

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Reducing multiple occurrences of meta-mark selection in relational data watermarking

MAIKEL LÁZARO PÉREZ GORT¹, MARTINA OLLIARO¹, AND AGOSTINO CORTESI¹

¹Università Ca' Foscari di Venezia. Campus Scientifico Via Torino, 155 30172 Mestre (VE), Italy

Corresponding author: Maikel Lázaro Pérez Gort (maikel.perezgort@unive.it)

This work was partially supported by the project "VIR2EM - Virtualization and Remotization for Resilient and Efficient Manufacturing" – POR FESR VENETO 2014-2020.

ABSTRACT Contrary to multimedia data watermarking approaches, it is not recommended that relational data watermarking techniques consider sequential selection for marks in the watermark and embedding locations in the digital asset being protected. Indeed, considering the database relations' elements, i.e., tuple and attributes, when watermarking techniques are based on sequential processes, watermark detection can be easily compromised by performing subset reverse order attacks. As a result, attackers can obtain owner evidence-free high-quality data since no data modifications for mark removing are required for the malicious operation to succeed. A standard solution to this problem has been pseudo-random selection, which however often leads to choosing the same marks multiple times, ignoring others, thus compromising the embedding of the entire watermark. This work proposes an engine that contributes to control marks' recurrent selection, allowing marks excluded by previous approaches to be considered. The experiments performed show a dramatic improvement of the embedded watermark quality when the proposed engine is included in watermarking techniques' architecture. They also provide evidence that this proposal leads to higher resilience against common malicious operations such as subset and superset attacks.

INDEX TERMS Mark exclusion, Pseudo-random selection, Redundant selection, Relational data, Watermarking.

I. INTRODUCTION

DIGITAL watermarking has become popular for providing evidence of ownership [1]–[3] or data tampering [4], [5], among others, and it allows the benefits of data spreading on the internet while protecting intellectual property. Firstly proposed as a field of information hiding techniques to secure multimedia data, watermarking approaches were also defined for relational data, considering the growth of web-based services with the needs of storing and sharing countless information. Two main processes characterize watermarking techniques [6]: the watermark embedding and its verification. Watermark verification is performed when there are suspicious of piracy, false ownership claims over protected data, among others.

The design and implementation of the watermark insertion and extraction actions highly depend on the nature of the protected data.

For instance, multimedia data are featured by a sequence of bits that offer high watermark embedding coverage. Contrary,

relational databases do not present data redundancy, thus decreasing the chances for watermark coverage on database relations. In terms of updates frequency, multimedia data are not meant to suffer highly-frequent updates, except for special applications. For this reason, any hidden information, e.g., the watermark, on multimedia objects is expected to endure. Instead, relational data are updated daily. Operations such as insertion, deletion, and updates of tuples cause a degradation of the watermark signal in time. Also, multimedia data are meant to be perceived by human systems, such as the visual (HVS) or the auditory (HAS). However, the limitations of those systems offer a high coverage to hide information, e.g., changes caused by embedding marks in frequencies outside the 20Hz - 20 kHz interval of an audio file are overlooked by HAS. Differently, data stored in relational databases are meant to be interpreted with the help of programming layers oriented to implement business rules and deploy the information generated in a graphical user interface (GUI) and printed in digital reports. Therefore, any distortion caused in

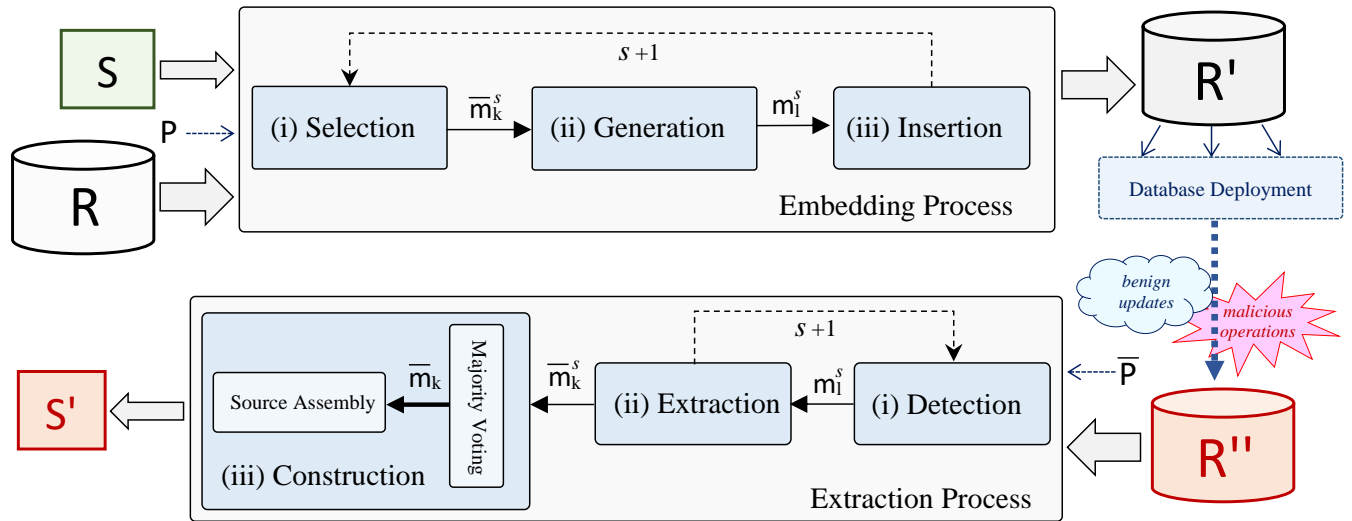


FIGURE 1. Processes implemented by watermarking techniques.

relational data will contribute to decision-making based on distorted information. Finally, the binary stream of multimedia data cannot be resorted without affecting the perception of the multimedia object, which allows the use of sequential watermark embedding over the spatial domain of multimedia objects. In other words, the selection of positions to embed the marks following the order of the bits in the binary stream that constitutes the digital asset.

In comparison, attributes and tuples in a relation do not present a fixed order, making sequential embedding not recommended. The order in which the database elements are presented is purely for the human comprehension of the database structure. For that reason, reordering tuples and/or attributes of relation do not compromise the result of the query performed over the data, not their usability. Indeed, relational data watermarking techniques implementing sequential synchronization compromise watermark detection if the protected data are reordered, e.g., hit by a subset reverse order attack [6], not requiring any updates in the process. This happens considering sequential watermark synchronization is based on embedding and extracting the watermark signal following the order of the marks in the watermark source and the embedding positions in the protected digital asset. In this way, the attacker can cause reasonable doubts in ownership claims while preserving data quality.

To avoid sequentiality in watermarking of relational data, some techniques propose the use of pseudo-random selection for picking the marks from the generated watermark and selecting their embedding place in the relation, e.g., Sardroudi & Ibrahim [7], Pérez Gort et al. [1], Hu et al. [8]. This approach successfully creates robustness against subset reverse order attacks, but once applied, another problem emerges. Indeed, some marks can be recurrently selected during the embedding, while others are entirely ignored. Embedding the same mark multiple times is good to overcome benign updates and update-

based attacks effects over the watermark signal if a majority voting is performed during the watermark extraction process. However, the recurrent selections of some marks comes at the expenses of the partial embedding of the watermark, compromising the insertion of a comprehensible watermark signal or resulting in the embedding of a signal weak enough, not requiring a high degree of attacks to compromise the detection of the watermark before compromising data quality.

This work aims to provide distribution of repetition of mark selection as uniform as possible, allowing marks previously excluded to be considered and improving the quality of the embedded watermark.

A. ARCHITECTURE OF RELATIONAL DATABASE WATERMARKING

In the following, we address the protection of a single relation in a relational database. Watermarking techniques for relational databases can be classified in several ways. For instance, by the type of watermark information to embed, the type of attributes into which the marks are embedded, or if the watermark distorts the data or not [9]. In particular, techniques modifying the relation content for its protection are defined as distortion-based, whereas the others as distortion-free. For distortion-based watermarking techniques, marks are first generated and then embedded into a relation with the condition that the embedding will not compromise data usability.

We introduce the definition of meta-mark, consisting of a bit extracted from the source used to generate the watermark, to generate then the mark being embedded. Notice that previous works do not make a distinction between meta-marks and marks. This distinction is of high relevance to this paper, considering the recurrent selection occurs with meta-marks, while marks are uniquely generated from them.

Fig. 1 highlights the steps generally characterizing the

embedding and extraction phases of a distortion-based watermarking technique, featured by pseudo-random selection. The embedding phase is an iterative process. Given a relation R in a relational database DB , a watermark source S , and a set of parameters P , each iteration s pseudo-randomly selects the so-called meta-mark \bar{m}_k from S . The meta-mark \bar{m}_k is combined with a bit of the attribute value being watermarked in the database to generate the mark m_l to be embedded.

Also, in the same iteration s , the location to embed the mark m_l in R is computed. Once m_l is generated, it is inserted into the chosen location of R .¹ After the mark of stage s is embedded, we enter into stage $s+1$ and the process proceeds with the generation of the next mark and the selection of the next embedding position. Thus the embedding process can be expressed by the function $\mathcal{I}(R, S, P) = R'$, where R' denotes the resulting watermarked relation. The embedding ends when all attributes and tuples from R has been considered according to the locomotion rules defined for the watermark synchronization. The locomotion rules establish the way to go through the tuples and attributes of R to find places suitable for mark embedding or extraction according to the parameters' values used by the embedding and extraction processes. In this work, AHK algorithm principles are applied (cf. Section II).

The extraction phase is given by the function $\mathcal{E}(R'', \bar{P}) = S'$, where R'' denotes the copy of the protected relation that can be distorted due to benign updates or malicious attacks performed once R' is deployed, \bar{P} is the set of parameters used for the extraction phase, and S' is the watermark source obtained from R'' . For the watermark extraction, the number of parameters and their values must be the same as those used during the watermark insertion, according to $P = \bar{P}$. The extraction phase is featured by an iterative process too. Indeed, a mark is detected and extracted at each iteration, and the corresponding meta-mark is generated. Several values for the same meta-mark can be extracted. After all the iterations are performed, a majority voting is carried out for the watermark source reconstruction, and each place's final meta-mark is produced. Once the meta-marks are collected, the watermark source is reconstructed by the "Source Assembly" step. The watermark source construction is the only sub-process of the watermark extraction that does not occur inside iterations.

B. PAPER CONTRIBUTION

This paper proposes the so-called Recurrence Reduction Engine (RRE) to limit the maximum number of times each meta-mark can be considered during watermark embedding (cf. Fig. 2) while preserving the pseudo-randomness property of the process. In particular, we add a 'Probability Box' to generate transition probabilities for meta-marks selection, according to the following rule: at stage $s+1$ neither the meta-mark selected by s nor the one chosen by $s-1$ can be used to generate the mark to embed. By adding this, we

¹The iteration stage and the meta-mark identifier k are not coincident since the number of meta-marks is constrained to the watermark source's length, and the number of generated embedding places gives stages (cf. Section II).

increase the chances of other meta-marks being selected. Furthermore, a 'Chaos Generator' is included to guarantee the same selection's order of the meta-marks during the embedding and the extraction processes, not compromising watermark synchronization. Our proposal functioning does not depend on the watermarking technique.

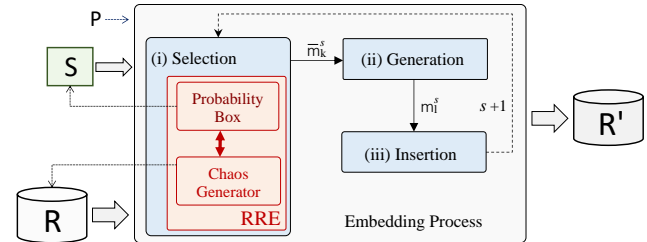


FIGURE 2. Architecture of the proposed embedding process.

The experiments performed to validate our work show how the quality of the embedded watermark improves due to considering more marks in the embedding. Despite recurrent selection reduction, most marks are still embedded multiple times, which, combined with majority voting during the extraction process, contributes to the resilience of the watermark against update-based malicious operations. By avoiding marks exclusion during the embedding and by using meaningful watermark sources, the watermark can be restored by applying enhancement algorithms.²

C. PAPER STRUCTURE

The rest of the paper is organized as follows. Section II, introduces the preliminaries on relational database watermarking and details the problem statement. Section III presents the watermarking architecture of our proposal along with the mark Recurrence Reduction Engine (RRE). Section IV presents a theoretical analysis of our approach. Section V depicts the experimental results. Section VI presents details of the related work. Section VII concludes.

II. PROBLEM STATEMENT

Let $R[\nu, \eta]$ be a relation, where ν denotes the number of attributes and η the number of tuples in R . According to the relational model [11], the corresponding relation schema is defined as $R(PK, A_0, \dots, A_{(\nu-1)})$, where PK is the relation's primary key, and A_j , with $j \in [0, \nu-1]$, represent the rest of the attributes of R . Tuples in $R[\nu, \eta]$ are denoted by T_i with $i \in [0, \eta-1]$, and $T_i[A_j]$ is the database element in the i -th tuple corresponding to the j -th attribute. Similarly, $T_i[PK]$ is the database element in the i -th tuple corresponding to the Primary Key. Fig. 3 presents an example of a database relation structure.

²Note that watermark signals can be meaningless or meaningful. Meaningful watermark signals are generated from the meaningful textual content, images, audio, among other sources [6], [10]. For meaningless watermark, meta-mark concept does not apply.

		Attributes					
		A ₀	A ₁	A ₂	...	A _(v-1)	
Primary Key		ID	NAME	AGE	GENDER	...	HEIGHT
Tuples	T ₀	1011	Alex	32	Male	...	175
	T ₁	1012	Cindy	24	Female	...	160
	⋮
	T _(η-1)	1015	Layla	30	Female	...	177

FIGURE 3. Table structure of a database relation [12].

Moreover, let S be the source used to generate the watermark, i.e., a binary array of meta-marks, and \bar{m}_k the k -th meta-mark in S , with $k \in [0, n-1]$, where $\bar{m} \in \{0, 1\}$ and n is the size of S . Also, let WM denote a watermark signal, i.e., a binary array collecting the marks generated during the embedding process, and m_l the l -th mark in WM , with $l \in [0, u-1]$, where $m \in \{0, 1\}$ and u is the size of WM [13]. The watermark WM being embedded into R can be generated from sources of different data types. When the watermark source is an image i.e., image-based watermarking technique (IBW) [14]–[16], we represent S as a matrix whose dimensions are given by the height H and width W of the image I used to generate the watermark. Watermarking techniques using binary images provoke less distortion on R while embedding WM , considering each pixel value is a bit, i.e., 0 for black and 1 for white.

To give an idea of the consequences arising from the embedding of marks generated from pseudo-random selected meta-marks, consider the source $S = \langle 1, 0, 1, 1, 1, 0, 0, 0 \rangle$ and the relation $R[4, 10]$. We assume the second meta-mark, i.e., $\bar{m}_1 = 0$, is used to generate the marks embedded in the positions $T_0[A_0]$, $T_2[A_0]$, $T_5[A_3]$, and $T_8[A_2]$. After a time, the database is updated and, during the watermark detection phase, when the marks for those positions are extracted, some of them might have a different value. Let $\text{extr}(T_i[A_j])$ be the function extracting the mark value embedded in $T_i[A_j]$. We assume that the values extracted for \bar{m}_1 are $\text{extr}(T_0[A_0]) = 0$, $\text{extr}(T_2[A_0]) = 0$, $\text{extr}(T_5[A_3]) = 0$, and $\text{extr}(T_8[A_2]) = 1$. Regarding the positions into which \bar{m}_1 was considered for the embedding, the mark value 0 has been extracted 3 times and the mark value 1 has been extracted just once. Thus, performing a majority voting, the value 0 is assigned to \bar{m}_1 . Let WM'' be the watermark signal extracted from R'' (cf. Section I-A), the function $\text{sim}(WM, WM'')$ computes the similarity level between the embedded and extracted signals. Suspicion of piracy will be confirmed if $\epsilon \leq \text{sim}(WM, WM'')$, where ϵ is a threshold of detection for piracy assertion. Details about the design and implementation of the $\text{sim}(\cdot)$ function depend on the nature of the watermark source.

Note that majority voting allows overcoming minor inconsistencies, like ignoring the mark with a value contradicting the values of the other marks extracted. Still, for it to

contribute to resilience against benign updates and update-based attacks, the more the same meta-mark is considered for the embedding, the better. But, while some meta-marks are considered multiple times, the chances for others to be chosen by the insertion process reduce, leading to the partial embedding of the watermark. This fact compromises the embedding of a comprehensible watermark signal or results in the embedding of a signal weak enough, not requiring a high degree from attacks to compromise the detection of WM before affecting data quality.

A. MOTIVATING EXAMPLE

The watermark partial embedding caused by pseudo-random selection of meta-marks affects watermarking techniques generating WM from images. In this case, missing pixels of images reconstructed from the embedded watermark are a direct consequence of ignoring some meta-marks.

To illustrate the problem, we use the IBW technique presented in [17] to watermark the numerical data set “Forest Cover Type”, available in [18]. The technique uses a binary image as a watermark source, which allows the generation of a single mark per pixel, considering the pixels’ values are one bit. Contrary, if colored images are selected as watermark sources, several marks are generated for each pixel to reconstruct the three-channel value, i.e., red, green, and blue, between 0 to 255, causing higher distortion over R compared to techniques using binary images.

We choose images of different sizes to generate the watermark (cf. Fig. 4) and later in this section we use red color to identify the positions where no pixels are selected. By using images with a different number of pixels, we can analyze the role of the source size n in pseudo-random selection when watermarking relations with the same number of attributes v and tuples η .

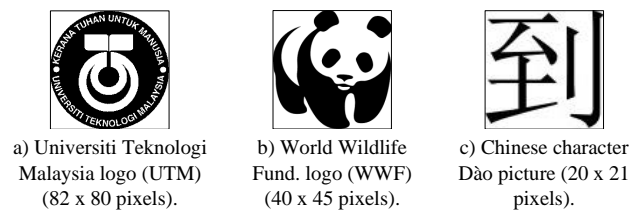


FIGURE 4. Binary images used as watermark sources.

Technique in [17] selects for marking approximately γ tuples out of η , where $\gamma \in [1, \eta]$ is defined as the Tuple Fraction (TF). The lowest values of γ will be responsible for involving a higher number of tuples in the process. Instead, the number of attributes marked per tuples, defined as the Attribute Fraction (AF), is denoted as $\delta \in [1, v]$. All attributes of the selected tuples are marked if $\delta = 1$.

The pseudo-random selection of tuples, attributes within chosen tuples, and bits within the available number ξ of least significant bits (*lsb*) of selected attributes is performed according to the value generated by using a one-way hash

function \mathcal{H} . As stated in [12], \mathcal{H} takes as input the primary key PK identifying the tuples of \mathbf{R} and the secret key SK given by the data owner according to (1), where \circ denotes the join operator.

$$F(T_i[\text{PK}]) = \mathcal{H}(\text{SK} \circ \mathcal{H}(\text{SK} \circ T_i[\text{PK}])) \quad (1)$$

The selection of embedding positions is carried out according to the combination of the AHK algorithm [19] and the analysis of each attribute per selected tuple considering the value of the attribute fraction δ (cf. Algorithm 1). The attributes analysed are those belonging to tuples satisfying the condition $F(T[\text{PK}]) \bmod \gamma = 0$ (line 2).

For each attribute A in the attribute list AL of \mathbf{R} , the Attribute Virtual Value (AVV), denoted as a_v , is generated using its β -th most significant bits (msb), to guarantee watermark synchronization, considering mark embedding, benign updates, and attacks do not modify their values to preserve data usability. Next, a_v is combined with $F(T_i[\text{PK}])$, and a new value a_h is created to decide if the attribute will be marked (line 5). Mark embedding proceeds only for attributes accomplishing the condition $a_h \bmod \delta = 0$ of line 6. Finally, the lsb position b to embed the mark generated from the selected \overline{m}_k is chosen and the embedding proceeds for the attribute A within the tuple T (line 8).

Algorithm 1: Selection of embedding positions [17].

```

1 foreach tuple  $T \in \mathbf{R}$  do
2   if  $F(T[\text{PK}]) \bmod \gamma = 0$  then
3     foreach attribute  $A \in \text{AL}$  do
4        $a_v \leftarrow \text{generate\_AVV}(A, \beta)$ 
5        $a_h \leftarrow \text{generate\_AVH}(a_v, F(T[\text{PK}]))$ 
6       if  $a_h \bmod \delta = 0$  then
7          $\text{bit\_index } b \leftarrow F(T[\text{PK}]) \bmod \xi$ 
8          $A \leftarrow \text{mark}(T, A, b, \overline{m}_k)$ 

```

Besides avoiding the downsides of sequential embedding stated in Section I, the selection of embedding places in \mathbf{R} is pseudo-randomly performed to difficult the attackers from guessing the locations of marks in \mathbf{R} as well as to scatter the distortion caused by the embedding. Enhancing the selection of embedding places with pseudo-randomness positively affects the watermarking architecture. Contrary, when applying pseudo-randomness to selecting the meta-marks from \mathbf{S} , new challenges arise.

Considering that multiple marks can be embedded in one tuple, the generation of the pixel's coordinates depends on the attributes selected within the tuple according to (2) and (3), where w and h are the selected width (out of W) and height (out of H) of the image I respectively.

To avoid obtaining the same coordinate every time the same value of a_h is generated, a different seed is applied for each one of the equations. Ideally, the seeds and a_h should be different every time, making the selection of new pixels

possible. However, due to using the same values of W , H , and msb for generating a_h , this is very unlikely.

The implementation of the seeds generation in [17] was done by considering γ , δ , and values of msb combined with a_h . Nevertheless, since both coordinates are generated for the same attribute, a constant element was added to differentiate seed1 from seed2.

$$w = \mathcal{H}(W, a_h, \text{seed1}) \quad (2)$$

$$h = \mathcal{H}(H, a_h, \text{seed2}) \quad (3)$$

To show the consequences of uncontrolled pseudo-random pixel selection we performed several experiments, watermarking \mathbf{R} each time using different values of γ whereas $\delta = 10$. The scatter of meta-marks' selection is obtained by using the binary capacity c_b introduced in [1] according to (4), where \overline{m} denotes the number of meta-marks selected during the embedding, out of n meta-marks composing \mathbf{S} .³

$$c_b = \overline{m} * 100/n \quad (4)$$

The behavior of recurrent selection is described by the weight-based capacity metric c_w also introduced in [1] according to (5), where the number of times the meta-mark \overline{m}_k is selected, is given by the function $\varkappa(\overline{m}_k)$. The weight-based capacity constitutes the accumulation of every time each meta-mark was selected.

$$c_w = \sum_{k=0}^{n-1} (\varkappa(\overline{m}_k))/n \quad (5)$$

To obtain the scattering of recurrent selection of meta-marks in \mathbf{S} , we use the standard deviation of the recurrent selection according to (6). Best results will be featured by values of σ_w as close as zero as possible, meaning that recurrent selection of meta-marks is uniform.

$$\sigma_w = \sqrt{\frac{\sum_{k=0}^{n-1} (\varkappa(\overline{m}_k) - c_w)^2}{n}} \quad (6)$$

Considering E as the set of values obtained by $\varkappa(\overline{m}_k)$ for all meta-marks, i.e., $E = \{\overline{m}_k \in \mathbf{S} \mid \varkappa(\overline{m}_k)\}$, the function $\max(E)$ returns the set's maximum element, which lets us know the extreme recurrence. Finally, the general situation of the selection is depicted in (7), where the number of meta-marks considered from \mathbf{S} (given by \overline{m}) is analyzed along with the scattering of the meta-marks' recurrent selection (given by σ_w).

$$\Theta(E) = \overline{m}/\sigma_w \quad (7)$$

According to (4), the more values are considered for the process the highest \overline{m} . Uniform embedding is described by

³The number of times the same meta-mark is selected is not considered to obtain c_b

high values of $\Theta(E)$, which are proportional to \bar{m} , and directly affected by σ_w . For cases describing more differences in the number of times each meta-mark is selected, σ_w presents a high value, resulting in a low value of $\Theta(E)$. On the other hand, the lowest σ_w the better.⁴

Tables 1, 2, and 3 show the results from the experiments carried out to illustrate the consequences of uncontrolled pseudo-random selection, when different watermark sources are used (cf. Fig. 4). The results of each column correspond to the embedding performed using different parameters. In general, γ varies whereas AF is kept the same ($\delta = 10$).

The rest of the metrics are computed for each experiment, along with the quality of the images reconstructed from the watermark embedded in R. As expected, for values of γ closer to 1, the number of red pixels in the image reconstructed is lower. Also, the values of c_b decrease, whereas the number of red pixels increases.

Regarding the comparison between the tables, reducing n while marking the same number of tuples and attributes in R has a direct effect on the metrics describing the spreading of meta-marks' selection. Table 3 shows better results for c_b and c_w . Nevertheless, σ_w and $\max(E)$ are also higher, which evidences the problem of meta-marks exclusion. The values given by these metrics let clear that the meta-mark exclusion problem can be solved without compromising embedding recurrence. Indeed, there are much more marks embedded than meta-marks considered for the generation of marks to embed, which means other meta-marks can be included in the process without drastically reducing recurrent selection.

TABLE 1. Differences among embedded UTM watermark using Algorithm 1.


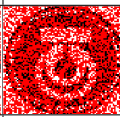
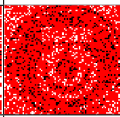
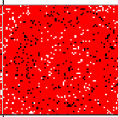
γ	5	10	20	30
Generated Image				
$\max(E)$	6	4	3	2
c_b	57.77	34.25	19.04	9.36
c_w	0.86	0.42	0.21	0.10
σ_w	0.92	0.64	0.46	0.31
$\Theta(E)$	41.09×10^2	35.03×10^2	27.37×10^2	19.60×10^2

TABLE 2. Differences among embedded WWF watermark using Algorithm 1.


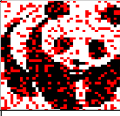
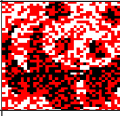
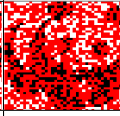
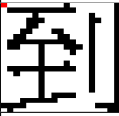
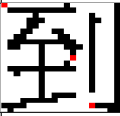
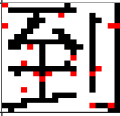

γ	5	10	20	30
Generated Image				
$\max(E)$	10	6	6	4
c_b	95.66	76.33	52.72	38.88
c_w	3.12	1.51	0.76	0.50
σ_w	1.81	1.29	0.90	0.71
$\Theta(E)$	95.29×10^1	10.65×10^2	10.52×10^2	98.78×10^1

TABLE 3. Differences among embedded Dào watermark using Algorithm 1.

γ	5	10	20	30
Generated Image				
$\max(E)$	24	13	9	7
c_b	99.76	99.29	95.71	87.38
c_w	13.34	6.47	3.28	2.14
σ_w	3.94	2.61	1.86	1.45
$\Theta(E)$	10.65×10^1	16.00×10^1	21.59×10^1	25.28×10^1

Partial consideration of meta-marks in S for the generation of WM directly affects the technique's robustness. In fact, when fewer meta-marks (out of n) are used, later updates on data will compromise a higher number of marks, and attackers will not need to update too many tuples to compromise watermark detection, being able to perform malicious operations while preserving data usability.

Fig. 5, 6 and 7 depict the quality of the watermark signal detected for the experiments shown in Tables 1, 2 and 3 respectively, when a tuple-deletion subset attack is performed. A different number of tuples are involved for each of the simulations, gradually increasing the attack's degree.

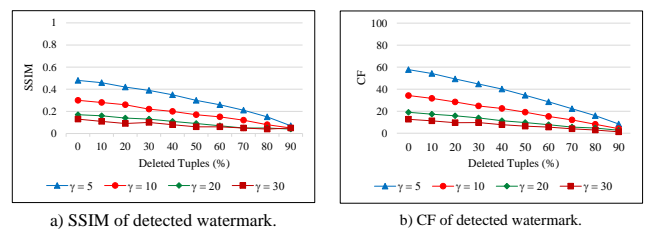


FIGURE 5. Quality of detected UTM WM after tuple deletion attacks.

In the figures, the quality of the watermark is measured by using the Correction Factor (CF) and the Structural Similarity Index (SSIM). The Correction Factor, according to (8), compares the values of each pixel for the same position of the embedded image vs. the extracted one. The embedded image is denoted as Img_{emb} whereas the extracted one is given by Img_{ext} . When CF is equal to 100, the two images are

⁴By definition, when $\sigma_w = 0$ then $\Theta(E) = \infty$.

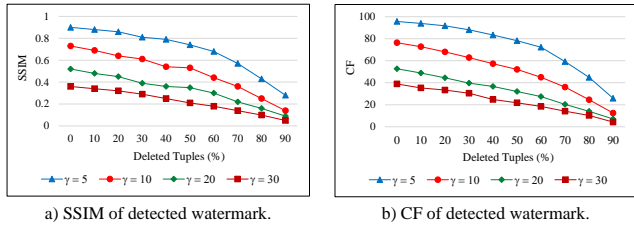


FIGURE 6. Quality of detected WWF WM after tuple deletion attacks.

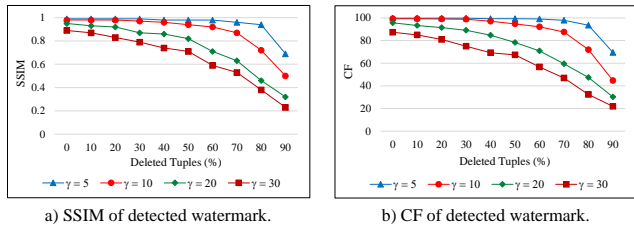


FIGURE 7. Quality of detected Dào WM after tuple deletion attacks.

identical. On the other hand, $CF = 0$ means that the images are entirely different.

$$CF = \frac{\sum_{i=1}^H \sum_{j=1}^W (\text{Img}_{\text{emb}}(i, j) \oplus \overline{\text{Img}_{\text{ext}}(i, j)})}{H \times W} \times 100 \quad (8)$$

The SSIM is obtained according to (9) and returns an appreciation of the extracted watermark quality closer to human perception. For this case, multiple windows of size $N \times N$ are defined by x and y . The domain of SSIM in this work is between 0 and 1, where 1 depicts the perfect structural similarity between the two images and 0 indicates no structural similarity. In the equation, the symbols μ_x and μ_y represent the average of x and y respectively, σ_x^2 and σ_y^2 their respective variance, and σ_{xy} their covariance. The elements C_1 and C_2 are two stabilization constants.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (9)$$

As is depicted in the figures, the quality of the detected WM is affected when the number of attacked tuples increases. Also, when the number of tuples marked is bigger (e.g., for $\gamma = 5$ or $\gamma = 10$), WM is compromised when more tuples are involved in the attack. Nevertheless, for the case of the watermark generated from the Dào character (cf. Fig. 7), since its size is smaller compared to the one generated from UTM and WWF sources, higher resilience is depicted.

Although the differences depicted in previous figures, allowing recurrent selection by managing its scattering is expected to have a direct benefit in the robustness of the technique, independently on the watermark source size.

The problem described in this section affects every technique implementing pseudo-random selection to prevent subset reverse order attack success. This is an inherent feature of the AHK algorithm, which is the base for a significant

number of watermarking approaches for relational data [7], [19], [20].

III. PROPOSED APPROACH

The problem addressed in this work has two main roots. First, there are no limitations defined for the recurrence of pseudo-random selection. Second, a source featured by low-scattered data limits the generation of the seed. The second root is harder to face by blind watermarking techniques, considering no-external content should be required for the watermark extraction [6]. Because of it, once a position is chosen to embed a mark, the elements involved in the mark generation must depend only on the value been watermarked (e.g., A, a_h, a_v) and the parameters used for watermark synchronization (e.g., $SK, \text{source size}, \zeta, \beta$).

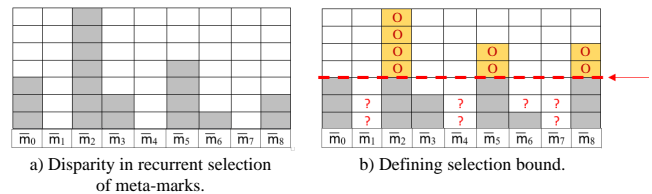


FIGURE 8. Normalizing redundant mark selection.

Let us start addressing the first root. The second root will be treated in Section III-B. The solution we propose addresses the first root with the help of the structures involved in the watermarking process. Our approach aims to restrict the selection of meta-marks until a specific limit $\rho \in \mathbb{Z}^+$, to increase the chances for other meta-marks to be selected. Fig. 8 presents an example with a watermark source composed of nine meta-marks. In Fig. 8.a) each time a meta-mark is selected, a gray rectangle is drawn on top of it. Some meta-marks, such as \bar{m}_2 , are selected multiple times, while others are ignored, e.g., \bar{m}_1 , \bar{m}_4 , and \bar{m}_7 . If a boundary is defined to only allow selecting the same meta-mark three times, as it is depicted in Fig. 8.b), there might be a chance for those meta-marks previously ignored to be considered.

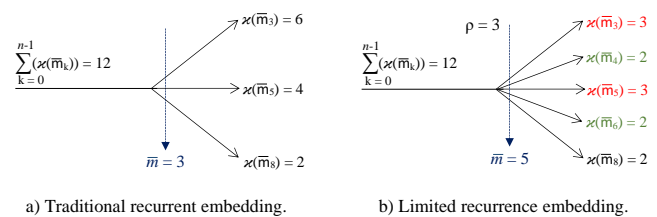


FIGURE 9. Increment of different number of marks considered.

Fig. 9 depicts a case of the benefits of recurrence selection limitation involving some of the metrics introduced in Section II. In this case, 12 marks are embedded. Fig. 9.a) shows the results of using traditional recurrence embedding, which considers only three meta-marks ($\bar{m} = 3$), some of them being embedded too many times considering others are ignored (e.g., \bar{m}_3 vs. \bar{m}_2). On the other hand, if the number of recurrence

embedding is limited to 3 according to $\rho = 3$ (cf. Fig.9.b)), then more meta-marks are selected, resulting in the embedding of a stronger watermark signal. This will be reflected in the value of σ_w , which will be lower for the case of the second figure, describing a more calibrated embedding.

In Fig. 9.b), values in red color represent recurrence embedding reduction compared to traditional recurrent embedding approaches. Furthermore, meta-marks in green are those previously ignored and considered for the recurrence limitation scheme when the excessive recurrence of other marks is avoided.

The challenge is that recurrent selection limitation must also be implemented as a pseudo-random process since any sequentiality can compromise watermark synchronization if data ordering is redefined. In this scenario it is also important to guarantee the same selection order of the meta-marks during their embedding and extraction, to avoid watermark synchronization failures in case of subset reverse order attack.

A. DYNAMIC TRANSITION PROBABILITIES GENERATION FOR META-MARKS' SELECTION

The challenge is to design a transition probabilities generation for meta-marks' selection that is as blind as possible. For this aim, considering sequential selection is not recommended, we would like probabilities depending only on the number of the meta-marks in the watermark source \mathbf{S} . Table 4 shows a matrix representing a generic watermark source. We assign sequential character values to the meta-marks instead of a single bit to better illustrate the approach.

TABLE 4. Generic watermark source example.

A	B	C
D	E	F

To increase the chances of other meta-marks' selection, first we do not allow the meta-mark selected at stage s to be considered in the next embedding stage. Assume $M(s)$ being the function returning the meta-mark chosen for the embedding stage s . If $M(s) = B$, the rule $M(s+1) \neq B$ is mandatory. Also, since previous selections want to be avoided, if $M(s-1) = A$ another mandatory rule is $M(s+1) \neq A$.

In this example, we use different values for each meta-mark to facilitate the reader's comprehension; nevertheless, different meta-marks can have equal values in the binary string composing \mathbf{S} . For this reason, instead of defining previous rules based on the marks' values, their positions must be used. Let $M(s) = \bar{m}_k^s$. Considering the function $P(\bar{m}_k^s)$ returning the position of the meta-mark selected in the stage s , the following statements formalize our approach.

$$P(\bar{m}_k^{s+1}) \neq P(\bar{m}_k^s) \quad (10)$$

$$P(\bar{m}_k^{s+1}) \neq P(\bar{m}_k^s) - 1 \quad (11)$$

The binary stream composed of the meta-marks in \mathbf{S} will have a circular structure. According to this, the meta-mark immediately before the first one will be the last one. As well, the meta-mark following the last one will be the first one. At each stage, the current and immediately previous positions of meta-marks must be excluded from the pseudo-random selection in the next stage, given by $s + 1$. Considering $\mathbb{P}_h(P(\bar{m}_k^s))$ as the function that returns the horizontal probability for the position $P(\bar{m}_k^s)$ of being selected in the stage $s + 1$ given the current one, then $\mathbb{P}_h(P(\bar{m}_k^s) - 1) = 0$ and $\mathbb{P}_h(P(\bar{m}_k^s)) = 0$, according to (10) and (11). The rest of meta-marks' positions probabilities are assigned according to (13). The number of meta-marks considered by the selection is $n - 2$ since two of them have already zero probability assigned.

$$U_n = 1 / \sum_{i=0}^{n-3} (n-2-i) \quad (12)$$

$$\mathbb{P}_h(P(\bar{m}_k^s) + 1) = \mathbb{P}_h(P(\bar{m}_k^s) + 2) + U_n \quad (13)$$

The position $P(\bar{m}_k^s) + 1$ will have a probability higher by U_n than the following position and the last position available for selection will have probability U_n . So probabilities strictly decrease. Table 5 shows an example for the current selection of the meta-mark B (highlighted in red color and underlined) where meta-marks' order in \mathbf{S} reflects the ordering of the weights of the assigned probabilities.

TABLE 5. Probability weights assigned given current selection of the mark B .

Marks:	A	<u>B</u>	C	D	E	F
\mathbb{P}_H values for $M(s) = B$	0	0	4/10	3/10	2/10	1/10

The probabilities for next meta-marks for being selected should depend just on the current selection, and we can model them by Markov chains according to $\mathbb{P}_h(X_{s+1} = x | X_s = x_s)$, where X_s denotes the current stage selection and X_{s+1} the selection for the next stage. By considering again example of Table 4, the generation transition matrix obtained this way results in:

$$T = \begin{bmatrix} 0 & 0.4 & 0.3 & 0.2 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.3 & 0.2 & 0.1 \\ 0.1 & 0 & 0 & 0.4 & 0.3 & 0.2 \\ 0.2 & 0.1 & 0 & 0 & 0.4 & 0.3 \\ 0.3 & 0.2 & 0.1 & 0 & 0 & 0.4 \\ 0.4 & 0.3 & 0.2 & 0.1 & 0 & 0 \end{bmatrix} \quad (14)$$

Formally, the generation transition matrix is obtained according to (15).

$$T = \begin{bmatrix} 0 & (n-2) & (n-3) & (n-4) & \dots & 1 & 0 \\ 0 & 0 & (n-2) & (n-3) & \dots & 2 & 1 \\ 1 & 0 & 0 & (n-2) & \dots & 3 & 2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (n-2) & (n-3) & (n-4) & (n-5) & \dots & 0 & 0 \end{bmatrix} \times U_n \quad (15)$$

Since the selection of the next meta-mark will depend on the current selection and T, there is the possibility to increase chaos in the process by assigning zero to other meta-marks' probabilities modifying the transition matrix. In this work, we present a symmetric matrix representing linear behavior to understand our proposal better.

According to the current selection, the probability of selecting the element is also combined with another factor, i.e., the recurrent selection boundaries. In order to enforce our approach we consider how the probability of selecting one element varies each time that element is chosen. In this case, the selection instead of being conceived horizontally (from the storing structure perspective) is analyzed vertically (from the stack formed every time the same meta-mark is selected).

Every time the same meta-mark is chosen, the probability for its recurrent selection \mathbb{P}_v , i.e., the vertical probability, is reduced in $1/\rho$ units. For a given \bar{m}_k , \mathbb{P}_v is obtained according to (16), where the function $\bar{\alpha}(s, k)$ returns the number of times the meta-mark in the position k has been selected so far.⁵

$$\mathbb{P}_v(s, k) = 1 - \bar{\alpha}(s, k)/\rho \quad (16)$$

Note that the differences in the meta-marks position notation in (13) and (16) are given by the different domains analyzed. Indeed, when meta-marks are identified according to their spatial location in S, we use k. On the other hand, we use $P(\bar{m}_k^s)$ when the temporal domain is used. The overall probability for selecting \bar{m}_k considering recurrent selection and the stage following the current one is obtained according to (17).

$$\mathbb{P}_{sel}(s, k) = \mathbb{P}_v(s, k) \times \mathbb{P}_h(P(\bar{m}_k^s)) \quad (17)$$

Table 6 shows \mathbb{P}_{sel} values for the watermark source of Table 4 when the meta-mark selected at stage s is B. Each probability depends on the number of times each meta-mark has been selected. The last column shows values for \mathbb{P}_v where we allow to select no more than 6 times. Every time the same position is chosen, \mathbb{P}_v reduces in $1/6$ units until it is equal to 0. When a meta-mark has not been selected, \mathbb{P}_v remains 1.

TABLE 6. \mathbb{P}_{sel} probability table corresponding to Table 4 with $M(s) = B$.

\mathbb{P}_H						\mathbb{P}_V
A	B	C	D	E	F	
0	0	0.4	0.3	0.2	0.1	1
0	0	0.4000	0.3000	0.2000	0.1000	0.8333
0	0	0.3333	0.2500	0.1667	0.0833	0.6667
0	0	0.2667	0.2000	0.1333	0.0667	0.5000
0	0	0.2000	0.1500	0.1000	0.0500	0.3333
0	0	0.1111	0.0833	0.0556	0.0278	0.1667
0	0	0.0444	0.0333	0.0222	0.0111	0
0	0	0	0	0	0	0

⁵Notice the difference between $\bar{\alpha}(s, k)$ and $\bar{\alpha}(\bar{m}_k)$, where the last one returns the number of times the meta-mark was selected when the embedding process is concluded (cf. Section II).

Values in light blue colored cells show the probability \mathbb{P}_h considering the meta-mark currently selected is B. For this reason, and according to (10) and (11), no matter the value of \mathbb{P}_v , A and B will not be considered for the next stage. According to this, $\mathbb{P}_{sel} = 0$ for all combinations involving meta-marks A and B (cf. columns 1 and 2). Even if those columns store always 0 as value, they have been kept in the table to stress the concept of next stage exclusion according to the current selection.

The rest of the meta-marks are assigned a probability depending on how distant they are from the meta-mark currently selected, and by the number of times they have been selected in previous stages. The final probability of selection \mathbb{P}_{sel} is obtained according to (17) and depicted in light orange colored cells.

B. THE CONTEXTUAL-CHAOTIC SEQUENCE GENERATOR

The challenge derived by trying recurrent selection control is that keeping track of $\bar{\alpha}$ requires considering the same sequence to embed and extract WM; otherwise, synchronization gets compromise. For example, having the relation $R[2, 7]$, lets assume that the meta-mark \bar{m}_D of a generic source S is selected 4 times to generate the marks being embedded into positions $T_0[A_0]$, $T_1[A_1]$, $T_2[A_0]$, and $T_6[A_1]$ in that order. If recurrent selection is limited according to $\rho = 2$, positions $T_2[A_0]$, and $T_6[A_1]$ are excluded.

If during WM extraction the order for detecting the marks relative to \bar{m}_D varies, i.e., $T_2[A_0]$ and/or $T_6[A_1]$ are considered and $T_0[A_0]$ and/or $T_1[A_1]$ are excluded, then false positives might get involved in the majority voting performed to assign the final value of \bar{m}_D . If the number of false positives is too high, the final value assigned to the meta-mark will be wrong. Also, when using positions previously denied to marks generated with \bar{m}_D , other meta-marks linked to marks using those places for the embedding will be affected (cf. Fig. 10).

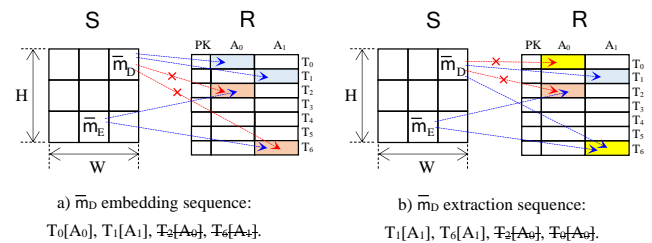


FIGURE 10. Collusion of locations while keeping track of $\bar{\alpha}$.

Fig. 10.a) shows the case when positions excluded from the ones selected to embed marks generated with \bar{m}_D (represented with red arrows) are assigned to marks generated using \bar{m}_E . On the other hand, Fig. 10.b) shows conflicting positions (highlighted in yellow) due to not respecting the same sequence's order for the extraction. In particular, the position $T_0[A_0]$ is excluded and wasted since it is never linked to any other meta-mark. While position $T_6[A_1]$, which was previously excluded and assigned to a mark generated with \bar{m}_E , is included.

As soon as sequential nature is added to the processes, synchronization becomes vulnerable to subset reverse order attacks. Also, suppose the strategy of keeping track of the positions' order consists of storing their sequence in third structures used as a reference for the extraction. In that case, the blindness requirement of the technique is affected.

To face this problem, we combine \mathbb{P}_{sel} with a component pseudo-randomly generated, defined as g_I , providing the same order for the embedding and extraction of WM. We define the region G_I of range λ_I containing the elements used to generate g_I . Considering multiple g_I are generated, we use the notation $\lambda_I^D, G_I^D, g_I^D$ denoting the case of the pseudo-random number D-th (cf. Fig. 11). All g_I numbers are stored in the set C_I , and a broader area of λ_{II} range, identified as G_{II} , is used to define the position of its correspondent g_I in C_I . Thus, even if the same g_I values are generated, their place in C_I will depend on a majority appreciation of their context.

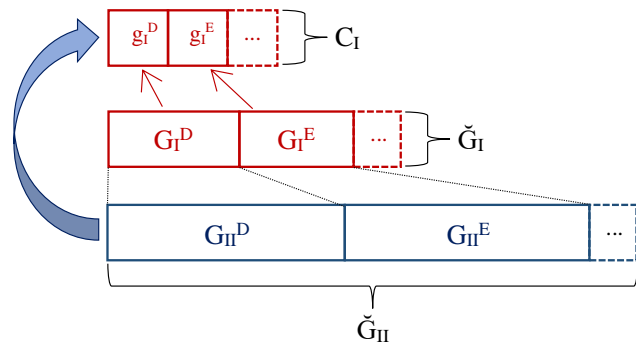


FIGURE 11. Structure of sets for generation and positioning of g_I values.

Both regions, G_{II} and G_I will have the same center g_c , that will be used as reference to embed the mark generated with the selected meta-marks in G_I . According to this, $G_I \subseteq G_{II}$ (cf. Fig. 12.a)).

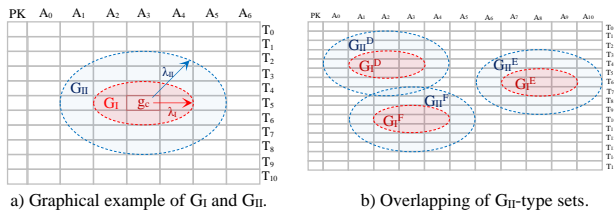


FIGURE 12. Graphical view of a definition of G_I and G_{II} .

The principle of the majority contextual appreciation is similar to the majority voting described in Section I. Their main difference is that for the majority voting are considered candidates' values for one meta-mark whereas for the majority contextual appreciation, the elements contained in the region G_{II} surrounding those contained in the region G_I are analyzed. Given the context analysis requirement, G_{II} sets can intersect each other without major consequences for the outcome. Nevertheless, this is not allowed for G_I sets (cf. Fig. 12.b)).

The way elements are analyzed to be considered by every region depends on values that cannot be modified without compromising data quality for both data owners and attackers (e.g., *msb* of carriers, locked attributes).⁶ This makes possible for the context to persist despite small variations, overcoming impacts due to G_{II} intersection, benign updates and malicious operations, and guaranteeing the persistence of C_I order.

$$\mathbb{T} \leq \Delta(\check{G}_{II}, \check{G}'_{II}) \quad (18)$$

Let \mathbb{T} be the threshold of allowed contextual differences and Δ be the function that evaluates them considering all the G_{II} sets generated for embedding the watermark signal (given by \check{G}_{II}) and those generated for the watermark extraction (given by \check{G}'_{II}). Then, with (18) we can obtain an accurate idea of the degree of changes in the context and of the effects of the latter over C_I . If the contextual difference between \check{G}_{II} and \check{G}'_{II} exceeds the threshold \mathbb{T} , the watermark synchronization will fail.

C. WATERMARKING ARCHITECTURE

This work focuses on robust watermarking techniques created for the copyright protection of relational data. These approaches are based on the principles of generation and embedding of WM, and the WM extraction is carried out under suspicion of piracy or false ownership claims. The main architecture of our approach is consistent with the processes described in Section I-A.

We perform the addition of Recurrence Reduction Engine (RRE) in the IBW technique described in Section II. Our proposal inherits features from [17] such as numerical cover type and multi-attribute. As depicted in Fig. 2 of Section I-B, the reduction of recurrent selection firstly takes place in the selection sub-process of the watermark embedding. The core of RRE consists of combining the pseudo-random generation of g_I numbers (handled by the 'Chaos Generator') and managing \mathbb{P}_{sel} probabilities (performed in the 'Probability Box'). The watermark source S and the relation being watermarked R are the main inputs of the embedding process. Nevertheless, they have a particular link with RRE, since the 'Chaos Generator' also uses R to define C_I and the 'Probability Box' interacts with S to assign the probability of each meta-mark for their later selection, according to the current stage.

Meta-marks resulting from the selection sub-process containing RRE must guarantee to consider more marks already at that early stage of the embedding compared to techniques that do not implement RRE. Once the meta-marks have been selected, the generation of marks is performed considering fixed elements from their embedding position.⁷ When the mark is generated, its insertion on R may apply.

A low-level description of the embedding process deploying RRE with the meta-marks' selection is formalized by Algorithm 2. In our approach, the role of the watermark source

⁶Locked attributes are attributes with nature and relevance not prompt to changes.

⁷Fixed elements are data that cannot be modified without compromising data usability.

Algorithm 2: Watermark embedding with RRE.

```

1  $C_H[n-1] = \rho$ 
2  $P_H[n-1] = 0$ 
3  $P_S[n-1] = 0$ 
4  $P_{acc} = 0$ 
5 for  $i = 1$  to  $n-1$  do
6    $P_H[i] = n - (i + 1)$ 
7    $P_S[i] = P_H[i] \times C_H[i]$ 
8    $P_{acc} += P_S[i]$ 
9  $C_I \leftarrow \text{generate\_}C_I(\lambda_I, \lambda_{II})$ 
10 foreach tuple  $T \in R$  do
11   if  $F(T[PK]) \bmod \gamma = 0$  then
12     foreach attribute  $A \in AL$  do
13        $a_v \leftarrow \text{generate\_AVV}(A, \beta)$ 
14        $a_h \leftarrow \text{generate\_AVH}(a_v, F(T[PK]))$ 
15       if  $a_h \bmod \delta = 0$  then
16          $b_\xi \leftarrow F(T[PK]) \bmod \xi$ 
17          $b_\beta \leftarrow \text{getMSB}([A]_2, \beta)$ 
18          $k \leftarrow \text{getMarkPos}(C_I, T, A, P_{acc})$ 
19          $m = \bar{m}_k \oplus b_\beta$ 
20          $A \leftarrow \text{mark}(T, A, b_\xi, m)$ 
21          $C_H[k] = C_H[k] - 1$ 
22          $P_H \leftarrow \text{rotate\_}P_H(P_H, n, k)$ 
23          $P_S \leftarrow \text{update\_}P_S(P_S, P_H, C_H)$ 
24          $P_{acc} \leftarrow \text{update\_}P_{acc}(P_S, P_{acc}, C_H)$ 

```

structure is higher since probabilities are given according to meta-marks position in S . Considering the positions of meta-marks are fixed, there is no risk of subset-reverse order attack over the watermark source, which backs up the feasibility for our proposal implementation.

Let's discuss some details of Algorithm 2. The array C_H contains the number of times each meta-mark can be considered for generation of marks to embed. For this reason, the values stored on each position are initialized to ρ (line 1). Values in C_H never switch positions, but are reduced by one each time the meta-mark correspondent to the same position is selected (line 21).

The array P_H stores the weight of selecting each meta-mark according to a number representing the probability in scale. Values of this array represents \mathbb{P}_h values and are switched according to the current selection given the rules introduced in Section III-A, by the function $\text{rotate_}P_H(P_H, n, k)$ (line 22), where k represents the position of the current meta-mark selected in S and n the watermark source's length. Initial values are assigned to each position of P_H (line 6). The set of numbers must describe the same slope of values obtained with equation (13). In the case of using decimal numbers, original values obtained with (13) can be used.

Fig. 13.b) depicts the projection with integer values of the original probability distribution generated for Dào WM, which might be used in case of engines taking integer values

as a virtual probability. By considering the image size and reducing by one the next value to assign in the correspondent P_H position, a distribution is generated with the same slope of the original probability distribution (cf. Fig. 13.a)). Fig. 13.c) shows the proportion between the values assigned to each meta-mark (original vs. projected ones), validating the projection with integers.

The array P_S represents the overall probability \mathbb{P}_{sel} of equation (17). Values assigned to each array's position are generated by considering the number of times available for selecting each meta-mark, according to \mathbb{P}_v defined in (16). The result of $C_H[i] \times P_H[i]$ returns $P_S[i]$, considering $C_H[i]$ being a scaled value of \mathbb{P}_v . Note that values stored in P_S are computed in line 7. Once an embedding is performed and C_H and P_H are updated, P_S is updated as well (line 23).

The term P_{acc} (line 4) represents the accumulated probability value to generate the random number inside the available set of probabilities (line 18). Each time P_S varies, P_{acc} is updated (lines 8 and 24). The generation of random numbers to increase the entropy of meta-marks selection is performed according to the techniques discussed in Section III-B. In line 9, the function $\text{generate_}C_I(\lambda_I, \lambda_{II})$ takes the radii of regions G_I and G_{II} to generate the set C_I , which contains g_I numbers with the order to be used for the meta-marks' selection.

Lines 10-15 detail the selection of elements being watermarked according to AHK and [17]. Line 16 computes the *lsb* position b_ξ to store the mark, and line 17 computes a *msb* value b_β out of the ones given by β . In line 18, the function $\text{getMarkPos}(C_I, T, A, P_{acc})$ uses C_I to extract the g_I corresponding to position $T[A]$ and uses it as seed of a random engine to select a position of the meta-mark according to the accumulated probabilities P_{acc} .

Once the position k is obtained, the meta-mark \bar{m}_k is extracted and the mark to embed is generated by *x-oring* \bar{m}_k and b_β (the symbol \oplus denotes the x-or operator) (line 19). Next, the mark is embedded in the *lsb* position b_ξ of the value stored in $T[A]$. Strategies for reducing distortion caused by the mark embedding according to [7] and [17] are implemented by the $\text{mark}(T, A, b_\xi, m)$ function of line 20.

Finally, for the selected meta-mark, C_H is decreased by one unit, P_H is switched according to the current selection, P_S and P_{acc} are updated. The iteration continues until all values of R are analyzed.

The extraction process takes place following the same principle for watermark embedding. In this case, the process is featured by a flow of detected marks from R to S . For every stage, a meta-mark is extracted following AHK and marked values are checked using rules defined in [17]. If a value is stored in a position identified as mark container, the meta-mark is extracted according to $\bar{m} = b_\xi \oplus b_\beta$, where b_ξ contains the mark stored during the embedding in the correspondent *lsb*. The position k to store the extracted meta-mark is obtained the same way as for the embedding process.

At last, different meta-marks' values can be extracted for the same position k . A middle layer will store all repeated values, and after the extraction is finished, a majority voting

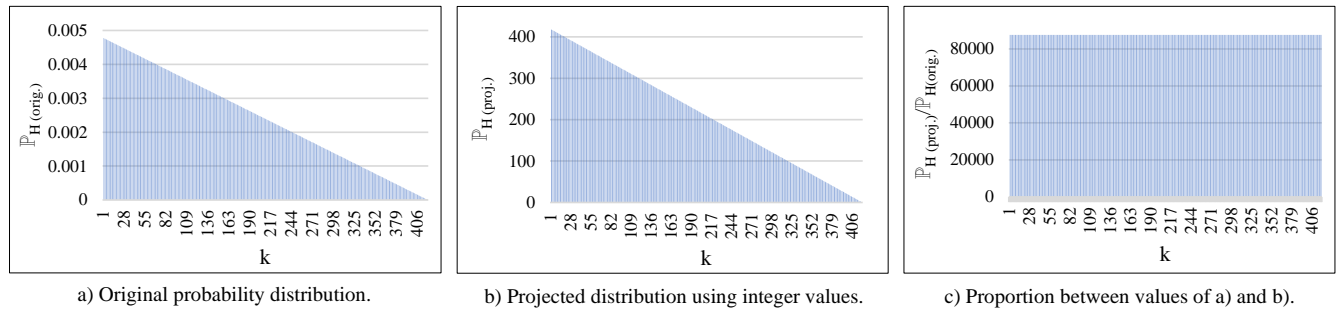


FIGURE 13. Dao WM probability distribution projected as integer numbers set.

is performed the same as with previous techniques. After the majority voting takes place, the source S is built and compared with the original according to (8) and (9). The conclusions of false ownership claims or piracy are delivered depending on the results.

IV. ANALYSIS

Limiting the number of times the same meta-mark is selected for the embedding should be done carefully. The value assigned to ρ could derive in higher robustness or proceeding with a weak protection of R as well.

There are some options about how to apply the engine proposed in this work, but all of them must take into account the number of times the marks are expected to be embedded into R , which requires taking into account the relation's dimension v and η , and the embedding parameters γ and δ .

A. THE PROBABILITY-BASED SYNCHRONIZATION

The probability-based embedding with *pseudo-random* seed generation and recurrence limitation has advantages of meta-marks uniform selection and scattering of embedding locations. Thanks to this, resilience against malicious attacks increases. Nevertheless, an important downside of this approach is that the detection is not 100% accurate. Majority voting can help to overcome this problem if ρ is not too low. On the other hand, the right detection of neighboring meta-marks can help to correct false-positives added to S' . To proceed this way, correction algorithms must try to recover missed meta-marks as well as correct wrong meta-marks' values.

To know which action to take to benefits majority voting but also allowing correction algorithms to be useful, it is important to monitor the performance of the watermark detection process. According to AHK, ω denotes the number of marked tuples. When only one mark per tuple is embedded, then ω can also be used to know the number of embedded marks and can be obtained according to $\omega \approx \eta/\gamma$. In a multi-attribute embedding setting $\omega \approx \eta/\gamma \times v/\delta$ and it reflects the number of embedded marks.

The false positive rate $\mathcal{FP} \in \mathbb{Q}^+ : 0 \leq \mathcal{FP} \leq 1$ is expressed according to (19), where $\hat{\omega} \in \mathbb{Z}^+$ is the number of marks extracted with the incorrect value out of the number of marks embedded given by $\omega \in \mathbb{Z}^+$. When $\mathcal{FP} = 1$ means all

extracted marks do not correspond to the expected ones, rising suspicious about malicious operations being applied over R .

$$\mathcal{FP} = \hat{\omega}/\omega \quad (19)$$

Another metric helping to measure the quality of the detection process is the detection accuracy $\mathcal{A} \in \mathbb{Q}^+ : 0 \leq \mathcal{A} \leq 1$, defined in (20). In the equation, $E_{\bar{m}} \in \mathbb{Z}^+$ is the number of meta-marks embedded and $C_{\bar{m}} \in \mathbb{Z}^+$ the number of meta-marks having the correct value once the majority voting is performed and S is reconstructed.

$$\mathcal{A} = C_{\bar{m}}/E_{\bar{m}} \quad (20)$$

The best case-scenario of WM extraction with respect to accuracy is when $\mathcal{A} = 1$. This means all meta-marks of S were correctly identified, even if some marks were extracted with the wrong value (i.e., $\mathcal{FP} \neq 0$). This last situation is possible thanks to the role played by the majority voting step in the construction sub-process.

B. VERTICAL DIGGING VS. HORIZONTAL SPREADING

Fig. 14 and 15 allow us to compare the distribution of the meta-mark selection when RRE is applied or not, respectively. Blue boxes and blue dots represent meta-marks selected in a particular recurrence level. Red boxes and red dots denote meta-marks being ignored in the first recurrence level, or considered by previous recurrence level and ignored at that point. Considering S as an image, axis x represents the source's width, axis y its height and axis z the number of times the meta-mark of position (x,y) have been selected.

Vertical digging and horizontal spreading are definitions that arises when using meaningful sources for the watermark generation and the performance of recurrence embedding. When using meaningful sources, neighboring meta-marks of a contrasting value can be used to correct it. This can be performed by applying enhancement algorithms such as noise-reduction methods for images.

This type of operation increases the probability of success if more meta-marks are considered for the embedding. We define this property as horizontal spreading of meta-marks' selection (cf. Fig. 14) and it is directly benefited by applying RRE proposed in this work.

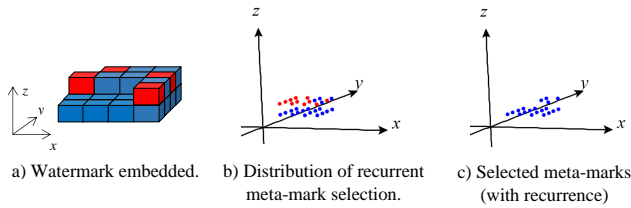


FIGURE 14. Horizontal spreading of meta-mark selection with controlled recurrence RRE.

On the other hand, recurrent embedding contributes to avoid the extraction of false positives meta-marks, which contributes as well to the technique's robustness. We define the recurrent embedding as vertical digging over S (cf. Fig. 15) and even if is a recommended feature, it should not be performed at expenses of horizontal spreading's costs. For this reason, it is very important to exploit all possible locations from R for the embedding, as long as data quality remains.

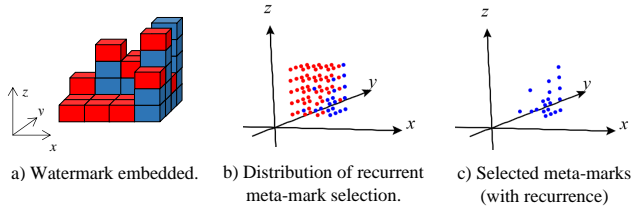


FIGURE 15. Vertical digging of meta-marks selection due to uncontrolled recurrence.

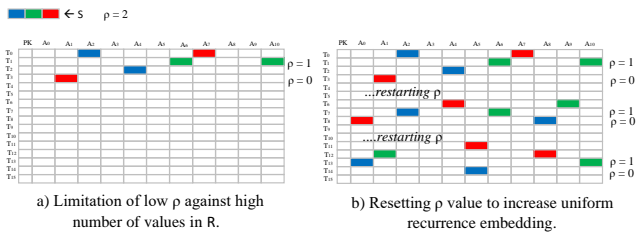


FIGURE 16. Controlling recurrence selection when $\omega \gg n$ considering low values of ρ can limit ω whereas there is still room for marks' embedding (each meta-mark is highlighted using different colors, for their easy identification in R).

In this case, there is no need to assign a value too low to ρ if the values of ν and η , considering γ and δ , guarantee the condition $\omega \gg n$, where the operator \gg describes the relation *much more greater than* (e.g., the number of embedded marks ω is much more greater than the number of meta-marks n). Then vertical digging does not have to be compromised to the benefit of horizontal spreading.

Another way to assign a value for ρ that does not restrict the use of all embedding places that R offers is by knowing first the values of $\max(E)$ and \bar{m} introduced in Section II. Nevertheless, this option is not optimal regarding performance, considering it requires first the embedding of the watermark without considering RRE to obtain the values of those metrics.

Another downside of selecting a low value of ρ , if n is relatively small, is limiting marks embedding to just a portion of R while all the meta-marks generating them were considered (cf. Fig. 16.a)). Indeed, in this scenario, all meta-marks are selected a number of times too low, compromising the benefits expected from the majority voting in the extraction process. Then, a low-level update-based attack could affect the watermark recognition. This can be overcome by restarting the value of ρ once all meta-marks are considered and $\mathbb{P}_{acc} = 0$, being possible to continue the meta-mark selection in a uniform way until all tuples and attributes of R are analyzed (cf. Fig. 16.b)).

C. WATERMARK EMBEDDING EXAMPLE

Before experimentally validate our approach, we show the effect of performing the watermark embedding while considering a small binary image S . According to the pixels' values, $S = \langle 1, 0, 1, 0, 0, 0, 1, 0, 1 \rangle$.⁸ In this case, $n = 9$ and $\rho = 3$. Also, as previously stated, for $s = 0$ it is assumed that the first meta-mark is already selected. Taking this into consideration P_H and P_S are initialized as $P_H = \langle 0, 7, 6, 5, 4, 3, 2, 1, 0 \rangle$ and $P_S = \langle 0, 21, 18, 15, 12, 9, 6, 3, 0 \rangle$.

Fig. 17 depicts the arrays representing the watermark source S , the tracker of embedding times C_H , P_H , and P_S aligned for each stage s . The red lined rectangle represents the pseudo-random position selected for the next stage. The blue boxes represent items selected already once (i.e., for $\rho = 3$, $C_H[i] = 2$), orange boxes represent items selected twice (i.e., for $\rho = 3$, $C_H[i] = 1$), and yellow boxes elements selected already three times (i.e., for $\rho = 3$, $C_H[i] = 0$).

On each stage, after updating C_H , P_H is rotated according to the position selected and P_S is generated considering C_H and P_H according to Algorithm 2. Once $C_H[i] = 0$ for a particular position, the same position will be assigned $P_S[i] = 0$, taking that meta-mark out of the random-selection's consideration.

At every stage, the figure shows the accumulated probability \mathbb{P}_{acc} obtained by summing all items from P_S . The process stops when $\mathbb{P}_{acc} = 0$ (stage $s = 26$).

V. EXPERIMENTAL EVALUATION

This section analyzes the components of the proposed watermarking architecture and the watermarking features directly affected by applying RRE, such as capacity and robustness. In addition, most of the metrics introduced in previous sections are used to evaluate and compare the results with those previously presented in Section II.

A. EXPERIMENTAL SETUP

We use watermark sources of different sizes to illustrate the effects of involving different values of n , similarly to Section II. The sources selected are the binary images introduced in Fig. 4.

The data used to embed the marks is the widely known numerical data set *Forest Cover Type*, available in [18]. To

⁸For convenience, we also represent the binary stream composing S as a binary array.

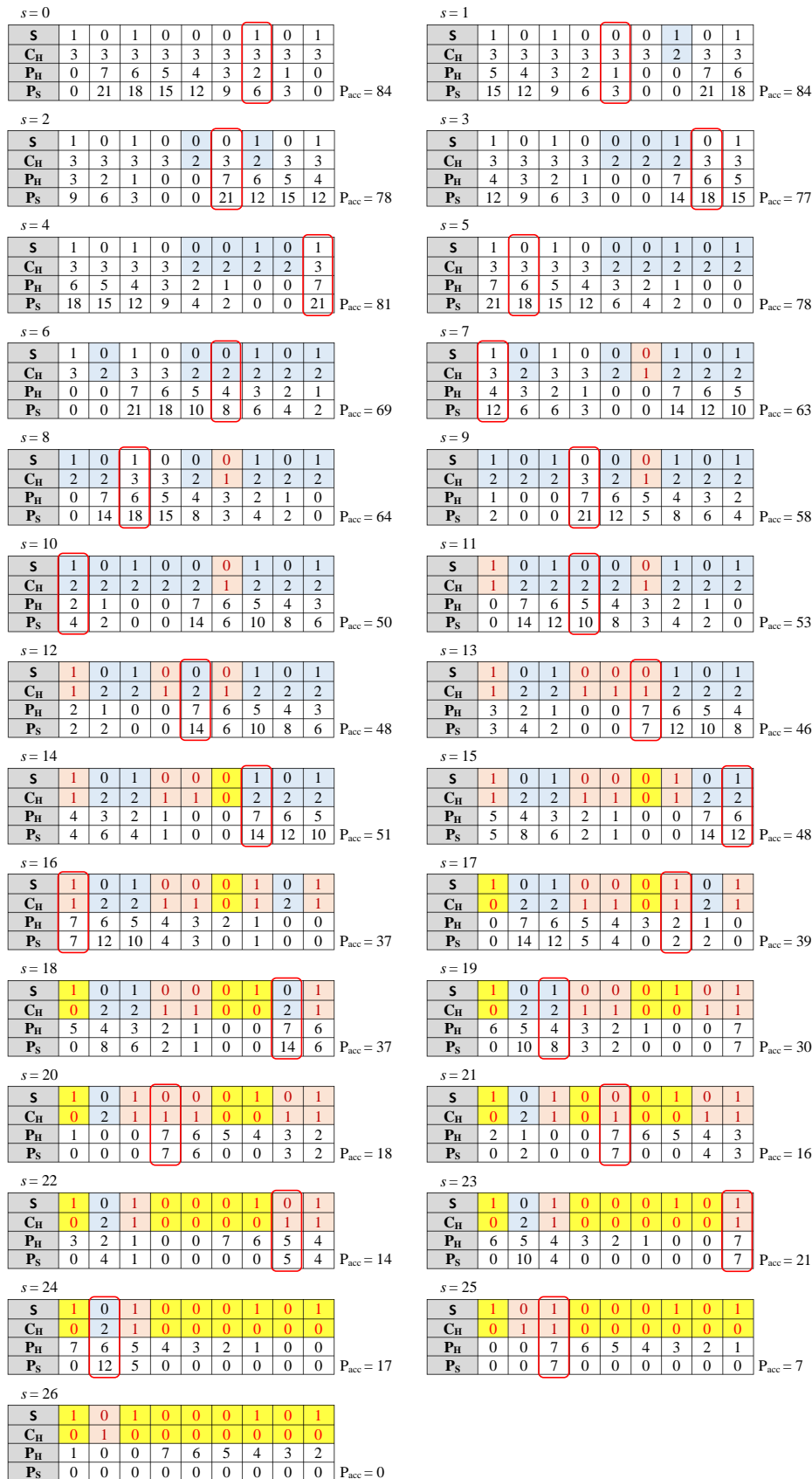


FIGURE 17. Simulation of pseudo-random meta-mark's selection ($\rho = 3$).

obtain results under the same conditions of previous works and allow a fair comparison, we perform the embedding in the first 30K tuples out of the 581K stored in the relation. Also, we consider only the first ten attributes (out of 54).

The implementation of our proposal follows a client-server architecture. The client layer was developed using Java 1.8 and Eclipse Integrated Development Environment (IDE) 4.21. The database server engine was Oracle Database 19C. The considered IDE for database management was Oracle SQL Developer 21.4. The runtime environment was a 4.20GHz GHz Intel i7-7700K PC with 32.0 GHz of RAM with Windows 10 Pro OS.

B. BENEFITS OBTAINED IN TERMS OF WATERMARK CAPACITY

Fig. 18 shows the values of \mathbb{P}_h for each meta-mark of Dào and WWF image sources at the initial stage. Since the watermark generated from Dào image has fewer meta-marks, the probabilities will experience a faster drop than those related to WWF.

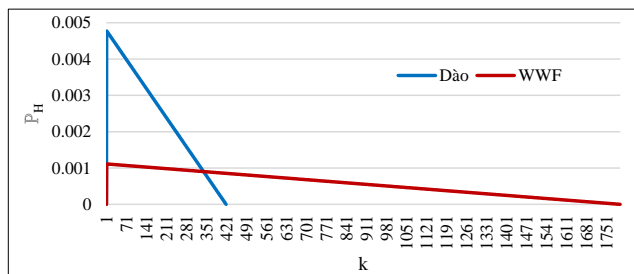


FIGURE 18. Initial values of \mathbb{P}_h for Dào and WWF meta-marks.

When the number of meta-marks n is low, the differences among values of \mathbb{P}_h is high. This means that, for two neighbors meta-marks \bar{m}_i and \bar{m}_j (with $i \neq j$), $\bar{m}_i - \bar{m}_j$ will be higher for Dào than WWF. Also, if γ and ν values do not change, i.e., if the amount of data being watermarked does not vary, all meta-marks of a small source are reconsidered first with respect to those belonging to a bigger source.

We used $\lambda_I = 1$ and $\lambda_{II} = 3$ for the generation of C_I . Table 7 shows the quality of the embedded watermark generated from the UTM source for different values of γ . Given the high value of n and the reduction of tuples selected for marking when γ increases, the quality of the embedded watermark reduces. Nevertheless, compared to embedding without RRE (cf. Table 1), the quality improves thanks to the uniform meta-marks selection. The presence of red pixels in the extracted signal shows that all possible embedding places from R were considered, but not all meta-marks were selected ρ times.

Results shown in Table 8 presents a general improvement compared to those obtained by using the same WWF source and not applying RRE (cf. Table 2). Note that in this case, since n is lower than for UTM, the improvement of the capacity is higher.

Table 9 presents the results when the Dào image is used as the source. For this case, since n is very small, independently

TABLE 7. Differences among embedded UTM WM in R using Algorithm 2.

γ	5	10	20	30
ρ	3	2	1	1
Generated Image				
$\max(E)$	3	2	1	1
c_b	63.13	37.27	21.01	13.70
c_w	0.86	0.42	0.21	0.14
σ_w	0.79	0.57	0.41	0.34
$\Theta(E)$	52.63×10^2	42.68×10^2	33.83×10^2	26.14×10^2

TABLE 8. Differences among embedded WWF WM in R using Algorithm 2.

γ	5	10	20	30
ρ	5	3	2	1
WM Image				
$\max(E)$	5	3	2	1
c_b	98.94	87.66	62.11	49.88
c_w	3.12	1.51	0.76	0.50
σ_w	1.11	0.86	0.68	0.50
$\Theta(E)$	16.07×10^2	18.26×10^2	16.36×10^2	17.96×10^2

TABLE 9. Differences among embedded Dào WM in R using Algorithm 2.

γ	5	10	20	30
ρ	10	5	3	2
WM Image				
$\max(E)$	10	5	3	2
c_b	99.76	99.76	99.76	99.76
c_w	9.98	4.99	2.99	2.00
σ_w	0.49	0.24	0.15	0.10
$\Theta(E)$	85.97×10^1	17.19×10^2	28.66×10^2	42.99×10^2

from the value of γ , the quality of the image generated from the embedded watermark is higher. Also, compared to results of Table 3 and despite the simplicity of S , the capacity experiences the more significant improvement.

It is important to notice that, for experiments shown in Tables 7, 8 and 9, c_b increases compared to embedding not applying RRE. On the other hand, c_w remains almost always the same, whereas σ_w reduces due to the uniform meta-marks selection. Finally, when considering more meta-marks $\Theta(E)$ increases, whereas a reduction of red pixels is spotted in the images generated from the embedded watermark signal.

Tables 10, 11 and 12 present a more straight comparison of the quality of the embedded signals using different values of ρ against the signal embedded not applying RRE. Each table presents the case of each watermark source. The column having “N/A” (does not apply) as ρ value denotes the case of watermark embedding without RRE. For all the other cases, i.e., when RRE is applied, the same value of δ and γ are used to show how ρ affects the quality of the embedded watermark signal without varying the expected number of values being watermarked from R .

TABLE 10. Relevance of ρ ($\gamma = 10$, UTM WM).

ρ	N/A	1	2	3
WM Image				
max(E)	4	1	2	3
c_b	34.25	41.54	37.27	35.64
c_w	0.42	0.42	0.42	0.42
σ_w	0.64	0.49	0.57	0.61
$\Theta(E)$	35.03×10^2	55.30×10^2	42.68×10^2	38.55×10^2

TABLE 11. Relevance of ρ ($\gamma = 10$, WWF WM).

ρ	N/A	1	2	3	4
WM Image					
max(E)	6	1	2	3	4
c_b	76.33	99.94	94.11	87.66	85.06
c_w	1.51	1.00	1.51	1.51	1.51
σ_w	1.29	0.02	0.61	0.86	0.95
$\Theta(E)$	10.65×10^2	76.35×10^3	27.94×10^2	18.26×10^2	16.04×10^2

TABLE 12. Relevance of \mathcal{R} ($\gamma = 10$, Dào WM).

ρ	N/A	1	3	5	7
WM Image					
max(E)	13	1	3	5	7
c_b	99.28	99.76	99.76	99.76	99.76
c_w	6.48	1.00	2.99	4.99	6.48
σ_w	2.61	0.05	0.15	0.24	0.73
$\Theta(E)$	16.00×10^1	85.97×10^2	28.66×10^2	17.19×10^2	57.02×10^1

Similarly to the results of Tables 7, 8 and 9, Tables 10, 11 and 12 depict the benefits of using a source with a low value of n in terms of the image reconstructed from the embedded watermark. Nevertheless, the case of WWF from Table 11 shows the more remarkable achievements considering the number of red pixels recovered in comparison with the embedding without applying RRE.

The tables above confirm also the relevance of the metric $\Theta(E)$ as an objective way to measure the benefits of RRE when the number of tuples and attributes considered for the embedding does not vary. From the data shown in these tables, the relationship $\Theta(E) \propto 1/\rho$ (where \propto denotes the proportionality relationship) can be empirically derived.

C. DETECTABILITY EVALUATION

When no malicious operations or benign updates are performed over R , the watermarking architecture without RRE guarantees a perfect detection of all marks. Nevertheless, due to the exclusion of meta-marks, the embedding of S is often partially compromising its recognition even when all embedded marks are correctly detected. On the other hand, the integration of RRE adds uniformity to the meta-marks selection. Still, since the process randomly selects the meta-marks based on their probability, the detection is not 100% accurate. However, the effects of wrong meta-marks values over the reconstructed S' reduce when the number of tuples in R is high.

A way to measure the quality of detectability by spotting the number of false hits during the watermark extraction is to compute the false positive rate \mathcal{FP} and the detection accuracy \mathcal{A} introduced in Section IV according to (19) and (20) respectively.

Table 13 shows the quality of the detected signal for $\gamma = 5$ and $\gamma = 10$, when the embedded watermark was generated using each one of the sources introduced in Figure 4. In the Table, the bigger image represents the reconstructed S' from the detected watermark. On the other hand, the smallest image depicts the part of S considered for the embedding.

The random engine used was based on the Random class of the package `java.util` contained in Java Development Kit (JDK). The methods `setSeed(long arg)` and `nextInt(int arg)` were used to assign the seed and for the random generation of numbers respectively. Despite the randomness added with the engine, for the same seeds and without any operations performed over the protected data, the process guarantee for each case a perfect detection (i.e., $\mathcal{FP} = 0$, and $\mathcal{A} = 1$).

TABLE 13. Accuracy of detection process ($\rho = 3$).

TF	UTM	WWF	Dào
5			
10			

D. ROBUSTNESS ANALYSIS

As previously stated, the uniform selection of meta-marks contributes to embedding a stronger watermark signal. Then, it is

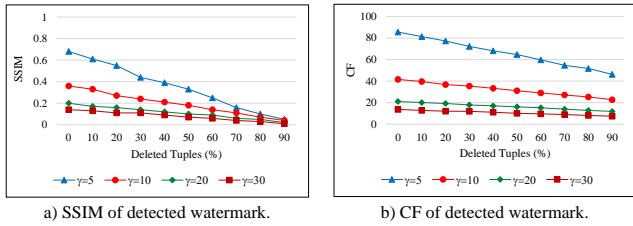


FIGURE 19. Detected UTM WM with RRE after tuple deletion attacks ($\rho = 1$).

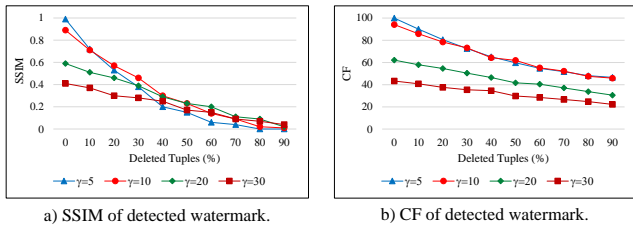


FIGURE 20. Detected WWF WM with RRE after tuple deletion attacks ($\rho = 2$).

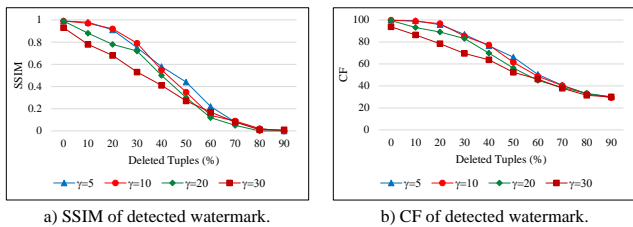


FIGURE 21. Detected Dào WM with RRE after tuple deletion attacks ($\rho = 5$).

expected to achieve higher robustness to malicious operations since the watermark quality is higher than embedding without RRE. Also, the generation of C_I and RRE must be resilient against updates over R' avoiding the computing of seeds for the watermark detection different from the ones used for the embedding.

In Section V-C was already tested the detection under lack of updates. Here we test the proposal with respect to the quality of the detected signal while increasing the attacks' degree.

The case of deletion of tuples was used considering it as the most aggressive form of subset attack. Then, resilience to other types of subset attacks, such as updates and insertion of new data will be featured by a higher resilience than the one spotted in these experiments. Figures 19, 20, and 21 show the resilience of the watermark signal after different levels of tuple deletion attacks are performed over R' . The comparisons are done with respect to results of Fig. 5, 6, and 7, where the extracted remaining signal after performing the same attack is shown.

The watermark generated from UTM and embedded considering RRE experiences higher robustness against tuple deletion (see Fig. 19). For this case, $\rho = 1$, which is not good vertical digging, but considering not all meta-marks are considered due to the number of tuples in R , horizontal

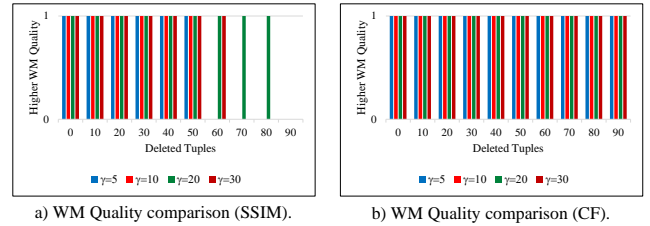


FIGURE 22. Cases when UTM WM present higher resilience.

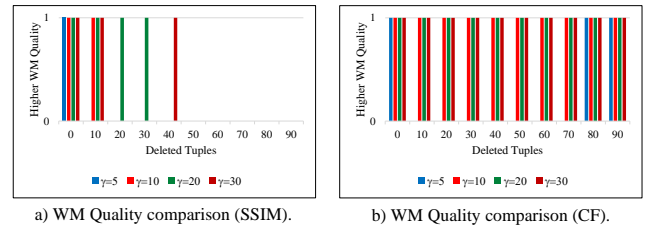


FIGURE 23. Cases when WWF WM present higher resilience.

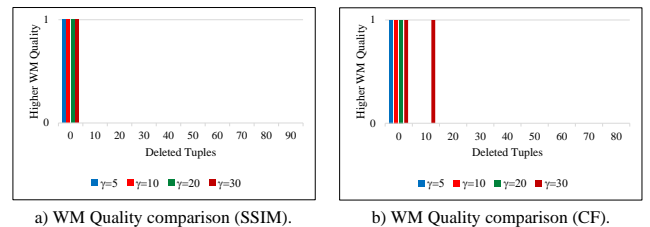


FIGURE 24. Cases when Dào WM present higher resilience.

spreading gets the direct benefits, allowing enhancement algorithms to reconstruct the signal if need it.

For the case when the embedded watermark is generated from WWF (cf. Fig. 20), it was used $\rho = 2$. The value is low, but since some meta-marks are still not considered when no RRE is applied, we intent a small recurrence whereas other meta-marks are included in the process. For this case, the quality of the remaining watermark signal drops faster than the previous case, which suggest a better performance of our approach with sources of bigger sizes.

Finally, we perform the text with the watermark generated from Dào (cf. Fig. 21) by using $\rho = 5$. This case confirms previous statement, but also contributes to reduce relevance for certain parameters, such as γ , which helps to obtain the same results performing the embedding without causing to much distortion over the data. This is a direct consequence of applying RRE over sources with low values of n .

Fig. 22, 23, and 24 depict the difference for each source between embedding the same watermark considering RRE and using uncontrolled recurrent selection. For each case, the bar means positive difference, which indicated higher resilience when RRE is applied. When no bar is shown, the difference is negative, describing higher resilience for no recurrent controlling.

The reduction of the resilience goes directly with the

reduction of n of S . Since the value of ρ is too low for those cases, it is important to achieve a compromise between vertical digging and horizontal spreading by using higher values of ρ .

Fig. 25, 26, and 27 show the accuracy of the detection process after tuple deletion attacks are performed. In this case, it is reflected how the pseudo-random nature depends on the use of same seeds, otherwise it gets affected when some elements from R vary. On the other hand, with this experiments it is confirmed once more then reduction of the role played by γ , particularly when n is higher.

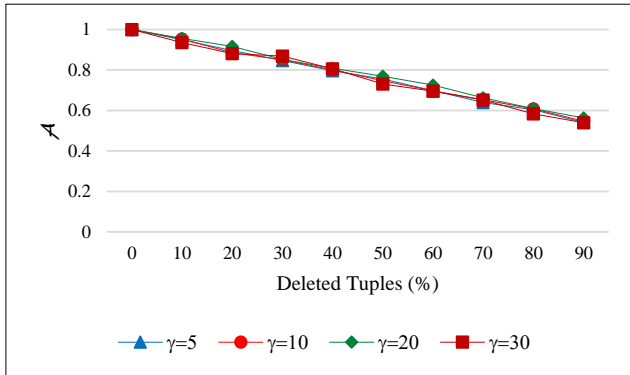


FIGURE 25. Accuracy for UTM WM detection after deletion attacks.

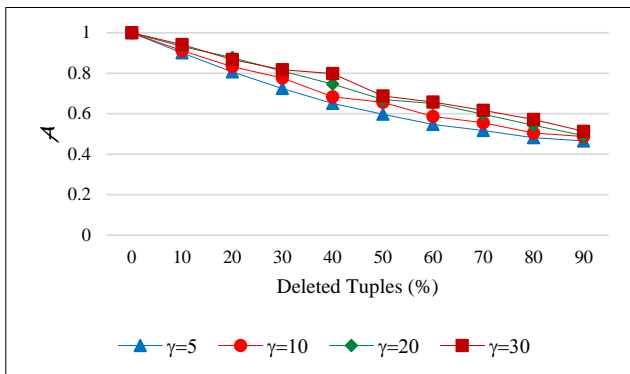


FIGURE 26. Accuracy for WWF WM detection after deletion attacks.

E. IMPACT OF RRE WITH RESPECT TO PERFORMANCE

It is essential to evaluate the additional cost of integrating RRE into the watermarking architecture. The performance of the “Probability Box” and the “Chaos Generator” must not compromise the time taken to perform the watermark synchronization. Furthermore, the integration of RRE must perform accordingly to the number of tuples in R . For its application to be feasible, the technique must be featured by a linear increment of the time required for watermark synchronization when the number of tuples in R increases linearly.

The first experiment is meant to analyze the complexity of RRE regarding the generation of C_I . Considering that this

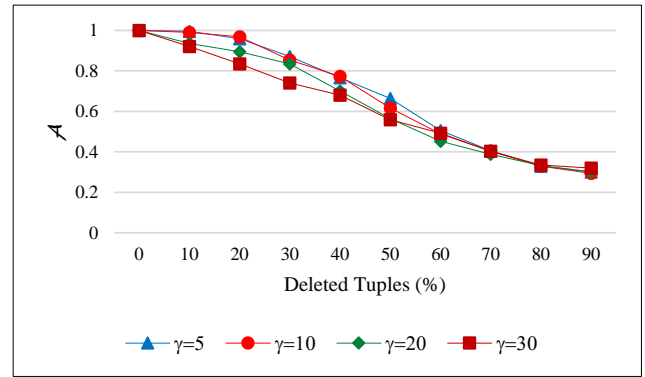


FIGURE 27. Accuracy for Dào WM detection after deletion attacks.

step is not performed when RRE is not applied, this should be considered as an additional performance cost. On the other hand, if there is no risk of primary key elimination, the values of columns used to identify each tuple can be used as C_I , and there is no need for extra time consumption generating C_I .

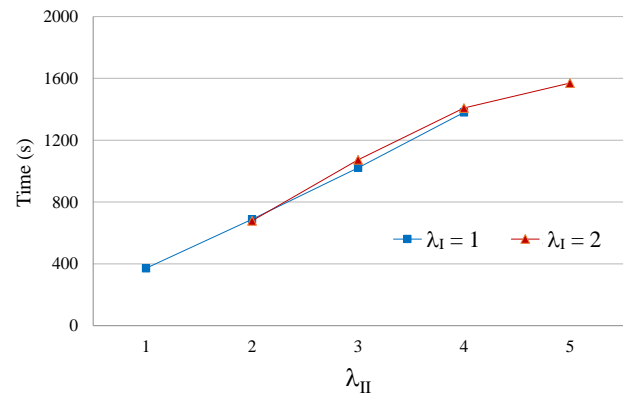


FIGURE 28. Performance of random generation of C_I .

Fig. 28 depicts the time required to generate C_I for two cases, when $\lambda_I = 1$ and $\lambda_I = 2$. For each case, λ_I keeps the same value whereas λ_{II} increases. In the figure can be depicted the linear increment of time required to generate C_I , directly proportional to λ_{II} increment.

Another important feature to analyze is the quality of each version of C_I to perform watermark synchronization. One important feature to track is the number of repeated elements, since a high number of duplicates can compromise watermark synchronization. Nevertheless, with the parameter used to generate C_I no duplicate values of g_I are obtained. Because of this, high watermark capacity is achieved. Tables 14 and 15 compare the value of the standard deviation of each set of Fig. 28 with the one composed by the primary keys values. Considering that for the primary keys, the value is assigned using a serial integer i.e., increasing in one the value tuple per tuple, the scatter of G_I and G_{II} is much higher. For each case of C_I , no duplicate values were spotted.

TABLE 14. Quality of G_I vs. PKs ($\lambda_I = 1$).

λ_{II}	1	2	3	4
PK	86.60×10^2	86.60×10^2	86.60×10^2	86.60×10^2
$\sigma(G_I)$	12.39×10^8	12.39×10^8	12.39×10^8	12.39×10^8
$\sigma(G_{II})$	12.39×10^8	12.47×10^8	12.35×10^8	12.44×10^8

TABLE 15. Quality of G_I vs. PKs ($\lambda_I = 2$).

λ_{II}	2	3	4	5
PK	86.60×10^2	86.60×10^2	86.60×10^2	86.60×10^2
$\sigma(G_I)$	12.44×10^8	12.44×10^8	12.44×10^8	12.44×10^8
$\sigma(G_{II})$	12.47×10^8	12.35×10^8	12.44×10^8	12.41×10^8

Finally, we evaluate the time required to perform the embedding using RRE. We compared our approach with a technique not applying RRE. Figure 29 shows the case when the sets of primary keys are used as C_I .

The watermarking was performed using as parameter values $\gamma = 10$, $\delta = 10$, and $\rho = 5$. As watermark source was selected the image Dào. The number of tuples considered was increased in 15×10^3 each time from 15×10^3 to 90×10^3 . As shown in the Figure, when C_I , the embedding applying RRE is performed almost in the same time as when RRE is not considered.

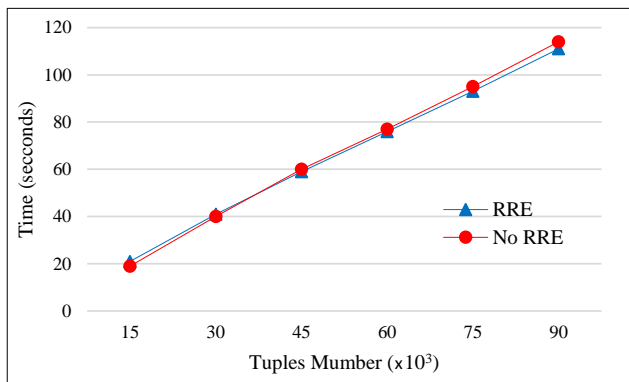


FIGURE 29. Time consumed by embedding process (RRE vs. no RRE).

Finally, depending on the parameters used to generate C_I , the process can be very costly. Figure 30 shows how the cost is even higher than performing the embedding when $\lambda_I = 1$ and $\lambda_{II} = 3$.

VI. RELATED WORK

Several works focus on the role of information hiding techniques in digital forensics [21]–[23]. Among them, some address alternatives for secret communication through means of steganography [22], [24], [25] whereas others apply watermark following different goals [26]–[29].

The main difference between steganography and watermarking is that the first one does not require robustness considering the hidden message is transported by the digital asset to be protected, while others ignore its presence [24],

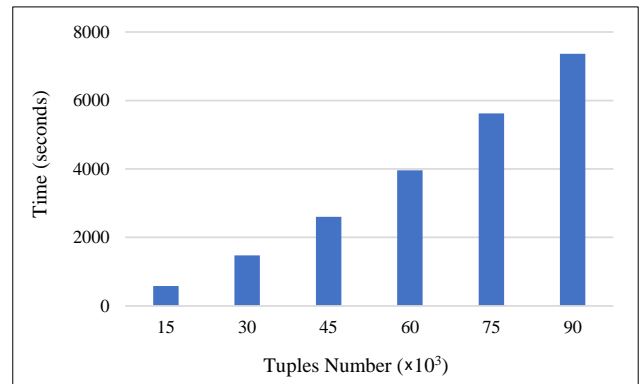


FIGURE 30. Time consumed for the generation of C_I considering the same number of tuples of Figure 29.

[30]. If secrecy is compromised, the hidden message can also be easily decoded. Contrary, watermarking techniques must implement the public system requirement following Kerckhoffs' principle [31], [32], which establishes that the cryptosystem security must rely on the secrecy of its parameter values (in particular, the cryptographic keys) and not in hiding its details. For this reason, watermarking techniques must guarantee robustness and security [33], [34], considering protected digital assets are expected to be attacked, and the watermark embedded into them has to resist the malicious operations trying to remove the marks or compromise the watermark detection [6], [35].

Watermarking techniques have been applied over different data types with cybersecurity and Internet development [36], [37]. The first approaches were proposed to protect multimedia data. Among techniques for multimedia data, some have been specialized on copyright protection and tamper detection on video [38]–[40], audio [41], [42] and images [43], [44]. Later some approaches oriented to protect documents [45], [46] or textual content stored in relational databases [1], [47] emerged. Other techniques have been created to protect source code, and software [48], [49]. Their diversity regarding data types and protection intents is wide.

We are interested in watermarking techniques developed for relational data protection. Differences between multimedia and relational data are such watermarking techniques created for multimedia data protection cannot be applied to relational data, especially when they implement sequential watermark embedding. Indeed, a watermarking technique developed for relational data protection that sequentially embeds the watermark might severely compromise its detection, especially in the case in which data are reordered as a consequence of both a subset reverse order attack or a benign update.

The first relational data watermarking approach was proposed in 2002 by [19]. The authors introduced the AHK algorithm in this work, defining how to analyze the values stored in a database's relation for watermark embedding. Many techniques have been proposed to protect relational data from that moment on and different classification criteria have been

defined to facilitate relational watermarking approaches study. For instance, relational watermarking techniques can be classified as distortion-based, and distortion-free [6]. Distortion-based watermarking approaches introduce small changes in the relation's content without affecting its usability. Among distortion-based techniques, there are schemes focused on returning the data to its original quality once the watermark is extracted. This subset of distortion-based technique is defined as reversible [50]–[52]. On the contrary, distortion-free techniques aim to preserve the integrity of the protected data [53], [54]. Usually, distortion-free techniques are defined as fragile, while distortion-based as robust approaches, to the extent that the embedded information survives at malicious or accidental attempts to remove it [1], [55]. Relational watermarking techniques can also be classified by their (i) cover-type, defining the type of data of the attribute in R selected to embed the marks; (ii) intent, i.e., ownership protection [1]–[3], [56], data tampering detection [4], [5], [57], traitor tracing [58]–[62], among others; (iii) watermark source, which can be meaningless such as a random binary stream [19], [31], [63], or meaningful, i.e., a source for watermark generation presenting a meaning that does not depend on the watermarking technique [17], [64], [65]. Regarding the cover-type, some techniques embed the watermark into attributes storing categorical values [66]–[69], information regarding date or time [70], textual [1], [47] or numerical content [17], [71], [72]. Note that the relational watermarking approach we propose in this work is distortion-based, oriented to protect the ownership of the data to which it is applied. As presented in the sections above, it exploits meaningful information as the source to generate the watermark. Moreover, we validated our approach embedding the watermark into numerical attributes; however, it is not limited to a particular cover-type. It can be used to improve techniques marking different types of attributes.

Concerning IBW techniques, to avoid consequences of sequentiality in watermarking approaches, it has been substituted by the pseudo-random selection of marks and embedding positions, as proposed in [1], [7]. A relational watermarking technique leveraged with pseudo-randomness has been proved to be robust against subset reverse order attacks, but it introduces a new issue to address. Indeed, the random nature of pixels selection leads to the multiple selection of certain meta-marks, ignoring the others available in the source. On the one hand, these techniques can contrast the effects of update-based actions (malicious or not) when majority voting is implemented during the watermark extraction process. However, they lead to the partial embedding of the watermark, exposing the protected data to leaks due to the watermark degradation. Our approach face this issue monitoring the random component integrated in the embedding process, as presented in Section III.

VII. CONCLUSION

Pseudo-random selection of meta-marks and embedding places in database relations is a perfect way to overcome

vulnerabilities of watermarking technique against subset reverse order attacks. While pseudo-random allows some meta-marks been used multiple times (recurrent selection) for generation and embedding of marks during the embedding process, this contributes to overcome minor update attacks if during the extraction process a majority voting is performed over each meta-mark candidate values. As well, chaotic nature of pseudo-random selection contributes to increase difficulty for attackers to find and delete or overwrite the marks.

Nevertheless, one important downside of recurrent selection is that, while some meta-marks are selected excessively, others are ignored, resulting in a partial use of the watermark source. Then, the attacks can be low-degree featured, guaranteeing watermark signal compromising while keeping data quality, since the embedded watermark signal is already weak. In this work, we proposed a recurrent meta-mark selection control engine (RRE) to limit the number of times a meta-mark is selected, increasing the opportunities for others to be considered, resulting in the increment of watermark capacity. The engine mixes the nature of probabilistic and chaotic frameworks, making hard for attackers to predict marks positions and increasing in a diverse way the positions of considered meta-marks.

Experimental results depict an important increment of the watermark capacity, and also clarify the role played by the number of meta-marks being considered according to the number of times the embedding is performed. Also, in terms of complexity, the experiments show an increment of the time required by applying RRE directly linked to the number of data being protected, with minor discrepancies with respect to techniques not considering RRE, which makes our proposal feasible for implementation.

As a future work, there is space for further optimizations related to the chaotic seeds generation without compromising the probabilistic features of the approach.

REFERENCES

- [1] M. L. Pérez Gort, M. Oliaro, A. Cortesi, and C. F. Uribe, "Semantic-driven watermarking of relational textual databases," *Expert Systems with Applications*, vol. 167, p. 114013, 2021.
- [2] R. Hou and H. Xian, "A graded reversible watermarking scheme for relational data," *Mobile Networks and Applications*, vol. 26, no. 4, pp. 1552–1563, 2021.
- [3] D. Hu, D. Zhao, and S. Zheng, "A new robust approach for reversible database watermarking with distortion control," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 6, pp. 1024–1037, 2018.
- [4] S. A. Shah, I. A. Khan, S. Z. H. Kazmi, and F. H. B. M. Nasaruddin, "Semi-fragile watermarking scheme for relational database tamper detection," *Malaysian Journal of Computer Science*, vol. 34, no. 1, pp. 1–12, 2021.
- [5] S. Siledar and S. Tamane, "A distortion-free watermarking approach for verifying integrity of relational databases," in 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC). IEEE, 2020, pp. 192–195.
- [6] R. Halder, S. Pal, and A. Cortesi, "Watermarking techniques for relational databases: Survey, classification and comparison," *J. Univers. Comput. Sci.*, vol. 16, no. 21, pp. 3164–3190, 2010.
- [7] H. M. Sardroudi and S. Ibrahim, "A new approach for relational database watermarking using image," in 5th International Conference on Computer Sciences and Convergence Information Technology. IEEE, 2010, pp. 606–610.
- [8] Z. Hu, Z. Cao, and J. Sun, "An image based algorithm for watermarking relational databases," in 2009 International Conference on Measuring

- Technology and Mechatronics Automation, vol. 1. IEEE, 2009, pp. 425–428.
- [9] S. Rani and R. Halder, “Comparative analysis of relational database watermarking techniques: An empirical study,” *IEEE Access*, 2022.
- [10] K. Stefan, P. Fabien AP *et al.*, “Information hiding techniques for steganography and digital watermarking,” 2000.
- [11] C. Date, *An Introduction to Database Systems*, 8th ed. USA: Addison-Wesley Longman Publishing Co., Inc., 2003.
- [12] M. L. Pérez 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, p. 112770, 2019.
- [13] —, “A double fragmentation approach for improving virtual primary key-based watermark synchronization,” *IEEE Access*, vol. 8, pp. 61 504–61 516, 2020.
- [14] C. Wang, J. Wang, M. Zhou, G. Chen, and D. Li, “Atbam: an arnold transform based method on watermarking relational data,” in *2008 International Conference on Multimedia and Ubiquitous Engineering (MUE 2008)*. IEEE, 2008, pp. 263–270.
- [15] Y. Wang, G.-M. Zhu, and S.-B. Zhang, “Research on the watermarking algorithm based on numerical attribute in the relational database,” in *2012 International Conference on Computer Science and Electronics Engineering*, vol. 2. IEEE, 2012, pp. 363–367.
- [16] U. P. Rao, D. R. Patel, and P. M. Vikani, “Relational database watermarking for ownership protection,” *Procedia Technology*, vol. 6, pp. 988–995, 2012.
- [17] M. L. Pérez Gort, C. Feregrino Uribe, and J. Nummenmaa, “A minimum distortion: High capacity watermarking technique for relational data,” in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, 2017, pp. 111–121.
- [18] Colorado-State-University, “Forest CoverType, The UCI KDD Archive,” Jun. 1999. [Online]. Available: <http://kdd.ics.uci.edu/databases/covertyp/covertime.html>
- [19] R. Agrawal and J. Kiernan, “Watermarking relational databases,” in *VLDB’02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 2002, pp. 155–166.
- [20] J. Sun, Z. Cao, and Z. Hu, “Multiple watermarking relational databases using image,” in *2008 International Conference on MultiMedia and Information Technology*. IEEE, 2008, pp. 373–376.
- [21] C. Wang, L. Yang, Y. Wu, Y. Wu, X. Cheng, Z. Li, and Z. Liu, “Data provenance with retention of reference relations,” *IEEE Access*, vol. 6, pp. 77 033–77 042, 2018.
- [22] M. Dalal and M. Juneja, “Steganography and steganalysis (in digital forensics): a cybersecurity guide,” *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 5723–5771, 2021.
- [23] L. Singh, A. K. Singh, and P. K. Singh, “Secure data hiding techniques: a survey,” *Multimedia Tools and Applications*, vol. 79, no. 23, pp. 15 901–15 921, 2020.
- [24] O. O. Evsutin, A. S. Melman, and R. V. Meshcheryakov, “Digital steganography and watermarking for digital images: a review of current research directions,” *IEEE Access*, 2020.
- [25] M. S. Hossen, M. A. Islam, T. Khatun, S. Hossain, and M. M. Rahman, “A new approach to hiding data in the images using steganography techniques based on aes and rc5 algorithm cryptosystem,” in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, 2020, pp. 676–681.
- [26] C.-C. Chang, T.-S. Nguyen, and C.-C. Lin, “A reversible database watermark scheme for textual and numerical datasets,” in *2021 IEEE/ACIS 22nd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE, 2021, pp. 208–212.
- [27] T. Ji, E. Ayday, E. Yilmaz, and P. Li, “Differentially-private fingerprinting of relational databases,” *arXiv preprint arXiv:2109.02768*, 2021.
- [28] P. Kadian, S. M. Arora, and N. Arora, “Robust digital watermarking techniques for copyright protection of digital data: A survey,” *Wireless Personal Communications*, pp. 1–25, 2021.
- [29] Y. Zhang, “A robust and adaptive watermarking technique for relational database,” *Cyber Security*, p. 3, 2022.
- [30] Q. Giboulot, R. Cogranne, and P. Bas, “Detectability-based jpeg steganography modeling the processing pipeline: the noise-content trade-off,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2202–2217, 2021.
- [31] R. Agrawal, P. J. Haas, and J. Kiernan, “Watermarking relational data: framework, algorithms and analysis,” *The VLDB journal*, vol. 12, no. 2, pp. 157–169, 2003.
- [32] F. Cayre, C. Fontaine, and T. Furon, “Watermarking security: theory and practice,” *IEEE Transactions on signal processing*, vol. 53, no. 10, pp. 3976–3987, 2005.
- [33] M. L. Pérez Gort, M. Olliaro, C. Feregrino-Uribe, and A. Cortesi, “Preventing attacks to relational database watermarking,” in *International Conference on Research and Practical Issues of Enterprise Information Systems*. Springer, 2019, pp. 131–140.
- [34] S. Xiang, G. Ruan, H. Li, and J. He, “Robust watermarking of databases in order-preserving encrypted domain,” *Frontiers of Computer Science*, vol. 16, no. 2, pp. 1–9, 2022.
- [35] N. Agarwal, A. K. Singh, and P. K. Singh, “Survey of robust and imperceptible watermarking,” *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8603–8633, 2019.
- [36] M. Ahmad, A. Shahid, M. Y. Qadri, K. Hussain, and N. N. Qadri, “Fingerprinting non-numeric datasets using row association and pattern generation,” in *2017 International Conference on Communication Technologies (ComTech)*. IEEE, 2017, pp. 149–155.
- [37] S. Kumar, B. K. Singh, and M. Yadav, “A recent survey on multimedia and database watermarking,” *Multimedia Tools and Applications*, vol. 79, no. 27, pp. 20 149–20 197, 2020.
- [38] A. A. Elrowayati, M. A. Alrshah, M. F. L. Abdullah, and R. Latip, “Hevc watermarking techniques for authentication and copyright applications: Challenges and opportunities,” *IEEE Access*, vol. 8, pp. 114 172–114 189, 2020.
- [39] C. Priya and C. Ramya, “Robust and secure video watermarking based on cellular automata and singular value decomposition for copyright protection,” *Circuits, Systems, and Signal Processing*, vol. 40, no. 5, pp. 2464–2493, 2021.
- [40] X. Yu, C. Wang, and X. Zhou, “A survey on robust video watermarking algorithms for copyright protection,” *Applied Sciences*, vol. 8, no. 10, p. 1891, 2018.
- [41] V. L. Narla, S. Gulivindala, S. R. Chanamallu, and D. Gangwar, “Bch encoded robust and blind audio watermarking with tamper detection using hash,” *Multimedia Tools and Applications*, vol. 80, no. 21, pp. 32 925–32 945, 2021.
- [42] L. Xu, D. Huang, S. F. A. Zaidi, A. Rauf, and R. K. Das, “Graph fourier transform based audio zero-watermarking,” *IEEE Signal Processing Letters*, vol. 28, pp. 1943–1947, 2021.
- [43] S. M. Darwish and L. D. S. Al-Khafaji, “Dual watermarking for color images: a new image copyright protection model based on the fusion of successive and segmented watermarking,” *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 6503–6530, 2020.
- [44] O. P. Singh, A. Singh, G. Srivastava, and N. Kumar, “Image watermarking using soft computing techniques: A comprehensive survey,” *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30 367–30 398, 2021.
- [45] A. W. Bitar, R. Darazi, J.-F. Couchot, and R. Couturier, “Blind digital watermarking in pdf documents using spread transform dither modulation,” *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 143–161, 2017.
- [46] A. K. Singh, S. Thakur, A. Jolfaei, G. Srivastava, M. Elhoseny, and A. Mohan, “Joint encryption and compression-based watermarking technique for security of digital documents,” *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 1, pp. 1–20, 2021.
- [47] A. Al-Haj and A. Odeh, “Robust and blind watermarking of relational database systems,” *Journal of Computer Science*, vol. 4, p. 1024–1029, 12 2008.
- [48] G. Gupta and J. Pieprzyk, “Source code watermarking based on function dependency oriented sequencing,” in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 2008, pp. 965–968.
- [49] M. Shirali-Shahreza and S. Shirali-Shahreza, “Software watermarking by equation reordering,” in *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*. IEEE, 2008, pp. 1–4.
- [50] M. Zhao, C. Jiang, and J. Duan, “Reversible database watermarking based on differential evolution algorithm,” in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. IEEE, 2019, pp. 120–124.
- [51] S. B. Sileedar and S. Tamane, “Quadratic difference expansion based reversible watermarking for relational database,” *Journal of Integrated Science and Technology*, vol. 9, no. 2, pp. 107–112, 2021.
- [52] S. Sileedar and D. S. Tamane, “Reversible database watermarking with distortion control,” *Indian J. Comput. Sci. Eng.*, vol. 12, no. 5, pp. 1503–1509, 2021.

[53] S. Bhattacharya and A. Cortesi, "Distortion-free authentication watermarking," in *International Conference on Software and Data Technologies*. Springer, 2010, pp. 205–219.

[54] S. Yan, S. Zheng, B. Ling, and D. Hu, "Lossless database watermarking based on order-preserving encryption," in *ACM Turing Award Celebration Conference-China (ACM TURC 2021)*, 2021, pp. 216–223.

[55] J. He, Q. Ying, Z. Qian, G. Feng, and X. Zhang, "Semi-structured data protection scheme based on robust watermarking," *EURASIP Journal on Image and Video Processing*, vol. 2020, no. 1, pp. 1–10, 2020.

[56] J. Franco-Contreras and G. Coatrieux, "Robust watermarking of relational databases with ontology-guided distortion control," *IEEE transactions on information forensics and security*, vol. 10, no. 9, pp. 1939–1952, 2015.

[57] S. Sun, Y. Xu, and Z. Wu, "Research on tampering detection of material gene data based on fragile watermarking," in *International Conference on Artificial Intelligence and Security*. Springer, 2020, pp. 219–231.

[58] J. Franco-Contreras and G. Coatrieux, "Robust watermarking of relational databases with ontology-guided distortion control," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 9, pp. 1939–1952, 2015.

[59] F. Guo, J. Wang, and D. Li, "Fingerprinting relational databases," in *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC)*, Dijon, France, April 23–27, 2006, H. Haddad, Ed. ACM, 2006, pp. 487–492.

[60] Y. Li, V. Swarup, and S. Jajodia, "Fingerprinting relational databases: Schemes and specialties," *IEEE Trans. Dependable Secur. Comput.*, vol. 2, no. 1, pp. 34–45, 2005.

[61] X. Shen, Y. Zhang, T. Wang, and Y. Sun, "Relational database watermarking for data tracing," in *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, 2020, pp. 224–231.

[62] E. Al Solami, M. Kamran, M. Saeed Alkathiri, F. Rafiq, and A. S. Alghamdi, "Fingerprinting of relational databases for stopping the data theft," *Electronics*, vol. 9, no. 7, p. 1093, 2020.

[63] G. Gupta and J. Pieprzyk, "Database relation watermarking resilient against secondary watermarking attacks," in *International Conference on Information Systems Security*. Springer, 2009, pp. 222–236.

[64] M. Huang, J. Cao, Z. Peng, and Y. Fang, "A new watermark mechanism for relational data," in *2004 International Conference on Computer and Information Technology (CIT 2004)*, 14–16 September 2004, Wuhan, China. IEEE Computer Society, 2004, pp. 946–950.

[65] D. Gross-Amblard, "Query-preserving watermarking of relational databases and xml documents," *ACM Trans. Database Syst.*, vol. 36, no. 1, pp. 3:1–3:24, 2011.

[66] E. Bertino, B. C. Ooi, Y. Yang, and R. H. Deng, "Privacy and ownership preserving of outsourced medical data," in *21st International Conference on Data Engineering (ICDE'05)*. IEEE, 2005, pp. 521–532.

[67] R. Sion, "Proving ownership over categorical data," in *Proceedings. 20th International Conference on Data Engineering*. IEEE, 2004, pp. 584–595.

[68] 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.

[69] C.-C. Lin, T.-S. Nguyen, and C.-C. Chang, "Lrw-crdb: Lossless robust watermarking scheme for categorical relational databases," *Symmetry*, vol. 13, no. 11, p. 2191, 2021.

[70] M. E. Farfoura, S.-J. Hornig, J.-L. Lai, R.-S. Run, R.-J. Chen, and M. K. Khan, "A blind reversible method for watermarking relational databases based on a time-stamping protocol," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3185–3196, 2012.

[71] H. Cui, "A watermarking algorithm for cloud database based on chaos cryptography," in *2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*. IEEE, 2020, pp. 586–589.

[72] Y. Xu and B. Shi, "Copyright protection method of big data based on nash equilibrium and constraint optimization," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1520–1530, 2021.



MAIKEL LÁZARO PÉREZ GORT is currently a research fellow in Università Ca' Foscari. He received a M.Sc. degree in applied informatics from Universidad Tecnológica de La Habana, Cuba, in 2010; and a Ph.D. degree in computer sciences from the National Institute of Astrophysics, Optics, and Electronics (INAOE) of Puebla, México, in 2020. From 2006 to 2013 he worked as a full-time professor at Universidad Tecnológica de La Habana. Also, from 2019 to 2021 was a part-time professor at Universidad Iberoamericana, Puebla, in Mexico. During his career has worked with different institutions regarding database management, usability, and security issues. In 2008 he was a visiting researcher at the Department of Computer Science in the University of Tampere, Finland; and in 2010 worked as a visiting professor at the University of Puerto Ordaz, Venezuela. His research interests are relational databases theory, information security and privacy, and data usability and authenticity.



MARTINA OLLIARO is currently a research fellow in Università Ca' Foscari. She received her Ph.D. in Computer Science at Ca' Foscari University of Venice (Italy) and Masaryk University of Brno (Czech Republic) under Professors Agostino Cortesi and Vashek Matyas. She started her double Ph.D. degree program in 2017, and she spent one year at the Faculty of Informatics, Masaryk University, where she successfully defended her Ph.D. thesis proposal. Her main research interest

concerns string static analysis through abstract interpretation theory, focusing on string-related security issues. She is also interested in watermarking relational databases techniques and studying relational data semantics preservation.



AGOSTINO CORTESI is a full professor of computer science at Ca' Foscari University, Venice, Italy. He has extensive experience in the area of static analysis and software verification techniques. In particular he contributed to the design and practical evaluation of abstract domains within the Abstract Interpretation framework. He coordinates the MAE Italy-India project 2017–2020 "Formal Specification for Secured Software System".

...