International Journal of Advances in Electronics and Computer Science, ISSN: 2393-2835 Volume-4, Issue-9, Sep.-2017
http://iraj.in

AN EFFICIENT AUTHENTICATED KEY AGREEMENT PROTOCOL BASED ON GEOMETRIC APPROACH

[1]REN-JUNN HWANG, [2]HSIU-PING KANG

[1,2]Department of Computer Science and Information Engineering, TamKang University, Tamsui, Taipei, Taiwan
Email: [1]junhwang@ms35.hinet.net, [2]691190259@s91.tku.edu.tw

Abstract - This paper proposed an efficient strong password authenticated key agreement protocol based on the straight-line property of geometry. The proposed scheme does not perform public key cryptographic functions while most of the previous strong password authenticated key agreement schemes based on them. All of the operations used in the proposed scheme are more efficient than the modular exponentiation which is the kernel operation of the public key cryptographic functions. The proposed scheme is more efficient than the previous research results. It withstands the unknown key share attack and password guessing attack, although clients select weak password. It provides forward secrecy, known-key security and robust characteristic. The proposed scheme is categorized to be a bilateral commitment mode which is defined in IEEE P1363.2 / D13. It is secure. The proposed scheme provides best solution with results of efficiency, security and functionalities.

Keywords - Strong Password Authentication, Key Agreement,Secure Communications

I. INTRODUCTION

Identity authentication and confidentiality are primary services of secure communications. It is important to authenticate each other when the client and the server want to communicate through the open network [5, 6, 7]. How to protect the confidential data transmitted between the client and server is also an important issue. Authenticated key agreement protocol is a good solution to provide authentication and confidentiality services. By this protocol, the client and server not only authenticate each other but also generate a secret session key. They can use the session key and cryptosystem to protect the confidential transmitted data. Password-based mechanism is the most widely used method for user authentication since it allows people to choose and keep password by himself. However, people are used to choose easy-to-remember passwords. These weak passwords are vulnerable to password guessing attacks because of limited password spaces are searchable and low-entropy. Jablon [4] considers the protocols should be designed to protect weak-willed and weak-minded clients. Authenticated key agreement protocol based on weak password is referred to as strong password authenticated key agreement protocol. IEEE P1363.2 / D13 [3] regulates its definition and security requirements. In general, the strong password authenticated key agreement protocol considers the scenario in which there are two entities: a client A and a server B; A holds a weak password. These two parties would like to engage in a conversation at the end of which each one generates a secret common session key that is known to nobody but the two of them.

The strong password authenticated key agreement protocol must satisfy many security properties. SRP-1 [11] strengths the requirements further as follows:

1. Revealing useless information about password or secret key to attacker in communication for preventing attacker can able to guess and verify.
2. Revealing useless information about session key to attacker, since session key is a strong cryptographic key instead of a low-entropy password to protect the transmitted data.
3. Even if an attacker has the ability to interfere with clients and server, the protocol should prevent the attacker to get resources from server or learn any information about passwords or session keys. At worst, the attacker only causes the protocol to fail between two parties.
4. If the server's verification table is captured or stolen, the attacker is still unable to impersonate the client; but for mutual authentication reason, the attacker is also unable to impersonate server (termed stolen-verifier attack).
5. Revealing the password to an attacker, he does not obtain the session key of past sessions (forward secrecy).
6. Even though attacker steals session key, he does not carry out a brute-force guessing attack on password (forward secrecy).

A strong password authenticated key agreement protocol with above properties is robust. Bellovin and Merritt first proposed the strong password authenticated key agreement protocol based on pre-shared weak password known as Encrypted Key Exchange (EKE) [1] by combining symmetric and public-key cryptography. There are a series of protocols proposed after EKE [1]. SPEKE [4] is based on Diffie-Hellman key exchange technology (DH for short). Zhang [16] showed SPEKE [4] is susceptible to off-line password guessing attack.

We proposed a verifier-based protocol known as Secure Remote Password (SRP-1) [11] based on DH

that the server's verification table stored the static-verifier instead of password. The advantage of verifier is that it provides a high-entropy than that of password. SRP-1 [11] cannot defeat stolen-verifier attack. Wu proposed SRP-3[12] later on. Since SRP-3 [12] is vulnerable to two-for-one guessing attack and message ordering problem, SPR-6 [13] is then presented. However, in SRP-6 [13], the attacker can pose as server by stealing verifier and then to cheat client's secret information. It is also insecure against unknown key share attack [3] and user table insert / update attack [10]. Yeh et al. proposed a simple authenticated key agreement protocol resistant to password guessing attacks called SAKA [14] which is based on DH. Lee et al. [8] improved the efficiency of the SAKA [14] using parallel computation technology called parallelizable SAKA (PSAKA).

However, both of SAKA [14] and PSAKA [8] are vulnerable to undetectable on-line password guessing attack, and unknown key share attack. Different from above protocols that are based on DH, Yeh et al. [15] proposed a protocol, which combined symmetric, RSA cryptography and suitable imbalanced network. However, as Yeh et al. [15] uses insecure parameter (the public key e of RSA is 3) for imbalanced network, it is unable to defeat off-line password guessing attack and user table insert / update attack. In a word, some of these research results had been considered as unsecured in the related papers, and some do not satisfy the strong definition. It is clearly that all of these research results are based on public-key cryptography. Most public-key functions such as DH or RSA have to perform modular exponentiation, they consume higher computational cost.

This paper proposes a new efficient strong password authenticated key agreement protocol based on the straight-line property of geometry. All of the operations included in the proposed scheme take less computation cost than modular exponentiation. It takes lower computation cost. The clients are allowed to use weak passwords. The proposed scheme conforms to forward secrecy, robust [11] and the bilateral commitment mode. The bilateral commitment mode is more secure than both safe unilateral commitment mode and unilateral commitment mode [3]. These three modes are defined in IEEE P1363.2 / D13 [3].Furthermore, it includes the advantages of all previous schemes. The proposed scheme provides best solution with results of efficiency, security and functionalities.

The remainder of this paper is organized as follows. Section 2 shows the proposed scheme. Then, the security of the proposed scheme is demonstrated in Section 3. Section 4 will discuss and compare performance, security and functionalities of proposed scheme with pervious schemes. Finally, the conclusion is made in Section 5.

II. THE PROPOSED SCHEME

The proposed strong password authenticated key agreement protocol does not include Diffie-Hellman key exchange functions while most of the related scheme based on it. It is more efficient and secure than the previous results. There are two entities: server and clients in the strong password authenticated key agreement protocol. The server maintains a verification table that stored clients' verifier. The verifier will be altered after each legal login process for security consideration. The server keeps a secret key. Clients keep devices such as smart card that stores parameters for authentication. Table 1 defines the notations that are used in this paper.

A	the identification code of Client A
B	the identification code of Server B
q	a large prime number
d	the secret key of Server B
d_A	the important secret value for Client A generated by Server B
v, v_i	verifier, v_i is the verifier for the ith login
V_A	the information of Client A, which is stored in server's verification table
P	the password of client
N_1, N_2, N_3	random numbers, $1 \le N_1, N_2, N_3 < q$
f(·)	a straight line function, L: $y = f(x) = ax + b \bmod q$
h(·)	a strongly collision-resistance hash function, such as SHA-2
T	timestamp
K	session key
$E_K(\cdot)$, $D_K(\cdot)$	symmetric encryption/decryption functions with a secret key K
$\oplus$	bitwise eXclusive OR operator, XOR for short
|m|	the binary bit length of m

Table 1. Notations

The proposed scheme contains two phases, the registration phase and the login phase. In the registration phase, a new client chooses his identification code and password to register and obtains a smart card from the server. In the login phase, the client and server authenticate each other and exchange data to generate a common session key. These two phases are described in the following subsections.

2.1. Registration Phase

When a new client wants to login to Server B, the client must perform the following steps to register at Server B beforehand. Firstly, the new client chooses

his own identity A and password P, and passes to Server B at the time T secretly. Server B generates parameters through the following steps and stores them into smart card to hand over to Client A.

Step R1. Computes $d_A = h(A \oplus d)$, $v_1 = h(P \oplus T)$, $V_A = d_A \oplus v_1$,where v_1 is Client A's initial verifier, and stores V_A in the verification table.

Step R2. Computes $C = d_A \oplus P$ and $D = h(v_1) \oplus P$ and saves {A, C, D} into smart card.

Step R3. Server B delivers the smart card to Client A.

Client A is allowed to login to Server B by using the smart card which keeps the authenticated parameters secretly.

2.2. Login Phase

In this phase, clients and Server B authenticate each other and exchange data to generate a shared session key K. They perform key confirmation operations to assure the session key. Assuming a registered clientA wants to login to Server B at ith time, A inserts the smart card and inputs password P. Server B and Client A perform the following steps to complete the login process.

Step L1. Client A inputs password P and computes $d_A = C \oplus P$, $h(v_i) = D \oplus P$.

Step L2. Client A chooses two random numbers N_1 andN_2. He also constructs a straight line L based on (N_1, N_2) and $(d_A, h(v_i))$. Here, L: $y = f(x) = ax + b \bmod q$, where $a = ((N_2 - h(v_i)) / (N_1 - d_A)) \bmod q$, $b = (h(v_i) - d_A \cdot a) \bmod q$, moreover $a \neq 0$ and $b \neq 0$.

Step L3. Client A calculates $C_1 = a \oplus d_A$, $C_2 = b \oplus h(v_i)$ and $v_{i+1} = f(h(v_i))$. Next, computes $R_1 = h(A, v_{i+1}, a)$, $h(\cdot)$ is a strongly collision-free hash function. Finally, Client A sends $\{A, C_1, C_2, R_1\}$ to Server B.

Step L4. Server Bretrieves Client A's V_A from verification table and computes the following parameters:

$d_A = h(A \oplus d)$, $v_i = V_A \oplus d_A$,

$a = C_1 \oplus d_A$,

$b = C_2 \oplus h(v_i)$,

and reconstructs the line L: $y = f(x) = ax + b \bmod q$. Later, Server B generates $v_{i+1} = f(h(v_i))$. Using the hash function to compute $h(A, v_{i+1}, a)$ and checking whether it equals to R_1. If the equation is false, Server Brejects Client A's login request; otherwise, Server B confirms the identity of Client A.

Step L5. Server B chooses a random number N_3. Then computes $R_2 = h(B, v_{i+1}, b)$ and $C_3 = E_{R_2}(N_3, h(N_3))$. Server B sends $\{B, C_3\}$ to Client A.

Step L6. Client A calculates $R_2 = h(B, v_{i+1}, b)$ where v_{i+1} and b are generated in Step L3, and decrypts C_3 to get N_3' and $h(N_3)$ with key R_2, i.e. $D_{R_2}(C_3)$. Next, Client A uses hash function to compute $h(N_3')$ and check whether it equals to $h(N_3)$. If the equation is false, Client Acan not assure himself that it is Server B and hewill abort the connection; otherwise, Client Aassures himself that it isServer B and believes Server B has a proper value v_{i+1}. Finally, Client A sends the message $\{h(N_3')\}$ to Server B.

Step L7. Upon receiving $h(N_3')$. If it is incorrect, Server Bcan not confirm Client A in producing the common session key, then Server B denies the login request; otherwise Server B believes Client A gets the correct N_3 and the same value v_{i+1}.

Step L8. Client A and Server B generate the share session key $K = N_3 \oplus v_{i+1}$ separately.

Step L9. Client A updates D stored in smart card with $h(v_{i+1}) \oplus P$. Server B updates Client A's V_A with $d_A \oplus v_{i+1}$ in the verification table.

In Steps L4 and L6, both Client A and Server B confirm the identity of each other at first and then exchange the parameter of the session key. They perform the key confirmation operations in Steps L5, L6 and L7 to confirm that theother party actually possess the session key K. We conclude the proposed scheme is a mutual authentication mechanism and conforms to the bilateral commitment mode defined in IEEE P1363.2 / D13 [**Error! Reference source not found.**].

In addition, the proposed scheme provides clients to change the password themselves without connecting to Server B. The step of changing password is to change $C = d_A \oplus P \oplus P \oplus P'andD = h(v_i) \oplus P \oplus P \oplus P'$ stored in smart card, where P is theold password and P′ is the new password. That is Server B is not to know clients' password.

III. SECURITY ANALYSIS

Strong password authenticated key agreement protocol should be satisfy security requirements and defeat some possible attacks. Especially, password guessing attack is always considered as the most vulnerable one and it caused some past proposed schemes to be considered as insecure. Moreover, the scheme should provide forward secrecy. This section demonstrates the proposed scheme withstands these attacks.

3.1. Stolen-Verifier Attack

In the proposed scheme, the client's verifier is generated by a randomly straight line function in each login process. After each login successfully, the server will update the client's verifier, which is protected by server's secret key d with V_A ($= d_A \oplus v_i$; $d_A = h(A \oplus d)$). Suppose the adversary intrudes Server B and steals Client A's V_A, according to XOR characteristic, the probability to compute v_i without d_A is $2^{-|v_i|}$.

The adversary without corrected $v_i$can notfigure out suitable C_1, C_2, and R_1 to pass through server's examinationin Step L4, so the adversary can not impersonate Client A to finish the ith login process. The adversary also cannot masquerade Server B

although he stole the verifier, because hestill cannot construct the correct straight line L to compute C_3 and pass through Client A's examination in Step L6.

3.2. Replay Attack

The adversary fails in impersonating Client A to finish the (i + 1)th login process by replaying Client A'sith login message {A, $C_{1,i}$, $C_{2,i}$, $R_{1,i}$}. Server B verifies the equation h(A, v_{i+2}, a) = $R_{1,i}$ in Step L4 of the (i + 1)th login process. It is false because the values v_i, v_{i+1}, and v_{i+2} are dependent upon three randomly unrelated straight line functions individually. The adversary can not replay previous login message for impersonating Client A to login Server B.

3.3.Denial Of Service Attack (Dos Attack)

The DoS attack [**Error! Reference source not found.**] is an attack that causing the legitimate client to bedenied by server. If the adversary modifies the transmitted message when a legitimate client changes password or verifier, then the server or the legitimate client will update the wrong password or verifier. The legitimate client will fail to login server. We will demonstrate the proposed scheme defeats DoS attack. There are three transmitted messages in the login phase. To begin with, assumingthe adversary modifies the first transmitted message {A, C_1, C_2, R_1} which Client A sent, that bring Server B to get two different values a′ and b′ in Step L4, and then constructs the different straight line function f′(·). However, the probability of two different straight line functions map the same output v_{i+1} is $2^{-|q|}$. It is very low on condition that q is large enough. Moreover, it should satisfy the verifying equation h(A, v_{i+1}, a′) = R_1 in Step L4. According to strongly collision-free hash function properties, it is computationally infeasible that the adversary can find two distinct messages that will hash to the same value. Others, if the adversary modifies the second transmitted message {B, C_3} or third {h(N_3′)}. Client A in Step L6 or Server B in Step L7, they will check whether the equation is true or false by hash function; by the strongly collision-free hash function properties, it is computationally infeasible that the adversary can make two distinct messages to the same hash-value to pass their checks. By the three cases above, the successful probability of adversary's actions is very low that is negligible. And the verifier in clients or server will not be updated improper. Thus, the proposed scheme defeats the DoS attack.

3.4. Password Guessing Attack

Ding [**Error! Reference source not found.**] divided password guessing attack into three classes: detectable on-line password guessing attack, undetectable on-line password guessing attack and off-line password guessing attack. In particular, off-line password guessing attack is considered as the heavy threat. According to these classes, we will show the proposed scheme against the attacks even if clients choose weak password.

It is clearly that our scheme can detect mistakes in each transmitted message described in DoS attack. The proposed scheme is easy to distinguish between a legitimate party and the adversary. The proposed scheme defeats the detectable and undetectable on-line password guessing attacks. We will demonstrate the scheme can also defeat the off-line password guessing attack in the following paragraph.

If the adversary collects the login messages transferred between Client A and Server B, the messages are {A, C_1, C_2, R_1} and {B, C_3}; in which, $C_1 = a \oplus d_A$, $C_2 = b \oplus h(v_i)$, $R_1 = h(A, v_{i+1}, a)$ with $v_{i+1} = f(h(v_i))$, $C_3 = E_{R_2}(N_3, h(N_3))$ where $R_2 = h(B, v_{i+1}, b)$. The values C_1, C_2, R_1 and C_3 are all included random numbers. Especially, these parameters are independent of the client's password. The adversary will fail to guess or learn the password from the transmitted message of the proposed scheme. Therefore, although our scheme allows clients to choose weak password, it still can defeat the heavy threat: off-line password guessing attack.

3.5. Forward Secrecy

SRP-1 [**Error! Reference source not found.**] defines forward secrecy in two meanings, we examines our scheme as follows:

1. The attacker can not get any information of the previous session keys by revealing Client A's password.

Assuming the adversary gets Client A's password P and smart card which stored C and D inside. The adversary will get d_A and h(v_i) by $d_A = C \oplus P$ and h(v_i) = D ⊕P. Since h(·) is a strongly collision-free hash function with irreversible, the adversary can not figure out v_i from h(v_i), where v_i is the verifier for next (ith) login. At the same time, the adversary also records (i - 1)th login message as {A, $C_{1,i-1}$, $C_{2,i-1}$, $R_{1,i-1}$} and {B, $C_{3,i-1}$}, he will get $a_{i-1} = C_{1,i-1} \oplus d_A$,but he can not get h($v_{i-1}$) from h($v_i$), and b_{i-1}. The probability of figuring out b_{i-1} is $2^{-|q|}$ that considerably guessed directly. That is to say the adversary can not construct the correct straight line function $f_{i-1}(\cdot)$ without getting b_{i-1} and h(v_{i-1}). He would fail to compute $R_{2,i-1}$anddecrypt $C_{3,i-1}$ to get $N_{3,i-1}$. Without $N_{3,i-1}$ and v_i, the adversary can not obtain (i- 1)th session key K (= $N_{3,i-1} \oplus v_i$). In the same situation, the adversary is hard to derive the session keys of (i- 2)th, (i- 3)th etc. by revealing password and getting smart card. Thus, our scheme provides forward secrecy.

2. Although the adversary stole the session key, it cannot assist him to get the password by the brute-force guessing attack.

The session key K (= $N_3 \oplus v_{i+1}$) was generated after Client A's ith login, where N_3 is a random number and v_{i+1} is calculated by a random straight line function f(·). It is clear that the session key K is

composed of two random factors, and does not include any information of password P and any verifiable data. Hence, the adversary is purely able to guess password by brute-force attack even thought session key has been stolen. Therefore, our scheme conforms to this definition.

By the above descriptions, our scheme satisfies the property of forward secrecy.

3.6. Known-Key Security

Known-key security is that an attacker compromises the session key, but he can not compromise future session keys based the compromised key. Moreover, he can not control the value of the future session keys. The objective of known-key security is to protect other sessions and limit damage only in the compromised session, which caused by the compromised session key.

Suppose the (i+ 1)th session key K_{i+1} is equal to $N_{3,i+1} \oplus v_{i+2}$ and the attacker gets the ith session key $K_i (= N_{3,i} \oplus v_{i+1})$ of the proposed scheme. However, Client A separately computes v_{i+1} and v_{i+2} from two unrelated straight lines L_i and L_{i+1} in Steps L3 of ith and (i+ 1)th sessions. L_i and L_{i+1} are randomly generated in Steps L2 of ith and (i+ 1)th sessions separately by Client A. $N_{3,i}$ and $N_{3,i+1}$ are randomly selected by Server B in Steps L5 of ith and (i+ 1)th sessions separately. The attacker is hard to decide $N_{3,i+1}$ and v_{i+2} to generate K_{i+1}, although he compromised K_i. Thus, the attacker cannot compute or control the future session keys based on the compromised session key.

3.7. Unknown Key Share Attack

There are two points of the unknown key share attack in IEEE P1363.2 / D13 [**Error! Reference source not found.**]. One is that Client A uses the same password P with Server B and B′, the adversary redirects the login messages to Server B′ when Client A sends login messages to Server B. The client does not correctly identify the target party and causes two wrong parties to share session key. Client A considers that he shares the key with Server B but actually with Server B′. Another is that both Clients A and C use the same password P with Server B. The adversary replaces the identification A of the login message with C when Client A sends the login messages to Server B. The server does not correctly identify theclient. Server B considers that it is talking with Client C but actual with Client A. According to these two points, we show that the proposed scheme withstands the unknown key share attack.

1. The adversary redirects Client A's login messages to Server B′

In this case, the adversary redirects Client A's login messages {A, C_1, C_2, R_1} to Server B′, where $C_1 = a \oplus d_A$, $C_2 = b \oplus h(v_i)$ and d_A is generated by Server B for Client A ($d_A = h(A \oplus d)$, d is the secret key of Server B). Server B′ receives Client A's login messages and computes $d_A' = h(A \oplus d')$ where d′ is thesecret key of Server B′ in Step L4. Server B′ gets $a' = C_1 \oplus d_A'$ and $b' = C_2 \oplus h(v_i')$, andv_i' is fetched from verification table in Server B′. As the results of $a' \neq a$,and $b' \neq b$, the equation $h(A, v_{i+1}, a) = R_1$ is false in Step L4, and Server B′ will abort the connection. The adversary will pass this checking on condition that $d = d'$, and $v_i = v_i'$, but the probability is $2^{-|d|} \times 2^{-|v_i|}$. It is too small to negligible.

2. The adversary replaces the identification of Client A with Client C

When Client A sends the login messages {A, C_1, C_2, R_1} to Server B, the adversary attempts to make Server B to consider that it's Client C by intercepting and replacing Client A's validity messages with {C, C_1, C_2, R_1}. After Server B receivingthe messages, it computes $d_C = h(C \oplus d)$, $v_{i,C} = V_C \oplus d_C$, $a' = C_1 \oplus d_C$and $b' = C_2 \oplus h(v_{i,C})$, but $a' \neq a$, and $b' \neq b$. Server B will construct another straight line L′: $y' = f'(x) = a'x + b'$ mod q and $v_{i+1,C} = f'(h(v_{i,C}))$. However, the equation $h(C, v_{i+1,C}, a') = R_1 (= h(A, v_{i+1,A}, a))$will be false in Step L4. Server B detects this attack,and the adversary failed. The adversary will achieve the goal on condition that $d_C = d_A$, and $V_C = V_A$, the probability is $2^{-|d_C|} \times 2^{-|V_C|}$. By the strongly collision-free hash function, it is computationally infeasible that the adversary could find two distinct values (A, C) with the same hash-value. Therefore, our scheme has ability to defeat the attack.

Clearly, the proposed scheme does securely identify the intended target.

3.8. User Table Insert / Update Attack

Wang [**Error! Reference source not found.**] points out that the verification table, whichstores the password kept in plaintext or hashed, is easy to retrieve or update client's data. It is not secure. Assuming the adversary can access the verification table of Server B, and he wants to add an illegal client C into the verification table for login to get resources. The adversary has to generate some parameters as Server B in the "registration phase", he chooses password P and computes $v_1 = h(P \oplus T)$, where T is the registering time. However, the adversary does not know Server B's secret key d, he cannot compute the suitable values $d_C (= h(C \oplus d))$ and $V_C (= d_C \oplus v_1)$. Therefore, even though the adversary has ability to access the verification table, he also cannot generate thecorrect information. The adversary mighteven write incorrect value V_C', whichwill be detected in Step L4. Hence, our scheme provides a secure verification table to prevent user table insert / update attack.

CONCLUSIONS

This paper proposed an efficient strong password authenticated key agreement protocol based on the

straight-line property of geometry. Clients allowed choosing weak password,but the scheme still can defeat password guessing attacks. The proposed scheme satisfies the definition of strong scheme, robust [**Error! Reference source not found.**], bilateral commitment mode [**Error! Reference source not found.**] and security requirements such as forward secrecy, known-key security and unknown key share attack etc. The characteristics of the proposed schemeare usingdynamic-verifier and changing password without connecting to server. The proposed scheme doesnot perform modular exponentiation computations. In comparison with previous well-known protocols [**Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.**], the proposed scheme is more efficient and secure.

REFERENCES

[1] Bellovin,S.M. and Merritt, M.,Encrypted key exchange: password-based protocols secure against dictionary attacks.IEEE Symposium on Research in Security and Privacy, 72-84 ,1992.

[2] Ding, Y. and Horster, P.,Undetectable on-line password guessing attack.ACM Operating System Review, 29(4), 77-86, 1995.

[3] IEEE P1363.2 / D13,Standard Specifications for Password-based Public Key Cryptographic Techniques.IEEE P1363 working group, 2004.

[4] Jablon, D. P., Strong password-only authenticated key exchange.ACM SIGCOMM Computer Communication Review, 26, 5-26, 1996.

[5] Moxie Marlinspike and Trevor Perrin,The X3DH Key Agreement Protocol,Open Whisper Systems,2016.

[6] Areej Omar Baalghusun, Olfa Fahad Abusalem, Zahra Abbas Al Abbas, JayaprakashKar,Authenticated Key Agreement Protocols: A Comparative Study.Journal of Information Security, 6, 51-58 , 2015.

[7] N. Unger and I. Goldberg, Deniable Key Exchanges for Secure Messaging, Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 1211-1223, 2015.

[8] Lee,S. W., Kim, W. H., Kim, H. S., Yoo, K. Y.,Parallizable simple authenticated key agreement protocol.ACM SIGOPS Operating Systems Review, 37, Issue 3, 17-22, 2003.

[9] Lin,C.L.,Sun, H.M., and Hwang, T., Attacks and solutions on strong-password authentication.IEICE Trans. Commun., E84-B, No.9, 2622–2627, 2001.

[10] Wang, S. J.,Remote table-based log-in authentication upon geometric triangle.Computer Standards and Interface, 26, Issue 2, 85-92 , 2004.

[11] Wu,T.,The secure remote password protocol.Proceedings of the Internet Society Network and Distributed System Security Symposium, March, pp. 97-111, 1998.

[12] Wu,T.,The SRP authentication and key exchange system.Stanford University. Request For Comments (RFC) 2945 , 2000.

[13] Wu, T., SRP-6: improvements and refinements to the Secure Remote Password Protocol. Submission to IEEE P1363 Working Group , 2002.

[14] Yeh,H.T. and Sun, H.M., Simple authenticated key agreement protocol resistant to password guessing attacks.ACM Operating Systems Review, 36, Issues 4 , 2002.

[15] Yeh, H. T., Sun, H. M., Yang, C. T., Chen, B. C., and Tseng, S. M., Improvement of password authenticated key exchange based on RSA for imbalanced wireless networks.IEICE TRANS. COMMUN., E86-B, No.11 , 2003.

[16] Zhang, M., Analysis of the SPEKE password-authenticated key exchange protocol.IEEE Communications letter, 8, No.1 , 2004.

★★★