Applied Mathematics & Information Sciences
An International Journal

An Efficient Password-based Group Key Exchange Protocol Using Secret Sharing

Wei Yuan, Liang Hu, Hongtu Li and Jianfeng Chu

School of Computer Science and Technology, Jilin University, Changchun 130012, China

Received: 13 May 2012; Revised 04 Jul. 2012; Accepted 16 Aug. 2012
Published online: 1 January 2012

Abstract: In this paper, a novel and efficent password based group key exchange protocol with secret sharing is proposed. Secret sharing technology is usually used to control the privileges of the authorized users to improve the robustness of the system in past years. The results are applied into designing the key exchange protocol directly, which clarify the proposed scheme. The security analysis shows the proposed scheme can provide the confidentiality and authentication, while being competitively efficent in comparison with other works in the literature.

Keywords: Group key exchange, password, secret sharing, interpolated polynomial.

1. Introduction

To communicate over an open channel secretly, all the participants need to share a same knowledge to differ from the outsiders, such as common session key and other secrets. If the number of participants is fixed and very small, each one can encrypt the useful messages once in a different secret and send it in an offline manner. Although the efficiency is low, this method is still feasible. However, when the number of the participants varies before the communication or the offline delivery is limited by the distance or other physical conditions, the key exchange protocols are necessary.

These protocols allow at least two participants to securely exchange their private information on open channels. In the beginning, they only know their own secret information. At the end of the protocol, they agree on a common secret but still know nothing about others' secret. Then one can use the same secret information to communicate with others secretly. We say that they own a common session key.

The key exchange protocol should guarantee the common session key computed by all participants is same and no person outside these participants can get this key even in the conditions that all messages between these participants are eavesdropped by the outsiders. Furthermore, if all the messages between the participants are intercepted and replaced, each participator can detect it and drop this session key.

The first practical key exchange protocol owe to Diffie and Hellman [1]. They divide the traditional encryption key into two kinds, public key and private key. The public key can be transmitted openly; user only needs to keep his private key as a secret. To achieve this aim, Diffie and Hellman introduced the discrete logarithm problem, which can not be solved in mathematics. For instance, if Alice wants to exchange a message with Bob, she randomly selects a secret x, and computes g^x, where g is a public parameter. Due to an efficient algorithm to compute x from g^x does not exist, she can open g^x and leave x as a secret at the same time. Then x is her private key, and is her public key. This protocol is simple and efficient. The information encrypted by the public key can only be decrypted by the private key. The eavesdropper can not get any information about the private key, because it is never transmitted on the network. But the private key x is a big number and impossible to be remembered by normal person. Then many efforts [2,3,4] have been made to construct more user friendly schemes, such as password-based protocols.

The password-based key exchange protocols require users only to remember a human-memorable low-entropy password and hope to provide the comparative intensity with the public key systems by internal cryptogrammic

* Corresponding author: Jianfeng Chu, e-mail: yuanwei1@126.com

operation. To the best of our knowledge, most password-based key exchange protocols originate from the basic DH protocol and its security basis is discrete logarithm. However, the DH protocol does not provide the authentication between the sender and receiver of the information. If the transmitted messages are replaced by the malicious attackers, they can send forged messages to cheat the normal users and both the sender and the receiver can not detect this attack. Many protocols that are based on DH protocol [5-9], especially in the wireless areas, easily suffer from this attack.

To overcome this shortage, some other mathematically hard problems are employed to design new group exchange protocols, such as large number factorization [10] and quadratic residue [11]. But the efficiency of these protocols is usually lower than those based on the DH protocol. In this paper, we mainly focus on another technology called secret sharing for group key exchange. This technology is first proposed to solve a combinatorial mathematical problem. The original problem [12] is as follows. Eleven scientists wish to lock up a document in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. How to compute the smallest number of locks needed and the number of keys each scientist must carry is the issue here. More generally, this problem can be expressed as how to divide one data into n pieces and at least k pieces are needed in order to recover it. Therefore, the privilege that used to belongs to one person is now distributed to multiple persons. For a long time, the secret sharing technology has been mainly applied in controlling the privileges of some pivotal notes to enhance the robustness of the system. In this paper, we apply it to key exchange in combination with the traditional password-based key exchange schemes. The performance analysis shows the efficiency can be greatly improved with the help of secret sharing technology.

The rest of this paper is organized as follows. The secret sharing technology is introduced in Section 2. The efficient password-based group key exchange protocol with trusted server is proposed in Section 3. The performance and security analysis of the proposed protocol are detailed in Section 4. Section 5 gives the conclusion remarks and the future works.

2. Preliminaries

In this part, we introduce the secret sharing technology. Secret sharing technology [13] is to divide a secret data into n pieces in such a way that any k or more pieces makes the secret data easily computable. However, any $k-1$ or fewer pieces leaves the secret data completely undetermined. This technology is also called (k,n) threshold technology.

This technology is based on the Lagrange interpolating polynomial and does not rely on any assumption. Generally, a trusted server is necessary in (k,n) threshold. It consists of two algorithms: secret distribution and secret reconstruction.

2.1. Secret distribution algorithm

In this phrase, the trusted server selects a polynomial function $f(x) = a_0 + a_1 x + \ldots, a_{k-1}x^{k-1}$. $a_0, a_1, \ldots, a_{k-1}$ can be any integers in particular group, in which the secret to be share is $a_0 = f(0)$. Then the trusted server computes n pieces of secret $s_i = f(i)$, $i = 1, \ldots, n$ and distributes each $s_i$to a corresponding person.

2.2. Secret reconstruction algorithm

In this phrase, at least k persons show their secret pieces, and then the secret s can be reconstructed as

$$a_0 = f(0) = \sum_{i=1}^{k} s_i (\prod_{j \neq i}^{k} \frac{x_j}{x_j - x_i}) \quad (1)$$

The two algorithms set a k variants equation and solving it. Thus, the Shamir secret sharing technology is mathematical theoretically secure.

3. Proposed protocol with trusted server

We assume the trusted server is well known to all users and no one can pretend it. The trusted server selects two large primes p and q and compute their product $n = pq$. When user registers, the trusted server shares a password with him and sends the number n to him. Suppose each user, U_i, shares a password pw_i with the server. In this protocol, U_i and pw_i are number strings and pw_i can be regarded as two parts: pw_{ix} and pw_{iy}. For example, one's pw is 12345678. We can regard pw_{ix} as 1234 and pw_{iy} as 5678. h_1 and h_2 are two unidirectional hash functions. We define all computations are on the group Z_n. Fig.1 shows the procedure clearly and the protocol runs as follows:

1. The initial user, U_1, selects a random number $k_1 \in Z_n$, computes $K_1 = pw_{1x} + k_1$, $M_1 = h_1(U_1, \ldots, U_t, k_1)$, sends $U_1, \{U_1, \ldots, U_t\}, K_1, M_1$ to the trusted server.

2. After receiving the message $U_1, \{U_1, \ldots, U_t\}, K_1, M_1$, the trusted server decrypts k_i with the password it shares with U_i and verifies whether the equation $M_1 \stackrel{?}{=} h_1(U_1, \ldots, U_t, k_1)$ holds or not. If the verification succeeds, it calls each user $\{(U_2, \ldots, U_t\}$, respectively.

3. After receiving the notification from the trusted server, $U_i, 2 \leq i \leq t$, selects a random number $k_1 \in Z_n$, computes $K_i = pw_{ix} + k_i$, and $M_i = h_1(U_1, \ldots, U_t, k_i)$ as in the first step, and finally sends $U_i, \{U_1, \ldots, U_t\}, K_i, M_i$ to the trusted server.

4. After receiving all messages from each user, the trusted server find corresponding password pw_i of U_i to

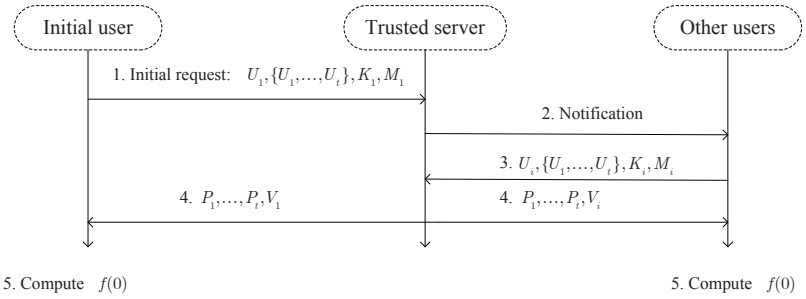

Figure 1 The process of the proposed group key exchange protocol

get each random number k_i, and verifies whether the equation $M_i \stackrel{?}{=} h_1(U_1, \ldots, U_t, k_i)$ holds. Then the trusted server selects two random number x_{ta} and y_{ta}, which have the same length with pw_{ix} and pw_{iy}, and generates an interpolated polynomial $f(x)$ with degree t to pass through $(t+1)$ points, $(pw_{1x}, pw_{1y}+k_1), \ldots, (pw_{tx}, pw_{ty}+k_t)$, and (x_{ta}, y_{ta}). At last, the trusted server computes t additional points $P_1, P_2, \ldots, P_t$ on $f(x)$, t additional verification messages $V_1 = h_2(U_1, \ldots, U_t, P_1, \ldots, P_t, k_1), \ldots, V_t = h_2(U_1, \ldots, U_t, P_1, \ldots, P_t, k_t)$ and sends $\{P_1, \ldots, P_t, V_i\}$ to U_i.

5. After receiving the message from the trusted server, each user U_i verifies whether the equation $V_i \stackrel{?}{=} h_2(U_1, \ldots, U_t, P_1, \ldots, P_t, k_i)$ holds with the kept user list $\{U_1, \ldots, U_t\}$, the received additional points $\{P_1, \ldots, P_t\}$ and the random number k_i, recovers $f(x)$ with the $(t+1)$ points $P_1, \ldots, P_t$ and $(pw_{ix}, pw_{iy}+k_i)$, and finally computes the group key $f(0)$.

4. Security and efficiency analysis of the proposed protocol

4.1. Security definition

In this protocol, we suppose that anyone in outside not only can eavesdrop on any intermediate messages between normal participator but also can intercept them and impersonate a normal participator to send forged messages. That is to say, the adversary has the entire control of the communications channel, and tries to break the privacy of the group key. Note it is the strongest assumption in secure communication areas. The potential adversaries can be sorted into two types: outside adversaries and inside adversaries. The outside adversaries are out of a particular group. The aim of their attack is to recover the group key of a particular group. Meanwhile the inside adversaries can be normal participants of a particular group. The aim of their attack is to get other participants' password shared with the trusted server. When other participator attends another particular group, these adversaries can recover the group key of that group as outside adversaries.

For outside adversaries, they success to break the privacy if they can gain any information about the group key by the messages they eavesdrop or impersonate a normal participator and are not detected by other participants. For inside adversaries, they success to break the privacy if they can gain other participator's password in the processes of the protocol executes.

To prevent above adversaries, our protocol should at least achieve the following security goals: key freshness, key confidentiality, and key authentication. Key freshness is to ensure that a group key has never been used before. With this property, a participator in previous group can not gain any useful information in other groups, even the participants are the same. Key confidentiality is to protect the group key against the outside adversaries. Key authentication is to provide assurance to authorize the messages from the participants by the trusted server and to authorize the messages from the trusted server by participants, which is to prevent the possible man-in-the-middle attacks.

4.2. Security analysis

In this part, we summarize the security properties with theorem 1 and then point out the inside attack is useless in this protocol.

Theorem1. The proposed protocol can achieve the attributes of key freshness, key confidentiality, and key authentication.

Proof. In this protocol, the group key k is determined by x_{ta}, y_{ta}, and the t numbers $k_1, \ldots, k_t$. x_{ta} and y_{ta} are random, and each k_i is selected by a participator randomly. Thus, any group key has not been used in other established group. It is obviously that the key is freshness in this protocol.

The key confidentiality in this protocol is ensured by the security feature of Lagrange interpolated polynomial. The trusted server generates a t degree polynomial $f(x)$ with $(t+1)$ points, $(pw_{1x}, pw_{1y}+k_1), \ldots, (pw_{tx}, pw_{ty}+k_t)$, and (x_{ta}, y_{ta}), and broadcasts t additional points $P_1, \ldots,$

P_t to each user. Each group user, U_i, can reconstruct $f(x)$ with the $(t+1)$ points, $P_1, \ldots, P_t$, and his secret $(pw_{ix}, pw_{iy}+k_i)$, then recovers $f(0)$. However, for the outsiders of that group, only t points $P_1, \ldots, P_t$ on $f(x)$ are known, guessing another point is impossible in mathematics. This property is information theoretically secure since we do not use any computational assumption.

The key authentication is bidirectional. The trusted server can authenticate participants' message by the shared password. Since only the trusted server knows the right password, except the corresponding participator, others can not compute k_i with K_i. Any modification on K_i may lead a failure, when the trusted server verifies the equation $M_i \stackrel{?}{=} h_1(U_1, \ldots, U_t, k_i)$, except that the adversary can break the unidirectional hash function to find the appropriate M_i. Thus, the trusted server can authenticate the messages from each participator. Each participator can authorize the trusted server's message by the random number k_i.Actually, this procedure references the one-time signature mechanism [14,15]. For k_i in $V_i = h_2(U_1, \ldots, U_t, P_1, \ldots, P_t, k_i)$ is never transmitted alone, the participator can detect it if some point P_x is modified or forged. Thus, this protocol has the attribute of key authentication

Theorem 2. Any participator can not gain other participator's password and any t-1 participants can not gain the group key.

Proof. The key confidentiality of this protocol has shown that directly attack on this protocol is invalid. However, due to the special status, the inside adversaries may initiate the key exchanges to the trusted server for many times. Thus he can select relevant random numbers in different key exchanges to compute the password of the target participator. For example, he initiates two processes of key exchange, and each one contains the target participator. He can send the same random number k_i, then the trusted server will return two different polynomials $f_1(x) = a_t x^t + \ldots + a_1 x + a_0 mod n$ and $f_2(x) = b_t x^t + \ldots + b_1 x + b_0 mod n$. We know that $pw_{iy}+k_i = f_1(pw_{ix}) = f_2(pw_{ix})$. The adversary can compute $f_3(pw_{ix}) = f_2(pw_{ix}) - f_1(pw_{ix}) = c_t x^t + \ldots + c_1 x + c_0$ and solve this equation to get $pw_i x$. However, to solve this equation, he has to solve two equations $f_3(x) = 0 mod p$ and $f_3(x) = 0 mod q$ at first. That is, he should have the ability to overcome the factoring assumption. This assumption is the basis of some well-known modern cryptosystems, such as RSA algorithm [11] and Rabin algorithm [12]. It is commonly believed that no efficient polynomial can overcome this assumption. Thus, an inside adversary can not crack this protocol by initiating key exchange process for many times.

As the insider attacks may exists, all the other participants may be regarded as the inside adversaries except the user himself. If they combine to guess the password of the target participator, we call it collusive attack. To discuss this attack, we suppose that only two users participate in the group. One is normal user, the other is the adversary. In this protocol, the password of the user consists of pw_x and pw_y. However, the password only be used by the trusted server and can not be got directly. Since K_u is open, the adversary can computes pw_x if he gets the random number k_u. Then he can guess pw_y by the polynomial $f(x)$ and pw_x . Thus, the problem to guess the password equals to guess the random number k_i. As we know, the adversary knows his random number, and the group key $f(0)$. But pw_x and pw_y are also unknown for the adversary and have never been transmitted directly. Thus, he still can not get normal user's password and the collusive attack still can not crack this protocol.

4.3. Performance analysis

In this protocol, each participator needs to select a random number, computes two hash functions in the message exchanging stage. Corresponding with the participants, trusted server needs to select two random numbers, computes $2t$ hash functions. Thus, t+2 random numbers and $4t$ hash operations are needed to exchange the transmitted messages between the server and t participants in total. When all participants gain the exchanged messages, they should compute the group with the formula (1) themselves. In this stage, each participator should run t-1 multiplications, t-1 divisions, and t addition. Duo to that the addition operation is very fast, we don't count this operation. Thus, $t(t$-1) multiplications and $t(t$-1) divisions are need in this stage. Correspondingly, the trusted server needs 2(t-1) multiplications and 2(t-1) divisions to construct the interpolated polynomial $f(x)$. The total computation cost in this stage is: t^2+t-3 multiplications and t^2+t-3 divisions. The total data are list in table 1.

To show the efficiency clearly, we list another three typical group key exchange and their computation cost are listed in the Table 1. Note, due to the management manners are different, we do not count the notification messages to all protocol. The messages can be preprocessed are also ignored. To treat all protocol fairly, the standard of calculating the computation cost is same, which causes the computation cost seemed like higher than the other paper claimed to all these protocols. For example, the sender computes a hash value of a message and passes it to the receiver. The receiver computes this value again to verify it. Then we regard it as twice computations. However, some papers regard it as once computation, and some papers regard it as twice. I hope this would not cause confusion to readers, because the comparing standard is uniform.

The protocol in reference [16] is based on the well-known Burmester and Desmedt group key exchange [17], the protocol in reference [18] is based on Horng's multi-party key establishment [19], they can both be regarded as the extensions of the basic DH two-party key exchange protocol. The security basis of them is the discrete logarithm problem. From Table 1, we can easily find that the multiplications operations of the three protocols are in the same level. The divisions operations in our protocol are

Table 1 Performance comparison with some other schemes

Contributions		Proposed protocol	Zheng et al. [18]	Abdalla et al. [16]	Jiang et al. [20]
Transmission messages		2t	4t	3t	2t
Computation costs	Power	0	4t-2	3t	2t
	Division	t^2+t-3	t	t	0
	Encryption	0	t	t	2t
	Decryption	0	2t	2t	2t
	Hash	4t	4t	6t	2t
	Random number	t+2	2t	2t	2t
	Multiplication	t^2+t-3	t^2/16	t^2	9t

higher than the others but no powers operations in our protocol. A fact should be attention: to compute 2^{100}, which the exponent is only 3, we need to at least run $log_2 100$ multiplications operations but we only need to run 2 multiplications operations to get 2×2. In addition, the bits of exponent usually are no less than 512. It means that the overhead of our protocol is lower than the other two. It should be highlight that we do not embed the symmetrical encryption and decryption algorithms into our protocol. It leads the efficiency of our proposed protocol is much lower than the others. This owns to the robustness of the basic method of interpolated polynomial.

The protocol in reference [20] is based on Yi et al.'s secure conference scheme [21], and its security basis is the quadratic residues problem. For this scheme only achieve the aim that each participator shares a secret with the trusted server but not the group key between the participants in the key establishment phase, its powers and multiplication operations are less than the other two schemes, but the encryptions and decryptions operations are still embedded into this protocol. We know the symmetrical encryptions or decryptions consist of many basic operations, such as additions, subtractions, multiplications, division, and powers operations. So its computation cost is greatly larger than these basic operations. Owe for the basic interpolated polynomial, we can replace these complex operation with the basic operations and hash operations.

5. Conclusions and future works

In this paper, we have proposed an efficient password-based group key exchange protocol using secret sharing. This protocol provides the properties of key freshness, key confidentiality, and key authentication to prevent many kinds of potential attacks. With the help of the secret sharing technology, we can replace the symmetrical encryptions or decryptions operations with the hash operations and other basic operations so that the efficiency of this protocol can be improved.

In some wireless environment, the trusted server doesn't exist or can only run in an offline manner. To protect the user's privacy, the anonymity is needed in some special situations. Thus, we will try to construct new schemes with the mentioned properties above in our future work.

References

[1] Diffie, M.E. Hellman, IEEE Transaction on Information Theory 22, 644 (1976).
[2] M. K. Boyarsky, ACM CCS 99: 6th Conference on Computer and Communications Security, (1999), pp.63-72.
[3] S. M. Bellovin and M. Merritt, 1992 IEEE Symposium on Security and Privacy, (1992), pp.72-84.
[4] M. Bellare, D. Pointcheval, and P. Rogaway, Advances in Cryptology-EUROCRYPT 2000, Lecture Notes in Computer Science 1807, (2000), pp.139-155.
[5] M. S. Hwang, IEEE Trans. Veh. Technol. 48 1469, (1999).
[6] K. F. Hwang and C. C. Chang, IEEE Trans. on Wireless Communications. 2, 400, (2003).
[7] Feng Bao, IEEE Trans. on Wireless Communications 5, 1984, (2006).
[8] C.-C. Chang and H.-C. Tsai, IEEE Trans. Wireless Commun. 9, 3346 (2010).
[9] Guomin Yang,, IEEE Trans. on Wriel. Common 10, 2015, (2011).
[10] M.O. Rabin, Technical Report LCS/TR-212, MIT Laboratory for Computer Science, (1979).
[11] R.L. Rivest, A. Shamir, and L. Adleman, Communications of ACM 21, 120, (1978).
[12] Liu, C.L. Introduction to Combinatorial Mathematics. McGrawHill, New York, (1968)
[13] A. Shamir, communications of the ACM 22,.612, (1979)
[14] Dan Boneh, Ben Lynn, Hovav Shacham, ASIACRYPT 2001, (2001), pp.514-532.
[15] L Lamport, Communications of the ACM 24, 770, (1981).
[16] M. Abdalla, E. Bresson, O. Chevassut, D. Pointcheval, PKC 2006, Lecture Notes in Computer Science 3958, (2006), pp.427-442.
[17] M. Burmester, Y. Desmedt, Information Processing Letters 94, (2005), pp.137-143
[18] M. H. Zheng, H. H. Zhou, J. Li and G. H. Cui, Computer Standards & Interfaces 31, 948, (2009).
[19] G. Horng, The Computer Journal 44, 464, (2001).
[20] Y. X. Jiang, C. Lin, M.H. Shi, X. M. Shen, IEEE J. Select. Areas Commun. 24, 1738,(2006).

[21] X. Yi, C. K. Siew, C. H. Tan, and Y. Ye, IEEE Trans. Wireless Commun..2, 1168, (2003).

Wei Yuan was born in Chengde of Hebei province of China in 1984. He began the study of computer science at Jilin University in 2003 and got his bachelor degree in 2007. Then he continued his research on information security and received his master degree in 2010. Now he is a PhD candidate of the college of computer science and technology of Jilin University. His main research interests include cryptography and information security. he have participated in several projects include two National Natural Science Foundations of China and one National Grand Fundamental Research 973 Program of China and published more than 20 research papers from 2007.

Liang Hu was born in 1968. He has his BS degree on Computer Systems Harbin Institute of Technology in 1993 and his PhD on Computer Software and Theory in 1999. Currently, he is the professor and PhD supervisor of College of Computer Science and Technology, Jilin University, China. His main research interests include distributed systems, computer networks, communications technology and information security system, etc. As a person in charge or a principal participant, Dr Liang Hu has finished more than 20 national, provincial and ministerial level research projects of China.

Li Hongtu was born in Siping of Jilin, China on Mar. 17 1984. In 2002, Li Hongtu began the study of computer science at Jilin University in Jilin, Changchun, China. And in 2006, Li Hongtu got bachelor's degree of computer science. In the same year, Li Hongtu began the master's degree study in network security at Jilin University. After 3 years study, Li Hongtu got his master's degree in 2009. From then on, Li Hongtu began the doctor's degree in the same field of study at the same University. From 2009, he has got a fellowship job. He worked in grid and network security laboratory as an ASSISTANT RESEACHER at Jilin University. From 2006 to now, he has published several papers.

Jianfeng Chu, born in 1978, Ph.D. , Now he is the teacher of the College of Computer Science and Technology, Jilin University, Changchun, China. He received the Ph.D. degree in computer structure from Jilin University in 2009. His current research interests focus on information security and cryptology. An important objective of the projects is to probe the trend of network security, which can satisfy the need of constructing high-speed, large-scale and multi-services networks. Various complex attacks can not be dealt with by simple defense. And to add mechanisms to network architecture results in decreasing performance. In a word, fundamental re-examination of how to build trustworthy distributed network should be made.