A Two-Phase Encryption Scheme for Enhancing Database Security *

Min-Shiang Hwang ‡
Email: m24@twnmoctl.bitnet

Wei-Pang Yang †,0
Email: wpyang@twnctu01.bitnet

Department of Computer and Information Science †
National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.

Directorate General of Telecommunication Laboratories ‡
Ministry of Transportation and Communications
P.O. Box 71, Chung-Li, Taiwan 320, R.O.C.

November 13, 2004

In this paper, we propose a two-phase encryption algorithm for database 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 database 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 database are the following [1]: (1) shared access; (2) minimal redundancy; (3) data consistency; (4) data integrity, so that data values are protected against accidental or malicious unauthorized changes; and (5) controlled access, so that only authorized users are allowed to access data values. A database management system (DBMS) with security facility is designed to provide all of these advantages efficiently.

In general, there are four methods of enforcing database security [2]: First, physical security, such as storage medium safekeeping and fire protection [3]; second, operating system security, such as the use of an access control matrix, capability-list, and accessor-list [4, 5, 6]; third, DBMS security, such as protection mechanisms and query modification [7]; and fourth, data encryp-

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no. NSC82-0408-E-009-161, and by the Telecommunication Laboratories, Taiwan, R.O.C., under contract no. TL-NSC-82-5206.

†

‡

†

†

‡

[0] Responsible for correspondence.

tion, such as the data encryption standard (DES) [8, 9] and RSA scheme [10]. The first three methods, however, are not totally satisfactory solutions to the database security problem, for the following four reasons: First, it is hard to control the disclosure of raw data, because the raw data exists in readable form inside a database [11]. Second, it is invalid to operating system security control and DBMS security control the disclosure of sensitive data, because the sensitive data must frequently be backed up in storage media in case of system failure or disk crash. Third, it is hard to control the disclosure of confidential data in a distributed database 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 [12, 13]. A practical solution to the above problems is to using encryption methods to enforce database security [11, 14, 15, 16, 17, 18, 19].

An encryption database security can solve the above problems in the following manner: (1) Data are encrypted into ciphertext, which only can be decrypted with the proper decryption key, thus eliminating the problem of data disclosure. (2) Since an intruder cannot change the ciphertext without knowing the encryption keys, thus the data authenticity problem is also resolved.

Database security methods based on encryption include database encryption systems with a single key [18] and database encryption systems with subkeys [11]. The first type of method needs a trusted centralized access control scheme with which to control all access to data stored in the database system (DBS). All encryption and decryption is executed by the trusted access control scheme with privacy key. In the second type of method, however, decryption is executed by users themselves with their own subkeys.

The first database encryption/decryption system with subkeys was proposed by Davida, Wells, and Kam [11]. 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.

In order to eliminate the above drawback. Lin et al. modified the method in [12, 13]. Basically, they generalized the Chinese remainder theorem. 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 Lin et al.'s 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 databases.

In this paper, we propose a two-phase encryption scheme for enhancing database security. Our scheme does not require extending the raw data, nor is an extra privacy key needed for each record. The paper is organized as follows. In Section 2, we introduce an architecture for a secure database 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 database system in order 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 of solving the key management problem in the subkey scheme. Section 6 is the conclusion of the paper.

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 decryption module, DBMS, locker module, and encrypted database. A user makes queries

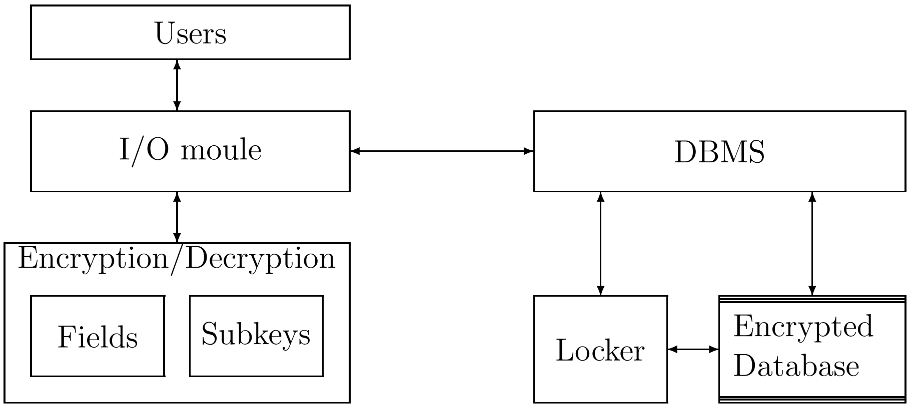

Figure 1: The architecture of the secured database system.

via the I/O module to access data objects in the database. The I/O module is a general purpose input and output device (i.e., a terminal) except 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 [8, 9]) or an asymmetric cryptosystem (RSA [10]) to the field E/D module. We shall 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 using the subkeys. The concept of subkeys was first proposed by Davida, Wells, and Kam [11]. The details of the subkey scheme will be discussed in Section 3.1.

The DBMS module is a general purpose DBMS used for managing database systems [20]. Another function of the DBMS is to control the operation of the encrypted database using the locker module. The two responsibilities of the locker module are to encrypt data item of fields within a record with the subkeys of fields and to operate the encrypted database using relational algebra. We will 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 database 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 database system with subkeys has the following advantages over conventional systems [11, 12, 13]. First, each encrypted record is a single encrypted value which 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 has. Not all fields need be available to everyone.

A database encryption/decryption scheme with subkeys was first proposed by Davida, Well, and Kam [11] in 1981. Their scheme was based on the Chinese remainder theorem (CRT) [21]. 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

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

where $D = \prod_{j=1}^{n} d_j$; x_{ij} is the value of field j of record i; $x_{ij} \leq d_j$; $e_j = (D/d_j)b_j$ 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:

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

Using 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 $C_i \bmod d_j$. The CRT has been used widely in security control, such as in access control schemes [22], in secure broadcasting schemes [23], in identification and authentication schemes [24], and in public-key cryptosystems [25]. Unfortunately, some schemes that use the CRT alone are not truly secure [11, 26, 27]. Davida, Wells, and Kam noted that a subkey scheme based on the CRT cannot withstand known-plaintext attacks [11]. 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:

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

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 use a one-way function as a field encryption scheme to replace the redundancy bits in [11] while maintaining security.

3.2 ONE-WAY FUNCTION

In order to describe the two-phase encryption algorithm in the following section, we first propose our solution for eliminating the redundancy bits in [11]. The method is based on the well-known idea of one-way functions. This is a family of functions $f : x \rightarrow y$ with the following properties [28, 29, 30]:

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 [18]; for access control in a hierarchy [31, 32]; for key management in a group-oriented scheme [33]; for a user authentication scheme [24]; and for other fields [34]. Merkle [28] 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 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 [8, 9], is the best known and most widely used encryption function. Generating one-way functions is secure if DES is random [28]. 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 in hardware. Hardware implementations achieve encryption rates of several million bits per second [35]. Thus, using the DES chip to encrypt/decrypt field data will affect system performance only slightly.

3.3 THE TWO-PHASE ENCRYPTION ALGORITHM

Now we describe the two-phase encryption algorithm. To illustrate the scheme, we assume that there are n fields in each record of a database. 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

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

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:

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

where $D = \prod_{j=1}^{n} d_j$, $e_j = (D/d_j)b_j$ is the writing key for field j, and b_j is the multiplication inverse of D/d_j with moduli 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

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

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

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

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

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

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

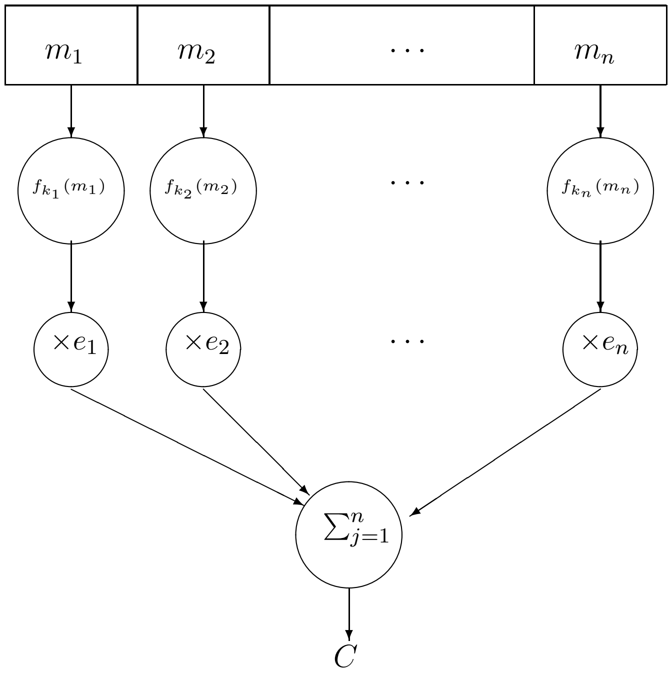

Figure 2: Encryption procedure.

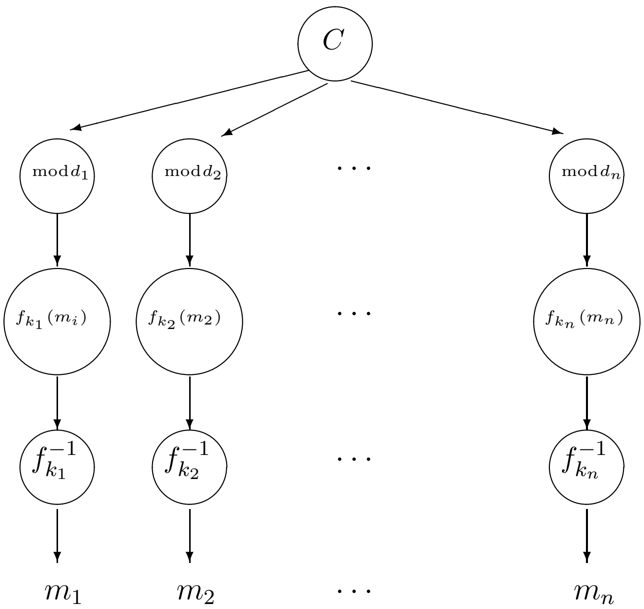

Figure 3: Decryption procedure.

3.4 CRYPTANALYSIS

Using the CRT in a subkey encryption scheme is not an effective scheme for security [11, 36]; the scheme has the following weaknesses:

1. 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 using the greatest common divisor (GCD).

2. 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, the system exists a 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

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

since 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 $\gcd(C''_i, C''_h)$.

3. 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.

In order to eliminate the above weaknesses, Davida, Wells, and Kam [11] 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 from computing the writing key e_j. Lin, Chang, and Lee [12, 13] generalized the CRT. Although their method does not require extra redundancy bits in each field, as does that in [11], 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 the 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

$$C \bmod d_j \quad = \quad f_{k_j}(m_j) \tag{15}$$

$$C' \bmod d_j = f_{k_j}(m'_j)$$

implying

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

The above 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, there are $t+1$ unknown variables to be determined with t simultaneous equations. Hence it will be much more difficult to mount a known-plaintext attack against our scheme than against the scheme in [11]. Since 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. Since 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_j's. Therefore, it seems unavoidable that we need extra dummy fields in our scheme to prevent collusion 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 number of bits of each field is q on the average. The computation time needed for each phase in Section 3.3 is as follows.

Phase E 1: 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), sixteen rounds of one table-lookup and one XOR operation each are required:

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

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

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

<u>Phase E 2:</u> 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

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

<u>Phase D 1:</u> Decryption Equation (8) requires only one module operation:

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

Phase D 2: The computation time required is the same as that of Phase E1:

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

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

$$t_{decryption} \quad = \quad t_{d1} + t_{d2}.$$

Some efficient implementations of the CRT have been developed [37, 38, 39]. Dirr and Taylor [39] have 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 database with 32 fields and 1000 records. Thus, our subkey scheme is practical to implement.

3.6 STORAGE SPACE

We assume that there are m records in a database, 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 scheme [11] needs rmn extra redundancy bits, where r is suggested to be 32 bits (or longer for greater security). Lin et al.'s scheme [12, 13] needs a large number of secret keys m for each record. In general, the number of records is much larger than the number of fields (i.e., $m \gg n$) in the database. Table 1 compares the storage space needed by the three schemes. Lin et al.'s scheme requires expanding an encrypted field data in each record of the database.

Table 1: Comparison of storage space.

Scheme	Number of keys	Space for raw data	Space for encrypted data
Davida et al.	$2n$	$mn(32+q)$	mnq
Lin et al.	$m+2n$	mnq	$m(n+1)q$
Our scheme	$3n$	mnq	mnq

4 CRYPTOGRAPHIC RELATIONAL ALGEBRA

In this section, we show how to perform the relational operations in our scheme. Codd [40] 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 [20]. 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 shall treat only the five primitive operations.

Since our scheme is a so-called record-oriented (tuple-oriented) subkey scheme, it is easy to see that the restrict, union, intersection, and difference are the same as in a traditional database. By the CRT [35, 41], we develop two algorithms for projection and production, as shown in Table 2 and Table 3, respectively. Step 8 in Table 2, $C_i' = C_i \bmod D'$, can be proved to be correct as follows:

$$\begin{aligned}
& C_i' \bmod d_j', \\
= \; & (C_i \bmod D') \bmod d_j', \\
= \; & C_i \bmod d_j',
\end{aligned}$$

Table 2: Algorithm for projection.

Input:	Ciphertext C_i, $i = 1, \cdots, m$, where m is the number of records in the database.
	Read field subkeys d_j, $j = 1, \cdots, n$, where n is the number of fields (attributes) in the database.
	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 $D' = \prod_{j=1}^{k} d'_j$;
7.	for $i = 1, \cdots, m$ do
8.	$C'_i = C_i \bmod D'$;

$$= \quad m_{ij}.$$

5 KEY MANAGEMENT

In the subkey scheme, the fields that a user can read depend on the reading field subkeys he holds, as shown in Figure 4. In the figure, each user may be authorized to read many of the fields, he then own many of reading subkeys. Management of the number of subkeys for users is a difficult key management problem [33, 42]. In this section, we propose two simple but efficient methods of handling this problem in our scheme.

Method 1: Using the Chinese remainder theorem. The master key for user i is generated by the following steps:

1. Assign each field a public prime number p_j, for $j = 1, 2, \cdots, n$.

Table 3: Algorithm for Cartesian production.

Input:	Ciphertext C'_i, $i = 1, \cdots, m$, where m is the number of records in the database.
	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 database.
	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 $D_1 = \prod_{j=1}^{n'} d'_j$
2.	Computer $D_2 = \prod_{j=1}^{n''} d''_j$
	/∗ 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.	$K_i \longleftarrow (C'_i G_j G'_j + C''_i G_j G'_j) \bmod D$;

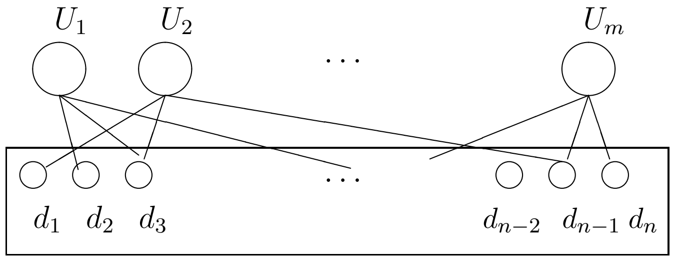

Figure 4: An example of that each user has his own field subkeys.

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 computes the following equation with his secret master key MK_i and the public parameter p_j of field j:

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

Method 2: Using 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)s and (x_{n+1}, d_{n+1}), for some j, $1 \leq j \leq (n+1)$, 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:

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

Each coefficient $f(x_0, x_1, \cdots, x_j)$ in the above formula is found by computing the divided differences [43] in the following equations:

$$f(x_0, x_1, \cdots, x_j) = (f(x_1, x_2, \cdots, x_j) - f(x_0, x_1, \cdots, x_{j-1}))/(x_j - x_0)$$
$$\text{and} \;\; f(x_j) = d_j, \quad \text{for some} \;\; j, 1 \leq j \leq (n+1).$$

4. When user i wants to read field j, he computes the following equation with his 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 database security. The characteristics of our two-phase enciphering scheme are:

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 in [11], 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 et al.'s scheme [12, 13] requires as many secret keys as there are records. The number of records is much larger than the number of fields in most database systems.

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

References

[1] Charles P. Pfleeger, *Security in Computing*, Prentice-Hall, 1989.

[2] Eduardo B. Fernandez, Rita C. Summers, and Christopher Wood, *Database Security and Integrity*, Addison-Wesley, Massachusetts, 1980.

[3] James Arlin Coper, *Computer & Communication Security: Strategies for the 1990s*, McGraw-Hill, New York, 1989.

[4] R. W. Conway, W. L. maxwell, and H. L. Morgan, On the Implementation of Security Measures in Information Systems, *Communications of the ACM* 15(4), 211–220 (Apr. 1972).

[5] G.S. Graham and P.J. Denning, Protection-Principles and Practice, In *Proc. Spring Jt. Computer Conf., Vol. 40, AFIPS*, Montrale, NJ, 1972, pp. 417–429.

[6] M. S. Hwang and W. P. Yang, A New Dynamic Access Control Scheme via Subject-Object-List, *Submitted publication*, (1992).

[7] Jeffrey D. Ullman, *Principles of Database and Knowledge-Base Systems, Volumn 1*, Computer Science, Maryland, 1988.

[8] National Bureau of Standard, *Data Encryption Standard*, FIPS, NBS, 1977.

[9] M. E. Smid and D. K. Branstad, The Data Encryption Standard: Past and Future, *Proc. of the IEEE* 76(5), 550–559 (May 1988).

[10] R. L. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Communications of the ACM* 21(2), 120–126 (Feb. 1978).

[11] George I. Davida, David L. Wells, and John B. Kam, A Database Encryption System with Subkeys, *ACM Transactions on Database Systems* 6(2), 312–328 (June 1981).

[12] C. H. Lin, C. C. Chang, and C. T. Lee, A Record-Oriented Cryptosystem for Database Sharing, In *International Computer Symposium*, R.O.C., Dec. 1990, pp. 328–329.

[13] C. S. Lin, *An Application of an Encryption Algorithm to Database Security, Chap. 3*, Ph.D. thesis, NCHU, R.O.C., 1991.

[14] Yair M. Babad and Jeffrey A. Hoffer, Data Element Security and Its Effects on File Segmentation, *IEEE Transactions on Software Engineering* SE-6(5), 402–410 (Sep. 1980).

[15] R. Bayer and J. K. Metzger, On the Encipherment of Search Trees and Random Access Files, *ACM Transactions on Database Systems* 1(1), 37–52 (Mar. 1976).

[16] Ragnar Eriksson and Kristian Beckman, Protection of Data-Bases Using File Encryption, In *Proceedings of the First Security Conference, IFIP/Sec'83*, 1983, pp. 217–221.

[17] Joan Feigenbaum, Mark Y. Liberman, and Rebecca N. Wright, Cryptographic Protection of Databases and Software, In *DIMACS Series in*

Discrete Math. and Theoretical Computer Science, Vol. 2, 1991, pp. 161–172.

[18] Ehud Gudes, The Design of a Cryptography Based Secure File System, *IEEE Transactions on Software Engineering* SE-6(5), 411–420 (Sep. 1980).

[19] Neal R. Wagner, Paul S. Putter, and Marianne R. Cain, Encrypted Database Design: Specialized Approaches, In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, 1986, pp. 148–153.

[20] C. J. Date, *An Introduction to Database Systems, Vol. 1, Fifth Edition*, Addison-Wesley, Massachusetts, 1990.

[21] Niven I and H. Zuckerman, *Introduction to the Theory of Numbers*, Wiley, New York, 1966.

[22] C. C. Chang, On the Design of a Key-Lock-Pair Mechanism in Information Protection System, *BIT* 26(3), 410–417 (1986).

[23] Guang-Huiou Chiou and Wen-Tsuen Chen, Secure Broadcasting Using the Secure Lock, *IEEE Transactions on Software Engineering* 15(8), 929–934 (Aug. 1989).

[24] C. C. Chang and T. C. Wu, Remote Password Authentication with Smart Cards, *IEE Proceedings-E* 138(3), 165–168 (May 1991).

[25] S. C. Lu and L. N. Lee, A Simple and Effective Public-Key Cryptosystem, *COMSAT Technical Review* 9(1), 15–23 (1979).

[26] C. C. Chang and C. S. Laih, Remote Password Authentication with Smart Cards (Correspondence), *IEE Proceedings-E* 139(4), 372 (July 1992).

[27] L. N. Lee, Note on Cryptosystems, *COMSAT Technical Review* 9(2B), 717–721 (1979).

[28] R. C. Merkle, One-Way Hash Functions and DES, In *Advances in Cryptology, CRYPTO'89*, Lecture Notes in Computer Science, Vol. 435, 1990, pp. 428–446.

[29] M. Naor and M. Yung, Universal One-Way Hash Functions and Their Cryptographic Applications, In *Proc. of the 21st STOC*, 1989, pp. 33–43.

[30] J. Rompel, One-Way Functions are Necessary and Sufficient for Secure Signatures, In *Proc. of the 22nd STOC*, 1990, pp. 387–394.

[31] S. G. Akl and P. D. Taylor, Cryptographic Solution to a Problem of Access Control in a Hierarchy, *ACM Transactions on Computer Systems* 1(3), 239–248 (July 1983).

[32] R. S. Sandhu, Cryptographic Implementation of a Tree Hierarchy for Access Control, *Information Processing Letters* 27, 95–98 (1988).

[33] Dorothy E. R. Denning, H. Meijer, and F. B. Schneider, More on master keys for group sharing, *Information Processing Letters* 13(3), 125–126 (Jan. 1981).

[34] I. Ingemarsson and C. K. Wong, A User Authentication Scheme for Shared Data Based on Trap-Door One-Way Functions, *Information Processing Letters* 12(2), 63–67 (1981).

[35] Dorothy E. R. Denning, *Cryptography and Data Security*, Addison-Wesley, Massachusetts, 1982.

[36] D. L. Wells, *A Short Note on the Dangers of Loading CRT Subkeys*, Technical Report TTTR-CSE-8106, Technical Report, Department of Computer Science and Engineering, SMU., Sep. 1981.

[37] D. E. Knuth, *The Art of Computer Programming, Vol. 2 (Seminumberical Algorithm), 2'nd ed.*, Addison-Wesley, Massachusetts, 1980.

[38] Thu Van Vu, Efficient Implementations of the Chinese Remainder Theorem for Sign Detection and Residue Decoding, *IEEE Transactions on Computers* C-34(7), 646–651 (July 1985).

[39] William Jr. Dirr and Fred J. Taylor, On Implementing the CRT in Residue Arithmetic, *The Journal of Computers Math.* 17, 155–163 (July 1985).

[40] E. F. Codd, *Relational Completeness of Data Base Sublanguages*, Prentice-Hall, Englewood Cliffs, N. J., 1972.

[41] George I. Davida and Y. Yeh, Cryptographic Relational Algebra, In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, 1982, pp. 111–116.

[42] Gerald C. Chick and Stafford E. Tavares, Flexible Access Control with Master Keys, In *Advances in Cryptology, CRYPTO'89*, Lecture Notes in Computer Science, Vol. 435, 1990, pp. 316–322.

[43] M. K. Jain, S. R. K. Lyengar, and R. K. Jain, *Numberical Methods for Scientific and Engineering Computation*, Wiley Eastern Limited, New Delhi, 1985.