Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 796726, 16 pages
http://dx.doi.org/10.1155/2013/796726

*Research Article*

# A Fragile Zero Watermarking Scheme to Detect and Characterize Malicious Modifications in Database Relations

**Aihab Khan**[1] **and Syed Afaq Husain**[2]

[1] *Department of Computing & Technology, Iqra University, Islamabad 44000, Pakistan*
[2] *Faculty of Computer Science & IT, King Faisal University, Ahsaa 31982, Saudi Arabia*

Correspondence should be addressed to Aihab Khan; aihabkhan@yahoo.com

Received 19 February 2013; Accepted 10 April 2013

Academic Editors: P. Melin and Y. Zhu

We put forward a fragile zero watermarking scheme to detect and characterize malicious modifications made to a database relation. Most of the existing watermarking schemes for relational databases introduce intentional errors or permanent distortions as marks into the database original content. These distortions inevitably degrade the data quality and data usability as the integrity of a relational database is violated. Moreover, these fragile schemes can detect malicious data modifications but do not characterize the tempering attack, that is, the nature of tempering. The proposed fragile scheme is based on zero watermarking approach to detect malicious modifications made to a database relation. In zero watermarking, the watermark is generated (constructed) from the contents of the original data rather than introduction of permanent distortions as marks into the data. As a result, the proposed scheme is distortion-free; thus, it also resolves the inherent conflict between security and imperceptibility. The proposed scheme also characterizes the malicious data modifications to quantify the nature of tempering attacks. Experimental results show that even minor malicious modifications made to a database relation can be detected and characterized successfully.

## 1. Introduction

Digital watermarking is a class of information hiding technique that provides measures for copyright protection, broadcast monitoring, covert communication, copy control, tamper, and integrity proof of digital assets. The watermarking techniques were primarily proposed for multimedia content [1–4]; however, in the last decade, the research community has extended these techniques to relational databases for its copyright protection, temper detection, and integrity proof. Most of the existing watermarking schemes for relational databases [5–20] introduce intentional errors or distortions as marks in the underlying data with some error tolerance so that it does not have a significant impact on the usefulness of data. However, this results in degrading data quality as the integrity of a relational database is violated. A large collection of real-world datasets has a strong usability constraint that disallows any permanent distortions or intentional errors. For example, the safety critical datasets are designed to minimize errors rather than to introduce intentional errors. Similarly, a business application may require that local properties like item-cost, ordered-quantity, and so forth, are preserved as well as global properties like natural join between item and sales, employees and department, and so forth. Moreover, in business datasets, the semantic constraints are not violated, like dissimilarity in attribute value for two similar transactions [21]. Query processing is sensitive due to selection criteria and has well-defined semantics; therefore, the watermarking schemes that introduce distortion into the database original content are not appropriate for certain applications.

Based on the intent of marking, the watermarking schemes presented in the literature can be categorized into robust and fragile schemes. The robust schemes [5–16] are aimed at copyright protection, whereas the fragile schemes [17–25] are used for tamper detection and integrity proof of database relations. Most of the robust schemes for copyright protection [5–16] introduce distortions into the database original content which affects data integrity and usability. These robust schemes may work for numeric [5–10] and categorical attributes [11, 12] of relational databases to embed watermarks. Some techniques embed meaningless bit pattern [5, 6]; whereas in other techniques meaningful bit patterns like image [13–15] and owner's speech [16] are used as watermarks for embedding in relational databases. In data sales environment, some of these robust schemes are extended to fingerprinting domain for unique identification of each buyer and also for traitor detection [21, 26–28]. Compared with the robust schemes, the fragile watermarking schemes are not adequately addressed and relatively little work is available for integrity proof of relational databases [20]. In this paper, we focus on fragile watermarking schemes for temper detection and integrity proof of database relations.

The initial work on fragile watermarking schemes can be found on images [29–31], which is extended to audio [32, 33] and video [3, 34] schemes. Recently, the importance of other data domains is recognized and fragile schemes for text [35, 36] and relational databases [17–20, 22–25] are proposed. Like robust schemes, most of the fragile schemes for relational databases [17–20] introduce distortion into the database original contents that degrades data quality and also affects data usability. These schemes are based on the content characteristics of database relation itself to create a secure hash (used as a watermark) which is stored in Least Significant Bits (LSBs) of database original contents, thus introducing distortion.

A fragile watermarking scheme presented by Guo et al. [17] detects malicious modifications made to a database relation. In their scheme, the watermark generation is based on the content characteristics of the database relation itself. The generated watermarks are embedded in at most two LSBs of all attributes in the database relation that introduces considerable distortion in the database original contents. The fragile scheme presented by Khataeimaragheh and Rashidi [18] is also a distortion-based scheme for integrity proof of database relations. Like [17], the watermarks are embedded in at most two LSBs of all attributes in the relation that forms a two-bit watermark grid. The fragile scheme presented by Iqbal et al. [19] logically partitions the database relation into three groups and generates self-constructing fragile watermark information from each group. The generated watermarks are embedded at LSBs of numerical attributes in each group of a database relation which introduces distortion in database original contents. Prasannakumari [20] presented a fragile scheme for temper detection in database relations. This technique also introduces distortion as it inserts a fake attribute in database relation to act as a watermark. The data values for the newly inserted attribute are determined by applying aggregate function on original database content.

Beside distortion-based techniques, some researches also presented distortion free fragile watermarking schemes [22–25] for integrity proof of database relations. The main feature of these schemes is that the watermark embedding in actual fact is the tuples or attributes reordering based on the content characteristics of database relation. A fragile scheme proposed by Li et al. [22] detects and localizes malicious modifications made to the database relations. Their scheme partitions the database relation into disjoint groups and the watermark is embedded and verified in each group independently. In their scheme, the watermark is embedded as tuple reordering and the order of each tuple pair in group is changed or unchanged depending on the tuple hash values and the corresponding group hash value. Though their technique does not introduce any distortion in the database relation, but it works only for categorical data type. Kamel [23] presented a fragile scheme to protect the integrity of database relations. Their scheme divides the database relations in groups and each group is marked independently. As in [22], the watermark embedding is reordering of tuples in each group that corresponds to the value of some secret watermark. The fragile scheme proposed by Bhattacharya and Cortesi [24] detects malicious modifications in database relations having categorical attributes. Their scheme divides the database relation into groups on the basis of categorical attribute values. Like [22, 23], tuple hash value is used to obtain a watermark as permutation of tuples. A fragile zero watermarking scheme is presented by Hamadou et al. [25] for authentication of database relations. Their technique is distortion-free and is based on attribute reordering method. Initially, the attributes of database relation are virtually sorted on hash values of attribute names to define a secret initial order of attributes. For each attribute in database relation, the Most Significant Bits (MSBs) are extracted and used for watermark generation. The generated watermark is then registered with the Certification Authority (CA) for certification purpose. As their technique is based on virtual sorting of attributes by their names, so any change in attribute name by attacker would fail the temper detection process.

In the previous discussion, we have identified two important issues in existing fragile watermarking schemes. First, the fragile schemes are distortion based [17–20] that inevitably degrade data integrity and thus affect data usability; therefore, these schemes are not applicable to non-error-tolerant data like safety critical datasets, and so forth. Second, though there exist some fragile schemes like [22–25] that are distortion-free, but the watermarking approach is based on reordering of tuples or attributes; so, they are vulnerable to sorting attacks. Also, if the modification is small, such that, it does not affect the order of tuples, the temper detection would fail. To address these issues, we propose a fragile scheme based on zero watermarking approach that does not modify any part or properties of the database relations itself; therefore, the proposed scheme assures imperceptibility and overcomes weaknesses like data integrity and data usability in existing fragile watermarking schemes. Also, the proposed scheme is independent of tuple ordering as well as attributes ordering and naming, so it is not vulnerable to sorting attacks. The watermark generation in the proposed scheme is based on algorithmically evaluating the local characteristics of database relation like frequency distribution of digit count, length and range of data values. This enables us to characterize the malicious data modifications on parameters like the fraction of digit, length and range of data values attacked, the type of attack (insertion, deletion, or update), and the effect of attack (low to high, high to low, or no change) on data values. Also, to the best of our knowledge, there is no such distortion-free fragile watermarking scheme that can characterize the tempering attacks, that is, the nature of tempering. Experimental results show that the proposed scheme can detect and characterize malicious data modifications successfully.

## 2. Materials and Methods

In this section, we present our proposed fragile zero watermarking scheme to detect and characterize malicious modifications made to a database relation. The proposed scheme exhibits the following important properties of a fragile watermarking system as discussed in [17].

(1) *Fragility.* The proposed scheme is designed to be fragile; that is, if there are any malicious data modifications, the embedded watermark is not detectable (destroyed).

(2) *Imperceptibility.* As the proposed scheme is based on zero watermarking approach, it does not introduce any distortion in the underlying data; therefore, the embedded watermark is invisible or imperceptible.

(3) *Key-Based System.* The watermark generation and verification in the proposed scheme is a key-based system. Also, to detect and characterize malicious data modifications, a secret key is required.

(4) *Blindness.* In the proposed scheme, the original database relation is not required to detect and characterize malicious data modifications.

(5) *Tuple and Attribute Ordering.* The existing fragile schemes are based on tuple ordering [22–24] and attribute ordering and naming [25]. The proposed scheme is independent of tuple and attributes ordering so it is not vulnerable to sorting attacks.

(6) *Characterization.* The proposed scheme not only detects but also characterizes the malicious data modifications in database relation to quantify the nature of tempering attacks.

*2.1. Watermark Generation.* Let $R$ be a database relation with primary key PK and $v$ attributes denoted by $R(\mathrm{PK}, A1, A2, \ldots, A_v)$. The watermark generation in the proposed scheme is based on the content characteristics of numeric data values, so we assume that some attributes of the database relation are numeric. Figure 1 shows the watermark generation process that comprises of subwatermark generation for digit count, length, and range of data values. The generated watermark is registered with the Certification Authority (CA) for certification purpose. Table 1 presents the list of notations used in our algorithms and discussion.

The algorithm for watermark generation is presented in Algorithm 1. At lines 1–3, the digit, length, and range of data values in a database relation are algorithmically evaluated to generate the subwatermarks as presented in Algorithms 2–4. These subwatermarks are then used to generate a database relation watermark $\omega_R$ as shown at line 4. At line 5, the relation watermark $\omega_R$ is encrypted with a secret key SK known only to the database owner. We assume that the secret key is selected from large key space such that it is computationally infeasible for attacker to guess a key. At lines 6-7, the encrypted relation watermark $E\omega_R$ is concatenated with owner Id along with date and time stamp to generate a watermark certificate $\omega_C$, which is then registered with the CA before publishing the database for certification purpose.

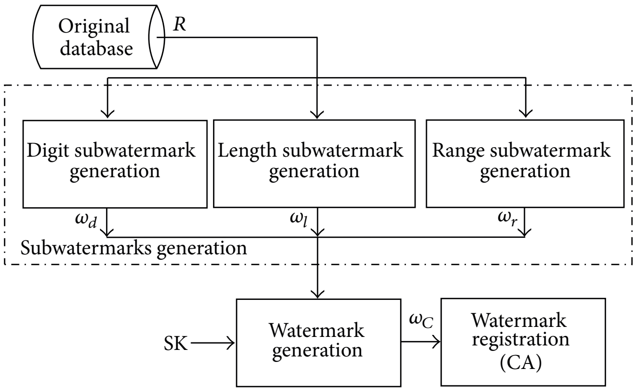

Original database
R
Digit subwatermark generation
Length subwatermark generation
Range subwatermark generation
$\omega_d$
$\omega_l$
$\omega_r$
Subwatermarks generation
SK
Watermark generation
$\omega_C$
Watermark registration (CA)

Figure 1: Proposed model for watermark generation and registration.

```
(1) $\omega_d$ = digit sub-watermark generation()
                                   //See Algorithm 2
(2) $\omega_l$ = length sub-watermark generation()
                                   //See Algorithm 3
(3) $\omega_r$ = range sub-watermark generation()
                                   //See Algorithm 4
(4) $\omega_R = \omega_d \parallel \omega_l \parallel \omega_r$
(5) $E\omega_R = Encrypt(\omega_R, SK)$
(6) $\omega_C = E\omega_R \parallel owner\_id \parallel date \parallel time$
(7) Register $\omega_C$ to CA
```

Algorithm 1: Watermark generation.

Algorithm 2 generates a digit subwatermark which is based on digit frequency for all data values present in adatabase relation. At lines 1–3, the length of each data value is determined which is then used to extract the individual digits as shown at lines 4-5. Lines 6-7 compute the frequency of each digit and the total number of digits present in the database relation. At line 11, the relative frequency of each digit $rfd_i$ is determined which is then used to generate a digit subwatermark $\omega_d$ as shown at line 13. At lines 15-16, the digit subwatermark $\omega_d$ is concatenated with total digit count and is returned to the watermark generation algorithm. It is to be noted that the digit subwatermark is composed of each digit relative frequency $rfd_i$ and the total count of all digits. In fact, this information is used for characterization of attacks as discussed in Section 3.

The subwatermark generation for length of data values in a database relation is presented in Algorithm 3. At lines 1–3, the length of each data value is determined. Lines 4-5 determine the frequency for each length of data values and the total count of data values length present in the database relation. At line 9, the relative frequency for each length of data value $rfl_j$ is computed which is then used to generate

```
For Each tuple $r_i$ € relation $R$ Do
For Each attribute $A_j$ € relation $R$ Do
    length = Len($r_i \cdot A_j$)
    For $i = 0$ to length − 1 Do
        $d_i$ = Mid\$($r_i \cdot A_j, i, 1$)
        digit_frequency[$d_i$]++
        total_digit_count++
    End For
End For
End For
For Each $i$ € digit Do
    $rfd_i$ = (digit_frequency[$i$]/total_digit_count) ∗ 100
    $\omega_d = \omega_d \parallel rfd_i$
End For
$\omega_d = \omega_d \parallel$ total_digit_count
Return $\omega_d$
```

Algorithm 2: Digit sub-watermark generation.

Table 1: Notations.

| Symbol | Description |
|---|---|
| $R$ | Database relation |
| PK | Primary key attribute |
| $r_i$ | The $i$th tuple |
| $A_j$ | The $j$th attribute |
| $\eta$ | Number of tuples in a database relation |
| $\nu$ | Number of attributes in a database relation |
| $\omega_d$ | Digit sub-watermark |
| $\omega_l$ | Length sub-watermark |
| $\omega_r$ | Range sub-watermark |
| $\omega_R$ | Watermark for database relation $R$ |
| $\omega_C$ | Watermark certificate |
| SK | Secret key |
| $d_i$ | The $i$th digit |
| $l_j$ | The $j$th length |
| $r_k$ | The $k$th range |
| $fd_i$ | Frequency for digit $i$ of data values |
| $fl_j$ | Frequency for length $j$ of data values |
| $fr_k$ | Frequency for range $k$ of data values |
| $rfd_i$ | Relative frequency for digit $i$ of data values |
| $rfl_j$ | Relative frequency for length $j$ of data values |
| $rfr_k$ | Relative frequency for range $k$ of data values |
| $\Delta fd_i$ | Change in frequency of digit $i$ |
| $\Delta fl_j$ | Change in frequency for length $j$ |
| $\Delta fr_k$ | Change in frequency for range $k$ |
| $\Delta \mathscr{F} d_i$ | Fractional change in digit frequency for digit $i$ |
| $\Delta \mathscr{F} l_j$ | Fractional change in length frequency for length $j$ |
| $\Delta \mathscr{F} r_k$ | Fractional change in range frequency for range $k$ |
| CA | Certification authority |
| WAR | Watermark accuracy rate |
| WDR | Watermark distortion rate |

```
(1)  For Each tuple $r_i$ € relation $R$ do
(2)  For Each attribute $A_j$ € relation $R$ do
(3)      length = Len($r_i \cdot A_j$)
(4)      length_frequency[length]++
(5)      total_length_count++
(6)  EndFor
(7)  End For
(8)  For Each $i$ € length do
(9)      $rfl_j$ = (length_frequency[$i$]/total_length_count) ∗ 100
(10)     $\omega_l = \omega_l \parallel rfl_j$
(11) End For
(12) $\omega_l = \omega_l \parallel$ total_length_count
(13) Return $\omega_l$
```

Algorithm 3: Length sub-watermark generation.

length subwatermark $\omega_l$ as shown at line 10. At lines 12-13, the length subwatermark $\omega_l$ is concatenated with total length count and is returned.

Algorithm 4 presents the algorithm for subwatermark generation for range of data values in a database relation. At line 1, different data ranges are defined in which the data value of a database relation may fall. It is to be noted that the defined data ranges may be adjusted as per the nature of data values in the database relation and also for more precise characterization of malicious data modifications, as discussed in Section 3. Lines 1–3 determine the attribute value, within each tuple. Lines 5–13 determine the frequency for different data ranges in which the data value may fall and the total number of data ranges present in the database relation. At lines 16-17, the relative frequency for each range of data value $rfr_k$ is computed, which is then used to generate range subwatermark $\omega_r$. Lines 19-20 show that the range subwatermark $\omega_r$ is concatenated with total range count and is returned.

```
range = {0–99, 100–999, 1000–9999, 10000–99999, 100000–999999}
For Each tuple $r_i$ € relation $R$ do
For Each attribute $A_j$ € relation $R$ do
    x = $r_i \cdot A_j$
    Select Case x
        x in range 0: range_frequency[0]++
        x in range 1: range_frequency[1]++
        x in range 2: range_frequency[2]++
        x in range 3: range_frequency[3]++
        x in range 4: range_frequency[4]++
    End Select
    total_range_count++
End For
End For
For Each $i$ € range do
    $rfr_k$ = (range_frequency[$i$]/total_range_count) $*$ 100
    $\omega_r = \omega_r \parallel rfr_k$
End For
$\omega_r = \omega_r \parallel$ total_range_count
Return $\omega_r$
```

ALGORITHM 4: Range sub-watermark generation.

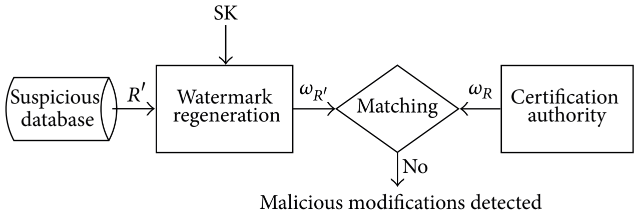


FIGURE 2: Proposed model for detection of malicious tempering.

*2.2. Watermark Verification.* Figure 2 shows the model for detection of malicious modifications in suspicious database relation $R'$. For detection of malicious data modifications, the relation watermark $\omega_R{}'$ is regenerated for suspicious database relation $R'$ and compared with the relation watermark $\omega_R$ registered at CA; if both watermarks are different then the suspicious database relation $R'$ is considered as a tempered relation.

The algorithm for watermark detection is presented in Algorithm 5. At line 1, the watermark $\omega_R{}'$ is generated by using Algorithm 1 for suspicious database relation $R'$. The watermark certificate $\omega_C$ which is already registered at CA is used to extract database relation watermark $\omega_R$ as shown at lines 2–4. At lines 5–10, each digit of $\omega_R$ is compared with the corresponding digit of $\omega_R{}'$ and match_count is incremented on each successful match. At line 9, the total_count is computed to know the number of digits tested. At lines 11-12, the WAR (Watermark Accuracy Rate) and WDR (Watermark Distortion Rate) are computed. If the distortion exists in the suspicious database relation $R'$, then $R'$ is rejected as a tempered relation with distortion rate WDR as shown at lines 13–15.

The algorithm for characterization of malicious data modifications is presented in Algorithm 6. At line 2, the relative frequency of each digit $rfd_i$ is extracted from digit subwatermark $\omega_d$ as $\omega_d \subseteq \omega_R$ and $\omega_R$ is already registered at CA. The frequency distribution of each digit $fd_i$ in relation $R$ is determined at line 3. At line 4, the frequency distribution of each digit $fd_i{}'$ for suspicious database relation $R'$ is determined. The change in frequency distribution of each digit $\Delta fd_i$ is computed at line 5 and the fractional change in each digit $\Delta\mathscr{F}d_i$ is determined at line 6. The computed value of $\Delta\mathscr{F}d_i$ is then used to characterize the malicious modifications made to the database relation $R$. For example, if $\Delta\mathscr{F}d_i$ is zero, then the suspicious relation $R'$ is not tempered. A positive $\Delta\mathscr{F}d_i$ indicates that $\mathscr{F}$ fraction of digit $d_i$ is maliciously inserted by attacker as an attempt to transform low data values to high in database relation $R$. Similarly, a negative $\Delta\mathscr{F}d_i$ indicates that $\mathscr{F}$ fraction of digit $d_i$ is maliciously deleted by attacker as an attempt to transform high data values to low in database relation $R$. At lines 8–14 and 15–21, a similar method as discussed earlier is used to determine $\Delta\mathscr{F}l_j$ and $\Delta\mathscr{F}r_k$ to characterize the attacks on length and range of data values in database relation $R$. The characterization of malicious data modifications is further elaborated in Section 3.2 with experimental results.

## 3. Results and Discussion

Suppose that Alice is the database owner and she has used the proposed algorithms along with the secret key to generate a watermark for the database relation $R$. The attacker Mallory for his own nefarious objectives may attempt to make malicious modifications in Alice watermarked database relation. We conducted our experiments in Microsoft Visual Basic and Microsoft Access, on 3.2 GHz Intel core i3 CPU with 2 GB of RAM. The proposed watermarking scheme is evaluated on a real-life dataset namely Forest Cover Type data set, available at UCI Machine Learning Repository [37]. This dataset has

```
Compute $\omega_R{}'$ for $R'$ (by using Algorithm 1)
Obtain $\omega_C$ from CA
$Extract\ E\omega_R$ from $\omega_C$
$\omega_R = Decrypt(E\omega_R, SK)$
For $i = 1$ to length($\omega_R$) Do
    If $\omega_R[i] = \omega_R{}'[i]$ Then
        $match\_count = match\_count + 1$
    End if
    $total\_count = total\_count + 1$
End For
$WAR = match\_count/total\_count * 100$
$WDR = 1 - WAR$
If $WDR \neq 0$ Then
    Database Relation $R'$ is tempered with distortion rate $WDR$
End if
```

Algorithm 5: Watermark verification.

```
For Each $d_i$ € $digit$ do
    Extract $rfd_i$ from $\omega_d$
    $fd_i = rfd_i$/total_digit_count
    $Generate\ fd_i{}'$ from $R'$ (by using Algorithm 2)
    $\Delta fd_i = fd_i - fd_i{}'$
    $\Delta \mathscr{F} d_i = \Delta fd_i / fd_i * 100$
End For
For Each $l_j$ € $length$ do
    Extract $rfl_j$ from $\omega_l$
    $fl_j = rfl_j$/total_length_count
    $Generate\ fl_j{}'$ from $R'$ (by using Algorithm 3)
    $\Delta fl_j = fl_j - fl_j{}'$
    $\Delta \mathscr{F} l_j = \Delta fl_j / fl_j * 100$
End For
For Each $r_k$ € $range$ do
    Extract $rfr_k$ from $\omega_r$
    $fr_k = rfr_k$/total_range_count
    $Generate\ fr_k{}'$ from $R'$ (by using Algorithm 4)
    $\Delta fr_k = fr_k - fr_k{}'$
    $\Delta \mathscr{F} r_k = \Delta fr_k / fr_k * 100$
End For
```

Algorithm 6: Characterization of malicious data modifications.

581,102 tuples, each with 10 integer attributes, 44 Boolean attributes, and 1 categorical attribute. In our experiments, we have used all 10 integer attributes. It is to be noted that in robust watermarking schemes, the aim of Mallory is to destroy the Alice watermark without affecting the database relation, whereas in fragile schemes, Mallory attempts to make malicious modifications in Alice watermarked database relation without affecting the watermark. The experimental results presented in this section show that the watermark is adversely affected by even minor malicious data modifications; therefore, the generated watermark is fragile.

*3.1. Detection of Malicious Modifications.* In this set of experiments, we randomly introduce malicious modifications in Forest Cover Type data set [37]. As discussed in Algorithm 5, these malicious modifications are detected by generating the watermark for the suspicious database relation $R'$ to obtain $\omega_R{}'$, which is then compared with the registered watermark $\omega_R$ to determine the WAR (Watermark Accuracy Rate) and WDR (Watermark Distortion Rate).

Table 2: Detection of malicious insertion of tuples with different attack rates ($\eta = 10^6$).

| Insertion attack rate | WAR | WDR | Temper detection |
|---|---|---|---|
| 10% | 18.14 | 81.86 | Yes (High) |
| 30% | 18.56 | 81.44 | Yes (High) |
| 50% | 20.41 | 79.59 | Yes (High) |
| 70% | 16.67 | 83.33 | Yes (High) |
| 90% | 16.32 | 83.68 | Yes (High) |

Table 3: Detection of malicious deletion of tuples with different attack rates ($\eta = 10^6$).

| Deletion attack rate | WAR | WDR | Temper detection |
|---|---|---|---|
| 10% | 24.32 | 75.68 | Yes (High) |
| 30% | 17.88 | 82.12 | Yes (High) |
| 50% | 20.95 | 79.05 | Yes (High) |
| 70% | 13.08 | 86.92 | Yes (High) |
| 90% | 14.14 | 85.86 | Yes (High) |

Table 2 shows the WAR and WDR for the malicious insertions made to the database relation with different attack rates. For example, when 10% of the fake but similar tuples are randomly inserted into the database relation $R$, the WDR is found to be high and malicious insertions are detected with low WAR.

Tables 3-4 show similar results as of insertion attack for malicious deletions and updates made to the database relation $R$.

Figure 3 summarizes the insertion, deletion, and update attacks and shows that the WDR is always high for different volume of malicious data modifications.

Table 4: Detection of malicious update of tuples with different attack rates ($\eta = 10^6$).

| Update attack rate | WAR | WDR | Temper detection |
|---|---|---|---|
| 10% | 20.42 | 79.58 | Yes (High) |
| 30% | 19.89 | 80.11 | Yes (High) |
| 50% | 19.89 | 80.11 | Yes (High) |
| 70% | 18.94 | 81.06 | Yes (High) |
| 90% | 14.13 | 85.87 | Yes (High) |

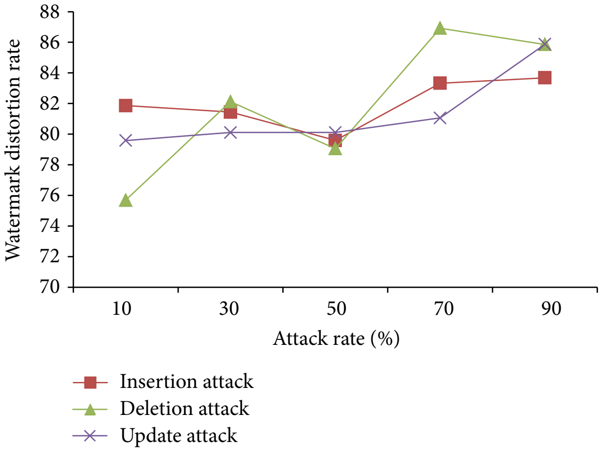


Figure 3: Watermark distortion rate for malicious insertion, deletion, and update of tuples with different attack rates ($n = 10^6$).

In another set of attacks, we simultaneously perform malicious insertion, deletion, and update of tuples with different attack rates in database relation $R$. Table 5 shows the WDR for this set of attack.

The experimental results presented in Tables 2–5 show that the malicious modifications are always detected and fragility of the registered watermark $\omega_R$ is observed for even low volumes of attack. The WAR is low and WDR is high for different volume of malicious insertions, deletions, and updates made to the database relation. The low WAR indicates the extent to which the database relation has been attacked, whereas the high WDR indicates that the database relation has been tampered and is not authentic. The accuracy of watermark is adversely affected even with minor malicious data modifications and the watermark fragility proves that the database relation has been attacked.

*3.2. Characterization of Malicious Modifications.* One of the important features of the proposed watermarking scheme is to characterize the malicious modifications made to the database relations. As discussed in Algorithm 1, the watermark generation is based on the content characteristics of database relation itself which enable us to characterize the malicious data modifications. Algorithm 6 elaborates the algorithm for characterization of malicious data modifications by evaluating the fractional change in each digit $\Delta\mathcal{F}d_i$, length $\Delta\mathcal{F}l_j$ and range $\Delta\mathcal{F}r_k$ of data values in the tempered database relation $R'$.

We have conducted experiments for both random and deterministic attacks for characterization of malicious data modifications. In random tempering attacks, we randomly attack the digit frequency, length, and range of data values in the database relation, whereas in deterministic attacks, the attack is performed with the specific attack rates. The random tempering attacks are presented in this section and the results of detailed deterministic attacks are shown in the Appendix for reference.

*3.2.1. Attacks on Digit Frequency.* In this set of attacks, Mallory randomly performs malicious insertion, deletion, and update attacks on digit frequency in Alice's watermarked relation $R$. For example, in insertion attack, Mallory may attempt to maliciously insert some digits in $R$. Table 6 shows the experimental results obtained for characterization of malicious insertion attack on digits 9 and 0 as discussed in Algorithm 6. A positive value of $\Delta\mathcal{F}d_i$ indicates that $\mathcal{F}$ fraction of digits 9 and 0 is maliciously inserted by Mallory in the database relation $R$. The characteristic of this attack is an attempt to relatively increase the low data values to high in database relation $R$ as an increase of 35.84% and 24.42% is observed in $\Delta\mathcal{F}d_i$ of digits 9 and 0, respectively. As the other digits are not attacked, so $\Delta\mathcal{F}d_i$ is zero for digits 1–8 and there is no change in the digit frequency $\Delta fd_i$ of these digits. This characteristic of attack, when combined with the nature of data, may provide useful information about the attacker intention. For example, in the product sales environment, these malicious insertions indicate that the attacker may have attempted to increase the low volume and amount of product sales.

Table 7 shows the result for random malicious deletions of digits 9 and 0 made to the database relation $R$. A negative value of $\Delta\mathcal{F}d_i$ indicates that $\mathcal{F}$ fraction of digits 9 and 0 is maliciously deleted by the attacker. The characteristic of this attack is an attempt to relatively decrease the high data values to low in the database relation $R$. In this attack, 14.70% of digit 9 and 12.44% of digit 0 are randomly deleted from the database relation. As the other digits are not deleted, so $\Delta\mathcal{F}d_i$ is zero for digits 1–8. Table 8 shows similar result for random malicious update for digits 9 and 0 made to the database relation. In this attack, digits 9 and 0 are randomly replaced with some other digits, so the digit frequency $\Delta fd_i$ of digits 9 and 0 is decreased (high to low), where as the digit frequency $\Delta fd_i$ of digits 1–8 is increased (low to high).

Figure 4 summarizes the malicious insertion, deletion, and update attacks on digits 9 and 0. The insertion attack shows a positive increase (low to high) on attacked digits, where as a negative trend (high to low) is observed in attacked digits for deletion attack. In update attack, both negative (high to low) and positive trends (low to high) are observed for attacked and unattacked digits, respectively.

In another set of attacks, we randomly insert, delete and update 10% (lower bound) and 90% (upper bound) of the tuples from the database relation $R$. Table 9 shows the effect on fractional change in digit frequency $\Delta\mathcal{F}d_i$ for each digit. It is to be noted that, in insertion attack, a $k$ fraction of positive trend (low to high) is being observed in each digit frequency of database relation $R$. For example, when 10% of

Table 5: Detection of malicious data modifications with different attack rates ($\eta = 10^6$).

| Insertion attack rate | Deletion attack rate | Update attack rate | WAR | WDR | Temper detection |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 15.85 | 84.15 | Yes (High) |
| 30% | 30% | 30% | 10.98 | 89.02 | Yes (High) |
| 50% | 50% | 50% | 10.28 | 89.72 | Yes (High) |
| 70% | 70% | 70% | 13.33 | 86.67 | Yes (High) |
| 90% | 90% | 90% | 10.98 | 89.02 | Yes (High) |

Table 6: Characterization of malicious insertion attacks on digit frequency.

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i'$ | $fd_i'$ | $\Delta fd_i$ | $\Delta \mathscr{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| **0** | **8.63** | **1435163** | **10.28** | **1785659** | **+350496** | **+24.42%** | **↑Low to High** |
| 1 | 18.02 | 2995771 | 17.24 | 2995771 | 0 | 0 | No change |
| 2 | 19.48 | 3238818 | 18.64 | 3238818 | 0 | 0 | No change |
| 3 | 11.70 | 1945572 | 11.20 | 1945572 | 0 | 0 | No change |
| 4 | 8.22 | 1366062 | 7.86 | 1366062 | 0 | 0 | No change |
| 5 | 7.48 | 1244089 | 7.16 | 1244089 | 0 | 0 | No change |
| 6 | 6.65 | 1105210 | 6.36 | 1105210 | 0 | 0 | No change |
| 7 | 6.45 | 1072363 | 6.17 | 1072363 | 0 | 0 | No change |
| 8 | 6.60 | 1097669 | 6.32 | 1097669 | 0 | 0 | No change |
| **9** | **6.76** | **1123819** | **8.78** | **1526545** | **+402726** | **+35.84%** | **↑Low to High** |

Table 7: Characterization of malicious deletion attacks on digit frequency.

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i'$ | $fd_i'$ | $\Delta fd_i$ | $\Delta \mathscr{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| **0** | **8.63** | **1435163** | **7.72** | **1256568** | **−178595** | **−12.44%** | **↓High to Low** |
| 1 | 18.02 | 2995771 | 18.40 | 2995771 | 0 | 0 | No change |
| 2 | 19.48 | 3238818 | 19.89 | 3238818 | 0 | 0 | No change |
| 3 | 11.70 | 1945572 | 11.95 | 1945572 | 0 | 0 | No change |
| 4 | 8.22 | 1366062 | 8.39 | 1366062 | 0 | 0 | No change |
| 5 | 7.48 | 1244089 | 7.64 | 1244089 | 0 | 0 | No change |
| 6 | 6.65 | 1105210 | 6.79 | 1105210 | 0 | 0 | No change |
| 7 | 6.45 | 1072363 | 6.59 | 1072363 | 0 | 0 | No change |
| 8 | 6.60 | 1097669 | 6.74 | 1097669 | 0 | 0 | No change |
| **9** | **6.76** | **1123819** | **5.89** | **958569** | **−165250** | **−14.70%** | **↓High to Low** |

Table 8: Characterization of malicious update attacks on digit frequency.

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i'$ | $fd_i'$ | $\Delta fd_i$ | $\Delta \mathscr{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| **0** | **8.63** | **1435163** | **5.71** | **948993** | **−486170** | **−33.88%** | **↓High to Low** |
| 1 | 18.02 | 2995771 | 19.01 | 3159784 | +164013 | +5.47% | ↑Low to High |
| 2 | 19.48 | 3238818 | 20.97 | 3485451 | +246633 | +7.61% | ↑Low to High |
| 3 | 11.70 | 1945572 | 11.91 | 1980325 | +34753 | +1.79% | ↑Low to High |
| 4 | 8.22 | 1366062 | 8.52 | 1416889 | +50827 | +3.72% | ↑Low to High |
| 5 | 7.48 | 1244089 | 8.22 | 1365803 | +121714 | +9.78% | ↑Low to High |
| 6 | 6.65 | 1105210 | 6.90 | 1146565 | +41355 | +3.74% | ↑Low to High |
| 7 | 6.45 | 1072363 | 7.14 | 1187586 | +115223 | +10.74% | ↑Low to High |
| 8 | 6.60 | 1097669 | 6.78 | 1127651 | +29982 | +2.73% | ↑Low to High |
| **9** | **6.76** | **1123819** | **4.85** | **805489** | **−318330** | **−28.33%** | **↓High to Low** |

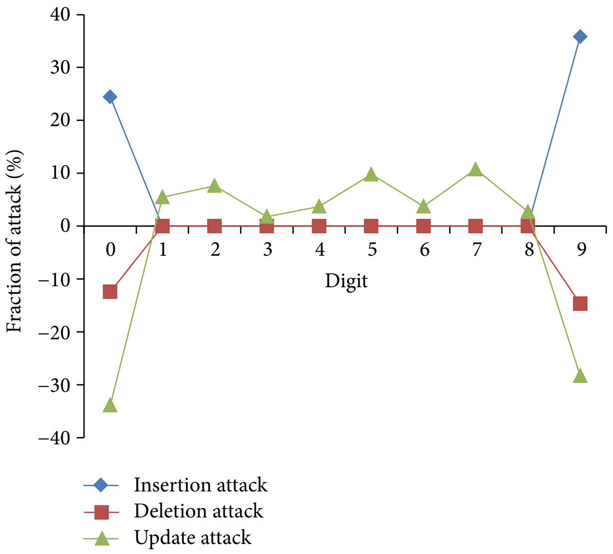


Figure 4: Characterization of malicious insertion, deletion, and update attacks on digits 9 and 0 of data values.

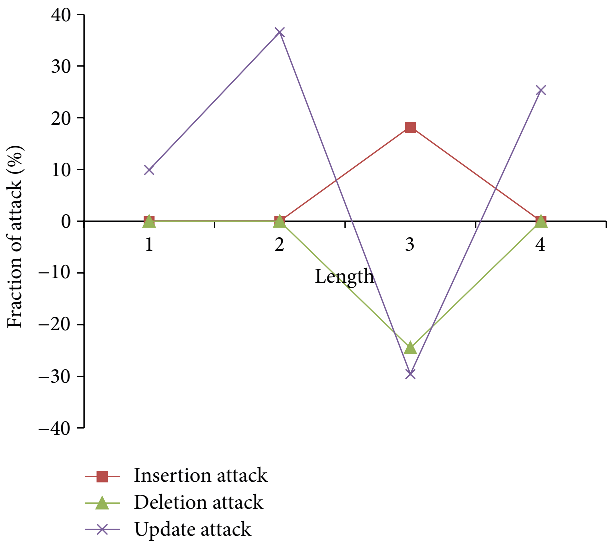


Figure 5: Characterization of malicious insertion, deletion, and update attacks on length 3 of data values.

similar tuples are inserted in database relation, an increase of approximately 10% is being observed in $\Delta\mathcal{F}d_i$ for each digit of database relation. Similarly, in deletion attack, a $k$ fraction of negative trend (high to low) is observed in $\Delta\mathcal{F}d_i$ for each digit of database relation. In update attack, no specific trend is observed in $\Delta\mathcal{F}d_i$ as $k$ fractions of digits are randomly replaced by some other digits.

It is to be noted that the attack on digit frequency (as discussed above) can be characterized on parameters like the digits being attacked, the fraction of each digit attacked, the type of attack (insertion, deletion, or update) on each digit, and the effect of attack (low to high, high to low, or no change) on data values.

*3.2.2. Attack on Length of Data Values.* In this set of attacks, Mallory randomly performs malicious insertion, deletion, and update attacks on length of data values. Table 10 shows the experimental result for characterization of malicious insertion on data values of length 3 in the database relation $R$. A positive value of $\Delta\mathcal{F}l_j$ indicates that $\mathcal{F}$ fraction of length $l_j$ is maliciously inserted in the database relation $R$. The characteristic of this attack is to relatively increase the low data values to high as an increase of 18.27% is observed in $\Delta\mathcal{F}l_j$ for data values of length 3. Also, $\Delta\mathcal{F}l_j$ is zero for lengths 1, 2, and 4, which shows that the data values of these lengths are not attacked.

Table 11 shows result of random malicious deletion for data values of length 3. As in deletion of digit frequency attack, a negative value of $\Delta\mathcal{F}l_j$ indicates that $\mathcal{F}$ fraction of length $l_j$ is maliciously deleted with characteristic of decreasing high data values to low in database relation. Also, as in malicious insertion, the $\Delta\mathcal{F}l_j$ is zero for lengths 1, 2, and 4, which indicates that the data values of these lengths are not deleted. Table 12 shows results for malicious updates on data values of length 3. In this attack, the data values of length 3 are randomly replaced by lengths 1, 2, and 4. This attack shows a decrease in $\Delta\mathcal{F}l_j$ for length 3, where as the $\Delta\mathcal{F}l_j$ for lengths 1, 2, and 4 is increased.

Figure 5 summarizes the malicious insertion, deletion, and update attacks on length 3 of data values. The insertion attack shows a positive increase (low to high) in attacked length, where as a negative trend (high to low) on attacked length is observed in deletion attack. In modification attack, a negative trend (high to low) is observed on attacked length, where as a positive trend (low to high) is observed on unattacked length of data values.

Table 13 shows the effect on fractional change in length frequency $\Delta\mathcal{F}l_j$, when 10% (lower bound) and 90% (upper bound) of tuples are maliciously inserted, deleted, and updated in the database relation. In insertion attack, the fractional change in length frequency $\Delta\mathcal{F}l_j$ has a $k$ fraction of positive trend (low to high) for each length of data values. Similarly, in deletion attack, a $k$ fraction of negative trend (high to low) is observed for each length of data values. For example, when 10% of tuples are randomly deleted from a database relation, a decrease of approximately 10% is observed in $\Delta\mathcal{F}l_j$ for each length of data values. The update attack does not show any specific trend as $k$ fraction of different length of data values are randomly replaced by some other length of data values.

It is to be noted that the attack on length of data values can be characterized on parameters like the length of data values being attacked, the fraction of each length of data values attacked, the type of attack (insertion, deletion, or update), and the effect of attack (low to high, high to low, or no change) on each length of data values.

*3.2.3. Attack on Range of Data Values.* In this set of attacks, Mallory randomly performs insertion, deletion, and update attack on range 1, that is, (100–999) of data values present in the database relation $R$. Table 14 shows the experimental results for characterization of malicious insertion for range 1

Table 9: Characterization of malicious modifications on digit frequency.

| | Insertion attack | | Deletion attack | | Update attack | |
|---|---|---|---|---|---|---|
| Attack rate | 10% | 90% | 10% | 90% | 10% | 90% |
| $d_i$ | $\Delta\mathscr{F}d_i\%$ | $\Delta\mathscr{F}d_i\%$ | $\Delta\mathscr{F}d_i\%$ | $\Delta\mathscr{F}d_i\%$ | $\Delta\mathscr{F}d_i\%$ | $\Delta\mathscr{F}d_i\%$ |
| 0 | +9.51 | +90.77% | −9.63% | −89.65% | +0.41 | +0.45 |
| 1 | +10.51 | +94.47% | −10.11% | −89.05% | +0.80 | +15.91 |
| 2 | +10.49 | +90.14% | −10.67% | −90.20% | −0.57 | −3.72 |
| 3 | +9.63 | +94.72% | −9.87% | −90.30% | −2.12 | +3.39 |
| 4 | +9.47 | +83.78% | −10.07% | −90.82% | −0.28 | −7.34 |
| 5 | +9.01 | +82.00% | −9.43% | −91.26% | +0.40 | −11.43 |
| 6 | +9.08 | +81.84% | −9.48% | −91.01% | +0.13 | −7.78 |
| 7 | +10.17 | +88.20% | −9.95% | −89.83% | +0.05 | +1.46 |
| 8 | +10.23 | +88.86% | −10.06% | −89.89% | −0.22 | +2.16 |
| 9 | +10.25 | +88.36% | −10.37% | −90.66% | −0.96 | −2.62 |

The detailed experiments for this set of attacks are presented in the Appendix (Tables 18(a)–18(f)).

Table 10: Characterization of malicious insertion attacks on length of data values.

| $l_j$ | $rfl_j$ | $fl_j$ | $rfl_j'$ | $fl_j'$ | $\Delta fl_j$ | $\Delta\mathscr{F}l_j$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 1 | 5.74 | 325894 | 5.27 | 325894 | 0 | 0 | No change |
| 2 | 20.19 | 1146469 | 18.54 | 1146469 | 0 | 0 | No change |
| **3** | **48.66** | **2762791** | **52.85** | **3267609** | **+504818** | **+18.27%** | **↑Low to High** |
| 4 | 25.41 | 1442906 | 23.34 | 1442906 | 0 | 0 | No change |

of data values. The characteristic of this attack is to relatively increase the low data values to high as an increase of 17.33% is observed in $\Delta\mathscr{F}r_k$ for range 1 of data values. The $\Delta\mathscr{F}r_k$ for range 0 and 2 is zero as the data values of these ranges are not attacked.

Table 15 shows the results of random malicious deletion for data values of range 1. As in deletion of digit frequency attack, a negative value of $\Delta\mathscr{F}r_k$ indicates that $\mathscr{F}$ fraction of range 1 is maliciously deleted with characteristic of transforming high data values to low in database relation $R$. As the data values of ranges 0 and 2 are not attacked, so the $\Delta\mathscr{F}r_k$ is zero for these ranges. Table 16 shows the results for malicious updates on data values of range 1. In this attack, the data values of range 1 are randomly replaced by ranges 0 and 2. This attack shows a decrease in $\Delta\mathscr{F}r_k$ for range 1, where as the $\Delta\mathscr{F}r_k$ for range 0 and 2 is increased.

The malicious insertion, deletion, and update attacks on range 1 of data values are summarized in Figure 6. A positive increase is observed in the attacked range for insertion attack (low to high) and a negative trend (high to low) is observed in attacked range for deletion attack. The modification attack shows a negative trend (high to low) for attacked range, that is, range 1 of data values and a positive increase for nonattacked ranges, that is, range 0 and 2 of data values.

In another set of attacks, we randomly inserted, deleted, and updated 10% (lower bound) and 90% (upper bound) of tuples from the database relation $R$. Table 17 shows the effect on fractional change in range frequency $\Delta\mathscr{F}r_k$, for each range of data values. The fractional change in range frequency $\Delta\mathscr{F}r_k$ has a $k$ fraction of positive trend (low to high) for malicious insertion in each range of data values. Similarly, in deletion attack, a $k$ fraction of negative trend (high to low) is observed for each range of data values. For example, when 10% of tuples are randomly deleted from a database relation, a decrease of approximately 10% is observed in $\Delta\mathscr{F}r_k$ for each range of data values. The update attack does not show any specific trend as $k$ fraction of different range of data values are randomly replaced by some other range of data values.

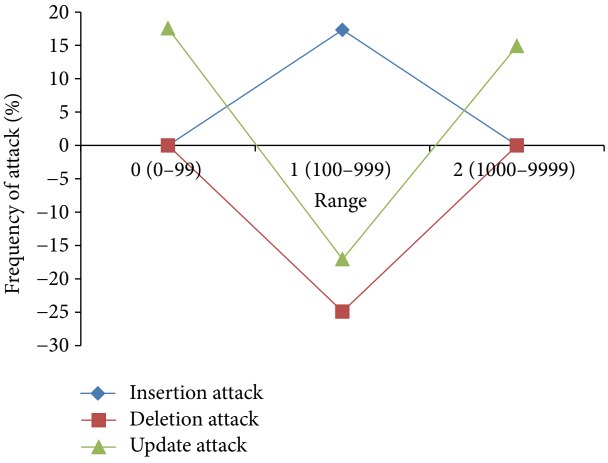


Figure 6: Characterization of malicious insertion, deletion, and update attacks on range 1 (100–999) of data values.

It is to be noted that the data characteristics used for our experiments like digit, length, and range of data values are cohesive to each other. Due to this relationship, we evaluated the effect of malicious data modifications on these three data characteristics. For example, if Mallory maliciously

Table 11: Characterization of malicious deletion attacks on length of data values.

| $l_j$ | $rfl_j$ | $fl_j$ | $rfl_j{}'$ | $fl_j{}'$ | $\Delta fl_j$ | $\Delta \mathcal{F}l_j$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 1 | 5.74 | 325894 | 6.52 | 325894 | 0 | 0.00 | No change |
| 2 | 20.19 | 1146469 | 22.92 | 1146469 | 0 | 0.00 | No change |
| **3** | **48.66** | **2762791** | **41.71** | **2085761** | **−677030** | **−24.51%** | **↓High to Low** |
| 4 | 25.41 | 1442906 | 28.85 | 1442906 | 0 | 0.00 | No change |

Table 12: Characterization of malicious update attacks on length of data values.

| $l_j$ | $rfl_j$ | $fl_j$ | $rfl_j{}'$ | $fl_j{}'$ | $\Delta fl_j$ | $\Delta \mathcal{F}l_j$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 1 | 5.74 | 325894 | 6.31 | 358142 | +32248 | +9.90% | ↑Low to High |
| 2 | 20.19 | 1146469 | 27.57 | 1565462 | +418993 | +36.55% | ↑Low to High |
| **3** | **48.66** | **2762791** | **34.27** | **1945657** | **−817134** | **−29.58%** | **↓High to Low** |
| 4 | 25.41 | 1442906 | 31.86 | 1808799 | +365893 | +25.36% | ↑Low to High |

Table 13: Characterization of malicious modifications on length of data values.

| | Insertion attack | | Deletion attack | | Update attack | |
|---|---|---|---|---|---|---|
| Attack rate | 10% | 90% | 10% | 90% | 10% | 90% |
| $l_j$ | $\Delta \mathcal{F}l_j$% | $\Delta \mathcal{F}l_j$% | $\Delta \mathcal{F}l_j$% | $\Delta \mathcal{F}l_j$% | $\Delta \mathcal{F}l_j$% | $\Delta \mathcal{F}l_j$% |
| 1 | +9.55 | 100.58 | −8.94 | −91.25 | −3.65 | −38.67 |
| 2 | +10.20 | 89.64 | −10.21 | −89.28 | 1.89 | 6.13 |
| 3 | +10.35 | 89.91 | −9.98 | −88.92 | 1.57 | 11.03 |
| 4 | +9.41 | 87.71 | −10.16 | −91.93 | −2.94 | −12.33 |

The detailed experiments for this set of attacks are presented in the Appendix (Tables 19(a)–19(f)).

Table 14: Characterization of malicious insertion attacks on range of data values.

| Range | $r_k$ | $rfr_k$ | $fr_k$ | $rfr_k{}'$ | $fr_k{}'$ | $\Delta fr_k$ | $\Delta \mathcal{F}r_k$ | Characteristic |
|---|---|---|---|---|---|---|---|---|
| 0 | 0–99 | 25.59 | 1436986 | 23.60 | 1436986 | 0 | 0 | No change |
| **1** | **100–999** | **48.73** | **2736731** | **52.72** | **3210988** | **+474257** | **+17.33%** | **↑Low to High** |
| 2 | 1000–9999 | 25.68 | 1442223 | 23.68 | 1442223 | 0 | 0 | No change |

Table 15: Characterization of malicious deletion attacks on range of data values.

| Range | $r_k$ | $rfr_k$ | $fr_k$ | $rfr_k{}'$ | $fr_k{}'$ | $\Delta fr_k$ | $\Delta \mathcal{F}r_k$ | Characteristic |
|---|---|---|---|---|---|---|---|---|
| 0 | 0–99 | 25.59 | 1436986 | 29.12 | 1436986 | 0 | 0 | No change |
| **1** | **100–999** | **48.73** | **2736731** | **41.65** | **2054875** | **−681856** | **−24.92%** | **↓High to Low** |
| 2 | 1000–9999 | 25.68 | 1442223 | 29.23 | 1442223 | 0 | 0 | No change |

Table 16: Characterization of malicious update attacks on range of data values.

| Range | $r_k$ | $rfr_k$ | $fr_k$ | $rfr_k{}'$ | $fr_k{}'$ | $\Delta fr_k$ | $\Delta \mathcal{F}r_k$ | Characteristic |
|---|---|---|---|---|---|---|---|---|
| 0 | 0–99 | 25.59 | 1436986 | 30.07 | 1688965 | 251979 | +17.54% | ↑Low to High |
| **1** | **100–999** | **48.73** | **2736731** | **40.42** | **2269854** | **−466877** | **−17.06%** | **↓High to Low** |
| 2 | 1000–9999 | 25.68 | 1442223 | 29.51 | 1657121 | 214898 | +14.91% | ↑Low to High |

Table 17: Characterization of malicious data modifications on range of data values.

| | | Insertion attack | | Deletion attack | | Update attack | |
|---|---|---|---|---|---|---|---|
| Attack rate | | 10% | 90% | 10% | 90% | 10% | 90% |
| Range | $r_k$ | $\Delta \mathcal{F}r_k$% | $\Delta \mathcal{F}r_k$% | $\Delta \mathcal{F}r_k$% | $\Delta \mathcal{F}r_k$% | $\Delta \mathcal{F}r_k$% | $\Delta \mathcal{F}r_k$% |
| 0 | 0–99 | +9.97 | +92.33 | −9.84 | −89.76 | 0.39 | −5.91 |
| 1 | 100–999 | +10.34 | +89.86 | −9.99 | −88.90 | 1.48 | 10.46 |
| 2 | 1000–9999 | +9.41 | +87.71 | −10.16 | −91.93 | −2.94 | −12.41 |

The detailed experiments for this set of attacks are presented in the Appendix (Tables 20(a)–20(f)).

Table 18: Characterization of malicious attacks on digit frequency (deterministic).

(a) Characterization of malicious insertion attacks on digit frequency ($\eta = 10^6$, attack rate = 10%)

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i'$ | $fd_i'$ | $\Delta fd_i$ | $\Delta \mathcal{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 0 | 8.44 | 247345 | 8.40 | 270858 | 23513 | +9.51% | ↑Low to High |
| 1 | 16.28 | 477362 | 16.37 | 527535 | 50173 | +10.51% | ↑Low to High |
| 2 | 19.88 | 582616 | 19.97 | 643745 | 61129 | +10.49% | ↑Low to High |
| 3 | 11.37 | 333378 | 11.34 | 365471 | 32093 | +9.63% | ↑Low to High |
| 4 | 8.83 | 258748 | 8.79 | 283251 | 24503 | +9.47% | ↑Low to High |
| 5 | 8.31 | 243554 | 8.24 | 265510 | 21956 | +9.01% | ↑Low to High |
| 6 | 7.14 | 209399 | 7.09 | 228418 | 19019 | +9.08% | ↑Low to High |
| 7 | 6.40 | 187586 | 6.41 | 206659 | 19073 | +10.17% | ↑Low to High |
| 8 | 6.49 | 190268 | 6.51 | 209737 | 19469 | +10.23% | ↑Low to High |
| 9 | 6.86 | 201050 | 6.88 | 221653 | 20603 | +10.25% | ↑Low to High |

(b) Characterization of malicious insertion attacks on digit frequency ($\eta = 10^6$, attack rate = 90%)

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i'$ | $fd_i'$ | $\Delta fd_i$ | $\Delta \mathcal{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 0 | 8.44 | 247345 | 8.51 | 471852 | 224507 | +90.77% | ↑Low to High |
| 1 | 16.28 | 477362 | 16.73 | 928331 | 450969 | +94.47% | ↑Low to High |
| 2 | 19.88 | 582616 | 19.97 | 1107797 | 525181 | +90.14% | ↑Low to High |
| 3 | 11.37 | 333378 | 11.70 | 649150 | 315772 | +94.72% | ↑Low to High |
| 4 | 8.83 | 258748 | 8.57 | 475532 | 216784 | +83.78% | ↑Low to High |
| 5 | 8.31 | 243554 | 7.99 | 443279 | 199725 | +82.00% | ↑Low to High |
| 6 | 7.14 | 209399 | 6.86 | 380768 | 171369 | +81.84% | ↑Low to High |
| 7 | 6.40 | 187586 | 6.36 | 353031 | 165445 | +88.20% | ↑Low to High |
| 8 | 6.49 | 190268 | 6.48 | 359339 | 169071 | +88.86% | ↑Low to High |
| 9 | 6.86 | 201050 | 6.83 | 378696 | 177646 | +88.36% | ↑Low to High |

(c) Characterization of malicious deletion attacks on digit frequency ($\eta = 10^6$, attack rate = 10%)

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i'$ | $fd_i'$ | $\Delta fd_i$ | $\Delta \mathcal{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 0 | 8.44 | 247345 | 8.48 | 223528 | −23817 | −9.63% | ↓High to Low |
| 1 | 16.28 | 477362 | 16.27 | 429114 | −48248 | −10.11% | ↓High to Low |
| 2 | 19.88 | 582616 | 19.74 | 520475 | −62141 | −10.67% | ↓High to Low |
| 3 | 11.37 | 333378 | 11.40 | 300483 | −32895 | −9.87% | ↓High to Low |
| 4 | 8.83 | 258748 | 8.83 | 232703 | −26045 | −10.07% | ↓High to Low |
| 5 | 8.31 | 243554 | 8.37 | 220577 | −22977 | −9.43% | ↓High to Low |
| 6 | 7.14 | 209399 | 7.19 | 189555 | −19844 | −9.48% | ↓High to Low |
| 7 | 6.40 | 187586 | 6.41 | 168921 | −18665 | −9.95% | ↓High to Low |
| 8 | 6.49 | 190268 | 6.49 | 171131 | −19137 | −10.06% | ↓High to Low |
| 9 | 6.86 | 201050 | 6.83 | 180196 | −20854 | −10.37% | ↓High to Low |

(d) Characterization of malicious deletion attacks on digit frequency ($\eta = 10^6$, attack rate = 90%)

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i'$ | $fd_i'$ | $\Delta fd_i$ | $\Delta \mathcal{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 0 | 8.44 | 247345 | 8.89 | 25612 | −221733 | −89.65% | ↓High to Low |
| 1 | 16.28 | 477362 | 18.14 | 52276 | −425086 | −89.05% | ↓High to Low |
| 2 | 19.88 | 582616 | 19.80 | 57072 | −525544 | −90.20% | ↓High to Low |
| 3 | 11.37 | 333378 | 11.22 | 32342 | −301036 | −90.30% | ↓High to Low |
| 4 | 8.83 | 258748 | 8.24 | 23758 | −234990 | −90.82% | ↓High to Low |
| 5 | 8.31 | 243554 | 7.38 | 21281 | −222273 | −91.26% | ↓High to Low |

(d) Continued.

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i{}'$ | $fd_i{}'$ | $\Delta fd_i$ | $\Delta \mathscr{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 6 | 7.14 | 209399 | 6.53 | 18815 | −190584 | −91.01% | ↓High to Low |
| 7 | 6.40 | 187586 | 6.62 | 19071 | −168515 | −89.83% | ↓High to Low |
| 8 | 6.49 | 190268 | 6.67 | 19239 | −171029 | −89.89% | ↓High to Low |
| 9 | 6.86 | 201050 | 6.52 | 18780 | −182270 | −90.66% | ↓High to Low |

(e) Characterization of malicious update attacks on digit frequency ($\eta = 10^6$, attack rate = 10%)

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i{}'$ | $fd_i{}'$ | $\Delta fd_i$ | $\Delta \mathscr{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 0 | 8.438048 | 247345 | 8.49 | 248357 | 1012 | +0.41% | ↑Low to High |
| 1 | 16.28496 | 477362 | 16.46 | 481186 | 3824 | +0.80% | ↑Low to High |
| 2 | 19.87565 | 582616 | 19.81 | 579315 | −3301 | −0.57% | ↓High to Low |
| 3 | 11.37302 | 333378 | 11.16 | 326311 | −7067 | −2.12% | ↓High to Low |
| 4 | 8.827055 | 258748 | 8.82 | 258020 | −728 | −0.28% | ↓High to Low |
| 5 | 8.30872 | 243554 | 8.36 | 244519 | 965 | +0.40% | ↑Low to High |
| 6 | 7.143539 | 209399 | 7.17 | 209680 | 281 | +0.13% | ↑Low to High |
| 7 | 6.3994 | 187586 | 6.42 | 187678 | 92 | +0.05% | ↑Low to High |
| 8 | 6.490895 | 190268 | 6.49 | 189845 | −423 | −0.22% | ↓High to Low |
| 9 | 6.858718 | 201050 | 6.81 | 199111 | −1939 | −0.96% | ↓High to Low |

(f) Characterization of malicious update attacks on digit frequency ($\eta = 10^6$, attack rate = 90%)

| $d_i$ | $rfd_i$ | $fd_i$ | $rfd_i{}'$ | $fd_i{}'$ | $\Delta fd_i$ | $\Delta \mathscr{F} d_i$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 0 | 8.438048 | 247345 | 8.46 | 248462 | 1117 | +0.45% | ↑Low to High |
| 1 | 16.28496 | 477362 | 18.84 | 553334 | 75972 | +15.91% | ↑Low to High |
| 2 | 19.87565 | 582616 | 19.10 | 560965 | −21651 | −3.72% | ↓High to Low |
| 3 | 11.37302 | 333378 | 11.74 | 344665 | 11287 | +3.39% | ↑Low to High |
| 4 | 8.827055 | 258748 | 8.16 | 239767 | −18981 | −7.34% | ↓High to Low |
| 5 | 8.30872 | 243554 | 7.35 | 215727 | −27827 | −11.43% | ↓High to Low |
| 6 | 7.143539 | 209399 | 6.58 | 193116 | −16283 | −7.78% | ↓High to Low |
| 7 | 6.3994 | 187586 | 6.48 | 190330 | 2744 | +1.46% | ↑Low to High |
| 8 | 6.490895 | 190268 | 6.62 | 194384 | 4116 | +2.16% | ↑Low to High |
| 9 | 6.858718 | 201050 | 6.67 | 195785 | −5265 | −2.62% | ↓High to Low |

inserts a digit in a data value, the length and range of the data value are also increased. Similarly, if Mallory maliciously decreases the length of a data value, the digit count and range of the data value are also decreased (Tables 9, 13, and 17).

At the end, we summarize our findings and observations for characterization of malicious data modifications as follows.

(i) If there is a positive trend in fractional change $\Delta\mathscr{F}$ of data values in tempered database relation $R'$, it means that $\mathscr{F}$ fraction of digit, range, and length of data values is maliciously inserted by Mallory in Alice's watermarked relation $R$. The characteristic of this attack is to relatively increase the low data values to high in database relation $R$ (Tables 6, 10, and 14).

(ii) If there is a negative trend in fractional change $\Delta\mathscr{F}$ of data values in tempered database relation $R'$, it means that $\mathscr{F}$ fraction of digit, range, and length of data values is maliciously deleted by Mallory from Alice's watermarked relation $R$. The characteristic of this attack is to relatively decrease the high data values to low in database relation $R$ (Tables 7, 11, and 15).

(iii) If there is both positive and negative trends in fractional change $\Delta\mathscr{F}$ for digit, range, and length of data values in tempered database relation $R'$, it means that the negative trend fractional change $\Delta\mathscr{F}$ of data values is maliciously replaced (updated) by positive trend fractional change $\Delta\mathscr{F}$ of data values (Tables 8, 12, and 16).

(iv) If there is a uniform increase of $k$ in fractional change $\Delta\mathscr{F}$ of all data values in tempered database relation $R'$, it means that $k$ fraction of similar tuples is maliciously inserted by Mallory in Alice's watermarked relation $R$. The characteristic of this attack is to relatively increase the low data values to high in database relation $R$ (Tables 9, 13, and 17).

(v) If there is a uniform decrease of $k$ in fractional change $\Delta\mathscr{F}$ of all data values in tempered database relation $R'$, it means that $k$ fraction of tuples is maliciously deleted by Mallory from Alice's watermarked relation

Table 19: Characterization of malicious attacks on length of data values (deterministic).

(a) Characterization of malicious insertion attacks on length of data values ($\eta = 10^6$, attack rate = 10%)

| $l_j$ | $rfl_j$ | $fl_j$ | $rfl_j'$ | $fl_j'$ | $\Delta fl_j$ | $\Delta \mathcal{F} l_j$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 1 | 7.11 | 71085 | 7.08 | 77875 | 6790 | +9.55% | ↑Low to High |
| 2 | 19.76 | 197605 | 19.80 | 217767 | 20162 | +10.20% | ↑Low to High |
| 3 | 45.18 | 451787 | 45.32 | 498531 | 46744 | +10.35% | ↑Low to High |
| 4 | 27.95 | 279523 | 27.80 | 305827 | 26304 | +9.41% | ↑Low to High |

(b) Characterization of malicious insertion attacks on length of data values ($\eta = 10^6$, attack rate = 90%)

| $l_j$ | $rfl_j$ | $fl_j$ | $rfl_j'$ | $fl_j'$ | $\Delta fl_j$ | $\Delta \mathcal{F} l_j$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 1 | 7.11 | 71085 | 7.50 | 142584 | 71499 | +100.58% | ↑Low to High |
| 2 | 19.76 | 197605 | 19.72 | 374737 | 177132 | +89.64% | ↑Low to High |
| 3 | 45.18 | 451787 | 45.16 | 857986 | 406199 | +89.91% | ↑Low to High |
| 4 | 27.95 | 279523 | 27.62 | 524693 | 245170 | +87.71% | ↑Low to High |

(c) Characterization of malicious deletion attacks on length of data values ($\eta = 10^6$, attack rate = 10%)

| $l_j$ | $rfl_j$ | $fl_j$ | $rfl_j'$ | $fl_j'$ | $\Delta fl_j$ | $\Delta \mathcal{F} l_j$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 1 | 7.11 | 71085 | 7.19 | 64733 | −6352 | −8.94% | ↓High to Low |
| 2 | 19.76 | 197605 | 19.71 | 177421 | −20184 | −10.21% | ↓High to Low |
| 3 | 45.18 | 451787 | 45.19 | 406721 | −45066 | −9.98% | ↓High to Low |
| 4 | 27.95 | 279523 | 27.90 | 251125 | −28398 | −10.16% | ↓High to Low |

(d) Characterization of malicious deletion attacks on length of data values ($\eta = 10^6$, attack rate = 90%)

| $l_j$ | $rfl_j$ | $fl_j$ | $rfl_j'$ | $fl_j'$ | $\Delta fl_j$ | $\Delta \mathcal{F} l_j$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 1 | 7.11 | 71085 | 6.22 | 6221 | −64864 | −91.25% | ↓High to Low |
| 2 | 19.76 | 197605 | 21.19 | 21186 | −176419 | −89.28% | ↓High to Low |
| 3 | 45.18 | 451787 | 50.04 | 50039 | −401748 | −88.92% | ↓High to Low |
| 4 | 27.95 | 279523 | 22.55 | 22554 | −256969 | −91.93% | ↓High to Low |

(e) Characterization of malicious update attacks on length of data values ($\eta = 10^6$, attack rate = 10%)

| $l_j$ | $rfl_j$ | $fl_j$ | $rfl_j'$ | $fl_j'$ | $\Delta fl_j$ | $\Delta \mathcal{F} l_j$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 1 | 7.11 | 71085 | 6.85 | 68488 | −2597 | −3.65% | ↓High to Low |
| 2 | 19.76 | 197605 | 20.13 | 201333 | 3728 | +1.89% | ↑Low to High |
| 3 | 45.18 | 451787 | 45.89 | 458869 | 7082 | +1.57% | ↑Low to High |
| 4 | 27.95 | 279523 | 27.13 | 271310 | −8213 | −2.94% | ↓High to Low |

(f) Characterization of malicious update attacks on length of data values ($\eta = 10^6$, attack rate = 90%)

| $l_j$ | $rfl_j$ | $fl_j$ | $rfl_j'$ | $fl_j'$ | $\Delta fl_j$ | $\Delta \mathcal{F} l_j$ | Characteristic |
|---|---|---|---|---|---|---|---|
| 1 | 7.11 | 71085 | 4.36 | 43599 | −27486 | −38.67% | ↓High to Low |
| 2 | 19.76 | 197605 | 20.97 | 209711 | 12106 | +6.13% | ↑Low to High |
| 3 | 45.18 | 451787 | 50.16 | 501622 | 49835 | +11.03% | ↑Low to High |
| 4 | 27.95 | 279523 | 24.51 | 245068 | −34455 | −12.33% | ↓High to Low |

*R*. The characteristic of this attack is to relatively decrease the high data values to low in database relation *R* (Tables 9, 13, and 17).

## 4. Conclusions

In this paper, a fragile watermarking scheme to detect and characterize malicious tempering made in database relations is presented. The proposed scheme is based on zero watermarking approach that does not alter the database original content, and thus it overcomes the limitation of data integrity and data usability in existing watermarking schemes. In the proposed scheme, the watermarks are generated by using the local characteristics of database relation itself, like frequency distribution of various digits, lengths, and ranges of data values. This enables us to characterize the malicious modifications made to the database relations. Experimental results showed that the proposed scheme can detect and characterize malicious data modifications successfully. In the future, we intend to work on some other local characteristics

Table 20: Characterization of malicious attacks on range of data values (Deterministic).

(a) Characterization of malicious insertion attacks on range of data values ($\eta = 10^6$, attack rate = 10%)

| Range No | $r_k$ | $rfr_k$ | $fr_k$ | $rfr_k{}'$ | $fr_k{}'$ | $\Delta fr_k$ | $\Delta\mathcal{F}r_k$ | Characteristic |
|---|---|---|---|---|---|---|---|---|
| 0 | 0–99 | 25.59 | 1436986 | 26.49 | 288506 | 26164 | +9.97% | ↑Low to High |
| 1 | 100–999 | 48.73 | 2736731 | 45.43 | 494709 | 46373 | +10.34% | ↑Low to High |
| 2 | 1000–9999 | 25.68 | 1442223 | 28.08 | 305806 | 26304 | +9.41% | ↑Low to High |

(b) Characterization of malicious insertion attacks on range of data values ($\eta = 10^6$, attack rate = 90%)

| Range No | $r_k$ | $rfr_k$ | $fr_k$ | $rfr_k{}'$ | $fr_k{}'$ | $\Delta fr_k$ | $\Delta\mathcal{F}r_k$ | Characteristic |
|---|---|---|---|---|---|---|---|---|
| 0 | 0–99 | 25.59 | 1436986 | 26.83 | 504566 | 242224 | +92.33% | ↑Low to High |
| 1 | 100–999 | 48.73 | 2736731 | 45.27 | 851193 | 402857 | +89.86% | ↑Low to High |
| 2 | 1000–9999 | 25.68 | 1442223 | 27.90 | 524663 | 245161 | +87.71% | ↑Low to High |

(c) Characterization of malicious deletion attacks on range of data values ($\eta = 10^6$, attack rate = 10%)

| Range No | $r_k$ | $rfr_k$ | $fr_k$ | $rfr_k{}'$ | $fr_k{}'$ | $\Delta fr_k$ | $\Delta\mathcal{F}r_k$ | Characteristic |
|---|---|---|---|---|---|---|---|---|
| 0 | 0–99 | 25.59 | 1436986 | 26.54 | 236532 | −25810 | −9.84% | ↓High to Low |
| 1 | 100–999 | 48.73 | 2736731 | 45.28 | 403562 | −44774 | −9.99% | ↑Low to High |
| 2 | 1000–9999 | 25.68 | 1442223 | 28.18 | 251104 | −28398 | −10.16% | ↓High to Low |

(d) Characterization of malicious deletion attacks on range of data values ($\eta = 10^6$, attack rate = 90%)

| Range No | $r_k$ | $rfr_k$ | $fr_k$ | $rfr_k{}'$ | $fr_k{}'$ | $\Delta fr_k$ | $\Delta\mathcal{F}r_k$ | Characteristic |
|---|---|---|---|---|---|---|---|---|
| 0 | 0–99 | 25.59 | 1436986 | 27.08 | 26857 | −235485 | −89.76% | ↓High to Low |
| 1 | 100–999 | 48.73 | 2736731 | 50.18 | 49757 | −398579 | −88.90% | ↑Low to High |
| 2 | 1000–9999 | 25.68 | 1442223 | 22.74 | 22547 | −256955 | −91.93% | ↓High to Low |

(e) Characterization of malicious update attacks on range of data values ($\eta = 10^6$, attack rate = 10%)

| Range No | $r_k$ | $rfr_k$ | $fr_k$ | $rfr_k{}'$ | $fr_k{}'$ | $\Delta fr_k$ | $\Delta\mathcal{F}r_k$ | Characteristic |
|---|---|---|---|---|---|---|---|---|
| 0 | 0–99 | 25.59 | 1436986 | 26.61 | 263370 | 1028 | +0.39% | ↑Low to High |
| 1 | 100–999 | 48.73 | 2736731 | 45.97 | 454956 | 6620 | +1.48% | ↑Low to High |
| 2 | 1000–9999 | 25.68 | 1442223 | 27.41 | 271271 | −8231 | −2.94% | ↓High to Low |

(f) Characterization of malicious update attacks on range of data values ($\eta = 10^6$, attack rate = 90%)

| Range No | $r_k$ | $rfr_k$ | $fr_k$ | $rfr_k{}'$ | $fr_k{}'$ | $\Delta fr_k$ | $\Delta\mathcal{F}r_k$ | Characteristic |
|---|---|---|---|---|---|---|---|---|
| 0 | 0–99 | 25.59 | 1436986 | 25.01 | 246838 | −15504 | −5.91% | ↓High to Low |
| 1 | 100–999 | 48.73 | 2736731 | 50.18 | 495248 | 46912 | +10.46% | ↑Low to High |
| 2 | 1000–9999 | 25.68 | 1442223 | 24.81 | 244819 | −34683 | −12.41% | ↓High to Low |

of relational databases for watermark generation and to extend the proposed scheme to semifragile watermarking schemes.

## Appendix

(i) Characterization of malicious attacks on digit frequency (deterministic) (see Table 18).

(ii) Characterization of malicious attacks on length of data values (deterministic) (see Table 19).

(iii) Characterization of malicious attacks on range of data values (deterministic) (see Table 20).

## References

[1] J. Dittmann, P. Schmitt, E. Saar, J. Schwenk, and J. Ueberberg, “Combining digital watermarks and collusion secure fingerprints for digital images,” *SPIE Journal of Electronic Imaging*, vol. 9, no. 4, pp. 456–467, 2000.

[2] E. T. Lin and E. J. Delp, “Temporal synchronization in video watermarking,” *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 3007–3022, 2004.

[3] M. Chen, Y. He, and R. L. Lagendijk, “A fragile watermark error detection scheme for wireless video communications,” *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 201–211, 2005.

[4] F. Hartung, P. Eisert, and B. Girod, “Digital watermarking of mpeg-4 facial animation parameters,” *Computers and Graphics*, vol. 22, no. 4, pp. 425–435, 1998.

[5] R. Agrawal, P. J. Haas, and J. Kiernan, “Watermarking relational data: framework, algorithms and analysis,” *VLDB Journal*, vol. 12, no. 2, pp. 157–169, 2003.

[6] G. Gupta and J. Pieprzyk, “Database relation watermarking resilient against secondary watermarking attacks,” in *Proceedings of the 5th International Conference on Information Systems*

*Security (ICISS '09)*, vol. 5905 of *Lecture Notes in Computer Science*, pp. 222–236, Springer, Kolkata, India, 2009.

[7] Y. Zhang, X. Niu, and D. Zhao, "A method of protecting relational databases copyright with cloud watermark," *International Journal of Information and Communication Engineering*, vol. 1, pp. 337–341, 2005.

[8] F. Guo, J. Wang, Z. Zhang, X. Ye, and D. Li, "An improved algorithm to watermark numeric relational data," in *Proceedings of the 6th International Workshop on Information Security applications (WISA '05)*, vol. 3786 of *Lecture Notes in Computer Science*, pp. 138–149, Springer, Jeju Island, Republic of Korea, 2005.

[9] M. Huang, J. Cao, Z. Peng, and Y. Fang, "A new watermark mechanism for relational data," in *Proceedings of the 4th International Conference on Computer and Information Technology (CIT '04)*, pp. 946–950, IEEE Computer Society, Wuhan, China, 2004.

[10] T. Hu, G. Chen, K. Chen, and J. Dong, "Garwm: towards a generalized and adaptive watermark scheme for relational data," in *Proceedings of the 6th International Conference in Advances in Web-Age Information Management (WAIM '05)*, vol. 3739 of *Lecture Notes in Computer Science*, pp. 380–391, Springer, Hangzhou, China, 2005.

[11] R. Sion, "Proving ownership over categorical data," in *Proceedings of the 20th International Conference on Data Engineering (ICDE '04)*, pp. 584–595, IEEE Computer Society, Boston, Mass, USA, April 2004.

[12] R. Sion, M. Atallah, and S. Prabhakar, "Rights protection for categorical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 7, pp. 912–926, 2005.

[13] C. Wang, J. Wang, M. Zhou, G. Chen, and D. Li, "Atbam: an arnold transform based method on watermarking relational data," in *Proceedings of the International Conference on Multimedia and Ubiquitous Engineering (MUE '08)*, pp. 263–270, IEEE Computer Society, Beijing, China, 2008.

[14] X. Zhou, M. Huang, and Z. Peng, "An additive-attack-proof watermarking mechanism for databases copyrights protection using image," in *Proceedings of the ACM symposium on Applied Computing (SAC '07)*, pp. 254–258, ACM Press, Seoul, Republic of Korea, March 2007.

[15] A. Al-Haj and A. Odeh, "Robust and blind watermarking of relational database systems," *Journal of Computer Science*, vol. 4, no. 12, pp. 1024–1029, 2008.

[16] H. Wang, X. Cui, and Z. Cao, "A speech based algorithm for watermarking relational databases," in *Proceedings of the International Symposiums on Information Processing (ISIP '08)*, pp. 603–606, IEEE Computer Society, Moscow, Russia, 2008.

[17] H. Guo, Y. Li, A. Liu, and S. Jajodia, "A fragile watermarking scheme for detecting malicious modifications of database relations," *Information Sciences*, vol. 176, no. 10, pp. 1350–1378, 2006.

[18] H. Khataeimaragheh and H. Rashidi, "A novel watermarking scheme for detecting and recovering distortions in database tables," *International Journal of Database Management Systems*, vol. 2, no. 3, pp. 1–11, 2010.

[19] S. Iqbal, A. Rauf, S. Mahfooz, S. Khusro, and S. H. Shah, "Self-constructing fragile watermark algorithm for. relational database integrity proof," *World Applied Sciences Journal*, vol. 19, no. 9, pp. 1273–1277, 2012.

[20] V. Prasannakumari, "A robust tamperproof watermarking for data integrity in relational databases," *Research Journal of Information Technology*, vol. 1, no. 3, pp. 115–121, 2009.

[21] J. Lafaye, D. Gross-Amblard, C. Constantin, and M. Guerrouani, "WATERMILL: an optimized fingerprinting system for databases under constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 4, pp. 532–546, 2008.

[22] Y. Li, H. Guo, and S. Jajodia, "Tamper detection and localization for categorical data using fragile watermarks," in *Proceedings of the 4th ACM Workshop on Digital Rights Management (DRM '04)*, pp. 73–82, ACM Press, Washington, DC, USA, October 2004.

[23] I. Kamel, "A schema for protecting the integrity of databases," *Computers and Security*, vol. 28, no. 7, pp. 698–709, 2009.

[24] S. Bhattacharya and A. Cortesi, "A distortion free watermark framework for relational databases," in *Proceedings of the 4th International Conference on Software and Data Technologies (ICSOFT '09)*, pp. 229–234, INSTICC Press, Sofia, Bulgaria, July 2009.

[25] A. Hamadou, X. Sun, L. Gao, and S. A. Shah, "A fragile zero-watermarking technique for authentication of relational databases," *International Journal of Digital Content Technology and its Applications*, vol. 5, no. 5, pp. 189–200, 2011.

[26] Y. Li, V. Swarup, and S. Jajodia, "Fingerprinting relational databases: schemes and specialties," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 1, pp. 34–45, 2005.

[27] S. Liu, S. Wang, R. H. Deng, and W. Shao, "A block oriented fingerprinting scheme in relational database," in *Proceedings of the 7th International Conference in Information Security and Cryptology (ICISC '04)*, vol. 3506 of *Lecture Notes in Computer Science*, pp. 455–466, Springer, Seoul, Republic of Korea, 2004.

[28] F. Guo, J. Wang, and D. Li, "Fingerprinting relational databases," in *Proceedings of the ACM symposium on Applied Computing (SAC '06)*, pp. 487–492, ACM Press, Dijon, France, 2006.

[29] C. T. Li, F. M. Yang, and C. S. Lee, "Oblivious fragile watermarking scheme for image authentication," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3445–3448, IEEE Press, Orlando, Fla, USA, 2002.

[30] H. Yang, X. Sun, B. Wang, and Z. Qin, "An image-adaptive semi-fragile watermarking for image authentication and tamper detection," in *Proceedings of the International Conference on Computational Science and its Applications*, Kuala Lumpur, Malaysia, 2007.

[31] J. Hu, J. Huang, D. Huang, and Y. Q. Shi, "Image fragile watermarking based on fusion of multi-resolution tamper detection," *Electronics Letters*, vol. 38, no. 24, pp. 1512–1513, 2002.

[32] J. Zhang and H. Bin, "Fragile audio watermarking scheme based on sample mean sequence," in *Proceedings of the IEEE International Conference on Multimedia Technology (ICMT '11)*, pp. 333–336, 2011.

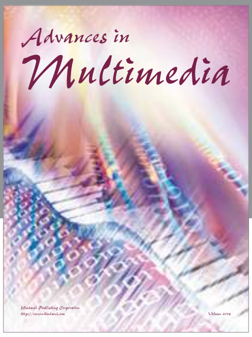
[33] H. Wang and M. Fan, "Centroid-based semi-fragile audio watermarking in hybrid domain," *Science in China F*, vol. 53, no. 3, pp. 619–633, 2010.

[34] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 6, pp. 774–778, 2007.

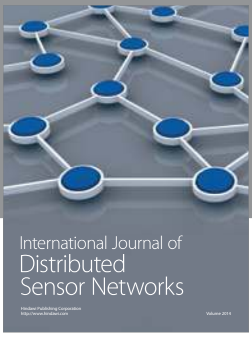
[35] X. Huazheng, S. Xingming, and T. Chengliang, "New fragile watermarking scheme for text documents authentication," *Wuhan University Journal of Natural Sciences*, vol. 11, no. 6, pp. 1661–1666, 2006.

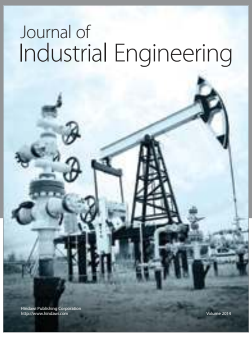
[36] H. Yang, A. C. Kot, and J. Liu, "Semi-fragile watermarking for text document images authentication," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, pp. 4002–4005, May 2005.

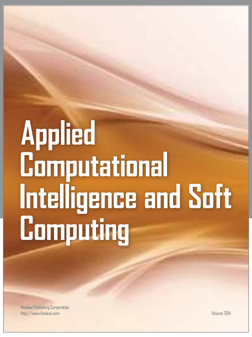
[37] http://archive.ics.uci.edu/ml/datasets/Covertype.

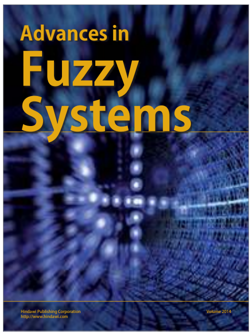

Advances in
Multimedia

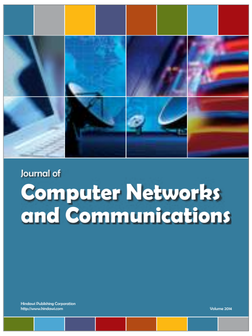
The Scientific
World Journal

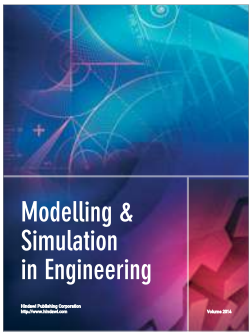

International Journal of
Distributed
Sensor Networks

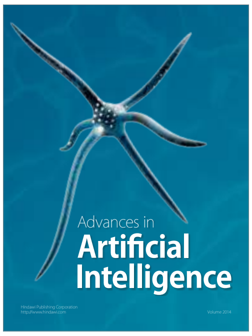

Journal of
Industrial Engineering

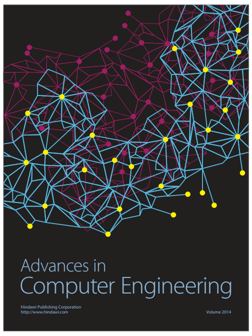

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

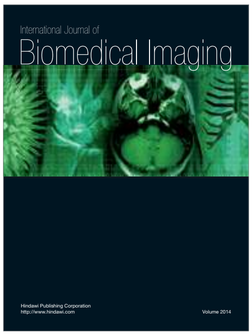

Journal of
Computer Networks
and Communications

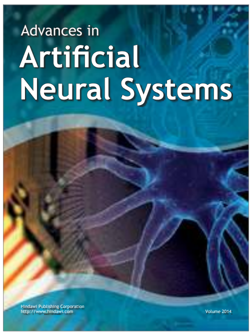

Modelling &
Simulation
in Engineering

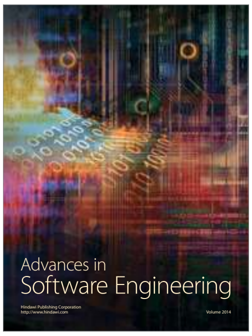

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

International Journal of
Computer Games
Technology

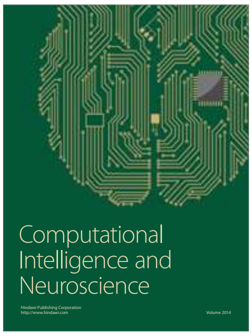

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics

Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering