Chaotic Modeling and Simulation (CMSIM) 3: 221-231, 2014

Key agreement protocol based on extended chaotic maps with anonymous authentication

Ping Zhen[1], Geng Zhao[2], Lequan Min[3] and Xiaodong Li[2]

[1] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, 100083, China
(E-mail: zhenping1989@126.com)
[2] Beijing Electronic Science and Technology institute, Beijing, 100070, China
(E-mail: zg@besti.edu.cn, lxd@besti.edu.cn)
[3] School of Mathematics and Physics, University of Science and Technology Beijing, Beijing, 100083, China
(E-mail: minlequan@sina.com)

Abstract. Key agreement protocol is used to establish shared secret key for the network system, which is quite important to guarantee secure communication. This paper proposes a two-party key agreement protocol. In order to improve the efficiency and enhance the security, we utilize extended chaotic maps to generate the shared key, which can be used to encrypt and decrypt the transmitted messages in the subsequent communications. The proposed protocol can guarantee anonymity of user's identity and provide mutual authentication. In addition, it also can resist various attacks. The explicit analysis show that the protocol is secure, reliable and applicable in practice.
Keywords: Key agreement protocol, Chaotic maps, Anonymous authentication.

1 Introduction

Key agreement protocols are basic to modern cryptography, which are used to guarantee the security of secret keys which are exchanged over the insecure public network. The shared keys are used in the subsequent communication for encryption, authentication, access control, and so on. In 1976, Diffie and Hellman[1] introduced the first key agreement protocol. However, both of communication parties don't verity the identity of each other and it is vulnerable to man-in-the-middle attack. In order to solve the problem, an authenticated key agreement protocol[2] is proposed. The authenticated key agreement not only allow two parties to agree on a session key, but also ensure the authentication of the participant. Since then, many related key agreement protocols have been proposed[3-5].

Chaotic systems have complicated behaviors, which are sensitive to initial conditions and system parameters, and are not predictable in the long term. These properties, as required by several cryptographic primitives, render chaotic systems a potential candidate for constructing cryptosystem. The application of

Received: 14 July 2014 / Accepted: 30 July 2014
 ISSN 2241-0503

chaotic maps in cryptography has been studied for more than twenty years. There are chaos-based symmetry key cryptosystem[6,7], public key cryptosystem[8,9], Hash functions [10,11], and so on.

In 2005, Xiao et al.[12] proposed a chaos-based key agreement protocol, which utilizes Chebyshev chaotic maps. Alvarez[13] demonstrated this protocol is vulnerable to man-in-the-middle attack. Xiao et al.[5] proposed an improved key agreement to enhance the security, but Han et al.[14] pointed out the improved protocol cannot resist the replay attack. Tseng et al.[15] proposed an anonymous key agreement protocol using smart cards. Niu et al.[16] demonstrated the protocol is vulnerable to the insider attacker and cannot protect user anonymity and then proposed a new key agreement protocol, which is also proved to have low computational efficiency problem by Yoon[17].

Recently, Tan[18] proposed a novel authenticated key agreement protocol with strong anonymity, which is based on smart cards. However, the expense of smart cards and readers will make the protocols costly in practical use. In Ref.[19], Gong et al. proposed a secure chaotic maps-based key agreement protocol without using smart cards and claimed that the protocol is secure. Wang et al.[20] pointed out that there are some problems existing in Gong et al.'s protocol, such as the stolen-verifier attack, forged message flood and key management problems. Then they proposed a new key agreement protocol. We have explicitly analyzed Wang et al.'s protocol. The protocol cannot provide the anonymity of users' identities. But in many insecure channels, especially in e-commerce applications, anonymity is also an very important issue. There also exits key distribution and management problems, which can be easily avoided. Lee et al.[21] proposed a three-party password-based authenticated key exchange protocol with user anonymity. However, the introduced trusted third party not only adds extra overhead, but also becomes another security and performance bottleneck, which will bring potential threats to the system. Motivated by this, this paper proposed a two-party key agreement protocol with anonymous authentication. an anonymous authenticated key agreement protocol based on extended chaotic maps to solve these problems. It doesn't need smart cards and at the same time preserves user anonymity. Besides, "two-party" will decrease the computation and communication cost and at the same time make the protocol secure and efficient. Explicit security analysis and performance analysis of the proposed protocol are also given in this paper.

This paper is organized as follows. Section 2 introduces the preliminaries about extend Chebyshev chaotic maps. Then the proposed two-party key agreement protocol is described in section 3. Security and performance analysis are given in section 4 and section 5 separately. The last section presents the conclusions.

2 Preliminaries

Definition 1. Let $n \in Z^+$ and $x \in [-1,1]$, then a Chebyshev polynomial 0 of order n, $T_n(x):[-1,1] \to [-1,1]$ is defined as:

$$T_n(x) = \cos(n \cdot \arccos(x))$$

It is recursively defined using the following recurrent relation:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), n \ge 2$$

where $T_0(x) = 1$ and $T_1(x) = x$.

The first few Chebyshev polynomials are

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x$$

$$T_4(x) = 8x^4 - 8x^2 + 1$$

$$\ldots$$

The Chebyshev polynomials exhibit the following important properties: the semigroup property and the chaotic property.

(1) The semi-group property:

$$\begin{aligned} T_r(T_s(x)) &= \cos(r\cos^{-1}(\cos(s\cos^{-1}(x)))) \\ &= \cos(rs\cos^{-1}(x)) \\ &= T_{sr(x)} \\ &= T_s(T_r(x)) \end{aligned}$$

r and s are positive integer numbers and $x \in [-1,1]$.

(2) The chaotic property

When the degree $n > 1$, the Chebyshev polynomial map $T_n(x):[-1,1] \to [-1,1]$ of degree n is a chaotic map with its invariant density $f^*(x) = 1/(\pi\sqrt{1-x^2})$, and positive Lyapunov exponent $\lambda = \ln n > 0$.

To improve security, Zhang[22] proved that the semi-group property holds for extend Chebyshev polynomials defined on $(-\infty, +\infty)$, which can enhance the property, as follows:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \bmod P$$

where $n \ge 2$ and P is a large prime. We can also obtain:

$$T_r(T_s(x)) \equiv T_{sr}(x) \equiv T_s(T_r(x)) \bmod P$$

Definition 2 The discrete logarithm problem (DLP) is explained by the following: Given an element y, the task of DLP is to find the integer s, such that $T_s(x) = y$.

Definition 3 The Diffie-Hellman problem (DHP) is explained by the following: Given the elements $T_r(x)$ and $T_s(x)$, the task of DHP is to compute $T_{rs}(x)$.

It is generally believed that there is no polynomial time algorithm to solve the DLP and DHP problems with non-negligible probability.

Table 1. The notations in the protocol

Notations	Descriptions
ID_i	Identity of client U_i
ID_S	Identity of server S
$E_k(\cdot), D_k(\cdot)$	Secure symmetric encryption and decryption
$H(\cdot)$	Secure one-way hash function
$T_k(\cdot)$	Cheybeshev chaotic map
x	The seed of Chebyshev chaotic map
r, s, r_1, r_2	The degree of Chebyshev chaotic map
PW_i	Password of client U_i
K_S	The secret key of server S
T_1, T_2, T_3	Time stamps
$\Delta T_1, \Delta T_2$	The specified valid time period
sn	The session identifier
KA	The established shared session key

3 The proposed protocol

This section will present our proposed two-party key agreement protocol based on extended Chebyshev chaotic maps. It consists of four phases: (1) the parameter generation phase; (2) the registration phase; (3) the key agreement phase; (4) the password updation phase. For the easy understanding of subsequent content, the commonly used notations are listed in Table 1.

1. Parameter generation phase

In order to perform the protocol, the server S firstly needs to generate some parameters as follow:

(1) S selects a secure symmetric cryptosystem with encryption $E_k(\cdot)$ and decryption $D_k(\cdot)$, where k is the key of symmetric cryptosystem;

(2) S selects a secure one-way hash function $H(\cdot)$;

(3) S select a private key K_S ,which is specialized for client registration.

(4) Utilizes the public key cryptosystem based on Cheybshev chaotic maps, S chooses two random large integers x and s as the seed and degree of Chebyshev maps respectively and computes $T_s(x)$. Then publish $(x, T_s(x))$ as the public parameters and keep s private.

2. Registration phase

The Client U_i with the identity ID_i registers with server S by the following two steps:

(1) U_i selects a password PW_i ,and sends the ID_i and PW_i to S through a secure channel.

(2) After receiving ID_i and PW_i , S use its private key K_S to computes $M_{reg} = H(ID_i, PW_i, K_S)$ and store M_{reg} as the register message securely.

3. Key agreement phase

The client and server need to perform the following four steps to realize mutual authentication and establish a common session key to complete the protocol. The simplified description of the phase is shown in Fig.1. The details are described in the following steps:

(1) $U_i \rightarrow S$: $M_1 = \{T_{r_1}(x), C_1 = E_{SK}(sn, ID_i, ID_S, PW_i, T_{r_1}(x), T_1)\}$.

U_i selects a random large integer r_1, and computes $T_{r_1}(x)$ and $SK = T_{r_1}(T_s(x))$. SK is used as the temporary key of symmetric cryptosystem to compute $C_1 = E_{SK}(sn, ID_i, ID_S, PW_i, T_{r_1}(x), T_1)$, where sn is a session identifier and T_1 is a timestamp. Then U_i sends the message $M_1 = \{T_{r_1}(x), C_1\}$ to the server.

(2) $S \rightarrow U_i$: $M_2 = \{sn, C_2 = E_{SK}(sn, T_{r_2}(x), H_1 = H(KA, ID_S), T_1)\}$.

After receiving the message M_1, S first compute $SK = T_s(T_{r_1}(x))$ and use it to decrypt C_1. Then S checks whether $|T_2 - T_1| \le \Delta T_1$,where T_2 is the current timestamp and ΔT_1 is the specified valid time period. S continues to compute $M_{reg}' = H(ID_i, PW_i, K_S)$ and validates whether $M_{reg}' = M_{reg}$. If so, S can authenticate the identity of client U_i , otherwise, the process will be terminated immediately. S selects a random large integer r_2, and computes $T_{r_2}(x)$, $KA = T_{r_2}(T_{r_1}(x))$, $H_1 = H(KA, ID_S)$ and $C_2 = E_{SK}(sn, T_{r_2}(x), H(KA, ID_S), T_1)$. S sends the message $M_2 = \{sn, C_2\}$ to the client.

(3) $U_i \rightarrow S$: $M_3 = \{sn, H_2 = H(sn, ID_i, KA)\}$.

Upon receiving the message M_2 from S , U_i first decrypts C_2 with the secret key SK . Then U_i checks whether $|T_3 - T_1| \le \Delta T_2$,where T_3 is the current timestamp. U_i computes $KA = T_{r_1}(T_{r_2}(x))$ and $H_1' = H(KA, ID_S)$, and validates whether $H_1' = H_1$. If so , U_i will authenticate the identity of S . Any fail will lead to the termination of the protocol. U_i continues to compute $H_2 = H(sn, ID_i, KA)$ and sends $M_3 = \{sn, H_2\}$ to the server.

(4) Having received the message M_3 from the client U_i, S will compute $H_2' = H(sn, ID_i, KA)$ and check whether $H_2' = H_2$. If so, the server S can affirm that U_i has received KA and KA will be the common session key used in the subsequent communications.

4. Password updation phase

If the client U_i want to update the password, U_i and S need to perform the following steps:

(1) U_i selects a random large integer r, and computes $T_r(x)$ and $K_{PW} = T_r(T_s(x))$. Similar with the first step in key agreement phase, K_{PW} will be used as the temporary key of symmetric cryptosystem. Then U_i encrypts $C_{PW} = E_{K_{PW}}(ID_i, PW_i, PW_i', T_r(x))$ and sends and $M_{PW} = \{T_r(x), C_{PW}\}$ to the server, where PW_i' is the updated password.

(2) Having received the message M_{PW} from U_i, S firstly computes $K_{PW} = T_s(T_r(x))$ and decrypts M_{PW}. Then S checks the validity of ID_i and PW_i. If so, then S continues to computes $M_{reg}' = H(ID_i, PW_i', K_S)$ and store M_{reg}' as the updated register message securely.

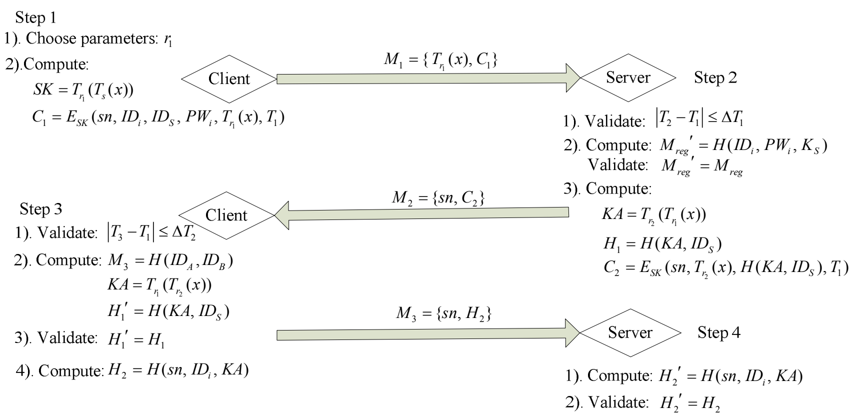

Fig. 1. The key agreement phase of the proposed protocol

4 Security analysis

In this section, we will analyze the security of the proposed protocol and show it can resist various attacks. Here, we claim that our protocol satisfy the following security properties:

(1) **Identity anonymity** With the popularization of internet application, identity privacy has become an important requirement. Identity anonymity means that in the key agreement phase, the attacker cannot find the information about user's ID by intercepting the communication messages. The attacker may eavesdrop the communication channel and try to find some sensitive information to trace the real identity. In the proposed protocol, the identity of Client and Server are encrypted by secure symmetric cryptosystem $C_1 = E_{SK}(sn, ID_i, ID_S, PW_i, T_{r_1}(x), T_1)$. In order to decrypt , the attack need the temporary secret key , which involve the DHP difficult problem mentioned in section 2. Only the server can decrypt the message and get the identity information. Thus, anonymity can be achieved during the key agreement phase.

(2) **Mutual authentication** The goal of mutual authentication is to confirm both the identities of the client and server and establish a common shared session key between them. In step 2 of the key agreement phase, only the server can decrypt the message $C_1 = E_{SK}(sn, ID_i, ID_S, PW_i, T_{r_1}(x), T_1)$ and authenticate the identity of the client by comparing the ID_i and PW_i with registered message M_{reg}. Client can authenticate the identity of server by the session identifier sn and comparing hash value $H_1' = H(KA, ID_S)$. The illegal attacker may modify the communication messages being transmitted over an insecure network. It is extremely difficult for the attacker to fabricate the false authentication information and any message modification during transmission will be detected by the protocol participant. So the proposed protocol can achieve the mutual authentication.

(3) **Resistance to tamper attacks** A tamper attack is an attempt by an adversary to modify information in an unauthorized manner. This is an attack against the integrity of the information. We have stressed the problem in the analysis above and will explain how our protocol can resist this attack in this part. In the key agreement phase, the session identifier sn and $T_{r_1}(x)$ are transmitted in the plaintext form and ciphertext form, respectively, which is used to validate whether the plaintext or cipherctext is being tampered. What is more, hash function is also utilized to further realize message integrity. If the adversary forges the message, the receiver can detect it by checking Hash value immediately. This leads to the termination of the protocol. According to the analysis, our protocol can resist the tamper attacks.

(4) **Fairness in the key agreement** The property fairness in the key agreement is also called the contributory property, which means that the session key is determined cooperationally by both the communicating parties. In 0, the author has given a strictly formal definition. The fairness in key agreement means that any communicating party cannot decide a shared session key in advance. In this protocol, we can see client and server choose random integers r_1 and r_2 separately. Through the commutative property of extended Chebyshev chaotic map, they can compute the shared session

key $KA = T_{r_1}(T_{r_2}(x)) = T_{r_2}(T_{r_1}(x))$. Therefore, the protocol can ensure the fairness in the key agreement.

(5) **Resistance to man-in-the-middle attack** Man-in-the-middle means that an active attacker intercepts the communication messages between communication participants and adopts some special means to successfully masquerade as the both parties communicate with each other. From previous analysis, the attack even doesn't know the identities of communicating parties since they are kept anonymous and any modification to the transmitted message will be detected. So the attacker cannot impersonate one participant to another during key agreement process. Therefore, the proposed protocol can withstand man-in-the-middle attack.

(6) **Resistance to replay attack** A replay attack is an offensive action in which an adversary impersonates or deceives another legitimate participant through the reuse of information obtained in a protocol. The proposed protocol can resist the replay attacks, which is realized by using the session identifier sn and time stamps (T_1, T_2, T_3). Time stamp is attached to verify freshness of every transmitted message. Furthermore, it cannot be modified because it is encrypted during transmission process. Thus, it is impossible for the replayed message to pass the verification with incorrect session identifier and timestamp. Therefore, our protocol can resist replay attack.

(7) **Resistance to password-based attacks** Dictionary attack is always used to crack the password in the protocol. There are three kinds of dictionary attack[21]: Off-line dictionary attack, undetectable on-line dictionary attack and detectable on-line dictionary attack. Both off-line and undetectable on-line dictionary attack can cause serious consequences among them. In the key agreement phase, the attacker needs to decrypt the message $C_1 = E_{SK}(sn, ID_i, ID_S, PW_i, T_{r_1}(x), T_1)$ to steal the password PW_i. To obtain the secret key SK, the attack faces the DHP difficult problem. So the attacker cannot launch any of these attacks. Therefore, our protocol is quite effective to resist password-based attacks.

(8) **Resistance to stolen-verifier attack** Then stolen-verifier attack means that an adversary who steals the password verification information from the server can use it directly to masquerade as a legitimate user in authentication phase[16]. In the protocol, we assume the registered message $M_{reg} = H(ID_i, PW_i, K_S)$ is safely stored by the server and cannot be accessed by the attacker. Even if it is stolen, the attacker still cannot carry out the stolen-verifier attack to get the client's password PW_i without the server's secret key K_S. So the secret key K_S can strength the security of password and resist the stolen-verifier attack.

(9) **High efficiency in key distribution and management** It need Server S to publish its public parameters $(x, T_s(x))$ and store the registered value $M_{reg} = H(ID_i, PW_i, K_S)$. Each entity only needs to keep his own password PW_i. This will improve the performance of the key distribution.

What's more, the symmetric secret keys SK are established temporarily utilizing the Chebyshev semigroup property and will be altered in each session according to the selected random numbers r_1 .So the communication entity does not need to store SK and it can decrease the key management cost and strengthen the security.

5 Performance analysis

In this section, we will compare the performance and security of our protocol with Tseng et al.'s protocol[15] and Wang et al.'s protocol[20]. For the convenience of evaluating the computational complexity, let T_X, T_S, T_C and T_H be the computation cost of one XOR operation, one symmetric encryption/decryption operation, one Chebyshev polynomial computation and one Hash operation, respectively. From table 2, we can see that our key agreement protocol need $(T_S + T_C)$ more computation cost for the client and $(T_S + T_C + T_H)$ more for the server than Wang et al.'s. In practical use, symmetric encryption/decryption and hash function can be quite efficient. As for the Chebyshev operation, the authors in[5,24,25] gave some implementation methods to decrease the computational cost. Our protocol provides user anonymity and can be more efficient in key distribution and management compared to Wang et al.'s protocol. What's more, our two-party protocol can decrease the communication cost. Our protocol only needs 3 times message transmission, which the number is 4 in Wang et al.'s protocol.

Table 2: Performance analysis and comparisons

	Tseng et al.'s	Wang et al.'s	Our protocol
User anonymity	No	No	Yes
Mutual authenticity	No	Yes	Yes
Fairness	Yes	Yes	Yes
Man-in-the-middle attack	No	No	No
Replay attack	No	No	No
Password-based attack	No	No	No
Stolen-verifier attack	No	No	No
Cost of Client	$2T_X + 2T_S + 2T_C + 5T_H$	$T_s + 2T_C + 2T_H$	$2T_s + 3T_C + 2T_H$
Cost of Server	$T_X + 2T_S + 2T_C + 3T_H$	$T_s + 2T_C + 2T_H$	$2T_s + 3T_C + 3T_H$

Conclusions

In this paper, we propose a two-party key agreement protocol based on extended chaotic maps. It securely establishes a shared session key, and provides identity anonymity and mutual authentication at the same time. It is demonstrated that the protocol can resist various attacks, such as man-in-the-middle attack, replay attack, stolen-verifier attack, and so on. The protocol is also very efficient in key distribution and management. Compared with some previously proposed protocols, our protocol has shown its advantage in security and efficiency, which can be applicable in practical use. However, the two-party party protocol may not be suitable in large peer-to-peer network situations, which still needs further research.

Acknowledgements

The authors would like to thank the anonymous reviewers for helpful comments and suggestions. This research is supported by the National Natural Science Foundation of China (No. 61170037) and the specialized Research Fund for Doctoral Program of Higher Education of China (No. 06198016).

References

1. W. Diffe, Hellman, and M. E., New directions in cryptography, IEEE Trans. Inf. Theory, vol.22, no.6, pp. 644-654, 1976, doi:10.1109/TIT.1976.1055638.
2. M. Bellare, D. Pointcheval and P. Rogaway, Authenticated key agreement secure against dictionary attacks, Advances in Cryptography, Eurocrypt'00, Bruges, Belgium. LNCS, 2000, vol. 1807, pp. 139-155.
3. T. Y. Change, W. P. Yang and M. S. Hwang, Simple authenticated key agreement and protected password change protocol, Comput. Math. Appl., vol.49, no.5-6, pp.703-714, April-May 2005, doi: 10.1016/j.camwa.2004. 11.007.
4. T. Y. Change, M. S. Hwang, and W. P. Yang, A communication efficient three-party password authenticated key exchange protocol, Inf. Sci., vol.181, no.1, pp.217-226, January 2011, doi: 10.1016/j.ins.2010. 08.032.
5. D. Xiao, X. F. Liao and S. J. Deng, A novel key agreement protocol based on chaotic maps, Inf. Sci., vol.177, no.4,15, pp. 1136-1142, February 2007, doi: 10.1016/j.ins.2006.07.026.
6. H. Liu and X. Wang, Color image encryption based on one-time keys and robust chaotic maps, Computers & Mathematics with Applications, vol.59, no.10, pp.3320-3327. May 2010, doi: 10.1016/j.camwa.2010.03.017.
7. W. Zhen, H. Xia, L. Ning and S. X. Na, Image encryption based on a delayed fractional-order chaotic logistic system, Chin. Phys. B, , Vol. 21, No.5, 2012, doi: 10.1088/1674-1056/21/5/050506.
8. B. Ranjan, Novel public key encryption technique based on multiple chaotic systems, Phys. Rev. Lett, vol.95, no.9, pp. 098702, September, 2005, doi:10.1103/Phys RevLett. 95.09870.
9. L. Kocarev and Z. Tasev, Public-key encryption based on Chebyshev maps, In: Proceedings of the IEEE international symposium on circuits system, 25-28 May 2003, vol.3, pp.28-31, doi: 10.1109/ISCAS.2003.1204947.
10. Y. Wang, X. F. Liao, D. Xiao D and K. W. Wong, One-way Hash function construction based on 2D coupled map lattices, Inf. Sci., vol.178, no.5, pp.1391–1406, March 2008, doi: 10.1016/j.ins.2007.10.008.

11. M. Amin, O. S. Faragallah, A. A. A. El-latif, Chaos-based Hash function (CBHF) for cryptographic applications, Chaos Solitons & Fractals, vol.42, no.2,30, pp.767–772, October 2009, doi: 10.1016/j.chaos.2009. 02.001.
12. D. Xiao, X. F. Liao and K. W. Wong, An efficient entire chaos-based scheme for denial authentication, Chaos Solitons & Fractals, vol.23, no.4, pp.1327-1331,February 2005, doi: 10.1016/j.chaos.2009.02.001.
13. G. Alvarez, Security problems with a chaos-based denial authentication scheme, Chaos Solitons & Fractals, vol.26, no.1, pp.7-11, October 2005, doi: 10.1016/j.chaos. 2004.12.023.
14. S. Han and E. Chang, Chaotic map based key agreement with/out clock synchronization, Chaos Solitions & Fractals, vol.39, no.3, pp.1283-1289, February 2009, doi: 10.1016/j.chaos.2007.06.030.
15. H. Tseng, R. Jan, and W. Yang, A chaotic maps-based key agreement protocol that preserves user anonymity, In: IEEE Inter. Conf. Commun., ICC'09, Dresden, Germany, 14-18 June 2009, pp. 1-6, doi: 10.1109/ICC.2009.5198 581.
16. Y. Niu and X. Y. Wang, An anonymous key agreement protocol based on chaotic maps, Commun. Nonlinear Sci. Number. Simul., vol.16, no.4, pp.1986-1992, April 2011, doi: 10.1016/j.cnsns.2010.08.015.
17. E. J. Yoon, Efficiency and security problems of anonymous key agreement protocol based on chaotic maps, Commun. Nonlinear Sci. Number. Simul., vol.17, no.717, pp.2735-2740, July 2012, doi: 10.1016/j.cnsns. 2011.11.010.
18. Z. Tan. "A chaotic maps-based authenticated key agreement protocol with strong anonymity," Nonlinear Dyn., vol.72, no.1-2, pp. 311-320, April 2013, doi: 10.1007/s11071-012-0715-5.
19. P. Gong, P. Li and W. Shi, A secure chaotic maps-based key agreement protocol without using smart cards, Nonlinear Dyn., vol.70, pp. 2401-2406, 2012, doi: 10.1007/s11071-012-0628-3.
20. X. Y. Wang and D. P. Luan, A secure key agreement protocol based on chaotic maps, Chin. Phys. B., vol.22, no.11, 2013, doi: 10.1088/1674-1056/22/11/110503.
21. C. C. Lee, C. T. Li and C. W. Hsu, A three-party password-based authenticated key exchange protocol with user anonymity using extened chaotic maps, Nonlinear Dyn., Vol.73, no.1-2, pp.125-132, July 2013, doi: 10.1007/s11071-013-0772-4.
22. L. Zhang, Cryptanalysis of public key encryption based on multiple chaotic systems, Chaos Solitons & Fractals, vol.37, no.3, pp.669-674, August 2008. Doi: 10.1016/j. chaos.2006.09.047.
23. A. C. Yao and Y. Zhao, Computationally-Fair group and identity-based key-exchange, Proceedings of the 9th international conference on Theory and Application of Models of Computation, TAMC'12, Beijing, China, May 16-21, 2012, pp.237-247, doi: 10.1007/978-3-642-29952-0_26.
24. X. Y. Wang and J. F. Zhao, An improved key agreement protocol based on chaos, Commun. Nonlinear Sci. Number. Simul., vol.15, no.12, pp.4052-4057, December 2010, doi: 10.1016/j.cnsns.2010.02.014.
25. Z. H. Li, Y. D. Cui and H. M. Xu, Fast algorithms of public key cryptosytem based on Chebyshev polynomials over finite field, The Journal of China Universities of Posts and Telecommunications, vol.18, no.2, pp.86-93, April 2011, doi: 10.1016/S1005-8885(10)60049-0.