REVERSIBLE AND SEMI-BLIND RELATIONAL DATABASE WATERMARKING

Gaurav Gupta and Josef Pieprzyk
Centre for Advanced Computing - Algorithms and Cryptography
Department of Computing, Division of Information and Communication Sciences
Macquarie University, Sydney, NSW - 2109, Australia

Keywords: Reversible, relational database, watermarking, copyright, ownership.

Abstract: In 2002, Agrawal and Kiernan proposed a relational database watermarking scheme that modifies least significant bits (LSBs) of numerical attributes selected using a secret key. The scheme does not address query preservation (some queries give different results when executed on the original and watermarked relation). Additive and secondary watermarking attacks on the watermarked relation are also possible. Such attacks can render the original watermark undetectable. Hence, an attacker who embeds his watermark in a previously watermarked relation can claim ownership of that relation. However, if the scheme is reversible, then a previous watermark, if any, can be detected in the reversed relation. In this paper, we propose an enhanced reversible, semi-blind and query-preserving watermarking scheme. Using this scheme, the correct owner of a relation can be identified even if the relation has been watermarked by multiple parties. If required, the database can be restored to it's original state too. This finds applications in high-precision settings such as military operations or scientific experiments.

1 INTRODUCTION

Watermarking (embedding owner information in a multimedia object) and fingerprinting (embedding buyer information) have received significant research attention in the past decade. Since a multimedia object is sold (usually) by one owner to multiple users, each copy has the same *watermark* but a different *fingerprint*. Images, audio-visual and text documents, software, and databases are general multimedia objects considered for watermarking. Images have been the primary focus of research in watermarking (Bors and Pitas, 1996; Braudaway, 1997; Cox et al., 1995; Cox et al., 1997; Cox et al., 1996) due to two main reasons. First, the human visual system (HVS) cannot distinguish between images with minor differences, and second, image pixels' LSBs can be flipped without causing significant distortion in the visual quality.

Database watermarking is a relatively new field which has seen contributions in the last five years. A typical scenario requiring database watermarking is when a company $\mathcal{C}$ provides confidential customer data to an external organization $\mathcal{O}$ (eg. call center). To ensure that $\mathcal{O}$ does not exploit the information and doesn't sell it, $\mathcal{C}$ embeds it's watermark in the database relation. Another application is in web services, where data provider $\mathcal{D}$ makes a database relation available online for remote query. An attacker may try to steal the relation using multiple "intelligent" queries. To prevent this, $\mathcal{D}$ watermarks the databases before making them available online using a blind or non-blind watermarking scheme. In a blind watermarking model, only the watermarked media and a secret key are required to detect/extract watermark whereas in a non-blind watermarking scheme, the unmarked multimedia object is also required in addition to the watermarked copy and secret key. This creates a situation where one needs to store the unmarked object at a secondary secure location. In this paper, we present a database watermarking scheme that is reversible and semi-blind. We call the scheme semi-blind because it does not require the original database to detect watermark but the insertion algorithm stores the original bits selected for modification as an embed trace $\mathcal{ET}$, which is input to the detection algorithm. Size of $\mathcal{ET}$ is proportional to

Gupta G. and Pieprzyk J. (2007).
REVERSIBLE AND SEMI-BLIND RELATIONAL DATABASE WATERMARKING.
In *Proceedings of the Second International Conference on Signal Processing and Multimedia Applications*, pages 279-286
DOI: 10.5220/0002136902790286

the number of tuples being marked. Previous watermarking schemes such as (Sion et al., 2004) have also presented similar semi-blind watermarking models. Our scheme is an enhancement of the irreversible watermarking model proposed by Agrawal and Kiernan (Agrawal and Kiernan, 2002). We show that secondary watermark attacks are feasible on (Agrawal and Kiernan, 2002). Further, we modify the model to eliminate this shortcoming and propose an additional algorithm to achieve identify the rightful owner from n contenders.

1.1 Organization of the Paper

Section 2 describes related work and provides a detailed account of database watermarking from (Agrawal and Kiernan, 2002) to eliminate shortcomings. The modified algorithms are presented in Section 4 and Section 5 contains their analysis. The paper is concluded in Section 6 with a discussion on the added advantages provided by the new scheme.

1.2 Notations Used

The following notations will be used through the paper,

- R : Relation
- r : Tuple
- A_i : i^{th} attribute
- $r.A_i$: i^{th} attribute in tuple r
- A_i^j : j^{th} LSB of i^{th} attribute
- $r.A_i^j$: j^{th} LSB of i^{th} attribute in tuple r
- $r.P$: Primary key of tuple r
- $\circ$: Concatenation
- $\mathcal{H}()$: One-way hash function
- $R \xrightarrow{ins(p)} R_w$: R is watermarked by party p to become relation R_w, or in other words, a watermark is inserted by party p into a document R resulting in its watermarked version R_w
- $R_w \xrightarrow{det(p)} R$: R is restored when party p's watermark is detected in R_w
- $|x|$: Size of x in bits
- $abs(x)$: Absolute value of x
- $\lfloor x \rfloor$: Floor value of x
- *Distance* for attribute $r.A_i$: $\delta_{r.A_i} = min_{\tilde{r} \neq r}\{abs(r.A_i - \tilde{r}.A_i)\}$

2 RELATED WORK AND AGRAWAL-KIERNAN SCHEME

Several relational database watermarking models have been proposed in (Agrawal and Kiernan, 2002; Agrawal et al., 2003; Sion et al., 2004; Gross-Amblard, 2003; Guo et al., 2006; Li and Deng, 2006; Li et al., 2004; Zhang et al., 2004a; Zhang et al., 2006; Zhang et al., 2004b). These schemes are irreversible with the exception of (Zhang et al., 2006) and except (Gross-Amblard, 2003), do not preserve queries. *Irreversible* watermarking implies that the original relation cannot be restored from the watermarked relation. Ownership disputes might be unresolved if an attacker successfully embeds a secondary watermark. But if the watermarking is reversible, the original database can be restored and the correct owner identified using suitable algorithm (discussed in section 4.1).

The notions of *local and global distortions* are presented in (Gross-Amblard, 2003) which achieve the property of query-preservation. Local distortion refers to the minimum difference between values of an attribute. For formal definitions of local and global distortions, please refer to (Gross-Amblard, 2003). In a nutshell, the attributes should only be modified by a value lesser than the local distortion.

Agrawal and Kiernan (Agrawal and Kiernan, 2002) were the first to present a database watermarking scheme that modifies LSBs of numerical attributes (selected using the private key and tuple's primary key value). This *key-based attribute selection* is common to other proposals (Agrawal et al., 2003; Li and Deng, 2006).

2.1 Agrawal-Kiernan Scheme

The watermarking scheme consists of two algorithms; ***Insertion***, and ***Detection***. The bits modified during insertion are checked for correctness in the detection algorithm for establishment of watermark presence. Parameters to the insertion algorithm are,

- Database Relation R containing η tuples and υ modifiable attributes $\{A_0, A_1, \ldots, A_\upsilon\}$
- Number of modifiable LSBs ξ
- Fraction of tuples to be watermarked $1/\gamma$
- Private key $\mathcal{K}$

The secret parameter set is given be $\phi = (\mathcal{K}, \gamma, \upsilon, \xi)$. Algorithm 1 illustrates the watermark insertion process. Tuples are selected using message authentication code (MAC) $\mathcal{F}(r.P)$ defined as $\mathcal{H}(\mathcal{K} \circ \mathcal{H}(\mathcal{K} \circ (r.P))$ (Schneier, 1996) and appropriate bit in the tuple set to $\mathcal{H}(\mathcal{K} \circ r.P)\%2$ till $\omega = \frac{\eta}{\gamma}$

bits are marked. Converse procedure is applied on the watermarked copy to detect the watermark (Algorithm 2) by verifying the modified bit is equal to $\mathcal{H}(\mathcal{K} \circ \tilde{r}_w.P)\%2$. The primary key value is unchanged. Parameters to the watermark detection algorithm are watermarked database relation R_w containing η tuples and υ attributes $\{A_0, A_1, \ldots, A_\upsilon\}$, number of LSBs modified ξ, fraction of tuples watermarked $1/\gamma$, upper bound on probability of falsely detecting watermark α, minimum number of correctly marked attributes for successful detection τ, and private key $\mathcal{K}$.

Input: Relation R, private key K, fraction $\frac{1}{\gamma}$, LSB usage ξ
Output: Watermarked relation R_w

forall *tuple* $r \in R$ **do**
  if $\mathcal{F}(r.P)\%\gamma = 0$ **then**
    $i = \mathcal{F}(r.P)\%\upsilon$;
    $j = \mathcal{F}(r.P)\%\xi$;
    $r.A_i^j = \mathcal{H}(\mathcal{K} \circ r.P)\%2$;
  end
end
return R;

Algorithm 1: Agrawal-Kiernan watermark insertion algorithm.

Input: Watermarked Relation $\tilde{R}_w$, private key K, fraction $\frac{1}{\gamma}$, LSB usage ξ
Output: Detection Status $\in \{true, false\}$

$totalcount = matchcount = 0$;
forall *tuple* $\tilde{r}_w \in \tilde{R}_w$ **do**
  if $\mathcal{F}(r.P)\%\gamma = 0$ **then**
    $i = \mathcal{F}(r.P)\%\upsilon$;
    $j = \mathcal{F}(r.P)\%\xi$;
    if $\tilde{r}_w.A_i^j = \mathcal{H}(\mathcal{K} \circ \tilde{r}_w.P)\%2$ **then**
      $matchcount = matchcount + 1$;
    end
    $totalcount = totalcount + 1$;
  end
end
$\tau = min\{\theta : \mathcal{B}(\theta, totalcount, 1/2) < \alpha\}$; // $\mathcal{B}$ defined in Equation 1
if $matchcount \geq \tau$ **then**
  return $true$;
end
return $false$;

Algorithm 2: Agrawal-Kiernan watermark detection algorithm.

Equation 1 gives the binomial probability of having at least k successes from n trials where probability of success in a single trial is p. During detection, at least τ bits need to be detected correctly in order to extract the correct watermark or in other words the probability of τ out of ω bits matching by sheer chance $B(\tau, \omega, \frac{1}{2})$ should be less than the upper bound of false positive α.

$$\mathcal{B}(k,n,p) = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i} \qquad (1)$$

2.2 Security Provided by Agrawal-Kiernan Scheme

While discussing the security of the scheme, Agrawal and Kiernan consider the following collection of attacks,

A1: Bit attack: Updating some bits in numerical attributes.

A2: Randomization attack: Assigning random values to some bits.

A3: Rounding attack: Rounding off a fixed number of bits.

A4: Translation attack: Transforming numerical values to another data type.

A5: Subset attack: Removing a small subset of tuples/ attributes.

A6: Mix and match attack: Applying A4 on multiple relations and merging them.

A7: Additive attack: Re-watermarking an already watermarked relation.

A8: Invertibility attack: Checking if detection returns true for a random key.

Inserting new tuples to destroy watermark will not succeed as $\mathcal{F}(r.P)$ identifies marked tuple and two tuples cannot have the same primary key. Success of removing the watermark by deleting tuples depends on the parameter γ. Probability of destroying watermark by deleting a few tuples is extremely low when the fraction of tuples marked when γ is high. If γ is high for a fixed n, $1/\gamma$ is low and hence the fraction of tuples marked are low. Thus the probability of the attacker modifying the watermarked tuples is low. Bit flipping attacks (A1–A3) are probabilistically ineffective since the identification of correct tuples, attributes and LSBs is dependent on MAC. Additive and invertibility attacks are still feasible.

3 ANALYSIS OF AGRAWAL-KIERNAN WATERMARKING SCHEME

Based on our observations, Agrawal and Kiernan scheme has three weaknesses.

1. **Susceptibility of secondary watermarking:** Secondary watermarking refers to an attacker who is trying to insert his watermark in an already watermarked relation. The scheme does not protect against secondary watermarking as the attacker can choose his/her own parameter list $\tilde{\phi}$ and insert a new watermark in the original watermarked relation. The new watermark will establish the ownership of the attacker over the relation and might also destroy the original watermark. If the watermarking is reversible, the actual owner's watermark can be recovered from the reversed relation.

2. **Lack of query-preservation:** If an attribute $r.A_i = x_1$ is modified to x_2, then query **"Select** r **from** R **where** $r.A_i = x_1$**"** cannot be preserved. Thus, it is obvious that not all queries are preservable in watermarked database. *Distance* $\delta_{r.A_i}$, that refers to the minimum difference between value of $r.A_i$ from values of A_i in other tuples, is not considered in (Agrawal and Kiernan, 2002), due to which queries might not be preserved. If we change value of an attribute beyond it's distance, the ordering of the tuples is modified when the relation is sorted on that attribute and hence query results change. Consider the following relations that contains foreign exchange rate data of some countries against 100 US Dollars. Table 1 is the original relation and Table 2 is the watermarked relation.Result of queries **"Select *Nation* from *ForEx* where *Selling rate*<*130*"** and **"Select *Currency* from *ForEx* where *Buying rate is maximum*"** are different when executed on the original and watermarked relations.

3. **Lack of tolerance of attributes:** The number of LSBs that can be used for watermarking are not dependent on the *tolerance* of the attributes. This results in the possibility that the relation becomes unusable from a user's perspective. *Tolerance* is different from *distance*. For example, even if population of the two countries differ by millions, modifying population values beyond a couple of thousands might render the data useless. Hence, the number of bits that one can change does not depend only on distance, but also on tolerance.

Table 1: Original **ForEx** relation.

Currency code	Nation	Buying rate	Selling rate
AUD	Australia	133	125
INR	India	4500	4300
THB	Thailand	3740	3510
SLR	Sri Lanka	4430	4210
NZD	New Zealand	151	134

Table 2: Watermarked **ForEx** relation.

Currency code	Nation	Buying rate	Selling rate
AUD	Australia	133	125
INR	India	4500	4300
THB	Thailand	3740	3510
SLR	Sri Lanka	***4530***	***4310***
NZD	New Zealand	151	***124***

We propose the following modifications to eliminate each of these weaknesses.

1. **Secondary Watermarking**

   To defeat secondary watermarking attacks, the step $r.A_i^j = \mathcal{H}(\mathcal{K} \circ r.P)\%2$ in Algorithm 1 is changed to

   $$\mathcal{ET} = \mathcal{ET} \circ r.A_i^j$$
   $$r.A_i^j = \mathcal{H}(\mathcal{K} \circ r.P \circ r.A_i^j)\%2,$$

   $r.A_i^j$ is concatenated to embed trace $\mathcal{ET}$ and then modified. The scheme is semi-blind and reversible, since the original values can be restored from $\mathcal{ET}$. The size of $\mathcal{ET}$ is proportional to $\frac{n}{\gamma}$. At the detection time, the value of *matchcount* is incremented only if

   $\tilde{r}_w.A_i^j == \mathcal{H}(\mathcal{K} \circ \tilde{r}_w.P \circ \mathcal{ET}[totalcount])\%2$
   $(i, j, totalcount, matchcount$ are counters updated during the detection)

   The owner stores $\mathcal{ET}$ at a secondary location. $(\mathcal{ET}, \mathcal{K})$ is the watermark detection key. Subsection 4.1 discusses how the rightful owner is identified if multiple parties watermark a relation in some sequence. Implementation is given in Algorithm 5.

2. **Query preservation**

   The value of an attribute $r.A_i$ should be modified by less than the distance $\delta_{r.A_i}$. Thereby, the number of bits available for watermarking are $\lfloor \log_2(\delta_{r.A_i}) \rfloor$ (For example, if the smallest difference between values of an attribute in two rows is

57.68, then only 5 bits can be used for watermarking as $\log_2(57.68) = 5.85$ and $\lfloor 5.85 \rfloor = 5$). This would guarantee query-preservation for the existing relation. Since the watermarking scheme is reversible, it facilitates incremental watermarking. The steps involved in incremental watermarking are,

(a) Restore relation to unmarked version.
(b) Add (or delete) required tuples (or attributes).
(c) Re-watermark the updated relation.

3. **Tolerance**
Since each attribute has a different tolerance limit beyond which it should not be modified, it is recommended that the number of LSBs to utilize for watermarking should be a function of tolerance of the attributes. Hence, ξ_i LSBs of attribute A_i can be modified. The list of all these values $\Xi = \{\xi_1, \xi_2, \ldots, \xi_\upsilon\}$ where υ attributes are available for watermarking.

4 MODIFIED ALGORITHMS

With the above modifications, the secret parameter list for watermark detection becomes $\phi = (\mathcal{K}, \mathcal{ET}, \gamma, \upsilon, \Xi)$. We present a reversible and semi-blind watermarking scheme that comprises of three algorithms; *Insertion*, *Detection*, and $(1,n)$*identification*. The algorithms are presented in Algorithm 3, Algorithm 4, and Algorithm 5 respectively. They contain comments illustrating the purpose served by various steps. The acronym WM refers to "watermark" in the three algorithms.

4.1 Identifying Rightful Owner

In this additional algorithm, ownership disputes can be resolved through backtracking. If $R \xrightarrow{ins(p_1)} R_1$ is followed by $R_1 \xrightarrow{ins(p_2)} R_2$, then $R_2 \xrightarrow{det(p_2)} R_1$ will show that the restored relation R_1 has already been watermarked by another party (p_1) and hence p_2 is not the original owner. For all potential owners u_i, we compare relations restored $R_{restored}$ after detecting watermark of party u_i, and if it matches any other party's watermarked relation R_w within a preset tolerance limit ε, then u_i is eliminated from the list of possible owners. Each party supplies its secret parameter list ϕ. and the relation R_w on which it claims ownership. A limitation is that each party that has watermarked the relation should participate in the owner-identification process. If this condition is not satisfied, we might not be able to associate the restored relation with another user, in which case the algorithm will fail.

Input: Relation R, private key K, fraction γ, number of markable attributes υ, LSB usage $\Xi = \{\xi_1, \xi_2, \ldots, \xi_\upsilon\}$
Output: Watermarked relation R_w, Embed Trace $\mathcal{ET}$

```
1  count = 0 ;            // index in WM to be generated
2  forall tuples r ∈ R do
3      if F(r.P)%γ = 0 then
4          i = F(r.P)%υ;          // identify attribute
5          j = F(r.P)%ξ_i;              // identify bit
6          if j < ⌊log2(δ_{r.A_i})⌋ then
7              ET[count] = r.A_i^j;   // store old value
               in WM
8              count = count + 1;     // next watermark
               bit's index
9              r.A_i^j = H(K ∘ r.P ∘ r.A_i^j)%2 ;    // modify
               bit in relation
10         end
11     end
12 end
```

Algorithm 3: Watermark insertion.

5 ANALYSIS

According to (Agrawal and Kiernan, 2002), the attacker Mallory needs to flip at least $\overline{\tau} = \omega - \tau + 1$ marked bits to carry out a successful attack, where $\omega = \frac{\eta}{\gamma}$. Let us assume that Mallory somehow knows the values of ξ and υ and randomly chooses ζ tuples. The probability that this attack will succeed when Mallory flips A_i^ξ for all υ attributes in all randomly selected ζ tuples is given in Equation 2 (Agrawal and Kiernan, 2002), and the values are provided in Table 3. For our modified watermarked scheme, $\xi = \frac{\sum_{i=1}^{\upsilon} \xi_i}{\upsilon}$. Note that if the attacker flips more than 50% bits, the watermark will be detected when the entire bits in the relation are flipped. This also gives us a fair idea about the value of γ that should be chosen. It should be fairly low and somewhere in between 10 and 100 as the attack is ineffective for values in this range.

$$\mathcal{P}(\mathcal{A}) = \sum_{i=\overline{\tau}}^{n} \frac{\binom{\omega}{i}\binom{\eta-\omega}{\zeta-i}}{\binom{\eta}{\zeta}} \qquad (2)$$

Without the knowledge of ξ, γ, υ, Mallory's task is much tougher. To compensate for the lack of knowledge, Mallory might need to choose an estimated ξ' and flip that bit of each of the attribute which degenerates the data quality. The security analysis

Input: Watermarked Relation $\tilde{R}_w$, Secret parameter list $\phi = (\mathcal{K}, \mathcal{ET}, \gamma, \upsilon, \Xi)$
Output: {Watermark Status $\in \{true, false\}$, Restored Relation R}

1 $R = \tilde{R}_w$;
2 $matchcount = 0$; // matching WM bits counter
3 $totalcount = 0$; // total WM bits counter
4 **forall** *tuples* $\tilde{r}_w \in \tilde{R}_w$ **do**
5 **if** $\mathcal{F}(r.P)\%\gamma = 0$ **then**
6 $i = \mathcal{F}(r.P)\%\upsilon$; // identify marked attribute
7 $j = \mathcal{F}(r.P)\%\xi_i$; // identify marked bit
8 **if** $j < \lfloor \log_2(\delta_{r.A_i}) \rfloor$ **then**
9 **if** $\mathcal{H}(\mathcal{K} \circ \tilde{r}_w.P) \oplus \mathcal{ET}[totalcount]\%2 = \tilde{r}_w.A_i^j$ **then**
10 $matchcount = matchcount + 1$; // bit authenticated
11 $r.A_i^j = \mathcal{ET}[totalcount]$; // restore bit in relation
12 **end**
13 $totalcount = totalcount + 1$;
14 **end**
15 **end**
16 **end**
17 $\tau = min(\theta) : B(\theta, totalcount, 1/2) < \alpha$; // threshold check
18 **if** $matchcount \geq \tau$ **then**
19 return $\{true, R\}$;
20 **else**
21 return $\{false, \tilde{R}_w\}$;
22 **end**

Algorithm 4: Watermark detection.

for (Agrawal and Kiernan, 2002) also holds for our scheme as the underlying operations are retained.

The advantages of our reversible watermarking scheme as compared to (Agrawal and Kiernan, 2002) are,

1. **Ownership resolution amongst *n* parties**

   This is not possible in the absence of a reversible watermarking scheme. Consider a situation in Figure 1 where a company C and five data servers d_1, d_2, d_3, d_4, d_5 are contesting for ownership over a relation, each party having a slightly different version of the same relation. The dotted line represents a relation being distorted by a party in an attempt to destroy any watermark it con-

Input: Potential owners $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$. Secret parameter list of each u_i, $\phi_{u_i} = \{\mathcal{K}_\backslash, \mathcal{ET}_i, \gamma_i, \upsilon_i, \Xi_i\}$, tolerance ε, Potential owners' versions of the watermarked relation $\{\tilde{R}_w^{u_1}, \tilde{R}_w^{u_2}, \ldots, \tilde{R}_w^{u_n}\}$
Output: Owner O

1 **forall** $u_i \in \mathcal{U}$ **do**
2 **if** $detect(\tilde{R}_w^{u_i}, I_{u_i}) == \{true, R'\}$ **then**
3 **forall** $u_j \in \mathcal{U}$ **do**
4 **if** *(*$difference(\tilde{R}_w^{u_j}, R') < \varepsilon$*) OR (*$detect(R', u_j) == true$*)* **then**
5 $\mathcal{U} = \mathcal{U} \setminus u_i$;
6 **end**
7 **end**
8 **end**
9 **end**

Algorithm 5: $(1,n)identification$.

Table 3: Probability of success for bit flipping attack.

γ	**bits flipped**	**success probability**
10000	40%	0.64
1000	46%	0.44
100	48%	0.11
10	>50%	≈ 0

tains. We assume that $\gamma < 100$ for all the parties who have watermarked the relation, which gives a high probability of the watermark being preserved if the relation is distorted or re-watermarked (Table 3). Hence, C's watermark is also detected in $\tilde{R}_1, \tilde{\tilde{R}}_1$ and d_1's watermark is also detected in $\tilde{R}_2, \tilde{\tilde{R}}_2, \tilde{\tilde{\tilde{R}}}_2$. When we execute Algorithm 5, the relation restored upon detecting watermark of each party matches another party's relation (except when C's watermark is detected). For example, the relation $\tilde{R}_2$ restored upon detecting d_3's watermark in R_4 matches R_2 within tolerance limit ε. But the relation R restored upon detecting the watermark of the authentic owner C, does not match any other party's relation, which establishes C's ownership.

Line 4 of Algorithm 5 maximizes the probability of identifying if the restored relation has been previously marked by another party. If the attacker only slightly modifies the relation before watermarking, the value of $difference(\tilde{R}_w^{u_j}, R')$ is less than ε, and even if the attacker changes the relation extensively, the value of $detect(R', u_j)$ is $true$ with a high probability (by Equation 2).

If some of the parties do not participate in the owner identification process, the algorithm might

fail. If we model parties watermarking relations as nodes of a tree where the actual owner is the root of the tree, then the probabilities with which the owner will be correctly identified despite nodes from n levels of the tree abstaining from participation is given by $(1-\mathcal{P}(\mathcal{A}))^n$. These probabilities are calculated taking into consideration the modifications the attacker might make in the relation before watermarking it. The probability that an attacker will succeed in destroying the watermark is $\mathcal{P}(\mathcal{A})$ and hence the probability of the relation surviving an attack is $1-\mathcal{P}(\mathcal{A})$. The probability of a relation surviving n sequential attacks is $(1-\mathcal{P}(\mathcal{A}))^n$. It is extremely rare that the relation will be distributed beyond three or four levels as there usually a few companies dealing with similar data. It is shown in (Agrawal and Kiernan, 2002) that the attacker has a probability of 11% success if he changes 48% of the tuples assuming $\gamma = 100$. Hence, if only C and d_3 participate in the correct algorithm, C will be identified as the correct owner with a probability of 89% since parties from only one level (d_1, d_2) abstain. This probability is 100% if C, d_1 participate or C, d_2 participate. In general, successful detection of the correct watermark occurs with the probability of 0.89^n where n levels abstain from participation for $\gamma = 100$. Thus the probability of finding rightful owner if two levels abstain is $0.89 * 0.89 = 0.7921$.

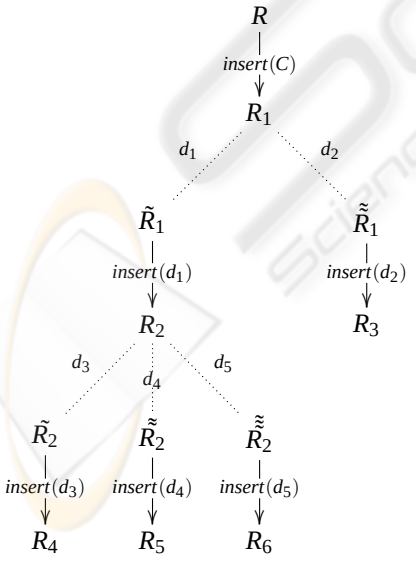

Figure 1: Dotted lines denote relation manipulated and solid lines denote relation watermarked.

2. **Situations requiring original dataset**

   Often, companies require precise data where a difference of even one bit might be disastrous such as stock markets and military operations. Once a relation is watermarked in (Agrawal and Kiernan, 2002), it cannot be restored to it's original state if needed. Since our watermarking scheme is reversible, the original data can be restored by executing the detection algorithm. It is also possible to distribute low-quality data free of cost and users can then purchase the key to extract original data.

5.1 Semi-blindness

As mentioned in the introduction, the two alternatives to facilitate reversibility are 1) Store original bits at a secondary location before modifying (This is the implemented solution in our paper or 2) Original bits and watermark bits should be recoverable from modified bits.

There are a few ways of implementing the second option,

Algorithm 3, statement 9 can be replaced by $r.A_i^j = \mathcal{H}(\mathcal{K} \circ r.P) \oplus r.A_i^j \% 2$. But an attacker $\mathcal{A}$ can run the insertion algorithm with inputs $(R_w, K', \gamma', \upsilon', \Xi')$ and get output R', W' such that $R' \xrightarrow{ins(\mathcal{A})} R_w$. Also $R \xrightarrow{ins(C)} R_w$, thus making it impossible to decide who (owner/ attacker) watermarked the relation first. Thus the this solution is vulnerable to pre-image attacks.

There have been reversible watermarking algorithms, primarily for images (Alattar, 2004; Tian, 2003). These schemes facilitate watermarking by encoding watermark bit and original value in the modified value at at the cost of watermarking capacity. Another option is to use lossless compression to first compress the original bits, append watermark bits and embed resulting bitstream (Celik et al., 2002). Since lossless compressions are sensitive to modifications, such schemes are not very resilient as suggested in (Jen-Bang Feng and Chu, 2006).

The first challenge in designing a blind reversible scheme for database relations is that lossless compression technique is not resilient against attacks. The second problem is that adapting reversible image watermarking schemes is harder because neighboring attributes or tuples do not have correlation unlike images, which is a prerequisite for schemes such as (Alattar, 2004). Our next research endeavor is to implement a fully blind

database watermarking model by working around these two limitations.

6 CONCLUSION

The watermarking scheme proposed by Agrawal and Kiernan is irreversible, resulting in problems during owner identification in case of additive or secondary watermarking attacks. Our modified scheme is reversible and thus the rightful owner can be identified from *n* candidates. The major advantages of our proposed scheme are 1)It provides query preservation, 2) It identifies rightful owner if relation is watermarked by multiple parties, and 3) It facilitates reversibility.

The current model requires modified bits to be stored at a secondary location ($\mathcal{ET}$). Our future research is directed towards eliminating this requirement and formulate a reversible blind watermarking scheme. The second enhancement is watermarking relations that do not contain a primary key. Concatenated attributes in a tuple can act as a primary key in such cases. However, the possibility of duplicate attributes makes identification of marked tuples difficult. One possibility is to treat tuples with duplicate attributes as a single tuple.

REFERENCES

Agrawal, R., Haas, P. J., and Kiernan, J. (2003). Watermarking relational data: framework, algorithms and analysis. *The VLDB Journal*, 12(2):157–169.

Agrawal, R. and Kiernan, J. (2002). Watermarking relational databases. In *Proceedings of the 28th International Conference on Very Large Databases VLDB*.

Alattar, A. (2004). Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Transactions on Image Processing*, 13(8):1147–1156.

Bors, A. and Pitas, I. (1996). Image watermarking using dct domain constraints. In *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, volume III, pages 231–234.

Braudaway, G. W. (1997). Protecting publicly-available images with an invisible image watermark. In *Proceedings of IEEE International Conference on Image Processing (ICIP'97)*, Santa Barbara, California.

Celik, M. U., Sharma, G., Tekalp, M. A., and Saber, E. (2002). Reversible data hiding. In *Proceedings of International Conference on Image Processing*, volume 2, pages 157–160.

Cox, I., Kilian, J., Leighton, T., and Shamoon, T. (1995). Secure spread spectrum watermarking for multimedia. Technical Report 128, NEC Research Institute.

Cox, I., Kilian, J., Leighton, T., and Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687.

Cox, I. J., Killian, J., Leighton, T., and Shamoon, T. (1996). Secure spread spectrum watermarking for images, audio, and video. In *IEEE International Conference on Image Processing (ICIP'96)*, volume III, pages 243–246.

Gross-Amblard, D. (2003). Query-preserving watermarking of relational databases and xml documents. In *Proceedings of the 20th ACM Symposium on Principles of Database Systems*, pages 191–201.

Guo, F., Wang, J., and Li, D. (2006). Fingerprinting relational databases. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 487–492, New York, NY, USA. ACM Press.

Jen-Bang Feng, Iuon-Chang Lin, C.-S. T. and Chu, Y.-P. (2006). Reversible watermarking: Current status and key issues. *International Journal of Network Security*, 2(3):161–171.

Li, Y. and Deng, R. H. (2006). Publicly verifiable ownership protection for relational databases. In *Proceedings of the ACM Symposium on Information, computer and communications security*, pages 78–89, New York, NY, USA. ACM Press.

Li, Y., Guo, H., and Jajodia, S. (2004). Tamper detection and localization for categorical data using fragile watermarks. In *DRM '04: Proceedings of the 4th ACM workshop on Digital rights management*, pages 73–82, New York, NY, USA. ACM Press.

Schneier, B. (1996). *Applied Cryptography*. John Wiley, second edition.

Sion, R., Atallah, M., and Prabhakar, S. (2004). Rights protection for relational data. *IEEE Transactions on Knowledge and Data Engineering*, 16(12):1509–1525.

Tian, J. (2003). Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8):890–896.

Zhang, Y., Niu, X.-M., and Zhao, D. (2004a). A method of protecting relational databases copyright with cloud watermark. *Transactions of Engineering, Computing and Technology*, 3:170–174.

Zhang, Y., Yang, B., and Niu, X.-M. (2006). Reversible watermarking for relational database authentication. *Journal of Computers*, 17(2):59–66.

Zhang, Z. H., Jin, X. M., Wang, J. M., and Li, D. Y. (2004b). Watermarking relational database using image. In *Proceedings of 3rd International Conference on Machine Learning and Cybernetics*, volume 3, pages 1739–1744.