Research on Implementation Method of Key Management Based on Data Encryption Technology

Shi Yan
Bengbu University, Bengbu, China

Abstract. Key management is a vital problem which must be solved when a database encryption system is designed and implemented. It is very difficult to solve key management. Key management is directly related to the performance of the whole encryption system and security of encrypted data. Key management is discussed by the author from the aspects of generation, distribution, storage, exchange and change. A concrete scheme of key management is proposed. The security of the scheme is discussed. The discussion result shows that it is a feasible key management scheme which can meet the security requirements of the database encryption system.

1. Introduction
Key management occupies a decisive position in database encryption system. Even if encryption algorithm has high strength, and it even cannot be broken by cipher text-only attack or plaintext attack, once the key is disclosed or stolen if the key is not managed well, the data becomes unsafe completely, and encryption is meaningless.

However, key management is a difficulty in database encryption system design. It is related to general process of key production, inspection, distribution, transmission, storage, use and destruction as a comprehensive technology. Though the encryption strength can be fully guaranteed by the key management scheme proposed in article [1], it has two obvious drawbacks. Firstly, it fails to propose an excellent master key protection scheme. Once the master key is leaked or breached, all the encrypted data will be exposed. Secondly, a lot of time is required for change of master key. Because the operator should decrypt all encrypted data every time the master key is changed, then a new master key should be used for encrypting the deciphered data. In contrast, though the master key can be protected well by the two-level conversion table key management scheme proposed in articles [2, 3, 5], the intensity of data is greatly reduced by the method of data encryption adopted in the scheme, thereby greatly reducing the security of the encrypted data. We discuss key management from the most important aspects of key management. In particular, we discuss how to organically combine the two above-mentioned key management solutions for protecting the master key which has a decisive impact on encrypted data security.

2. Key in database encryption system
Keys of the database encryption system can be divided into two categories mainly: one is user key, and the other is data key. User keys can be divided into user private key and user class key. Data can be divided into encrypted data and clear data after data encryption in the database. Corresponding database users can be divided into two categories, one is database administrator with the maximum authority, who can handle both clear data and encrypted data. The other category refers to general visitor who can
process clear data only. Because each kind of users contains different users, different users have different permissions, which can be realized through establishing a user permission table. Users can be divided into different classes (or groups) according to different access rights. These different classes are called user classes. Key corresponding to user classes is called user class key. It is obvious that the different user classes have different user class keys. Users belonging to the same user class have the same corresponding user class key though their user keys are different generally.

Data key is used for encrypting data, which is divided into master key and working key. Master key is divided into clear data master key and encrypted data master key according to the classification of database data. Clear data master key is mainly used for verifying the legality of user identity accessing clear data. Encrypted data master key is mainly used for generating the working key for encryption and densification, and it is not directly used as the data encryption key. Different field values have different working keys. The working key is regarded as the encryption key of the field value. OTP encryption mode is adopted here, and it is the safest encryption mode [1].

3. Key generation

Key generation is the first and very important step of key management. How to generate an excellent key has very significant influence on encrypted data security.

3.1. Generation of user private key and master key

User private key generation and master key generation are based on chaos theory [2]. The chaos is a motion state which widely exits in the nature. It has no cycle, no order, and nonlinear change, upward and downward fluctuation. Chaos is characterized by nonlinearity, butterfly effect, fractal dimension and impossibility of long-term forecast. Chaos results which are unpredictable theoretically are obtained in the determined mathematical form according to random sequence obtained from the chaos. It is nearly impossible to analyze the system comprehensively and descriptively. Therefore, it is an excellent key source.

3.2. Generation of user class key

3.2.1. Generation process of user class key. Data processed by the database encryption system can be divided into two categories: one is clear data, and the other is encrypted data. Users also can be divided into two categories correspondingly: one is database administrator, and the other is general visitor, users of different categories have different access rights to database. Database administrator can operate both clear data and encrypted data. General visitor can process data of clear text only. The database processed by the encrypted database system is set as \( D = \{d_1, d_2\} \), where \( d_1 \) refers to encrypted data, \( d_2 \) refers to clear data, they respectively correspond to data key \( K_D = \{k_{d_1}, k_{d_2}\} \) and user class \( U = \{uc_1, uc_2\} \), \( uc_1 \) refers to database administrator, \( uc_2 \) refers to general visitor, \( uc_1 \) grade is higher than \( uc_2 \), and user class key \( k_{uc_1} \) and \( k_{uc_2} \) were respectively mastered. Each security class data corresponds to a parameter, and it is called multidirectional lock, which is expressed by \( ML = \{ml_1, ml_2\} \). It is assumed that \( f \) represents transfer function, and the above conditions can be expressed as follows formally:

\[
\begin{align*}
 f (k_{uc_1}, ml_1) & = k_{d_1} \\
 f (k_{uc_1}, ml_2) & = k_{d_2} \\
 f (k_{uc_2}, ml_2) & = k_{d_2} \\
 f (k_{uc_2}, ml_1) & \neq k_{d_1}
\end{align*}
\]
Quadratic residue function meeting the four conditions can be discovered:

$$F(x, y) = (x + y) \equiv 0 \pmod{n}$$  \hspace{1cm} (5)

In the formula, \(n = pq\), \(p = q = 3 \pmod{4}\), \(p\) and \(q\) are two major prime numbers. If there are solutions for the equation \(x^2 \equiv a \pmod{n}\) \((n\) meets the condition of formula \((5))\), there are four solutions about module \(n\) according to relevant theorem of quadratic residue, namely \(\pm x_0, \pm x_1\). When \(p\) and \(q\) are known, the 4 solutions can be obtained from \((a, p, q)\). However, if \(p\) and \(q\) are unknown, and it is assumed that it is difficult to decompose \(p\) and \(q\) from most \(n\). It is impossible to obtain \(x_0\) and \(x_1\) from \(a\) and \(n\). It is also impossible to obtained \(x_1\) from \((x_0, a and n)\). On the contrary, \(x_0\) (or \(-x_0\)) and \(x_1\) (or \(-x_1\)) are called a pair of roots which are diverse.

Key generation process is shown as follows:

1. General visitor key \(k_{uc2}\) and encrypted data multidirectional lock \(ml_2\) are generated according to above-mentioned chaos theory;
2. Clear data key is obtained according to quadratic residue function \((5)\)

\[
k_{d_2} = (k_{uc2} + ml_2)^2 \equiv 0 \pmod{n}
\]  \hspace{1cm} (6)

3. \(P\) and \(q\) are utilized, a root \(r\) diverse from \((k_{uc2} + ml_2)\) is obtained from \(k_{d_2}\), \(k_{uc1} = r - ml_2 \pmod{n}\) is obtained, which meets the following conditions.

\[
k_{d_2} \equiv (k_{uc1} + ml_2)^2 \pmod{n}
\]  \hspace{1cm} (7)

4. Multidirectional lock \(ml_1\) of the cipher text is generated by chaos method, \((ml_1 - ml_2)\) is relatively prime with \(n\), and the calculation is shown as formula \((8)\).

\[
k_{d_1} = (k_{uc1} + ml_1)^2 \equiv 0 \pmod{n}
\]  \hspace{1cm} (8)

Entire household keys \((k_{uc1}, k_{uc2}, ml_1, ml_2, k_{d_1}, k_{d_2})\) required for encrypting a database is obtained accordingly. Formulas \((6), (7), \text{ and } (8)\) respectively correspond to conditions \((3), (2), \text{ and } (1)\).

5. Conditional \((4)\) formula is verified. \((6)\)–\((8)\) Formulas show that

\[
\begin{align*}
(k_{uc2} + ml_1)^2 & \equiv k_{uc2}^2 + 2k_{uc2} \cdot ml_1 + ml_1^2 \\
& \equiv k_{d_1} + 2(k_{uc2} \cdot ml_2) \pmod{n}
\end{align*}
\]

It is assumed that \((k_{uc2} + ml_1) \equiv k_{d_1} \pmod{n}\), then \(n | 2(k_{uc2} - k_{uc1}) (ml_1 - ml_2)\). Process \((4)\) shows that \(n\) and \((ml_1 - ml_2)\) are relatively primes, \(n \mid (k_{uc2} - k_{uc_1})\), \(k_{uc2} + ml_2 = k_{uc_1} + ml_2 \pmod{n}\) accordingly, and it is contradictory to the follows: \((k_{uc2} + ml_2)\) and \((k_{uc1} + ml_2)\) are two diverse roots of \(k_{d_2}\) in process \((3)\). Therefore:

\[
\begin{align*}
(k_{uc2} + ml_1)^2 & \equiv k_{d_1} \pmod{n}
\end{align*}
\]

Namely, conditional \((6)\) formula is established.

3.2.2. Security of user class key. since there are only database administrator key \(k_{u1}\) and encrypted data class key \(k_{d1}\) which are calculated through quadratic residue function \(f(x, y)\) in user keys of database encryption system, other keys are generated by chaos method (the security has been discussed in the former text), the security of \(k_{uc1}\) or \(k_{d1}\) should be only investigated accordingly, namely whether general visitor can obtain \(k_{uc1}\) or \(k_{d1}\), through own known key and multidirectional lock\((k_{uc2}, ml_2, ml_1, k_{d_2})\).
Obviously, the security of \( k_{uc1} \) and \( k_{d1} \) depends on the security of quadratic residue function, while the security is guaranteed according to the following theorem.

**Theorem:** if it is difficult to decompose \( n, k_{uc1} \) and \( k_{d1} \) are secure.

**Proof:** it is assumed that they are not secure, general visitor \( u_2 \) can calculate \( k_{d1} \) through a group of visible parameters \( ( k_{uc2}, ml2, ml1, k_{d2}), k_{u1} \) and other parameters meet (6) – (9) formulas, therefore:

\[
(k_{uc2} + ml1)^2 = kd1 + (k_{uc2} - k_{u1}) (ml1 - ml2) \pmod{n}
\]

\( U_{C2} \) has known \( k_{uc2}, ml1 \) and \( k_{d1} \), the follows can be obtained:

\[
2(k_{uc2} - k_{u1}) (ml1 - ml2) = (k_{uc2} + ml1)^2 - kd1 \pmod{n}
\]

Since \( (k_{uc1} + ml2) \) and \( (k_{uc2} + ml2) \) are diverse roots, \( (k_{uc2} - k_{uc1}, n) = p \) or \( q \) according to number theory results. In addition \( (ml1 - ml2) \) and \( n \) are relatively prime, \( 2 (k_{uc2} - k_{uc1}) (ml1 - ml2), n) = p \) or \( q \) accordingly, \( uc2 \) can decompose \( n \) itself accordingly, and it is contradictory to difficult decomposition of \( n \) in the supposition. Therefore, \( k_{uc1} \) and \( k_{d1} \) are secure [2, 3].

### 3.3. Generation of working key

Working key is used for data encryption and decryption in database, and it is generated through master key which encrypting working key information. Working key information is composed of recorded information (such as record name) and list information (such as field attribute). Different records have different record names. Field attribute information of each list is also different, thereby ensuring that each data (field value) has different information, and guaranteeing that each two working keys are diverse mutually. Each two working keys are diverse mutually, OTP encryption mode with the highest security coefficient is adopted for encrypting data, data security is sufficiently guaranteed in the aspect of encryption; working key diversity and dynamics guarantee the security of working key (unless master key is stolen or disclosed).

### 4. Key storage

Safe storage of the key is a very headachy problem. Different storage methods are adopted for different keys in order to ensure the security of key and facilitate the use of legitimate users. Working key is generated dynamically, and it is unnecessary to consider its storage.

#### 4.1. Master key storage

Master key storage is an important problem affecting the encrypted data security, since all encrypted data will be leaked once the key is disclosed or stolen. In addition, the master key change is also a difficult problem because all encrypted data must be decrypted firstly in order to replace the master key. Since data size for encryption or decryption is quite large, quite a long time is required, and it is intolerable for the user. The master key is encrypted by the user class key, and then it is stored in the encryption master key table in view of these two reasons. The user class key is further encrypted by user private key, which is stored in the encryption user class table. Therefore, users can not directly operate the encryption master key table. It is always unable to obtain the master key, thereby effectively preventing the key from being disclosed. Since two-level encryption protection is implemented on the master key here, it is almost impossible to break the master key. Therefore, it is unnecessary to replace the master key, thereby preventing huge trouble from the master key change. In addition, the user class key should be changed regularly in order to guarantee the security of the master key. Similarly, the user key should also be changed regularly in order to ensure the security of the user class key if the user is changed.

#### 4.2. Storage of user private key

User private key should be long enough, and it is generated based on chaos theory itself in order to prevent the dictionary attack. The key is irregular. It is impossible for users to remember the key. Therefore, we divide the user private key into two parts. Most former digits and user ID are stored in the smart card. The later digits should be remembered by the user as password. Users should insert the smart card in the card reader during access to the database. Password is input, and the key information
in the card is read out by the system. The information and password should be connected in series to form user private key [4].

5. Key transformation

5.1. Table of key transformation

Database encryption system can obtain a master key through key transformation table by the user key, the above-mentioned encryption user class key table and encryption master key table belong to key transformation table, which are respectively expressed with $T_1$ and $T_2$.

$T_1$ is a two dimensional table, $T_1 = t_1[i, m, 1 \ldots 3]$, wherein:

$t_1[i, 1] = E'(k_u, k_w) \quad j = 1$ or 2

$t_1[i, 2] = j$ User class ID, $j = 1$ represents database administrator, and $j = 2$ represents general visitor.

$t_1[i, 3] = E'(k_{ui} j, k_{ui})$ Is used for identity authentication.

$t_1[i, 3] = E'(k_{uj}, k_{ui})$ is the result of user class key $k_{ui}$ for user key $k_{ui}$. User class key $k_{ui}$ is obtained through $D'(k_{ui}, t_1[i, 1])$ firstly during user login. Then, $k_{uj}$ is used for decryption $D'(k_{ui}, t_1[i, 3])$, $k_{ui}$ is obtained, if $k'_{ui} = k_{ui}$, it is verified that user ID is correct, otherwise the user is refused [5].

$T_2$ refers to two-dimensional table, $T_2 = t_2[1 \ldots 2, 1 \ldots 2]$, wherein:

$t_2[i, j] = E'(k_{ui}, k_{uj})$ $d_k D_k$

5.2. Key transformation

After user inputs own key $k_{ui}$ and ID $i$, the identity is verified through the system, and the key is transformed if the identity is effective. Key transformation process is shown as follows:

1. User key $k_{ui}$ is used in the system for decrypting encrypted user class key $t_1[i, 1]$ in user class key table $T_1$, user class key $k_{ui}$ is obtained, and user class ID is achieved, namely:

$a \leftarrow D'(k_{ui}, t_1[i, 1]), \quad b \leftarrow t_1[i, 2]$

2. Encrypted master key $t_2[b, j]$ is decrypted by the user class key $an$ in the system, thereby obtaining master key, namely

$c \leftarrow D'(a, t_2[b, j])$

General visitor only can obtain the clear data master key. He can obtain clear data master key on the one hand, and achieve encrypted data master key on the other hand aiming at database administrator.

6. Key transformation

The above discussion shows that user key and user class key should be changed regularly in order to guarantee the security of the encrypted data. Key change is easy according to the above-mentioned key storage and key transformation plan.

6.1. User key change

Users only need to provide the original user key and the new user key to the system in order to change the user key. The system can calculate the user class key of the user based on the original user key. Then, new user key is further used for user class key encryption. Then, the encrypted user class key is filled into table $T_1$, thereby changing the user key.

6.2. User class key change
The system automatically generates a new user class key $k_{ucj}$ during change of user class key, according to above-mentioned user class key generation method, and all user keys of the category can be calculated according to originally user class key $k_{ucj}'$ and ID $i$:

$$k_{uj} = D'(k_{ucj}, t1[i, 3]) \quad u_i \in ucj$$

Table $T_1$ and table $T_2$ are changed as follows:

$$t1[i, 1] = E'(D'(k_{ucj}, t1[i, 3]), k_{ucj}') \quad u_i \in ucj$$

$$t1[i, 3] = E'(k_{ucj}', D'(k_{ucj}, t1[i, 3]))$$

$$t2[j, k] = E'(k_{ucj}', D'(k_{ucj}, t2[j, k]))$$

It is obvious from the above change process that the change process is very independent aiming at both user key change or user class key change. Other user or user class cannot be affected. More importantly, the change work is completed by the system in the whole change process. Users always cannot obtain user class key or master key, thereby ensuring the security of user class key and master key during key change.

7. Conclusion

How to effectively protect the master key in the key management scheme proposed in article [1] is mainly discussed. Discussion shows that the key management scheme proposed in article [1] and two-level transformation table key management plan proposed in articles [2, 3 and 5] are organically combined for fully exerting respective advantages and overcoming respective shortcomings well. The security of the master key can be effectively protected. The data encryption strength can be fully guaranteed. In addition, we also provide the key management scheme of the database encryption system on the basis of in-depth discussion of key generation, distribution, storage, transformation, change, etc. The influence of all links in the scheme on database encryption system performance and encrypted data security is analyzed. The analysis results show that the scheme is practical as the key management scheme of database encryption system.

References

[1] Zhu Luhua, Chen Rongliang. Design and implementation of database encryption system. Computer Engineering, 2002, 28 (8): 61 - 63.

[2] Song Yu, Zhao Wenqing. Application research of key management in management information system. Computer Engineering and Application, 1999, (10): 95 - 97.

[3] Chen Wei. Database encryption key allocation and management technology. Journal of Tsinghua University (natural science edition), 1994, 34 (1): 99 - 103.

[4] Bruce Schneier. Applied Cryptography Second Edition: protocols, algorithms, and source in. Beijing: China Machine Press, 2000.

[5] Shang Jie, Dai Yiqi, Li Xiangyang. Encrypted database and its key management. Computer Application Research, 1996, (3): 98 - 100.