Journal of Information Hiding and Multimedia Signal Processing ©2014 ISSN 2073-4212
Ubiquitous International Volume **5**, Number **4**, October **2014**

A Database Encryption Scheme Based on the Generalized Aryabhata Remainder Theorem

[1,2]Yanjun Liu, and [2,3]Chin-Chen Chang

[1]Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education
School of Computer Science and Technology
Anhui University, Hefei, 230039, China
yjliu104@gmail.com

[2]Department of Computer Science and Information Engineering
Asia University, Taichung, 413, Taiwan
No.26 Hexing Road Xiangfang District, Harbin, P.R.China
yjliu104@gmail.com, alan3c@gmail.com

[3]Department of Information Engineering and Computer Science
Feng Chia University, Taichung, 407, Taiwan
alan3c@gmail.com

Received August, 2013; revised January, 2014

Abstract. *Database security prevents the disclosure of confidential data within a database to unauthorized users, and has become an urgent challenge for a tremendous number of database applications. Data encryption is a widely-used cryptographic technique for realizing database security in which the data kept in the database are encrypted into ciphertext. Inspired by Lin et al.'s database encryption system, we propose a record-oriented database encryption scheme based on the Generalized Aryabhata Remainder Theorem (GART). Analysis showed that our proposed scheme is more efficient than Lin et al.'s scheme, while retaining the same security.*

Keywords: Database security, Data encryption, Generalized aryabhata remainder theorem (GART)

1. Introduction.

A database usually plays an important role for information sharing among different users in an organization or distributed system. Compared with conventional data storage methods, the use of a database has the following features [5]: shared access, minimal redundancy, data consistency, data integrity, and controlled access. Since databases have been applied extensively in the areas of information security, e-commerce, and distributed computing, a large amount of critical and sensitive data are now stored in databases. Therefore, solving the problem of database security [10-13, 16, 18, 23, 27, 29], that is, protecting confidential data from being disclosed to unauthorized users has become an urgent challenge for a considerable number of database applications.

There are three traditional approaches [7, 19] for providing database security: 1) physical security; 2) operating system security; and 3) database management system (DBMS) security. These three approaches, however, cannot guarantee that the database is completely secure for two reasons [14, 15]. First, it is hard to prevent the disclosure of data. Since raw data are maintained in readable form in the database, any person who has the access right to the raw data according to the access control mechanism can easily bypass the three traditional approaches and then obtain the correct content of the data. In addition, operations for backing up important data and keeping data confidential in a distributed system make it more difficult to control the disclosure of data. Second, it is hard to achieve data authentication. If an intruder is able to access the raw data in readable form, they may subsequently forge the data, resulting in authorized users not being able to obtain the original data from the database.

In recent years, data encryption [1, 2, 7, 8, 14, 15, 19, 24, 26] has been used as a common method for realizing database security and it can overcome the drawbacks of the three approaches mentioned previously. Data encryption is a cryptographic technique that encrypts the data kept in the database into ciphertext. Therefore, data encryption possesses the following advantages over conventional methods: 1) it solves the problem of data disclosure, as the data are encrypted with an encryption key rather than staying in readable form, making it unreadable to any unauthorized users who do not know the corresponding decryption key; and 2) solves the problem of data authentication. It is impossible for an intruder to alter the encrypted data without knowing the encryption key, thereby ensuring that the information gained from the database is authentic.

In 1981, Davida et al. [7] proposed a database encryption system with subkeys, which is a significant breakthrough for data encryption. Their encryption system is record-oriented in such a way that each record can be encrypted by subkeys of different fields and each field can be separately decrypted by a subkey. More specifically, the ciphertext of each encrypted record is a single encrypted value that is a function of all fields based on the Chinese Remainder Theorem (CRT). A user cannot read the whole record, and can only obtain some field values of the record if authorized to have the corresponding decryption subkey. Encryption at the level of records and decryption at the level of fields can effectively enhance database security.

In the past decades, many database encryption schemes that originated from the concept of Davida et al.'s method have been developed in the literature [2, 8, 14, 15, 19, 24]. In 1992, Lin et al. [19] pointed out that although Davida et al.'s method can satisfy security requirements and withstand many types of attack, such as a pattern matching attack, substitution attack, and known-plaintext attack, it needs to concatenate a random redundancy value with each field, resulting in a large storage overhead. In order to overcome this weakness, they proposed a record-oriented database cryptosystem based on the Generalized Chinese Remainder Theorem (GCRT) in which an extra private key is associated with each record instead of a random redundancy value with each field. In 1995, Hwang and Yang [14] presented a two-phase record-oriented encryption scheme that uses both the one-way function and the subkey enciphering approach to enhance database security. In this method, there is no need to extend the raw data within the database, and as such, storage efficiency is improved. In 1997, Hwang and Yang [15] proposed a multilevel secure database encryption scheme with subkeys. The scheme is also record-oriented and can encrypt each record with subkeys of different fields with different security classes, whereas each field can only be decrypted separately by the subkey of a security class that is higher than or equal to that of the encrypted subkeys. In 2003, Chang and Chan [2] proposed two database encryption systems based on the concept of RSA public key, where one is record-oriented and the other is field-oriented. Their systems enable authorized users to make encryption and decryption operations on the protected fields directly and reduce the key management problem. In 2004, Elovici et al. [8] pointed out that since the basic encryption unit is a record in a record-oriented database encryption scheme, it can cause a problem when changing the structure of the database. Therefore, they proposed a structure-preserving database encryption scheme to encrypt all the content of the database. Recently, Manivannan and Sujarani [24] presented a lightweight database encryption technique. They use a TSTF (Transposition, Substitution, Folding, Shifting) algorithm that only encrypts sensitive data to achieve high security and efficiency.

In this paper, we propose a record-oriented database encryption scheme that is based on the Generalized Aryabhata Remainder Theorem (GART). Our proposed scheme is a variation of Lin et al.'s scheme [19] and its objective is to improve the efficiency in the procedures of encryption and decryption. The main contributions of our proposed scheme are listed below:

(1) For the encryption procedure, GART is used to convert the complete contents of a record into a ciphertext. Authorized users can then use their decryption keys to recover the ciphertext to the original field values of the record with GART in the decryption procedure.
(2) Due to the intrinsic characteristics of the GART, an extra private key k is required for each record. When the ciphertext needs to be updated, only the value of k must be modified without changing other parameters. This implies flexibility in our proposed scheme.
(3) The computational cost of our proposed scheme is lower than that of scheme by Lin et al.
(4) Our proposed scheme achieves the same degree of security as Lin et al.'s scheme.

The rest of this paper is organized as follows. In Section 2, we briefly introduce some background related to a record-oriented database encryption system. In Section 3, we propose our database encryption system based on the GART. Section 4 gives security and efficiency analyses of the proposed scheme. Finally, our conclusions are presented in Section 5.

2. Preliminaries.

In this section, we briefly review some background related to a record-oriented database encryption system. In Subsections 2.1 and 2.2, we review two basic constructing elements for many database encryption systems, i.e., the Chinese Remainder Theorem (CRT) and the Generalized Chinese Remainder Theorem (GCRT), respectively. In Subsection 2.3, we introduce Lin et al.'s database encryption system [19] based on the GCRT.

2.1. The Chinese remainder theorem.

The Chinese Remainder Theorem (CRT) is a mathematical method for computing an integer in a specific range. It has been used extensively in the field of cryptology [6, 9, 21, 22, 28], such as for a group key distribution protocol, an access control mechanism, oblivious transfer protocol, secret sharing scheme, and a secure broadcasting method. The CRT can be defined as follows. Assume that there are t positive, pairwise, relative prime integers, $p_1, p_2, \ldots, p_t$, and t positive integers, $x_1, x_2, \ldots, x_t$. We can construct a system of equations by using $p_1, p_2, \ldots, p_t$ and $x_1, x_2, \ldots, x_t$ to find an integer X. The system of equations is shown as follows:

$$x_1 = X(\text{mod} p_1),$$

$$x_2 = X(\text{mod} p_2),$$

$$x_t = X(\text{mod} p_t).$$

The CRT solves the above system of equations in such a way that $X = \sum\limits_{i=1}^{t} M_i \cdot M_i' \cdot x_i(\bmod P)$, where $P = \prod_{i=1}^{t} p_i$, $M_i = \frac{P}{p_i}$, and $M_i \cdot M_i' \equiv 1(\bmod p_i)$. It must be noticed that X is the unique solution in Z_p.

2.2. **Generalized Chinese remainder theorem.** Although the CRT is applied in many applications of cryptology, its generalized version X generalized Chinese remainder theorem (GCRT) [3, 17] can provide more flexibility and thus is more practical than the CRT. Suppose that there are t positive,pairwise, relative prime integers, $p_1, p_2, \ldots, p_t$,t positive integers, $x_1, x_2, \ldots, x_t$, and an integer k, where k is selected to satisfy the condition $\text{Max}\{x_i\}_{1\leq i\leq t} < k < \text{Min}\{p_i\}_{1\leq i\leq t}$. Then, a system of equations can be constructed as follows:

$$x_1 = \left\lfloor \frac{X}{p_1} \right\rfloor (\text{mod} k),$$

$$x_2 = \left\lfloor \frac{X}{p_2} \right\rfloor (\text{mod} k),$$

$$\vdots$$

$$x_t = \left\lfloor \frac{X}{p_t} \right\rfloor (\text{mod} k).$$

The GCRT computes the unique solution X in Z_{kp} by $X = \sum\limits_{i=1}^{t} A_i \cdot A'_i \cdot B_i(\bmod k \cdot P)$, where $P = \prod_{i=1}^{t} p_i$, $A_i = k \cdot \frac{P}{p_i}$, $A_i \cdot A'_i = k(\text{mod} k \cdot p_i)$ and $B_i = \left\lceil \frac{x_i \cdot p_i}{k} \right\rceil$. In addition, the values of $k \cdot P$ and $A_i \cdot A'_i$ can be pre-computed to speed up the computational process.

From the computational process of the GCRT, it is can be implied that the GCRT is more flexible than the CRT since it contains an extra integer k. More specifically, the flexibility of the GCRT relies on the fact that if we want to change the integer X, we only need to change the value of k without modifying the whole system of equations.

2.3. **Review of Lin et al.s database encryption system.** Lin et al.'s database encryption system [19] that is a valuable extension of Davida et al.s encryption system [7], is record-oriented and is based on the GCRT. We first introduce the notations used in their database encryption system and then review how the system works.

Let C denote the ciphertext of an encrypted record in the database and each record contains t fields. Let f_i denote the value of field i for $i = 1, 2, ..., t$. $p_1, p_2, \ldots, p_t$ denote t positive integers that are pairwise relatively prime to each other. The integer k denotes an extra private key for each record that satisfies $\text{Max}\{f_i\}_{1\leq i\leq t} < k < \text{Min}\{p_i\}_{1\leq i\leq t}$. Each pair (k, p_i) is the decryption key for field i. Lin et al.'s database encryption system consists of two procedures, i.e., encryption and decryption procedure. The encryption procedure is to encrypt all the field values of a record into the ciphertext C by using the GCRT, which is executed as follows:

$$C = \sum_{i=1}^{t} A_i \cdot A'_i \cdot B_i(\bmod k \cdot P), \tag{1}$$

where $P = \prod_{i=1}^{t} p_i$, $A_i = k \cdot \frac{P}{p_i}$, $A_i \cdot A'_i = k(\text{mod} k \cdot p_i)$ and $B_i = \left\lceil \frac{f_i \cdot p_i}{k} \right\rceil$.

On the other hand, the decryption procedure is responsible for decrypting the ciphertext C to raw data in each field according to the GCRT. It can be conducted easily by the following operations:

$$f_i = \left\lfloor \frac{C}{p_i} \right\rfloor (\text{mod}k).$$

After decryption, the user can read the original data in field i if authorized to have the decryption key (k, p_i) for field i. Consequently, a user can only read some field values of the record depending on which decryption key they know.

Lin et al.'s database encryption system surpasses Davida et al.'s encryption system due to the use of GCRT.Davida et al.'s encryption system is based on the CRT, which needs to add a random redundancy value for each field to prevent known-plaintext attack, thereby increasing the storage overhead.In contrast, Lin et al.'s database encryption system only utilizes an extra private key k for each record according to the GCRT.This can withstand a known-plaintext attack and also significantly reduce storage overhead. The efficiency of Lin et al.'s system, however, can still be improved, especially for computational efficiency in the encryption and decryption procedures.As such, we propose a new database encryption scheme to achieve higher efficiency.

3. **Our proposed scheme.**

In this section, we propose a new record-oriented database encryption scheme, which is a variation of Lin et al.'s scheme.The aim of our proposed scheme is to enhance computational efficiency. We use the generalized Aryabhata remainder theorem (GART) instead of GCRT in the encryption and decryption procedures.Similar to any record-oriented scheme, the data can be protected by using the strategy of encryption at the level of records and decryption at the level of fields.In the encryption procedure, the GART is used to convert the entire contents of a record into a ciphertext. Then authorized users can use their decryption key to recover the ciphertext to the original field values of the record by the GART in the decryption procedure.

In the following, we will introduce the definition of GART and the process of computing an integer by using the GART. Then we propose our new database encryption scheme in detail, using the GART as the main building block.

3.1. **Generalized Aryabhata remainder theorem.**

In 2006, Rao and Yang proposed the Aryabhata remainder theorem (ART) [25] with two relatively prime moduli. Later in 2010, Chang et al. [4, 20] extended the ART to t moduli to propose the generalized Aryabhata remainder theorem (GART). Let $p_1, p_2, \ldots, p_t$ be t positive, pairwise, relative prime integers, $x_1, x_2, \ldots, x_t$ be t positive integers, and k be an integer that is selected to satisfy the condition $\text{Max}\{x_i\}_{1 \le i \le t} < k < \text{Min}\{p_i\}_{1 \le i \le t}$. Then, a system of equations can be constructed as follows:

$$x_1 = \left\lfloor \frac{X}{p_1} \right\rfloor (\text{mod}k),$$

$$x_2 = \left\lfloor \frac{X}{p_2} \right\rfloor (\text{mod}k),$$

$$\vdots$$

$$x_t = \left\lfloor \frac{X}{p_t} \right\rfloor (\text{mod}k).$$

In order to compute the solution X by using the GART, an iterative algorithm is proposed as shown below:

GART Algorithm
Input:$(k, \{x_1, x_2, \ldots, x_t\}, \{p_1, p_2, \ldots, p_t\})$
Output: integer X

```
1. $P_1 = p_1, X_1 = x_1 \cdot p_1$
2. for $i = 2$ to $t$ do
3. $P_i = P_{i-1} \cdot p_i$
4. $X_i = k \cdot P_{i-1} \cdot ((\lceil (x_i \cdot p_i - X_{i-1})/k \rceil \cdot (P_{i-1})^{-1}) \bmod p_i) + X_{i-1}$,
   where $(P_{i-1})^{-1} \cdot P_{i-1} = 1 \bmod p_i$.
5. end for
6. return $X_t$
```

In the GART algorithm, the values of P_i, $k \cdot P_{i-1}$, $(P_{i-1})^{-1} \bmod p_i$, and $k \cdot P_{i-1} \cdot ((P_{i-1})^{-1} \bmod p_i)$ can be pre-determined to accelerate the computational process.After t-1 rounds of the GART algorithm are done, the solution X can be obtained.

3.2. **Our proposed scheme.** We now describe our new database encryption scheme based on the GART. Assume that there are t fields in each record within the database and f_i denotes the value of field i for $i = 1, 2, ..., t$. C denotes the ciphertext of an encrypted record. $p_1, p_2, \ldots, p_t$ denote positive integers that are pairwise relatively prime to each other. The integer k denotes an extra private key for each record that satisfies $\text{Max}\{f_i\}_{1 \le i \le t} < k < \text{Min}\{p_i\}_{1 \le i \le t}$. Each pair (k, p_i) is the decryption key for field i.

Our proposed database encryption system consists of two procedures, i.e., an encryption and decryption procedure. In the encryption procedure, the GART is used to convert all the field values of a record into the ciphertext C. Since the GART computes an integer in an iterative way, the ciphertext C can be obtained as follows:

$$C_i = \begin{cases} f_i \cdot p_i, \text{for} i = 1, \\ g(C_{i-1}, k, P_{i-1}, f_i, p_i), \text{for} i = 2, 3, ..., t, \end{cases} \tag{2}$$

where $C = C_t$, $P_1 = p_1$, $P_i = P_{i-1} \cdot p_i$, and $g(C_{i-1}, k, P_{i-1}, f_i, p_i) = k \cdot P_{i-1} \cdot ((\lceil (f_i \cdot p_i - C_{i-1})/k \rceil \cdot (P_{i-1})^{-1}) \bmod p_i) + C_{i-1}$ for $i = 2, 3, ..., t$. A diagram of the encryption procedure is illustrated in Figure 1.

Figure 1 indicates that the encryption procedure is divided into t steps. In Step 1, C_1 is calculated by $C_1 = f_1 \cdot p_1$; while in Step $i(i = 2, 3, ..., t)$, C_i is calculated by the function g of parameters $C_{i-1}, k, P_{i-1}, f_i, \text{and} p_i$ based on the GART. After t steps are executed, the encryption procedure is completed so that the raw data in all fields of a record, $f_1, f_2, ..., f_t$, are encrypted into the ciphertext C. In addition, whenever any field value is altered, all fields must be re-encrypted into a new ciphertext by selecting new k and $p_1, p_2, \ldots, p_t$ according to Equation (2).

On the other hand, the value of field i can be recovered from the ciphertext C which was obtained in the encryption. The decryption procedure is done by the following equation:

$$f_i = \left\lfloor \frac{C}{p_i} \right\rfloor (\bmod k), \tag{3}$$

where(k, p_i) is the decryption key for field i. When a user has the decryption keys for all fields of a record, they can successfully recover all the original data in this record; otherwise, a user can only read some parts of the record depending on the decryption key

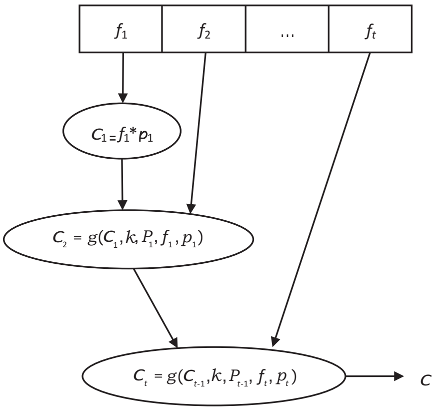
f_1
f_2
...
f_t
$C_1=f_1*p_1$
$C_2 = g(C_1,k,P_1,f_1,p_1)$
$C_t = g(C_{t-1},k,P_{t-1},f_t,p_t)$
C

FIGURE 1. Encryption procedure of our proposed database encryption scheme

that they own. The decryption of our proposed scheme is the same as Lin et al.'s scheme [19] and is depicted in Figure 2.

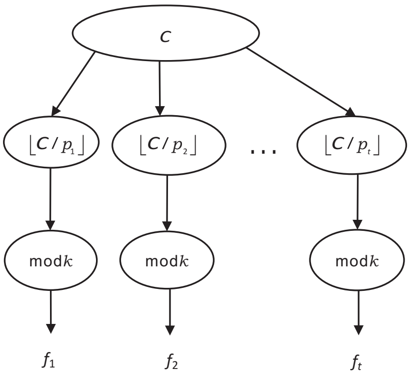

FIGURE 2. Decryption procedure of our proposed database encryption scheme

By adopting the GART, our proposed scheme has all of required characteristics of a database encryption scheme [7]. Moreover, our proposed scheme is flexible since an extra private key k is provided for each record. When the ciphertext C needs to be updated, only the value of k must be modified, while the other parameters are left unchanged.

3.3. **Example.** This example is given to illustrate the encryption and decryption procedures of our proposed scheme.

Example 3.1. Assume that there are four fields of a record within the database and let$\{f_1, f_2, f_3, f_4\} = 4,\ 7,\ 5,\ 2,\{p_1, p_2, p_3, p_4\} = 13,\ 19,\ 17,\ 11$, and $k = 9$.

Encryption procedure

The ciphertext C is computed through three steps as shown below:

STEP1:

$C_1 = f_1 \cdot p_1 = 4 \cdot 13 = 52.$

STEP2:

$$\begin{aligned} C_2 &= k \cdot P_1 \cdot ((\lceil (f_2 \cdot p_2 - C_1)/k \rceil \cdot (P_1)^{-1}) \bmod p_2) + C_1 \\ &= 9 \cdot 13 \cdot ((\lceil (7 \cdot 19 - 52)/9 \rceil \cdot 13^{-1}) \bmod 19) + 52 \\ &= 9 \cdot 13 \cdot 8 + 52 \end{aligned}$$

STEP3:

$$\begin{aligned}C_3 &= k \cdot P_2 \cdot ((\lceil (f_3 \cdot p_3 - C_2)/k \rceil \cdot (P_2)^{-1}) \bmod p_3) + C_2\\ &= 9 \cdot 247 \cdot ((\lceil (5 \cdot 17 - 988)/9 \rceil \cdot 247^{-1}) \bmod 17) + 988\\ &= 9 \cdot 247 \cdot 4 + 988\\ &= 9880.\end{aligned}$$

STEP4:

$$\begin{aligned}C_4 &= k \cdot P_3 \cdot ((\lceil (f_4 \cdot p_4 - C_3)/k \rceil \cdot (P_3)^{-1}) \bmod p_4) + C_3\\ &= 9 \cdot 4199 \cdot ((\lceil (2 \cdot 11 - 9880)/9 \rceil \cdot 4199^{-1}) \bmod 11) + 9880\\ &= 9 \cdot 4199 \cdot 2 + 9880\\ &= 85462 = C.\end{aligned}$$

Decryption procedure

Each field value is recovered as follows:

$f_1 = \lfloor C/p_1 \rfloor \bmod k = \lfloor 85462/13 \rfloor \bmod 9 = 4$

$f_2 = \lfloor C/p_2 \rfloor \bmod k = \lfloor 85462/19 \rfloor \bmod 9 = 7$

$f_3 = \lfloor C/p_3 \rfloor \bmod k = \lfloor 85462/17 \rfloor \bmod 9 = 5$

and $f_4 = \lfloor C/p_4 \rfloor \bmod k = \lfloor 85462/11 \rfloor \bmod 9 = 2$

Figures 3 and 4 demonstrate the encryption and decryption procedures of Example 3.1, respectively.

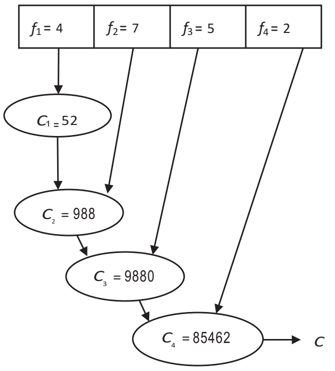

Figure 3. Encryption procedure of Example 3.1

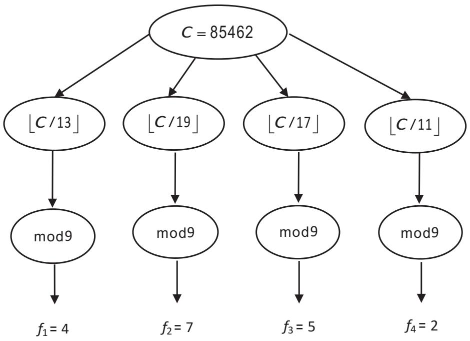

Figure 4. Decryption procedure of Example 3.1

4. **Analysis.** In this section, we analyze the security and efficiency of our proposed database encryption scheme. The security of a database encryption scheme depends on how much effort a cryptoanalyst must put in to break it. In the following, we examine whether our proposed scheme can withstand a known-plaintext attack. Assume that there are two record R and R'. Let C and C'be the ciphertexts of R and R', respectively, and f_i and f_i'

be the values of field i of R and R', respectively. C, C', f_i and ${f_i}'$, are revealed to a cryptoanalyst,whereas k, k' and P_i are kept secret. According to Equation (2), a cryptoanalyst can establish some simultaneous equations as follows:

$\lfloor C/p_i \rfloor \bmod k = f_i$, and $\lfloor C'/p_i \rfloor \bmod k' = {f_i}'$

implying

$\lfloor C/p_i \rfloor - f_i = d_1 k$ and $\lfloor C'/p_i \rfloor - {f_i}' = d_2 k'$

where d_1 and d_2 are two integers.

The above two simultaneous equations have three unknown variables, P_i, k, and k', which indicates that P_i has infinite possible solutions. There is, however, only one solution for P_i and is the one selected in our database encryption scheme; therefore, the possibility for the cryptoanalyst to obtain the correct P_i is very low. This case can be generalized if the cryptoanalyst knows the t records to construct t simultaneous equations, in which t+1 unknown variables need to be solved. In this case, the cryptoanalyst still cannot obtain P_i Thus, the known-plaintext attack can be successfully prevented. This implies that our proposed scheme achieves the same degree of security as Lin et al.'s scheme.

Next, we compare the efficiency of our proposed scheme with that of Lin et al.'s scheme [19]. Since our scheme has the same decryption procedure as Lin et al.'s scheme, we will only investigate the encryption procedure of both schemes that dominates the computational efficiency.

In the encryption of Lin et al.'s scheme, all field values of a record are converted to the ciphertext C by using the GCRT according to Equation (1). Let us use the number of required bit operations to measure the computational cost of Equation (1). Assume that f_i,p_i and k are all assigned m digits. Equation (1) requires $2t$ multiplications, t divisions, (t-1) additions, and one modular operation. Because the multiplication of two integers requires m^2 bit operations, the addition requires m bit operations, and the modular operation with an m-bit integer needs m^2 bit operations, the computational cost of computing C is about $[2t \times m^2 + t \times m^2 + (t-1) \times m + ((t+1) \times m)^2$ bit operations. Therefore, the time complexity is $O(t^2m^2)$.

In contrast, the encryption of our proposed scheme utilizes the GART according to Equation (2). The difference between the GCRT and the GART comes from the GCRT use of a modular operation with a large integer, say kP, as its last operation, whereas the GART uses a modular operation with a relatively smaller integer, say p_i, in each iteration. Thus, the use of the GART can reduce the computational cost. In Equation (2), computation of C_i requires two multiplications, one subtraction, one division, one addition, and one modular operation. Consequently, the total computational cost of computing C is $(t-1) \times (2m^2 + m + m^2 + m + m^2)$ and thus the time complexity is $O(tm^2)$. The analysis shows that our proposed scheme is more efficient than Lin et al.'s scheme. Table 1 compares our proposed scheme with Lin et al.'s scheme.

Table 1. Comparison of our proposed scheme and Lin et al.'s scheme

Scheme	Method	Time complexity of encryption	Time complexity of decryption
Lin et al. [19]	GCRT	$O(t^2m^2)$	$O(m^2)$
Ours	GART	$O(tm^2)$	$O(m^2)$

5. **Conclusions.**

In this paper, we proposed a record-oriented database encryption scheme based on the GART. In the procedure of encryption, the GART is used to convert the entire content of a record into a ciphertext. Authorized users can then use their decryption key to recover the ciphertext to the original field values of the record by the GART in the decryption procedure. Analysis showed that our proposed scheme is more efficient than the scheme by Lin et al., while maintaining the same security.

Acknowledgements. This research was supported in part by the National Nature Science Foundation of China (grant number: 61202228) and the College Natural Science Key Project of Anhui Province of China (grant number: KJ2012A008).

REFERENCES

[1] L. Bouganim, and Y. Guo, Database encryption, *Encyclopedia of Cryptography and Security*, pp. 1-9, 2009.

[2] C. C. Chang, and C. W. Chan, A Database record encryption scheme using the RSA public key cryptosystem and its master keys, *Proc. of International Conference on Computer Networks and Mobile Computing*, pp. 345-348, 2003.

[3] C. C. Chang, and Y. P. Lai, A fast modular square computing method based on the generalized chinese remainder theorem for prime module, *Applied Mathematics and Computation*, vol. 161, no. 1, pp. 181-194, 2005.

[4] C. C. Chang, J. S. Yeh, and J. H. Yang, Generalized aryabhata remainder theorem, *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 4, pp. 1865-1871, 2010.

[5] C. Y. Chen, C. Y. Ku, and D. C. Yen, Cryptographic relational algebra for databases using the field authenticator, *Computers and Mathematics with Applications*, vol. 54, no. 1, pp. 38-44, 2007.

[6] Y. Cheng, J. Park, and R. Sandhu, A user-to-user relationship-based access control model for online social networks, *Data and Applications Security and Privacy*, LNCS 7371, Springer, pp. 8-24, 2012.

[7] G. I. Davida, D. L. Wells, and J. B. Kam, A database vncryption system with subkeys, *ACM Transactions on Database Systems*, vol. 6, no. 2, pp. 312-328, 1981.

[8] Y. Elovici, R. Waisenberg, E. Shmueli, and E. Gudes, A structure preserving database encryption Scheme, *Secure Data Management*, LNCS 3178, Springer, pp. 28-40, 2004.

[9] C. Guo, and C. C. Chang, An authenticated group key distribution protocol based on the generalized chinese remainder theorem, *International Journal of Communication Systems*, vol. 27, no. 1, pp. 126-134 , 2014.

[10] T. Hardjono, Y. Zheng, and J. Seberry, Database authentication revisited, *Computers and Security*, vol. 13, no. 7, pp. 573-580, 1994.

[11] D. He, J. Chen, and J. Hu, A pairing-free certificateless authenticated key agreement protocol, *International Journal of Communication Systems*, vol. 25, no. 2, pp. 221-230, 2012.

[12] M. S. Hwang, W. G. Tzeng, and W. P. Yang, An access control scheme based on chinese remainder theorem and time stamp concept, *Computers and Security*, vol. 15, no. 1, pp. 73-81, 1996.

[13] M. S. Hwang, and W. P. Yang, A new dynamic access control scheme based on subject-object-list, *Data and Knowledge Engineering*, vol. 14, no. 1, pp. 45-56, 1994.

[14] M. S. Hwang, and W. P. Yang, A two-phase encryption scheme for enhancing database security, *Journal of Systems and Software*, vol. 31, no. 3, pp. 257-265, 1995.

[15] M. S. Hwang, and W. P. Yang, Multilevel secure database encryption with subkeys, *Data and Knowledge Engineering*, vol. 22, no. 2, pp. 117-131, 1997.

[16] S. Jajodia, and R. Sandhu, Toward a multilevel secure relational data model, *SIGMOD Record*, vol. 20, no. 1, pp. 50-59, 1991.

[17] Y. P. Lai, and C. C. Chang, Parallel computational algorithms for generalized chinese remainder theorem, *Computers and Electrical Engineering*, vol. 29, no. 8, pp. 801-811, 2003.

[18] C. F. Lee, H. Y. Chien, and C. S. Laih, Server-less RFID authentication and searching protocol with enhanced security,*International Journal of Communication Systems*, vol. 25, no. 3, pp. 376-385, 2012.

[19] C. H. Lin, C. C. Chang, and R. C. T. Lee, A record-oriented cryptosystem for database sharing, *The Computer Journal*, vol. 35, no. 6, pp. 658-660, 1992.

[20] Y. Liu, C. C. Chang, and S. C. Chang, A residual number system oriented group key distribution mechanism,*International Journal of Information Processing and Management*, vol. 4, no. 3, pp. 146-155, 2013.

[21] Y. Liu, C. C. Chang, and S. C. Chang, An efficient oblivious transfer protocol using residue number system,*International Journal of Network Security*, vol. 15, no. 3, pp. 212-218, 2013.

[22] Y. Liu, L. Harn, and C. C. Chang, An authenticated group key distribution mechanism using theory of numbers,*International Journal of Communication Systems*, 2013.

[23] T. F. Lunt, D. E. Denning, R. R. Schell, M. Heckman, and W. R. Shockley, The seaView security model, *IEEE Transactions on Software Engineering*, vol. 16, no. 6, pp. 593-607, 1990.

[24] D. Manivannan, and R. Sujarani, Light weight and secure database encryption using TSFS algorithm, *Proc. of International Conference on Computing Communication and Networking Technologies*, pp. 1-7, 2010.

[25] T. R. N. Rao, and C. H. Yang, Aryabhata remainder theorem: relevance to public-key crypto-algorithms, *Circuits, Systems, and Signal Processing*, vol. 25, no. 1, pp. 1-15, 2006.

[26] E. Shmueli, R. Vaisenberg, Y. Elovici, and C. Glezer, Database encryption: an overview of contemporary challenges and design considerations, *ACM SIGMOD Record*, vol. 38, no. 3, pp. 29-34, 2010.

[27] P. D. Stachour, and B. Thuraisingham, Design of LDV: a multilevel secure relational database management system, *IEEE Transactions on Knowledge and Data Engineering*, vol. 2, no. 2, pp. 190-209, 1990.

[28] T. C. Wu, and Y. S. Chang, Authorization-based group-oriented secure broadcasting system, *Journal of Information Science and Engineering*, vol. 15, no. 5, pp. 653-667, 1999.

[29] T. C. Wu, Y. S. Yeh, and C. C. Chang, Algebraic operations on encrypted relational databases, *Information Systems*, vol. 18, no. 1, pp. 55-62, 1993.