A Novel Proposal of Exchanging Key for Symmetric Key Cryptosystems

Viet Hoang Van† , Truyen Bui The††, Dung Luu Hong††† and Duc Tong Minh††††,

Le Quy Don University, Ha Noi, Viet Nam

Summary
In today's digital world, the demand for secret information transmitted between two parties over insecure channel is more and more popular. This paper proposed two new key established algorithms for symmetric key cryptosystems. The most advantage of the two new algorithms is the secret key is established requires only a single round of transportation. Moreover, the secret key can be authenticated the origin so it can effectively resistant to spoofing attacks. The article also presents the analysis and assessment of the security level of the new proposed algorithms, indicating its application posibility of practice..

Key words:
Key Establishment, Key Agreement Protocols, Key Exchange Protocol, Key Transport Protocols.

1. Introduction

In the field of information security, the symmetric key cryptosystems has significant advantages in speed of execution than the public key cryptosystems, so they are often used to encode the data block size large, especially in online transactions.
In symmetric key cryptosystems, the common key establishment for both the sender/encoder and receiver/decoder is a very important and complex issue. Key establishment can be broadly subdivided into key agreement and key transport protocol:
a) A key agreement protocol is a key establishment technique in which a shared secret is derived by two (or more) parties as a function of information contributed by, or associated with, each of these, (ideally) such that no party can predetermine the resulting value. The key agreement also known as the key exchange protocol that was first proposed by W. Diffie and M. Hellman in 1976 and called the Diffie-Hellman Key Exchange Protocol [1];
b) Key transport protocols or mechanism is a key establishment technique where one party creates or otherwise obtains a secret value, and securely transfers it to the other(s) using public key algorithm such as RSA [2] or ElGamal [3]. However, using the Diffie-Hellman Key Exchange Protocol or public key algorithm such as RSA or ElGamal in key transport protocol has a common weakness which is the inability to fight several types of spoofing attacks such as Man-In-Ihe-Middle Attack [4-6], because of the fact that they do not have the authentication mechanism of receiving messages.
The paper proposed two new key establishment algorithms in which the secret key can be authenticated the origin so it can effectively resistant to some well known spoofing attacks. In addition, the shared key established procedures by the algorithms are required only one round public key transfer so the algorithms are suitable for high speed applications.

2. New key establishment algorithms

2.1. The first algorithm

2.1.1 Procedures for system parameters and public key

The procedure consists of following steps:
1 - Select a group Zp where p is a large prime number so that the discrete logarithm problem is intractable and g is the generator element of Z_p^*.
2 - Private key x is random chosen such that: $1 < x < (p-1)$.
3 - Public key y is computed as:

$$y = (g)^x \bmod p \qquad (1.1)$$

4- Public the values of (p, g, y). Keep secret : x.

2.1.2 Key establishment procedure

Assuming the exchange key participants are A and B. A and B also agree to use the same symmetric key cryptography algorithm (eg DES, AES, ...) to encrypt information (text, document, ...) which will be exchanged with each other. A's private key is xA, the corresponding public key is yA; B's private key and public key are xB and yB respectively. The public key of A and B is formed under the procedure for system parameters and public key in Section 2.1.1. The values yA and yB need to be authenticated by a trust CA (Certificate Authority). The

Manuscript received February 5, 2016
Manuscript revised February 20, 2016

algorithm allows the parties A and B to establish a shared secret key K. It is done by following two steps below:
Step 1: Done by A.

1 - Select a random integer k such that: $0 < k < (p-2)$.
Computes R as formular:

$$R = (y_A)^{k+1} \bmod p \quad (1.2)$$

2 - Send R to B.
Step 2: Done by A and B.
1 - A computes the shared secret key KAB as formular:

$$K_{AB} = (y_B)^{(k.x_A)} \bmod p \quad (1.3)$$

2 - B computes the shared secret key KBA as formular:

$$K_{BA} = \left(R \times (y_A)^{-1}\right)^{x_B} \bmod p \quad (1.4)$$

The shared secret key K of A and B is:
$K = K_{AB} = K_{BA}$.
Note:
Using hash function H(.) can increase randomly for the shared secret key K of A and B as shown:

$$K_A = H(K_{AB})$$

and: $K_B = H(K_{BA})$

The shared secret key of A and B is: $K = K_A = K_B$
In which: KA is created by the A, and KB is created by the B.
For example: Suppose A and B agree to use AES with 128 or 256-bit encryption to exchange confidential information. Meanwhile, A and B can use the hash function such as: SHA-256 or MD5 to increase randomly for the shared secret key, as follows:
$K_A = MD5(K_{AB})$ or: $K_A = SHA\_256(K_{AB})$
and:
$K_B = MD5(K_{BA})$ or: $K_B = SHA\_256(K_{BA})$
Discussion:
In the first algorithm, the shared secret key of two communication parties of A and B is $K = K_{AB} = K_{BA} = g^{k.x_A.x_B} \bmod p$, A sends $R = (y_A)^{k+1} \bmod p$ to B. In other way, the message A sends to B is not the secret value as the key transport protocol (using RSA or El Gamal) but it is the information to form the shared key. From A's information B can create its own the shared secret key. Therefore, the proposed algorithm is not any known key transport protocols. In adddition, the information, which uses to form shared key, is created by only one side. This is different from the DH protocol which requires both sides exchange their own public keys. So the proposed key establishment algorithm is the new algorithm for symmetric key cryptosystem.

2.1.3. The correctness of the new proposed algorithm

What is to prove is: Let say p is a prime number and g is a generator element of Z_p^*, $1 < x_A, x_B < (p-1)$, $y_A = g^{x_A} \bmod p$, $y_B = g^{x_B} \bmod p$, $0 < k < (p-2)$.
If: $R = (y_A)^{k+1} \bmod p$,
$K_{AB} = (y_B)^{(k.x_A)} \bmod p$,
$K_{BA} = \left(R \times (y_A)^{-1}\right)^{x_B} \bmod p$
then:
$K_{AB} = K_{BA}$.
Proof:
Indeed, from (1.1) and (1.3) we have:

$$\begin{aligned} K_{AB} &= (y_B)^{(k.x_A)} \bmod p \\ &= \left(g^{x_B} \bmod p\right)^{(k.x_A)} \bmod p \\ &= g^{k.x_A.x_B} \bmod p \end{aligned} \quad (1.5)$$

And, from (1.1), (1.2) and (1.4) we also have:

$$\begin{aligned} K_{BA} &= \left(R \times (y_A)^{-1}\right)^{x_B} \bmod p \\ &= \left((y_A)^{k+1} \bmod p \times (y_A)^{-1} \bmod p\right)^{x_B} \bmod p \\ &= \left((y_A)^{k} \bmod p\right)^{x_B} \bmod p \\ &= (y_A)^{k.x_B} \bmod p = g^{k.x_A.x_B} \bmod p \end{aligned} \quad (1.6)$$

From (1.5) and (1.6) we have: $K_{AB} = K_{BA}$.
This concludes the proof.

2.1.4 The security of the first proposed algorithm.

The security of the new proposed algorithm can be assessed through the following capabilities:

a) Resistance to reveal secret keys attack

From (1.3) and (1.4) it can be seen that, a third communication party (attacker) wants to computes the key (K), he has to know both k and xA or xB based on solution of (1.1) and (1.2). Solving equation of (1.1) and (1.2) is the same as solving a DLP (Discrete Logarithm Problem) [4-6]. Thus, resistance to reveal secret keys attack of the new algorithms generally depending on the degree of difficulty of the discrete logarithm problem. Currently, discrete logarithm problem is considered a difficult problem if the parameter p and k, xA, xB is selected large enough to attack as "brute force" is not feasible in the practical application.

b) Resistance to spoofing attacks

Suppose C is spoofing attacker, the question is C can impersonate A to establish the shared secret key with B or C impersonate B to get the shared secret key with A by the new proposed algorithm or not?

The first case, C impersonates A to establish the shared secret key with B. First C selects a random value k satisfy: $1 < k < p-1$ then computes: $R = (y_A)^{k+1} \bmod p$. Second, C sends R to B. When receiving R, B computes the shared secret key with A as fomular (1.4) $K_{BA} = g^{k.x_A.x_B} \bmod p$. Meanwhile, C computes the shared secret key with B according to (1.3), but C do not know xA, so C has to choose a random value x_A^* in order to compute K_{AB} as: $K_{AB} = g^{k.x_A^*.x_B} \bmod p$.

Because of the fact that $x_A^* \neq x_A$ then $K_{AB} \neq K_{BA}$, it can be seen that C fails to impersonate A to set the shared secret key with B. It should be emphasized that, the selection x_A^* satisfying $x_A^* = x_A$ is excluded, because if it happens well meaning that the discrete logarithm problem has been solved.

The second case, C impersonates B to generate the shared secret key with A. Meanwhile, A chooses k such that: $1 < k < p-1$ then computes R according to equation (1.2). A sends R to B, in this case, the actually receiver is C. A computes the shared secret key KAB according to equation (1.3). Because of the fact that C impersonates B, then A uses B's public key to generate the shared secret key as: $K_{AB} = g^{k.x_A.x_B} \bmod p$. When receiving R, C calculates the shared secret key with A according to (1.4). Because C does not know B's secret key x_B then C has to choose a random value x_B^* and the probability of $x_B^* = x_B$ is very small. Therefore K_{BA} value, that C get, would be $K_{BA} = g^{k.x_A.x_B^*} \bmod p$. Because $x_B^* \neq x_B$ then clearly that $K_{AB} \neq K_{BA}$. In other words, C failed to establish the shared secret key with A. Therefore, in both cases C can not impersonate A or B.

2.2. The second algorithm

2.2.1 Procedure for system parameters

The procedure consists of following steps:

1 - Randomly select two large, strong prime numbers: p and q, such that: $q \mid (p-1)$ or: $p = N \times q + 1$, with N is an integer number.

2 - Select $g = \alpha^{(p-1)/q} \bmod p$, is a generator element order q of group Z_p^*, such that: $1 < g < p$ and: $g^q \equiv 1 \bmod p$, with: $\alpha \in Z_p^*$.

3 - Select a random private key x satifies : $1 < x < q$.

4 - Public key is computed as:

$y = g^x \bmod p$ (2.1)

5 - Public the values of (p, g, y). Keep secret: x.

2.2.2 The second key establishment procedure

Communicating parties agree to choose the parameters (p, q and g). They choose their own private key then compute the public key is formed under the procedure for system parameters and public key in section 2.2.1. Suppose that, A is a sender/encoder. The A's private key is xA, the corresponding public key is yA. B is a receiver/decoder. The B's private key is xB, the corresponding B's public key is yB. The values yA and yB need to be authenticated by a trust CA (Certificate Authority). A and B agree to use the same symmetric-key cryptography algorithm (eg DES, AES, ...) to encrypt information (text, document, ...) which will be exchanged with each other, The algorithm establishes a shared secret key K. In the algorithm allows, A to encode and B to decode information or vice versa. It is done by following steps below:

Step 1: Done by A.

1 - Choose a random values k such that:

$$0 < k < q-1$$

2 - Compute R as follow:

$$R = (y_A)^{k+1} \bmod p \quad (2.2)$$

3 - Compute *E* as equation:

$$E = R \bmod q \quad (2.3)$$

4 - Compute *S* as equation:

$$S = \left((k+1) + (x_A)^{-1} \times E\right) \bmod q \quad (2.4)$$

5 - Send (*E*,*S*) to B.

Step 2: Done by A and B.

1 - A generates the shared secret key K_{AB} according to formular (2.5):

$$K_{AB} = (y_B)^{x_A.k} \bmod p \quad (2.5)$$

2 - B generates the secret key K_{BA} by three small steps:

2.1 - Compute $\overline{R}$ as:

$$\overline{R} = g^{-E} \times (y_A)^S \bmod p \quad (2.6)$$

2.2 - Compute $\overline{E}$ as:

$$\overline{E} = \overline{R} \bmod q \quad (2.7)$$

2.3 - Check if $\overline{E} = E$ then establish the secret key KBA as:

$$K_{BA} = \left(\overline{R} \times (y_A)^{-1}\right)^{x_B} \bmod p \quad (2.8)$$

The shared secret key K of A and B is:

$$K = K_{AB} = K_{BA}.$$

if $\overline{E} \neq E$ then (E,S) has been changed and $K_{AB} \neq K_{BA}$. Therefore, in this case, the shared secret key K between A and B can not be established.

Discussion:

- Like the first algorithm was introduced in section 2.1, the second key established algorithm is the new algorithm which does not use any previously known public key cryptography protocols in practice.
- Instead of sending R to B as the first protocol, in the second one, A sends to B a pair values (E,S), which is generated from R and A's private key xA according to (2.3) and (2.4). From the pair values (E,S) B can recomputes R and A's public key yA according to (2.6). From the R (in Step 2 is denoted by $\overline{R}$), B will create a shared secret key with A. The key issue is that R is recovering from the pair values (E,S) through A's public key, according to (2.6). It shows that (E,S) must be generated from A's private key according to (2.3) and (2.4). That mean (E, S) and KBA must originate from A. This is the shared key authentication mechanism of both proposed algorithms, so they can resistance to spoofing attacks.
- In step 2, B just creates a secret key with A only if the equation $\overline{E} = E$ is satisfied. From (2.3), (2.4), (2.6) and (2.7) shows that this condition is satisfied only if (E,S) is transmitted from A to B without any change at all. A change in the value of E or S or both at the same time results in the equation: $\overline{E} = E$ does not satisfied. It means that, if the equation have been satisfied, pointing out the value of R is generated from A and will be restored exactly on B ($\overline{R} = R$) and therefore shared secret key KBA is generated. The secret key is generated on the A or B has the same value. In other words, the secret key is integrity transmitted from A to B. This is the integrity of authentication mechanism of the secret key in the second protocol.

2.2.3 The correctness of the second algorithm

What to prove is: Let p, q are two independence prime number, such that:

$$q \mid (p-1),\ g = \alpha^{(p-1)/q} \bmod p,\ \alpha \in Z_p^*,$$

$$1 < x_A, x_B < q,\ y_A = g^{x_A} \bmod p,$$
$$y_B = g^{x_B} \bmod p,\quad 0 < k < q-1,$$
$$R = (y_A)^{k+1} \bmod p,$$
$$E = R \bmod q,$$
$$S = \left((k+1) + (x_A)^{-1} \times E\right) \bmod q.$$

If:

$$K_{AB} = (y_B)^{x_A.k} \bmod p,$$
$$\overline{R} = g^{-E} \times (y_A)^S \bmod p \quad ,$$
$$K_{BA} = \left(\overline{R} \times (y_A)^{-1}\right)^{x_B} \bmod p,\ \overline{E} = \overline{R} \bmod q$$

then:

$$\overline{E} = E \text{ and } K_{AB} = K_{BA}.$$

Proof:

From equation (2.1) and (2.4) we have:

$$\begin{aligned}\overline{R} &= \left(g^{-E} \times (y_A)^S\right) \bmod p \\ &= \left(g^{-E} \times \left(g^{x_A} \bmod p\right)^S\right) \bmod p \\ &= g^{-E} \times g^{x_A.\left((k+1)+(x_A)^{-1}.E\right)} \bmod p \\ &= g^{-E} \times g^{x_A.(k+1)} \times g^E \bmod p \\ &= g^{(k+1).x_A} \bmod p \\ &= \left(g^{x_A} \bmod p\right)^{k+1} \bmod p \\ &= (y_A)^{k+1} \bmod p\end{aligned} \quad (2.9)$$

From (2.2) and (2.9) deduce:

$$\overline{R} = R \quad (2.10)$$

From equation (2.3), (2.7) and (2.10) can deduce the first what to proof:

$$\overline{E} = \overline{R} \bmod q = R \bmod q = E$$

From (2.1), (2.2) and (2.5) we have:

$$\begin{aligned}K_{AB} &= (y_B)^{x_A.k} \bmod p \\ &= \left(g^{x_B} \bmod p\right)^{x_A.k} \bmod p \\ &= g^{k.x_A.x_B} \bmod p\end{aligned} \quad (2.11)$$

On the other hand, from (2.1), (2.8) and (2.10) we also have:

$$\begin{aligned}K_{BA} &= \left(\overline{R} \times (y_A)^{-1}\right)^{x_B} \bmod p \\ &= \left(R \times (y_A)^{-1}\right)^{x_B} \bmod p \\ &= \left((y_A)^{k+1} \times (y_A)^{-1} \bmod p\right)^{x_B} \bmod p \\ &= (y_A)^{k.x_B} \bmod p = g^{k.x_A.x_B} \bmod p\end{aligned} \quad (2.12)$$

From (2.11) and (2.12) infer what we have to proof :

$$K_{AB} = K_{BA} = K$$

2.2.4 The security level of the second algorithm

The security of the second proposed algorithm can be accessed by two security attributes:

a) Resistance to reveal secret keys attacks

Similar analyzing as section 2.1.4 a), It can be seen that Both new proposed algoritms can resistance to reveal secret keys attacks and they depend on the difficulty of DLP (Discrete Logarithm Problem).

b) Resistance to spoofing attacks

As discussion in section 2.2.2, if the equation: $\overline{E} = E$ is not satisfied then B can confirm that the received values (E,S) are not sent from A or the values have been changed during transmission from A, so the shared secret key can not be established. As discussion in section 2.1.4 b) shows that even if the impersonate attacker can create a pair (E,S) satisfies the condition checked as indicated, it is not possible to set the shared secret key with B.

3. Conclusion

This paper proposes two new key established algorithms for symmetric key cryptosystems, the new algorithms have following characteristics:

- It needs only one time to transmit information to establish a shared secret key between two communication parties. It is similar to any key transport protocol which uses public key cryptography.
- Information transmitted from sender/encoder to receiver/decoder or vice versa, is not a shared secret key, but it helps to establish a shared secret key between two parties. The information plays the same role in the key exchange protocol. However, this information is transferred only one time, so the new proposed algorithms need only a single round to establish the shared secret key.
- Shared secret key is authenticated of the origin and the integrity (only the second protocol), so this algorithm is able to resist to known spoofing attacks in practice.

Proof of the correctness and assessment of the security level of the new proposed algorithms shows very positive potential application in practice.

References

[1] W. Diffie & M. Hellman, “New Directions in Cryptography”, IEEE Trans. On Info. Theory, IT-22(6):644-654, 1976.

[2] R. L. Rivest, A. Shamir, and L. M. Adleman, “A Method for Obtaining Digital Signatures and Public Key Cryptosystems”, Commun. of the ACM, Vol. 21, No. 2, 1978, pp. 120-126.

[3] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Transactions on Information Theory, 1985, Vol. IT-31, No. 4. pp.469–472.

[4] A. Menezes, P. van Oorschot and S. Vanstone, “Handbook of Applied Cryptography”, Boca Raton, Florida: CRC Press, 1997.

[5] D.R Stinson, “Cryptography: Theory and Practice”, CRC Press 1995.

[6] Wenbo Mao, “Modern Cryptography: Theory and Practice”, Prentice Hall PTR, 2003.

Viet H.V received the B.S in Math and informatic applied from VNU University of Science in 1999 and M.S. degrees in Network management from University of Sedney in 2006. I am interested in information security, network security

DUC T.M, I graduated information technology of Le Qui Don Technical University (Ha Noi, Viet Nam). I am working in Information Technology Faculty. My research related to cryptography, information security and image processing. Emai: tmduc08@gmail.com

Truyen B.T graduated information technology of Military Technical Academy. My research related to cryptography, information security and simulation tecjnology.

Dung L.H is a Lecturer at the Le Qui Don Technical University (Ha Noi, Viet Nam). My research interests include cryptography, communication and network security. He published more than 3 papers in Scientific Journals and Proceedings of Conferences in the areas of his research. He received the Electronics Engineer degree (1989) and Ph.D (2013) from the Le Qui Don Technical University.