Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **2**, March **2016**

An Enhanced Dragonfly Key Exchange Protocol Using Chaotic Maps

Yang Sun, Hongfeng Zhu* and Xueshuai Feng

Software College
Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034 China
17247613@qq.com; zhuhongfeng1978@163.com; 1275064307@qq.com
*Corresponding author

Received May, 2015; revised November, 2015

Abstract. *Dragonfly is a key exchange protocol based on a password authentication. This protocol has been submitted to the IETF and became a candidate standard for general internet use. However, Harkins analysed the security of the protocol and designed an algorithm to attack Drangonfly key exchange protocol. This algorithm attacks the protocol successfully in a polynomial time. In this paper, we propose a key exchange protocol called Improved Dragonfly key exchange protocol. We use chaotic maps to design the algorithm. The proposed protocol based on CMBDLP and CMBDHP, and use multiplication in finite field algorithm to replace the traditional method of chaotic maps-symmetric cryptography for achieving high-efficiency. Therefore, this proposed protocol is more secure, more efficient, and more practical compared with the old Dragonfly protocol, and can resist the attack raised by Harkins specialy.*

Keywords: Key exchange, Mesh networks, Chaotic maps, Dragonfly protocol.

1. **Introduction.** Mesh network is one of the network topologies, which can transmit data on the network by mesh nodes. Mesh network is also known as multi hop networks, and it is a dynamic and extensible network structure. In this network, all nodes cooperate in the transmission of data. A mesh networks can be divided into two categories: wired mesh network and wireless mesh network. Recently, wireless mesh network is pretty popular. Wireless mesh network is a self-organizing network, the nodes on the network are connected by wireless links and they realize the transmission between wireless devices. And the wireless mesh network is better than the old wireless LAN. In conventional wireless LAN, if the user wants to communicate with others, they must firstly visit a fixed access point (AP), the network which use this method is called single hop networks. However, on mesh network (multi hop networks), any wireless device node can be used as the AP or router. The advantage of this network is that if the nearest node congested because of large flow, the data can also select a node which is in a small flow to finish the whole transmission. So the process of the transmission in the network situation can be described as follows, data from a node to some other multiple nodes, and then reach the final destination. This method mentioned above is called multi-hop access. According to the advantages and superiority of mesh networks, more and more researchers begin to design protocols on mesh network.

Many protocols have been used on the mesh networks[1-3]. In 2009, Clancy et al [1] came up with the idea of EAP-GPSK, the method is a lightweight shared-key authentication protocol. And the next year, in 2010, Kaufman et al. proposed the IKEv2[2], it is a scheme that use a password as a proof-of-knowledge of the password. However, all of these protocols cannot resist the dictionary attack. To solve this problem, in 2012, D.Harkins[3] wrote a document. In that document, he created a key exchange protocol called Dragonfly key exchange protocol based on discrete logarithm, the idea also has been mentioned in [4] early. And D.Harkins claimed Drangonfly protocol can resist active attack, passive attack, and off-line dictionary attack.

However, the Dragonfly protocol also has weaknesses, and the protocol has been attacked by Dylan Clarke [5] in 2013, they point out that the attackers computation raised by D.Harkins is not the best way to attack Dragonfly protocol. Instead, if attackers select a generator of a small subgroup rather than a group generator [5], they can use offline dictionary attacks to attack Dragonfly protocol. So we think some methods to improve the security of Dragonfly key exchange protocol. Conventional chaotic maps is a concept in mathematics. Chaotic maps can be parametric by both discrete time and continuous time. Nowadays, chaotic maps is widely used in the key agreement protocol because chaotic maps have many advantages, such as certainty, boun dedness, unpredictability and the randomness. Besides, chaotic maps can also produce a random phenomena, and this phenomena is pseudo-random. So chaotic maps do not need some mathematical calculation, such as timestamp, modular-exponentiation, elliptic curve [6], etc, can also resist the common attacks, such as active attack, passive attack, dictionary attack etc.

In this paper, we use the properties of chaotic maps to improve the security of Drogonfly protocol. We combine Dragonfly protocol with chaotic maps, and make sure that the improved Dragonfly protocol is securer and more efficient than the old one. Our main contributions are shown as below: (1) We effectively improve the security and efficiency of the Dragonfly protocols by using chaotic maps. (2) Our scheme can real resist active attacks, passive attacks, even the offline dictionary raised by Dylan Clarke. (3) Our schemes practicability, stability, security is better than old one.

The rest of the paper is organized as follows: In the next section, we review some preliminaries. Sect. 3 we describe the old dragonfly protocol and analyse the problem in that protocol. And in Sect. 4, describe our proposed scheme. Sect. 5 and 6 discuss the security and efficiency of the proposed scheme. Finally, the paper is concluded in Sect. 7.

2. Preliminaries.

In this part, the concepts of mesh networks and Chebyshev chaotic maps will be introduced.

2.1. Mesh networks.

A mesh network is a network topology. In this topology, each node relays data for the network. Mesh networks can transmit messages using either a flooding technique or a routing technique. With routing, the information is along a transmission path by hopping from node to node until it reaches its destination. The network must allows continuous connections and must reconfigure itself around broken paths in case some path be not available, so there is some typical self-healing algorithms such as Shortest Path Bridging on the mesh networks. Self-healing allows a routing-based network to operate when a node breaks down or when a connection becomes unreliable. As a result, the network is typically quite reliable, as there is often more than one path between a source and a destination in the network. Although mostly used in wireless situations, this concept can also apply to wired networks and to software interaction.

A mesh network is a fully connected network. Fully connected wired networks have the advantages of security and reliability: problems in a cable affect only the two nodes

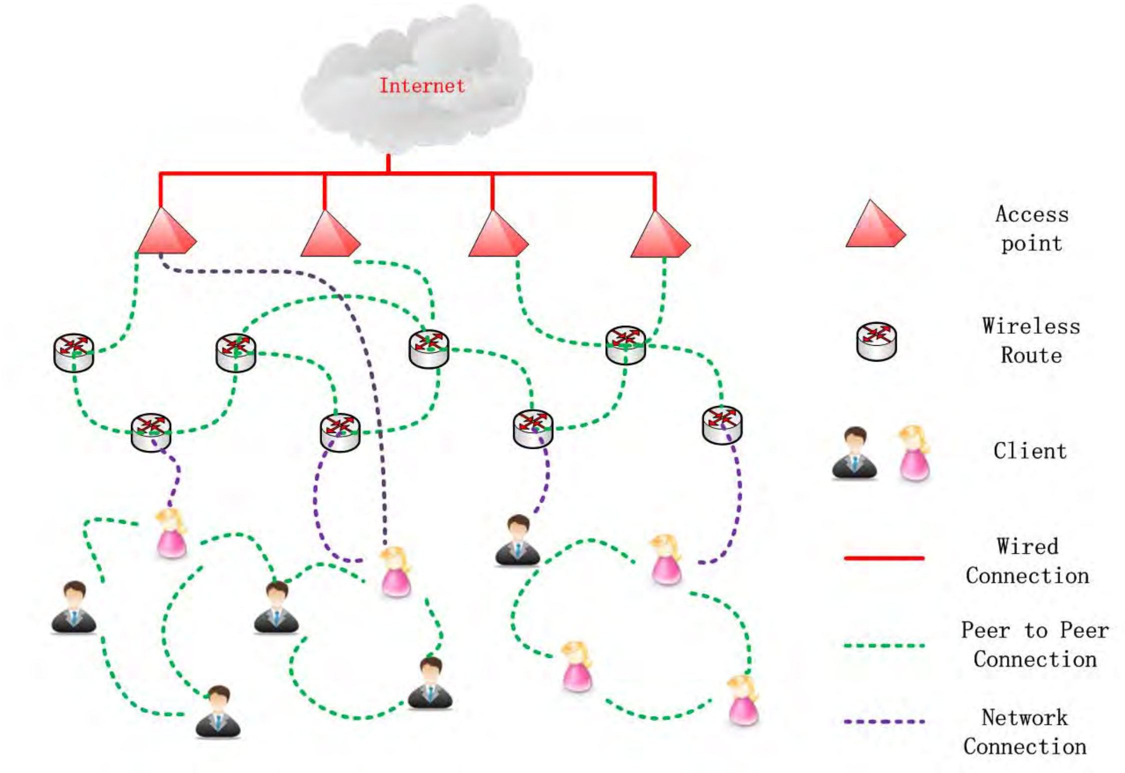


FIGURE 1. Mesh Network Architecture

attached to it. However, in such networks, the number of cables, and therefore the cost, goes up rapidly as the number of nodes increases.

Mesh networks can be considered a type of an ad hoc network. Thus, mesh networks are closely related to mobile ad hoc networks (MANETs), although MANETs also must deal with problems introduced by the mobility of the nodes.

Mesh networks architecture is made up of three parts: access point, wireless router, clients, and they work with each other. The facilities (access point, wireless router) make sure the connectivity in different networks. And the clients access the Internet via wireless router. In addition, the clients have routing capabilities so they can make a network by themselves. Figure 1 is display the architecture of mesh network.

2.2. **Definition and properties of Chebyshev chaotic maps.** Let be an integer and let be a variable with the interval $[-1, 1]$ .The Chebyshev polynomial The Chebyshev polynomial $T_n(x) : [-1, 1] \to [-1, 1]$ is defined as $T_n(x) = cos(narccos(x))$. Chebyshev polynomial map $T_n : R \to R$ of degree is defined using the following recurrent relation [15]:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \ , \qquad (1)$$
$$\text{where } n \geq 2 \ , \ T_0(x) = 1 \ , \text{ and } T_1(x) = x \ .$$

The first few Chebyshev polynomials are:

$$T_2(x) = 2x^2 - 1, T_3(x) = 4x^3 - 3x, T_4(x) = 8x^4 - 8x^2 + 1, ...$$

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that

$$T_r(T_s(x)) = T_{r \cdot s}(x) \qquad (2)$$

An immediate consequence of this property is that Chebyshev polynomials commute under composition

$$T_r(T_s(x)) = T_s(T_r(x)) \qquad (3)$$

In order to enhance the security, Zhang [16] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$. The enhanced Chebyshev polynomials are used in the proposed protocol:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x))(mod\ N) \qquad (4)$$

where $n \geq 2$, $x \in (-\infty, \infty)$, and N is a large prime number. Obviously,

$$T_{r \cdot s}(x) = T_r(T_s(x)) = T_s(T_r(x)) \qquad (5)$$

Definition 2.1. *Semi-group property of Chebyshev polynomials:*

$$T_r(T_s(x)) = cos(rcos^{-1}(scos{-}1(x))) = cos(rscos^{-1}(x)) = T_{sr}(x) = T_s(T_r(x))$$

Definition 2.2. *Given x and y, it is intractable to find the integer s, such that $T_s(x) = y$. It is called the Chaotic Maps-Based Discrete Logarithm problem (CMBDLP).*

Definition 2.3. *Given x, $T_r(x)$ and $T_s(x)$, it is intractable to find $T_{rs}(x)$. It is called the Chaotic Maps-Based Diffie-Hellman problem (CMBDHP).*

3. **Original Dragonfly protocol.**

Original Dragonfly protocol is a password authenticated key exchange protocol which is based on discrete logarithm cryptography. So the operations of Dragonfly protocol can be also used on an elliptic curve or a finite field. It defines two users, and then we can draw a flow of Dragonfly protocol. Dragonfly protocol can be described as follow steps will be shown in Figure 2.

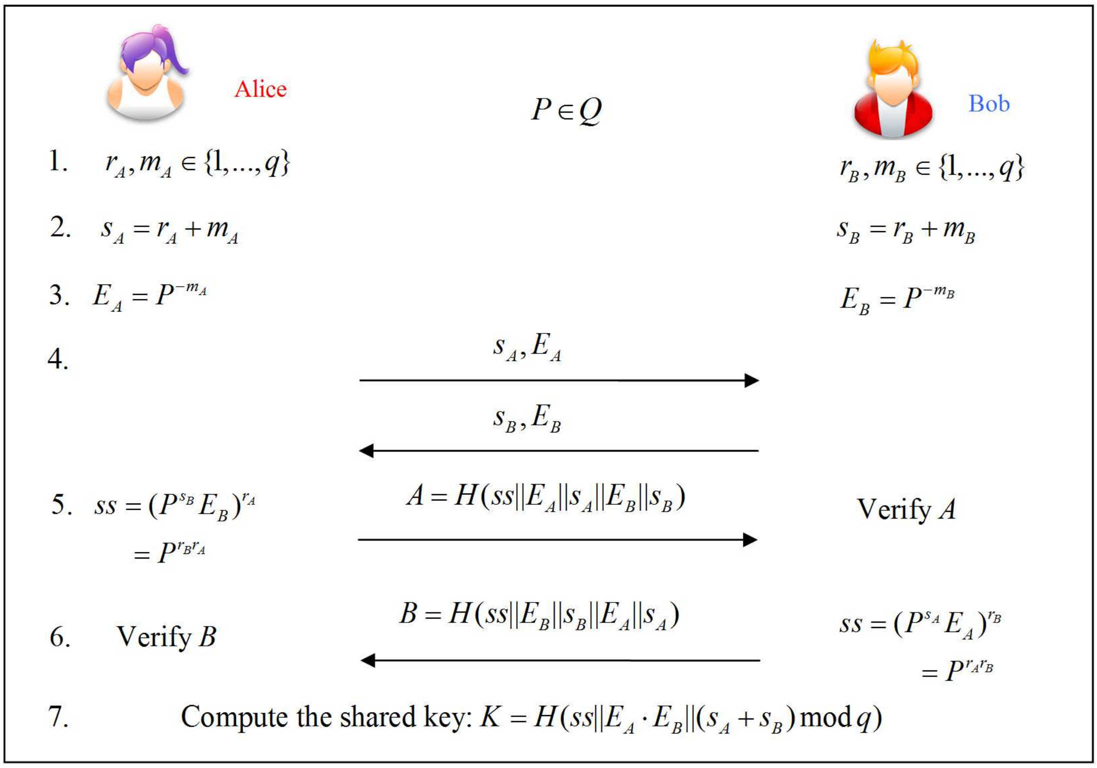

Alice
$P \in Q$
Bob
1. $r_A, m_A \in \{1,\dots,q\}$
$r_B, m_B \in \{1,\dots,q\}$
2. $s_A = r_A + m_A$
$s_B = r_B + m_B$
3. $E_A = P^{-m_A}$
$E_B = P^{-m_B}$
4.
s_A, E_A
s_B, E_B
5. $ss = (P^{s_B} E_B)^{r_A} = P^{r_B r_A}$
$A = H(ss \| E_A \| s_A \| E_B \| s_B)$
Verify A
6. Verify B
$B = H(ss \| E_B \| s_B \| E_A \| s_A)$
$ss = (P^{s_A} E_A)^{r_B} = P^{r_A r_B}$
7. Compute the shared key: $K = H(ss \| E_A \cdot E_B \| (s_A + s_B) \bmod q)$

FIGURE 2. The Flow of Dragonfly Protocol

The Dragonfly protocol works as follows:

(1) Alice and Bob will share a password. The password can deterministically generate as $P \in Q$.

(2) Alice randomly chooses two scalars r_A, m_A from 1 to q , and then computes the $s_A = r_A + m_A$,$E_A = p^{-m_A}$. And Alice sends s_A, E_A to Bob. By the same token, Bob will also do the same operations like Alice, and sends s_B, E_B to Alice.

(3) Alice and Bob will calculate the shared secret key $ss = (P^{s_B}E_B)^{r_A} = P^{r_B r_A}$ and $ss = (P^{s_A}E_A)^{r_B} = P^{r_A r_B}$ respectively.

(4) Alice and Bob use cryptographic hash function, Alice sends $A = H(ss||E_A||s_A||E_B||s_B)$ to Bob, and Bob sends $B = H(ss||E_B||s_B||E_A||s_A)$ to Alice.

(5) Alice and Bob verify each other and check the hashes are correct or not, if they are, they create a shared key $K = H(ss||E_A \cdot E_B||(S_A + S_B) mod\ q)$.

As mentioned above, we can find some questions during the process of the secret key negotiation:

(1) D.Clarke points that there is no assumption are made about the underlying group. And he also thinks the computation of discrete logarithms is not sufficiently for the level of security required. In the article writen by D.Clarke, he describes a method and a set of implementations to attack this protocol.

(2) The attack method that proposed by D.Clarke is dictionary attack. When crack the passwrod or key, the adversary attempts the combination of letters and numbers untill they make a correct combination and make sure the password or key.

(3) The attacker then uses three steps to launch an off-line dictionary attack [5]: 1) to obtain the victims password element P; 2) to forge a valid response B to bypass authentication (so the victim is unaware that the password has been compromised); 3) to compromise the secrecy of communication by deriving the session key K.

Therefore, we think some ways to extend the field and increase the number of the random scalars to resist the dictionary attack. We will introduce them in the section 4.

4. **The Proposed Protocol.** In this part, we will illustrate the improved Dragonfly protocol scheme in detail. Firstly, we formulate why our proposed scheme can resist dictionary attack. Secondly, illustrate why our proposed scheme is more efficient than the old one. In case of the problems appear in the old dragonfly protocol come again, we take a set of measures.

(1) Firstly, We use HPW instead of the old password element P. We have introduced that P belongs to the finite cyclic group Q and it is easily to be guessed. So we take a predefined cryptographic hash function $H(ID_A||ID_B||PW)$, and return the function value as the value of HPW . Then when an adversary makes a dictionary attack, he will waste more time to guess this password.

(2) We add a high entropy variable x and n, compare with the old Dragonfly protocol, this measure can make the new Dragonfly key exchange protocol has a high security.

(3) Make the random scalars r_A, r_B, m_A and m_B to become the number of the Chebyshev polynomial, the form is $E = T_m T_{HPW} T_r(x)$. This measure makes the dictionary attack can not be sucessful thoroughly. Because an adversary must guess two input values HPW (HPW is a low-entropy secret) and m (m is a high-entropy secret absolutely), it is infeasible.

Remark 4.1. *Most of chaotic maps-based protocols for achieving key agreement or encrypted messages usually adopt Chaotic Maps-Based Diffie-Hellman (CDH) problem to get the same session key to encrypting/decrypting messages transferred between user and server [9, 15]. But our proposed scheme only uses CDH problem to get temporary key for attaching messages to it, which can make our scheme more efficient, and the users privacy information is protected. In other words, we change the usage of chaotic maps from the form $E_{T_aT_b(x)}(messages)$ to another form $T_aT_b(x) \cdot messages$, obviously, the latter is much more efficient than the former.*

(4) We take Chebyshev chaotic maps instead of discrete logarithms. The Chebyshev polynomial $T_n(x) : [-1, 1] \to [-1, 1]$ is defined as $T_n(x) = cos(narccos(x))$. It provides smaller key sizes, faster computation, as well as memory, energy and bandwidth savings. So our protocol is more efficient. About the specific analysis of security and efficiency, we will illustrate in section 5 and 6.

Then we will introduce the improved Dragonfly protocol by steps in Figure 3.

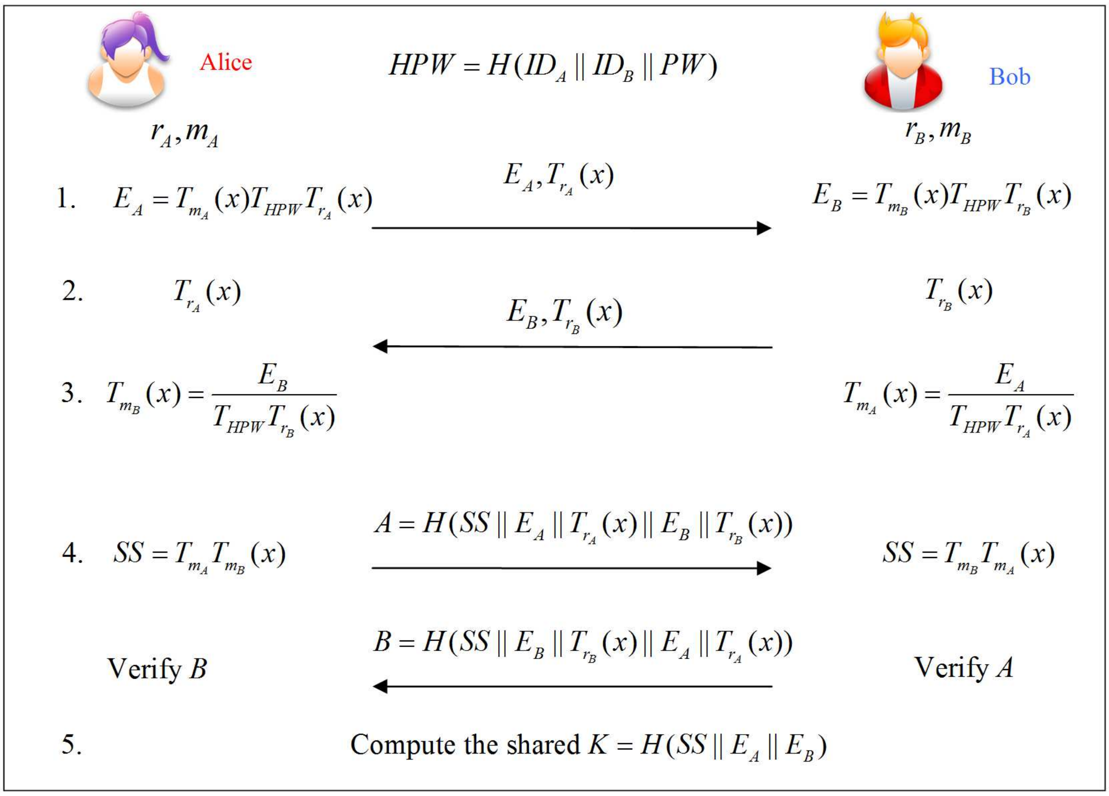

Alice
$HPW = H(ID_A \| ID_B \| PW)$
Bob
r_A, m_A
r_B, m_B
1. $E_A = T_{m_A}(x) T_{HPW} T_{r_A}(x)$
$E_A, T_{r_A}(x)$
$E_B = T_{m_B}(x) T_{HPW} T_{r_B}(x)$
2. $T_{r_A}(x)$
$E_B, T_{r_B}(x)$
$T_{r_B}(x)$
3. $T_{m_B}(x) = \frac{E_B}{T_{HPW} T_{r_B}(x)}$
$T_{m_A}(x) = \frac{E_A}{T_{HPW} T_{r_A}(x)}$
4. $SS = T_{m_A} T_{m_B}(x)$
$A = H(SS \| E_A \| T_{r_A}(x) \| E_B \| T_{r_B}(x))$
$SS = T_{m_B} T_{m_A}(x)$
Verify B
$B = H(SS \| E_B \| T_{r_B}(x) \| E_A \| T_{r_A}(x))$
Verify A
5. Compute the shared $K = H(SS \| E_A \| E_B)$

FIGURE 3. Improved Dragonfly Protocol

(1) Alice and Bob have a shared password PW and compute $HPW = H(ID_A || ID_B || PW)$.

(2) Alice randomly chooses two scalars r_A and m_A , then Alice computes $E_A = T_{m_A}(x) T_{HPW} T_{r_A}(x)$ and sends $E_A, T_{r_A}(x)$ to Bob. Bob will do the similar operations concurrently. Bob computes $E_B = Tm_B(x) T_{HPW} T_{r_B}(x)$ and sends $E_B, T_{r_B}(x)$ to Alice.

(3) Alice computes $T_{m_B}(x) = \frac{E_B}{T_{HPW} T_{r_B}(x)}$ and sends $A = H(SS||E_A||T_{r_A}(x)||E_B||T_{r_B}(x))$ to Bob, then Bob computes $T_{m_A}(x) = \frac{E_A}{T_{HPW} T_{r_A}(x)}$ and verifies the numerical value of A.

(4) Alice calculates the shared secret $SS = T_{m_A} T_{m_B}(x)$. Then it is same to Bob. And Bob will calculate the shared secret $SS = T_{m_B} T_{m_A}(x)$.

(5) Bob sends $B = H(SS||E_B||T_{r_B}(x)||E_A||T_{r_A}(x))$ to Alice, and Alice verifies the numerical value of B.

(6) Alice and Bob check that verifications are correct or not, and if they are then they create a shared key $K = H(SS||E_A||E_B)$.

Our protocol can resist common attacks, and also can resist the attack methodology proposed by Dylan Clarke [5]. In his attack implementation, successful search the hash value of A and the value of s_B, E_B is the key of the algorithm. But in our protocol, it is impossible to search the value of E_B and r_B , because the difficulty of CMBDLP and CMBDHP. The security analysis of other common attacks will be mention in the next part.

5. **Security Analysis.** In this section, we analyse the security of the improved Dragonfly protocol.

5.1. **Perfect forward secrecy.**

Definition 5.1. *An authenticated multiple key establishment protocol provides perfect forward secrecy if the compromise of both the nodes secret keys cannot results in the compromise of previously established session keys [17].*

Theorem 5.1. *The proposed scheme can realize perfect forward secrecy.*

Proof: In this proposed scheme, the session key SS is related with r_A, r_b, m_A and m_B, which were chosen by Alice and Bob randomly. So even adversary can obtain the value of r_A, r_B, m_A, m_B in last time, and he can not compute the r_A, r_B, m_A, m_B this time. So he cant compute the share secret key without the value of these parameters. According to the above, our protocol can achieve perfect forward secrecy.

5.2. **Known-key secrecy.**

Definition 5.2. *A protocol can protect the subsequent session keys from disclosing even if the previous session keys are intercepted by the adversaries, what will not affect other session keys is called known-key security.*

Theorem 5.2. *The proposed scheme can realize known-key security.*

Proof: As m_A,m_B are independent and different in all sessions, if the adversary knows a session key $SS = T_{m_A}T_{m_B}(x)$ and a pair random numbers m_A and m_B, he can also not compute the previous or the future session keys without knowing the previous or the future random numbers m_A and m_B.

5.3. **Password guessing attack.**

Definition 5.3. *Password guessing attack is an attack in which an adversary can guess and confirm the password of user in a system or in a communication protocol.*

Theorem 5.3. *The proposed scheme can resist password guessing attack.*

Proof: If the adversary starts a password guessing attack, he will fail because the key is compute with $K = H(SS||E_A||E_B)$, and in the process of the calculation include four random and large variables r_A,m_A,r_B and m_b.

Password guessing attack can only crack a function with one low entropy variable (password), so if we at least insert one large random variable which can resist this attack. And our protocol has some high entropy variables r_A,m_A,r_B,m_b with HPW to makeup one functional expression (such as $E_A = T_{m_A}(x)T_{HPW}T_{r_A}(x)$). Moreover, when we use Chebyshev chaotic maps, which the form is $T_n(x) = cos(narccos(x))$, we also have high entropy variables x and n. So this attack can not be sucessful.

5.4. **Replay attack.**

Definition 5.4. *A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed.*

Theorem 5.4. *The proposed protocol can resist replay attack.*

Proof: Our protocol can resist replay attack, because the anonymous of m_A,m_B, they never appear in the process of transmission. So the adverary can never obtain m_A,m_B. In addition, every time the value of m, r are not same, so the adversary can not replay the message in the data transmission which be encrypted by our protocol.

5.5. Session key security.

Definition 5.5. *A communication protocol exhibits session key security if the session key cannot be obtained without any long-term secrets.*

Theorem 5.5. *The proposed protocol can achieve session key security.*

Proof: In the authenticated key agreement phase, a session key K is generated from m_A,m_B,r_A and r_B. These values are different in each session, and the values of m_A,m_B is only known by Alice or Bob. Whenever the communication ends between Bob and Alice, the key will immediately self-destruct and will not be reused. Therefore, assuming the attacker has obtained a session key, and Alice will be unable to use this session key to decode the information in other communication processes. Because the random elements m_A,m_B,r_A and r_B are all generated randomly and are protected by the CMBDLP, CMBDHP, and the secure symmetric encryption, a known session key cannot be used to calculate the value of the next session key.

5.6. Spoofing attack.

Definition 5.6. *A spoofing attack is an attack in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage.*

Theorem 5.6. *The proposed scheme can resist spoofing attack.*

Proof: In this scheme, Alice and Bob have to verify each other through the value of the hash function $A = H(SS||E_A||T_{r_A}(x)||E_B||T_{r_B}(x))$ and $B = H(SS||E_B||T_{r_B}(x)||E_A||T_{r_A}(x))$. Only they pass the verification, they compute the K.

5.7. Insider attack.

Definition 5.7. *An insider attack is a malicious attack perpetrated on a network or computer system by a person with authorized system access. Compare with the outsider, insider attack has a better access, which is more trusted, and has better information about internal procedures, high-value targets, and potential weak spots in the security.*

Theorem 5.7. *The proposed scheme can resist insider attack.*

Proof: We compute the password $HPW = H(ID_A||ID_B||PW)$, the ID information is a parameter of hash function, the attacker cant obtain ID directly, he can only obtain the value of hash function which is no use. So our protocol can resist insider attack.

5.8. Denial-of-service attack.

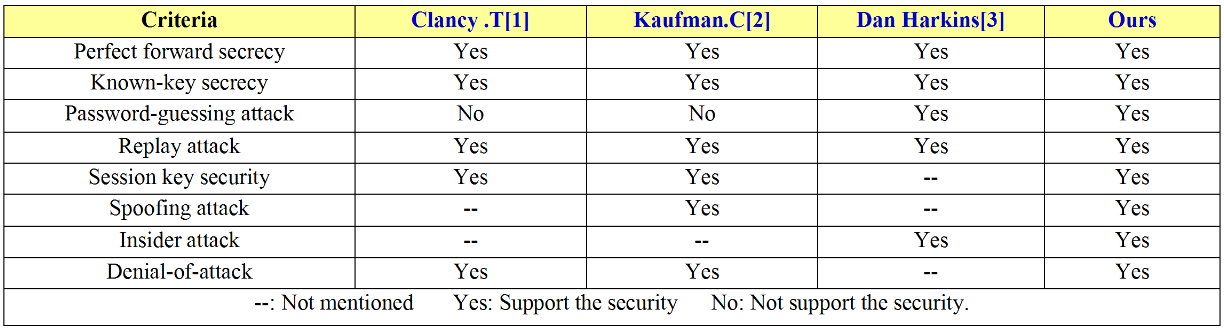
Definition 5.8. *A denial-of-service attack is an attempt to make a machine or network resource unavailable to its intended users.*

Theorem 5.8. *The proposed scheme can resist denial-of-service attack.*

Proof: Chebyshev polynomial computation problem offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings. So it hardly burdens the gateway CPU.

Table 1 can show the security comparisons between our scheme and recent related schemes especialy the precious Dragonfly protocol vividly.

Table 1. Security comparisons between our scheme and related scheme



Criteria	Clancy .T[1]	Kaufman.C[2]	Dan Harkins[3]	Ours
Perfect forward secrecy	Yes	Yes	Yes	Yes
Known-key secrecy	Yes	Yes	Yes	Yes
Password-guessing attack	No	No	Yes	Yes
Replay attack	Yes	Yes	Yes	Yes
Session key security	Yes	Yes	--	Yes
Spoofing attack	--	Yes	--	Yes
Insider attack	--	--	Yes	Yes
Denial-of-attack	Yes	Yes	--	Yes

--: Not mentioned Yes: Support the security No: Not support the security.

6. **Efficiency Analysis.** Wang [7] proposed several methods to solve the Chebyshev polynomial computation problem. To be more precise, on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, the computational time of a one-way hashing operation, a symmetric encryption/decryption operation, an elliptic curve point multiplication operation and Chebyshev polynomial operation is 0.0005s, 0.0087s, 0.063075s and 0.02102s separately [18]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations. According to the results in [19], one pairing operation requires at least 10 times more multiplications in the underlying finite field than a point scalar multiplication in ECC does in the same finite field. Through the above mentioned analysis, we can reached the conclusion approximately as follows:

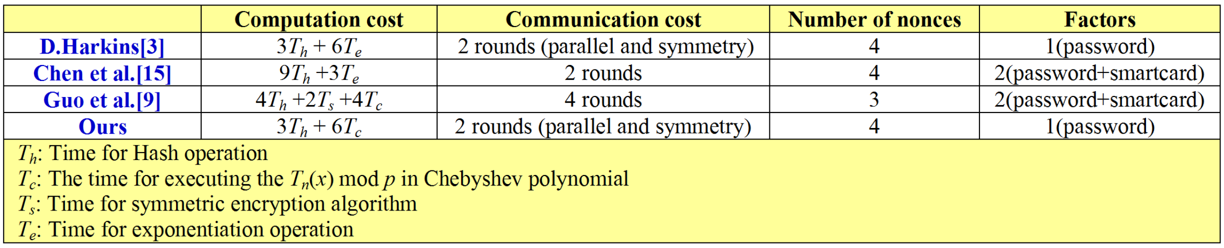
$$T_p \approx 10T_m, T_m \approx 3T_c, T_c \approx 2.42T_s, T_s \approx 17.4T_h,$$

we sum up these formulas into one so that it can reflect the relationship among the time of algorithms intuitively.

$$T_p \approx 10T_m \approx 30T_c \approx 72.6T_s \approx 1263.24T_h,$$

where: T_p: Time for bilinear pair operation, T_m: Time for a point scalar multiplication operation, T_c: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial, T_s: Time for symmetric encryption algorithm, T_h: Time for Hash operation. According to [7, 18, 19], the execution times of each phase in our proposed scheme and related schemes are shown in Table 2.

Table 2. Efficiency comparisons between our scheme and related scheme in the authentication phase



	Computation cost	Communication cost	Number of nonces	Factors
D.Harkins[3]	$3T_h + 6T_e$	2 rounds (parallel and symmetry)	4	1(password)
Chen et al.[15]	$9T_h + 3T_e$	2 rounds	4	2(password+smartcard)
Guo et al.[9]	$4T_h + 2T_s + 4T_c$	4 rounds	3	2(password+smartcard)
Ours	$3T_h + 6T_c$	2 rounds (parallel and symmetry)	4	1(password)

T_h: Time for Hash operation
T_c: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial
T_s: Time for symmetric encryption algorithm
T_e: Time for exponentiation operation

7. **Conclusion.** In this paper, we propose an improved Dragonfly key exchange protocol which based on chaotic maps. This scheme enjoys some good properties such as simple, efficent, and good performance. Review of the whole article, we use chaotic maps instead of discrete logarithm (modular exponentiation and scalar multiplication on elliptic curves)

firstly. Then, we introduce the knowledge about the mesh network, chaotic maps, and the old dragonfly protocol. Finally, we compare the new scheme with the old one and some other protocols. According to the comparison result, we can see our protocol is more suitable for applications.

Acknowledgement. This work is supported by the Liaoning Provincial Natural Science Foundation of China (Grant No. 201602315).

REFERENCES

[1] T. Clancy and H. Tschofenig, Extensible Authentication Protocol-Generalized Pre-Shared Key (EAP-GPSK) Method,*RFC* 5433, 2009.

[2] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen, Internet Key Exchange Protocol Version 2 *(IKEv2), RFC 5996*, 2010.

[3] D. Harkins, Dragonfly key exchange - internet research task force internet draft, http://tools.ietf.org/html/draft-irtf-cfrg-dragonfly-00, 2012.

[4] D. Harkins, Simultaneous authentication of equals: A secure, password based key exchange for mesh networks, *IEEE computer society*, pp. 839-844, 2008.

[5] D. Clarke, H. Feng, Cryptanalysis of the Dragonfly Key Exchange Protocol, *IACR Cryptology ePrint Archive*, 2013.

[6] Q. Xie, J. Zhao and X. Yu, Chaotic maps-based three-party password authenticated key agreement scheme, Nonlinear Dynamics, vol.74, no.4, pp.1021-1027, 2013.

[7] X. Wang, T. Zhao, An improved key agreement protocol based on chaos, *Communications in Nonlinear Science and Numerical Simulation*, vol.15, pp.4052-4057, 2010.

[8] Lai, H. Xiao, J. Li, L. Yang, Applying semi-group property of enhanced Chebyshev polynomials to anonymous authentication protocol, *Math, Probl, Eng*, 2012.

[9] C. Guo and C.C. Chang, Chaotic maps-based password-authenticated key agreement using smart cards, *Communications in Nonlinear Science and Numerical Simulation*, vol.18, no.6, pp.1433-1440, 2013.

[10] K. Chain and W.C. Kuo, A new digital signature scheme based on chaotic maps, *Nonlinear Dynamics*, vol.74, no.4, pp.1003-1012, 2013.

[11] C.C. Lee, C.L. Chen, C.Y. Wu and S.Y.Huang, An extended chaotic maps-based key agreement protocol with user anonymity, *Nonlinear Dynamics*, vol. 69, no.1-2, pp.79-87, 2012.

[12] H. Zhu, X. Hao, Y. Zhang and Man Jiang, A Biometrics-based Multi-server Key Agreement Scheme on Chaotic Maps Cryptosystem, 2014.

[13] D. B. He, Y.T. Chen and J.H. Chen, Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol *Nonlinear Dynamics.* vol. 69, no. 3, pp.1149-1157, 2012.

[14] M. Inuma, A. Otsuka and H. Imai, Theoretical framework for constructing matching algorithms in biometric authentication systems, *Lecture Notes in Computer Science*, vol. 5558, pp.806-815, 2009.

[15] T. Y. Chen, C. C. Lee, M. S. Hwang and J. K. Jan, Towards secure and efficient user authentication scheme using smart card for multi-server environments. *J Supercomput*, vol.66, pp.10081032, 2013.

[16] M. Inuma, A. Otsuka and H. Imai, Theoretical framework for constructing matching algorithms in biometric authentication systems, *Lecture Notes in Computer Science*, vol. 5558, pp.806-815, 2009.

[17] J. Kar and B. Majhi, An Efficient Password Security of Multiparty Key Exchange Protocol based on ECDLP, *International Journal of Computer Science and Security (IJCSS)*, vol. 3, no. 4, pp. 405-413, 2009.

[18] Kocarev L, and Lian S, Chaos-Based Cryptography: *Theory, Algorithms and Applications*, pp. 5354, 2011.

[19] P. Barreto, B. Lynn, M. Scott, On the selection of pairing-friendly groups, In: Selected Areas in Cryptography, *In: LNCS, Springer-Verlag*, vol. 3006 pp. 1725, 2004.