ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2013, Vol. 42, No. 3

Improvement of a Three-Party Password-Based Key Exchange Protocol with Formal Verification

Qi Xie[1], Na Dong[1], Xiao Tan[2], Duncan S. Wong[2], Guilin Wang[3]

[1] School of Information Science and Engineering,
Hangzhou Normal University, China
e-mail: qixie68@yahoo.com.cn

[2] Department of Computer Science,
City University of Hong Kong, China
e-mail: xiaotan4-c@my.cityu.edu.hk

[3] School of Computer Science & Software Engineering,
University of Wollongong, Australia
e-mail: guilin@uow.edu.au

crossref http://dx.doi.org/10.5755/j01.itc.42.3.1905

Abstract. A Three-Party Password-based Authenticated Key Exchange (3PAKE) protocol allows two users to establish a secure session key over an insecure communication channel with the help of a third party, which is a trusted server. Recently, Lou and Huang proposed a 3PAKE which is efficient and suitable for running on resource-constrained devices such as smart cards and mobile phones. In this paper, we show that their scheme is vulnerable to off-line password guessing attack and partition attack. We then propose an efficient method to fix these problems. Additionally, the mutual authentication and session key secrecy of the proposed protocol are verified using a formal verification tool.

Keywords: key exchange; Password Based Authenticated Key Exchange (PAKE); three-party PAKE; ProVerif.

1. Introduction

Two-party password-based authenticated key exchange protocol was introduced by Bellovin and Merritt [1] in 1992. The protocol allows two parties to authenticate each other via a public, insecure network and establish a secure session key which is to be used for protecting their subsequent communication. However, the protocol is not scalable in a large-scale peer-to-peer system, since every pair of communication parties needs to share a password, so that each party in an *n*-party system has to maintain *n-1* passwords. To solve this problem, Three-Party Password-based Authenticated Key Exchange (3PAKE) protocols were introduced [2-11]. In a 3PAKE protocol, each user only shares a password with a trusted third-party server which gets involved in every session for helping the parties to establish secure session keys. A secure 3PAKE protocol should defend against both passive and active adversaries. One of the well-known attacks against password-based cryptographic protocols is password guessing attack, since users' passwords are usually short and easy to remember. Password guessing attacks can generally be classified into three categories [3]:

(1) Detectable on-line password guessing attack: an attacker tries a possible password on-line each time and determines the correctness of the guessed password by the response from the server. An incorrect password can be detected and logged by the server.

(2) Undetectable password guessing attack: an attacker verifies the guessed passwords through other channels with the server, such that an incorrect password cannot be detected or logged by the server.

(3) Off-line password guessing attack: an attacker verifies the guessed passwords off-line. No participation of the server is required, so the attack cannot be detected by the server.

1.1. Related work

In 2007, Lu and Cao [4] proposed an efficient 3PAKE which is found vulnerable against off-line password guessing attack and man-in-the-middle attack [5-10]. In 2009, Huang [11] proposed another scheme in which the server does not need to have a public key. However, Yoon and Yoo [12] showed that the protocol is vulnerable to undetectable password guessing attack and off-line password guessing attack.

In 2011, Lou and Huang [13] proposed a new 3PAKE protocol which can be implemented on an elliptic curve group, and is suitable for resource-constrained devices such as mobile phones and smartcards. They claimed that the protocol can achieve security against various password guessing attacks.

In this paper, we show that Lou and Huang's scheme is vulnerable to off-line password guessing attack and partition attack. In addition, we propose an improved scheme to solve these problems. The protocol also enjoys low computational complexity and is suitable for resource-constrained devices. There were also several recent schemes proposed in the literature [14-19], however, none of them attempted to give an appropriate solution to the issue above.

1.2. Paper organization

The rest of the paper is organized as follows. In Section 2, we review Lou and Huang's scheme. In Section 3, an off-line password guessing attack and a partition attack against their scheme are described in details. In Section 4, we propose an improved scheme and analyze its security in Section 5. After that, we use the ProVerif tool to prove the mutual authentication and security of the proposed protocol in Section 6. The paper is concluded in Section 7.

2. Review of Lou-Huang 3PAKE protocol

In this section, we briefly review Lou and Huang's 3PAKE protocol [13]. The system chooses a large prime q, an elliptic curve E defined over a finite field F_q, a cyclic group $G = <P>$ of points over E, where P is a generator of E with order n. Suppose that pw_A (resp. pw_B) is the password of the user with identity *A* (resp. *B*) shared with the trusted server *TS*. Let $(d, F = dP)$ be *TS*'s private-public key pair, and $h()$ be a secure hash function, D_x and D_y be the *x*-coordinate and *y*-coordinate of point $D = (D_x, D_y)$. Lou and Huang's 3PAKE protocol is described as follows.

Round 1: User *A* randomly chooses t_a, computes two points $Q_A = t_aP = (Q_{Ax}, Q_{Ay})$, $F_A = t_aF = t_adP = dQ_A$, and $Z_A = (Q_{Ax} \| Q_{Ay}) \oplus h(pw_A, A, B)$. Then $\{A, Z_A, F_A\}$ is sent to *B*.

Round 2: User *B* randomly chooses t_b, computes two points $Q_B = t_bP = (Q_{Bx}, Q_{By})$, $F_B = t_bF = t_bdP = dQ_B$, and $Z_B = (Q_{Bx} \| Q_{By}) \oplus h(pw_B, A, B)$. Then *B* sends $\{A, Z_A, F_A, B, Z_B, F_B\}$ to *TS*.

Round 3: Upon receiving $\{A, Z_A, F_A, B, Z_B, F_B\}$, the trusted server *TS* computes

$(Q_{Ax} \| Q_{Ay}) = Z_A \oplus h(pw_A, A, B)$, $F_A^{'} = d(Q_{Ax}, Q_{Ay})$,

$(Q_{Bx} \| Q_{By}) = Z_B \oplus h(pw_B, A, B)$, $F_B^{'} = d(Q_{Bx}, Q_{By})$.

Then TS checks if $F_A^{'} = F_A$ and $F_B^{'} = F_B$. If the checking holds, *TS* randomly chooses t, computes:

$R_A = t(pw_A)Q_A = t(pw_A)t_aP$,

$R_B = t(pw_B)Q_B = t(pw_B)t_bP$,

and sends R_A and R_B to *B*. Otherwise, TS terminates the protocol.

Round 4: After obtaining R_A and R_B, *B* computes

$K = t_b(pw_B)R_A = t_b(pw_B)t(pw_A)t_aP = (K_x, K_y)$,

$S_B = h(K_x, K_y, B)$,

and sends S_B and R_B to *A*.

Round 5: After obtaining S_B and R_B, *A* computes

$K = t_a(pw_A)R_B = t_a(pw_A)t(pw_B)t_bP = (K_x, K_y)$,

and checks if $S_B = h(K_x, K_y, B)$. If the checking holds, *A* computes and sends $S_A = h(K_x, K_y, A)$ to *B***.** Otherwise, *A* terminates the protocol.

Round 6: After obtaining S_A, *B* checks if $S_A = h(K_x, K_y, A)$. If the checking does not hold, *B* terminates the protocol. Otherwise, *A* and *B* has established the session key $K = t_b(pw_B)t(pw_A)t_aP$.

3. Attacks on Lou-Huang 3PAKE protocol

In this section, we show that Lou and Huang's 3PAKE is vulnerable to off-line password guessing attack and partition attack.

3.1. Off-line password guessing attack

It seems that Lou and Huang's protocol can defend against off-line password guessing attack as pw_A, pw_B and t cannot be computed from R_A and R_B due to the intractability of Elliptic Curve Discrete Logarithm Problem (ECDLP). However, we will show that this kind of attack can be amounted against their protocol. The reason is that the users *A* and *B* have no direct authentication on whether R_A and R_B are sent by *TS*. According to the security model proposed by Dolev and Yao [21], an active attacker can control the communication channels through intercepting the communication and inserting data into the channels. Below are the details of our attacks.

Suppose *A* is a malicious user who targets for user *B*'s password, *A* performs as follows.

Step 1: *A* randomly chooses an integer C and a point $R_B^{'}$ over E, and computes $R_A^{'} = CP$.

Step 2: *A* sends the request to user *B* for setting up a session key with the help of *TS*. *B* accepts the request and performs the protocol with both *A* and *TS*. Round 1, Round 2 and Round 3 are the same as in Lou and Huang's protocol without any modification. After Round 3, *TS* sends R_A and R_B to *B*. *A* intercepts the communication between *TS* and *B* and change (R_A, R_B) to $(R_A^{'}, R_B^{'})$.

Step 3: After obtaining $(R_A^{'}, R_B^{'})$, *B* computes

$$K = t_b(pw_B)R_A^{'} = t_b(pw_B)CP = (K_x, K_y),$$

$$S_B = h(K_x, K_y, B),$$

and sends S_B and $R_B^{'}$ to *A*.

Step 4: After getting S_B and $R_B^{'}$ from *B*, *A* computes

$$(Q_{Bx}^{'} \,\|\, Q_{By}^{'}) = Z_B \oplus h(pw_B^{'}, A, B),$$

$$C(pw_B^{'})(Q_{Bx}^{'}, Q_{By}^{'}) = (K_x^{'}, K_y^{'})$$

where $pw_B{}'$ is a guessed password. Then A verifies if $S_B = h(K_x^{'}, K_y^{'}, B)$ holds or not. If it holds, the guessed password is correct, otherwise *A* makes another guessing and performs above attack again.

Therefore, Lou and Huang's 3PAKE protocol cannot resist off-line password guessing attack. To solve this problem, one method is to let *TS* sign R_A and R_B for authentication, but this will make the protocol less efficient and therefore, less suitable for resource-constrained devices.

Remark: The above attack can also be launched by an outsider. The outsider just replays *A*'s message $\{A, Z_A, F_A\}$ to *B***.** After *TS* sends R_A and R_B to *B*, the attacker intercepts the communication between *TS* and *B* and replaces (R_A, R_B) with $(R_A^{'}, R_B^{'})$. Then the attacker can launch the above off-line password guessing attack.

3.2. Partition attack

We now describe another attack against Lou-Huang 3PAKE. In the protocol, the output value of the hash function is a random number. We show that this allows an attacker to launch partition attack to eliminate more than one trial password by simply eavesdropping the communication among *A*, *B* and *TS*. The details are as follows.

Note that $Z_A = (Q_{Ax} \,\|\, Q_{Ay}) \oplus h(pw_A, A, B)$, where Q_{Ax} and Q_{Ay} are the *x*-coordinate and *y*-coordinate of Q_A, respectively. Consider a typical elliptic curve equation $y^2 = x^3 + ax + b \pmod q$. Only a half of the *x*-coordinate values in Z_q have solutions. So an eavesdropper can get Z_A and use a guessed password, say $pw_A^{'}$, to check if $Z_A \oplus h(pw_A^{'}, A, B)$ is a valid elliptic curve point or not. If not, $pw_A^{'}$ must be an invalid password. Therefore, an eavesdropper can eliminate at least half of the passwords in the password space in Lou and Huang's 3PAKE.

To solve this problem, we can choose a secure hash function which maps into the points on elliptic curve. In particular, Z_A, $h(pw_A, A, B)$ and Q_A are points on the elliptic curve. In the next section, we propose a new protocol which can resist both off-line password guessing attack and partition attack.

4. The improved protocol

In this section, we propose an improved 3PAKE and provide a security analysis of this scheme against various attacks. The basic ideas of our constructions are as follows: (1) user *A* and user *B* directly authenticate that R_A and R_B are sent by *TS* and unmodified by anyone else. Instead of using digital signature, we propose a more efficient method which allows R_A or R_B to be recovered only by the one who knows pw_A or pw_B; (2) we use a secure hash function which maps to points on elliptic curve to resist partition attack.

The system parameters are generated in the same way as Lou and Huang's protocol except that the definition of hash function is changed so that $h()$ maps the input to an elliptic curve point.

Round 1: User *A* randomly chooses t_a, computes two points $Q_A = t_aP$, $F_A = t_aF = t_adP = dQ_A$, and sets $Z_A = Q_A \oplus h(pw_A, A, B)$, where $h(pw_A, A, B)$ is a point on elliptic curve. Then A sends $\{A, Z_A, F_A\}$ to B.

Round 2: User *B* randomly chooses t_b, computes two points $Q_B = t_bP$, $F_B = t_bF$, and sets $Z_B = Q_B \oplus h(pw_B, A, B)$. *B* sends $\{A, Z_A, F_A, B, Z_B, F_B\}$ to *TS*.

Round 3: Upon receiving $\{A, Z_A, F_A, B, Z_B, F_B\}$, the trusted server *TS* computes

$$Q_A = Z_A \oplus h(pw_A, A, B),\ F_A^{'} = dQ_A,$$

$$Q_B = Z_B \oplus h(pw_B, A, B),\ F_B^{'} = dQ_B.$$

TS checks if $F_A^{'} = F_A$ and $F_B^{'} = F_B$. If the checking holds, TS randomly chooses t, computes

$$r_A = tQ_A = tt_aP, r_B = tQ_B = tt_bP$$

$$R_A = r_A \oplus h(pw_B, B, A), R_B = r_B \oplus h(pw_A, B, A),$$

and sends R_A and R_B to *B*. Otherwise, *TS* terminates the protocol.

Round 4: When *B* obtains R_A and R_B , he computes

$K_1 = R_A \oplus h(pw_B, B, A) = tt_a P$,

$K = t_b K_1 = t_b tt_a P$, $S_B = h(K, B)$,

and sends S_B and R_B to *A*.

Round 5: When *A* gets S_B and R_B , he computes

$K_2 = R_B \oplus h(pw_A, B, A) = tt_b P$,

$K = t_a K_2 = t_a tt_b P$,

and verifies whether $S_B = h(K, B)$ or not. If it holds, *A* computes and sends $S_A = h(K, A)$ to *B*. Otherwise, he terminates the protocol.

Round 6: When *B* obtains S_A , he verifies whether $S_A = h(K, A)$ or not. If it does not hold, *B* terminates the protocol. Otherwise, *A* and *B* has established the session key $K = t_a tt_b P$.

5. Security analysis and performance comparison

5.1. Security analysis

1. Offline password guessing attack

Suppose an adversary (e.g. a malicious user *A*) eavesdrops the communication between *B* and TS, and gets Z_B, F_B, R_A and R_B , and launches off-line password guessing attack. As described above, the adversary may randomly choose an integer C and a point $R_B^{'}$ over the elliptic curve E , and compute $R_A^{'} = CP$, then send $(R_A^{'}, R_B^{'})$ to *B*. *B* computes

$K_1 = R_A^{'} \oplus h(pw_B, B, A) = C^{'} P$,

$K = t_b K_1 = C^{'} t_b P$,

$S_B = h(K, B)$

and sends S_B and $R_B^{'}$ back to the adversary. So the adversary guesses *B*'s password $pw_B{}'$, and computes

$R_A^{'} \oplus h(pw_B^{'}, B, A) = C^{''} P$,

$Q_B^{'} = Z_B \oplus h(pw_B^{'}, A, B) = t_b^{'} P$.

However, the adversary cannot get $C^{''}$ or $t_b^{'}$ from $C^{''} P$ or $t_b^{'} P$ due to the intractability of ECDLP, and also cannot compute $C^{''} t_b^{'} P$ from $C^{''} P$ and $t_b^{'} P$ due to the intractability of the Computational Diffie-Hellman (CDH) problem. Therefore, the adversary cannot verify if the guessed password $pw_B{}'$ is correct or not.

2. Perfect forward secrecy

In the improved scheme, the session key is $K = t_a tt_b P$, where t_a , t_b and t are nonce chosen by user *A*, user *B* and the trusted server *TS*, respectively. Even if an adversary gets *TS*'s secret key d , *A* and *B* 's passwords, he can only get $tt_b P$ and $tt_a P$, but he is not able to compute the session key of any previously established sessions due to the intractability of CDH problem.

3. Replay attack

Suppose that an adversary impersonates *A* and replays *A*'s message $\{A, Z_A, F_A\}$ to *B*. He cannot verify $S_B = h(K, B)$ and respond with the correct $S_A = h(K, A)$ to *B* as t_b and t are new nonce chosen by *B* and *TS* in each new session so that the adversary has no control over it.

Suppose that an adversary impersonates *B* and replays *B*'s message $\{A, Z_A, F_A, B, Z_B, F_B\}$ to *TS*. Then he cannot respond with the correct $S_B = h(K, B)$ to *A* since t and t_a are new nonce chosen by *TS* and *A* in each new session so that the adversary has no control over it.

Suppose that the adversary replays *TS*'s message R_A and R_B . The replayed message cannot pass the verification of both *A* and *B*, and cannot get the session key as t_a and t_b are new nonce chosen by *A* and *B* in each new session so that the adversary has no control over it.

4. Forgery attack and impersonation

An adversary may impersonate *A* (or *B*) and send $\{A, Z_A, F_A\}$ (or $\{B, Z_B, F_B\}$) to *B* (or *TS*). However, the adversary's response message S_A (or S_B) cannot pass the verification process of *B* (or *A*) as the password is unknown.

5. Denning-Sacco attack

Even if an adversary gets the session key $K = t_a tt_b P$, he cannot get $t_a P$, $t_b P$, $tt_b P$ and $tt_a P$ due to the intractability of ECDLP. Therefore, the adversary cannot get *TS*'s secret key d , *A* and *B*'s passwords from Z_A, F_A , Z_B, F_B , R_A and R_B .

6. Known-key security

Due to the randomness and independence of generating t_a , t_b and t in all the sessions, the session key $K = t_a tt_b P$ of each session is independent. Therefore, an adversary is unable to compute either previous or future session keys given a session key.

7. Man-in-the-middle attack

If an adversary mounts man-in-the-middle attack by impersonation and replay attack, the adversary cannot gain any advantage due to the reasons given above. Next, we analyze if a malicious insider Eve can succeed in launching man-in-the-middle attack.

When *B* sends $\{A, Z_A, F_A, B, Z_B, F_B\}$ to *TS*, suppose that Eve intercepts and sends $\{A, Z_A, F_A, ID_E, F_E, Z_E\}$ and $\{ID_E, F_E, Z_E, B, Z_B, F_B\}$ to *TS*, where ID_E is Eve's identity. *TS* randomly chooses t_1, computes and returns R_A and R_E; and randomly chooses t_2, computes and returns $R_E^{'}$ and R_B, respectively. Since R_A, R_E, $R_E^{'}$ and R_B include users' passwords and identities, Eve cannot impersonate *B* to successfully establish a session key with A, or vice versa, without knowing A and B's passwords.

Therefore, the improved scheme can resist man-in-the-middle attack.

5.2. Performance comparison

The differences between the improved scheme and Lou-Huang scheme are in the generation of R_A, R_B and K and the hash function. As we can see, our scheme has four more hash operations than Lou-Huang scheme, more precisely, the user *A* and user *B* have one more hash operation respectively, while the trusted server *TS* has two more hash operations. On the other hand, our scheme has four less modular multiplication computations than Lou-Huang scheme, more precisely, the user *A* and user *B* have one less modular multiplication computation respectively, while *TS* does not need to perform modular multiplication. Therefore, the improved scheme not only enhances security, but also keeps efficiency.

Table 1: The performance comparison

	Lou-Huang's scheme		**Our scheme**	
	A/B	***TS***	***A/B***	***TS***
Modular Exponentiation	0	0	0	0
Scalar Multiplication	3	4	3	4
Hash Operation	3	2	4	4
Modular Multiplication	2	2	1	0

6. Protocol verification

In this section, we use ProVerif tool [20] to prove that the proposed protocol satisfies the mutual authentication and session key secrecy. In the formal model, the protocol was modeled as the parallel execution of three distinct processes: the user *A*, the user *B* and the server:

```
process pUserA | pUserB |!pTS
```

The processes are replicated in order to model that several users may communicate with the server at the same time. The processes of the users *A* and *B* are defined as:

```
let pUserA=
  new ta:bitstring;
  let QA=mult(ta,P) in
  let F=mult(d,P) in
  let FA=mult(ta,F) in
  let ZA=add(QA,h(((PWA,A,B)))) in
  out(sch1,(A,B,ZA,FA));
  event beginUserA(A,B);
  in(sch1,(tA:bitstring,tB:bitstring,tRB:bitstring,tSB:bitstring));
  let K2=add(tRB,h(((PWA,tB,tA)))) in
  let K'=mult(ta,K2) in
  let SB'=h((K',tB)) in
  if SB'=tSB then
  let SA=h((K',tA)) in
  out(sch1,(tA,tB,SA));
  event endUserA(tA,tB).

let pUserB=
  new tb:bitstring;
  in(sch1,(xA:bitstring,xB:bitstring,xZA:bitstring,xFA:bitstring));
  let QB=mult(tb,P) in
  let F'=mult(d,P) in
  let FB=mult(tb,F') in
  let ZB=add(QB,h(((PWB,A,B)))) in
  out(sch2,(xA,xZA,xFA,xB,ZB,FB));
  event beginUserB(A,B);
  in(sch2,(zA:bitstring,zB:bitstring,zRA:bitstring,zRB:bitstring));
  let K1=add(zRA,h(((PWB,zB,zA)))) in
  let K=mult(tb,K1) in
  let SB=h((K,zB)) in
  out(sch1,(zA,zB,zRB,SB));
  in(sch1,(pA:bitstring,pB:bitstring,pSA:bitstring));
  let SA'=h((K,pA)) in
  if SA'=pSA then
  let sk=mult(tb,K1) in
  event endUserB(zA,zB).
```

The server process is modeled as:

```
let pTS=
  in(sch2,(yA:bitstring,yZA:bitstring,yFA:bitstring,
  yB:bitstring,yZB:bitstring,yFB:bitstring));
  let QA=add(yZA,h(((PWA,yA,yB)))) in
  let FA'=mult(d,QA) in
  let QB=add(yZB,h(((PWB,yA,yB)))) in
  let FB'=mult(d,QB) in
  if FA'=yFA then
  if FB'=yFB then
  new t:bitstring;
  let rA=mult(t,QA) in
```

```
    let rB=mult(t,QB) in
    let RA=add(rA,h(((PWB,yB,yA)))) in
    let RB=add(rB,h(((PWB,yB,yA)))) in
    out(sch2,(yA,yB,RA,RB)).
```

The secrecy of the session key is modeled as the following query and events:

```
    query attacker(sk).
    event beginUserA(bitstring,bitstring).
    event endUserA(bitstring,bitstring).
    event beginUserB(bitstring,bitstring).
    event endUserB(bitstring,bitstring).
```

The mutual authentication of the protocol is modeled as the following queries:

```
    query id:bitstring; inj-event(endUserA(id,id))
      ==> inj-event(beginUserA(id,id)) .
    query id:bitstring; inj-event(endUserB(id,id))
      ==> inj-event(beginUserB(id,id)) .
```

The readers may refer to the online demo for ProVerif: http://proverif.rocq.inria.fr/index.php to test above codes. The outputs by this formal verification tool show that the proposed scheme can pass all the evaluations. Hence, our protocol is secure, in the sense that it provides both mutual authentication and session key secrecy.

7. Conclusion

In this paper, we showed that Lou and Huang 3PAKE protocol is vulnerable to off-line password guessing attack and partition attack. In addition, we not only propose a security-enhanced scheme for solving these problems, but also keep the efficiency of the scheme. One of our future work is to study on how to build a provably secure protocol while maintaining the efficiency when compared with the protocol we proposed in this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No. 61070153), the Major State Basic Research Development (973) Program of China (No. 2013CB834205), Natural Science Foundation of Zhejiang Province (No. LZ12F02005), and a grant from the RGC of the HKSAR, China (Project No. CityU 121512).

References

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

[2] **C. L. Lin, H. M. Sun, M. Steiner, T. Hwang.** Three-party encrypted key exchange without server public-keys. In: *IEEE Communication Letters*, 2001, Vol. 5, pp. 497-499.

[3] **Y. Ding, P. Horster.** Undetectable on-line password guessing attacks. *ACM Operating Systems Review*, 1995, Vol. 29, 77-86.

[4] **R. X. Lu, Z. F. Cao.** Simple three-party key exchange protocol. *Computers and Security*, 2007, Vol. 26, 94-97.

[5] **H. Guo, Z. J. Li, Y. Mu, X. Y. Zhang**, Cryptanalysis of simple three-party key exchange protocol, *Computers and Security*, 2008, Vol. 27, 16-21.

[6] **Y. F. Chang**. A practical three-party key exchange protocol with round efficiency. *International Journal of Innovative Computing, Information and Control*, 2008, Vol. 4, 953-960.

[7] **H. R. Chung, W. C. Ku.** Three weaknesses in a simple three-party key exchange protocol. *Information Sciences*, 2008, Vol. 178, 220-229.

[8] **R. C. W. Phan, W. C. Yau, B. M. Goi**. Cryptanalysis of simple three-party key exchange protocol (S-3PAKE). *Information Sciences*, 2008, Vol. 178, 2849-2856.

[9] **J. Y. Nam, J. Y. Paik, H. K. Kang, U. M. Kim, D. H. Won**. An off-line dictionary attack on a simple three-party key exchange protocol. In: *IEEE Communication Letters*, 2009, Vol. 13, pp. 205-207.

[10] **H. S. Kim, J. Y. Choi**. Enhanced password-based simple three-party key exchange protocol. *Computers and Electrical Engineering*, 2009, Vol. 35, 107-114.

[11] **H. F. Huang.** A simple three-party password-based key exchange protocol. *International Journal of Communication Systems*, 2009, Vol. 22, 857-862.

[12] **E. J. Yoon, K. Y. Yoo.** Cryptanalysis of a simple three-party password-based key exchange protocol. *International Journal of Communication Systems,* 2011, Vol. 24, 532-542.

[13] **D. C. Lou, H. F. Huang**. Efficient three-party password-based key exchange scheme. *International Journal of Communication Systems*, 2011, Vol. 24, 504-512.

[14] **Cheng-Chi Lee, Chin-Ling Chen, Hsia-Hung Ou, Lung Albert Chen**. Extension of an efficient 3GPP authentication and key agreement protocol. *Wireless Personal Communications*, 2013, Vol. 68, No. 3, 861-872.

[15] **Chun-Ta Li**. Secure smart card based password authentication scheme with user anonymity. *Information Technology and Control*, 2011, Vol. 40, No. 2, 157-162.

[16] **Chun-Ta Li**. A more secure and efficient authentication scheme with roaming service and user anonymity for mobile communications. *Information Technology and Control*, 2012, Vol. 41, No. 1, 69-76.

[17] **Bae-Ling Chen, Wen-Chung Kuo, Lih-Chyau Wuu**. A secure password-based remote user authentication scheme without smart cards. *Information Technology and Control*, 2012, Vol. 41, No. 1, 53–59.

[18] **Shirisha Tallapally**. Security enhancement on simple three party PAKE protocol. *Information Technology and Control*, 2012, Vol. 41, No. 1, 15-22.

[19] **Chun-Ta Li, Cheng-Chi Lee.** A robust remote user authentication scheme using smart card. *Information Technology and Control*, 2011, Vol. 40, No. 3, 236-245.

[20] **M. Abadi, B. Blanchet, H. C. Lundh.** Models and proofs of protocol security: A progress report. *21st*

International Conference on Computer Aided Verification, 2009.

[21] **D. Dolev, A. C. Yao.** On the security of public key protocols In: *IEEE Transactions on Information Theory* 29, 1983, pp. 198-208.

Received June 2012.