ROBUST AND REVERSIBLE NUMERICAL SET WATERMARKING

Gaurav Gupta, Josef Pieprzyk
Centre for Advanced Computing - Algorithms and Cryptography
Macquarie University, Australia

Mohan Kankanhalli
School of Computing, National University of Singapore

Keywords: Watermarking, Copyright protection.

Abstract: Numeric sets can be used to store and distribute important information such as currency exchange rates and stock forecasts. It is useful to watermark such data for proving ownership in case of illegal distribution by *someone*. This paper analyzes the numerical set watermarking model presented by Sion et. al in *"On watermarking numeric sets"*, identifies it's weaknesses, and proposes a novel scheme that overcomes these problems. One of the weaknesses of Sion's watermarking scheme is the requirement to have a normally-distributed set, which is not true for many numeric sets such as forecast figures. Experiments indicate that the scheme is also susceptible to subset addition and secondary watermarking attacks. The watermarking model we propose can be used for numeric sets with arbitrary distribution. Theoretical analysis and experimental results show that the scheme is strongly resilient against sorting, subset selection, subset addition, distortion, and secondary watermarking attacks.

1 INTRODUCTION

Piracy of digital objects such as audio, video and software is becoming a major concern for the owners of these documents. It is becoming easier for the pirates to obtain and distribute the data using peer-to-peer networks and file-sharing hosts and it is getting more difficult for the owners to prevent this illegal distribution, in which case it becomes important for the owner to be able to at least establish his ownership over the object if a person is found in possession of a digital object, believed to be pirated. This is accomplished be *watermarking*; the process of embedding information by introducing small changes in digital data. Successful retrieval of this information with a secret key establishes the ownership of the key-holder over the concerned object. An effective watermarking scheme should have the following features:

1. *Detectability*: The watermark should be detectable in order to establish ownership.
2. *Robustness*: The watermark should survive attacks such as cropping, data modification, and more.
3. *Low false positives*: There should be negligible possibility of *accidentally* detecting a watermark in an unmarked object.
4. *Blindness*: Only the watermarked object and a secret key should be needed to detect watermark.
5. *Imperceptibility*: The watermark should have minimal impact on the quality of data.

Several watermarking schemes have been proposed in the past for images (Cox et al., 1996; Bors and Pitas, 1996), software (Venkatesan et al., 2001; Collberg and Thomborson, 1999; Qu and Potkonjak, 1998), databases (Sion et al., 2004; Zhang et al., 2004) and text documents (Bolshakov, 2005; Atallah et al., 2001), but numerical sets have not been given the deserved attention, with only two major works known to the authors (Sion et al., 2002; Sebe et al., 2005). Numeric sets are extensively useful in several fields such as weather, military, scientific results and bio-informatics to name a few. For instance, management consulting firms such as McKinsey provide financial institutions with valuable data concerning currency and stock fluctuations. Their clients employ thousands of employees who have access to this in-

Gupta G., Pieprzyk J. and Kankanhalli M. (2009).
ROBUST AND REVERSIBLE NUMERICAL SET WATERMARKING.
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 141-148
DOI: 10.5220/0002186901410148

formation. The data provider would prefer that the information is not distributed without it's permission. But since this is not always possible, the company at least should have provable ownership over the data.

The numeric watermarking model proposed in (Sebe et al., 2005) focuses on preserving the statistical properties of a numerical set, including arithmetic mean and variance. The scheme has high false positive rates of up to 30.85%, thereby possibly incriminating innocent users and also lacks a substantial analysis of the model's security against an active adversary. In this paper, we present a watermarking scheme for numeric sets that satisfies the desirable features of a watermarking scheme mentioned above.

1.1 Organization of Paper

In Section 2, we discuss and analyze the numeric set watermarking scheme from *"On watermarking numeric sets"* (Sion et al., 2002), present experimental results demonstrating the weaknesses of the model and also discuss solutions to eliminate these drawbacks. In Section 3, we propose a fresh watermarking model and present the algorithms in Section 3.1. Section 4 includes a comprehensive analysis of the proposed model and the experimental results. The paper is concluded in Section 5 with a note on future research direction.

2 SION'S WATERMARKING SCHEME (SWS)

This section analyzes *SWS* that they later extend for database watermarking (Sion et al., 2004). The watermarking model is based on the statistical distributions of subsets. The subset partitioning is based on the most significant bits (MSBs) of set items. Items of a subset are then modified such that they satisfy some data usability conditions (*DUC*), such as maximum allowable mean square error. When a value s_i is changed to v_i during watermark insertion, the following two conditions must be satisfied:

$$(s_i - v_i)^2 < t_i \quad 1 \leq i \leq n \tag{1}$$

$$\sum((s_i - v_i)^2) < t_{max} \tag{2}$$

where $\mathbb{T} = \{t_i | 1 \leq i \leq n, t_i \in \mathbb{R}\}$ is the set of individual bounds and $t_{max} \in \mathbb{R}$ is the overall bound.

The authors address subset selection, addition, alteration and re-ordering attacks. However, secondary watermarking or additive attacks, is left out. The detailed watermarking process is as follows:

1. Numeric set to watermark is $\mathbb{S} = \{s_i | 1 \leq i \leq n, s_i \in \mathbb{R}\}$, k' is the private key to generate item indices, $MSB(f, s_i)$ are the f MSBs of s_i, and *NORM* a normalization operation.
2. Items are ordered using hashing performed on the secret key k' and the f MSBs of the items' normalized value using Equation 3, thereby imposing a secret key-based order on the items.

$$index(s_i) = H(k', MSB(f, NORM(s_i)), k') \tag{3}$$

3. Create subsets S_i of equal sizes where,

$$S_i = \{x_{i_j} | x_{i_j} = s_{i \times y + j}, 0 \leq i < \frac{n}{y}, 1 \leq j \leq y\} \tag{4}$$

   Each subset contains y adjacent items from the list of items sorted by indices (calculated using Equation 3).
4. The maximum of n/y bits can be embedded in each subset. Each bit of a b-bit watermark can be embedded up to $n/(y \times b)$ times. This provides error-correction capabilities.
5. Let $avg(S_i) = \sum_{j=0}^{y}(x_{i_j})$ and $\delta(S_i) = \sqrt{\frac{\sum(avg(S_i) - x_{i_j})^2}{y}}$ be the average and standard deviation of the items of S_i, respectively. Let $v_{true}, v_{false}, c \in (0,1)$ be real numbers. The value c is a confidence factor while (v_{true}, v_{false}) are confidence violators such that $v_{true} > v_{false}$. As an example, let $c = 0.9, v_{true} = 0.1, v_{false} = 0.07$. $v_c(S_i)$ is the number of items greater than $avg(S_i) + c \times \delta(S_i)$. Bit encoded in a subset S_i is '1' if $v_c(S_i) > v_{true} \times y$, '0' if $v_c(S_i) < v_{false} \times y$ and otherwise invalid.

Algorithm *Please place* `\label` ***after*** `\caption` embeds a single watermark bit b in a subset S_i. If it returns *success*, we insert the next bit, otherwise we insert the same bit, in the next subset. Detection algorithm works symmetrically, identifying watermarked bit in subsets created from Equations 3, 4.

The watermarking scheme is presented to be resilient against several attacks such as re-sorting (obviously, the actual sorting that the watermarking algorithm uses is based on hash of the secret key and the items' MSBs hence it is evident that re-sorting attacks do not alter the watermarking detection results), and subset selection (up to 50% data cuts). Although, subset addition attack is not discussed by the authors. The attacker inserts multiple instances of the same item in the set to distort the subsets used for watermark detection. On an average, $\frac{n}{2 \times y}$ subsets are distorted. The

watermark detection is affected based on the properties of elements that jump from one subset to another. The effectiveness of this attack needs to be measured experimentally, however, in Section 2.1, we provide a theoretical estimate of this *SWS*'s resilience against subset addition attack.

Input : Bit b, Subset S_i
Output: bit embedded status
return *success* if $((b = 1$ and $v_c(S_i) > v_{true} \times y)$ or $(b = 0$ and $v_c(S_i) < v_{false} \times y))$;
if $b = 1$ **then**
  while *true* **do**
    Select $it_1, it_2 \in S_i \leq avg(S_i) + c \times \delta(S_i)$;
    if it_1, it_2 *found* **then**
      while $it_1 \leq avg(S_i) + c \times \delta(S_i)$ **do**
        $it_1 = it_1 + incrementValue$;
        $it_2 = it_2 - incrementValue$;
        return $failure$ if *DUC* violated;
      end
      return *success* if $v_c(S_i) > v_{true} \times y$;
    end
  end
else
  while *true* **do**
    Select $it_1, it_2 \in S_i > avg(S_i) + c \times \delta(S_i)$;
    if it_1, it_2 *found* **then**
      while $it_1 > avg(S_i) + c \times \delta(S_i)$ **do**
        $it_1 = it_1 - incrementValue$;
        $it_2 = it_2 + incrementValue$;
        return $failure$ if *DUC* violated;
      end
      return *success* if $v_c(S_i) < v_{false} \times y$;
    end
  end
end
return $failure$;

Algorithm 1: Single watermark bit insertion.

2.1 Drawbacks of *SWS*

From our discussion above, we have identified the following drawbacks of *SWS*:

1. We need to preserve each subset's average during watermark insertion. If the watermark bit is 1, then we choose two items, $it_1, it_2 < avg(S_i) + c \times \delta(S_i)$ and increase it_1 while decreasing it_2 until $it_1 \geq avg(S_i) + c \times \delta(S_i)$. The condition increases the standard deviation and the value of $avg(S_i) + c \times \delta(S_i)$ is different during watermark detection. This value should be remain the same during insertion and detection. Hence, instead of using $avg(S_i) + c \times \delta(S_i)$ as a bound, $c \times avg(S_i)$ should be used.
2. The scheme is applicable to numeric set that follow a *normal distribution*; a theoretical bell-shaped data distribution that is symmetrical around the mean and has a majority of items concentrated around the mean. This is not practical in real life since a lot of candidate numeric sets watermarking might not be normally distributed. Secondly, even if we assume that the set is normally distributed, the chances of each subset following a normal distribution are even lower. Thus, a watermarking scheme should be independent of the data distribution.
3. The sorting mechanism assumes that small changes to the items do not alter the subset categorization, which is based on MSBs. However, small modifications can change an item's MSBs when the item lies in the neighborhood of 2^x (let the set containing such items be $\mathcal{N}$) for $x \in \mathbb{Z}$. For example, subtracting two from 513 $(1000000001)_2$ would change it to 511 $(0111111111)_2$, thereby modifying the MSBs. The attacker can hence, select these items and add a small value to the items in the left $\mathcal{N}$ so that they jump to the right neighborhood and vice-versa. *SWS* does not address this constraint and possible solutions.
4. The watermarking scheme actually relies on the enormity of available bandwidth with majority voting being used to determine the correct watermark bit. For an m-bit watermark that is embedded l times, the data set needs to have $m \times l \times y$ items. As an illustrative figure, for a 32-bit watermark to be embedded just five times in subsets containing 20 items, we need to have 3,200 items in the set.
5. *Vulnerable to addition attacks:* Assume that the adversary adds $\bar{n}$ instances of the same item to the original set of n items. The number of items in the new set is $n' = n + \bar{n}$. The added items are adjacent to each other in the sorted set, which is divided into y subsets, each containing n'/y items. The starting index of the added items can be $1, \ldots, n+1$ with equal probabilities . Let the probability of detecting the watermark correctly be $P(\mathcal{A}, i)$, where the starting index of the added items is i in the sorted set. Therefore, the overall detection probability is $= \frac{1}{n+1} \sum_{i=1}^{n+1} P(\mathcal{A}, i)$. From Figure 1, the modified subsets are divided into three categories:
   (a) $\mathcal{G}_1$: Subsets containing items with index lower than that of added items and not containing any added items.
   (b) $\mathcal{G}_2$: Subsets containing added items.

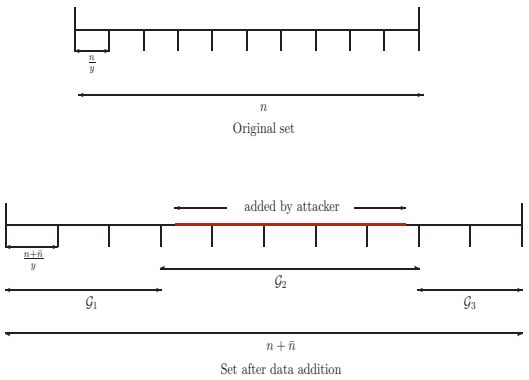

Figure 1: Subset generation after data addition attack (multiple instances of the same item added - their location in the sorted set represented in red line).

(c) $\mathcal{G}_3$: Subsets containing items with index higher than that of added items and not containing any added items.

Each modified subset $S'_i \in \mathcal{G}_1$ contains $\sigma_i = \frac{n}{y} - i \times \frac{\bar{n}}{y}$ items of original subset S_i and $\zeta_i = i \times \frac{\bar{n}}{y}$ items of the next original subset S_{i+1}. At some point of time, either the added items are encountered, or, σ_i becomes 0 (since $gcd(n,\bar{n}) > 1$). In the second condition, modified subset S'_i will contain $\frac{n}{y} - i \times \frac{\bar{n}}{y}$ items of next original subset S_{i+1} and ζ_i items of S_{i+2} (σ_i is 0 in this case, since the subset does not contain any of the original items). Thus, the probability of the correct watermark bit w_i being detected in subsets in $\mathcal{G}_1$ is $\mathcal{F}(\sigma_i, \frac{n'}{y})$ where $\mathcal{F}(a,b)$ is the probability of the correct watermark bit being detected in a subset with a of the original b items remaining. The probability of all $|\mathcal{G}_1|$ watermark bits being detected correctly is given as follows:

$$P(d_1) = \prod_{i=1}^{|\mathcal{G}_1|} \mathcal{F}(\sigma_i, \frac{n'}{y}) \quad (5)$$

The second group $\mathcal{G}_2$ can be further divided into two categories:

(a) $\mathcal{G}_{2_1}$: Subsets containing both original and new items from the same subset (the only possibility of this is with the first subset in $\mathcal{G}_2$).

(b) $\mathcal{G}_{2_2}$: Subsets containing none of the original items.

Watermark detection probability in $\mathcal{G}_{2_1}$ is $\mathcal{F}(\sigma_{(|\mathcal{G}_1|+1)_1}, \frac{n'}{y})$, and in $\mathcal{G}_{2_2}$ is $\mathcal{F}(0, \frac{n'}{y})$, achieving an overall watermark detection probability given below.

$$P(d_2) = \mathcal{F}(\sigma_{(|\mathcal{G}_1|+1)_1}, \frac{n'}{y}) \times \prod_{i=1}^{|\mathcal{G}_2-1|} \mathcal{F}(0, \frac{n'}{y}) \quad (6)$$

None of the subsets have the original items in $\mathcal{G}_3$ and therefore the probability of detecting the watermark correctly equals:

$$P(detect_3) = \prod_{i=1}^{|\mathcal{G}_3|} \mathcal{F}(0, \frac{n'}{y}) \quad (7)$$

The overall probability of detecting the watermark in the new set, $P(detected)$, is, $P(detect_1) \times P(detect_2) \times P(detect_3)$. $\mathcal{F}(0,-)$ is negligible since the subset contains none of the original items. It can be see that $P(detected)$ depends on the starting index of the added items in the modified set; if the added items are towards the front of the index-based sorted set, then the watermark is more likely to be erased.

3 PROPOSED SCHEME

We propose a watermarking scheme that inserts a single watermark bit in each of the items selected from a numeric set. During detection, we check if an item carries a watermark bit and verify whether the bit extracted from the watermarked item matches the expected watermark bit. If the proportion of items for which the extracted bit matches the watermark bit, to the total number of item carrying a watermark bit, is above a certain threshold, the watermark is successfully detected.

During the insertion algorithm, the watermark should ideally be spread evenly across the set and should be sparse enough so that the watermark can survive active attacks. We distribute the watermark evenly across the set by selecting the items based on their MSBs. It is possible to make it sparse enough by embedding a watermark bit in one of every γ items. This can be done by checking if γ divides λ, where λ is a one way hash on a concatenation of $MSB(f, s_i)$ and a secret key $\mathcal{K}$, shown as follows:

$$\lambda = \mathcal{H}(MSB(f, s_i) \| \mathcal{K}) \quad (8)$$

We assume that we have ξ LSBs that can be modified without substantially reducing the data's utility (value of ξ can be adjusted by the owner). The maximum distortion to the data without compromising it's quality is 2^{ξ}.

The watermark bit is $\lambda \pmod 2$. The owner marks v out of n items. If the detection algorithm identifies v' items to be watermarked, out of u' items carry the correct bit, then watermark presence is established if $\frac{u'}{v'} > \alpha$. Higher values of confidence level

α imply lower false positive probability but lower resilience against attacks. The value α should be set to an optimal value, usually between 0.6 and 0.8.

Finally, the bit location to be replaced by the watermark bit is identified. We input the maximum percentage change that can be introduced in an item, ε, and generate $\xi = \lceil log_2(s_i \times \varepsilon) \rceil$. We insert the replaced bit into the fractional part to enable reversibility. We can choose the location at which this bit is inserted in the fraction part as $\tau = \lambda \pmod{\beta}$, where β is the number of bits used to store the fraction part.

As discussed in Section 2.1, even a small distortion during insertion or by the attacker can result in modifying MSBs if the item lies in $\mathcal{N}$ and therefore affect the detection process.

Let there be one of out of ε items from S in $\mathcal{N}$. Upon inserting a bit in an item from $\mathcal{N}$, the watermarked item is ignored during detection with a probability of $(\gamma-1)/\gamma$, which simply reduces the number of items in which the watermark bits are detected. There is a $1/\gamma$ probability that the modified item is still detected as carrying a watermark bit (Algorithm 19, line 5). When this happens, there is a 50% probability that the bit detected is, in fact, the correct watermark bit (Algorithm 19, line 11). Thus the overall probability that the watermark bit being detected incorrectly is $1/(2\varepsilon\gamma)$. In normal circumstances, this is less than 1% since usually $\varepsilon < 10$ and $\gamma \approx 10$.

In stricter conditions where even a small proportion of watermark bits getting affected is unacceptable, a solution is to ensure that $abs(s_i - 2^x) > 2^\xi$, where $abs(x)$ is a function that returns the absolute value of a number $x \in \mathbb{R}$. Thus, an item s_i is chosen for carrying a watermark bit if $\lambda \pmod{\gamma} = 0$ *AND* $abs(s_i - 2^x) > 2^\xi$. From a security perspective, the attacker can ignore n/ε items that are in $\mathcal{N}$ while trying to remove the watermark, but apart from that, (s)he does not get any benefit.

3.1 Watermarking Algorithms

The insertion and detection processes are provided in Algorithms *Please place* `\label` ***after*** `\caption`, 19 respectively. In these algorithms $lsb(x,y)$ refers to y^{th} LSB of value x.

Input : Numeric set $\mathbb{S} = \{s_1, \ldots, s_n\}$, change limit ε, bits used for fraction part β, Secret key $\mathcal{K}$, Watermarking fraction γ
Output: Watermarked set $\mathbb{S}_w$
$\lambda = \mathcal{H}(MSB(f, s_i) \| \mathcal{K})$;
$\tau = \lambda \pmod{\beta}$;
for *$i = 1$ to n in steps of 1* **do**
  $\xi = \lceil log_2(s_i \times \varepsilon) \rceil$;
  if $\lambda \pmod{\gamma} = 0$ **then**
    //2^x is the power of 2 closest to s_i;
    if $abs(s_i - 2^x) > 2^\xi$ **then**
      $int = \lfloor s_i \rfloor$;
      $frac = s_i - int$;
      $b = \lambda \pmod{\xi}$;
      $lsb(frac, \tau) = lsb(int, b)$;
      $lsb(int, b) = \lambda \pmod 2$;
    end
  end
end

Algorithm 2: Watermark insertion.

Input : Watermarked set $\mathbb{S}_w$, change limit ε, bits used for fraction part β, Secret key $\mathcal{K}$, Watermarking fraction γ, confidence level α
Output: Watermark presence status, Original set $\mathbb{S} = \{s_1, \ldots, s_n\}$
1 $\lambda = \mathcal{H}(MSB(f, s_i) \| \mathcal{K})$;
2 $\tau = \lambda \pmod{\beta}$;
3 **for** *$i = 1$ to n in steps of 1* **do**
4 $\xi = \lceil log_2(s_i \times \varepsilon) \rceil$;
5 **if** $\lambda \pmod{\gamma} = 0$ **then**
6 //2^x is the power of 2 closest to s_i;
7 **if** $abs(s_i - 2^x) > 2^\xi$ **then**
8 $int = \lfloor s_i \rfloor$;
9 $frac = s_i - int$;
10 $b = \lambda \pmod{\xi}$;
11 **if** $lsb(int, b) = \lambda \pmod 2$ **then**
12 $match = match + 1$;
13 $lsb(int, b) = lsb(frac, \tau)$;
14 **end**
15 $total = total + 1$;
16 **end**
17 **end**
18 **end**
19 return *true* if $lsb(int, b) = \lambda \pmod 2$, otherwise *false*;

Algorithm 3: Watermark detection.

4 ANALYSIS AND EXPERIMENTAL RESULTS

4.1 False Positive Probability

First we discuss the false positive probability of our watermarking scheme. That is, what are the chances of a watermark detection algorithm detecting a watermark in an unmarked set S with parameters secret key $\mathcal{K}$, fraction γ and confidence level α. The number of items in a random set identified as containing watermark bit are $n' = \frac{n}{\gamma}$ and probability that the watermark bit will be detected correctly for an item is 1/2. Hence, at least α proportion of watermark bits identified correctly is given in Equation 9. This false positive probability is extremely and has shown to be around 10^{-10} in (Agrawal and Kiernan, 2002).

$$\sum_{i=\alpha\times n/\gamma}^{n/\gamma} \binom{n/\gamma}{i} (1/2)^i \times (1/2)^{n/\gamma - i}$$
$$= \sum_{i=\alpha\times n/\gamma}^{n/\gamma} \binom{n/\gamma}{i} (1/2)^{n/\gamma}$$
$$= 2^{-n/\gamma} \times \sum_{i=\alpha\times n/\gamma}^{n/\gamma} \binom{n/\gamma}{i} \quad (9)$$

4.2 Security

The attacks and our scheme's resilience to them is provided next:

1) Set Re-ordering. The re-ordering attack is ineffective against the watermarking model since each item is individually watermarked and checked for watermark bit presence.

2) Subset Addition. Let the attacker add subset S_1 containing n_{add} items to the watermarked set S_2 containing n items. $\frac{n}{\gamma}$ out of $\frac{n}{\gamma}$ watermark bits will still be detected correctly in S_2. From S_1, a total of $\frac{n_{add}}{\gamma}$ will probabilistically be detected as marked and for each item considered to be marked, watermark bit will be detected correctly with a a 0.5 probability. Thus, the expected number of correctly detected bits from S_2 is $\frac{n_{add}}{2\times\gamma}$. The overall watermark detection ratio is $\frac{1+n_{add}/(2n)}{1+n_{add}/n}$. For 50% ($\frac{n_{add}}{n} = \frac{1}{2}$), and 100% ($\frac{n_{add}}{n} = 1$) data additions, the expected watermark detection ratio is $\frac{5}{6}$ and $\frac{3}{4}$ respectively. For $\alpha = 0.7$. The adversary needs to add at least 150 items for every 100 items in the watermarked set to have a decent chance of destroying the watermark. For $\alpha = 0.6$, the number of items that need to be added to destroy the watermark increases to 400 items for every 100 items. Such levels of data addition are bound to have derogatory effect on data usability. Figure 1 illustrates the variation of watermark detection ratio with increasing levels of data addition.

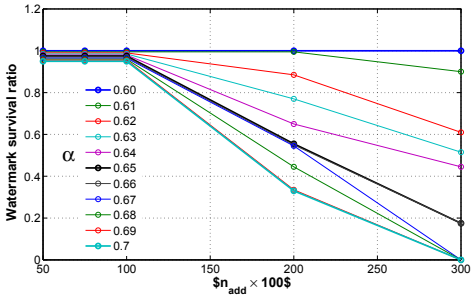

Figure 2: Watermark survival with varying α, n_{add}.

Experimental results from data addition attacks are given in Figure 2. The findings confirm our claim with all watermarked sets surviving attacks of up to 300% data addition for $\alpha = 0.6$, and 95% of the watermarked sets surviving attacks of up to 100% data addition for $\alpha = 0.7$.

3) Subset Deletion. We assume that the attacker deletes n_{remove} items from the watermarked set containing n items, leaving $n - n_{remove}$ items. The removed items have equal probability of being watermarked as the remaining items. Thus, the watermark detection ratio is $\frac{(n-n_{remove})/\gamma}{(n-n_{remove})/\gamma} = 1$. But this does not mean that the watermarking scheme is unconditionally secure against subtractive attacks. If the number of remaining elements is extremely low, the false positive probability becomes unacceptably high and the adversary can claim that the watermark detection was *accidental*. However, it is only in the interest of the adversary to leave sufficient items so that the set is still useful.

4 a) LSB Distortion. We assume that the attacker has the knowledge of ξ for this discussion.This is to provide additional strength to the attack and thereby provide the worst case security analysis of the watermarking model. The attacker chooses n_d items out of the total n items and flips all ξ bits in an attempt to erase the watermark. The watermark detection will detect the watermark bits incorrectly from the n_d items and, correctly from the other $n - n_d$ items. The watermark detection ratio in this case is $1 - \frac{n_d}{n}$. This ratio needs to be at least α to detect the watermark. Hence, the upper limit on items that can be deleted from the subset is $n_d \leq n \times (1 - \alpha)$. For $\alpha = 0.7$, a maximum of 30% items can be removed such that the watermark is preserved.

Experimental results of LSB distortion attack are

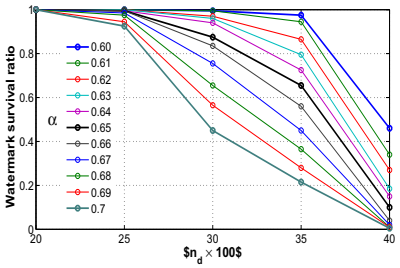

Figure 3: Watermark with varying α, n_d.

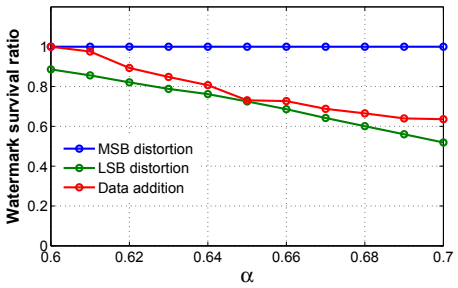

Figure 4: Watermark survival with varying α.

provided in Figure 3. The experiment was run on 200 numerical sets and computed the proportion of the times watermark survived when all ξ LSBs of 20% to 40% data items were flipped. The results show that the watermarking scheme is extremely secure against LSB bit flipping attacks for LSBs of 25% items being flipped. For 35% attack, the watermark survived an average of 62% times. For $\alpha = 0.60$, the watermark survival rate drops to 46% times when attack level increases to 40%. We infer from experimental results that the optimal value of α is around 0.65, with which watermark has a high survival possibility and at the same time has a low false positive probability.

4 b) MSB Distortion. We assume that the attacker has the knowledge of f for this discussion. Again, this makes the adversary stronger and provides us with an estimate of the watermark's resilience against acute attacks. The attacker chooses n_d items out of the total n items and flips all f MSBs, resulting in modified λ. The watermark detection will detect the watermark bits correctly from the other $n - n_d$ items. For the items with distorted MSBs, there are two cases:

1. With a probability of $\frac{\gamma-1}{\gamma}$, $\lambda \pmod{\gamma} \neq 0$ and the item is not considered as carrying a watermark bit.

2. With a probability of $\frac{1}{\gamma}$, $\lambda \pmod{\gamma} = 0$ and the item is still considered as carrying a watermark bit. There is a probability of $1/2$ that $(\lambda \pmod{\xi})^{th}$ LSB equals $\lambda \pmod 2$.

The following is an analysis of the expected value of watermark detection ratio. Within the distorted subset, the expected number of items considered as carrying a watermark bit is $\frac{n_d-\gamma+1}{\gamma}$ and the expected number of items in which watermark bit is detected correctly is $\frac{n_d-\gamma+1/2}{\gamma}$. Expected value of watermark detection ratio in the final set is $\frac{n-\gamma+\frac{1}{2}}{n-\gamma+1}$.

We can see that, on an average, for sufficiently large $n-\gamma$, the expected watermark detection ratio after MSB modification attack is very close 1. During our experiments, the watermarks were detected at all times with all f MSBs of 20% to 40% items being flipped.

The average watermark survival proportion under the three significant attacks of LSB distortion, MSB distortion, and data addition are presented in Figure 4. It can be seen from the figure that $\alpha = 0.65$ is the optimal value, where the watermark has a high chance of survival while having a low false positive probability.

5) Secondary Watermarking. The security of the watermarking scheme against secondary watermarking attacks comes from the reversibility operation (storing the original bit replaced by the watermark bit in the fraction part). If r parties, $O_1, \ldots, O_r$ watermark the same numeric set sequentially, then the objective is for the first party O_1 to be established as original and rightful owner. It has been shown in (Gupta and Pieprzyk,) that owner identification is facilitated by watermarking schemes that provide reversibility. Based on the experimental results, the current watermarking scheme provides security against secondary watermarking attacks with $r \leq 5$.

The watermark carrying capacity of the watermarking scheme is $|\{s_i : (abs(s_i - 2^x) > 2^{\xi})\}|/\gamma$, where 2^x is the power to 2 closest to s_i. This is much higher than the capacity of $\frac{|S|}{|S_i| \times m}$ offered by *SWS*, where $|S|$ is the size of the numeric set, $|S_i|$ is the size of the subsets and m is the number of times each watermark bit must be inserted. We designed experiments to test the watermarking capacities of both schemes with the sets ranging from 1000 to 3000 items, each watermark bit being embedded 1 to 5 times in subsets containing 25 to 200 items for SWS. Our scheme had an average watermarking capacity of 8.28% for the 60 experiments while the overall average of SWS was 0.86%. The summary of results is presented in Figure 5.

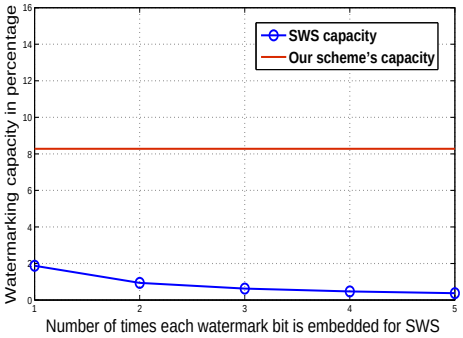

Figure 5: Comparison of our scheme's watermarking capacity with SWS.

5 CONCLUSIONS AND FUTURE WORK

We have proposed a watermarking model for numeric sets in this paper. The watermarking scheme embeds one watermark bit for every γ items in the numeric set of size n, thereby offering a watermark carrying capacity close to n/γ. The major improvement offered by our scheme is in terms of the lack of constraints on the characteristics of the numeric set to be watermarked. Unlike (Sion et al., 2002), where the numeric set to be watermarked should follow normal distribution, the watermarking scheme is applicable to a numeric set irrespective of it's distribution and it is shown that the watermark survives against data addition, deletion, distortion, re-sorting attacks as well as secondary watermarking attacks. The capacity of the watermarking scheme is also higher than that of the previous scheme.

The current scheme embeds a *detectable* watermark in the numeric set and not an *extractable* watermark. Our future work is directed towards finding ways to embed an extractable watermark in the numeric set whilst providing the same level of security and capacity offered by our current scheme.

REFERENCES

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

Atallah, M., Raskin, V., Crogan, M., Hempelmann, C., Kerschbaum, F., Mohamed, D., and Naik, S. (2001). Natural language watermarking: design, analysis, and a proof-of-concept implementation. In *Proceedings of 4th Information Hiding Workshop, LNCS*, pages 185–199, Pittsburgh, Pennsylvania. Springer-Verlag, Heidelberg.

Bolshakov, I. A. (2005). A method of linguistic steganography based on collocationally-verified synonymy. In *In Proceedings of 4th International Workshop on Information Hiding, IH 2004*, volume 3200 of *LNCS*, pages 180–191. Springer Verlag.

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

Collberg, C. and Thomborson, C. (1999). Software watermarking: Models and dynamic embeddings. In *Proceedings of Principles of Programming Languages 1999, POPL'99*, pages 311–324.

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

Gupta, G. and Pieprzyk, J. Reversible and blind database watermarking using difference expansion. *International Journal of Digital Crime and Forensics*, 1(2):42.

Qu, G. and Potkonjak, M. (1998). Analysis of watermarking techniques for graph coloring problem. In *Proceedings of International Conference on Computer Aided Design*, pages 190–193.

Sebe, F., Domingo-Ferrer, J., and Solanas, A. (2005). Noise-robust watermarking for numerical datasets. *Lecture Notes in Computer Science*, 3558:134–143.

Sion, R., Atallah, M., and Prabhakar, S. (2002). On watermarking numeric sets. In *Proceedings of First International Workshop on Digital Watermarking, IWDW 2002. LNCS*, volume 2163, pages 130–146, Seoul, Korea. Springer-Verlag, Heidelberg.

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

Venkatesan, R., Vazirani, V., and Sinha, S. (2001). A graph theoretic approach to software watermarking. In *Proceedings of 4th Information Hiding Workshop, LNCS*, volume 2137, pages 157–168.

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