NORTH-HOLLAND

A Two-Phase Encryption Scheme for Enhancing Database Security

Min-Shiang Hwang

Directorate General of Telecommunication Laboratories, Ministry of Transportation and Communications, Chung-Li, Taiwan

Wei-Pang Yang

Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan

In this article, we propose a two-phase encryption algorithm for data base systems. The system, a record-oriented cryptosystem, allows the encryption and decryption of fields within a record by means of writing and reading subkeys of fields. In addition, we develop two algorithms for cryptographic relational algebra in data base systems. Two simple methods of solving the key management problem in the subkey scheme are presented.

1. INTRODUCTION

Some of the advantages of using a data base are shared access, minimal redundancy, data consistency, data integrity (Pfleeger, 1989) so that data values are protected against accidental or malicious unauthorized changes, and controlled access, so that only authorized users are allowed to access data values. A data base management system (DBMS) with a security facility is designed to provide all of these advantages efficiently.

In general, there are four methods of enforcing data base security (Fernandez et al., 1980): physical security, such as storage medium safekeeping and fire protection (Coper, 1989); operating system security, such as the use of an access control matrix, capability-list, and accessor-list (Conway et al., 1972; Graham and Denning, 1972; Hwang and Yang, 1994); DBMS security, such as protection mechanisms and query modification (Ullman, 1988); and data encryption, such as the data encryption standard (DES) (National Bureau of Standards, 1977; Smid and Branstad, 1988) and the RSA scheme (Rivest et al., 1978). The first three methods, however, are not totally satisfactory solutions to the data base security problem, for the following four reasons. First, it is hard to control the disclosure of raw data because the raw data exist in readable form inside a data base (Davida et al., 1981). Second, it is invalid to operating system and DBMS security control to disclose sensitive data, because the sensitive data must frequently be backed up in storage media in case of system failure of disk crash. Third, it is hard to control the disclosure of confidential data in a distributed data base system. Fourth, it is hard to verify that the origin of a data item is authentic, because the original data may have been modified by an intruder (Lin et al., 1990; Lin, 1991). A practical solution to these problems is to use encryption methods to enforce data base security (Davida et al., 1981; Babad and Hoffer, 1980; Bayer and Metzger, 1976; Eriksson and Beckman 1983; Feigenbaum and Liberman, 1991; Gudes, 1980; Wagner et al., 1986).

Encryption data base security can solve these problems in the following manner. First, data are encrypted into ciphertext, which can only be decrypted with the proper decryption key, thus eliminating the problem of data disclosure. Second, an intruder cannot change the ciphertext without knowing the encryption keys; thus, the data authenticity problem is also resolved.

Data base security methods based on encryption include data base encryption systems with a single

Address correspondence to Prof. Wei-Pang Yang, Computer and Information Science, National Chiao Tung University, 1001 TA Hsueh Road, Hsinchu / TA Hsueh / 300 Taiwan, R.O.C.

J. SYSTEMS SOFTWARE 1995; 31:257–265

655 Avenue of the Americas, New York, NY 10010

0164-1212/95/$9.50
SSDI 0164-1212(94)00102-S

key (Gudes, 1980) and with subkeys (Davida et al., 1981). The first method needs a trusted centralized access control scheme with which to control all access to data stored in the data base system. All encryption and decryption is executed by the trusted access control scheme with a privacy key. In the second method, however, encryption/decryption is executed by users themselves with their own subkeys.

The first data base encryption/decryption system with subkeys was proposed by Davida et al. Their system, the so-called record-oriented cryptosystem, has the important property of having subkeys that allow the encryption and decryption of fields within a record. However, their scheme requires a random number generator for generating extra redundant bits in each field to withstand known plaintext attacks.

To eliminate this drawback, Lin and coworkers (1990, 1991) modified their method. Basically, they generalized the Chinese remainder theorem (CRT). Although their method does not require extra redundant bits in each field, it needs an extra privacy key for each record. There are two drawbacks to this scheme. One is that their scheme requires a great deal of storage space, because many privacy key values are needed to maintain better security. The other is that users or the DBMS needs to manage the privacy keys. Note that the number of records is in the thousands in most data bases.

In this article, we propose a two-phase encryption scheme for enhancing data base security. Our scheme does not require extending the raw data, nor is an extra privacy key needed for each record. The article is organized as follows. In Section 2, we introduce an architecture for a secure data base system. In Section 3, we describe both a one-way function and a subkey-enciphering method, and then apply the one-way function and the Chinese remainder theorem to a data base system to develop our two-phase encryption algorithm. In Section 4, we propose several algorithms for cryptographic relational operations. In Section 5, we propose two simple methods for solving the key management problem in the subkey scheme. Section 6 concludes the article.

2. THE SYSTEM ARCHITECTURE

In this section, we propose our system architecture, which is shown in Figure 1. There are six modules in the architecture: the input/output (I/O) module, field encryption/decryption (E/D) module, subkey

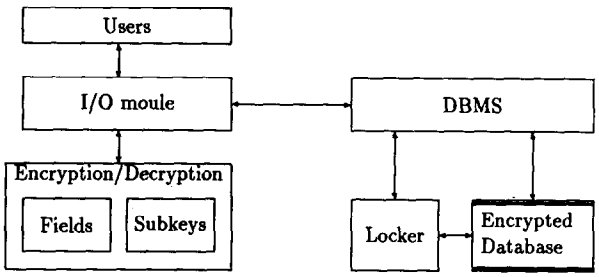

Figure 1. The architecture of the secured data base system.

decryption modules, DBMS, locker module, and encrypted data base. A user makes queries via the I/O module to access data objects in the data base. The I/O module is a general purpose I/O device (i.e., a terminal) for sending sensitive (encrypted) data objects to the E/D module for encrypting (decrypting). The E/D module has two components: the field E/D and the subkey module. The two phases of our E/D algorithm correspond to the operations of these components. In the first phase, the field E/D encrypts (decrypts) each individual field data item of a record. We can apply a symmetric cryptosystem (DES) (National Bureau of Standards, 1977; Smid and Branstad, 1988) or an asymmetric cryptosystem (RSA) (Rivest et al., 1978) to the field E/D module. We discuss these in detail in Section 3.2. In the second phase, the subkey module decrypts the ciphertext of a record into several data items of fields by use of the subkeys. The concept of subkeys was first proposed by Davida et al. The details of the subkey scheme are discussed in Section 3.1.

The DBMS module is a general purpose DBMS used for managing data base systems (Date, 1990). Another function of the DBMS is to control the operation of the encrypted data base by use of the locker module. The two responsibilities of the locker module are to encrypt data items of fields within a record with the subkeys of fields and to operate the encrypted data base by use of relational algebra. We present algorithms for the cryptographic relational algebra in Section 4.

3. A TWO-PHASE ENCRYPTION/DECRYPTION SCHEME

In this section, we present a two-phase encryption algorithm for enhancing security in data base systems. Our scheme is based on both the one-way function and the concept of subkeys. Our scheme

does not require that the length of raw data objects be extended for security considerations.

3.1 The Subkeys Scheme

A data base system with subkeys has the following advantages over conventional systems (Davida et al., 1981; Lin et al., 1990; Lin, 1991). First, each encrypted record is a single encrypted value that is a function of all fields, so the system is record oriented. Obviously, a small change in the encrypted value will cause a significant change in the decrypted value. Therefore, unauthorized modification of data can be prevented. Second, the system's properties can withstand pattern-matching attacks. Third, the possibility of substitution attacks is eliminated because the system encrypts all fields together. Finally, a user can read only some of the field data objects, depending on the reading field subkey he or she has. Not all fields need be available to everyone.

A data base E/D scheme with subkeys was first proposed by Davida et al. in 1981. Their scheme was based on the CRT (Niven and Zuckerman, 1966). Let C_i be the ciphertext of an encrypted record, let d_j be the reading subkey for field j, and let there be n fields in each record and m records in a relation. The encryption procedure is done by forming

$$C_i = \sum_{j=1}^{n} e_j x_{ij} \bmod D, \quad \text{for } i = 1,2,\cdots,m \tag{1}$$

where $D = \Pi_{j=1}^{n} d_j$; x_{ij} is the value of field j of record i, $x_{ij} \le d_j$, $e_j = (D/d_j)b_j$ is the writing subkey for field j, and b_j is the multiplicative inverse of D/d_j with moduli d_j.

The decryption can be done as follows:

$$x_{ij} = C_i \bmod d_j, \quad j = 1,\cdots,n \tag{2}$$

By use of the CRT, the subkey scheme has the following merit: The raw field data can be easily recovered within only one operation. That is, the field data are obtained from C_i by merely finding the remainder of C_i mod d_j. The CRT has been used widely in security control, such as in access control schemes (Chang et al., 1986), secure broadcasting schemes (Chiou and Chen, 1989), identification and authentication schemes (Chang and Wu, 1991), and public-key cryptosystems (Lu and Lee, 1979). Unfortunately, some schemes that use the CRT alone are not truly secure (Davida et al., 1981, Chang and Laih, 1992; Lee, 1979). Davida et al. (1981) noted that a subkey scheme based on the CRT cannot withstand known plaintext attacks. They improved the scheme by adding a random redundancy value r_{ij} to each field before enciphering. The encryption procedure (equation 1) is thus replaced by the following equation:

$$C_i = \sum_{j=1}^{n} e_j(r_{ij} \| x_{ij}) \bmod D, \quad \text{for } i = 1,2,\cdots,m \tag{3}$$

where $\|$ indicates concatenation. On the other hand, the decryption procedure (equation 2) is replaced by the following equation:

$$r_{ij} \| x_{ij} = C_i \bmod d_j, \quad j = 1,\cdots,n \tag{4}$$

By discarding the random bit r_{ij}, one can obtain the jth field data x_{ij} of record i.

In our scheme, we use the CRT in the subkey scheme and a one-way function as a field encryption scheme to replace the redundancy bits found in Davida et al. (1981) while maintaining security.

3.2 One-Way Function

To describe the two-phase encryption algorithm in the following section, we first propose our solution for eliminating the redundancy bits found in Davida et al. (1981). The method is based on the well-known idea of one-way functions. This is a family of functions $f: x \to y$ with the following properties (Merkle, 1990; Naor and Yung, 1989; Rompel, 1990):

1. The functions f are easy to compute, and it is also easy to pick a member of the function f at random.
2. The functions are computationally difficult to invert. This means it is computationally infeasible, given a string x, to compute another string $x \neq x'$ satisfying $f(x) = f(x')$ for a randomly chosen f.

The practical importance of such functions has been known for some time, and researchers have used them in a number of schemes. For example, they have been applied for safeguarding cryptographic keys (Gudes, 1980), access control in a hierarchy (Akl and Taylor, 1983; Sandhu, 1988), key management in a group-oriented scheme (Denning et al., 1981), a user authentication scheme (Chang and Wu, 1991), and other fields (Ingemarsson and Wong, 1981). Merkle (1990) showed that a good cryptosystem can be used to implement a one-way function. A commonly used approach is to encrypt some fixed constant c by using x as the key, i.e., $f(x) = E_x(c)$. Computing the inverse of $f(x)$ then amounts to computing the key x given that c encrypts as $f(x)$.

It is generally accepted that one-way functions are a major tool in cryptography. A commercial product, DES (National Bureau of Standards, 1977; Smid and Branstad, 1988), is the best known and most widely

used encryption function. Generating one-way functions is secure if DES is random (Merkle, 1990). Public-key cryptographic systems (i.e., the RSA scheme) are also based on one-way functions.

In our system, the raw data of fields should be encrypted by the DES chip before being sent to the subkey scheme. The run time of the DES algorithm should be as little as possible. It is well known that DES has been implemented both in software and hardware. Hardware implementations achieve encryption rates of several million bits per second (Denning, 1982). Thus, use of the DES chip to encrypt/decrypt field data affects system performance only slightly.

3.3 The Two-Phase Encryption Algorithm

We now describe the two-phase encryption algorithm. To illustrate the scheme, we assume that there are n fields in each record of a data base. Let $m_1, m_2, \cdots, m_n$ be the n raw data of fields of a record.

Phase E1. Encrypt m_j, for $j = 1, \cdots, n$ with a symmetric cryptosystem, such as DES. Let f be the encryption algorithm and k_j be a secret key of field j. This encryption is done as $f_{k_j}(m_j)$.

Phase E2. Encrypt $f_{k_j}(m_j)$ with writing subkeys $e_1, e_2, \cdots, e_n$. This encryption is done as

$$C = E((f_{k_1}(m_1), e_1), (f_{k_2}(m_2), e_2), \cdots, (f_{k_n}(m_n), e_n)) \quad (5)$$

where E is an encryption algorithm, e_j is a writing key for field j, and C is the encrypted data of a record. With the CRT, the encryption procedure is the following:

$$C = \sum_{j=1}^{n} e_j f_{k_j}(m_j) \bmod D \quad (6)$$

where $D = \Pi_{j=1}^{n} d_j$, $e_j = (D/d_j)b_j$ is the writing key for field j, and b_j is the multiplication inverse of D/d_j with module d_j. A diagram of the encryption scheme is shown in Figure 2.

The decryption procedure is the reverse of the encryption procedure.

Phase D1. Decrypt ciphertext C with reading subkeys $d_1, d_2, \cdots, d_n$. The decryption is done as

$$f_{k_j}(m_j) = S(C, d_j) \quad (7)$$

where S is a decryption algorithm based on the CRT and d_j is a reading key for field j. The decryption procedure is as follows:

$$f_{k_j}(m_j) = C \bmod d_j. \quad (8)$$

Phase D2. Decrypt $f_{k_j}(m_j) = m'_j$ with the secret key k_j as follows:

$$m_j = f_{k_j}^{-1}(m'_j), \quad (9)$$

A diagram of the decryption scheme is shown in Figure 3.

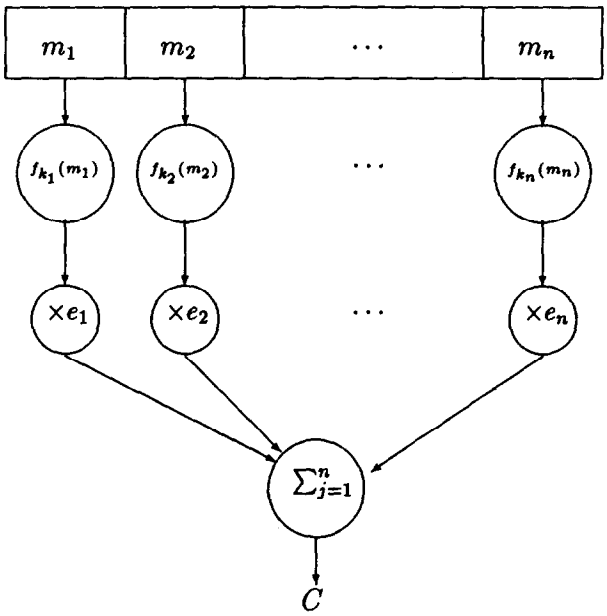

Figure 2. Encryption procedure.

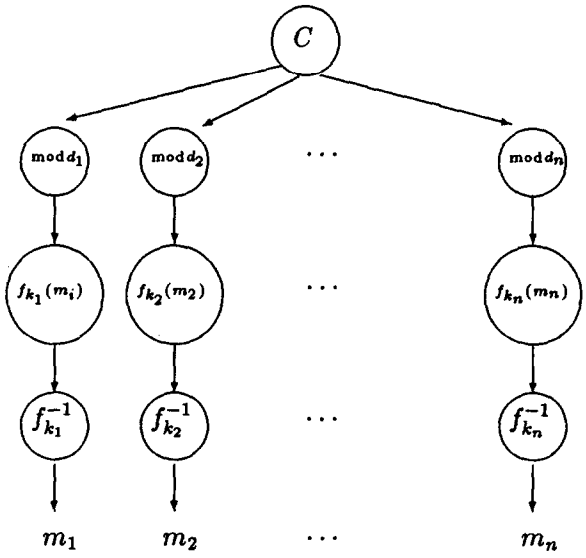

Figure 3. Decryption procedure.

3.4 Cryptanalysis

Use of the CRT in a subkey encryption scheme is not an effective scheme for security (Davida et al., 1981; Wells, 1981); the scheme has the following weaknesses:

It cannot withstand known plaintext attacks. Let C and C' be the ciphertexts of two different records R and R', respectively. If m_j and m'_j are the raw data of field j in R and R', respectively, and both are known to a cryptanalyst, then from equation 2 we have

$$\begin{aligned} C \bmod d_j &= m_j \\ C' \bmod d_j &= m'_j \end{aligned} \tag{10}$$

implying

$$\begin{aligned} C - m_j &= a_1 d_j \\ C' - m'_j &= a_2 d_j \end{aligned}$$

The subkey d_j thus can be derived from the above two equations by use of the greatest common divisor (gcd).

The following strategy can also be used to attack the scheme. Let C_i be the i encrypted record and m_{ij} be the field j raw data of record i. Thus, in the system exists an integer a_1 such that

$$C_i = a_1 d_j + m_{ij} \tag{11}$$

Assume that a field other than j is updated. Then

$$C'_i = a_2 d_j + m_{ij} \tag{12}$$

because m_{ij} is not changed. Then

$$C_i - C_i' = C''_i = (a_1 - a_2) d_j \tag{13}$$

If a similar operation is performed on another encrypted record C'', then

$$C_h - C'_h = C''_h = (a_3 - a_4) d_j \tag{14}$$

The subkey d_j can then be computed by finding the $\gcd(C''_i, C''_h)$.

The scheme cannot withstand collusion attacks. All users can, together, compute the writing key e_j, which is known only by the system, if they have all of the reading keys d_j.

To eliminate these weaknesses, Davida et al. (1981) concatenate a random redundancy value r_{ij} in each field (the length of the redundancy value r_{ij} is at least 32 bits, which leads to better security.). Although this improved scheme can prevent known plaintext attacks, all fields with different values of r_{ij} in the encrypted record must be recomputed whenever any field is updated. Also, extra dummy fields are needed to prevent colluding users for computing the writing key e_j. Lin and coworkers (1990, 1991) generalized the CRT. Although their method does not require extra redundancy bits in each field, as does that of Davida et al. (1981), it requires an extra secret key for each record. It also requires that an extra dummy field be recomputed and added to the system to eliminate the second and third weaknesses described above.

Now let us see whether a known plaintext attack is possible in our scheme. Let C and C' be the ciphertexts of two different records R and R', respectively. If m_j and m'_j are the fields in R and R', respectively, and both are known to a cryptanalyst, then from equation 8, we have

$$\begin{aligned} C \bmod d_j &= f_{k_j}(m_j) \\ C' \bmod d_j &= f_{k_j}(m'_j) \end{aligned} \tag{15}$$

implying

$$\begin{aligned} C - f_{k_j}(m_j) &= a_1 d_j \\ C' - f_{k_j}(m'_j) &= a_2 d_j \end{aligned}$$

These simultaneous equations have three unknown variables, $f_{k_j}(m_j)$, $f_{k_j}(m'_j)$, and d_j. Hence, there are infinite possible solutions for d_j. In general, if t corresponding fields of t records are known, then there are $t + 1$ unknown variables to be determined with t simultaneous equations. Hence, it is much more difficult to mount a known plaintext attack against our scheme than against the scheme in of Davida et al. (1981). Because our scheme is based on the CRT, the second weakness still remains. However, the security of our scheme depends on the one-way function in addition to the subkey scheme. Illegal users cannot read the raw data of a tuple unless they know both the reading subkey and the secret key of the symmetric cryptosystem. Thus, security is guaranteed in our scheme. Because the writing key for field j, e_j, is equal to $(D/d_j)b_j$, e_j can be obtained if we know all the d_js. Therefore, it seems unavoidable that we need extra dummy fields in our scheme to prevent collision attacks.

3.5 Computational Complexity

In this section, we examine the complexity of enciphering and deciphering each field. Assume that each record contains n fields and the average number of bits of each field is q. The computation time

needed for each phase discussed in Section 3.3 is as follows.

Phase E1. If DES is used as the symmetric cryptosystem, it partitions the data text into pieces of 64 bits each. This phase requires

$$t_{e1} = n * \lceil q/64 \rceil DES(64)$$

where $DES(64)$ is the time required to encipher 64 bits of text using the DES device. To compute $DES(64)$, 16 rounds of one table look-up and one XOR operation each are required:

$$DES(64) = 16(t_l + t_{xor})$$

where t_l is the time cost of a table look-up and t_{xor} is the time cost of an XOR operation. The total processing time of phase E1 is

$$t_{e1} = n * \lceil q/4 \rceil (t_l + t_{xor})$$

Phase E2. Encryption equation 6 requires a total of $2n$ multiplications, $(n - 1)$ additions, n divisions, and one module operation. Let $t_{op}(p, q)$ denote the time cost of an *op* operation (i.e., multiplication, division, addition, or module) with two bits p and q.

$$\begin{aligned} t_{e2} &= 2nt_{multiplication}(nq, q) + (n-1)t_{addition}(nq, nq) \\ &\quad + nt_{division}(nq, q) + t_{module}(nq, nq) \\ &= 2n^2 t_{multiplication}(q, q) + n(n-1)t_{addition}(q, q) \\ &\quad + nt_{division}(nq, q) + t_{module}(nq, nq) \end{aligned}$$

The total processing time of the two-phase encryption procedure is

$$t_{encryption} = t_{e1} + t_{e2}.$$

Phase D1. Decryption equation 8 requires only one module operation:

$$t_{d1} = t_{module}(nq, q)$$

Phase D2. the computation time required is the same as in phase E1:

$$t_{d2} = n * \lceil q/4 \rceil (t_l + t_{xor})$$

The total processing time of the two-phase decryption procedure is

$$t_{decryption} = t_{d1} + t_{d2}$$

Some efficient implementations of the CRT have been developed (Knuth, 1980; Vu, 1985; Dirr and Taylor, 1985). Dirr and Taylor (1985) designed a fast and efficient hardware implementation of the CRT in residue arithmetic. Their method incurs a time cost of $70\lceil \log_2 L \rceil$ ns for computing the equation $C = m_i \bmod d_i$, for $i = 1, 2, \cdots, L$. It only needs 0.35 ms to encipher a data base with 32 fields and 1,000 records. Thus, our subkey scheme is practical to implement.

Table 1. Comparison of storage space

Scheme	Number of Keys	Space for Raw Data	Space for Encrypted Data
Davida et al., 1981	$2n$	$mn(32 + q)$	
Lin and coworkers, 1990, 1991	$m + 2n$	mnq	mnq $m(n + 1)q$
Our scheme	$3n$	mnq	mnq

3.6 Storage Space

We assume that there are m records in a data base, n fields in each record, and an average of q bits in each field. Our scheme needs n reading field subkeys, n writing field subkeys, and n secret keys for the DES scheme. The total number of keys is $3n$. Both the raw data and the encrypted data are mnq bits. Davida et al.'s (1981) scheme needs rmn extra redundancy bits, where r is suggested to be 32 bits (or longer for greater security). Lin and coworkers' (1990, 1991) scheme needs a large number of secret keys, m, for each record. In general, the number of records is much larger than the number than the number of fields (i.e., $m \gg n$) in the data base. Table 1 compares the storage space needed by the three schemes. Lin and coworkers' scheme requires expanding encrypted field data in each record of the data base.

4. CRYPTOGRAPHIC RELATIONAL ALGEBRA

In this section, we show how to perform the relational operations in our scheme. Codd (1972) defined a very specific set of eight operations: restrict, project, Cartesian product, union, intersection, difference, natural join, and division. Basically, only the first five primitive operations are needed; the other operations can be derived from these five (Dale, 1990). For example, natural join is a projection of a restriction of a product, intersection is a difference twice, and division is the difference of a product of a difference. Thus, we treat only the five primitive operations.

Because our scheme is a so-called record-oriented (tuple-oriented) subkey scheme, it is easy to see that the restrict, union, intersection, and difference operations are the same as in a traditional data base. By use of the CRT (Denning, 1982; Davida and Yeh, 1982), we develop two algorithms for projection and production, as shown in Tables 2 and 3, respectively.

Table 2. Algorithm for Projection

Input:	Ciphertext $C_i, i = 1,\cdots, m$, where m is the number of records in the data base. Read field subkeys $d_j, j = 1,\cdots, n$, where n is the number of fields (attributes) in the data base. Input read subkeys $d'_j, j = 1,\cdots, k$, where $k < n$ (i.e., $\{d'_j : j = 1,\cdots, k\} \subset \{d_j : j = 1,\cdots, n\}$).
Output:	New encrypted record data $C'_j, j = 1,\cdots, m$
1.	for $j = 1,\cdots, k$ do
2.	begin
3.	if $d'_j \neq d_i$ for all $i = 1,\cdots, n$
4.	then return (Failure);
5.	end
6.	Computes $D' = \Pi_{j=1}^{k} d'_j$;
7.	for $i = 1,\cdots, m$ do
8.	$C'_i = C_i \bmod D'$;

Step 8 in Table 2, $C'_i = C_i \bmod D'$, can be proved to be correct as follows:

$$C'_i \bmod d'_j$$

$$= (C_i \bmod D') \bmod d'_j$$

$$= C_i \bmod d'_j$$

$$= m_{ij}$$

5. KEY MANAGEMENT

In the subkey scheme, the fields that a user can read depend on the reading field subkeys he or she holds, as shown in Figure 4. In the figure, each user owns many reading subkeys. Management of the number of subkeys for users is a difficult key management problem (Denning et al., 1981; Chick and Tavares, 1990). In this section, we propose two simple but efficient methods of handling this problem in our scheme.

Method 1

This method uses the CRT. The master key for user i is generated by the following steps:

1. Assign each field a public prime number p_j, for $j = 1, 2,\cdots, n$.
2. Compute the secret master key MK_i by the CRT for user i

$$MK_i = d_j \bmod p_j, \quad \text{for some } j, 1 \leq j \leq (n+1) \tag{16}$$

where d_j is possessed by user j and d_{n+1} and p_{n+1} are a random secret key and a prime number of a dummy field used to prevent users from colluding to disclose the secret master key of user i.

3. When user i wants to read field j, he or she computes the following equation with his or her secret master key MK_i and the public parameter p_j of field j:

$$d_j = MK_i \bmod p_j \tag{17}$$

Table 3. Algorithm for Cartesian Production

Input:	Ciphertext $C'_i, i = 1,\cdots, m$, where m is the number of records in the data base. Ciphertext $C''_i, i = 1,\cdots, m''$ in a relation table T'. Read field subkeys $d'_j, j = 1,\cdots, n'$, where n' is the number of fields (attributes) in the data base. Read field subkeys $d''_j, j = 1,\cdots, n''$ in a relation table T'', where $d''_j \neq d'_i$ for all i and j.
Output:	New encrypted record data $C_j, j = 1,\cdots, m$
1.	Computer $D_1 = \Pi_{j=1}^{n'} d'_j$
2.	Computer $D_2 = \Pi_{j=1}^{n''} d''_j$
	/* Computing the ciphertext by CRT */
3.	Computer $D = D_1 \times D_2$
4.	for $j = 1, 2$ do
5.	begin
6.	Compute $G_j = D/D_j$;
7.	Find G'_j such that $G_j G_j' \bmod D_j = 1$;
8.	end;
	/* Computes new ciphertext record */
9.	for $i = 1,\cdots, mm'$ do
10.	$K_i \leftarrow (C'_i G_j G_j' + C''_i G_j G'_j) \bmod D$;

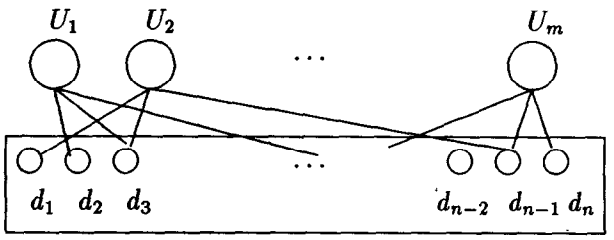

Figure 4. An example of that each user owns many subkeys.

Method 2

This method uses Newton's interpolation method. A secret interpolating polynomial is constructed through the following steps:

1. Choose a large prime number P.
2. Assign each field a public identification number x_j.
3. Construct the secret polynomial $F_i(X)$ over the Galois field $GF(P)$ by interpolating on points (x_j, d_j) and (x_{n+1}, d_{n+1}) for some $j, 1 \le j \le (n+1)$, where x_{n+1} and d_{n+1} are random numbers used to withstand collusion attacks by other users. The interpolating polynomial is computed as follows:

$$F_i(x) = \sum_j \left(f(x_0, x_1, \cdots, x_j) \prod_{i=0}^{j-1} (x - x_i) \right) \bmod P \tag{18}$$

Each coefficient $f(x_0, x_1, \cdots, x_j)$ in this formula is found by computing the divided differences (Jain et al., 1985) in the following equations:

$$f(x_0, x_1, \cdots, x_j) = (f(x_1, x_2, \cdots, x_j) - f(x_0, x_1, \cdots, x_{j-1}))/(x_j - x_0)$$

$$\text{and } f(x_j) = d_j, \text{for some } j, 1 \le j \le (n+1).$$

4. When user i wants to read field j, he or she computes the following equation with his or her secret polynomial $F_i(x)$ and the public parameter x_j of field j:

$$d_j = F_i(x_j) \bmod P \tag{19}$$

6. CONCLUSIONS

We have proposed a two-phase encryption scheme for enhancing data base security. The characteristics of our two-phase enciphering scheme are as follows:

1. The security of our scheme depends on the one-way function instead of depending fully on the subkey scheme. Thus, security is guaranteed in our scheme.
2. Our scheme does not require a large number of redundancy bits, as does the scheme of Davida et al. (1981), to withstand known plaintext attacks.
3. The number of subkeys and secret keys is equal to the number of fields in our scheme; in contrast, Lin and coworkers' (1990, 1991) scheme requires as many secret keys as there are records. The number of records is much larger than the number of fields in most data base systems.

We have developed two algorithms for cryptographic relational algebra. We have also introduced two simple methods of solving the key management problem in the subkey scheme.

ACKNOWLEDGMENT

We thank Dr. Che-Fn Yu for contributions to this work.

This research was funded by grants NSC82-0408-E-009-161 and TL-NSC-82-5206 from the National Science Council and Telecommunication Laboratories, Taiwan, R.O.C., respectively.

REFERENCES

Akl, S. G., and Taylor, P. D., Cryptographic Solution to a Problem of Access Control in a Hierarchy, *ACM Trans. Comp. Syst.* 1, 239–248 (1983).

Babad, Y. M., and Hoffer, J. A., Data Element Security and Its Effects on File Segmentation, *IEEE Trans. Software Eng.* SE-6, 402–410 (1980).

Bayer, R., and Metzger, J. K., On the Encipherment of Search Trees and Random Access Files, *ACM Trans. Database Syst.* 1, 37–52 (1976).

Chang, C. C., On the Design of a Key-Lock-Pair Mechanism in Information Protection Systems, *BIT* 26, 410–417 (1986).

Chang, C. C., and Wu, T. C., Remote Password Authentication with Smart Cards, *IEEE Proc.* 138, 165–168 (1991).

Chang, C. C., and Laih, C. S., Remote Passwork Authentication with Smart Cards, *IEEE Proceedings* 139, 372 (1992).

Chick, G. C., and Tavares, S. E., Flexible access control with master keys, in *Advances in Cryptology, CRYPTO '89,* 1990, pp. 316–322.

Chiou, G.-H., and Chen, W.-T., Secure Broadcasting Using the Secure Lock, *IEEE Trans. Software Eng.* 15, 929–934 (1989).

Codd, E. F., *Relational Completeness of Data Base Sublanguages,* Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

Conway, R. W., Maxwell, W. L., and Morgan, H. L., On the Implementation of Security Measures in Information Systems, *Commun. ACM* 15, 211–220 (1972).

Coper, J. A., *Computer & Communication Security: Strategies for the 1990s,* McGraw-Hill, New York, 1989.

Date, C. J., *An Introduction to Database Systems,* vol. 1, Addison-Wesley, 1990.

Davida, G. I., and Yeh, Y., Cryptographic relational algebra, in *Proceedings of the IEEE Symposium on Security and Privacy,* 1982, pp. 111–116.

Davida, G. I., Wells, D. L., and Kam, J. B., A Database Encryption System with Subkeys, *ACM Trans. Database Syst.* 6, 312–328 (1981).

Denning, D. E. R., *Cryptography and Data Security,* Addison-Wesley, 1982.

Denning, D. E. R., Meijer, H., and Schneider, F. B., More on Master Keys for Group Sharing, *Info. Proc. Lett.* 13, 125–26 (1981).

Dirr, W., Jr., and Taylor, F. J., On Implementing the CRT in Residue Arithmetic, *J. Comp. Math.* 17, 155–163 (1985).

Eriksson, R., and Beckman, K., Protection of data-bases using file encryption, in *Proceedings of the First Security Conference, IFIP / Sec '83,* 1983, pp. 217–221.

Feigenbaum, J., Liberman, M. Y., and Wright, R. N., Cryptographic protection of databases and software, in *DIMACS Series in Discrete Math and Theoretical Computer Science,* vol. 2, 1991, pp. 161–172.

Fernandez, E. B., Summers, R. C., and Wood, C., *Database Security and Integrity,* Addison-Wesley, 1980.

Graham, G. S., and Denning, P. J., Protection—Principles and practice, in *Proceedings Spring Joint Computer Conference,* Vol. 40, AFIPS, 1972, pp. 417–429.

Gudes, E., The Design of a Cryptography Based Secure File System, *IEEE Trans. Software Eng.* SE-6, 411–420 (1980).

Hwang, M. S., and Yang, W. P., A New Dynamic Access Control Scheme Based on Subject-Object-List, Data & Knowledge Engineering 14, 45–56 (1994).

Ingemarsson, I., and Wong, C. K., A User Authentication Scheme for Shared Data Based on Trap-Door One-Way Functions, *Info. Proc. Lett.* 12, 63–67 (1981).

Jain, M. K., Lyengar, S. R. K., and Jain, R. K., *Numerical Methods for Scientific and Engineering Computation,* Wiley Eastern Limited, New Delhi, India, 1985.

Knuth, D. E., *The Art of Computer Programming, Vol. 2: Seminumerical Algorithm,* Addison-Wesley, 1980.

Lee, L. N., Note on Cryptosystems, *COMSAT Tech. Rev.* 9, 717–721 (1979).

Lin, C. S., An Application of an Encryption Algorithm to Database Security, Chap. 3, Ph.D. Thesis, National Tsing Hua University, 1991.

Lin, C. H., Chang, C. C., and Lee, C. T., A record-oriented cryptosystem for database sharing, in *International Computer Symposium,* 1990, pp. 328–329.

Lu, S. C., and Lee, L. N., A Simple and Effective Public-Key Cryptosystem, *COMSAT Tech. Rev.* 9, 15–23 (1979).

Merkle, R. C., One-way hash functions and DES, in *Advances in Cryptology, CRYPTO '89,* 1990, pp. 428–446.

Naor, M., and Yung, M., Universal one-way hash functions and their cryptographic applications, in *Proceedings of the 21st STOC,* 1989, pp. 33–43.

National Bureau of Standards, *Data Encryption Standard,* FIPS, NBS, 1977.

Niven, I., and Zuckerman, H., *Introduction to the Theory of Numbers,* Wilcy, New York, 1966.

Pfleeger, C. P., *Security in Computing,* Prentice-Hall, 1989.

Rivest, R. L., Shamir, A., and Adleman, L., A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Commun. ACM* 21, 120–126 (1978).

Rompel, J., One-way functions are necessary and sufficient for secure signatures, in *Proceedings of the 22nd STOC,*: 1990, pp. 387–394.

Sandhu, R. S., Cryptographic Implementation of a Tree Hierarchy for Access Control, *Info. Proc. Lett.* 27, 95–98 (1988).

Smid, M. E., and Branstad, D. K., The Data Encryption Standard: Past and Future, *Proc. IEEE* 76, 550–559 (1988).

Ullman, J. D., *Principles of Database and Knowledge-Base Systems,* vol. 1, Computer Science, 1988.

Vu, T. V., Efficient Implementations of the Chinese Remainder Theorem for Sign Detection and Residue Decoding, *IEEE Trans. Comp.* C-34, 646–651 (1985).

Wagner, N. R., Putter, P. S., and Cain, M. R., Encrypted database design: Specialized approaches, in *Proceedings of the IEEE Symposium on Security and Privacy,* 1986, pp. 148–153.

Wells, D. L., A Short Note on the Dangers of Loading CRT Subkeys, Technical Report TTTR-CSE-8106, Department of Computer Science and Engineering, Southern Methodist University, 1981.