A Lightweight Three-party Authenticated Key Exchange Protocol with XOR-based operation

Chung-YiLin* and Chen-Hua Fu

Department of Information Management National Defense University

ABSTRACT

Traditional three-party authenticated key exchange (3PAKE) protocols face two common problems, one is that the shared key can be compromised, and the other is the high computational cost that is required to run them. With the rapid development of services such as cloud and ubiquitous computing, devices with relatively low computing power are becoming more widely used. Some studies have thus proposed 3PAKE protocols with elliptic curve cryptography to reduce the computational cost in a mobile commerce environment, but these not only cannot counteract impersonation attacks, but still have a relatively high computational cost. Therefore, this paper proposes a lightweight 3PAKE protocol with XOR（exclusive OR）-based operation that has the following characteristics: (1) less computational cost than the 3PAKE protocols with elliptic curve cryptography, (2) greater confidentiality with regard to sharing keys, and (3) the ability to resist impersonation attacks..

Keywords: 3PAKE, Authenticated Key Exchange, XOR Operation

以 XOR 運算為基之輕巧的三方認證鑰匙交換協定

林忠毅* 傅振華

國防大學管理學院資訊管理學系

摘　　要

傳統的三方認證鑰匙交換協定存在共享鑰匙被外洩的威脅及需要較大的運算成本。隨著雲端及無所不在運算服務快速的發展，低運算能力的設備將可能被廣泛的使用。一些研究提出以橢圓曲線密碼為基礎的認證鑰匙交換協定，藉以在行動商務環境減低運算成本，然而，這些協定不但無法抵抗冒名攻擊，並且，仍存在較高的運算成本。基於此，我們提出以 XOR 運算為基礎之輕巧的三方認證鑰匙交換協定達到（1）較橢圓曲線密碼方法更低的運算成本（2）共享鑰匙的保密性（3）抵抗冒名攻擊。

關鍵詞：3PAKE，三方認證鑰匙交換協定，XOR運算

文稿收件日期 103.06.26;文稿修正後接受日期 104.12.28; *通訊作者
Manuscript received June 26, 2014; revised December 28, 2015; * Corresponding author

I. INTRODUCTION

In traditional 3PAKE protocols [1-4], session users provide their passwords as shared keys to a trusted server, which then provides authentication to them. However, there are two common problems for such protocols, one is that the shared keys stored in the server in plaintext form can be stolen, compromising security, and the other is the use of modular exponentiation-based operations, which requires significant computational costs.

The rise of cloud and ubiquitous computing in recent years means that an increasing number of low computing devices are now being used, such as cell phones, e-readers, tablets and other gadgets [5-6], which are very limited in terms of computational capabilities, memory, communication bandwidth, and battery power [7][8]. It is thus necessary to develop an authenticated key exchange protocol that can be operated on such devices. From a computational cost perspective, since traditional 3PAKE protocols based on modular exponentiation require more computational cost, they cannot operate properly on such devices.

Some studies have focused on reducing the computational cost of 3PAKE protocols in a mobile commerce environment, with approaches based on elliptic curve cryptography being the most common [9][10], although these remain vulnerable to impersonation attacks [11], and the computational cost is still too high for certain devices. Therefore, this paper proposes a lightweight 3PAKE protocol with XOR-based operation to achieve the necessary efficiency and security requirements, thus making it more suitable for use with low computing power devices. The proposed protocol has the following advantages. (1) It requires less computational cost than 3PAKE protocols with elliptic curve cryptography; (2) it allows the shared authenticated keys to be stored in ciphertext form, and thus their security cannot be compromised; and (3) it can counteract impersonation attacks.

The rest of this paper is organized as follows: Section Ⅱ reviews the related literature, while we describe a proposed security mechanism in Section Ⅲ. The proposed protocol is presented in Section Ⅵ, while the we then provides a security analysis of the proposed protocol in Section Ⅶ, and an efficiency analysis of it in Section Ⅷ. Finally, we conclude this paper in Section Ⅸ.

Ⅱ. RELATED WORKS

Diffie and Hellman proposed the first key exchange protocol [12], but this does not include an authentication function, and thus is vulnerable to man-in-the-middle attacks. Many other protocols were thus developed, all of which include authentication, such as Wilson and Menezes [13], Bellovin and Merrit [14], and Abdalla and Pointcheval [15]. Bellovin and Merrit [14] proposed a two-party PAKE (2PAKE) protocol, which allows a pair of session users to share their passwords in advance to establish a common ephemeral session key in a secure fashion. However, this protocol has a problem, which is that each session user has to remember a considerable number of passwords in order to communicate with others session users in a large-scale communication environment.

To overcome this, Steiner et al. [16] proposed that session users share their passwords with a trusted server, which provides authentication to them, and this approach is called three-party PAKE (3PAKE). However, Steiner et al.'s protocol is still not fully secure, as noted by Lin et al. [17] and Sun et al. [18], because it is unable to detect password-guessing attacks that may occur both on-line and off-line. To avoid these, a number of schemes in which the server uses its own public key in 3PAKE have been proposed, such as in Lin et al. and Sun et al. However, these require PKI, and the complex calculations associated with this lead to a heavy computational overhead. Therefore, the use of a server's public key is not suitable in computationally limited environments.

A number of studies have thus developed protocols that do not use a server's public key [1-4], and these have much lower computational overheads. However, two common problems remain for 3PAKE protocols, one is that the shared keys stored in the server in plaintext form can be stolen, thus compromising security, and the other is the use of modular exponentiation-based computation, which requires significant computational resources. To avoid the shared keys in the server being compromised, Chen et

al. [19] proposed a protocol which does not storeany security-sensitive tables on the server, and is still able to authenticate users. However, Yang and Chang [9] pointed out that Chen et al.'s protocol cannot be applied to user authentication for mobile commerce, because it has to compute modular exponentiation, which requires a considerable amount of time and computational power. Yang and Chang also showed that the protocol cannot resist stolen-verifier attacks. To overcome these weaknesses, Yang and Chang proposed a 3PAKE protocol using elliptic curve cryptography (ECC-3PAKE) for mobile commerce. However, Pu et al. [20] showed that this protocol is potentially vulnerable to unknown key share attacks, and suggested changing the computation of the ciphertexts as a countermeasure to resist these. Tan [10] then demonstrated that Yang and Chang's protocol is vulnerable to impersonation attacks (impersonation-of-initiator and impersonation-of-responder), and proposed an enhanced ECC-3PAKE protocol to resolve this. However, Nose [11] further found that Tan's protocol cannot completely protect against impersonation attacks.

Ⅲ. A proposed security mechanism

The proposed security mechanism is based on the security objective as subsection 3.1, and it includes a novel three-party session key establish (3PSKE) mode and both authentication and self-encryption schemes within this.

3.1 Security objectives

This study proposes a lightweight computational model to develop a 3PAKE with a lower computational cost than ECC. However, the system still has the following threats that must be considered: (1) the shared authenticated key, which stores the server and user sites, may be compromised by the adversaries. (2) The adversaries can conduct both impersonation-of-initiator attack and impersonation-of-responder attack. (3) The adversaries can conduct known-plaintext attacks.

The objective is thus to improve these security threats.

3.2 The 3PSKE mode

The 3PSKE mode is shown in Fig. 1, where A represents the initiator and B represents the responder. In 3PSKE mode, both site users and the server collaborate to generate a session key. We will use the A-Server-B as an example. As illustrated in Fig. 1, A generates the identity hash value (as shown in Fig. 2) to encrypt the sub-key (i.e. the session key component) and delivers the sub-key to the server. The server adopts the same procedure to generate A's identity hash value and obtains its sub-key. The server then generates its sub-key and uses it to encrypt the sub-key of A into a confused sub-key. Next, the shared authenticated key of B is used by the server to encrypt the confused sub-key into the exchange value, which is then delivered to B. Using its authenticated key, B retrieves the confused sub-key from the exchange value, and generates a session key using this value combined with its sub-key.

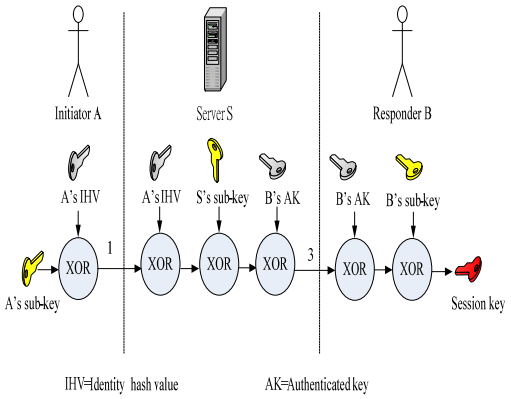
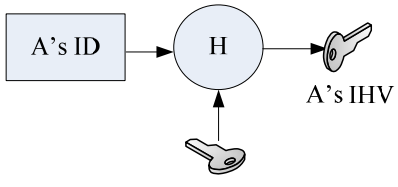


Fig. 1. The procedure by which A-Server-B generates the session key.



AK=Authenticated key H=One way hash function
IHV=Identity hash value

Fig. 2. The procedure of generating the identity hash value.

3.3 An authentication scheme

To counteract the adversaries conducting an impersonation attack, a coordinated A-Server-B and B-Server-A authentication scheme is designed used in the 3PSKE mode. As these two procedures employ an identical procedure, we will use the A-Server-B authentication procedure as an example. First, A inputs the IDB, sub-key, timestamp and authenticated key into hash to generate a message hash value, as shown in Fig. 3. This hash value provided to the server for authentication. Next, A delivers its message verification value to the server. After S retrieves the sub-key of A, it also employs the same procedure to generate a new verification value to authenticate A.

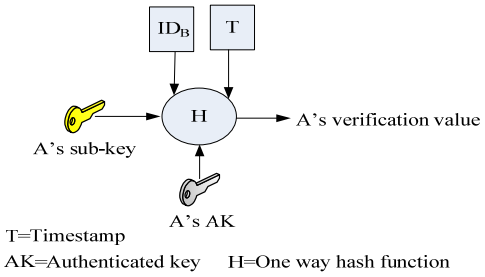


Fig. 3. The procedure by which A generates its verification value.

When A pass the authentication of the server, the server then input IDA, the confused sub-key and the authenticated key of B, to generate a message hash value, as shown in Fig. 4. This hash value is provided to B for authentication. The server then delivers its message verification value to B. After B retrieves the confused sub-key, it also employs the same procedure to generate a new verification value to authenticate the Server.

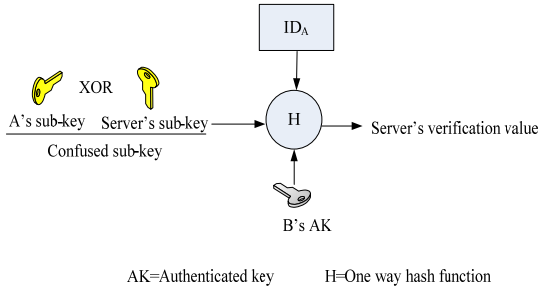


Fig. 4. The procedure by which the Server generates its message verification value.

3.4 A self-encryption scheme

A self-encryption scheme allows the user to generate an encryption key with its password, and also enables it to encrypt an important parameter, such as the shared authenticated key. Using a self-encryption scheme, the user can ensure the confidentiality of an important parameter. The self-encryption scheme developed in this work is described in detail below.

First, the user generates a random number (rn) and lets password and rn would be each other's salt, and inputs them into a hash (H) to generate the hash values, with PA and AG being the self-encryption keys of the user, as shown in Fig. 4. Since the password is simple for the user to memorize, the user can encrypt and retrieve the rn with its password, such as $C\_rn = rn\ XOR\ password$, where C_rn is the encrypted rn. Furthermore, the user can use PA and AG separately or jointly to encrypt the important parameter, such as $C\_par = par\ XOR\ PA\ XOR\ AG$ (par represents an important parameter). Therefore, even if C_par is stolen, it is infeasible for an adversary to retrieve par with unknown PA and AG. On the other hand, when par is needed by the user, it can be retrieved from C_par based on the PA and AG, which are generated from the password and rn.

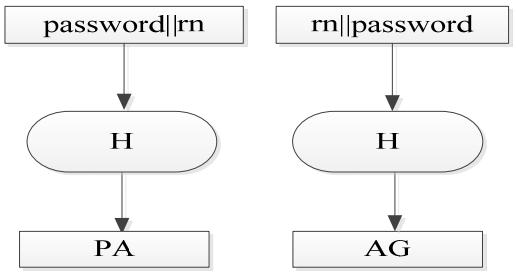


Fig. 5. The process by which the user generates self-encryption keys.

Ⅵ. The proposed protocol

In this section, we present an L-3PAKE protocol based the proposed security mechanism described in Section Ⅲ, and it is divided into initial and sub-keys exchange stages, and discussed in subsections 4.1 and 4.2,

respectively. Before describing the proposed protocol in more detail, the assumptions it is based on are presented. First, A is the initiator, B is the responder, and S is the trusted server. Collision-free hash and random number generation algorithm are able to generate outputs of sufficiently secure bit-lengths. The server has its secret key which can be kept secure. The notations used in the proposed protocol are defined in Table 1.

Table 1. Notations used in the proposed protocol

Parameter	Definition
H	collision-free one way hash function
⊕	exclusive OR operation
E_k ,D_k	symmetric encryption / ecryption process with k , where k is a secret key of the server.
ID_A,ID_B	identities of A and B, respectively
rad_A,rad_B	random numbers generated by the server for A and B, respectively
psw_A,psw_B	passwords of A and B, respectively
rn_A,rn_B	random numbers generated by A and B, respectively
PA_A,AG_A PA_B,AG_B	self-confusion keys generated by A and B, respectively
sk_A,sk_B	components of session key generated by A and B, respectively
s_A,s_B	shared authenticated keys with the server generated by A and B, respectively
SAG_A, SAG_B	cluesstored in A and Bsites to retrieve s_A and s_B, respectively
VS_A,VS_B	clues stored in the server to retrieve s_A and s_B
$token_A$, $token_B$	tokens used to help the server retrieve s_A and s_B
ihv_A, ihv_B	identity hash value generated by A and B, respectively
mv_A,mv_B, vs	message verification values generated by A and B ,and S, respectively
C_sk_A, C_sk_B	encrypted sub-keysgenerated by A and B, respectively
ex_sk_A , ex_sk_B	exchange values generated by the server for A and B, respectively
T_A,T_B	timestamps generated by A and B, respectively

4.1 Initialization stage

Before A and B participate in a session, they must register with the server. This procedurs is the same for A and B. We used the registration procedures of A in the following example.

（1） A sets its password psw_A and generates a random number rn_A .

（2） A compute $PA_A = H(psw_A || rn_A)$ and $AG_A = H(rn_A || psw_A)$.

（3） A computes $C\_rn_A = rn_A \oplus psw_A$.

（4） A generates a random number s_A as a shared authentication key with the server and computes $SAG_A = s_A \oplus AG_A$.

（5） A computes $SAG_A \oplus PA_A$ anddeliver $(ID_A, SAG_A \oplus PA_A)$ to S in a secure way. A then stores C_rn_A and SAG_A

（6） After S receives the message from A, it generates random number rad_A and computes $\gamma_A = H(rad_A)$.

（7） S uses γ_A toencrypt $SAG_A \oplus PA_A$ into VS_A , as $VS_A = SAG_A \oplus PA_A \oplus \gamma_A$.

（8） S computes $C\_rad_A = E_k(rad_A)$, and then stores (ID_A, VS_A, C_rad_A).

4.2 Sub-key exchange stage

Round 1

（1） A delivers $(ID_A, request)$to B

（2） A computes $rn_A = C\_rn_A \oplus psw_A$.

（3） A computes $PA_A = H(psw_A || rn_A)$ and $AG_A = H(rn_A || psw_A)$.

（4） A computes $s_A = SAG_A \oplus AG_A$.

（5） A computes $ihv_A = H(ID_A || s_A)$.

（6） A randomly generates sk_A , T_A and computes $mv_A = H(ID_B || T_A || sk_A || s_A)$.

（7） A computes $C\_sk_A = sk_A \oplus ihv_A$ and $token_A = PA_A \oplus AG_A$, and

delivers $(ID_A, ID_B, C\_sk_A, token_A, mv_A, T_A)$ to S.

Round 2

(1) After receives the message that A delivers, B computes $rn_B = C\_rn_B \oplus psw_B$.

(2) A computes $PA_B = H(psw_B \| rn_B)$ and $AG_B = H(rn_B \| psw_B)$

(3) B computes $s_B = SAG_B \oplus AG_B$.

(4) B computes $ihv_B = H(ID_B \| s_B)$.

(5) B randomly generates sk_B, T_B and computes $mv_B = H(ID_A \| T_B \| sk_B \| s_B)$.

(6) B computes $C\_sk_B = sk_B \oplus ihv_B$ and $token_B = PA_B \oplus AG_B$,and delivers $(ID_B, ID_A, C\_sk_B, token_B, mv_B, T_B)$ to S.

Round 3 (The procedure used by the Server)

(1) After receives messages from A and B, S check whether T_A, T_B are fresher than the one received in the last request. Then, S takes out (VS_A, C_rad_A) and (VS_B, C_rad_B) according to ID_A and ID_B.

(2) S computes $rad_A = D_k(C\_rad_A)$ and $\gamma_A = H(rad_A)$. Similarly, S also computes $rad_B = D_k(C\_rad_B)$ and $\gamma_B = H(rad_B)$.

(3) S computes $s_A = VS_A \oplus token_A \oplus \gamma_A$ and $s_B = VS_B \oplus token_B \oplus \gamma_B$ .When S retrieves s_A and s_B, it can compute $sk_A = C\_sk_A \oplus s_A$ and $sk_B = C\_sk_B \oplus s_B$.

(4) S computes $ihv_A' = H(ID_A \| s_A)$ and $ihv_B' = H(ID_B \| s_B)$.

(5) S computes $sk_A = C\_sk_A \oplus ihv_A'$ and $sk_B = C\_sk_B \oplus ihv_B'$.

(6) S computes $mv_A' = H(ID_B \| T_A \| sk_A \| s_A)$ and $mv_B' = H(ID_A \| T_B \| sk_B \| s_B)$. S then verifies $mv_A' = ?v_A$ and $mv_B' = ?v_B$.

(7) S computes $vs_A = H(ID_B \| sk_B \oplus sk_S \| s_A)$ and $vs_B = H(ID_A \| sk_A \oplus sk_S \| s_B)$.

(8) S computes $ex\_sk_A = sk_B \oplus sk_S \oplus s_A$ and $ex\_sk_B = sk_A \oplus sk_S \oplus s_B$.

(9) S delivers $(ID_A, ID_B, ex\_sk_A, vs_A)$ to A and delivers $(ID_B, ID_A, ex\_sk_B, vs_B)$ to B.

Round 3 (The procedure used by both site users)

Round 3.1 (The procedure used by A)

(1) After A receives the message, A computes $sk_B \oplus sk_S = ex\_sk_A \oplus s_A$ and $vs_A' = H(ID_B \| sk_B \oplus sk_S \| s_A)$.

(2) A verifies $vs_A' = ?vs_A$.

(3) A generates the common session key $session\,key = sk_B \oplus sk_S \oplus sk_A$.

Round 3.2 (The procedure used by B)

(1) After B receives the message, B computes $sk_A \oplus sk_S = ex\_sk_B \oplus s_B$ and $vs_B' = H(ID_A \| sk_A \oplus sk_S \| s_B)$.

(2) B verifies $vs_B' = ?vs_B$.

(3) B generates the common session key $session\,key = sk_A \oplus sk_S \oplus sk_B$.

Ⅶ. Security analysis

5.1 The proposed protocol ensures the confidentiality of the authenticated keys

To ensure the confidentiality of S_A, A use AG_A to encrypt $S_A$into SAG_A, such as $SAG_A = s_A \oplus AG_A$. Similarly, S uses γ_A to encrypt $SAG_A \oplus PA_A$ into VS_A, such as $VS_A = s_A \oplus AG_A \oplus PA_A \oplus \gamma_A$ in the initialization stage.

Assuming that an adversary has managed to steal VS_A and SAG_A, and also collect $token_A$ and $C_sk_A$in the communication channels. The adversary then perform the following computing: $PA_A \oplus \gamma_A = VS_A \oplus SAG_A$, $\gamma_A \oplus AG_A = token_A \oplus PA_A \oplus \gamma_A$, $s_A \oplus \gamma_A = SAG_A \oplus \gamma_A \oplus AG_A$ and $sk_A \oplus \gamma_A = C\_sk_A \oplus s_A \oplus \gamma_A$.

According to the presented above, if any one of γ_A,AG_A,$PA_A$and sk_A are cracked by the adversary, then she can retrieve s_A. In reality, regardless of the different operations that the adversary performs on SAG_A, VS_A, $token_A$ and C_sk_A, she will ultimately face two unknown parameters. Without being able to confirm either of these parameters, the adversary cannot retrieve s_A. Also, even if the adversary steals C_rad_A, because it is unaware of the secret key of the server, it cannot retrieve rad_A from C_rad_A and use rad_A to generate γ_A through hash. Therefore, the adversary cannot impersonate Ato

counterfeit C_sk_A and $v_A$to pass the authentication of the server.According to the analysis presented above, the proposed protocol prevents the shared authenticated keys from being compromised.

5.2 The proposed protocol resists impersonation attacks

1. Impersonation of initiator attack

If an adversary X wants to impersonate A in order to establish communication with B, it first delivers a message to B, such as $(ID_A \,||\, requst)$. X then delivers a message to S, as $(ID_X, ID_B, C\_sk_X, token_X, mv_X, T_X)$. When B receives this message, B delivers a message to the server, such as $(ID_B, ID_A, C\_sk_B, token_B, mv_B, T_B)$. X then interrupts the message in the communication channel between B and server and tampers with it, as $(ID_B, ID_X, C\_sk_B, token_B, mv_B, T_B)$ in an attempt to pass the authentication of the server and cause B to mistakenly believe that it is establishing communication with A.

In the above condition, since B generates the verification value, such as $mv_B = H(ID_A \,||\, T_B \,||\, sk_B \,||\, s_B)$, if X changes ID_A into ID_X, themv_B'which S generated with $mv_B$is not the same, and so X cannot pass the server's authentication.

2. Impersonation of responder attack

If A wants to establish a session with B, it needs to deliver a message to B, such as $(ID_A \,||\, requst)$, then deliver a message to S, such as $(ID_A, ID_B, C\_sk_A, token_A, mv_A, T_A)$. X intercepts these messages on the communication channels, and tampers with the one that A delivers to S, such as $(ID_A, ID_X, C\_sk_A, token_A, mv_A, T_A)$. X then delivers a message to S, such as $(ID_X, ID_A, C\_sk_X, token_X, mv_X, T_X)$, in an attempt to pass the authentication of S and cause A to mistakenly believe that it is establishing communication with B.

In the above condition, since A generates the verification value as $mv_A = H(ID_B \,||\, T_A \,||\, sk_A \,||\, s_A)$, if C changes ID_B into ID_C, the mv_A'which S generated with $mv_A$is not the same, and so X cannot pass the authentication of S. Therefore, the proposed protocol resists impersonation attacks.

5.3 The proposed protocol resists known-plaintext attacks

The 3PSKE mode is used to resist known-plaintext attacks from a malicious initiator or responder. As the initiator and the responder employ the same attack mode, we present the example of initiator A attempting to use known-plaintext to retrieve the authenticated key of B. To A, sk_A is its known-plaintext. Assuming that A intercepts ex_sk_B from the communication channel between the server and B, and attempts to use sk_A to retrieve the authenticated key of B, it encounters two unknown parameters:sk_S and s_B, as $sk_S \oplus s_B = ex\_sk_B \oplus sk_A$. Therefore, it is virtually impossible for A to infer the authenticated key of B. Based on the above analysis; we conclude that the proposed protocol resists known-plaintext attacks.

5.4 Security properties

The comparison of security properties among the protocols (the proposed, Yang and Chang's, and Tan's protocols) is shown in Table 2. The proposed protocol provides mutual authentication, known-key security, and key-compromise impersonation also provided by the other two protocols. However, the proposed protocol is able to further cope with the unknown key-share attack, while the other two protocols are not.

Table 2. Comparison of the security properties

Security properties	Ours	Yang and Chang	Tan
Mutual authentication	Yes	Yes	Yes

Known-key security	Yes	Yes	Yes
Key-compromise impersonation	Yes	Yes	Yes
Unknown key-share	Yes	No	No

Ⅷ. Performance analysis

This section examines the performance of the proposed protocol from two perspectives: communication cost and on-line computational cost, and compares the results with the ones produced by using different 3PAKE protocols (as shown in Table 5).

6.1 Communication cost

1. Round efficiency

The results show the proposed protocol requires only 3 rounds, so do Chen et al.'s, Yang and Chen's, and Tan's protocols. we can see that the proposed protocol only requires three rounds, the same as with Chen et al.'s, Yang and Chang's ,and Tan'sprotocols.

2. The size of the transmitted messages

To measure the size of the transmitted messages, we use the assumptions in Yang and Chang's protocol and Chen et al.'s protocol that the output size in the hash function is 128 bits, the timestamp field is 96 bits and the block in the secure secret key cryptosystem is 128 bits. Accordingly, the bit-lengths of both Round 1 and Round 2 in the proposed protocol are 128*6+96*2 bits, and that of Round 3 is 4*128 bits. Therefore, the total size of the transmitted message is 1,472 bits. Compared with the Yang and Chang's, Tan's, and Jaung's protocols, the one proposed requires larger size of the messages.

6.2 On-line computational cost

The proposed protocol includes both the initial and sub-key exchange stages. Only when a new user registers on the server will the initial phase need any computational cost, and thus the on-line computational cost of the proposed protocol is limited to sub-key exchange stage. The symbols used to analyze the computational cost are shown in Table 3. The relationships of the various computational costs are derived from the literature [21][22][23] and shown in Table 4.

Table 3. The symbols of used to analyze the computational cost

Symbol	Definition
T_{EXP}	The time needed for one modular exponential operation
T_{ECM}	The time needed for the multiplication of a number over an elliptic curve
T_{MUL}	The time needed for one modular multiplication
$T_{E/D}$	The time needed for one symmetric encryption/decryption
T_H	The time needed for one hash function operation
$T_{\oplus}$	The time needed for one XOR operation

Table 4. Relationships of computational costs

1 $T_{EXP} \approx 240\ T_{MUL}$
1 $T_{ECM} \approx 29\ T_{MUL}$
1 $T_H \approx 0.4\ T_{MUL}$
1$T_{E/D} \approx 2.25\ T_H$
$T_{\oplus}$ is negligible

Disregarding $T_{\oplus}$, the computational costs for the various parties in the proposed protocol are shown in Table 5 Both initiator A and responder B are 4_{TH}, and that of the server is $2T_{E/D} + 4T_H$. Therefore, the total computational cost of the proposed protocol is approximately $2T_{E/D} + 12T_H$.

Table 5.Comparisons among different 3PAKE protocols

Communication cost	Computational cost	SY/

3PAKE	R	Size (bits)	T_{EXP} A/B/S	T_{ECM} A/B/S	$T_{E/D}$ A/B/S	T_H A/B/S	Total	ASY
Chen et al.[19]	3	5824	2/2/0	0/0/0	0/0/0	4/4/6	4 T_{EXP} +14 T_H	ASY
Lv et al.[3]	4	11648	2/2/2	0/0/0	3/4/3	1/1/2	6 T_{EXP} +10 $T_{E/D}$ +4 T_H	ASY
Yang and Chang	3	1152	0/0/0	5/5/2	2/2/4	0/0/0	12 T_{ECM} +8 $T_{E/D}$	ASY
Tan	3	832	0/0/0	5/5/2	2/2/4	0/0/0	12 T_{ECM} +8 $T_{E/D}$	ASY
Jaung [24]	5	1024	0/0/0	0/0/0	4/4/4	2/2/4	12 $T_{E/D}$ +8 T_H	SY
Ours	3	1472	0/0/0	0/0/0	0/0/2	4/4/4	2 $T_{E/D}$ +12 T_H	SY

R Rounds SY/ASY(Symmetric/ Asymmetric encryption)

According to Table 5, Yang and Chang's protocol and Tan's protocol use the elliptic curve cryptography in substitution for the higher-cost modular exponential operation to reduce the overheads produced from the computation by session users. However, to follow the trend of ubiquitous cloud-computing, the proposed protocol is based on the XOR operation to further reduce the cost of protocols using the elliptic curve cryptography for overcoming the limitation of equipment with low computation capability. Under the circumstance that cryptography techniques is the same, the proposed protocol carries out the session key agreement with fewer rounds and reduces the computational cost for both A and B by 14TH (based on 1TE/D ≈2.25 TH) in comparison with the use of Jaung's protocol. Furthermore, the proposed protocol reduces the overall computational cost by 18.5 TH comparing with Jaung's protocol.

Althoughthe size of thetransmitted messages using the proposed protocol is larger than using Yang and Chang's, Tan's, and Jaung' protocols, However, the overall computational cost of the proposed protocol is less thanthat of the other 3 protocols.In terms of communication cost, we believe that the current limitations on communication cost will be gradually overcome as more network bandwidthbecomes available in the future.In terms of computational cost, it is likely that widely-used devices will continue to have relatively limited computing power, as ubiquitous and cloud computing become more popular. Therefore, since the proposed protocol requires less computational cost to implement 3PAKE, it can better meet the demands of these emerging computing service models.

Ⅸ.Conclusions

The proposed protocol presented in this work requires far less computational cost than Yang and Chang's and Tan's protocols, and can also overcome their weaknesses with regard to impersonation attacks. The proposed protocol is also better able to meet the computational demands of devices with low computing power. Finally, the protocol can also prevent the shared authenticated key from being compromised, as occurs with the traditional 3PAKE protocols, and thus it is expected to be very useful for 3PAKE applications in cloud and ubiquitous computing environments.

REFERENCES

[1] Lu, R., and Cao, Z.,"Simple three-party exchange protocol," Computers & Security,Vol. 52, No. 1, pp. 94-97, 2007.

[2] Lee, T. F., and Hwang, T., "Simple password-based three-party authenticated key exchange without server public keys," Information Sciences,Vol. 180, No. 9, pp.1702-1714, 2010.

[3] Lv, C., Ma, M., Li, H., Ma, J., and Zhang, Y.,"An novel three-party authenticated key exchange protocol using one-time key,"Journal of Network and Computer Application, Vol. 36, No. 9, pp.498-503, 2013.

[4] Yoon, E. J.,"Verifier-based computation- and communication-efficient 3EKE protocols," Mathematical and Computer Modelling, In Press, Corrected Proof, Available online 20, 2012.

[5] Henricksen, K., and Indulska, J.,"Developing context-aware pervasive computing applications: Models and approach," Journal of Pervasive and Mobile Computing,Vol. 2, No. 1, pp. 37–64, 2006.

[6] Shi,Q., Zhang, N., and Jones,D.L.,"Efficient autonomous signature exchange on ubiquitous networks,"Journal of Network and Computer Application, Vol. 35, No. 6, pp.1793-1806, 2012.

[7] Marin,L.,Jara,A. J.,and Skarmeta,A. F. G., "Shifting Primes: Extension of pseudo-Mersenne primes to optimize ECC for MSP430-based Future Internet of Things devices,"in: LNCS,Springer-Verlag Vol. 6908, pp.205-219, 2011.

[8] Khan,A.N.,Kiah,M.Mat., Khan, S.U., and Madani,S.A.,"Towards secure mobile cloud computing: a survey,"Future Generation Computer Systems,Vol. 28,No. 3, pp.583-592, 2012.

[9] Yang, J.H., and Chang, C.C., "An efficient three-party authenticated key exchange

protocol using elliptic curve cryptography for mobile-commerce environments,"The Journal of Systems and Software,Vol.82,No. 9, pp.1497-1502, 2009.

[10] Tan,Z.,"AnEnhanced Three-Party Authentication Key Exchange Protocol for Mobile Commerce Environments," Journal of Communications, Vol. 5, No. 5, pp.436-443, 2010.

[11] Nose, P.,"Security weaknesses of authenticated key agreement protocols,"Information Processing Letters,Vol. 111, No. 14, pp.687-696, 2011.

[12] Diffie, W., and Hellman, M., "New directions in cryptography," IEEE Trans Inform Theory, Vol. 22, pp.644–54, 1976.

[13] Wilson, S. B., and Menezes, A., "Authenticated Diffie-Hellman key agreement protocols," in:LNCS, Springer-Verlag, Vol. 1556, No. 630, pp.339-361, 1999.

[14] Bellovin, S. M., and Merrit, M., "Encrypted key exchanged: Password-based protocols secure against dictionary attacks," in: Proc. of IEEE Symposium on Security and Privacy,pp.72–84, 1992.

[15] Abdalla,M., and Pointcheval,D.,"Simple Password-based encrypted key exchange protocols," in:LNCS, Springer-Verlag, Vol. 3386, pp.191-208, 2005.

[16] Steiner, M., Tsudik, G., and Waidner, M., "Refinement and extension of encrypted key exchange," ACM SIGOPS Operating Systems Review, Vol. 29, No. 3, pp.22-30, 1995.

[17] Lin,C.L., Sun,H. M., and Hwang, T.,"Three-party encrypted key exchange: attacks and a solution," ACM SIGOPS Operating System Review Vol. 34, No. 4, pp.12-20, 2000.

[18] Sun,H.M., Chen,B.C., and Hwang, T., "Secure key agreement protocols for three-party against guessing attacks," The Journal of Systems and Software,Vol. 75, No. 1-2, pp.63-68, 2005.

[19] Chen, T.Z.,Lee, W.B., and Chen, H.B., "A round- and computation-efficient three-party authenticated key exchange protocol,"The Journal of Systems and Software, Vol. 81, No. 9, pp.1581-1590, 2008.

[20] Pu,Q., Zhao,X., and Ding, J., "Cryptanalysis of a three-party authenticated key exchange protocol using elliptic curve cryptography," in: International Conference on Research Challenges in Computer Science,pp.47-53, 2009.

[21] Wang,R.C., Juang,W.S. andLei, C.L.,"A Web Metering Scheme for Fair Advertisement Transactions," International Journal of Security and Its Applications, Vol. 2, No. 4, pp.49-56, 2009.

[22] Liao,Y.P., and Wang,S.S.,"A secure dynamic ID based remote user authentication scheme for multi-server environment,"Computer Standards & Interfaces, Vol. 31, No. 1, pp.24-29, 2009.

[23] Li, Z., Higgins, J., and Clement, M., "Performance of finite field arithmetic in an elliptic curve cryptosystem,"IEEE Computer Society, pp.249-258, 2001.

[24] Jaung, W.S., "Efficient three-party key exchange using smart cards,"IEEE Transactions on Consumer Electronics, pp.619-624, 2004.