based Remote User Authentication Scheme Using Smart Cards," *IEICE Transactions on Communications*, Vol. E88-B, No. 5, pp. 2165-2167, 2005.
- 12.M. Kumar, "New Remote User Authentication Scheme Using Smart Cards," *IEEE Transactions on Consumer Electronics*, Vol. 50, No. 2, pp. 597-600, 2004.
- 13.T. Kwon, Y. Park and H. Lee, "Security Analysis and Improvement of the Efficient Password-Based Authentication Protocol," *IEEE Communication Letters*, Vol. 9, No. 1, pp. 93-95, 2005.
- 14.L. Lamport, "Password Authentication with Insecure Communication," *Communications of ACM*, Vol. 24, No. 11, pp. 770-772, 1981.
- 15.T. Le, M. Burmester and B. Medeiros, "Forward-secure RFID Authentication and Key Exchange," *IACR Eprint*, 2007. <http://eprint.iacr.org/2007/051.pdf>.
- 16.S. Lee, H. Kim and K. Yoo, "Improvement of Chien et al.'s Remote User Authentication Scheme Using Smart Cards," *Computer Standards and Interfaces*, Vol. 27, No. 2, pp. 181-183, 2005.
- 17.NIST FIPS PUB 46-3, "Data Encryption Standard (DES)," National Institute of Standards and Technology, U. S. Department of Commerce, Reaffirmed, 1999.
- 18.NIST FIPS PUB 180-2, "Secure Hash Standard," National Institute of Standards and Technology, U. S. Department of Commerce, Draft, 2004.
- 19.NIST FIPS PUB 197, "Announcing the Advanced Encryption Standard (AES)," National Institute of Standard and Technology, U. S. Department of Commerce, 2001.
- 20.X. Wang, W. Zhang, J. Zhang and M. Khan, "Cryptanalysis and Improvement on Two Efficient Remote User Authentication Scheme Using Smart Card," *Computer Standards & Interfaces*, Vol. 29, No. 5, pp. 507 - 512, 2007.
- 21.S. Weis, S. Sarma, R. Rivest and D. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," In *Security in Pervasive Comp.*, Vol. 2802 of LNCS, pp. 201-212, 2004.
- 22.Y. Yang, S. Wang, F. Bao, J. Wang and R. Deng, "New Efficient User Identification and Key Distribution Scheme Providing Enhanced Security," *Computers and Security*, Vol. 23, No. 8, pp. 697-704, 2004.
- 23.M. Burrow, M. Abadi and R. Needham, "A Logic of Authentication," *ACM Transaction on Computer Systems*, Vol. 8, No. 1, pp. 18-36, 1990.
24. R. Rivest, "The MD5 Message-Digest Algorithm," *IETF RFC 1321*, April, 1992.