Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

The Key Distribution Protocol for Moible Communication Systems with Untrusted Centers

Hwang, Shin-Jia*; Chang, Chin-Chen**; and Yang, Wei-Pang*
*Institute of Computer and Information Science
National Chiao Tung University, Hsinchu, Taiwan 300, R. O. C.
**Institute of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi, Taiwan 621, R.O.C.

Abstract

In this paper, we propose a verifiable server-aided computation protocol to help a user with small computation power compute the modulo P operation, where P is a large strong prime. Utilizing the protocol, we propose a new key distribution protocol used in mobile systems without trusted centers. In this new protocol, any two users can share the session key via the help of a untrusted center. The center has no chance to know the session key. Therefore, this key distribution protocol is more practical than those previous ones that require trusted centers.

Keywords: Key distribution protocol, server-aided computation, secret key cryptosystem, session key.

1. Introduction

To transmit secret messages in insecure networks, secure cryptosystems are necessary. There are two kinds of cryptosystems: secret key cryptosystems and public key cryptosystems. In a secret key cryptosystem, two users must share the same secret key to encrypt (decrypt) messages (ciphertexts). Since the secret key is shared by two users, key distribution protocols are necessary to transmit secret keys from one user to another. Many key distribution protocols have been proposed [1, 2].

A mobile communication system is a special network consisting of two kinds of members. One is the centers with huge computational power and the other is the user terminals with small computational power. Because of the limited computational power of the user terminals, existing key distribution protocols are not suitable for mobile communication systems [10].

Tatebayashi et al.[10] proposed two key distribution protocols, KDP1 and KDP2, for mobile communication systems. They also showed that KDP1 is vulnerable to replaying attacks. Hence, they proposed KDP2. In KDP2, any two user terminals share a secret session key with the aid of the center. At Eurocrypt '93, KDP2 was broken by Park et al. [6]. Park et.al. also proposed their protocol.

A common characteristic of the above protocols is that the center knows the session key and then is able to know the transmitted messages. No existing scheme can detect and defend against this attack. Requiring trusted centers is not practical.

In 1976, Diffie and Hellman [2] proposed a public key distribution protocol based on a discrete logarithm problem. Any two user can generate the session key by themselves. The communication system needs only one system manager to manage the public key of each user. In their protocol, the center is removed when any two users compute their session keys. For the large computational cost to generate session keys, this protocol is not suitable for a mobile communication system.

Inspired by Diffie and Hellman's protocol, we propose a new key distribution protocol for mobile communication systems with untrusted centers. In the next section, some previous research results will be reviewed. Then a new verifiable server-aided computation protocol will be proposed in Section 3. The new key distribution protocol for mobile communication systems with untrusted centers will be presented in Section 4. Section 5 concludes the paper.

2. Review

We review Diffie and Hellman's public key distribution protocol in the following. In their scheme, each user, say User A, chooses a secret random number S_A and publishes the value $Q_A = \alpha^{S_A}$ mod P, where α is a primitive number of GF(P) and P is a strong prime number.

Suppose that User A and User B want to construct a session key CK_{AB}. User A computes the session key $CK_{AB} = (Q_B)^{S_A}$ mod P. User B computes the session key $CK_{AB} = (Q_A)^{S_B}$ mod P. Then they have to generate the same session key $CK_{AB} = \alpha^{S_A S_B}$ mod P. The security of Diffie and Hellman's public key distribution protocol is based on the difficulty of Diffie and Hellman's problem. In this protocol, the center is removed. Since the computational cost of the session key is large, user terminals with limited computational power cannnot generate the session key in a reasonable amount of time. This public key distribution protocol is not suitable for mobile communication systems.

3. A verifiable server-aided computation protocol for modular P exponential operation

Now we propose a new verifiable server-aided computation protocol in which a powerful server can help a client to perform the modular P exponential operation, X^S mod P, where S is a secret value of the client, X is a public integer, and P is a public strong prime. The secret value S is not released in the protocol. The server does not need to be trusted. After obtaining the result of X^S mod P, the client could verify whether the server does use X to compute X^S mod P.

Let P be a prime number and X be an integer such that X and P are relatively prime. Then $X^{P-1} \equiv 1$ (mod P).

[VSACP-MP]

Step 1: A client randomly generates an integer vector $R = (r_1, r_2, \ldots, r_M)$ and two binary vectors $Z1 = (z1_1, z1_2, \ldots, z1_M)$ and $Z2 = (z2_1, z2_2, \ldots, z2_M)$ satisfying the following requirements:

(1) $S \equiv \sum_{i=1}^{M} (r_i)(z1_i) \pmod{P-1}$,

(2) $1 \equiv \sum_{i=1}^{M} (r_i)(z2_i) \pmod{P-1}$,

(3) $Z1 \neq Z2$,

(4) Z1 and Z2 are not two disjoint vectors, and

(5) Weight(Z1) $\leq$ L and Weight(Z2) $\leq$ L, where Weight(W) denotes the number of 1's of the binary vector W and M and L are two predetermined integers.

Step 2: The client transmits {R, X, P} to the server.

Step 3: The server calculates $R' = (r'_1, r'_2, \ldots, r'_M)$ as $r'_i = (X)^{r_i}$ mod P. Then the server transmits R' to the client.

Step 4: The client verifies whether $X \equiv \prod_{i=1}^{M} (r'_i)^{z2_i} \pmod{P}$. If $\prod_{i=1}^{M} (r'_i)^{z2_i}$ mod P is not equal to X, then the client stops, because the server has sent an incorrect vector R' to the client.

Step 5: The client computes $(X)^S$ mod P= $\prod_{i=1}^{M} (r'_i)^{z1_i}$ mod P.

The reason the client can verify the value of X and to obtain $(X)^S$ mod P is given below. Since $1 \equiv \sum_{i=1}^{M} (r_i)(z2_i) \pmod{P-1}$, by Fermat's Little Theorem [9, pp. 37-42], $\prod_{i=1}^{M} (r'_i)^{z2_i} \equiv \prod_{i=1}^{M} ((X)^{r_i})^{z2_i} \equiv X^{\sum_{i=1}^{M}(r_i)(z2_i)} \equiv X \pmod{P}$. Since $S \equiv \sum_{i=1}^{M} (r_i)(z1_i) \pmod{P-1}$, by Fermat's Little Theorem, $\prod_{i=1}^{M} (r'_i)^{z1_i} \equiv \prod_{i=1}^{M} ((X)^{r_i})^{z1_i} \equiv X^{\sum_{i=1}^{M}(r_i)(z1_i)} \equiv (X)^S \pmod{P}$.

The computational cost of the client is at most 2(L-1) multiplication mod P operations. The amount of communication between the client and the server is 2(M+1) integers which have length are at most log P bits. Hence, the number of multiplication operations of our protocol is double that of RSA-S1. The additional multiplication mod P operations are the cost of the verification mechanism. The amount of communication of our protocol is approximately the same as that of RSA-S1.

To analyze the security of the server-aided computation protocol, we shall consider the passive and active attacks proposed by Pfitzmann and Waidner [7].

Since the binary vector Z1 can be decomposed into two disjoint binary vectors $Z1'_1$ and $Z1'_2$, Pfitzmann and Waidner's passive attack can be modified to attack our protocol. The modified passive attack is stated below. Then we discuss the impact of the modified passive attack on VSACP-MP.

[Modified Passive Attack on VSACP-MP]

Suppose that the server obtains the result value $Y=(X)^S \bmod P$. The server wants to derive the secret value S from X, R, R', and Y.

Step 1: The server computes all the products $Y_{Z1'} = \prod_{i=1}^{M} (r'_i)^{z1'_i} \bmod P$ for all binary vectors Z1' such that Weight(Z1') ≤ $\lceil L/2 \rceil$.

Step 2: The server also computes the value $Y^*_{Z1'} = Y(Y_{Z1'})^{-1} \bmod P$ for the binary vectors Z1'.

Step 3: The server rearranges the triple (Z1', $Y_{Z1'}$, $Y^*_{Z1'}$) by sorting all $Y_{Z1'}$.

Step 4: The server finds all pairs ($Z1'_1$, $Z1'_2$) satisfying the following requirements:
(1) $Y_{Z1'_1} = Y^*_{Z1'_2}$, and
(2) $Z1'_1$ and $Z1'_2$ are disjoint.
From a pair ($Z1'_1$, $Z1'_2$), the server finds a binary vector $Z1'_1+Z1'_2$ that is a candidate for the real binary vector Z1.

Step 5: If there is more than one candidate vector, the server tests all candidates by using another integer X'.

Through Pfitzmann and Waidner's passive attack, the searching space of RSA-S1 is reduced from $\sum_{i=1}^{L} C(M, i)$ to $\sum_{i=1}^{\lceil L/2 \rceil} C(M, i)$. Hence the search space of VSCAP-MP is also reduced to $\sum_{i=1}^{\lceil L/2 \rceil} C(M, i)$ by using the modified passive attacks.

[Modified Active Attack on VSACP-MP]

Can the server modify Pfitzmann and Waidner's active attack to break VSCAP-MP? In modifying Pfitzmann and Waidner's active attack, the server replaces the ith element r'_i and the jth element r'_j of R' with $wr'_i \bmod P$ and $(w^{-1})r'_j \bmod P$, respectively, to produce a new vector $\mathbf{R'}=(r'_1, r'_2, ..., r'_M)$, where w is an integer and i and j are two different numbers. Instead of sending R', the intruder sends the new vector $\mathbf{R'}$ to the client. Assume that the server could get the result value $Y'=\prod_{i=1}^{M} (r'_i)^{z1_i} \bmod P$ and know the correct result value $Y=(X)^S \bmod P$. The server is able to determine $z1_i$ and $z1_j$ of Z1 by comparing Y and Y'. There are three cases for the comparison result. If Y'= Y, then the server knows that either both $z1_i$ and $z1_j$ are equal to 0, or both $z1_i$ and $z1_j$ are equal to 1. If $Y' \equiv (w^{-1})Y \pmod P$, then $z1_i = 0$ and $z1_j = 1$. If $Y' \equiv wY \pmod P$, then $z1_i = 1$ and $z1_j = 0$.

What is the probability that the modified active attack will go undetected? If the server does not have the binary vector Z2, the undetectable probability of his attack is 0.25. The reason is that the attack escapes detection by the verification mechanism $X \equiv \prod_{i=1}^{M} (r'_i)^{z2_i} \pmod P$ only when both $z2_i$ and $z2_j$ are equal to 0. In order to pass the verification mechanism $X \equiv \prod_{i=1}^{M} (r'_i)^{z2_i} \pmod P$, the server should find the vector Z2 by exhaustive search in advance. If the server is able to know the result value Y', he determines at most two bits of Z1. Note that the modified active attack does not work if the server does not have the computing result Y'.

If the server repeats the same attack many times, he may find more than two bits of Z1. Fortunately, in VSACP-MP, the client changes the binary vector Z1 and Z2 each time, so the intruder can determine at most only two bits of the same Z1. Therefore, the modified active attack cannot be used to break VSACP-MP.

If the server mixes the modified passive and active attacks to attack VSACP-MP, then the searching space may be reduced to $\sum_{i=1}^{\lceil (L-1)/2 \rceil} C(M-2, i)$, for the case where only one of $z1_i$ and $z1_j$ is equal to 1, or $\sum_{i=1}^{\lceil (L-2)/2 \rceil} C(M-2, i)$, for the case where both $z1_i$ and $z1_j$ are equal to 1, or $\sum_{i=1}^{\lceil L/2 \rceil} C(M-2, i)$, for the case where both $z1_i$ and $z1_j$ are equal to 0.

Can the server pass the verification mechanism $X \equiv \prod_{i=1}^{M} (r'_i)^{z2_i}$ (mod P) of the client and cause the client to compute an incorrect result $Y' = (X')^{S}$ mod P? This type of attack will succeed only when Z1 and Z2 are two disjoint vectors. Since Z1 and Z2 are not disjoint in VSACP-MP, there is no chance that this attack will succeed. Therefore, the server cannot cheat the client by replacing the actual integer X with a wrong integer X'.

If Z2 is known by an intruder, what information related to Z1 can the intruder learn from Z2? Since Z1 and Z2 are not disjoint, there is at least one common position i such that $z1_i = z2_i = 1$. Then the searching space will be reduced from $\sum_{i=1}^{L} C(M, i)$ to Weight(Z2)($\sum_{i=1}^{L-1} C(M-1, i)$). If the intruder adopts the mixed passive and active attack, the searching space may be at most reduced to Weight(Z2)($\sum_{i=1}^{\lceil (L-3)/2 \rceil} C(M-3, i)$). Since Weight(Z2) $\geq$ 1, Weight(Z2)($\sum_{i=1}^{\lceil (L-3)/2 \rceil} C(M-3, i)) \geq (\sum_{i=1}^{\lceil (L-3)/2 \rceil} C(M-3, i)$). The exhaustive search needed to find Z2 is very time consuming, so this new attack is not effective.

The way in which the values of M and L are chosen is very important in our protocol. According to the security analysis, M and L should be chosen such that $(\sum_{i=1}^{\lceil (L-3)/2 \rceil} C(M-3, i)) \geq P-1$.

4. A new key distribution protocol with untrusted centers

A new key distribution protocol with unteusted centers is proposed in the first subsection. The security analysis of the protocol is given in the second subsections.

4.1 The new protocol

In our protocol, the mobile communication system consists of three kinds of members: a trusted system manager, several untrusted centers and many user terminals. Both the system manager and the centers have huge computation power. The user terminals have only small computational power. The goal of the new key distribution protocol is that two user terminals A and B can construct a session key CK_{AB} with the help of an untrusted center. The center cannot know the session key CK_{AB}, because it is not trusted.

The new key distribution protocol is divided into four parts: preparation, registration, key distribution, and handshaking verification. Each of the four parts is described below.

[Preparation]

The system manager is responsible for constructing the communication system. The construction procedure is stated below.

<u>Step 1</u>:. The system manager constructs two RSA public key cryptosystems (e_S, d_S, n) and (e_C, d_C, n), where n is the product of two large prime numbers, e_S and e_C are public keys, d_S and d_C are the corresponding secret keys, and e_C should be small enough that each user terminal computes $(X)^{e_C}$ mod P at a reasonable speed.

<u>Step 2</u>:. The system manager publishes (e_C, n) for each user terminal, publishes (e_S, n) for each center, and gives (d_C, n) to each center through a secure channel.

<u>Step 3</u>:. The system manager also publishes a one-way function $G(\alpha,X)=\alpha^{X}$ mod P, a primitive number α of GF(P), and a large strong prime number P.

[Registration]

A new user A executes the following steps to enter the mobile communication system.

<u>Step 1</u>:. User A selects a random secret integer S_A.

<u>Step 2</u>:. The system manager helps User A to compute $P_A = (\alpha)^{S_A}$ mod P by VSACP-MP.

<u>Step 3</u>:. User A sends P_A and ID_A to the system manager through a secure channel.

<u>Step 4</u>:. The system manager computes authentication pattern $V_A = (P_A \| ID_A)^{d_S}$ mod n and transmits V_A

to User A through a secure channel, where $||$ denotes the concatenation operation.

Now, with the help of a center, the user A can share a session key with another legal user in the communication system.

[Key distribution]

Suppose that User A wants to communicate with User B. User A needs the aid of a center to construct a session key CK_{AB} shared by Users A and B.

[On the terminal of User A]

Step A1: Randomly construct an integer vector $R_A = (r_{A,1}, r_{A,2}, \ldots, r_{A,M})$, and two binary vectors $Z1_A = (z1_{A,1}, z1_{A,2}, \ldots, z1_{A,M})$ and $Z2_A = (z2_{A,1}, z2_{A,2}, \ldots, z2_{A,M})$ satisfying the following requirements:

(1) $S_A \equiv \sum_{i=1}^{M} (r_{A,i})(z1_{A,i}) \pmod{P-1}$,

(2) $1 \equiv \sum_{i=1}^{M} (r_{A,i})(z2_{A,i}) \pmod{P-1}$,

(3) $Z1_A \neq Z2_A$,

(4) $Z1_A$ and $Z2_A$ are not disjoint, and

(5) $\text{Weight}(Z1_A) \leq L$ and $\text{Weight}(Z2_A) < L$.

Step A2: Generate a random number r_1 and select a center C to help with the computation of the session key CK_{AB}.

Step A3: Compute $C_A = (T_1||V_A||r_1)^{e_C} \bmod n$, where timestamp T_1 is the sending time.

Step A4: Send the package $\{R_A, C_A, (P_A||ID_A), ID_B\}$ to the selected center C and wait for the response of the center C.

[On the host of the center C]

Suppose that the center C receives the package sent from User A at T'_1. Then the center C performs the following steps:

Step C1: Decrypt C_A to obtain the sending timestamp of the package T_1 and the authentication pattern V_A.

Step C2: Verify whether $\Delta T_A \leq T'_1 - T_1$, where ΔT_A is the legal transmitting delay between the center C and User A. If the delay $T'_1 - T_1$ is larger than ΔT_A, then the center C rejects the request of the sender.

Step C3: Authenticate the identity of User A by checking whether $(P_A||ID_A) \equiv (V_A)^{e_S} \pmod{n}$. If the equation is not equal, the center C also rejects User A and refuses to construct the session key.

Step C4: Inform User B with whom User A wants to communicate. Wait for the response of User B.

[On the terminal of User B]

Step B1: Randomly construct an integer vector $R_B = (r_{B,1}, r_{B,2}, \ldots, r_{B,M})$, and two binary vectors $Z1_B = (z1_{B,1}, z1_{B,2}, \ldots, z1_{B,M})$ and $Z2_B = (z2_{B,1}, z2_{B,2}, \ldots, z2_{B,M})$ satisfying the following requirements:

(1) $S_B \equiv \sum_{i=1}^{M} (r_{B,i})(z1_{B,i}) \pmod{P-1}$,

(2) $1 \equiv \sum_{i=1}^{M} (r_{B,i})(z2_{B,i}) \pmod{P-1} = 1$,

(3) $Z1_B \neq Z2_B$,

(4) $Z1_B$ and $Z2_B$ are not disjoint, and

(5) $\text{Weight}(Z1_B) \leq L$ and $\text{Weight}(Z2_B) \leq L$.

Step B2: Generate a random number r_2.

Step B3: Compute $C_B = (T_2||V_B||r_2)^{e_C} \bmod n$, where timestamp T_2 is the sending time.

Step B4: Send the package $\{R_B, C_B, (P_B||ID_B), ID_A\}$ to the center C and wait for the data sent from the center C.

[On the host of the center C]

Suppose that the center C receives the package sent from User B at T'_2. After receiving the package, the center executes the following steps:

Step C5: Decrypt C_B to get the sending timestamp of the package T_2 and the authentication pattern V_B.

Step C6: Verify whether $\Delta T_B \leq T'_2 - T_2$, where ΔT_B is the legal transmitting delay between the center C and User B. If the delay $T'_2 - T_2$ is larger than ΔT_B, then the center C informs User B to send another package again.

Step C7: Authenticate the identity of User B by checking whether $(P_B||ID_B) \equiv (V_B)^{e_S} \pmod{n}$. If the equation is not equal, the center C infers that the sender is illegal and stops.

Step C8: Compute $R'_A = (r'_{A,1}, r'_{A,2}, ..., r'_{A,M})$ for User A by $r'_{A,i} = (P_B)^{r_{A,i}} \bmod P$, for i=1, 2, ..., M. Compute $R'_B = (r'_{B,1}, r'_{B,2}, ..., r'_{B,M})$ by $r'_{B,i} = (P_A)^{r_{B,i}} \bmod P$, for i=1, 2, ..., M.

Step C9: Transmit R'_A and R'_B to User A and User B, respectively.

[On the terminal of User A]

Step A5: Verify whether $P_B \equiv \prod_{i=1}^{M} (r'_{A,i})^{z2_{A,i}} \pmod P$. If the equation does not hold, then the center is an intruder.

Step A6: Compute the session key $CK_{AB} \equiv \prod_{i=1}^{M} (r'_{A,i})^{z1_{A,i}} \pmod P$.

[On the terminal of User B]

Step B5: Verify whether $P_A \equiv \prod_{i=1}^{M} (r'_{B,i})^{z2_{B,i}} \pmod P$. If the equation does not hold, then the center is an intruder.

Step B6: Compute the session key $CK_{BA} \equiv \prod_{i=1}^{M} (r'_{B,i})^{z1_{B,i}} \pmod P$.

After the key distribution, User A and User B hold the session keys CK_{AB} and CK_{BA}, respectively. Before they start to communicate, User A and User B need to know whether CK_{AB} is equal to CK_{BA}. In the handshaking verification, the two users only verify whether $CK_{AB}$is equal CK_{BA}. Neither knows the session key held by the other.

[Handshaking verification]

After the protocol, User A and User B confirm the correctness of the session key with one another.

[On the terminal of User A]

Step A1: Choose a random number vr_A and obtain a ciphertext CVA1 by encrypting vr_A with the session key CK_{AB}.

Step A2: Sent CVA1 to User B and wait for CVB1 sent from User B. If User B does not send CVB1, then User B may be an intruder.

Step A3: To recover vr_B by decrypting CVB1, encrypt vr_B+1 to produce CVA2 by the key CK_{AB}. Send CVA2 to User B and wait for CVB2.

Step A4: To use the session key CK_{AB}, recover vr'_A by decrypting CVB2. If vr'_A is not equal to vr_A+1, then the session key held by User A is wrong.

[On the terminal of User B]

Step B1: Choose a random number vr_B and get a ciphertext CVB1 by encrypting vr_B with the session key CK_{BA}.

Step B2: Sent CVB1 to User A and wait for CVB1 sent from User A. If User B does not send CVA1, then User A may be an intruder.

Step B3: To recover vr_A by decrypting CVA1, encrypt vr_A+1 to produce CVB2 by the key CK_{BA}. Send CVB2 to User A and wait for CVA2.

Step B4: To use the session key CK_{BA}, recover vr'_B by decrypting CVA2. If vr'_B is not equal to vr_B+1, then the session key is wrong.

After the handshaking verification, the users A and B know whether they hold the same session key $CK_{AB} = CK_{BA}$.

4.2 The Security Analysis and Discussion

To analyze the security of the new key distribution protocol, we consider the security of user authentication, replaying attacks, and the impact of passive and active attacks. In addition, we propose a new attack and show that this attack also fails.

The security of the user authentication is based on the security of the signature of the RSA public key cryptosystem [8]. Since the authentication pattern $V_A = (P_A \| ID_A)^{d_s} \bmod n$ of User A is a signature of $(P_A \| ID_A)$ by (e_S, d_S, n), and the RSA public key cryptosystem is secure, the signature $V_A = (P_A \| ID_A)^{d_s} \bmod n$ can not be forged. Therefore, the authentication pattern V_A can represent the identity of User A.

Next, we must examine whether the new key distribution protocol defends against replaying attacks. Since the actions of User A and User B in the new key distribution protocol are symmetric, we consider only the actions of User A. According to the key distribution protocol, there are a timestamp T_1, a random integer vector R_A, and two binary vectors $Z1_A$ and $Z2_A$ generated for the communication between the center C and User A. Since the timestamp T_1 is hidden in the package $\{R_A, C_A, (P_A \| ID_A), ID_B\}$, the package cannot be sent again by an intruder. Otherwise, the transmitting delay would be longer than the legal delay ΔT_A. Could an intruder replace only the R'_A with a previous intercepted vector $\mathbf{R}'$? Because the two binary vectors $Z1_A$ and $Z2_A$ are changed each time, User A then computes an incorrect session key CK_{AB} by the intercepted vector $\mathbf{R}'$. After the handshaking verification, User A finds that the session key was wrong. Replaying attacks would not work in the new protocol.

We now consider the modified passive and active attacks. Since the new key distribution protocol uses VSACP-MP to handle the cooperation between User A and the center C (or User B and the center C), the server cannot use the modified passive attacks to derive the secret value S_A of User A. In the modified active attack, the server needs to know the computing session key CK_{AB}. In the new key distribution protocol, no one knows the computing session key CK_{AB} except the user himself. Hence the modified active attack does not work for the new key distribution protocol. A mixed attack that adopts the modified active attack cannot work to break the new protocol. Since the modified active attack is useless, the search space of the secret value S_A of User A is at most reduced to $(\sum_{i=1}^{\lceil (L-1)/2 \rceil} C(M-1, i))$.

We now consider a new type of attacks. An untrusted center stands at the middle of the communication link between users A and B. The center generates an integer $P_C = (\alpha)^{S_C} \bmod P$. The center misleads User A into computing an incorrect session key $CK_{AC} = (P_C)^{S_A} \bmod P$. At the same time, User B computes an incorrect session key $CK_{CB} = (P_C)^{S_B} \bmod P$. After intercepting the ciphertext sent from User A, the center decrypts the ciphertext to plaintext by CK_{AC}, encrypts the plaintext to produce a new ciphertext by CK_{CB}, and sends the new ciphertext to User B. Then neither User A nor User B discovers the attack. Instead of finding the session key CK_{AB}, the center can use the key pair (CK_{AC}, CK_{CB}) to decipher the message communicated between Users A and B.

Fortunately, the untrusted center cannot mislead User A into computing $CK_{AC} = (P_C)^{S_A} \bmod P$ and User B into computing $CK_{CB} = (P_C)^{S_B} \bmod P$, because the center cannot pass both verification mechanisms $P_B \equiv \prod_{i=1}^{M} (r'_{A,i})^{z2_{A,i}} \pmod P$ and $P_A \equiv \prod_{i=1}^{M} (r'_{B,i})^{z2_{B,i}} \pmod P$. Our protocol is secure against this new attack. Therefore, our protocol is secure. Two integers M and L must be selected to satisfy $(\sum_{i=1}^{\lceil (L-1)/2 \rceil} C(M-1, i)) \geq P-1$.

Finally, we discuss the problem of whether the RSA public key cryptosystem (e_C, d_C, n) is secure when the public key e_C is small enough that each user terminal can compute $(X)^{e_C} \bmod P$ at a reasonable speed? According to [4, 11], it is insecure when there are many RSA cryptosystems whose public keys are all small and their products of two large primes are relatively prime. In this case, if the same message is sent to many users, then the message can be recovered by the Chinese Remainder Theorem. In our protocol, however, there is only one RSA public key cryptosystem with a small public key, so this case does not occur.

5. Conclusions

A new verifiable server-aided computation protocol for modular P exponentiation, VSACP-MP, has been proposed. In this protocol, a client with small computation power computes a modular P exponential operation, $(X)^S \bmod P$ with the help of a powerful server, where S is the secret value of the client, P is a public strong prime, and X is an integer given by the client. S is not released to the server. The server may be untrusted.

The modified passive and the modified active attacks cannot break VSACP-MP, nor attacks combining the modified passive and active attacks. The server cannot mislead the client into computing a wrong result $(X')^S \bmod P$ by replacing the real integer X with a different integer X', because a verification mechanism is used to check whether the server does use X to help the client compute $X^S \bmod P$. Therefore, VSACP-MP is secure.

On the basis of VSACP-MP, a key distribution protocol for mobile communication system with untrusted centers has been proposed. Since finding a trusted center is difficult, our protocol proposes a new approach. In the new protocol, the centers do not need to be trusted since the center cannot discover the session key of any two users.

Due to the security analysis in Section 4, neither replaying attacks, modified passive and active attacks, nor mixed attacks can break the new key distribution protocol. The security of the user authentication of our protocol is based on the security of the signature of the RSA cryptosystem. Hence our key distribution protocol for mobile communication systems with untrusted centers is secure.

References

[1] Denning, D. E. R. (1982): Cryptography and Data Security, Addison-Wesley, Reading, MA, 1982.

[2] Diffie, W. and Hellman, M. E. (1976): "New Directions in Cryptography," IEEE Transactions on Information Theory, Vol. IT-22, 1976, pp. 644- 654.

[3] ElGamal, T. (1985): "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory, Vol. IT-31, No. 4, July 1985, pp. 469- 472.

[4] Hastad, Johan (1988): "Solving Simultaneous Modular Equations of Low Degree," SIAM J. Comput., Vol. 17, No. 2, April 1988, pp.336-314.

[5] Matsumoto, T., Kato, K. and Imai, H. (1988): "Speeding up Secret Computations with Insecure Auxiliary Devices," Advances in Cryptology-CRYPTO '88, Springer Verlag, New York, 1989, pp. 497- 506.

[6] Park, C., Kurosawa, K., Okamoto, T. and Tsujii, S. (1993): "On Key Distribution and Authentication in Mobile Radio Networks," Proceedings of Eurocrypt ' 93, Hotel Ullensvang, Lofthus, Norway, May 24- 27, 1993, pp. T131-T138.

[7] Pfitzmann, B. and Waidner, M. (1992): "Attacks on Protocols for Server-Aided RSA Computation," EUROCRYPT '92, Balatonfured, 1992, pp. 139 -146.

[8] Rivest, R. L., Shamir, A. and Adleman, L. (1978): "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of ACM, Vol. 21, No. 2, 1978, pp. 120-126.

[9] Seberry, J. and Pieprzyk, J. (1988): Cryptography: An Introduction to Computer Security, Prentice Hall, Singapore, 1988.

[10] Tatebayashi, Makoto, Matsuzaki, Natsume, and Newman, Jr. David B. (1989): "Key Distribution Protocol for Digital Mobile Communication Systems," Proceedings of Crypto '89, Springer-Verlag, New York, 1989, pp. 324-334.

[11] Wiener, M. J. (1990): "Cryptoanalysis of Short RSA Secret Exponents," IEEE Transactions on Information Theory, Vol IT-36, May 1990, pp. 553-558.