JCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-6, June 2019 E-ISSN: 2347-2693

Smart Card Based Password Authenticated Key Agreement in Distributed Systems

Pritaj Yadav[1*], **Sitesh Kumar Sinha**[2], **S. Veenadhari**[3]

[123]Department of Computer Science & Engineering, Rabindranath Tagore University, Bhopal, India

[*]*Corresponding Author: yadavpritaj@gmail.com, Phone:9827454903*

DOI: https://doi.org/10.26438/ijcse/v7i6.99104 | Available online at: www.ijcseonline.org

Accepted: 15/Jun/2019, Published: 30/Jun/2019

Abstract- The distributed system protocol based on single password is old fascinated, nowadays security issues are important aspects for any kind or sort of system protocol. As per the security need key exchange protocol which shows better security and provide greater convenience i.e. smart card and key security password for the key exchange system protocol in data sharing distributed systems. The present study proposes a general architecture construction of smart card and key security password for the key exchange system protocol in data sharing distributed systems. The present study introduces the combinatorial method of password authentication key exchange (PAKE) without public key. This constructed architecture has additional exchange phase as compare to the scheme for the public encryption (original). As compared with the protocols used in distributed system, the proposed architecture construction shows great properties in term of security and quite better computational efficiency it means operation time is less and low cost at storage.

Keywords: *Smart Card, PAKE, TWO-PAKE, General Architecture Construction, Distributed System.*

I. INTRODUCTION

Password-Authenticated Key Exchange (PAKE) allows two communication entities to certify one another and established a session key via simply memorable passwords. The primary PAKE protocol was introduced by Bellovin and Merritt in 1992 referred to as Encrypted Key Exchange (EKE).

Two-party password-based attested key exchange (two-PAKE) protocol is comparatively helpful for client-server architectures. However, in large-scale client-client communication environments wherever a user must communicate with a great deal of different users, Two-PAKE protocol is incredibly tough in key management that the amount of passwords that the user would wish to remembers. A multilateral password-based key transfer protocol using server’s public key. Later, a multilateral PAKE (three-PAKE) protocol between two clients while not server’s public key.

Security in computers is the key aspect now a days. Data access from unauthorized or accidental persons is threat for security systems. Authentication protocols gives two entities to know the counterparty to anticipate who makes an attempt to communicate with over a diffident network. These protocols are often thought of from three dimensions: sort, potency and security.

In general, there are two forms of authentication protocols, the password-based and public key based. During a password primarily based protocol, a user registers his account and password to a remote server. Later, he will access the remote server if he will prove his information of the password. The server sometimes maintains a password or verification table however this can build the system simply subjected to a stolen-verifier attack. To deal with this drawback, recent studies recommend an approach with no password or verification table within the server. Moreover, to reinforce password protection, recent studies conjointly introduce a tamper-resistant smart card within the user end. During a public key-based system, a user ought to register himself to a trust party, named KGC (Key Generation Center) to get his public key and corresponding non-public key. Then, they'll be recognized by a network entity through his public key. To modify the key management, an identity-based public-key cryptosystem is sometimes adopted, during which KGC problems user is ID as public key and computes corresponding non-public key for a user.

The section-I discusses the introductory part of the paper, section II discusses the recent implements , their results, and projected methodology, the section III discusses results and their comparison depict. Section IV & V discusses the Conclusion.

II. RECENT DEVELOPMENTS

Zhang Gegei et al. [1]: In this paper proposed a general construction for KE protocols using smart card and password. The KE protocols generated can be used in various public key environments as a basic module. This new construction also satisfies the AKE security mentioned by Bellare, so that it can resist several attacks including off-line dictionary attack, while many other protocols can’t. Applying this construction to the Die-Hellman integrated encryption scheme (DHIES) mentioned by M. Abdalla et al. AKE protocol can be obtained, which has not only better security properties, but also better computational efficiency in storage cost and operation time.

Qi Xie et al. [2]: In this paper, planned an Anonymous Two-Factor AKE theme that preserves security against varied attacks together with de-synchronization attack, lost-smart-card attack and password estimation attack, and supports many fascinating properties together with excellent forward secrecy, obscurity or un-traceability, adaptively password modification, no centralized password storage, and no long public key. Furthermore, our protocol maintains high efficiency in terms of storage demand, communication value also as process complexness. Our protocol needs solely a couple of or some or many or number of message flows and every one the transmitted messages are short in size. Additional, the planned theme is incontrovertibly secure in our extended security model of AKE. Therefore, the planned theme is appropriate for readying in varied low-power networks, specially, the pervasive and mobile computing networks.

Hung-Min Sun et al. [3]: Planned a shoulder- surfing resistant authentication system supported graphical passwords, named Pass-Matrix. Employing a one-time login indicator per image, users will signifies the placement of their pass-square while not directly clicking or touching it that is associate action prone to shoulder surfing attacks. as a result of the look of the horizontal and vertical bars that cowl the complete pass-image, it offers no clue for attackers to slender down the word area notwithstanding they need quite one login records of that account. Moreover, we tend to enforce a Pass-Matrix model on mechanical man and dispensed user experiments to judge the memorability and usefulness. The experimental result showed that users will log into the system with a mean of 1:64 tries (Median=1), and also the Total Accuracy of all login trials is 93.33% even fortnight when registration. The overall time consumed to log into Pass-Matrix with a mean of 3:2 pass-images is between 31:31 and 37:11 seconds and is taken into account acceptable by 83.33% of participants in our user study.

R. Madhusudan et al. [4]: In this paper, presented the cryptanalysis of Wen and Lis an improved dynamic ID based remote user authentication scheme with key agreement, and identify its vulnerability. To overcome the security problems, we proposed improved scheme. Through security analysis, we have explained that, our scheme gives protection from all pointed weaknesses. By performance analysis, we compare the computation cost of our scheme with Wen and Li's scheme and illustrated that our scheme reduces 6 hash function, than their scheme. Hence our scheme is more efficient, particularly for user privacy, amplified security and low computation capability.

Zheng Xian Gao et el. [5]: In this paper, briefly reviews the recently development of Dynamic ID-based user authentication scheme using smart card. Then, they have reported security vulnerabilities on three well-designed remote user authentication schemes (Gao-Tu scheme, 2008; Yeh et al. scheme, 2010; Khan et al. scheme, 2011). Based on their cryptanalysis, Gao-Tu scheme is confronted with some threats including smart card forge attack, impersonation attack and message forge attack; Yeh et al. scheme cannot defend against smart card forge attack, impersonation attack and replay attack; insecure against resembling account attack, session key compromise attack and impersonation attack. In addition, the study demonstrates some interesting issues on dynamic ID-based authentication scheme using smart cards. Furthermore, they have defined all the security requirements and all the goals an ideal password authentication scheme should satisfy and achieve, which is useful for the authentication scheme designers.

III. PROPOSED METHODOLOGY

This section describes two types of attacks on proposed scheme, both of which can successfully uncover the password chosen by the user.

Offline-Dictionary attacks with Smart Cards

The smart card contains the public parameter IM and a private parameter V. As discussed in here, an adversary cannot directly use $V = h(ID||K_S) \oplus h(PW)$ to corrupt the user ’s authentication session. This is due to the fact that V does not provide any useful information about the password PW, if the server’s secret key K_S is selected from a large domain. In other words, the information V alone does not help the adversary to verify the guess of a user’s password. The question arises: With two (or more) Vs generated at different times, whether or not the adversary can uncover the user’s password?

Attacking Scenario.

In this section, we address the attacking scenario as follows.
1) At time T_1, the user invokes the password changing phase to change the password to PW_1.
At the end of this phase, the smart card contains (V_1, IM) where $V1 = h(ID||K_S) \oplus h(PW_1)$.

2) At some time later (say, T_2), the user changes the password $PW_1$to a new password PW_2, and the smart card contains (V_2, IM), where $V2 = h(ID||K_S) \oplus h(PW_2)$.$PW_2$ can be regarded as the current password.

3) A passive attacker with smart card (defined in Section 3.1) can obtain the data in the smart card at time T_1 and T_2. We note that such an adversary is stronger than that considered here, where the adversary can obtain the information in the smart card but only once. If the adversary can capture the information in the smart card once, we believe the adversary can also do it for the second time. As an example, one can obtain the information in the smart card via an illegal card reader. This could occur more than once without the awareness of the smart card owner (e.g., the attacker could steal the smart card and send it back after extracting the data stored in the smart card). In the above attacking scenario, the other assumption is that the user will change the password at least twice? We believe this is also a reasonable assumption as changing password on a regular basis has been regarded as one of good password habits.

This completes the description of the attacking scenario we are concerned about, which we believe falls into the category of passive attacker with smart card defined in Section 3.1. It remains to show how to extract the two passwords (PW_1, PW_2) with (V_1, V_2).

How Does the Attack Work?

(V_1, V_2), the adversary can XOR V_1 and V_2 to obtain the equation

$$V_1 \oplus V_2 = h(PW_1) \oplus h(PW_2).$$

This enables the adversary to verify the guess of PW_1 and PW_2, where PW_1 is an old password at time T_1 and PW_2 is the current password at time T_2.

SUCCESS PROBABILITY.

We now consider the success probability that an adversary can find (PW_1, PW_2) in the above attacking scenario. We will show that the success probability (in general) is at least $1 - \frac{SIZE_{PW}}{SIZE_h}$, where $SIZE_{PW}$ is the size of the password dictionary and $SIZE_h$ is the size of the output domain of the hash function h. In a concrete case, the adversary can find (PW_1, PW_2) with probability almost 1. The detail of our analysis is given as below.

1) By testing all password pairs in the password dictionary, the adversary will find at least one pair (pw_1, pw_2) such that

$$V_1 \oplus V_2 = h(pw_1) \oplus h(pw_2). \qquad (1)$$

2) If there is only one pair satisfying Equation. (1), it must be (PW_1, PW_2) and the adversary thus successfully finds the user's passwords.

3) The adversary, however, could find two or more password pairs using Equation. (1). We now consider the probability that there is only one pair satisfying Equation. (1) in the password dictionary.

4) For any two different passwords pw_1 and pw_2 in the password dictionary, we define

$$\mathcal{E}_1: \{pw_1, pw_2\} \neq \{PW_1, PW_2\} \text{ and}$$
$$\mathcal{E}_2: h(pw_1) \oplus h(pw_2) \neq V_1 \oplus V_2.$$

Then, $P_r[\mathcal{E}_1|\mathcal{E}_2]$ is the probability that there is only one pair (PW_1, PW_2) satisfying Equation. (1), i.e., $P_r[\mathcal{E}_1|\mathcal{E}_2]$ is the adversary's success probability to find (PW_1, PW_2).

5) Let $SIZE_{PW}$ be the size of the password dictionary, and let $SIZE_h$ be the size of the output domain of the hash function h.

6) In the password dictionary,

a) For any password $pw_1 \in \{PW/_1, PW_2\}$, the probability that there is another password pw_2

such that $h(pw_{/2}) = V1 \oplus V2 \oplus h(pw_1)$ is at most $SIZE_{PW}/SIZE_h$ (assuming the output of h is uniformly distributed). Therefore, for any password $pw_1 \in \{PW_1, PW_2\}$, $Pr[\mathcal{E}_2| \mathcal{E}_1] \geq 1 - SIZE_{PW}/SIZE_h$.

b) Otherwise, $pw_1 \in \{PW_1, PW_2\}$ but $pw_2 \in \{PW_1, /PW_2\}$ (since pw_1 and pw_2 are two different passwords, and $\{pw_1, pw_2\}$ _= $\{PW_1, PW_2\}$). We first suppose $pw_1 = PW_1$.

In this case, the probability that there is another password $pw_2 \in \{PW_1, PW_2\}$

such that $h(pw_2) = V_1 \oplus V_2 \oplus h(pw_1) = H(PW_2)$ is at most $SIZE_{PW}/SIZE_h$ (assuming the output of h is uniformly distributed). Similarly, for the other case when $pw_1 = PW_2$, the probability that there is another password $pw_2 \in \{PW_1, PW_2\}$ such that $h(pw_2) = V_1 \oplus V_2 \oplus h(pw_1) = H(PW_1)$ is at most $SIZE_{PW}/SIZE_h$ (assuming the output of h is uniformly distributed). Therefore, for any password $pw_1 \in \{PW_1, PW_2\}$, $Pr[\mathcal{E}_2|\mathcal{E}_1] \geq 1 - SIZE_{PW}/SIZE_h$.

Thus, in either case,

$Pr[\mathcal{E}_2|\mathcal{E}_1] \geq 1 - \frac{SIZE_{PW}}{SIZE_h}$

7) We now consider a concrete case. Let the password dictionary consists of 8-character passwords of digits and mixed-case letters, and the hash function h is SHA-256. In this case,$SIZE_{PW}=62^8=218340105584896$,$SIZE_h=2^{256}=11579208923731619542357098500868790785326998466564056403945758400791312963993 6$,

and $1 - \frac{SIZE_{PW}}{SIZE_h} \approx 1$. In other words, the adversary can find the passwords (PW_1, PW_2) with probability almost 1 in this case.

This completes the analysis of Sun et al.'s scheme under a passive attacker with smart card. We have shown that such an adversary can successfully uncover the passwords chosen by the user with overwhelming probability.

Online-Dictionary attacks with Smart Cards

An adversary with the Smartcard = {IM, V} can also break the protocol in via an online-dictionary attack. We first outline the active attacker with smart card and then provide the detail description.

The adversary first extracts {IM, V} in the smart card. Then, the adversary inserts the card in the card reader and sends a log-in request on behalf of the user (by inputting a randomly chosen password in a password dictionary). After that, the adversary can uncover the user's password using the response from the server. As in offline-dictionary attacks, we assume again that the user's password is chosen by him/her via the password change phase, rather than the initial one selected by the server. In other words, the password is chosen from a human-memorable domain. The detail of the attack is given below.

ONLINE-DICTIONARY ATTACK

1) The adversary first chooses a random number $r_C$from the interval [1, n − 1], and calculates GC = r_c× G.

2) The adversary sends (IM, G_c) to the Server. Here, IM is stored in the
Smartcard = {IM, V}.

3) The decryption of IM will be correct, and the server will respond with {M_s, G_s}, where G_S = r_s× G, r_s is randomly chosen in [1, n − 1], M_S = $h_2(K_{su}||G_c||G_s)$, and
$K_{SU} = h_1(h(ID||K_S)||(r_s\times G_C))$.

4) Upon receiving {M_S, G_S}, the adversary chooses a password pw in the password dictionary and calculates V `= V $\oplus$h(pw), K`= h_1 ($V||r_c\times G_s$). In this case, the adversary (most likely) will not log in successfully, but only one failed log-in attempt will not lead to the lock out of user's account.)

Recall that V is stored in the smart card as well.
5) If the guess of the password is correct, then $M_s = h_2$ (K $||G_c||G_s$).Otherwise, the adversary repeats the calculation at Step 4 with another password in the dictionary.

6) There is only one correct password in the dictionary, assuming the hash function is collision resistant.
7) The adversary, with the correct password, can either proceed with the remaining steps in the authentication phase or invoke the password-change phase.

This completes the analysis of Sun et al.'s scheme under online-dictionary attacks with the smart card. We have shown that an active attacker with smart card can successfully find the user's password using the response from the server. Note that the attacker described above does not log on the server by trying every possible password for a specific user, and is different from the common online-dictionary attacker. More precisely, the adversary described above mounts online-dictionary attacks in a more active way.

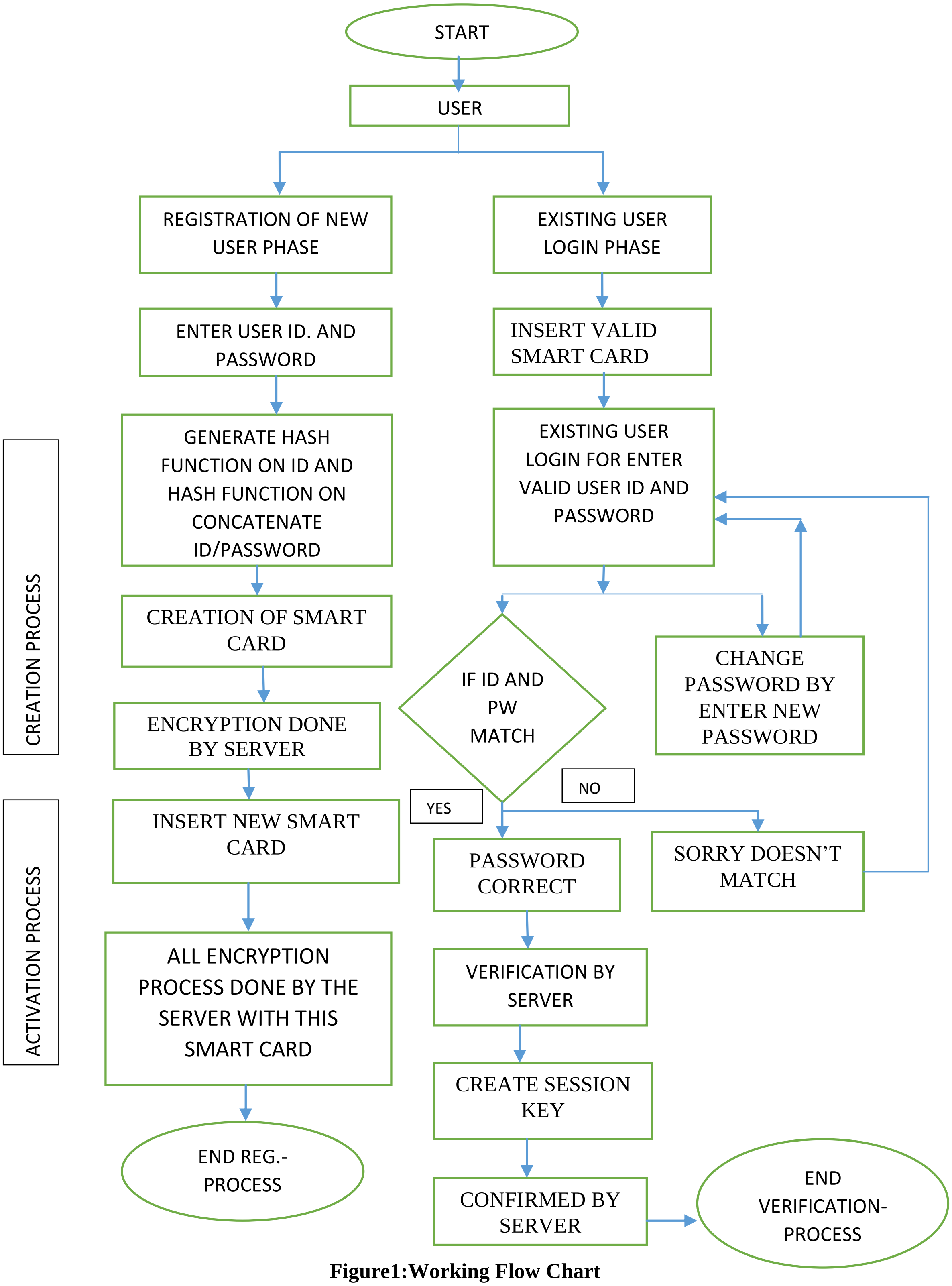
START
USER
REGISTRATION OF NEW USER PHASE
EXISTING USER LOGIN PHASE
ENTER USER ID. AND PASSWORD
INSERT VALID SMART CARD
GENERATE HASH FUNCTION ON ID AND HASH FUNCTION ON CONCATENATE ID/PASSWORD
EXISTING USER LOGIN FOR ENTER VALID USER ID AND PASSWORD
CREATION PROCESS
CREATION OF SMART CARD
CHANGE PASSWORD BY ENTER NEW PASSWORD
IF ID AND PW MATCH
ENCRYPTION DONE BY SERVER
NO
YES
ACTIVATION PROCESS
INSERT NEW SMART CARD
PASSWORD CORRECT
SORRY DOESN'T MATCH
ALL ENCRYPTION PROCESS DONE BY THE SERVER WITH THIS SMART CARD
VERIFICATION BY SERVER
CREATE SESSION KEY
END REG.- PROCESS
CONFIRMED BY SERVER
END VERIFICATION- PROCESS

Figure1:Working Flow Chart

IV. SIMULATION RESULTS

In this section we tend to mention the results associated with our projected methodology. NetBeans we tend to are exploitation Net-Beans IDE 8.1 as a simulation tool. NetBeans is an integrated development platform chiefly for Java, however additionally support to different languages, specifically PHP, C/C++, and HTML5. It’s additionally an application platform framework for Java desktop applications et al. The NetBeans IDE could be a platform within which code written in Java and may run on Windows OS.

Table 1: Prevention from Various Attacks

Attack	**Existing Scheme(attack prevention)**	**Our Scheme(attack prevention)**
Replay attack	**Yes**	**Yes**
Identity-disclosure attack	**Yes**	**Yes**
Insider attack	**No**	**Yes**
Outsider attack	**No**	**Yes**
Eavesdropping	**Yes**	**Yes**
Identity Spoofing	**Yes**	**Yes**
Password based attack	**Yes**	**Yes**
Man-in the middle attack	**Yes**	**Yes**

Table 2: No. of bit used

No. of bits in token	**No. of bits in conceal value**
32	128

Table 3: Storage judgment of the planned scheme

Storage/ scheme	**Existing Work**	**Our scheme**
Smart card	128 bits	256 bits
Server	64 bits	128 bits

Table 4: Comparison of Computation with previous work

Computation Cost		**Existing Scheme**	**Our Scheme**
Smart Card	Registration Operation	-	-
	Session Run	2M+4H	1M+2H
	Password Operation	2H	1H
Server	Registration Operation	2H+1E	1H+1E
	Session Run	2M+4H+1E	1M+2H+1E
	Password Operation	-	-

Where, H denotes the cryptographic hash computation & M denotes the scalar multiplication computation over the elliptic curve & E denotes the symmetric encryption or decryption computation. Table 1 shows comparison of prevention of attack from previous methods. Table 2 shows Time taken and No. of bit used in our scheme. Table 3 shows that comparison of Storage judgment of the planned scheme of our scheme with previous scheme and Table 4 shows the comparisons of computation of all the phases with previous work.

V. CONCLUSION

Simulation results shows that less number of hash functions and elliptical function used to design this system. The present study shows that how to combine the operation of registration and login section (i.e. user ID with PW) with fewer hash functions. The main objective of this paper is to reduce the complexity and cost of the machine. Less complexity and reduced cost has been achieved by implementing this proposed password authentication based smart card scheme. The achieved results shows that the improvement in terms of time complexity, size and cost.

REFERENCES

[1] ZHANG Gefei, FAN Dan, ZHANG Yuqing and LI Xiaowei, “*A Provably Secure General Construction for Key Exchange Protocols Using Smart Card and Password*”, Chinese. Journal of Electronics 2017.

[2] Qi Xie, Duncan S. Wong, Guilin Wang, Xiao Tan, Kefei Chen, Liming Fang,” *Provably Secure Dynamic ID-based Anonymous Two-factor Authenticated Key Exchange Protocol with Extended Security Model*”, IEEE Transaction 2016.

[3] Hung-Min Sun, Shiuan-Tung Chen, Jyh-Haw Yeh and Chia-Yun Cheng, “*A Shoulder Surfing Resistant Graphical Authentication System*”, IEEE Transaction 2016.

[4] R. Madhusudhan and Manjunath Hegde, “*Cryptanalysis and Improvement of Remote User Authentication Scheme Using Smart Card*”, IEEE 2016.

[5] Zheng xianGao, ShouHsuan Stephen Huang, Wei Ding, “*Cryptanalysis of Three Dynamic ID-Based Remote User Authentication Schemes Using Smart Cards*”, IEEE 2016