International Journal of Database Theory and Application
Vol.8, No.1 (2015), pp.23-28
http://dx.doi.org/10.14257/ijdta.2015.8.1.03

Tracing Mobile Database with Complex Watermark

Hequn Xian[1,2], Jing Li[1], and Xiuqing Lu[1]

[1]College of Information Engineering, Qingdao University
[2]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences
xianhq@126.com

Abstract

Relational database can be replicated and distributed in the mobile computing scenario. To protect the ownership of the data and to trace the trail of data distribution, we devise a complex database watermark scheme. In each data distribution process, the sender and the receiver agree on a watermark key pair, and a watermark is embedded into the data. In case of piracy or misusing, the sender can claim the ownership of the controversial data copy and accuse the receiver for his improper behavior. On the other hand, an innocent receiver can prove his innocence to arbitral authorities with watermark detection. Experiment results show that the scheme has a satisfying performance and it is qualified for practical use.

Keywords: *mobile database, database tracing, database watermark, complex watermark*

1. Introduction

Tremendous knowledge and great value reside in databases in our digital world. In the mobile computing environment, databases could be replicated, distributed, and sometimes even purchased as commodities by related parties from the network. Problems arise with the mobility of databases, such as piracy, misusing, or leakage of sensitive data. In this paper, we devise a tracing scheme in order to witness each step of the database's distribution. When a database copy is pirated or misused, the original owner can identify the controversial copy and accuse the responsible party with cryptographic evidence. On the other hand, an innocent defendant should be given the ability to prove his innocence when falsely accused.

Generally speaking, it takes two steps for us to build a tracing scheme for mobile databases that can fulfill the requirement mentioned above. First, we tailor the classic database watermark technique to construct a novel watermark mechanism, which is elaborated in later sections of this paper as the *complex watermark*. Second, we devise a database distributing protocol, which securely witnesses the handing over of a database copy from a sender to a receiver. We implement our scheme, and experiment on it. Results show that the scheme is qualified for practical use.

2. Related Works

Digital watermark has been used for decades to hide information in a certain kind of media [1]. For relational data, watermark schemes assume that the data can tolerate small errors that have no effect on the usability. Values of those least significant bits can be intentionally altered to carry marks which constitute the watermark for the database relation [2]. A secret key is used to control the watermark embedding and detecting process. Sion *et al*. utilized this method to provide a way for a database owner to claim his copyright [3]. Jajodia's group developed a new

ISSN: 2005-4270 IJDTA

watermark technique that can include an arbitrary bit string in the watermark, so that not only the copyright can be claimed but also the pirate can be identified [4]. There are other research works on watermark technique and its applications, most of which focus on detailed watermark mechanism or transformation from various kinds of information to database watermarks. Yang *et al.*, found a way of watermark design based on Steiner triple systems [5]. Bhattacharya *et al.*, devised a distortion-free authentication watermark [6]. Khanduja *et al.*, adopted bacterial foraging algorithm to construct a database watermark [7]. There are many other related works on data protection in the context of mobile computing. Lu *et al.*, proposed a framework for dynamic data protection, expecting to balance the security and resource consumption in mobile cloud computing environment [8]. Dinh *et al.*, analyzed security challenges as well as other issues in the field of mobile cloud computing [9]. In [10], Khan *et al.*, addressed the problem of data integrity, privacy preserving and secure storage in mobile computing environment. In those copyright protection systems, only the data owner's interest is protected. It is technically possible for a malicious owner to falsely watermark a data copy and accuse some innocent party for misusing or pirating. We solve this problem with our complex watermark scheme.

3. Complex Watermark

A complex watermark inherits the classic database watermark's basis principles, the difference is that the embedding and detection procedures involve two keys, and that most watermarked tuples have a pair of altered least significant bits.

Several parameters should be explained in order to understand the algorithms. The parameter γ is used to determine the percentage of marked tuples. Not every tuple carries the watermark, only $1/\gamma$ of them do. The hash function is a standard cryptographic routine which is supposed to uniformly map an arbitrary value into a binary value with fixed length. The parameter ξ stands for the number of least significant bits in each attribute. The following algorithm shows how a watermark is embedded into a relation with K1 and K2 as the controlling secrete keys.

Algorithm Embedding(key K1, key K2,relation R)

```
FOR EACH tuple t∈R
    IF (H(t.PrimaryKey | K1) mod γ = = 0)
    {
        int attIdxM = H(t.PrimaryKey | K1) mod υ;
        IF isnull(t. Attribute[attIdxM])
            Continue;
        int bitIdxM = H(t.PrimaryKey | K1) mod ξ;
        IF (H(t.PrimaryKey | K1) is an even number)
            Set the bitIdxMth bit of the bitIdxMth attribute of t to 0;
        ELSE
            Set the bitIdxMth  bit of the bitIdxMth  attribute of t to 1;
        int attIdxS = H(t.PrimaryKey | K2) mod υ;
        IF isnull(t. Attribute[attIdxS])
            continue
        int bitIdxS= H(t.PrimaryKey | K2) mod ξ;
        IF ((attIdxM = = attIdxS) && (bitIdxM = = bitIdxS)
            Continue;
        IF (H(t.PrimaryKey | K2) is an even number)
            Set the bitIdxSth bit of the bitIdxSth attribute of t to 0;
        ELSE
            Set the bitIdxSth  bit of the bitIdxSth  attribute of t to 1;
    }
```

As we can see from the algorithm *Embedding*, null value attributes don't carry watermark bits, and watermark bits in the same tuple never overlap. The watermark can be recognized to have two parts, one bound by K1 and the other with strong relation with K2. The two parts of a complex watermark can be detected respectively with the same keys used in the embedding process. The following two algorithms can be used in detecting each part of the complex watermark.

Algorithm Detect1 (key K1, relation R, int &ckN,int &hitN)

```
kN = hitN = 0;
FOR EACH tuple t∈R
IF (H(t.PrimaryKey | K1) mod γ = = 0)
  {
      int attIdxM = H(t.PrimaryKey | K1) mod υ;
      IF isnull(t. Attribute[attIdxM])
           Continue;
      ckN ++;
      int bitIdxM = H(t.PrimaryKey | K1) mod ξ;
      int bitValue = the bitIdxMth bit of the attIdxMth attribute of t;
      IF (H(t.PrimaryKey | K1) mod 2 = = bitValue)
          hitN ++;
  }
```

Algorithm Detect2 (key K1, key K2, relation R, int &ckN,int &hitN)

```
ckN = hitN = 0;
FOR EACH tuple t∈R
IF (H(t.PrimaryKey | K1) mod γ = = 0)
  {
      int attIdxM = H(t.PrimaryKey | K1) mod υ;
      IF isnull(t. Attribute[attIdxM])
           Continue;
      int attIdxS = H(t.PrimaryKey | K2) mod υ;
      IF isnull(t. Attribute[attIdxS])
           Continue;
      int bitIdxM = H(t.PrimaryKey | K1) mod ξ;
      int bitIdxS = H(t.PrimaryKey | K2) mod ξ;
      IF ((attIdxM = = attIdxS) && (bitIdxM = = bitIdxS)
        Continue;
      ckN ++;
      int bitValue = the bitIdxSth bit of the attIdxSth attribute of t;
      IF (H(t.PrimaryKey | K2) mod 2 = = bitValue)
          hitN ++;
  }
```

Both detection algorithms output variable *chN* and *hitN*, which indicate how many tuples are checked for watermark bits and how may watermark bits are discovered in these checked tuples. For an irrelevant copy, the probability of *hitN* out of *chN* bit positions match the expected watermark value by sheer chance abides by binomial probability distribution.

$$\Pr ob[hitN \ \ out \ \ of \ \ chN \ \ mathes] = binomial \ (hitN \ ; chN \ , 1/2)$$

So, if we observe a hit count with a probability that is too slim, we can safely deduce that the data are actually watermarked with the expected key. We don't elaborate on the probabilistic analysis here, which can be found in chapters of most text books on probability. With the complex watermark method, we develop a data transferring protocol, which should be followed each time a sender gives his database to a receiver.

4. Data Transferring Protocol

The data transferring is initiated by the receiver. A trusted third party (TTP) is needed to carry out the watermark embedding, to generated keys, and to function as a mediator. However, the TTP does not store any data or keys, which reduces the security risks.

Protocol DataTransfer

1) The receiver sends his public key PK-R to the sender
2) The sender forwards PK-R to TTP together with the original database
3) The TTP randomly chooses K1 and K2, embeds a watermark into the database using K1 and K2
4) The TTP sends H(K1) and H(K2) to the sender and the receiver
5) The TTP encrypts K1with the sender's public key and sends it to the sender
6) The TTP encrypts K2with the receiver's public key and sends it to the receiver
7) The sender and the receiver sign the concatenation of the hash values H(K1)| H(K2) respectively with their private key.
8) The sender sends his signature in step 7 to the receiver
9) The receiver sends his signature in step 7 to the TTP
10) The TTP sends the watermarked database to the receiver
11) The TTP forwards the receiver's signature to the sender

As shown in the protocol above, both the sender and the receiver receive signatures from each other and keep them. The sender holds K1, which can be used to carry out watermark detection as described in algorithm *Detect1* on a suspected database copy. A positive result will prove the existence of one part of the complex watermark, so that ownership can be claimed and the receiver is accused. On the other hand, if the receiver is wronged, he can use K2 to demonstrate the non-existence of the other part of the claimed complex watermark. The signatures from the sender and the receiver constitute the agreement of a database distribution between the two parties.

Digital signature operations and hash functions are common cryptographic routines that can be easily abstained from any PKI system, which we assume to be secure.

5. Experiment

We analyze the computational cost of watermark embedding and detecting by carrying out a set of experiments. The database is hosted in Microsoft SQL Server 2008 on a workstation with Intel CORE i3 processor and 4GB DDR3 memory. We populate the database with TPC-H benchmark data [11], 8 attributes in table *lineitem* are candidate watermarking attributes. Parameter γ and ξ are set to be 3. The watermark embedding time is compared with common database reading and writing operations. The watermark detection time is compared with reading operations. The results are shown in Table 1.

Calculated from the direct result, the average overhead is 7.17% for watermark detecting, and 11.29% for watermark embedding, which is fairly satisfying for practical applications.

Table 1. Execution Time Comparison (measured in millisecond)

tuples	Read	read-write	detecting	embedding
600	30	6988	36	7189
6000	141	141390	160	159625
20000	270	509102	300	560810
60000	413	943438	479	1009782
200000	990	1601090	1010	1698022
600000	3328	2508732	3499	2609557

6. Conclusions

In this paper, we propose a complex database watermark scheme to trace mobile databases. A key pair is used to determine the database watermark, and a data transferring protocol is devised for database redistribution processes. Cryptographic evidence is generated and given to the corresponding parties, so that both the sender and the receiver are fairly protected. Experiment shows that the proposed scheme is qualified for practical use.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (61303197) and the Natural Science Foundation of Shandong province (ZR2013FQ005).

References

[1] S. Khanna, F and Zane, "Watermarking maps: hiding information in structured data", Proceedings of the Int'1 Conf. SODA2000, San Francisco, California, USA, **(2000)**, pp. 596-605.

[2] R. Agrawal, P. Haas and J. Kiernan, "Watermarking relational data: framework, algorithms and analysis", The VLDB Journal, vol. 12, **(2003)**, pp. 157-169.

[3] R. Sion, M. J. Atallah and S. Prabhakar, "Rights protection for relational data", IEEE Trans. Knowledge and Data Engineering, vol. 16, **(2004)**, pp. 1509-1525.

[4] Y. Li, V. Swarup and S. Jajodia, "Fingerprinting Relational Databases: Schemes and Specialties", IEEE Trans, Dependable and Secure Computing, vol. 2, no. 1, **(2005)**, pp. 34-45.

[5] Z.-F. Yang, S.-S. Chiou and J.-T. Lee, "Watermark design based on Steiner triple systems", Multimedia Tools and Applications, Springer, US, **(2013)**.

[6] S. Bhattacharya and A. Cortesi, "Distortion-Free Authentication Watermarking", Software and Data Technologies Communications in Computer and Information Science, vol. 170, **(2013)**, pp. 205-219.

[7] V. Khanduja, O. Prakash Verma and S. Chakraverty, "Watermarking relational databases using bacterial foraging algorithm", Multimedia Tools and Applications, **(2013)**.

[8] H. Lu, X. Xia and X. Wang, "How to Dynamically Protect Data in Mobile Cloud Computing", LNCS - 7719, **(2013)**, pp. 364-371.

[9] H. T. Dinh, C. Lee, D. Niyato and P. Wang, "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches", Wireless Communications and Mobile Computing, vol. 13, no. 18, **(2013)**, pp. 1587-1611.

[10] A. Nasir Khan, M. L. Mat Kiah, S. U. Khan and S. A. Madani, "Towards Secure Mobile Cloud Computing: A Survey", Future Generation Computer Systems, vol. 29, no. 5, **(2013)**, pp. 1278-1299.

[11] TPC-H Benchmark Specification, Available at: http://www.tpc.org, **(2008)**.

Author

Hequn Xian, born in 1979, associate professor and Ph. D.. His main research interests include mobile computing, network and information security.