Computers, Materials & Continua
DOI:10.32604/cmc.2022.019936
Article

Tech Science Press

A Novel Database Watermarking Technique Using Blockchain as Trusted Third Party

Ahmed S. Alghamdi[1], **Surayya Naz**[2], **Ammar Saeed**[3], **Eesa Al Solami**[1], **Muhammad Kamran**[1,*] **and Mohammed Saeed Alkatheiri**[1]

[1]Department of Cybersecurity, College of Computer Science and Engineering, University of Jeddah, KSA
[2]Abdul Wali Khan University, Mardan, Pakistan
[3]Department of Computer Science, COMSATS University Islamabad, Wah Campus, Wah Cantt, Pakistan
[*]Corresponding Author: Muhammad Kamran. Email: mkkamran@uj.edu.sa

Received: 02 May 2021; Accepted: 03 June 2021

Abstract: With widespread use of relational database in various real-life applications, maintaining integrity and providing copyright protection is gaining keen interest of the researchers. For this purpose, watermarking has been used for quite a long time. Watermarking requires the role of trusted third party and a mechanism to extract digital signatures (watermark) to prove the ownership of the data under dispute. This is often inefficient as lots of processing is required. Moreover, certain malicious attacks, like additive attacks, can give rise to a situation when more than one parties can claim the ownership of the same data by inserting and detecting their own set of watermarks from the same data. To solve this problem, we propose to use blockchain technology—as trusted third party—along with watermarking for providing a means of rights protection of relational databases. Using blockchain for writing the copyright information alongside watermarking helps to secure the watermark as changing the blockchain is very difficult. This way, we combined the resilience of our watermarking scheme and the strength of blockchain technology—for protecting the digital rights information from alteration—to design and implement a robust scheme for digital right protection of relational databases. Moreover, we also discuss how the proposed scheme can also be used for version control. The proposed technique works with nonnumeric features of relational database and does not target only selected tuple or portion (subset) from the database for watermark embedding unlike most of the existing techniques; as a result, the chances of subset selection containing no watermark decrease automatically. The proposed technique employs zero-watermarking approach and hence no intentional error (watermark) is added to the original dataset. The results of the experiments proved the effectiveness of the proposed scheme.

Keywords: Watermarking; blockchain; digital copyright protection; relational databases security

1 Introduction

Due to ever-improving availability and efficiency of the Internet, sharing of digital data such as text, databases, images, videos, and software has become very common across the globe. Data may also be outsourced for educational, commercial, or official purposes. Accordingly, it can be redistributed in an unauthorized manner. An attacker (Mallory) can make changes to that data and redistribute it using different channels pretending being the owner of that data. So, protecting the data from such kind of misuse and protecting ownership of actual owner becomes challenging. It may also affect the owner of digital products financially. Existing methods, like watermarking, offer a variety of ways for protecting the digital data from such breaches. Similarly, emerging technologies like blockchain are also finding their way for utilizing their strong features for providing a firewall for data breaches [1,2]. New classes of utilization have emerged including blockchain and cooperative examination, including information forming, fork semantics, alter proof or any blend thereof. They present new open doors for capacity frameworks to productively support such applications [3,4].

Database systems are being used in many real-life applications. Sometimes, authors need to outsource their databases across Internet or using other medium. The data in relational database is first stored and later evaluated and is prone to the problem of piracy and copyright violation and false ownership claims [5]. As watermarking of multimedia objects is comparatively mature field and a lot of related work is available in this area, many techniques are available to deal with copyright issue of multimedia products efficiently including cryptography, steganography, blockchain, and watermarking. By using blockchain and watermarking, the actual owner of the data can be identified easily and efficiently [6,7].

While there are many techniques proposed to deal with the digital rights (or ownership rights) protection of relational databases, the researchers of the database fields are also showing their interest recently to use the idea of watermarking and blockchain together for the copyright protection of the relational databases. The main scope of blockchain is to give value to Internet transactions. They can also be used to timestamp digitized documents so that it is not possible to tamper them [8].

The watermarking and blockchain based techniques can be used for robust copyright marking that provided us the motivation for our proposed work. More specifically, the motivation behind our proposed work is that current techniques handle some of the attacks but with little probability of survival against strong subset attacks. Moreover, they often require the role of a trusted third party (TTP) to address the ownership dispute of the underlying data. For instance, consider a scenario where a user $U1$ is the owner of Data D. He embeds his Watermark $W1$ in D to create D' and shares it with some other users. Now, if another user $U2$ acts as malicious attacker and inserts his watermark $WM2$ in D' to create D'' which is essentially a tampered copy of D'. In case of ownership dispute over D'', the user $U1$ extracts his watermark from it and the user $U2$ can also extract his watermark from D'' because he has also inserted his watermark in D' to create D''. Now the dispute arises: *who is the actual owner of data D''*? To solve this issue, we get the motivation to use blockchain to store digital watermark information because blockchain can help to detect tampering and store time stamps. Accordingly, blockchain can be used to know who watermarked the data first and thus the actual owner of the data (D'' in this case) can be identified. So, we use blockchains to act as TTP to store watermark (or ownership) information.

Another challenge for enforcing right protection on digital data is to preserve the usability of the data that may be compromised while embedding the watermark as copyright information in the original data that brings distortions in the data [9–11]. Furthermore, the extraction of watermark should neither require the original data or the embedded watermark to make it a blind system. In this context, this work presents a robust distortion-free blind database watermarking technique particularly targeting resilience against subset attacks while most of the algorithms presented in the literature for digital rights protection are distortion based and introduce errors to the original data while in some applications the maintenance of the quality is of high priority. Fig. 1 shows a top-level architecture of a basic watermarking system.

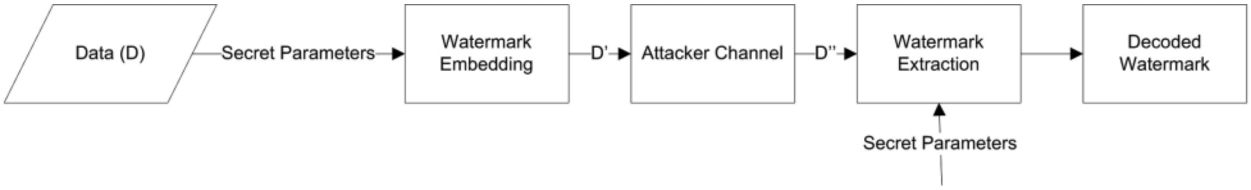

Figure 1: Basic watermarking system

Blockchains (see Fig. 2) are progressively perceived as an option in contrast to conventional information base frameworks for value-based information and dataset stockpiling. However, to be an achievable substitution to the, as of now, prevailing social information bases for the undertaking use, blockchains should likewise guarantee that honesty constraints hold for their put away information and for the interrelationships between such information passages [8]. These trustworthiness limitations guarantee exactness and consistency of information over its life cycle and are significant parts of the plan, execution, and utilization of information base arrangements. For this, we use the strong security features of blockchain alongside watermarking for providing digital protection of relational data.

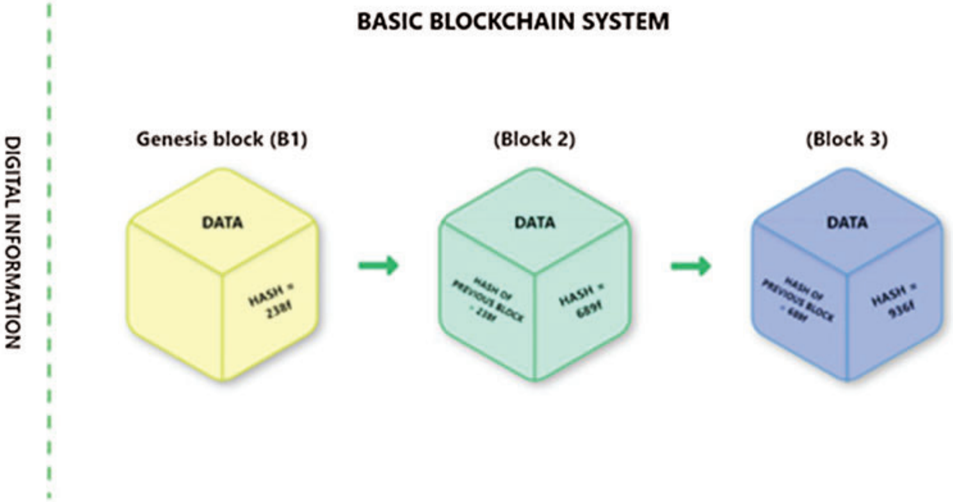

Figure 2: Basic blockchain system

The idea of using blockchain along with watermarking can also help in version control of the databases as the newly added block can facilitate making different versions of the data.

Database version control is versioning of objects stored in the database using a snapshot. At first, a snapshot of these objects which contains information about parameters of object is stored in the database. To store next version of the database, first version of the database object is replaced by its newest version of and previous snapshot of database is replaced by the newest one [12]. For this purpose, we propose to use blockchain to store time stamp information of latest copy (version) of the database. This is another contribution of the proposed work towards copyright protection of the database which can also be used for version control. Without the loss of generality, in our work, we use alpha numeric features as almost all the databases have at least one alpha numeric column in a table. However, since our technique does not bring any distortion to the original dataset; therefore, it can also be easily modified to use it with columns having other data types (for instance, numeric) as well. Moreover, certain malicious attacks like additive attacks can give rise to a situation when more than one parties can claim the ownership of the same data by inserting and detecting their own set of watermarks from the same data. To solve this problem, we take the advantage of the strength of use blockchain technology—as TTP—along with watermarking for providing a means of rights protection of relational databases. Adopting blockchain for writing the copyright information alongside watermarking helps to secure the watermark as changing the blockchain is very difficult. This way, we combined the resilience of our watermarking scheme and the strength of blockchain technology—for protecting the digital rights information from alteration—to design and implement a robust scheme for digital right protection of relational databases.

In a nutshell, the major contributions of this paper are: (i) we propose a distortion-free relational database watermarking technique based for addressing dispute of digital rights while preserving the data quality; (ii) we store the watermark as digital rights information in blockchain to utilize its strength of protecting the watermark from tampering; (iii) we provide a mechanism for using the proposed technique for version control of the database; and (iv) we use various attacks to test the resilience of proposed scheme against malicious attacks.

The rest of the paper is organized as follows. Section 2 discusses related work in the concerned domain. The proposed technique has been presented in Section 3 with Section 4 providing experimental details for showing the effectiveness of the proposed technique. Section 5 concludes the paper with some highlights to the future direction.

2 Related Work

Although the research on watermarking of relational databases started almost two decades ago in 2002, the application of same idea alongside blockchain technology is new. The idea of relational database watermarking was given for the first time in 2002 by Agrawal et al. [13] who suggested that the database copyright protection can be achieved by inserting hidden 'marks' into the relational database. This research got the attention of the researchers all over the world. After the work of [13] idea for watermarking relational database for the first time and its extension in [14], many researchers proposed their various algorithms for preserving relational database integrity. For instance, Hu et al. [15] suggested the use of an image as watermarks by proposing a technique in which an image is embedded into the relational data as copyright information. The image is considered as a sequence of 0's and 1's. The given method of using image proved to be correct, feasible, and robust and it supported easy watermarking identification for ownership claim. Pournaghshband [16] proposed a technique which embeds watermark information in the form of fake tuples inserted into the database. Although this technique is robust against different forms of malicious attacks; however, it requires the use of TTP for solving the ownership dispute. Moreover, the authors did not consider the additive attack where the attacker can watermark the already watermark dataset to claim the ownership of the data.

In a recent technique in [17], Farah et al. proposed a database watermarking technique that works for numeric as well as non-numeric databases. They provide their scheme as a service to provide robust ownership protection. However, they do not consider using the blockchain to reduce the role of TTP. Sahoo et al. [18] introduced a blockchain-technology based technique that uses the idea of controlling the number of users who can access the underlying data. However, this scheme does not itself focus on making the watermarking scheme robust enough to be robust against malicious attacks. Such techniques are not applicable in blockchain based database environment where the number of users is high and the watermarking scheme is required to provide robustness against malicious attacks.

Some recent works like [1,4,19,20] using blockchain and watermarking for ownership rights management cannot be used on relational databases because the watermarking of relational databases itself is different from other data formats as the part of a database (subset of the original database) is often useful as opposed to other data formats like images etc. While on the other hand, our proposed work works for relational databases. In [21], authors proposed a method of database watermarking using image with a focus on preventing additive attacks that create an ownership dispute when an already watermarked database is watermarked again. But again, this technique does not provide any mechanism to reduce the role of TTP for resolving ownership dispute. The works presented in [22–24] also present various recent techniques for ownership protection and user identification. But none of them considered any requirement to reduce the role of TTP for resolving ownership dispute. Furthermore, they also do not focus on the requirement of version control of the database. This further strengthens our motivation for our work to fill these missing gaps.

A technique in [25] uses blockchain for detecting tampering in the databases and does not provide any mechanism for solving ownership dispute for digital rights management. Another closely relevant technique in [26] uses blockchain technology for watermarking big data but it is focused on transfer of ownership and does not focus on version control of the underlying data. All the above techniques provide ownership protection mechanism for databases without using blockchain-based technology working both for: (i) digital right protection of the database; and (ii) version control of the database. On the other hand, the focus of the proposed technique is to use blockchain for enhancing the robustness of the database watermarking technique and also using the timestamp information stored in the blockchain to make versions of data after some new data is added into the databases. We believe that this is another novelty of the proposed scheme.

3 Proposed Methodology

In our proposed methodology, we present an algorithm that divides the whole database into groups (subsets) based on median occurred letter in the whole table. For the ease of the readers, we list the major symbols used in this paper in Tab. 1.

We generate a letter's occurrence table (LOT) that shows the occurrence of each letter in the data. The median occurred letter is calculated from the table. This median occurred letter is used as group separator to divide the whole data into groups. Then the most occurred letters i.e., 1st, 2nd, 3rd, 4th, and 5th MOLT (most occurred letters table) is created, and a secret key is generated using this MOLT. This key is also part of digital rights information and is stored in blockchain—thus reducing the use of CA (certifying authority) which usually acts the role of TTP—along with date and time stamp. If a suspicious database needs to be verified for data ownership rights, this digital rights information (secret key and watermark) can be used for addressing the dispute of data ownership. The overall architecture of the proposed technique has been presented in Fig. 3.

Table 1: Definitions of symbols used in the paper

Symbol	Description
LOT	Letter's occurrence table
MOL	Most occurred letters table
CA	Certifying authority
TTP	Trusted third party
GS	Groups separator
T	Database table
WM	Watermark
SS	Shift size

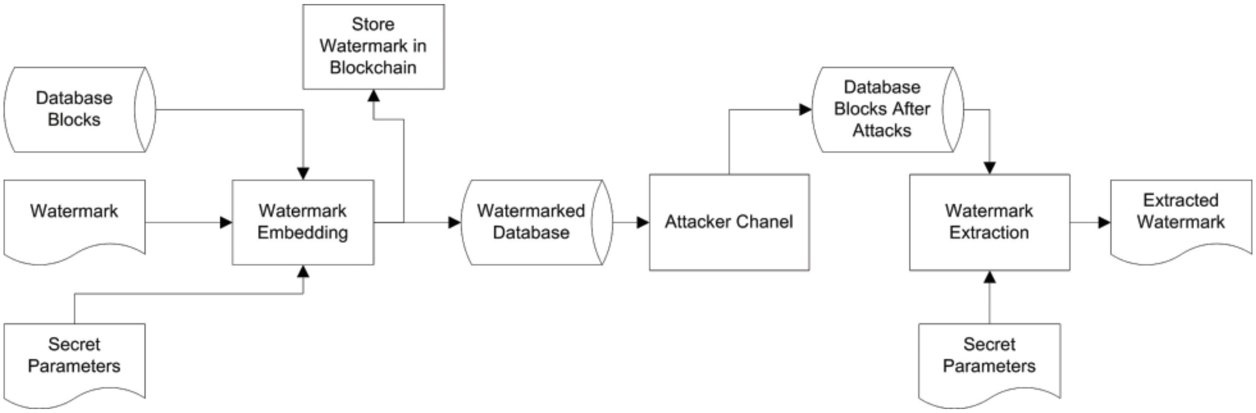

Figure 3: Architecture of proposed scheme

Before going into the details of the proposed scheme, we define some basic parameters that we use at various stages of the proposed scheme.

Watermark: It is security information which is being inserted in the original data to maintain the copyright protection. In the proposed technique, watermark is a string of letters provided as input to the algorithm which helps in generating secret key.

Watermarking: It is the process of embedding the watermarks into the actual data. But here we are not actually embedding the watermarks into the data and we just use the watermark to generate the key. We use all the tuples of the database table to ensure the robustness of embedded watermark because if an attacker attacks a subset of the watermarked data, the rest of the data can be sued to extract the embedded watermark.

TTP or CA: In traditional watermarking systems, it is a neutral third party with which an author (or data owner) can register his work with date and time stamp. The proposed technique minimizes the use of TTP by utilizing the strong security feature of blockchain technology by storing the data ownership related information in it. In this way, we ensure the robustness of the proposed scheme.

LOT: It is the list that contains the number of times each letter occurs in a given string. By removing digits and special characters from the given string, it is converted to a string of English letters only. The frequency of each letter in this string is used to generate LOT.

Groups Separator (GS): It is the median value of occurrence of letters which is calculated from the values in the LOT. GS is used as group separator to divide the whole data into groups. Suppose the median value of all the values in the list of letter occurrence is 155. We will see in the list which letter's occurrence value is equal to 155. Suppose in the list 'n' is the letter which occurred 155 in the whole data. So 'n' is our GS, and the data will be partitioned based on value of 'n'.

MOLT: It keeps the most occurred values. After dividing the whole string data based on GS, the 1^{st} most occurred letter, 2^{nd} most occurred letter and 3^{rd} most occurred letter is found. A table called MOLT (Most Occurred Letter Table) is created where the three most occurred letters for each group are stored.

Secret Key: The Secret key is generated by logically inserting the watermark in the data during the watermark embedding process.

Shifting: Shifting of a letter occurs when a specific letter in the watermark is not found in the MOLT. And after that, the shifted letter is searched for.

The proposed technique consists of a two-phase process. First, in the Embedding phase, watermark is inserted in the table to generate the secret key while in the watermark extraction phase watermark is extracted watermarks from the watermarked data.

3.1 Watermark Embedding Phase

This phase involves various steps and uses the following parameters. A table T that contains at least one non-numeric columns, an n-character long watermark string $WM = WM_1 WM_2 \ldots WM_n$, and shift size SS of length 2 to 6. The various steps involved in this phase have been discussed in the following.

3.1.1 Preprocessing

A database sample is converted to a string of letters by removing all special symbols; digits i.e., make it a string of English letters. The occurrence of each letter in the string (i.e., for a-z and A-Z) is calculated. The median occurred letter in the list is the formulated which will act as a group separator for the division of the string of table values. Similarly, the Mode, and Range among the list of letter occurrence is also found. The whole string is divided based on group separator letter, into separate groups. For each group, the most occurred values are found i.e., 1^{st} most occurred letter, 2^{nd} most occurred letter and 3^{rd} most occurred letter in that group to obtain MOLT. Algorithm 1 lists the steps of this phase.

3.1.2 Secret Key Generation

The secret key is generated as follows:

Step 1: First of all, searching is performed for the letters of the watermark (WM) one by one in MOLT in the relevant columns. Next step is to search WM_1 (the first letter of watermark WM) in first column of MOLT. Wherever found, the algorithm (see Algorithm 2) returns the location to the key and skip the rest of the options; otherwise, search WM1 in the second column, if found then note this location and if it is not found go to next option. Search the WM in the 3rd column and if it is found then return the location to the key else go to next option. In this option, shift the WM according to the SS, add the letter "i" to the key and repeat the above procedure for the shifted letter.

Algorithm 1 Preprocessing

```
Require: Database Table $T$
Ensure: $MOLT$
    Generate LOT
    $Mode =$ Compute Mode (LOT)
    $Range =$ Compute Range (LOT)
    $MOL = Mode$
    Set $GS = MOL$
    Divide the string in groups based on $GS$
    for each group do
    Compute 1st, 2nd, 3rd $MOL$
    Construct $MOLT$ using $MOL$ end
  for return $MOLT$
```

Step 2: If the letter of WM is not found in the whole MOLT, it is shifted according to the SS. After this shifting, the process given at Step 1 is repeated for this letter too.

Step 3: Do the same for each letter WM_1 to WM_n of the watermark WM and return the location, separating each entry with "-", to the key and hence generate the secret key.

Step 4: Store the generated secret key, the Shift Size (SS), and watermark (WM) in the blockchain along with the date and time stamp. This secret key is used to verify the ownership of the data and it does not bring any change to the original data; hence the proposed watermark embedding is distortion-free. Algorithm 2 lists the steps of watermark embedding phase.

3.2 Watermark Extraction Phase

This section explains the steps involved in watermark extraction phase. For watermark extraction, the following parameters are given as input to the algorithm. The table for which we want to extract the watermark, Shift Size (SS) which will be the same as given at the time of embedding of watermark process, and Secret key generated at the time of insertion of watermark.

3.2.1 Preprocessing

The preprocessing of data at the time of extraction of watermark is same as the embedding of watermarks. The table is converted to a string of letters by removing all special symbols; digits i.e., make it a string of pure letters. Then we find the occurrence of each letter in the string (i.e., for a-z and A-Z). The median occurred letter in the list is calculated. Similarly, the Mode, and Range among the list of letter occurrence is also found. The whole string obtained from the basis of group separator letter is divided into separate groups. For each group, the most occurred values are found out and hence MOLT for that group is obtained.

3.2.2 Extraction of Watermark from Data

The extraction of the watermark is nearly the reverse process of the generation of the key. For this, the proposed scheme first reads the first character from the key and goes to that specific location. Next, it stores the letter at that position. If it includes the letter *i* at the start of the key then shift-back the letter to its original one and store that letter in the extracted watermark. It continues the above two steps for finding all the letters of the key in the table and return the letter to form the extracted watermark *EWM*. This extracted watermark is then compared with the watermark stored in the blockchain for identification of the data owner and hence solving the data ownership dispute. The steps of this algorithm have been listed in Algorithm 3.

Algorithm 2 Watermark Embedding

```
Require: Database Table $T$, $WM$, $SS$
Ensure: Secret Key
$MOLT$ = Preprocessing($T$)
Search letters of $WM$ in $MOLT$
Set found = false
for each letter $l$ of $WM$ do
        Search $l$ in 1st column of $MOLT$
        if $l$ is found then $location_l$ = location of $l$
        else
            Search $l$ in 2nd column of $MOLT$
        end if
if $l$ is found then $location_l$ = location of $l$
else
           Search $l$ in 3rd column of $MOLT$
end if
if $l$ is found then $location_l$ = location of $l$
else
          Shift $WM$ according to $SS$ and append $i$ in Secret Key
          Repeat the above steps for shifted letter
end if
Append $location_l$ in Secret Key
end for
return Secret Key
```

4 Experiments and Results

In this section, we illustrate the details of various experiments performed on different samples of data (blocks) and their results to demonstrate the effectiveness of our proposed scheme. First, the preprocessing of the data was performed using mainly PHP WampServer. For supporting PHP, JavaScript and CSS were also used running on a Windows 7 operating system with a core i3 processor, 2.0 GB of memory, and 320-GB disk drive. Some experiments were also run in python using Matplotlib package.

4.1 Dataset

The database we used in the experiments DBLP-Citation-network V5 was taken from arnetminer.org [27]. The computer science bibliography DBLP consists of more than 1 million records of publications of about 1 million authors. As our work will consider mainly non-numeric data, we have chosen the attributes "Authors" and "Title" and make three different block sizes as mentioned in Tab. 2.

4.2 Parameter Settings

Before discussing the results of the experiments, we list the parameters that we used for the experiments in Tab. 3. Please note that shift size SS can vary from 1 to 26 (the total number of letters) but for our experiments, we used SS = 3.

Algorithm 3: Watermark Extraction

```
Require: Database Table $T$, Secret Key, $SS$
Ensure: Extracted watermark $EWM$
$MOLT =$ Preprocessing($T$)
for each character $s$ of Secret Key do
        Go to $location_s$
        l = Letter at $location_s$
        if $l == i$ then
        Shift back $l$ to its original position according to $SS$ and append $l$ in $EWM$
        end if
        Append $l$ in $EWM$
  end for
return EMW
```

Table 2: Different block sizes used in the experiments

S. No.	Block size	Number of records
1	Small block	100
2	Medium block	500
3	Large block	1000

4.3 Robustness Analysis

For examining the advantages of blockchain-based database technology regarding watermark security by protecting it from tampering, we evaluated the robustness of proposed watermarking scheme against subset selection attacks because the attacker may try to attack particular block (or group) of the dataset by taking a subset of the data. In subset selection attack, the attacker selects a subset from the watermarked database hoping that the subset does not contain watermark information. But our technique virtually—not physically, as the technique is distortion-free-embeds watermark in the whole table uniformly; therefore, there is no chance that the attacker selects any subset from the table that would not have the watermark. Such type of attacks can include subset selection attack, subset deletion attack, subset addition attack. In the proposed technique, we are dealing with all these three attacks collectively and calling it "subset modification attack".

Table 3: Parameters used in the experiments

Watermarks	Watermark string	Watermark length	Shift size
WM1	IIUI International Islamic University Islamabad Pakistan	100	3
WM2	Signal error that is securely and imperceptibly embedded into original content to describe information about owner.	500	3

For testing, we selected three samples of data i.e., small, medium, and large. Each sample is evaluated against 10%, 20%, 30% and 50% tuples modification attacks. Following six scenarios were used for testing the effectiveness of the proposed algorithm.

4.3.1 Scenario 1: Modification Attack on Small Dataset (Block)

In this set of experiments, the watermark WM1 is selected to insert it in the small size dataset block while attacking 10%, 20%, 30%, 50% of tuples as shown in Fig. 4. It is evident from this figure that when up to 20% tuples were attacked, the embedded watermark was detected with 100% accuracy. However, the decoding accuracy decreases when higher percentage of records is attacked as the embedded watermark gets disturbed (or corrupted). To investigate this effect further, we design our next experiment using watermark WM2.

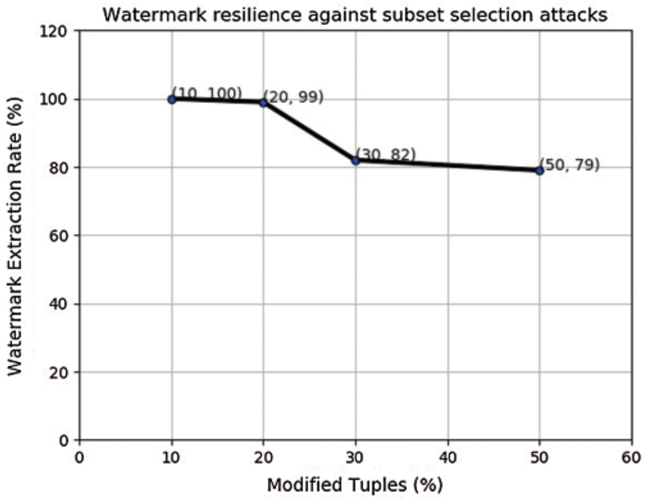

Figure 4: Modification attack on small size block watermarked with WM1

4.3.2 Scenario 2: Modification Attack on Small Dataset (Block)

To investigate the effect of watermark length on small sized blocks, the watermark WM2 is inserted in the small size block with the same attacks as in previous set of experiments. The watermark decoding accuracy with same sized data with larger watermark was observed to be better as shown in Fig. 5. We believe that this is due to the fact that the attacker could not attack the larger watermark when targeting larger number of tuples because he has to make sure that his attacks does not affect to the watermarked data to an extent where the data usability is affected. After this experiment, we performed the experiments with larger data size to observe the resilience of proposed watermarking scheme against malicious attacks.

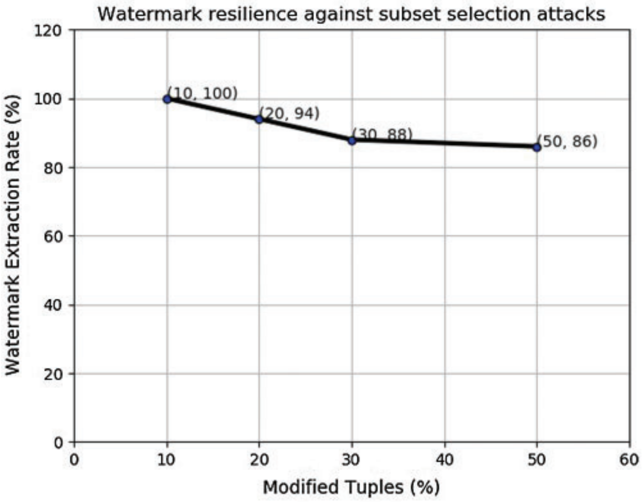

Figure 5: Modification attack on small size block watermarked with WM2

4.3.3 Scenario 3: Modification Attack on Medium Dataset (Block)

As stated earlier, we design the experiment to investigate the effect of data (block) size on the watermark decoding accuracy. In this experiment, the watermark WM1 is inserted in the medium size block. In this experimental setup again, 10%, 20%, 40%, and 50% tuples were attacked. Here again, we observed (see Fig. 6) that if higher number of tuples are attacked, the watermark decoding accuracy deteriorates. This gave us motivation to design our next experiment with large sized watermark WM2.

4.3.4 Scenario 4: Modification Attack on Medium Dataset (Block)

To investigate the resilience of larger watermark WM2, in this experiment, we chose to insert it in the medium size block. But here again, as evident from Fig. 7, the decoding accuracy keeps on deteriorating when more number of tuples are attacked. So, in next set of experiments, we increase the block size more.

4.3.5 Scenario 5: Modification Attack on Large Dataset (Block)

In this set of experiment, large sized block is selected for watermark insertion using the small watermark WM1. A similar pattern was again observed here as reported in Fig. 8. So, we moved further to our final set of experiment with large sized block and larger watermark WM2.

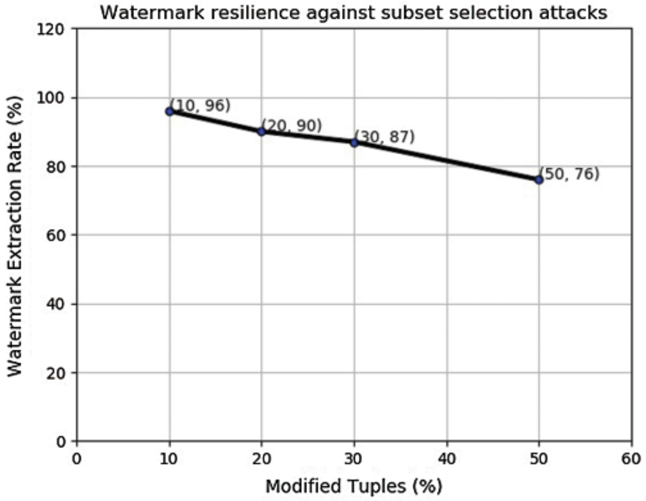

Figure 6: Modification attack on medium size block watermarked with WM1

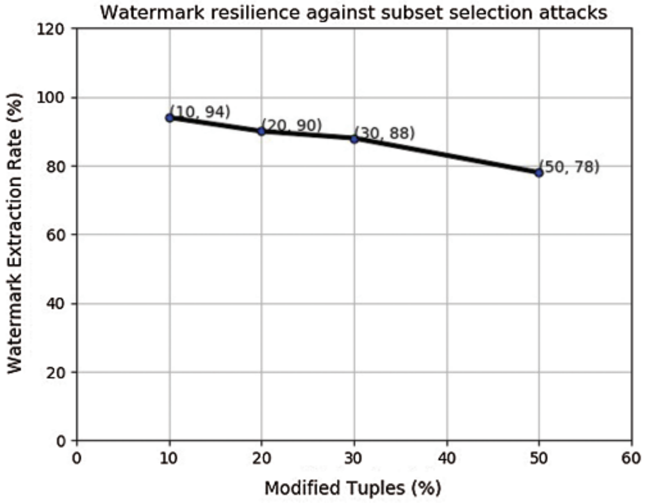

Figure 7: Modification attack on medium size block watermarked with WM2

4.3.6 Scenario 6: Modification Attack on Large Dataset (Block)

The final set of experiment uses the larger watermark WM2 to be inserted in the large size blocks when attacking 10%, 20%, 30%, and 50% of the tuples. Like previous experiments with larger watermark (WM2), this experiment again showed a similar behavior as shown in Fig. 9. Next, we analyze and discuss the results of our experiments.

4.4 Discussion on Results

In this section, we analyze and compare the experimental results and move towards generalization. The results are compared in Tab. 4 based on small block, medium block, and large block with respect to modification attack and watermark decoding accuracy for each sample size with different watermark lengths. It is evident from this table that as more number of tuples are attacked, watermark decoding accuracy decreases. However, if larger sized block is selected for watermark embedding with a large sized watermark, better decoding accuracy is achieved. Therefore, we recommend using larger sized watermark with larger sized blocks to get better resilience of embedded watermark by keeping in view the data quality to make sure that watermark embedding does not affect the data usability. Thus, the experimental results proved the robustness of the proposed scheme. Also, in case of additive attacks where the attacker—Mallory—inserts his own watermark in the watermarked database of the original owner, the watermark information, along with time stamp, stored in the blockchain can solve the ownership dispute as the data stored in blockchains is tamper free. As a result, the actual owner of the data can be identified using the ownership information (watermark) and associated time stamp stored in the blockchain.

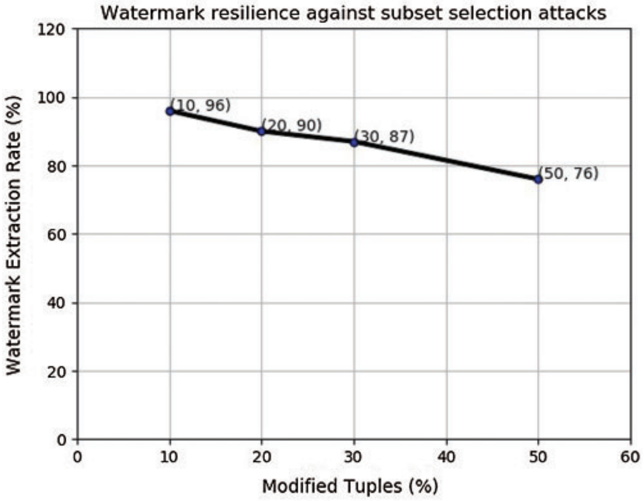

Figure 8: Modification attack on large size block watermarked with WM1

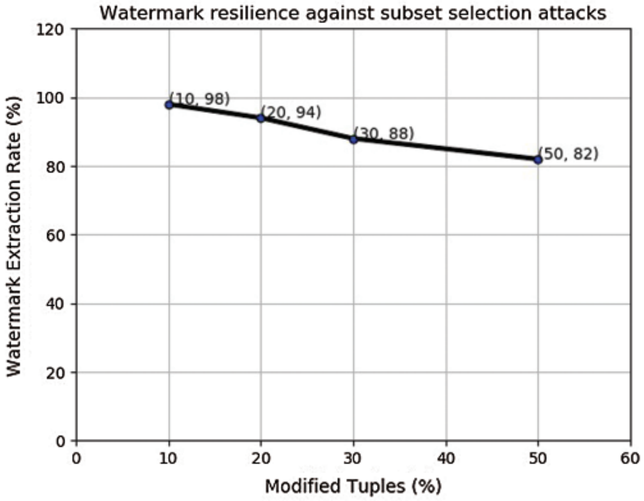

Figure 9: Modification attack on large size block watermarked with WM2

Table 4: Result comparison

Sample table (Block)	Modification %age	Watermark decoding accuracy	
		WM1	WM2
Small	10%	100%	100%
	20%	99%	98%
	30%	82%	92%
	50%	79%	86%
Medium	10%	96%	94%
	20%	90%	90%
	30%	87%	88%
	50%	76%	78%
Large	10%	98%	99%
	20%	94%	96%
	30%	88%	90%
	50%	82%	87%

As stated earlier. Recent works like [1,4,19,20] cannot be directly used for relational databases because the watermarking of relational databases itself is different from other data formats as the part of a database (subset of the original database) is often useful as opposed to other data formats like images etc. While on the other hand, our proposed work works for relational databases. However, a closely relevant technique [26] relies on the robustness of underlying watermarking technique but it does not itself present any mechanism to enhance the security of blockchain-based database watermarking technique against malicious attacks. Moreover, it neither focuses on version control of the underlying data nor on identification of owner of the data in case a database has been watermark by more than one user. On the other hand, our proposed technique itself uses the blockchain technology to enhance the security of database in such a way that helps to: (i) identify the owner of a particular of data in case of more than one user have watermarked the same data; and (ii) facilitate the version control process by using a different watermark when the data is updated by the insertion of a new block of data. Similarly, the technique presented in [18] recommends controlling the number of users for robustness of the scheme. However, our proposed scheme does not put any limitation for number of users and is yet robust. Similarly, the technique in [25] uses blockchain for detecting tampering in the databases and does not provide any mechanism for solving ownership dispute for digital rights management. Furthermore, to the best of our knowledge, the existing traditional database watermarking techniques consider the attacks that perform one type of data modification operation (insert, delete, update) at a time while the proposed technique considers them collectively as the attacker can also launch these attacks at the same time.

Since the proposed technique uses the blocks of data for watermark embedding, it can be used to make versions of data after some new blocks are added into the databases. In this case, by using the hash-based feature of blockchain technology, the version information of the database can be stored. So, if, after a certain number of new records have been inserted in the database, the database can be again watermarked using watermarking embedding module of the proposed technique along with storing the digital rights information in blockchain after this update in the database. Consequently, this information stored in blockchains can be used to get information about the versions of the database. We believe that this is another novel contribution of our proposed scheme.

5 Conclusions

Watermarking of relational database is playing a vital role in maintaining authentication, copy right protection, privacy, and integrity from last two decades. However, for taking full advantages, the watermarking techniques have yet to undergo through more challenges particularly involving the role of TTP. For this purpose, this paper presented a new blockchain technology based watermarking technique. In the proposed scheme, all the tuples in the database are used during the watermark insertion process, reducing the chances of subset selection attack. Moreover, the proposed technique is 100% distortion-free technique because it virtually inserts the watermarks in the data, and no changes are brought in the original data during watermark embedding. The technique has been further strengthened for its robustness by incorporating the concept of blockchain technology for storing the digital rights information and reducing the role of TTP. The proposed technique is restricted to non-numeric data only. In future, we are expecting to extend it by combining the same method for numeric features. In this research, the most efficient and the root cause of other attacks i.e., subset selection attack is considered but the proposed technique can be examined for resilience against other attacks as well. Similarly, the proposed technique can be extended to deal with the security issues of very large databases and using more than one secret keys. Furthermore, the proposed technique only considers pure English letters and can be extended for special character and symbols as well in the future.

Funding Statement: This project was supported by University of Jeddah under the Grant Number (UJ-02-014-ICGR).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] O. Evsutin and Y. Meshcheryakov, "The use of the blockchain technology and digital watermarking to provide data authenticityon a mining enterprise," *Sensors*, vol. 20, no. 12, pp. 1–20, 2020.

[2] Z. Zheng, S. Xie, H.-N. Dai, X. Chen and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[3] S. Wang, T. T. A. Dinh, Q. Lin, Z. Xie, M. Zhang *et al.*, "Forkbase: An efficient storage engine for blockchain and forkable applications," *Proceedings of the VLDB Endowment*, vol. 11, no. 10, pp. 1137–1150, 2018.

[4] Z. Ma, M. Jiang, H. Gao and Z. Wang, "Blockchain for digital rights management," *Future Generation Computer Systems*, vol. 89, no. 1, pp. 746–764, 2018.

[5] S. Iftikhar, M. Kamran and Z. Anwar, "A survey on reversible watermarking techniques for relational databases," *Security and Communication Networks*, vol. 8, no. 15, pp. 2580–2603, 2015.

[6] R. Halder, S. Pal and A. Cortesi, "Watermarking techniques for relational databases: Survey, classification and comparison," *Journal of Universal Computer Science*, vol. 16, no. 21, pp. 3164–3190, 2010.

[7] S. M. Thampi, "Information hiding techniques: A tutorial review," in *ISTE-STTP on Network Security & Cryptography*, Kerala, India, pp. 1–19, 2004.

[8] X. Li, P. Jiang, T. Chen, X. Luo and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, no. C, pp. 841–853, 2020.

[9] M. L. P. Gort, C. Feregrino-Uribe, A. Cortesi and F. Fernández-Peña, “Hqr-scheme: A high quality and resilient virtual primary key generation approach for watermarking relational data,” *Expert Systems with Applications*, vol. 138, no. 1, pp. 1–28, 2019.

[10] A. Hamadou, L. Camara, A. A. Issaka Hassane and H. Naroua, “Reversible fragile watermarking scheme for relational database based on prediction-error expansion,” *Mathematical Problems in Engineering*, vol. 2020, no. 1, pp. 1–9, 2020.

[11] M. Kamran and M. Farooq, “A comprehensive survey of watermarking relational databases research,” *ArXiv*, vol. 1801.08271, pp. 1–20, 2018.

[12] M. Fischer, M. Pinzger and H. Gall, “Populating a release history database from version control and bug tracking systems,” in *Proc. of the 2003 Int. Conf. on Software Maintenance*, Amsterdam, Netherlands, pp. 23–32, 2003.

[13] R. Agrawal and J. Kiernan, “Watermarking relational databases,” in *VLDB’02: Proc. of the 28th Int. Conf. on Very Large Databases*, Hong Kong SAR, China, pp. 155–166, 2002.

[14] 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.

[15] Z. Hu, Z. Cao and J. Sun, “An image based algorithm for watermarking relational databases,” in *2009 Int. Conf. on Measuring Technology and Mechatronics Automation*, Zhangjiajie, China, pp. 425–428, 2009.

[16] V. Pournaghshband, “A new watermarking approach for relational data,” in *Proc. of the 46th Annual Southeast Regional Conf. on XX*, Auburn, AL, USA, pp. 127–131, 2008.

[17] F. Naz, A. Khan, M. Ahmed, M. I. Khan, S. Din *et al.*, “Watermarking as a service (waas) with anonymity,” *Multimedia Tools and Applications*, vol. 79, no. 23, pp. 16051–16075, 2020.

[18] S. Sahoo and R. Halder, “Traceability and ownership claim of data on big data marketplace using blockchain technology,” *Journal of Information and Telecommunication*, vol. 5, no. 1, pp. 35–61, 2021.

[19] R. Latypov and E. Stolov, “A new watermarking method to protect blockchain records comprising small graphic files,” in *2019 42nd Int. Conf. on Telecommunications and Signal Processing*, Budapest, Hungary, pp. 668–671, 2019.

[20] D. Bouslimi and G. Coatrieux, “A crypto-watermarking system for ensuring reliability control and traceability of medical images,” *Signal Processing: Image Communication*, vol. 47, no. C, pp. 160–169, 2016.

[21] M. L. P. Gort, M. Olliaro, C. Feregrino-Uribe and A. Cortesi, “Preventing additive attacks to relational database watermarking,” in *Int. Conf. on Research and Practical Issues of Enterprise Information Systems*, Prague, Czech Republic, Springer, pp. 131–140, 2019

[22] K. Khatatneh, A. Odeh, A. Mashaleh and H. Hamadeen, “Secure digital databases using watermarking based on English-character attributes,” *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, vol. 13, no. 3, pp. 477–481, 2020.

[23] H. Tufail, K. Zafar and A. R. Baig, “Relational database security using digital watermarking and evolutionary techniques,” *Computational Intelligence*, vol. 35, no. 4, pp. 693–716, 2019.

[24] M. Kamran and E. U. Munir, “On the role of optimization algorithms in ownership-preserving data mining,” *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 2, pp. 151–164, 2018.

[25] K. Rani and C. Sharma, “Tampering detection of distributed databases using blockchain technology,” in *2019 Twelfth Int. Conf. on Contemporary Computing (IC3)*, Noida, India, pp. 1–4, 2019.

[26] S. Sahoo, R. Roshan, V. Singh and R. Halder, “Bdmark: A blockchain-driven approach to big data watermarking,” in *Asian Conf. on Intelligent Information and Database Systems*, Phuket, Thailand, Springer, pp. 71–84, 2020.

[27] 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.