THE DIGITAL WATERMARKING TECHNIQUE FOR NUMERICAL RELATIONAL DATABASES

Sonupriya P S, Rani M R
MCA Scholar, Dept of CSE, MCET, Pathanamthitta, Kerala
Asst Professor, Dept of CSE, MCET, Pathanamthitta, Kerala

ABSTRACT: Data is an important asset of any application and large amount of data are usually stored as database tables. The need for digital watermarking and encryption of the relational database is to ensure its integrity, authentication and confidentiality. In this paper, a new watermark embedding and de-embedding algorithm is proposed for secure insertion of a robust imperceptible watermark in relational database. The proposed Digital Watermarking technique is used to provide integrity to the database table and in order to provide confidentiality and authentication RSA technique is used. The proposed system can be divided into two phases: watermark embedding and de-embedding phase. The first step in watermark embedding phase is to partition the database table and assign the partition number to each tuple of the relation using Cryptographic Hashing function. Then the randomly generated watermark key is embedded in selected tuples of the selected partitions of the database table. After watermark embedding, the entire database table is encrypted. At the receiver side, Watermark de-embedding phase consists of the decryption of the watermarked table, watermark de-embedding and integrity verification process. The watermark key is de-embedded from each attributes of the tuples of the selected partitions. The proposed technique also handles the attack on database like tuple deletion, alteration and insertion.

Keywords: Digital watermarking, data integrity, confidentiality, authentication, relational database watermarking, data partitioning, watermark embedding and de-embedding, encryption and decryption.

1. INTRODUCTION

Security of a database is a crucial issue today. When database is shared, it is necessary to ensure its integrity, confidentiality and authentication at the receiver side. The proposed system is used to provide integrity to the numerical relational database by means of Digital Watermarking technique. In order to provide confidentiality and authentication RSA encryption technique is used. New
watermark embedding and de-embedding algorithm is proposed, which ensure no data modification by the insertion of watermark key in the attributes of the selected tuples of the selected partitions of the database table. At the sender side, data partitioning algorithm is used to partition the database table to choose partitions in which watermark key is embedded. The next step is to embed the randomly generated watermark key in each attribute of the selected partitions. After this, the entire database table is encrypted using RSA algorithm to provide confidentiality. The secret data such as watermark key, secret key used for data partitioning etc is made available at the receiver side also along with the encrypted watermarked database. Then at the receiver side, the secret data is used for checking integrity of the database table. The received database table is decrypted and submits for integrity verification process using watermark de-embedding algorithm.

In existing system, several techniques are proposed to provide integrity, but they sometimes cause serious alterations to the original data and have no means to provide confidentiality and authentication. In proposed system integrity, confidentiality and authentication is provided without modification to the original data. The proposed system has application in military, medical field and large organizations where huge amount of confidential databases are exchanged. Only authenticated users, who have secret data, can see the original contents of the database. The receiver can also ensure that database is originated from the specified source and there is no alteration or modification in the data received. If there is any insertion, deletion or modification of the data during transmission, it is detected at the receiver side during integrity verification process. So the objectives of the proposed system are

- To provide integrity to the database table.
- To provide confidentiality and authentication to the contents of the database table.

2. EXISTING SYSTEM

In existing there is no way to provide integrity, confidentiality and authentication together in one system. Arti Deshpande and Jayant Gadge [1] proposed a single bit encoding algorithm which changes the least significant bits of the attributes of the selected partitions of the database table. But this may sometimes cause
undesirable changes to the original data. There is no way to restore the original data values. In the case of confidential data, these little changes may result in unexpected results. M. Kamran, Sabah Suhail, and Muddassar Farooq [2] proposed techniques to resist data modification during watermarking of relational database, but does not gave complete guarantee against such alterations. They do not propose any technique to provide confidentiality and authentication to the database table along with integrity preservation.

3. PROPOSED SYSTEM

Ensuring integrity and confidentiality of relational databases is a crucial issue in today’s internet-based application environments and in many content distribution applications. In the proposed system, new mechanism is proposed for ensuring integrity based on the secure embedding of a robust imperceptible watermark in relational data. The first phase is to partition the original data and assign partition number to each and every tuple of the relation using Cryptographic Hashing Function. In the second phase, the randomly generated watermark key is embedded in each attribute of the selected tuples of the partitions. In the third phase, after inserting the watermark in the partition, merge all partitions and get the complete watermarked data. Watermarking is the piece of data securely embedded and this is said to be imperceptible. Imperceptible embedding means that the presence of the watermark is unnoticeable in the data. The technical side of the proposed system can be divided into two phase - Watermark Embedding Phase and Watermark De-embedding phase. Watermark embedding phase consists of Data Partitioning, watermark embedding and encryption of the entire the watermarked table using RSA. Watermark de-embedding phase consists of the decryption of the watermarked table, watermark de-embedding and integrity verification process. The flow chart of the proposed system is shown in the Figure 1.

Digital watermarking is to ensure data integrity, the assurance that data received are exactly as sent by an authorized entity i.e. contain no modification and insertion. RSA encryption/decryption algorithm is used to ensure authentication and confidentiality. Data Confidentiality means protection of data from unauthorized disclosure. The function of the
authentication service is to assure that it claims to be from.

![Figure 1: Flow Chart of the Proposed System](image)

**4. ALGORITHMS**

The following algorithms are used in the proposed system.

**4.1 Data Partitioning Algorithm**

The Cryptographic hash function Message Digest 5 (MD5) is used to partition the database table. The randomly generated secret key of the database table is used for data partitioning. The data set R is a database relation with scheme R (P, D0, D1……Dn-1) where P is the primary key attribute, D0, D1……Dn-1 are n attributes which are candidates for watermarking, and |R| is the number of tuples in R. The data set R is to be partitioned into m non overlapping partitions, namely, {S0; . . . ; Sm-1}. For each tuple r ∈ R, the data partitioning algorithm computes a MAC (Message Authentication Code), which is considered to be secure cryptographic hash function as given below:

\[ H(ks \ | \ H(r.P || Ks)) \]

where r.P is the primary key of the tuple r, H( ) is a secure hash function, and || is the concatenation operator. Using the computed MAC tuples are assigned to partitions. For a tuple r, its partition assignment is given by \( \text{Partition}(r) = H(Ks \ | \ H(r.p || Ks)) \mod m \). A cryptographic hash function is a transformation that takes an input (or 'message') and returns a fixed-size string,
which is called the hash value (sometimes termed a message digest, a digital fingerprint, a digest or a checksum). Assign the partition to each row using \( \text{Partition}(r) = H(K_s \parallel H(r.p \parallel K_s)) \mod m \).

An attacker cannot predict the tuple to partition assignment without the knowledge of the secret key \( K_s \) and the number of partitions \( m \), which are kept secret. However, keeping it secret makes it harder for the attacker to regenerate.

**Algorithm:**

Input: Data Set \( R \), Secret Key \( K_s \), Number of partitions \( m \)

1. \( \{S_0, S_1 \ldots \ldots S_{m-1}\} \rightarrow \{\} \)

2. For each tuple \( r \in R \)

3. \( \text{Partition}(r) = H(K_s \parallel H(r.p \parallel K_s)) \mod m \).

4. Insert \( r \) into \( S \) partition(r)

5. Return \( S_0, S_1 \ldots \ldots S_{m-1} \).

Output: Data Partitions \( S_0, S_1 \ldots \ldots S_{m-1} \)

**4.2 Watermark Embedding Algorithm**

After partition the database table using data partitioning algorithm, the next step is to embed the randomly generated watermark key to each attributes of the tuples of the selected partitions.

Input: Partitioned Data set \( S=\{s_0, s_1 \ldots \ldots s_{m-1}\} \), watermark key \( k_w \)

1. Select a partition \( s_k \) from \( S \).

2. For each tuple \( t_j \) of \( s_k \)
   2.1 \( \text{mark}(r) = H(k_s \parallel r.p) \mod 2 \)
   2.2 if \( \text{mark}(r) = 0 \) then
      Select that tuple for watermark.
      \( s_k^{w} = \text{embed\_watermark}(t_j, k_w) \).

3. Insert \( s_k^{w} \) into \( D_w \).

Output: watermarked data set \( D_w \).

Procedure: \( \text{embed\_watermark}(t_j: \text{tuple} \ j \text{ of} \ s_k, k_w) \)

1. For each attribute \( a_k \) in \( t_j \)
   append watermark key \( k_w \) with \( a_k \).

2. Return watermarked tuple \( t_j^k \).

**4.3 Watermark De-embedding Algorithm**

Watermark De-embedding algorithm is used at the receiver side to check the integrity, of the received data from sender. Received database table is partitioned using the data partitioning algorithm. The secret data such as secret key used for data partitioning, watermark key, the
partitions selected for watermarking etc used at the sender side is made available at the receiver side also. After partitioning the table, the next phase is to de-embed the watermark key from each attributes of the tuples of the selected partitions. This de-embedded watermark key is compared with the watermark key obtained from the sender for calculating the integrity status. If integrity status was hundred percentage, the receiver accepts the received data. This means that there is no alteration or modification takes place in the received data. Otherwise the received data is not accepted.

Input : Watermarked Data Set Dw, Secret key Ks, number of partition m, watermark key k_w

1. \{S0,S1……Sm-1\} = get_partition (Dw, Ks, m)

// apply data partitioning algorithm on watermarked data

2. Select watermarked partition sk.

3. success = 0.

4. fail = 0.

5. For each attribute  a_k  in tuple tj

5.1 remove the watermark key K_w' from a_k .

5.2 if  K_w' = k_w

success=success + 1;

else

fail = fail+1;

6. Compute total=success+ fail

8. Compute per = (success/ total) * 100

9. If per =100% then

Integrity is preserved.

Else

Integrity is not preserved.

4.4 RSA Algorithm

Asymmetric encryption transforms plaintext into Ciphertext using one of the key and an encryption algorithm. Using the paired key and a decryption algorithm, the plaintext is recovered from the Ciphertext. Asymmetric encryption can be used for confidentiality, authentication or both. The most widely used public key cryptosystem is RSA. The difficulty of attacking RSA is based on the difficulty of finding the prime factors of a composite number.

4.4.1 Key Generation

1. selecting two large primes at random: p, q

2. Compute n = p. q

3. Calculate \( \phi(n) = (p-1)(q-1) \)
4. selecting at random the encryption key $e$

Where $1 < e < \phi(n)$, $\gcd(e, \phi(n)) = 1$

5. solve following equation to find decryption key $d$

$e \cdot d \equiv 1 \pmod{\phi(n)}$ and $0 \leq d \leq n$

6. publish public encryption key: $PU = \{e, n\}$

7. keep secret decryption key: $PR = \{d, n\}$

### 4.4.2 RSA Encryption/Decryption

Public key of the receiver is used for the encryption of the database table at the sender side after watermark embedding process. Let $M$ be the plaintext, $M < n$ and public key $\{e, n\}$ is used for generating the corresponding Ciphertext as follows:

| Plaintext $M$ | Ciphertext $C = M^e \pmod{n}$ |

Private Key of the receiver is used for decryption of the received database table at the receiver side. Let $C$ be the Ciphertext and $\{d, n\}$ be the private key, and then the corresponding plaintext can be obtained as follows:

| Ciphertext $C$ | Plaintext $M = C^d \pmod{n}$ |

### 5. FUTURE ENHANCEMENTS

The proposed system is applicable to numerical database for providing integrity, authentication and confidentiality. It is expected that the proposed system is applicable in any field when consider non-numerical database and its conversion to numerical data.

### 6. CONCLUSION

In existing system, database table is send directly without using digital watermarking and encryption technique. In the proposed system, new mechanism is used for ensuring integrity based on the secure embedding of a robust imperceptible watermark in relational data. In order to provide confidentiality and authentication RSA Encryption is used. At the sender side, before sending database table, data partitioning algorithm is used to select the partitions in which watermark key is embedded. Watermark key is embedded in each attributes of the tuples of the selected partitions of the database table. After watermark embedding, the entire table is encrypted using RSA encryption algorithm. The watermark key, the secret key and other
secret data is sent to the receiver along with the database table. At the receiver side, the received location table is decrypted first. Then the next is integrity checking process. The watermark key is extracted from each attribute of the tuples of the selected partitions. The extracted watermark key is compared with the one obtained from the sender at the receiver side. The received database tables are accepted only if integrity is preserved, otherwise it is rejected.

7. REFERENCES


