A Reversible Watermarking Technique for Social Networks Datasets for Enabling Data Trust in Cyber, Physical and Social Computing

Saman Iftikhar, M. Kamran, Ehsan Ullah Munir, Member IEEE, and Samee U. Khan, Senior Member IEEE

Abstract—Social networks data is being mined for extracting interesting patterns. Such data is collected by different researchers and organizations and is usually also shared via different channels. This data usually has huge volume because there are millions of social networks users throughout the world. In this context, ownership protection of such datasets with huge volume becomes relevant. Digital watermarking is more demanding solution than any other technique for ensuring rights protection and integrity of the original datasets. The objective of this paper is to devise a reversible watermarking technique for the social networks data to prove ownership rights and also provide a mechanism for data recovery. Robustness of the proposed technique is evaluated through attack analysis using experimental study. In this paper, Z notation based formal specification is also provided to show the working of the proposed reversible watermarking technique for social networks datasets for enabling data trust in Cyber, Physical and Social Computing (CPSCom).

Index Terms—Digital Watermarking, Reversible Watermarking, Robust, Data Mining, Social Network Data.

1 INTRODUCTION

The use of social networks is increasing day by day. A recent statistics show that there were above 32 million social networks users worldwide [1]. The recent trends of the people all over the world indicates that this number would keep on increasing. The social networks’ users perform different actions such as: sharing their activities, news about their events, sharing news, sharing their personal data, discussing various topics and so on. Consequently, they are generating “Big Data” for social network computing – an essential component of Cyber, Physical and Social Computing (CPSCom). To extract and analyze useful information from this data, data mining and other knowledge extraction algorithms are applied [2], [3], [4], [5]. This work is usually done in collaboration and demands data sharing among different stakeholders; consequently, ownership protection of such data becomes relevant.

Nowadays social networks data is used excessively for statistical and data mining purposes. Reality Commons is an ongoing project in MIT Human Dynamics Lab [6] containing datasets about social communities such as: (1) Badge dataset; (2) Friends and Family; (3) Reality Mining; and (4) Social Evolution. These datasets were assembled as a result of two projects namely: an open-source sensing platform for Android phones and sociometric badges for sensing organizational behavior [7]. The digital ownership protection of such datasets is inevitable and highly demanding because an intruder – Mallory – may claim the false ownership of the data after acquiring access to the data and performing some modifications (malicious attacks) on the data. This provides us the motivation for the work reported in this paper.

For ownership protection of digital data, approaches like fingerprinting, data hashing, serial codes, and watermarking techniques are usually used. Fingerprints (or transactional watermarks) are used to monitor and identify the digital ownership by watermarking all copies of the same contents with different watermark for different recipients. The main purpose of such technique is to identify the source of leakage of data [8]. On the other hand, watermarking provides the ownership protection over the digital content by watermarking the data with one watermark and then proving the ownership of digital content after successful detection of the embedded watermark. This watermark can either be perceptible or imperceptible based on the requirements. A perceptible watermark is visible to the human user while an imperceptible watermark is not visible. A watermarking scheme should be robust and imperceptible. Furthermore, the watermark should be such that it can be

1. In information theory, Mallory is considered as an intruder that has some evil intentions.
2. Related Work and Motivation

This discussion presents the related work and the motivation behind this work.

2.1 Related Work

To the best of our knowledge, no work has been done for watermarking of social networks datasets yet. However, some closely relevant techniques include relational database watermarking techniques but they are either robust or reversible but not both. On the other hand, the proposed technique is both the robust and reversible.

The watermarking of relational databases primarily deals with the watermarking of numeric features, subject to the usability constraints. In this context, techniques, such as [18] and [19] require the presence of a primary key attribute to enforce ownership over the shared data. Primary keys are used in some watermarking techniques for various types of data. Its use is not dependent on any type of data and not required for every watermarking technique. In our context, social networks datasets do not need to have a primary key as the focus is not to identify each instance distinctly.

A number of recent techniques, such as [20], [21], [22], [23], [24] extend the work reported in [18] and embed a multi-bit watermark in selected Least Significant Bits (LSBs). The focus of all of the above mentioned techniques is towards the watermarking of relational databases and almost all of the techniques require a primary key for watermarking. However, often (if not always) there is no primary key or any other unique feature in social network datasets. Consequently, we consider our work to be novel.

2.2 Motivation

A scenario is demonstrated in Fig. 1 for the motivation of this work. A number of users are accessing data from the Reality Commons - a social network. The purpose is to perform data mining and other statistical measures over the provided datasets for useful analysis. It is possible that Mallory (or any of the user) may claim fake ownership over the dataset. Therefore, the ownership protection of all the given datasets in the social network is required. Digital watermarking and more specifically reversible watermarking is one good solution for ownership protection and data recovery in case of malicious intentions of the users.

3. Threat Model

This section systematically presents the facts regarding the system and adversary capabilities; thus, encompasses all the possible scenarios with the perspective of an attacker.
3.1 System Model

The owners of the social network datasets demand ownership protection for their shared data. Therefore, the proposed system is not allowed to modify the data in any way; otherwise, its meaning would be lost and it would become useless. The owners should be able to recover watermark and the original data. They do not need to keep a copy of the original data for two reasons: (1) watermark is embedded in the original data without changing values; and (2) watermark decoding does not require the original data and the original watermark for watermark decoding (or recovery). They should input: (1) numeric and non-numeric features; (2) a seed value for watermark string generation; (3) a secret key of number of characters and symbols for computing digram matrix; and, (4) a secret number of rounds (defined by the owner) for introducing randomness in the permutation matrix. There is no need to store digram and permutation matrix as these can be trivially regenerated while restoring watermark and the original data. They should define an integer value and a percent for watermark encoding and decoding. We assume that all the secret parameters and keys are not compromised and attackers cannot reproduce some or all of them. They will not share these secret parameters and keys with anyone.

3.2 Adversary Model

The intention of the adversary, Mallory, is to corrupt (or remove) the watermark from the marked dataset. It is assumed that in various types of malicious attacks, Mallory tries to destroy watermark W. These attacks consist of insertion, deletion, alteration, additive and counterfeiting attacks, as shown in Figure 2. The proposed technique of reversible watermarking should be robust against these attacks and the encoded watermark must be successfully extracted from the attacked dataset.

4 Proposed Technique

This section discusses the reversible watermarking of social network datasets. The main architecture of the proposed technique is presented in the Fig. 3. It includes the following four major phases: (1) preprocessing, (2) watermark encoding, (3) watermark decoding, and (4) data recovery.

For a quick reference, Table 1 lists the notations used in this paper.

4.1 Preprocessing

The preprocessing phase comprises three steps: (1) data selection; (2) feature selection; and (3) watermark creation. First, the dataset to be watermarked is selected. Next, the numeric or non-numeric feature is selected for watermarking. Then, a watermark is created through pseudo-random sequence generator to encode the selected feature of the selected dataset.

4.2 Watermark Encoding

After selecting the feature from the dataset, two further steps are performed for each type of selected feature before encoding watermark. In the first step, an evolutionary algorithm – Genetic Algorithm (GA) [25] – is used to create an optimum value to be embedded in the numeric type of dataset for ensuring robust watermark detection. In the second step, hashing and permutations are generated for non-numeric type of dataset to ensure reversible watermarking. After calculating a seeded watermark in step (3), watermark is embedded in each type of data in step (4) as shown in the Fig. 3.

4.3 Watermark Decoding

For watermark detection from the watermarked data; first, the preprocessing steps – hashing and permutation
are performed again for selected type of feature. Next, majority voting scheme is used to detect the watermark from the marked dataset on the basis of number of ‘1s’ and ‘0s’. Finally, the watermark is extracted from the whole dataset to prove ownership.

4.4 Data Recovery

In this phase, GA, hashing and permutation steps are performed again for selected type of feature. Data is recovered from the marked data of the selected feature type through employing GA, hashing and permutation after detecting the embedded watermark.

5 Formal Specification Model

Formal specification model includes schemas to describe the static and dynamic aspects of the proposed system. Schemas define states, invariants, operations and relationships between inputs and outputs. Metadata about different states and operations of the system are defined separately for each schema. Data owner requirements are defined in the form of properties and proposition using conjunction, disjunction, implication, equivalence and negation. Quantifiers used are: (1) for all ∀; and, (2) existential quantifier ∃. Mathematical notations for schema inclusion, used in this paper are ∆ and Ξ.

The formal specification model comprises SystemInfo, FeatureSelection, EvolutionaryAlgorithm, HashingPermutation, WatermarkCreation, WatermarkEncoding, WatermarkDecoding, and DataRecovery schemas.

5.1 The Proposed System

It is a system that provides ownership protection of numeric relational data and restores the original data without distortions. The proposed technique allows any social network data owners to watermark their datasets by giving some secret parameters. The system specification is provided in this schema by defining SETS for OWNER-ID, NAME, WATERMARK-ID, PASSWORD, DATA, WATERMARK-DATA, De-WATERMARK-DATA, FEATURES, STRING, λ, and β. The metadata of the proposed system is defined in Table 2, and its schema is given below. All the variables of the proposed system are declared in the SystemInfo schema.

<table>
<thead>
<tr>
<th>Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Set of Original Datasets</td>
</tr>
<tr>
<td>WATERMARK-DATA</td>
<td>Set of Data that is Watermarked</td>
</tr>
<tr>
<td>De-WATERMARK-DATA</td>
<td>Set of Data that is Decoded</td>
</tr>
<tr>
<td>FEATURES</td>
<td>Set of features of Datasets</td>
</tr>
<tr>
<td>STRING</td>
<td>Set of messages that are shown in the system</td>
</tr>
<tr>
<td>S[ω]</td>
<td>Set of Seed vectors for PseudoRandom Sequence Generator</td>
</tr>
<tr>
<td>λ</td>
<td>Set of usability constraints defined by OWNER</td>
</tr>
<tr>
<td>κ</td>
<td>Set of secret number of rounds for permutations defined by the owner</td>
</tr>
<tr>
<td>β</td>
<td>Set of Optimized values to Watermark the Data</td>
</tr>
</tbody>
</table>

[DATA, WATERMARK-DATA, De-WATERMARK-DATA, FEATURES, STRING, S[ω], κ, λ, β]
5.2 Feature Selection
Schema of the Feature Selection step of preprocessing phase is defined below. In the schema $F$ represents the numeric or non-numeric feature.

$$\text{FeatureSelection}$$

\[ [\text{SystemInfo}] \]

Display! = Select Features to set in your Data
Display! = Feature Selection Completed

5.3 Evolutionary Algorithm
Schema of the Evolutionary Algorithm sub-step of preprocessing phase is defined below: [httpb]

$$\text{EvolutionaryAlgorithm}$$

\[ [\text{SystemInfo}] \]

Display! = EncodingAlgorithm
Display! = Step Completed

5.4 Hashing and Permutation
Schema of the Hashing and Permutation sub-step of preprocessing phase is defined below. In the schema $P[N]$ represents a $1 \times 10$ vector and $P[\Delta\Sigma]$ represents all possible permutations.

$$\text{HashingPermutation}$$

\[ [\text{SystemInfo}] \]

Display! = hashing generated
Display! = Permutations generated

5.5 Watermark Creation
Schema of the Watermark Creation step of preprocessing phase is defined below:

$$\text{WatermarkCreation}$$

\[ [\text{SystemInfo}] \]

Display! = Step Completed

5.6 Watermark Encoding
Schema of the Watermark Encoding phase is defined below:

$$\text{WatermarkEncoding}$$

\[ [\text{SystemInfo}] \]

Display! = Step Completed

5.7 Watermark Decoding
Schema of the Watermark Decoding phase is defined below:
5.8 Data Recovery

Schema of the Data Recovery phase is defined below:

\[ \Delta \text{DataRecovery} \]
\[ \Xi \text{SystemInfo} \]
\[ \Xi \text{FeatureSelection} \]
\[ \Delta \text{EvolutionaryAlgorithm} \]
\[ \Delta \text{HashingPermutation} \]
\[ \Delta \text{WatermarkEncoding} \]
\[ \text{DRA}? : \text{DecodingAlgorithm} \]

Change? : Variable of type \( \Delta \) 
Counter = WATERMARK \( \rightarrow \) Length

Display1! : STRING

Loop of Counter :
Step1
\[ \eta_D = D'_W + \beta \ast \gamma \]
\[ \eta_D = \eta - \eta \]
CASE \( \eta_D \leq 0 \) : \( \text{THEN} \ w \rightarrow 1 \)
ELSE CASE \( \eta_D > 0 \) \( \wedge \eta_D \leq 1 \) : \( \text{THEN} \ w \rightarrow 0 \)
Step2
\[ \Delta_DW = (\gamma(DW(i)) \ast k + \gamma(DW(i+1)) + \zeta) \ast \tau \]
\[ \Delta_DW = \Delta_DW - \Delta_D \]
CASE \( \Delta_DW \leq 0 \) : \( \text{THEN} \ w \rightarrow 1 \)
ELSE CASE \( \Delta_DW > 0 \) \( \wedge \Delta_DW \leq 1 \) : \( \text{THEN} \ w \rightarrow 0 \)
WATERMARK = DATA' =
\[ \text{De} \rightarrow \text{WATERMARK} \rightarrow \text{DATA} \]

Display1! = Step Completed

6 RESULTS AND DISCUSSION

In this section, the proposed technique has been evaluated for providing: (1) reversible watermarking; and (2) robustness against malicious attacks. For brevity, experiments have been reported with a Badge dataset containing performance and dynamics of a real world organization. A relatively small watermark that consists of 8-bits is used in the analysis. The dataset consists of four features including: (1) BID (badges identified by unique numbers assigned to the employees), (2)(3) \( x \) and \( y \) (locations of employees’ cubicles and anchor nodes); and (4) roles of employees. The dataset has been shown in Table 3 with only three records for brevity.

<table>
<thead>
<tr>
<th>BID</th>
<th>( x )</th>
<th>( y )</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>266</td>
<td>5895.2</td>
<td>3075.8</td>
<td>Pricing</td>
</tr>
<tr>
<td>270</td>
<td>6312</td>
<td>3105.8</td>
<td>Configuration</td>
</tr>
<tr>
<td>291</td>
<td>592.3</td>
<td>3105.8</td>
<td>Coordinator</td>
</tr>
</tbody>
</table>

The proposed technique handles big data in a systematic way with logical grouping. For instance, for non-numeric data encoding, similar records for the same employees get hashed to same values and make logical groups; so, the data size do not increase. Moreover, the data owner has more interest in acquiring ownership rights and data recovery. She gets her data watermarked only once and watermarking is usually done offline, that is to say: it is done on the machine of the data owner and data is delivered to recipient only after completing the embedding process; so, she can afford large computation time for providing ownership rights and data recovery. For datasets involving large number of features or large number of rows “Big Data”, data owner may use a separate machine, with high computation power, for watermarking the datasets. This might incur some cost but gain the owner more security (false claim of ownership can be tackled by watermark encoding and decoding).

The computational time of the proposed technique is \((\omega \ast R \ast F)\) where \( \omega \) is the watermark length, \( R \) is the total number of tuples in the dataset and \( F \) is the feature selected for watermarking. The number of tuples are usually much larger as compared to the number of features in databases and the watermark length \( \omega \); so, \( F \ll R \) and \( \omega \ll R \). Therefore, for large databases, \( R \) termed as \( n \) the time complexity of the proposed technique for watermark encoding and decoding is \( O(n) \) (it is worth mentioning here that the time for computing the GA based optimal value is not included in this calculation because it is part of pre-processing.)

6.1 Reversible Watermarking

The preprocessing phase, watermark creation, watermark encoding, decoding and data recovery are demonstrated, to give an insight of how the proposed technique works.

6.1.1 Preprocessing Phase

In this phase, a dataset (Badge Dataset) and a numeric and non-numeric feature was selected for watermarking. A watermark string of 8-bits was generated with seeded pseudo-random sequence generator to encode all the...
tuples of the selected features. An optimum value of $\beta = 1$ is calculated by using an optimization scheme. This value might be different for different datasets. Sufficient experiments were performed to find out the most reliable set of parameters for GA. The detailed set of GA parameters found reliable are given in Table 4. The value of $\zeta = 0.1$ for both feature types.

### TABLE 4

<table>
<thead>
<tr>
<th>GA Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Generations</td>
<td>50</td>
</tr>
<tr>
<td>Population Size</td>
<td>10</td>
</tr>
<tr>
<td>Chromosome Length</td>
<td>8</td>
</tr>
<tr>
<td>Selection Mechanism</td>
<td>Tournament Selection</td>
</tr>
<tr>
<td>Tournament Size</td>
<td>$\Delta_\Sigma$</td>
</tr>
<tr>
<td>Crossover</td>
<td>Type: Single Point</td>
</tr>
<tr>
<td>Fraction</td>
<td>0.7</td>
</tr>
<tr>
<td>Mutation</td>
<td>Type: Uniform</td>
</tr>
<tr>
<td>Rate</td>
<td>0.1</td>
</tr>
<tr>
<td>Elitism Count</td>
<td>2</td>
</tr>
</tbody>
</table>

All the possible digrams $\Delta_\Sigma$ for the selected dataset were computed with all the possible pair combinations and hashing of the selected feature and permutation matrix $P[\Delta_\Sigma]$ was generated from the digram matrix $\Delta_\Sigma$. Permutations are performed to substitute the original characters with permuted characters [26]. The permutation matrix $P[\Delta_\Sigma]$ was then computed algorithmically by using previously calculated matrix of digrams $\Delta_\Sigma$. Thereafter, permutations are performed. A $1 \times 10$ vector $P[N]$ is initialized randomly that introduces randomness in the resultant permutation matrix $P[\Delta_\Sigma]$ by manipulating $P[j]$ and $P[j+1]$ values from $P[N]$ alternatively. The permutations were computed for a pre-specified number of times according to a secret number - $\kappa$, specified by the data owner. This step is performed to bring randomness in the initial digrams matrix $\Delta_\Sigma$.

#### 6.1.2 Watermark Encoding

In this phase, Badge dataset was taken under consideration, with only 3 tuples of the selected feature: BID as a numeric feature and role as a non-numeric feature to show the whole procedure of watermark encoding. The watermark encoding process has been explained through Tables 5 and 6. Numeric values were marked with a novel proposed reversible watermarking technique where the original data recovered fully without modifications. For non-numeric data, a digram $Pr$ from pricing – the value of the first tuple was encoded with watermark bits. Watermark bits have been represented with $b$ of length 8. For every watermark bit the watermarked data $D_W$ and percentage change $\eta_p$ in the original values were computed. Similarly, for permuted digrams $\xi_W$ and changes in data values $\Delta_D$ were computed for digram $Pr$. The same procedure was repeated for the rest of the tuples for the selected features.

The watermarked dataset after watermark encoding has been shown in Table 7.

### 6.1.3 Watermark Decoding

To show the intuition to how the decoding algorithm works, the same 3 tuples of the selected features were considered. The steps of watermark decoding have been shown in Tables 8 and 9. The changes in the original data values $\Delta_D$ computed during the encoding phase were used in the decoding phase. The changes in the encoded data values $\Delta_{DW}$ were computed while decoding the watermark from the watermarked data. The difference in change in the original data values and the encoded data values $\Delta_{Dig}$ was computed as given in schemas above. On the basis of these computed differences, watermark bits $b$ were extracted as it is for numeric data and in reverse order for non-numeric data.

### TABLE 5

**Watermark Encoding (Numeric Data)**

<table>
<thead>
<tr>
<th>BID</th>
<th>$\Delta_D$</th>
<th>$\Delta_{DW}$</th>
<th>$\Delta_{Diff}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>266</td>
<td>26.7</td>
<td>26.6</td>
<td>0.1</td>
</tr>
<tr>
<td>276</td>
<td>27.7</td>
<td>27.6</td>
<td>0.1</td>
</tr>
<tr>
<td>291</td>
<td>29.2</td>
<td>29.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

### TABLE 6

**Watermark Encoding (Non-Numeric Data)**

<table>
<thead>
<tr>
<th>Role</th>
<th>$\xi_W$</th>
<th>$\eta_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr</td>
<td>BI</td>
<td>292</td>
</tr>
<tr>
<td>Co</td>
<td>QD</td>
<td>291</td>
</tr>
</tbody>
</table>

### TABLE 7

**Badge Dataset (Watermarked Dataset)**

<table>
<thead>
<tr>
<th>BID</th>
<th>x</th>
<th>y</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>266</td>
<td>268.5</td>
<td>267.8</td>
<td>Pricing</td>
</tr>
<tr>
<td>276</td>
<td>267.8</td>
<td>267.9</td>
<td>Configuration</td>
</tr>
<tr>
<td>291</td>
<td>267.8</td>
<td>267.9</td>
<td>Coordinator</td>
</tr>
</tbody>
</table>

### TABLE 8

**Watermark Decoding (Numeric Data)**

<table>
<thead>
<tr>
<th>BID</th>
<th>$\Delta_D$</th>
<th>$\Delta_{DW}$</th>
<th>$\Delta_{Diff}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>266</td>
<td>26.7</td>
<td>26.6</td>
<td>0.1</td>
</tr>
<tr>
<td>276</td>
<td>27.7</td>
<td>27.6</td>
<td>0.1</td>
</tr>
<tr>
<td>291</td>
<td>29.2</td>
<td>29.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BID</th>
<th>$\Delta_D$</th>
<th>$\Delta_{DW}$</th>
<th>$\Delta_{Diff}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>266</td>
<td>26.7</td>
<td>26.6</td>
<td>0.1</td>
</tr>
<tr>
<td>276</td>
<td>27.7</td>
<td>27.6</td>
<td>0.1</td>
</tr>
<tr>
<td>291</td>
<td>29.2</td>
<td>29.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
6.1.4 Data Recovery

After detecting watermark string, post processing steps were carried out for error correction and data recovery. The optimized value of \( \beta \) computed through GA is used for regeneration of numeric data. The original data \( D_{r} \) and the permuted digrams \( \xi_{D} \) for the digram \( Pr, Co \) were decoded accordingly. Finally, the original data value \( pricing \) was computed from permuted digrams. The same procedure was repeated for the rest of the tuples for the selected features as shown in Tables 8 and 9. The original dataset has been recovered fully and same as shown in the Table 3

6.2 Robustness Analysis

Robustness analysis of the proposed technique is started with a supposition that the ownership protection of the social networks’ datasets is inevitable against the threat model (presented in Section 3). The datasets are shared in the social networks. Mallory has malicious intentions and she make several attempts to corrupt the data and the watermark encoded in the data.

Robustness of the proposed technique was examined through an extensive attack analysis. Our results showed high accuracy for the watermark detection with insertion, alteration and deletion attacks. We believe that this is due to the fact that the optimum GA parameters made sure that the maximum decoding accuracy is achieved by introducing sufficient randomness during watermark embedding. The GA do not guarantee the optimum result; however, due to their evolutionary nature, the tend to find a better solution for the given problem.

In the robustness study, we examined our technique against tuple insertion, deletion and insertion attacks. Mallory tries to insert, alter or delete 10\%, 20\%, 30\%, 40\%, 50\%, ..., 90\% of data. After such attacks, the proposed technique recovered 100\% tuples in all the cases. In case of tuple insertion and deletion attacks, the decoding accuracy is higher because Mallory is not quite “affecting” the watermarked tuples because the watermark embedding takes into account all the original tuples and hence insertion of new tuples or deletion of some tuples do not affect the watermark in the un-attacked tuples. On the other hand, in case of tuple alteration attacks, the robustness may suffer with the attacks that target large number of tuples because the embedded watermark is directly dependent on the values of the tuples. Moreover, the combination of tuple insertion, deletion and alteration attacks will also affect the accuracy of the proposed technique.

6.2.1 Insertion Attacks

In this type of attack, Mallory inserts new tuples to corrupt the watermark embedded in the data. Insertion of new tuples do not destroy the data integrity and the embedded watermark but may affect the watermark detection rate. The proposed technique was observed to be highly resilient against these types of attacks and recovered the watermark with 100\% accuracy even if Mallory inserts 100\% new and fake tuples. Data recovery has been shown in Fig. 4 for the proposed technique. The watermark decoding success rate was demonstrated in Fig. 5.

![Fig. 4. Data Recovery after Insertion Attack](image)

![Fig. 5. Watermark Decoding Accuracy of the proposed technique with Insertion Attack](image)

6.2.2 Deletion Attacks

In such attacks, Mallory deletes a subset of watermarked tuples from the datasets to corrupt the watermark. The watermark was encoded in the permuted digrams of \( n \) tuple, so the watermark was extracted even from a permuted digram of a single tuple. Mallory is unable to detect watermark as she is not aware of permuted digrams; therefore, she has the only choice to delete tuples with the concern of preserving the data usefulness of the remaining tuples. Experiments were performed to show the data recovery and the watermark detection. If she can delete \( n - 1 \) tuples, watermark would be restored from the remaining 1 tuple of the dataset. In

### Table 9

<table>
<thead>
<tr>
<th>Role</th>
<th>( \Delta_{D} )</th>
<th>( \Delta_{DW} )</th>
<th>( \Delta_{Eff} )</th>
<th>( \delta )</th>
<th>( \xi_{D} )</th>
<th>( \Delta_{D} )</th>
<th>( \Delta_{DW} )</th>
<th>( \Delta_{Eff} )</th>
<th>( \delta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr</td>
<td>42.0</td>
<td>42.0</td>
<td>42.0</td>
<td>42.0</td>
<td>42.0</td>
<td>42.0</td>
<td>42.0</td>
<td>42.0</td>
<td>42.0</td>
</tr>
<tr>
<td>Co</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>( r )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \xi_{D} )</td>
<td>UU, UU, MS, MS, MS, MS, UU, MS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 10

<table>
<thead>
<tr>
<th>Role</th>
<th>( \Delta_{D} )</th>
<th>( \Delta_{DW} )</th>
<th>( \Delta_{Eff} )</th>
<th>( \delta )</th>
<th>( \xi_{D} )</th>
<th>( \Delta_{D} )</th>
<th>( \Delta_{DW} )</th>
<th>( \Delta_{Eff} )</th>
<th>( \delta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr</td>
<td>6.6</td>
<td>6.6</td>
<td>6.6</td>
<td>6.6</td>
<td>6.6</td>
<td>6.6</td>
<td>6.6</td>
<td>6.6</td>
<td>6.6</td>
</tr>
<tr>
<td>Co</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>( r )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \xi_{D} )</td>
<td>QD, QD, B, B, B, B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 11

<table>
<thead>
<tr>
<th>Role</th>
<th>( \Delta_{D} )</th>
<th>( \Delta_{DW} )</th>
<th>( \Delta_{Eff} )</th>
<th>( \delta )</th>
<th>( \xi_{D} )</th>
<th>( \Delta_{D} )</th>
<th>( \Delta_{DW} )</th>
<th>( \Delta_{Eff} )</th>
<th>( \delta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>Co</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>( r )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \xi_{D} )</td>
<td>QD, QD, B, B, B, B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 12

<table>
<thead>
<tr>
<th>Role</th>
<th>( \Delta_{D} )</th>
<th>( \Delta_{DW} )</th>
<th>( \Delta_{Eff} )</th>
<th>( \delta )</th>
<th>( \xi_{D} )</th>
<th>( \Delta_{D} )</th>
<th>( \Delta_{DW} )</th>
<th>( \Delta_{Eff} )</th>
<th>( \delta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>Co</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>( r )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \xi_{D} )</td>
<td>QD, QD, B, B, B, B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
experiments, up to 90% of the data was deleted, so the watermark was recovered with 100% accuracy as shown in Fig. 7. Data recovery was also observed with various success rates under various ranges of deletion attacks as shown in Fig. 6.

![Fig. 6. Data Recovery after Deletion Attack](image)

![Fig. 7. Watermark Decoding Accuracy of the proposed technique with Deletion Attack](image)

6.2.3 Alteration Attacks

In such attacks, Mallory alters the values of the watermarked data to try to corrupt the watermark. She makes modifications into the data with the intention of destroying the encoded watermark. We experimented our technique with such type of attack and observed that the original data was recovered with this type of attack with high success rate. If she can alter \( n - 1 \) tuples, watermark and the original data would be recovered from the remaining 1 tuple of the dataset. In experiments, the proposed technique showed 100% data recovery when more than half of the tuples were modified as shown in Fig. 8. The results of experiments proved that the watermark was detected with accuracy rate of 100% as shown in Fig. 9. The watermark encoding is imperceptible; therefore, Mallory is unable to completely destroy the watermark.

![Fig. 8. Data Recovery after Alteration Attacks](image)

![Fig. 9. Watermark Decoding Accuracy of the proposed technique with Alteration Attack](image)

6.2.4 Additive Attacks

In additive attacks, Mallory attempts to claim fake ownership of data and embeds a forged watermark into the data. Mallory’s intentions include: (1) destroying the encoded watermark; and (2) proving her ownership over the data. However, imperceptible and distortion-free watermark embedding in the permuted digrams makes the proposed technique highly robust against additive attacks. As any data owner could easily prove his ownership by decoding his own watermark from the data. On the other hand, a certificate can be created as a watermark and registered with a trusted third party, known as certification authority (CA) [27].

In this scenario, Mallory is unable to add her own watermark in the datasets because it is almost impossible to create the copy of the original certificate that is registered with CA.

6.2.5 Counterfeiting Attacks

In this particular attack, Mallory attempts to achieve a forged copy of the data so that she can use it in some unauthorized manner. Consider a scenario where Mallory gets access to the watermarked data \( D \). However, she is unable to find out the watermark \( W \) and tries to construct a copy of watermarked data \( D' \) with counterfeiting watermark \( W \). Moreover, an imperceptible watermark is encoded in the data taking into account a novel reversible watermarking technique; consequently, the forgery would not be successful and would get detected later with the extraction of the encoded watermark. Thus, such type of attack has no effect on the datasets.

7 CONCLUSIONS

Ever increasing use of social networks is generating “Big Data”. To extract useful knowledge from this data, different shareholders work in collaboration that involves sharing of digital data. Since the digital data can be easily copied, moved and modified, it faces the challenge of ownership rights once the same has been shared. This paper presents a mechanism for providing ownership rights over the digital data through digital watermarking. The proposed watermarking technique is not only robust but it also ensures original data recovery after watermark decoding. A formal method
has been used to prove the effectiveness of the system. Experimental study is performed for evaluation of the proposed watermarking technique against the defined threat model.

REFERENCES


Saman Itikhar received her M.S and Ph.D. degrees in Information Technology in 2008 and 2014, respectively, from National University of Sciences and Technology (NUST), Islamabad, Pakistan. She has worked as Research Assistant in an Information and Communication Technology Research and Development (ICT R&D) project from 2008 to 2011 in NUST. Currently, she is working as Assistant Professor in the Department of Computer Science, Government College University Faisalabad. Her research interests include information security, distributed computing, semantic web and cloud computing.

Ehsan Ullah Munir received his Ph.D. degree in Computer Software & Theory from Harbin Institute of technology Harbin, China in 2008. He completed his Masters in Computer Science from Barani Institute of Information Technology, Pakistan in 2001. He is currently an Associate Professor and Head in the department of Computer Science at COMSATS Institute of Information Technology, Wah Cantt, Pakistan. His research interests include machine learning, evolutionary computations techniques, data security, health informatics, big data analytics and decision support systems.

Samee U. Khan received a BS degree from Gulham Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan, and a PhD from the University of Texas, Arlington, TX, USA. Currently, he is Assistant Professor of Electrical and Computer Engineering at the North Dakota State University, Fargo, ND, USA. Prof. Khans research interests include optimization, robustness, and security of: cloud, grid, cluster and big data computing, social networks, wired and wireless networks, power systems, smart grids, and optical networks. His work has appeared in over 250 publications. He is a Fellow of the Institution of Engineering and Technology (IET, formerly IEE), and a Fellow of the British Computer Society (BCS).