1998 International Computer Symposium
Workshop on Cryptology and Information Security
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

A Software Key Escrow System Suitable for Broadcasting

Chien-Yuan Chen, Wei-Bin Lee**, Chin-Chen Chang***

*Department of Information Engineering
I-Shou University, Kaohsiung County, Taiwan, 84008 R.O.C.
email: cychen@csa500.isu.edu.tw,
**Institute of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi, Taiwan 621, R.O.C.
email: ccc@cs.ccu.edu.tw

Abstract

The Escrowed Encryption Standard (EES) allows the Law Enforcement Agent to trace the sender by decrypting the Law Enforcement Access Field (LEAF). However, to prevent users from deleting the LEAF, the Clipper chip must be tamperproof. In this paper, we present a software key escrow system without a tamperproof hardware. In this system, the Law Enforcement Agent is allowed to wiretap the special designated communication once. Furthermore, to be more suitable for broadcasting, the LEAF is modified to make the tracing of the sender and the recipient possible.

1. Introduction

Privacy is the right of people by nature. However, the government should have a right to trace the communications when the crimes happen. For example, in a kidnapping case, to trace the kidnappers is required, while in a drug case, to trace the recipient is important. But, if the communication used is broadcast oriented, such as in a cellular telephone system, it may be hard to trace the sender and the recipient.

To guarantee the privacy of law-abiding citizens and traceability of the suspected, the Clinton Administration announced the Escrowed Encryption Standard (EES) in 1993 [2, 4]. In EES, the Law Enforcement Access Field (LEAF) is attached to the ciphertext to allow the Law Enforcement Agent (LEA) to trace the sender. To prevent the users from deleting the LEAF, the chip decryption procedure will not decrypt the cipher if the LEAF is not validated. Obviously, such a solution requires a tamperproof hardware.

Because the requirement of the tamperproof hardware seems unreasonable to researchers, many seek another solution. Recently, Desmedt [3] has presented a software key escrow system such that traceability of the recipient is possible. Unfortunately, it cannot be suitable for broadcasting because tracing the sender is impossible. This motivates us to design a key escrow system without a tamperproof hardware suitable for broadcasting. On the other hand, in [3, 4], once the LEA knows the parameters from the Key Escrow Agents, it can decrypt everything sent from the sender or transmitted to the recipient, without any time limit. This drawback is eliminated because our system allows the LEA only to obtain the access to special designated communications.

2. Our System

Before describing the system, two important concepts should be pointed out. First, to guarantee that the ciphertext cannot be decrypted without the LEAF, the function of the session key exchange is involved in the LEAF. Second, to trace both the sender and the recipient, the LEAF must contain the identities of the sender and the recipient. However, the identities are apt to cause the forgery by the intruder. So, we require some technique to verify the information.

The system is partitioned into four phases: registration, encryption, decryption, and legal wiretapping.

2.1 Registration

Registration phase is further divided into two parts: registering public keys with the Law Enforcement Agent (LEA) and registering secret keys with the Key Escrow Agents (KEAs), which are shown in Fig. 1.

Part 1. Registering Public Keys with the LEA

The LEA selects m distinct large primes q_1, q_2, ..., q_m, and computes $Q = \prod_{i=1}^{m} q_i$. The LEA generates a large prime P such that P -1 is divided by Q. The prime P is published and the factors of Q are kept secret. Let g be an integer with the order Q module P. Each user U_i with his identity ID_i registers a public key with the LEA. According to ID_i, the LEA generates a unique user's identifier $B_i = (b_1, b_2, ..., b_m)$ in binary form, $B_i \neq 0$, and stores (B_i, ID_i) in its data base. It then gives g_i to the user U_i, where

$$g_i = g^{\frac{Q}{\prod_{i=1}^{m} q_i^{b_i}}} \bmod P.$$

The order of g_i modulo P is $\prod_{i=1}^{m} q_i^{b_i}$. To uniquely identify the user, the LEA must guarantee that no two of all g_i's have the same order. The user U_i randomly chooses x_i, coprime to P - 1, computes $Y_i = g_i^{x_i} \bmod P$, and sends it to the LEA. If Y_i and g_i have the same order module P, the LEA publishes a pair (g_i, Y_i) as the public key of the user U_i.

Part 2. Registering Secret Keys with the KEAs

Assume there are two key escrow agents, say KEA1 and KEA2. After registering a public key with the LEA, each user U_i must divide his secret key x_i such that $x_i = x_{i1} + x_{i2} \bmod (P - 1)$. The user U_i will send x_{i1} and x_{i2} to the KEA1 and KEA2 in secret channels, respectively. Once receiving the secret components, two agents compute $Z_1 = g_i^{x_{i1}} \bmod P$ and $Z_2 = g_i^{x_{i2}} \bmod P$, and verify whether $Z = Z_1 \cdot Z_2 \bmod P$ is equivalent to Y_i, where the public key (g_i, Y_i) comes from the public file. If x_{i1} and x_{i2} are validated, they are stored in the data bases of the KEA1 and KEA2, respectively.

2.2 Encryption

Assume the user U_i wants to send the message m to the user U_j. He first gets the public key (g_j, Y_j) from the public file. Then, he randomly chooses an integer k and generates a session key h(k), where $h(\cdot)$ is a one-way public hashing function. He uses the ElGamal scheme to encrypt k into a pair (R, S) such that

$$R = g_j^r \bmod P,$$

and $$S = (k \| ID_i) Y_j^r \bmod P,$$

where r is a random number coprime to P - 1, and "||" means the concatenation operation. Further, he uses the ElGamal signature scheme to sign the integer k as a pair (R', S') such that

$$R' = g_i^{r'} \bmod P,$$

and $$S' = (x_i R' - k)(r')^{-1} \bmod (P - 1),$$

where r' is a random number coprime to P - 1. Finally, the user U_i sends the ciphertext $C = E_{h(k)}(m)$ together with the LEAF = (R, S, R', S') to the user U_j. Here $E_{h(k)}(m)$ denotes that the symmetric algorithm, i.e., DES or Skipjack algorithm, uses h(k) as the key to encrypt the message m.

2.3 Decryption

When the user U_j gets C and LEAF, he uses his secret key x_j to compute $Z = R^{x_j} \bmod P$ and recover $(k \| ID_i) = S \cdot Z^{-1} \bmod P$. Then he validates the integer k by verifying the following equation

$$g_i^k = Y_i^{R'} \cdot (R')^{S'} \bmod P.$$

If the integer k is validated, he computes the session key h(k) by using the one-way public hashing function h. Finally, the message m will be discovered by $m = D_{h(k)}(C)$. Here $D_{h(k)}(C)$ denotes that the symmetric algorithm uses the key h(k) to decrypt the ciphertext C.

2.4 Legal Wiretapping

If the LEA wants to wiretap the communication between two users U_i and U_j. He first gets R, S, R' and S' from the LEAF. The order of R modulo P is easily computed by setting $b_k = 0$ if $R^{\frac{Q}{q_k}} = 1 \bmod P$; otherwise, setting $b_k = 1$. Once $B_j = (b_1, b_2, ..., b_m)$ is computed, the LEA can find ID_j from the data base. Similarly, the LEA can find ID_i according to the order of R'. Knowing the identities of both, the LEA sends the count order, ID_j and R to the KEA1 and KEA2. The KEA1 finds the secret component x_{j1} according to ID_j and computes $Z_1 = R^{x_{j1}}$

mod P. Similarly, the KEA2 also computes $Z_2 = R^{xj2}$ mod P. Then both Z_1 and Z_2 are sent to the LEA. After receiving Z_1 and Z_2, the LEA computes $Z = Z_1 \cdot Z_2$ mod P and then gets $(k\|ID_i) = S \cdot Z^{-1}$ mod P. The session key is then generated by computing h(k). Finally, the message m is discovered by computing $m = D_{h(k)}(C)$.

To understand easier, we show the above phases in Fig. 2.

2.5 Security Analysis

Under the assumption that the LEA and the KEA are trusted, we consider the security from three viewpoints of the sender and the recipient, the eavesdropper, and the LEA.

The sender and the recipient

In general, when referring to the key escrow system, one has to assume that there is no subliminal channel in communications [3]. That is to say, the sender and the recipient cannot share a common key in advance. In [3], the question whether the sender can find a scheme to hide the identity of the recipient was addressed. This problem is similar to the subliminal problem in [5]. However, the solution of the subliminal problem has a requirement that the sender and the recipient share a secret key. But, this requirement violates the aforementioned assumption. Therefore, the sender cannot hide the identity of the recipient.

The Eavesdropper

We consider the case that the eavesdropper wants to trace the sender or the recipient. To trace the recipient, the eavesdropper must obtain the order of g_j. Since finding the order of g_j is believed as hard as factoring P - 1 [1], it seems that the eavesdropper cannot trace the sender and the recipient.

The LEA

We consider the case that the LEA wants to decrypt the ciphertext without any help of the Key Escrow Agents. Since the LEA can obtain the order of g_j, say d, he can compute

$$S^d = ((k\|ID_i)Y_j^r)^d = (k\|ID_i)^d \bmod P.$$

This equation has d solutions because $d \mid (P - 1)$. However, d contains at least one large prime. So, the LEA cannot recover k from these solutions by an efficient method.

3. Conclusions

In this paper, we have presented a software key escrow system suitable for broadcasting. The LEAF allows the LEA to trace the sender and the recipient. Because the recipient must get the session key from the LEAF, the LEAF cannot be deleted by the users before the ciphertext is decrypted. Therefore, requiring a tamperproof hardware is eliminated. Besides, it should be worth noting that our system can overcome the drawback that the LEA is able to decrypt everything transmitted by the sender or received by the recipient, without any time restriction if the LEA knows the parameters from the KEAs.

4. Reference

1. L. M. Adleman, K. S. McCurley, "Open Problems in Number Theoretic Complexity. In Discrete Algorithm and Complexity," Proceedings of the Japan-US Joint Seminar, June 4-6, Kyoto, Japan, 1986, Johnson, D., Nishizeki, T., and Wilf, H., Academic Press Inc., Orlando, Florida, pp. 263-286, 1986.

2. T. Beth, H. Knobloch, M. Otten, G. J. Simmons, and P. Wichmann, "Towards Acceptable Key Escrow System," Proceedings of the 2nd ACM Conference on Computer and Communications Security, pp. 51-58, 1994.

3. Y. Desmedt, "Securing Traceability of Ciphertexts--Towards a Secure Software Key Escrow System," Pre-proceedings of EUROCRYPTO'95, pp. 147-157, 1995.

4. National Institute for Standards and Technology, "Escrowed Encryption Standard," Federal Information Processing Standards Publication 185, U.S. Dept. of Commerce, 1994

5. G. J. Simmons, "The Prisoners' Problem and the Subliminal Channel," Advances in Cryptology: Proceedings of Crypto'83, Plenum, New York, pp. 51-67, 1983.

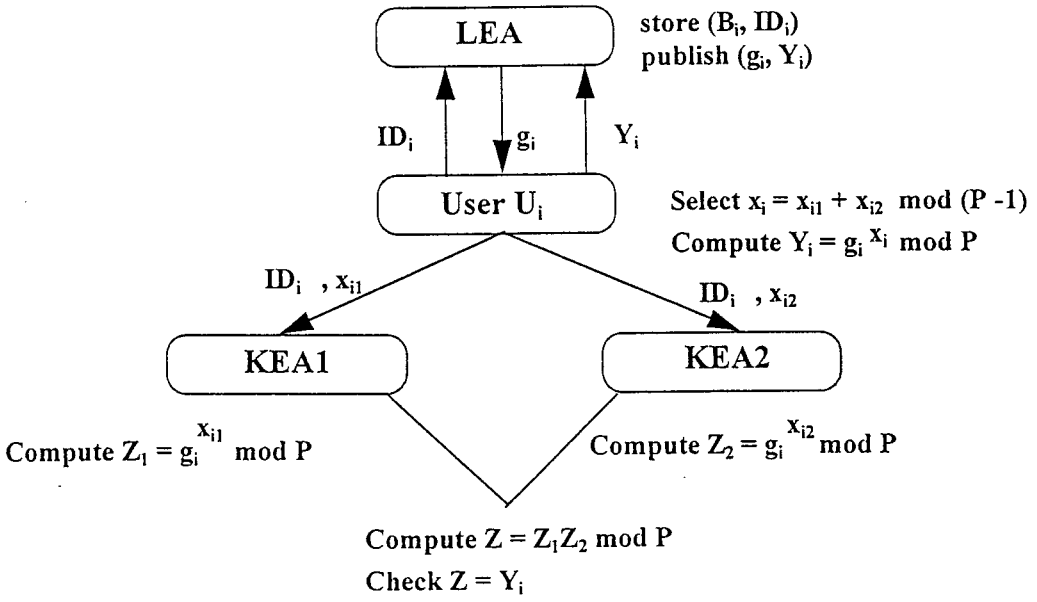

Fig. 1 Registration Phase

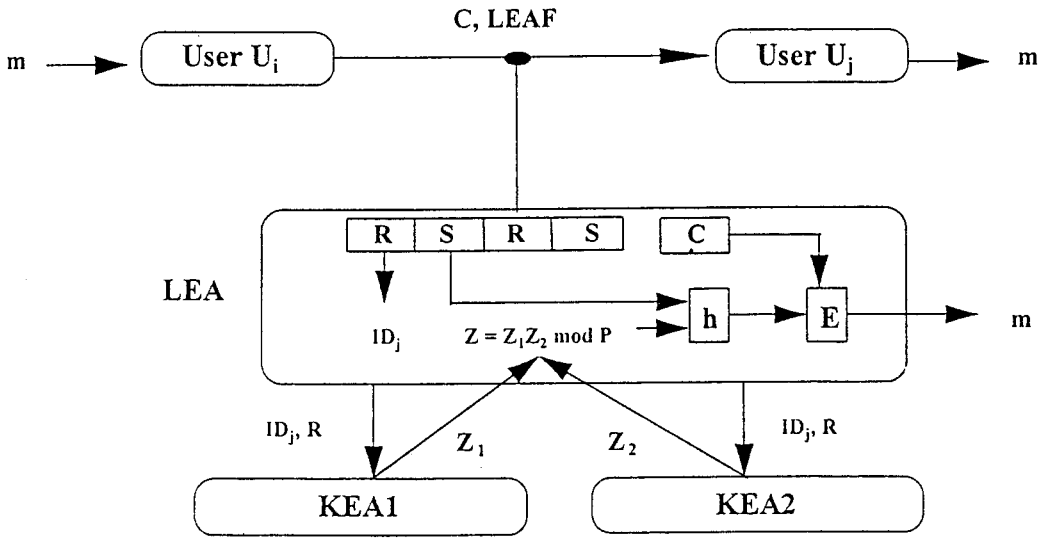

Fig. 2. Legal Wiretapping Phase