

# Secure Broadcasting Using the Secure Lock

GUANG-HUEI CHIOU, MEMBER, IEEE, AND WEN-TSUEN CHEN, MEMBER, IEEE

**Abstract**—In this paper, we propose the concept of a secure broadcasting, effected by means of a secure lock, on broadcast channels, such as satellite, radio, etc. This lock is implemented by using the Chinese Remainder Theorem (CRT). With the secure lock, we have the following advantages. First, only one copy of the ciphertext is sent. Second, the deciphering operation is efficient. Third, the number of secret keys held by each user is minimized. By using the secure lock, we also present protocols for secure broadcasting, based on the public-key cryptosystem as well as the private-key cryptosystem.

**Index Terms**—Chinese Remainder Theorem (CRT), cryptosystem, secure broadcasting, secure lock, session key.

## I. INTRODUCTION

THE main property of a broadcast channel is that a single transmission from a source station may be received simultaneously by many destination stations. Examples of broadcast channels include various forms of local area networks, satellite channels, packet radio networks, etc. Frequently, we need to send a secret message to many people at the same time. Applications of this type are called *secure broadcasting* applications. These applications, such as document distribution, teleconferencing, etc., have considerably changed the nature of data traffic. In the future, the volume of secure broadcasting data traffic will increase significantly.

In this paper, we assume that there are  $n$  users, called a group  $U$ , in a broadcast network. Let  $ek_i$  be the enciphering key and  $dk_i$  be the deciphering key of user  $u_i$ . Generally, in the private-key cryptosystem,  $ek_i$  is equal to  $dk_i$  and in the public-key cryptosystem,  $ek_i$  is different from  $dk_i$ . Each user can communicate directly with every other user through the broadcast channel. But, working on a "need to know" basis, the sender may want to send a message  $M$  just to a group of users  $G$ , within  $U$ . The ciphertext of  $M$  should only be decipherable by the users in  $G$  but not every other user in  $U$ . Most of the cryptosystems that have been proposed are concerned with the point-to-point type of communication [3].

If one uses point-to-point approach for secure broadcasting of a message, then for each user in  $G$ , the sender must perform encryption and send the ciphertext separately. Clearly, using this approach is very inefficient since multiple copies of the ciphertext must be sent [3], [4].

Another approach is that each group  $G$  within  $U$  is associated with a *group key* for secure communication and this key is known only to users in  $G$ . In the public-key cryptosystem, this group key is a pair of keys; one is a public enciphering key and the other is the deciphering key. Since this group key is only known by the users in  $G$ , the sender can use it to encipher the message before broadcasting. Clearly, using this method only one copy of the ciphertext needs to be sent. However, the sender must hold up to  $(2^{n-1} - 1)$  group keys, one for each possible nonempty subgroup of  $U$ . The disadvantage of this solution is that each user needs to hold possibly too many group keys.

To avoid keeping too many secret group keys we can arbitrarily select a key, called *session key*, to encipher the message. In the public-key cryptosystem, this session key will be a pair of keys. Let  $\hat{e}$  be the *enciphering session key* and  $\hat{d}$  be the *deciphering session key* and let  $R_i$  be the ciphertext of  $\hat{d}$  which is enciphered under  $u_i$ 's enciphering key  $ek_i$ ; i.e.,  $R_i = E_{ek_i}(\hat{d})$  where  $u_i$  is a user in  $G$ . We can directly concatenate  $m$  pieces of  $R_i$  together and append them onto the sent-out message. In this method, only one copy of ciphertext is needed because the message is enciphered by the session key, and no extra keys need to be kept in secret because the session key is selected for each transmission. However, it is inefficient in that the receiver needs to decipher each piece of  $R_i$ , in average of  $m/2$  decipherings, with his deciphering key until he gets the session key  $\hat{d}$ .

To avoid doing so many times of deciphering, we can modify the above method by labeling each  $R_i$  in certain predefined and advertised sequence so that the receiver can locate his  $R_i$  and decipher it directly. However, the  $m$  labels will increase the length of sent-out message. And by using traffic analysis, those unciphered labels will disclose some secret information to the intruder, such as the purpose or the security level of the message.

In this paper, we shall propose a secure lock to lock the deciphering session key. Using this secure lock only a single deciphering operation is needed to obtain the session key. In Section II, the secure lock, with implementation based on the Chinese Remainder Theorem (CRT), is presented. In Section III, two efficient protocols for secure broadcasting are presented. One is based on the public-key cryptosystem and the other on the private-key cryptosystem. In Section IV, we examine the security of our proposed protocols. Finally, conclusions are given in Section V.

Manuscript received June 15, 1987.

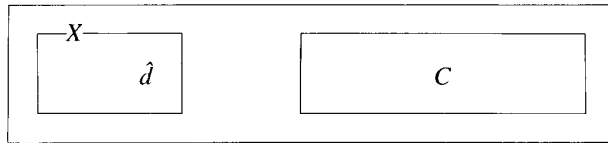
The authors are with the Institute of Computer Science, National Tsing-Hua University, Hsinchu, Taiwan 30043, Republic of China.

IEEE Log Number 8928900.

## II. OUR PROPOSED APPROACH

Consider a broadcasting system consisting of a broadcast network and a group  $U$  containing  $n$  users. Let the  $n$  users be denoted as  $u_1, u_2, \dots, u_n$ . Let  $G$  denote a nonempty subgroup of  $m$  users within  $U$ ,  $m \geq 2$ . In order to send a message  $M$  to the users in  $G$  securely,  $M$  is enciphered before being sent. It is clear that if there exists a secure method which can send the deciphering session key together with the ciphertext of  $M$ , then the disadvantages of secure broadcasting can be easily solved. Since the message is enciphered by the enciphering session key, only one copy of ciphertext needs to be sent, and since the session key is selected for each transmission, no extra keys need to be kept in secret. Based on the above idea, we propose a new approach in which a lock is superimposed onto the front of the sent-out message to lock the deciphering session key, and only matches the keys of the users in  $G$ .

According to the locking concept, we can describe the format of the sent-out message as follows:



where

$X$  is the lock.

$\hat{d}$  is the deciphering session key; i.e.,  $M = D_{\hat{d}}(C)$ .

$C$  is the ciphertext of the message  $M$ ; i.e.,  $C = E_{\hat{e}}(M)$ .

$\hat{e}$  is the enciphering session key.

The lock  $X$  should satisfy the following requirements. First, it can only be opened by users in  $G$  since we only allow users in  $G$  to read the message. Second, it must be dependent on  $\hat{d}$ . Otherwise, each time the sender broadcasts a message to the users in  $G$  using the same lock, it will disclose some secret information to the intruder, such as the purpose or the security level of the message. Thus, in order to meet these requirements the lock must be functionally dependent on both the ciphering key used between the sender and users in  $G$  and the deciphering session key  $\hat{d}$ .

The remaining problem is how to construct the secure lock. A lock construction method based on the Chinese Remainder Theorem will be presented in the next subsection.

### A. The Construction and Performance of the Secure Lock

Before presenting the construction method of the secure lock, let us briefly recall the Chinese Remainder Theorem.

**Chinese Remainder Theorem (CRT):** Let  $N_1, N_2, \dots, N_n$  be  $n$  positive integers that are pairwise relatively prime,

and let  $R_1, R_2, \dots, R_n$  be positive integers, and let  $L = N_1 * N_2 * N_3 * \dots * N_n$ . Then the set of congruous equations

$$X \equiv R_1 \pmod{N_1}$$

$$X \equiv R_i \pmod{N_i} \quad \text{where “} \equiv \text{” is the congruous sign}$$

$$X \equiv R_n \pmod{N_n}$$

have a common solution  $X$  which is in the range of  $[1, L - 1]$  and

$$X = \left( \sum_{i=1}^n (L/N_i) * R_i * f_i \right) \pmod{L}$$

$$\text{where } 1 \equiv f_i * (L/N_i) \pmod{N_i}.$$

For further information on the CRT, the reader can consult [1], [2].

Now, based on the CRT, the construction of the secure lock is described as follows. Let  $\hat{d}$  be the deciphering session key and let  $R_i$  be the ciphertext of  $\hat{d}$  which is enciphered with  $u_i$ 's enciphering key  $ek_i$ ; i.e.,  $R_i = E_{ek_i}(\hat{d})$ . Let  $G$  be a group of users within  $U$ , and  $N_1, N_2, \dots, N_n$  be pairwise relatively prime integers which are publicly known in the system.  $N_i$  is associated with the user  $u_i$  and greater than  $R_i$ . Then we have the congruous equations as follows:

$$X \equiv R_1 \pmod{N_1}$$

.

$$X \equiv R_i \pmod{N_i} \quad \text{for all } u_i \text{ in } G$$

.

$$\text{for all } R_i, R_i \leq N_i$$

$$X \equiv R_m \pmod{N_m}$$

$$\text{where } R_i = E_{ek_i}(\hat{d}).$$

From the above congruous equations, the common solution  $X$  can be computed using the CRT.

Let the common solution  $X$  of the CRT be the secure lock. According to the congruous equations of the CRT,  $u_i$  can compute  $R_i$  from the lock  $X$ . Therefore,  $u_i$  can decipher  $R_i$  to get  $\hat{d}$  in one single deciphering operation.

To see that the lock  $X$  satisfies the requirements mentioned previously in this section, first we note that each of the congruous equations is associated with a user in  $G$ . In order to obtain  $\hat{d}$ , the user needs the associated deciphering key  $dk_i$  to decipher  $R_i$  for opening the lock  $X$ . Therefore, only the users in  $G$  have the deciphering key to open this lock  $X$ . Second, since  $R_i = E_{ek_i}(\hat{d})$ , we know that  $R_i$  is functionally dependent on  $\hat{d}$ . Therefore, the second requirement is also met. Finally, since the user only needs his own secret key to open the lock, users in the system need not to hold extra keys. Thus, the lock  $X$  is the lock which we want to construct.

In summary, appending the lock  $X$  onto the sent-out ciphertext has the following advantages.

- 1) Only one copy of the ciphertext is sent.
- 2) For each user in the system, no extra secret keys are needed.

### B. The Performance of the Construction Method

Since the performance of our proposed construction method is dependent on the CRT algorithm, we shall first consider the complexity of the CRT algorithm. Let  $N_1, N_2, \dots, N_k$  denote the  $k$  moduli in the CRT and assume that each  $N_i$  at most has  $b$ -bit. Then the complexity of the CRT algorithm is  $O_B(M(bk) \log k) + O_B(kM(b) \log b)$  (see [1, Theorem 8.21]), where  $M(n)$  is the time used to multiply two  $n$ -bit integers, and  $O_B$  is measured in bit operations.

Since the complexity of the CRT algorithm is dominated by  $M(n)$ , we shall examine the algorithm for multiplying two  $n$ -bit integers. Assume that a multiplier and an adder, 32 bits, are available. Using divide-and-conquer strategy, one can multiply two  $n$ -bit integers  $X, Y$  as follows [1].

Let

$$\begin{aligned} X &= a * 2^{n/2} + b \\ Y &= c * 2^{n/2} + d \\ U &= (a + b) * (c + d) \\ V &= a * c \\ W &= b * d. \end{aligned}$$

Then

$$XY = V * 2^n + (U - V - W) * 2^{n/2} + W.$$

Thus,

$$M(n) = \begin{cases} 1 \text{ multiplication} & \text{if } n \leq 32 \\ 3M(n/2) + 4A(n) + 2A(n/2) + 2S \\ \quad = 3M(n/2) + 5A(n) + 2S, & \text{if } n > 32 \end{cases}$$

$$A(n) = \begin{cases} 1 \text{ addition} & \text{if } n \leq 32, \\ k \text{ addition} & \text{if } (k-1) * 32 \leq n \leq k * 32, \end{cases}$$

and  $S = 1$  shift.

Now, let us multiply two 1024-bit integers

$$\begin{aligned} M(1024) &= 3M(512) + 5A(1024/32) + 2 \text{ shifts} \\ &= 3(3M(256) + 5A(512/32) + 2 \text{ shifts}) \\ &\quad + 5 * 32 \text{ additions} + 2 \text{ shifts} \\ &\vdots \\ M(1024) &= 243 \text{ multiplications} \\ &\quad + 2110 \text{ additions} + 34 \text{ shifts.} \end{aligned}$$

According to the CRT algorithm and using divide-and-conquer strategy, we can say our proposed construction method is efficient. If the parallel multiplication hardware and the special CRT computing hardware are available, then the computing time can be improved significantly [5].

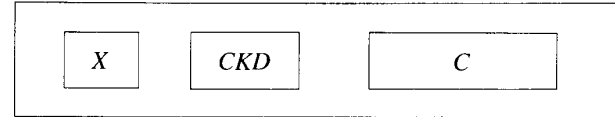
The security of the locking concept and its detailed solution will be presented in the following section.

## III. OUR PROPOSED BROADCASTING PROTOCOLS

First, we shall describe a protocol for secure broadcasting which is based on a public-key cryptosystem. Next, we shall describe a private-key based protocol, and then compare these two protocols.

### A. Public-Key Cryptosystem Based Broadcast Protocol (PUBP)

Assume each user  $u_i$  in the broadcast system has been assigned an integer  $N_i$ . Let  $N_1, N_2, \dots, N_n$  be pairwise relatively prime, and public in the broadcast system. Let  $u_s$  be the sender. Let  $ek_i$  be  $u_i$ 's public enciphering key and  $dk_i$  be  $u_i$ 's secret deciphering key. The format of the sent-out message is given as follows:



where

$X$  is the lock, being the solution of the following congruous equations:

$$X \equiv E_{ek_i}(\hat{d}) \pmod{N_i}, \quad \text{for all } u_i \text{ in } G.$$

$CKD$  is the ciphertext of  $\hat{d}$  which is enciphered by  $\hat{e}$ ; i.e.,  $CKD = E_{\hat{e}}(\hat{d})$ , and used by the receivers to check whether the sender wants to communicate with him or not.

$C$  is the ciphertext of the message  $M$  which is enciphered by  $\hat{e}$ ; i.e.,  $C = E_{\hat{e}}(M)$ .

*The Encryption Algorithm:*

*Input:* The secret message  $M$ , the public relatively prime integers  $N_1, N_2, \dots, N_n$ , and the public enciphering keys of the users in  $G$ .

*Output:* The message to be broadcast; i.e.,  $X, CKD$ , and  $C$ .

*Step 1:* For the secret message  $M$ , the sender arbitrarily selects an enciphering session key  $\hat{e}$  for encrypting the message  $M$ ; i.e.,  $C = E_{\hat{e}}(M)$ , and a deciphering session key  $\hat{d}$  for decrypting the ciphertext  $C$ ; i.e.,  $M = D_{\hat{d}}(C)$ .

*Step 2:* Use the CRT algorithm to compute the common solution from the following congruous equations:

$$\text{i.e., } X \equiv R_1 \pmod{N_1}$$

$$X \equiv R_i \pmod{N_i} \quad \text{for all } u_i \text{ in } G$$

$$X \equiv R_m \pmod{N_m}$$

where  $R_i$  is the ciphertext of  $\hat{d}$ , which is enciphered by  $ek_i$ ; i.e.,  $R_i = E_{ek_i}(\hat{d})$ .

*Step 3:* Compute the  $CKD$  and encipher  $M$  with  $\hat{e}$ ;

$$\text{i.e., } CKD = E_{\hat{e}}(\hat{d}), \quad \text{and } C = E_{\hat{e}}(M).$$

*Step 4:* Broadcast the message; i.e., send  $X$ ,  $CKD$ , and  $C$  out.

*The Decryption Algorithm:*

*Input:* The sent-out message, and the deciphering key  $dk_i$  of the receiver  $u_i$ .

*Output:* The secret message  $M$ .

*Step 1:* Compute  $\hat{d}$  from  $X$  by using the receiver's deciphering key,  $dk_i$ . Then compare  $\hat{d}$  to the  $CKD$ ;

$$\begin{aligned} \text{i.e., to compute } \hat{d} &= D_{dk_i}(X \pmod{N_i}) \\ &= D_{dk_i}(R_i), \end{aligned}$$

and check whether  $D_{\hat{d}}(CKD)$  is equal to  $\hat{d}$  or not. If not, the receiver knows that this message was not sent to him, and stops.

*Step 2:* Decipher the ciphertext  $C$  with  $\hat{d}$ ; i.e., compute

$$M = D_{\hat{d}}(C).$$

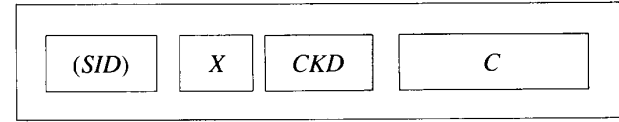
*Step 3:* End.

The public-key cryptosystems, one who sends a message to a receiver  $u_i$ , shall encipher the message under  $u_i$ 's public enciphering key. Then the receiver can decipher the ciphertext under his secret deciphering key. From the above decryption algorithm, in order to compute  $\hat{d}$ , the user  $u_i$  in  $G$  only needs to decipher  $R_i$  with his own deciphering key  $dk_i$ . Hence, for each user in the system, only one key needs to be kept in secret. To compute this to the group key approach mentioned in Section I, our proposed solution, based on public-key cryptosystems, needs only  $O(n)$  secret keys instead of  $O(2^n)$ .

#### B. Private-Key Cryptosystem Based Broadcast Protocol (PRBP)

In public-key cryptosystems, the receiver can use his own secret deciphering key to open the lock. However, in private-key cryptosystems, since the receiver has  $n - 1$  deciphering keys, one for each other user in the system, the receiver needs to select the correct deciphering key to open the lock. Consequently, the sender's ID must be included in the sent-out message.

Let  $ek_i$  be the enciphering key used by the sender  $u_s$  to encipher a secret message  $M$  and send it to the receiver  $u_i$ . Let  $dk_s$  be the deciphering key used by the receiver  $u_i$  to decipher a ciphertext  $C$  which is sent by the sender  $u_s$ . The format of the sent-out message for the private-key cryptosystem is given as follows:



where

$X$ ,  $CKD$ , and  $C$  retain the same definition as in the Public-Key Cryptosystem Based Protocol.

$SID$  is the sender's identification number.

The encryption algorithm of the PRBP is the same as that of the PUBP. In private-key cryptosystems, the receiver needs to select the deciphering key to decipher the ciphertext which is sent by the sender  $u_s$ . Therefore, an additional step to select the deciphering key for opening the lock  $X$  is inserted at the beginning of the PUBP decryption algorithm to form the PRBP decryption algorithm. This additional step is shown as follows.

*Step 0:* Receiver  $u_i$  selects the deciphering key  $dk_s$  to compute  $\hat{d}$  from  $X$ , where  $s$  is obtained directly from  $SID$ .

It is clear that the PRBP and the PUBP have a strikingly similar solution for secure broadcasting. According to the above solutions, we only need to encipher the message once and only send out one ciphertext.

In private-key cryptosystems, each user has  $(n - 1)$  secret keys since for each other user  $u_i$ , the sender needs a key  $ek_i$  for encrypting the sent-out message. Compared to the group key approach mentioned in Section I, our proposed solution based on the private-key cryptosystem needs only  $O(n^2)$  secret keys instead of  $O(2^n)$  keys.

#### IV. SOME ATTACKS ON OUR PROPOSED PROTOCOLS

In this section, we state that the security of our proposed broadcasting protocols is the same as that of the cryptosystem used. In our protocols, for each user in  $G$ , we encipher the deciphering session key  $\hat{d}$  with the receiver's enciphering key  $ek_i$  separately and mix these ciphertexts by the CRT algorithm. Thus, the congruous equations are as follows:

$$\begin{aligned} X &\equiv E_{ek_1}(\hat{d}) \pmod{N_1} \\ &\vdots \\ X &\equiv E_{ek_i}(\hat{d}) \pmod{N_i} \\ &\vdots \\ X &\equiv E_{ek_m}(\hat{d}) \pmod{N_m} \end{aligned} \quad (1)$$

where the values of  $X$  and  $N_i$  are public. In the public-key cryptosystem,  $ek_i$  is also public.

Since the  $X$  and  $N_i$  are public, the congruous equations (1) can be transformed into the following set of equations:

$$\begin{aligned} R_1 &= E_{ek_1}(\hat{d}) \\ &\vdots \\ R_i &= E_{ek_i}(\hat{d}) \\ &\vdots \\ R_m &= E_{ek_m}(\hat{d}). \end{aligned} \quad (2)$$

A cipher is breakable if it is possible to deduce the plaintext or key from a given ciphertext. In general, the basic measure of a secure broadcasting protocol is whether the broadcasting protocol downgrades the security of the cipher which is applied to the protocol, or not. Ciphers are usually considered acceptable if they can withstand a known-plaintext attack with the assumption that the cryptanalyst has an arbitrary number of plaintext-ciphertext pairs [2]. In other words, any given secure broadcasting protocol which uses a cipher that can withstand the known-plaintext attack, and can still withstand the known-plaintext attack, is secure.

Our proposed protocols may be attacked by an intruder using the following means.

*Attack 1: Attack to find the deciphering session key  $\hat{d}$ .*

In this case, we stipulate that the intruder does not belong to  $G$ . An intruder must try to solve the equations shown in (2) to get  $\hat{d}$ . But, since each equation in (2) is enciphered with a different secret key, he must break the cipher  $E$  which is secure against the known-plaintext attack. Thus, our protocols do not downgrade the security of  $E$ .

*Attack 2: Attack to obtain the secret key.*

The intruder may be a legitimate receiver in  $G$ . He tries to obtain the secret key which is used by another receiver  $u_i$ . If the intruder has the secret key of user  $u_i$ , then he will know all of the messages sent to  $u_i$ . In the private-key cryptosystem, the secret key used between the sender and receiver  $u_i$  is  $ek_i$ . Even though the intruder knows the ciphertext-plaintext pairs  $(R_i, \hat{d})$ , he also has to break the cipher  $E$  used in (2) to get  $ek_i$ . Since the cipher used in our protocols is secure against the known-plaintext attack, the result is the same as that of attack 1.

In the public-key cryptosystem, the secret key is the deciphering key  $dk_i$  which is associated with the public enciphering key  $ek_i$ . Since the public-key cipher can withstand the known-plaintext attack, (actually, the public-key cryptosystem is secure against the chosen-plaintext attack), the intruder can not obtain more information from (2). Therefore, the intruder intending to get the secret key  $dk_i$  from (2) has the same difficulty of trying to break the cipher  $E$ .

From the above discussion, we conclude that if the cipher which is used in our proposed protocols can withstand the known-plaintext attack, then our proposed protocols also can withstand the known-plaintext attack, therefore our proposed broadcasting protocols are secure.

## V. CONCLUSION

In this paper, we have proposed the locking concept and a secure lock implemented based on the Chinese Remainder Theorem. By using this secure lock, we have proposed secure broadcasting protocols for solving the secure broadcasting problem. It is clear that these are distributed protocols since each user can compute the lock by himself. In the last section, we have shown that the security of the cipher which is used in our protocols is not downgraded. So, our proposed protocols are secure.

The first point of merit in our proposed protocols is their efficiency because only one copy of the ciphertext is broadcasted.

Second, in our proposed protocols, the number of secret keys which are kept by each user is equal to the number of secret keys which are needed in the one-to-one applications. Actually, the same secret keys are used. Thus, the number of secret keys which are used in our protocol is minimized.

The third merit is that the sent-out message need not contain the names of the receivers. Furthermore, in the public key based protocol, the sender's ID (SID) is also enciphered. Since the receiver's addresses are inherent in the ciphertext of  $X$ , the security level of the sent-out message is not disclosed and the attack using traffic analysis is also infeasible.

Finally, the Chinese Remainder Theorem is used to implement the secure lock. However, it is efficient only when the number of users in a group is small, since the time to compute the lock and the length of the lock (hence the transmission time) is proportional to the number of users. In most typical applications, the number of users in a group is small, therefore the secure lock implemented by the Chinese Remainder Theorem is suitable. In case that there are a large number of users in a group, we can partition the users into a number of subgroups with suitable size. Then a lock is constructed for each subgroup by using the Chinese Remainder Theorem. A lock is sent while the lock for the next subgroup is constructed, so that construction and transmission time of these locks can be overlapped.

## REFERENCES

- [1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley, 1974.
- [2] D. E. Denning, *Cryptography and Data Security*. Reading, MA: Addison-Wesley, 1982.
- [3] I. S. Gopal and J. M. Jaffe, "Point-to-multipoint communication over broadcast links," *IEEE Trans. Commun.*, vol. COM-32, no. 9, pp. 1034-1044, 1984.

- [4] A. S. Tanenbaum, *Computer Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1981.
- [5] F. J. Taylor, "Residue arithmetic: A tutorial with examples," *Computer*, pp. 50-62, May 1984.



**Guang-Huei Chiou** (M'88) was born in Taiwan, Republic of China, on April 17, 1954. He received the B.S. degree in computer science from Tamkang University, Taiwan, in 1977, and the M.S. and Ph.D. degrees in computer science from National Tsing-Hua University, Taiwan, in 1979 and 1987, respectively.

Since 1987, he has been the Deputy Director of the division of research and development, Institute of Information Industry, Taiwan. His current research interests are data security, software engineering, and artificial intelligence.



**Wen-Tsuen Chen** (M'87) was born in Taiwan, Republic of China, on May 27, 1948. He received the B.S. degree in nuclear engineering from National Tsing-Hua University, Taiwan, in 1970, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1973 and 1976, respectively.

He joined the faculty of the Institute of Computer Science, National Tsing-Hua University in March 1976 as an Associate Professor. Since 1979, he has been a Professor and from 1983 to 1988 he served as the Director of the Institute. From 1984 to 1985, he was elected as an IEEE Distinguished Visitor in region 10. Since 1988, he has been a member of the technical consulting board of the Ministry of Education, Taiwan. His current research interests include computer networks, ISDN, multiprocessing systems, parallel algorithms, and software engineering.

Dr. Chen is a member of the Association for Computing Machinery.