ELSEVIER

Computer Communications 21 (1998) 279–283

computer communications

Secure broadcasting in large networks

Hung-Min Sun[a], Shiuh-Pyng Shieh[b,*]

[a]*Department of Information Management, Chaoyang University of Technology, Wufeng, Taichung County, Taiwan 413, Republic of China*
[b]*Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China*

Received 6 January 1997; revised 22 October 1997; accepted 22 October 1997

Abstract

It is common that users or hosts in a large network are partitioned and organized as a hierarchical tree where children of the same parent form a group. Secure broadcasting intends to provide a secure communication channel from a sending principal to a group of legal receiving principals. Only legal receiving principals can decrypt the message, and illegal receiving principals cannot acquire any information from the broadcasted message. In this paper, we propose a secure broadcasting protocol in which only one packet is transmitted for every broadcast, and the size of the broadcasted packet is small. © 1998 Elsevier Science B.V.

Keywords: Secure systems; Broadcasting; Network security; Cryptography; Computer networks

1. Introduction

With the rapid development of computer networks, network security becomes an important issue. Generally, there are two types of communication in networks. One is the point-to-point type of communication which only concerns about two communicating principals. (A principal is an entity in a computer network which can send or receive messages. For example processes, users and stations can be principals.) Messages of this type can be protected either by private key cryptosystems or by public key cryptosystems [1–5]. The other one is the point-to-multipoint communication, also called broadcasting, in which a single transmission from a sending principle may be received simultaneously by many receiving principals [6]. Broadcasting technique is very useful in many applications, such as local area networks, satellite channels, and packet radio networks, when a message must be sent to several principals at the same time. As an example consider the sending principal and the legal receiving principals form a conference group. Secure broadcasting protocols can be regarded as conference-key distribution protocols in which the shared conference session key is broadcasted by the key distribution center to the conference members [7,8].

Secure broadcasting protocols intend to provide a secure communication channel. In the protocol, the sending principal broadcasts an encrypted message in a computer network, and only legal receiving principals can decrypt the message. Illegal receiving principals cannot acquire any information from the broadcasted message. In contrast, the sending principal using point-to-point approach must encrypt the message and transmit the ciphertext to the receiving principals, individually. Clearly, it is very inefficient because multiple copies of the ciphertext need to be transmitted [6].

Chiou and Chen proposed two secure broadcasting protocols, one is based on the public-key cryptosystem (PUBP in short), the other is based on the private-key cryptosystem (PRBP in short) [9]. These protocols are summarized as follows. The sending principal broadcasts the ciphertext (C_1, C_2) in the computer network, where C_1 is used by the legal receiving principals to obtain the communication key CK, and C_2 is the ciphertext of message M, encrypted by CK. When a legal receiving principal receive the ciphertext (C_1, C_2), he first uses his secret key to recover CK from C_1, and then uses CK to decrypt C_2 and acquire message M. Although these broadcasting cryptosystems are able to provide secrecy, the size of the packet being broadcasted are much larger than the original message. Assume that the number of legal receiving principals is n. Then the length of C_1 is n times longer than that of CK in their protocols. Comparing with the point-to-point approach, these protocols will not lead to a better result. This is the case because the length of C_1 is even longer than the total length of CKs being transmitted in the point-to-point approach (the length of C_1 will be n times of the length of CK in this approach).

Other broadcasting protocols employ a central authority server (CAS) in a computer network [10,11]. In these

* Corresponding author.

PII S0140-3664(97)00187-4

protocols, the ciphertext C_1 is saved in a public table which is only used once. The ciphertext (C_2) broadcasted in these protocols is smaller than the ciphertext (C_1, C_2) used in other protocols. However, the nonce public table needs to be broadcasted for every transmission in their protocols, and its size is not smaller than C_1. Therefore, these protocols have the same problem.

In this paper, we propose a secure broadcasting protocol which is capable of reducing the size of network message C_1 in a hierarchical tree of principals. In our protocol, only one packet is transmitted for every broadcast, and the total size being broadcasted is smaller than that in other protocols. This paper is organized as follows. In Section 2, we introduce the hierarchical tree of principals and discuss key management in the environment. In Section 3, we propose and analyze our secure broadcasting protocol. In Section 4, we give a comparison of our protocol with other protocols. Finally, we conclude this paper in Section 5.

2. Key management in a hierarchical tree of principals

It is common in many applications that members of a working group are organized as a hierarchical tree. For example a university has the tree structure containing colleges, departments, laboratories, and students (teachers), from root to leaves. The hierarchical structure can be shown in Fig. 1, where each node represents a group of principals, and each group may contain one or more principals. A lower level group is contained in its ancestral groups. We formulate the concept of hierarchical groups of principals by using graphic interpretation as follows.

We use tree structure to represent the structure of groups of principals. Each node in a tree corresponds to a group. A group may contain one or many principals. Every principal in a network system is regarded as a group and is represented by a leaf in a tree. Groups (children) sharing some characteristics form a sup-group (parent group), represented by a subtree. That is, the union of groups in a subtree forms their parent group. The root of the tree represents the universal group which is the group of all principals in the network system. The following example helps us to demonstrate the tree structure of the principal groups.

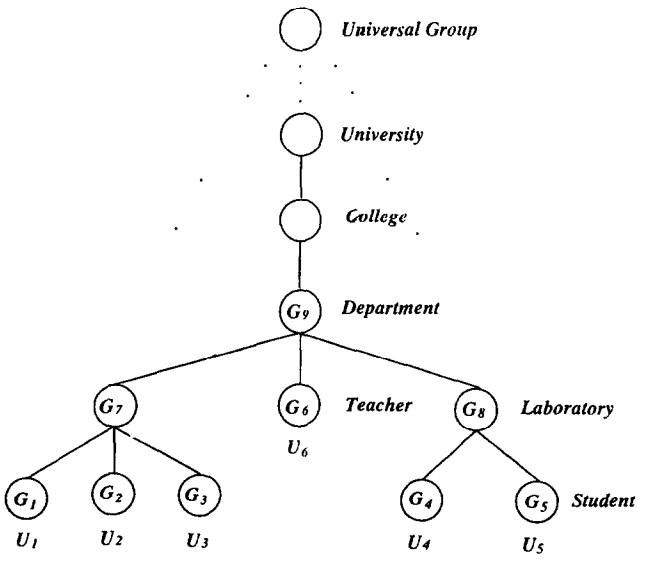

Fig. 1. The tree structure of user groups.

2.1. Example 1

Assume that a department consists of six members $u_1,\ldots,u_6$. Let $G_1 = \{u_1\},\ldots,G_6 = \{u_6\}$ be the leaves of a tree. Among them, u_1, u_2, and u_3 belong to the same laboratory; u_4, and u_5 belong to the other one. u_6 is a teacher of the department. So, u_1, u_2, and u_3 are combined into group G_7 ($= \{u_1, u_2, u_3\}$); and u_4 and u_5 are combined into group G_8 ($= \{u_4, u_5\}$). The group of the department G_9 is $\{u_1,\ldots,u_6\}$ which is the union of G_6, G_7, and G_8. The tree structure of the department is shown in Fig. 1.In the following, we propose the key assignment method for groups of principals with a tree structure. This will help us to realize the concept of secure broadcasting. We assume that every principal in a network system has only a secret key. The main purpose of the key assignment is to provide that every group G_i has a group key k_i and every principal in G_i can derive the group key from his secret key, but a principal outside G_i cannot. In the group tree, $G_i \subset G_j$ if the node of G_i is the descendant of the node G_j in the tree. It means that one knowing the group key of G_i can also derive the group key of G_j in the tree.

We assume that there is a center authority (CA), which is responsible for key generation and key distribution. The key generation and derivation algorithms for the groups in the tree structure can be described as follows.

2.2. Key generation algorithm

CA first selects two large prime numbers, p and q, satisfying the RSA assumption [5]. Let $N = p \times q$. CA travels the nodes in the tree of hierarchical principal groups from the root to leaves, and from left to right.

1. If the node is G_u which is the root of the tree, then CA assigns a random number k_u (mod N) as the group key of G_u, and selects a pair of (T_u, S_u) such that $T_u \cdot S_u = 1$ [mod $\phi(N)$], where T_u is public and S_u is secret.
2. If the node G_i is not the root or a leaf, we assume that node G_j is the parent of node G_i and the group key of G_j is k_j. CA computes $k_i = (k_j)^{S_j}$ (mod N) as the group key of G_i and selects a pair of (T_i, S_i) such that $T_i \cdot S_i = 1$ (mod $\phi(N)$), where T_i is public and S_i is secret.
3. If the node G_i is a leaf (the group contains only one principal) of the tree, we assume that node G_j is the parent of node G_i and the group key of G_j is k_j. CA computes $k_i = (k_j)^{S_i}$ (mod N) as the group key of G_i (the secret key of the principal).

2.3. Key derivation algorithm

Assume that u_s is a principal in the group G_i, who wants to get the group key k_i of G_i. We assume that the principal corresponds to the group G_s, i.e. $G_s = \{u_s\}$, and G_f is the parent of G_s.

1. If $G_s = G_i$, then the group key k_i of G_i is equal to k_s (the secret key of the principal).
2. If $G_s \neq G_i$, then $G_s \subset G_f \subseteq G_i$. u_s who owns the group key k_s can compute the group key k_f of G_f by $k_f = (k_s)^{T_f} \pmod N$. Upon the group key k_f of G_f is determined, the group key k_r of G_r can be computed by $k_r = (k_f)^{T_r} \pmod N$ where G_r is the parent of G_f. The same processes are repeated until the group key k_i is derived.

2.4. Security analysis

We herein consider the security problem that whether a principal outside the group G_i can derive the group key k_i of G_i. The first possible case is that a principal outside the network wants to get the group key k_i of G_i. In this case, he can obtain only the public information T_i. Assume group G_j is the child of G_i. Therefore, $k_i = (k_j)^{T_i} \pmod N$. Because k_i is unknown, he is unable to get the exact group key k_i of G_i. The second possible case is that a principal outside group G_i wants to get the group key k_i of G_i. The possible solution is first to get the group key k_t of G_t where G_t is the ancestor of G_i. And then, he tries to compute the group key k_i of G_i from k_t of G_t. Let us consider the relation between the group key of parent G_f and the group key of child G_c. We assume that the group key k_f of G_f is known. Because $k_f = (k_c)^{T_f} \pmod N$, it is infeasible to compute k_c as a result of the difficulty of factoring the product of two large prime numbers [5]. So, it is infeasible to compute the group key of children from the group key of parent. Therefore, the principal outside group G_i cannot get the group key k_i of G_i.

From the characteristics of the tree structure, we develop a key management scheme for it. Every principal in a group can recover the group key of the group by using his secret key, while principals outside this group cannot get the group key. Note that each group key is randomly decided from the top to the bottom in the tree structure, and will be used as the decryption key of a cryptosystem. In general, the decryption key of a cryptosystem needs to satisfy some special conditions, such as RSA which the decryption key, d, needs to satisfy $e \cdot d = 1 \pmod{\phi(N)}$ where e is the encryption key and N is the product of two large prime numbers. Therefore, we need a mapping function $F(\cdot)$ so that $F(k_i) = d_i$, where k_i is the group key of G_i and d_i is the extended group key which is suitable to serve as the decryption key of a cryptosystem. For example we define $F(k)$ is the function of the maximum prime number which is less than or equal to k. Let $N_i = p \cdot q$ and $\phi(N_i) > k_i$, where p and q are two large prime numbers. Then d_i [$= F(k_i)$] is a prime number which is less than $\phi(N_i)$. Thus there exists an e_i so that $e_i \cdot d_i = 1 \pmod{\phi(N_i)}$. This means the decryption key d_i of a RSA cryptosystem can be derived from k_i. In the next section, we will discuss secure broadcasting in a hierarchical tree of principals.

3. Secure broadcasting in a hierarchical tree of principals

A principal can be a user, a station, etc., depending on the environment where messages are broadcasted. In this section, we discuss secure broadcasting in hierarchical groups of principals with tree structure. The communicating protocol is stated as follows. We assume that each group G_i in the tree structure has a public value e_i so that (e_i, d_i) is the pair of public key and secret key of a public-key cryptosystem. Let $E_e(\cdot)$ denote the encryption function of the public-key cryptosystem using the public key e, and $D_d(\cdot)$ denotes the decryption function of the public-key cryptosystem using the secret key d. On the other hand, we use $E_{CK}'(\cdot)$ and $D_{CK}'(\cdot)$ to denote the encryption and decryption functions of a private-key cryptosystem with the symmetric key CK [4], e.g. DES. Without loss of generality, let u_0 be the sending principal of a broadcasting, $G_1,\ldots,G_k$ be the groups of legal receiving principals in a hierarchical tree where $G_i \not\subset G_j$ for all $i,j \in \{1,\ldots,k\}$, $i \neq j$, and there does not exist any selection of $G_i,\ldots,G_j$ such that $G_i \cup \ldots \cup G_j$ is in the hierarchical tree.

In our secure broadcasting protocol, the sending principal broadcasts the ciphertext (C_1, C_2) in the computer network, where C_1 is used by the legal receiving principals to obtain the communication key CK, and C_2 is the ciphertext of message M, encrypted by CK. A message can be directly encrypted in C_1 when the message space is not larger than the communication key space. In this case, C_2 is not needed.

3.1. The encryption algorithm

Case I: the message space is larger than the communication key space.
Case II: the message space is not larger than the communication key space.
Input: message M
Output: (Case I): ciphertext (C_1, C_2).
(Case II): ciphertext (C_1).

Step 1. (Case I): u_0 selects a random number CK as the communication key.
(Case II): let $CK = M$.
Step 2. u_0 computes $E_{e_i}(CK)$ which is the ciphertext of CK encrypted by using the public key e_i of the public-key cryptosystem, for $1 \leq i \leq k$.
Step 3. u_0 generates a polynomial $f(x)$ of degree $k-1$

interpolating those points $(ID_i, E_{e_i}(CK))$ for $1 \le i \le k$. Assume that $f(x) = t_{k-1} \cdot x^{k-1} + \ldots + t_1 \cdot x + t_0$.
Step 4. u_0 computes $t_k = E_{CK}'(CK)$ which is the ciphertext of CK encrypted by a private-key cryptosystem with the secret key CK. Let $C_1 = (t_0, \ldots, t_k)$.
Step 5. (Case I): u_0 divides M into blocks $M_1, M_2, \ldots, M_h$ where M_i is suitable for encryption, for $1 \le i \le h$. And then u_0 computes $C_2 = [E_{CK}'(M_1), E_{CK}'(M_2), \ldots, E_{CK}'(M_h)]$ where $E_{CK}'(M_i)$ is the ciphertext of message M_i encrypted by a private-key cryptosystem with the secret key CK, for $1 \le i \le h$.
(Case II): Go to step 6.
Step 6. (Case I): u_0 broadcasts the ciphertext (C_1, C_2) through the network.
(Case II): u_0 broadcasts the ciphertext (C_1) through the network.

3.2. The decryption algorithm

Input. (Case I): Ciphertext (C_1, C_2).
(Case II): Ciphertext (C_1)
Output. Message M

Step 1. (Case I): The legal receiving principal u_j who belongs to G_i receives (C_1, C_2), for $1 \le i \le k$.
(Case II): The legal receiving principal u_j who belongs to G_i receives (C_1), for $1 \le i \le k$.
Step 2. u_j finds the correct extended group key d_i and communication key CK by

1. Computes $E_{e_i}(CK) = f(ID_i)$ where $f(x)$ can be obtained from C_1.
2. Computes k_i from his secret key by the key derivation algorithm in Section 2.
3. Computes the extended group key d_i from k_i,
4. Computes CK by $D_{d_i}[E_{e_i}(CK)] = CK$,
5. Check whether the group key d_i is correct by testing whether $D_{CK}'(t_k) = D_{CK}'[E_{CK}'(CK)] = CK$.

Step 3. (Case I): u_j Computes message $M = M_1 \| M_2 \| \ldots \| M_h$, where $\|$ denotes concatenation and $M_i = D_{CK}'[E_{CK}'(M_i)]$, for $1 \le i \le .h$.
(Case II): u_j Obtains message $M = CK$.

3.3. Security analysis

Our secure broadcasting protocol may be attacked by an intruder using the following means. We show that our protocol is secure against these attacks.

3.3.1. Attack 1: Attack to find the extended group key d_i of G_i.

First, the intruder may try to find d_i of G_i from the ciphertext $E_{e_i}(CK)$. So, if the public-key cryptosystem used in our protocol is secure, then the intruder will fail. On the other hand, the intruder may try to find k_i of G_i from the key management of the hierarchical groups of principals. However, we have proven the security of our proposed key management scheme in Section 2.

3.3.2. Attack 2: Attack to find the communication key CK.

First, the intruder may try to find CK from the ciphertext $E_{e_i}(CK)$. So, if the public-key cryptosystem used in our protocol is secure, then the intruder will fail. On the other hand, the intruder may try to find CK from the ciphertext $E_{CK}'(CK)$ and $E_{CK}'(M)$. So, if the private-key cryptosystem used in our protocol is secure, then the intruder will fail.

3.3.3. Attack 3: Attack to find the message M.

The intruder may try to find message M from the ciphertext $E_{CK}'(M)$. So, if the private-key cryptosystem used in our protocol is secure, then the intruder will fail.

From the discussions above, we conclude that if the public-key cryptosystem and the private-key cryptosystem used in our protocol are secure, then our protocol is secure against these attacks.

4. Comparison with other protocols

In this section, we compare the size of broadcasted ciphertext (C_1, C_2) in our protocol with other well-known secure broadcasting protocols. First, we compare our protocol with PUBP and PRBP proposed by Chiou and Chen [9]. The size of C_2 in our protocol is the same as their protocols. The size of C_1 in their protocols is in $O(n)$ complexity where n is the number of legal receiving principals. In contrast, the size of C_1 in our protocol is only in $O(k)$ complexity where k is the number of groups. In general, a group in a broadcasting environment contains many members, that is, k is far smaller than n. (This is the environment where secure broadcasting protocols are frequently used.) In this case, the improvement is significant. In the worst case that every group contains only a legal receiving principal, our protocol is at least as good as other protocols. Secondly, we compare our protocol with other broadcasting protocols which employ a central authority server (CAS) in a computer network [10,11]. In these protocols, the size of C_2 is the same as the one used in our protocol. The ciphertext C_1 in these protocols is saved in a nonce public table. However, the nonce public table still needs to be broadcasted for every transmission in these protocols, and its size is in $O(n)$ complexity where n is the number of legal receiving principals. Therefore, the total size being broadcasted in these protocols is much larger than that in our protocol. The comparison of existing protocols with our protocol is summarized in Table 1.

5. Conclusions

In this paper, we propose an efficient secure broadcasting protocol. In the protocol, only one packet needs to be

Table 1
Summary of secure broadcasting protocols' performance

Protocol	Size of broadcasted packet (C_1)	Size of nonce public table	Total size being broadcasted
Chiou and Chen's PUBP [9]	$O(n)$	NA	$O(n)$
Chiou and Chen's PRBP [9]	$O(n)$	NA	$O(n)$
Chang and Wu [10]	NA	$O(n)$	$O(n)$
Chang and Wu [11]	NA	$O(n)$	$O(n)$
Our protocol	$O(k)$	NA	$O(k)$

transmitted for every broadcast. The length of the packet is only in $O(k)$ which is small, comparing with $O(n)$ in other protocols. Every receiving principal in our protocol holds only one secret key which can be used to derive the group keys of his ancestral groups in the hierarchical tree structure. Furthermore, our protocol also ensures that a principal outside a legal group fails to get the group key.

Acknowledgements

This work was supported in part by the National Science Council, Taiwan, under contract NSC-85-2622-E-009-006R.

References

[1] D.E. Denning, Cryptography and Data Security, Addision–Wesley, Reading, MA, 1982.

[2] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Trans. Inf. Theory IT-22 (6) (1976) 644–654.

[3] T. ElGamal, A public-key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Inform. Theory IT-31 (4) (1985) 469–472.

[4] National Bureau of Standards, Data Encryption Standard, FIPS Publication 46, NBS, 1977.

[5] R.L. Rivest, A. Shamir, L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126.

[6] I.S. Gopal, J.M. Jaffe, Point-to-multipoint communication over broadcast links, IEEE Trans. Commun. IT-32 (9) (1984) 1034–1044.

[7] I. Ingemarsson, D.T. Tang, C.K. Wong, A conference key distribution system, IEEE Trans. Inform. Theory IT-28 (5) (1982) 714–720.

[8] C.S. Laih, J.Y. Lee, L. Harn, A new threshold scheme and its application in designing the conference key distribution cryptosystem, Info. Process. Lett. 32 (3) (1989) 95–99.

[9] G.H. Chiou, W.T. Chen, Secure broadcasting using the secure lock, IEEE Trans. Software Engng IT-15 (8) (1989) 929–934.

[10] C.C. Chang, T.C. Wu, A broadcasting cryptosystem based upon Euclidean geometry, Int. J. Policy Inf. 13 (2) (1989) 179–186.

[11] C.C. Chang, T.C. Wu, Broadcasting cryptosystem in computer networks using interpolating polynomials, Comput. Systems Sci. Engng 6 (3) (1991) 185–188.