DEFENDING AGAINST ADDITIVE ATTACKS WITH MAXIMAL ERRORS IN WATERMARKING RELATIONAL DATABASES

Yingjiu Li, Vipin Swarup and Sushil Jajodia

Abstract Recently, several database watermarking techniques have been developed to fight against database piracy. In watermarking, a database owner's identification information is embedded into a database such that proof of ownership can be established by detecting the information in pirated data. However, most watermarking systems are vulnerable to the severe threat of additive attacks and this threat has not been studied formally. In an additive attack, a pirate inserts an additional watermark such that the proof of ownership becomes ambiguous. In this paper, we present an effective approach to defending against additive attacks. Our strategy is to raise the errors introduced during watermark insertion to a predetermined threshold such that any additive attack would introduce more errors than the threshold. Exceeding the error threshold means that the pirated data is less useful or less competitive; thus, the owner does not need to claim ownership for such pirated data.

1. INTRODUCTION

With database systems extensively deployed nowadays, database piracy has become a severe concern in database applications. Compared to the cost of building and maintaining a database, it is much easier for a pirate to copy and redistribute it without its owner's knowledge or permission. Parallel to the recent efforts made in the passage of database protection laws [8], watermarking techniques are being developed to fight against database piracy [2, 5, 10–1, 9, 6, 7]. In these techniques, a database owner embeds a unique information, called watermark, into his data. The owner can detect the embedded information from pirated data so as to claim his ownership of data.

People have identified additive attacks as a severe threat to many watermarking schemes. In an additive attack, a pirate inserts an additional watermark before distributing a pirated database. The pirate may insert a watermark to claim ownership of the database or he may insert a watermark to claim that the database was provided to a buyer legitimately.

Additive attacks are easy to launch but hard to defend against. Though this problem has been identified for some time [2–1, 10, 9], a formal study has not been conducted. Some initial defending strategies [1] include:

- Assume that both the pirate's watermark and the owner's watermark can be detected from a pirated relation. To resolve the conflict, the owner may prove to a trusted third party that she has a data copy such that the pirate's watermark cannot be detected from that copy, while the pirate does not possess a copy without the owner's watermark.

- Involve a trusted third party in a secure append-only registry. When the owner publishes her data, she appends her key to the registry. To resolve disputes due to additive attacks, the owner may prove that her key was appended to the registry before the pirate's key and that her watermark was indeed present in the pirated data.

Both methods involve a trusted third party to resolve the dispute incurred by an additive attack. Involving a trusted third party is not always realistic and may be very expensive. This paper presents an alternative solution that makes dispute resolution less necessary.

Our approach is based on the observation that an additive attack inevitably introduces additional errors to the underlying data while rendering the original watermark less robust. Our strategy is to raise watermarking errors to a predetermined threshold such that any additive attack would introduce more errors than the threshold. The threshold represents the threshold of database usefulness. If the error exceeds the threshold significantly, the underlying database is considered to be less useful or less competitive; thus, the owner does not need to claim her ownership for such pirated data copies.

2. PRELIMINARIES

We start from a typical watermarking scheme that embeds watermark in numerical values. The scheme was proposed by Agrawal and Kiernan [2] for watermarking database relations. In the following, we briefly summarize this scheme based on the analysis given in [2, 1].

Given a database relation R which has ν numerical attributes $A_1, \ldots, A_\nu$ and η tuples, a watermark bit is embedded to a selected attribute value in each selected tuple. Tuple selection is determined by an integer parameter γ such that each tuple is selected with probability $1/\gamma$; on average $\omega \simeq \eta/\gamma$ tuples are selected for watermarking. For each selected attribute value, one of its ξ least significant bit is set to be a computed mark bit. It is assumed that database usage can tolerate the alteration caused by changing one of ξ least significant bits; however, the value of data will be severely degraded if all least significant bits are randomized. We call this *error tolerance assumption.*

The watermark insertion can be described as follows. For each tuple $r \in R$, a cryptographic pseudo-random sequence generator $\mathcal{S}$ is seeded with the tuple's primary key $r.P$ and the secret key $\mathcal{K}$ of the owner of R. Here is assumed that R has a primary key attribute P (besides the ν numerical attributes) and the owner has a secret key. Let $\mathcal{S}_i(\mathcal{K}, r.P)$ be the i-th number in the sequence generated by $\mathcal{S}$. Tuple r is selected if $\mathcal{S}_1 \mod \gamma = 0$. For each selected tuple, attribute i is selected if $i = \mathcal{S}_2 \mod \nu$. Then, for each selected attribute value, least significant bit j is selected if $j = \mathcal{S}_3 \mod \xi$. Finally, the selected bit is set to be a computed mark bit which is zero if $\mathcal{S}_4$ is even and one otherwise.

In watermark detection, the same procedure and parameters are used to locate all mark bits. A located mark bit should be the same as the one computed from $\mathcal{S}_4$ if the data has not changed. In this case there are total ω matches. However, because of possible updates or attacks, one may not be able to detect all the matches. The ownership of the detected data is claimed if the fraction of the matches is than $\tau \in [0.5, 1)$. Note that in Agrawal and Kiernan's original paper [2], τ is defined to be the minimum number of matches (absolute measure) rather than a fraction (relative measure).

Errors. Watermark insertion introduces errors by altering the underlying data. In [1], the errors are analyzed in computing mean and variance for an integer-valued attribute[1]. Assume the original attribute values are $x_1, \ldots, x_\eta$. After watermark insertion, value x_i becomes $x_i + e_i$, where e_i is a random variable that represents the perturbation to x_i caused by watermark insertion. Because watermark insertion changes a selected least significant bit pseudo-randomly, e_i equals 2^j or -2^j $(j = 0, 1, \ldots, \xi - 1)$ with the same probability $p/2$, where $p = \frac{1}{2\gamma\nu\xi}$ is the probability that a least significant bit is modified by watermark insertion. The mean of e_i is $E[e_i] = 0$ and variance of e_i is $V[e_i] = E[e_i^2] = \sum_{j=0}^{\xi-1} p \cdot 2^{2j} = \frac{p \cdot (2^{2\xi}-1)}{3}$. In this paper, we use $E[\cdot]$ and $V[\cdot]$ to denote mean and variance of a random variable, respectively.

Let $\mu = \frac{\sum_{i=1}^{\eta} x_i}{\eta}$ be the mean of original attribute values and let $\mu_e = \frac{\sum_{i=1}^{\eta} e_i}{\eta}$ be the error in computing μ after watermarking. One can derive that the expected error in computing μ is $E[\mu_e] = 0$ and the variance of the error is $V[\mu_e] = E[\mu_e^2] = \frac{\sum_{i=1}^{\eta} E[e_i^2]}{\eta^2} = \frac{E[e_i^2]}{\eta} = \frac{p \cdot (2^{2\xi}-1)}{3\eta}$.

Let $\sigma^2 = \frac{\sum_{i=1}^{\eta} (x_i - \mu)^2}{\eta}$ be the variance of original attribute values and let $\sigma_e^2 = \frac{1}{\eta} \cdot \sum_{i=1}^{\eta} [(x_i + e_i) - (\mu + \mu_e)]^2 - \frac{1}{\eta} \sum_{i=1}^{\eta} (x_i - \mu)^2$ be the error in computing σ^2 after watermarking. According to the standard theory for sample variance [1, 3], one can derive that the expected error in comput-

[1] a floating point number can be marked with its mantissa and treated like an integer.

ing σ^2 is $E[\sigma_e^2] = \frac{p\cdot(\eta-1)\cdot(2^{2\xi}-1)}{3\eta}$ and the variance of the error is $V[\sigma_e^2] = \frac{1}{\eta}\cdot[\frac{p\cdot(2^{4\xi}-1)}{15} - \frac{p^2\cdot(2^{2\xi}-1)^2}{9} + \frac{4p\cdot(2^{2\xi}-1)\sigma^2}{3}] + O(\frac{1}{\eta^2})$.

Robustness. To make sure that a detected watermark is correct, one needs to show that the following probabilities are low enough: (i) the probability of detecting a valid watermark from unmarked data, (ii) the probability of detecting no watermark from marked data even in the presence of attacks, and (iii) the probability that an attacker obtains the secret key from examining the pirated data. We call these probabilities *robustness measures.*

Let $b(k;n,p) = \binom{n}{k}p^k q^{n-k}$ be the binomial distribution function which gives the probability of obtaining *exactly* k successes out of n Bernoulli trials, where the result of each Bernoulli trial is true with probability p and false with probability $q = 1 - p$. Let $B(k;n,p) = \sum_{i=k+1}^{n} b(i;n,p)$ be the binomial distribution survival function which returns the probability of having *more than* k successes in n independent Bernoulli trials.

If watermark detection is applied to unmarked data, then it may possibly return "watermark detected" purely by chance. Let the "watermark" be extracted from data $\omega > 0$ times. Due to the use of pseudo-random mark bits, each time the watermark is extracted, it has probability 0.5 to match the corresponding pseudo-random mark bit. Therefore, the probability of detecting a valid watermark from unmarked data is $fh = B(\lfloor\tau\omega\rfloor;\omega,0.5)$. We call this probability *false hit.*

Now consider the probability that a pirate discovers a fictitious secret key that extracts a valid watermark from *pirated data.* A pirate can use the discovered key to claim legitimate ownership of the data. Alternately, a pirate can claim innocence by claiming that database owner used this type of "invertibility attack" [4, 2] to obtain evidence of piracy.

If a pirate randomly selects a secret key, then the probability that this key causes a valid watermark to be detected from pirated data is $p_{invert} = \max(\frac{1}{2^{|K|}}, B(\lfloor\tau\omega\rfloor;\omega,0.5))$, where the first term $\frac{1}{2^{|K|}}$ is the probability that the tried key is the real secret keys, and the second term is the false hit for detecting the watermark using a random "secret key."

Watermarking schemes should be robust against malicious attacks or benign update operations that may destroy the embedded watermark. The probability of detecting no watermark from marked data under various attacks is considered. We call such probability *false miss.*

First consider bit-flipping attack by which an attacker randomly selects some bits and toggles their values. Assume that the attack toggles each least significant bit with probability p_{flip}. Also assume that less than a half of the watermarkable bits are flipped (i.e., $p_{flip} \leq 0.5$) otherwise watermark detection can be applied to transformed data by flipping each watermarkable bit

back. The flipping of a single bit is modelled as an independent Bernoulli trial with probability p_{flip} of success and $1 - p_{flip}$ of failure.

Bit flipping attack does not change the size of data. Now, each watermark is embedded $\omega > 0$ times and it is extracted exactly the same times. The detection algorithm fails to detect the watermark only if at least $(1 - \tau)\omega$ embedded bits that correspond to the watermark are toggled (or, equivalently, more than $\omega - \lfloor \tau\omega \rfloor - 1$ bits are toggled). Thus, the false miss rate is $fm_{flip} = B(\omega - \lfloor \tau\omega \rfloor - 1; \omega, p_{flip})$.

Then consider subset attack where the pirated data is a subset of the tuples in a watermarked relation. Suppose that the attack examines each tuple independently and selects it with probability p_{subset} for inclusion in the pirated relation. The detection algorithm fails to detect the watermark only if all embedded bits are not included in the pirated relation. Therefore, the false miss rate is $fm_{subset} = (1 - p_{subset})^{\omega}$.

Consider superset attack where an attacker takes a pirated relation and mixes it with tuples from other sources to create a relation. Assume that each watermark is embedded ω times in the pirated relation, and that it can be extracted from the additional tuples $\omega \cdot p_{supset}$ times. After the superset attack, $\omega \cdot p_{supset}$ "watermark" bits will be extracted from the tuples that are added in the attack. Total $\omega(1 + p_{supset})$ watermark bits will be extracted from the entire data. The detection algorithm fails to detect the watermark only if at least $\omega(1 + p_{supset}) - \lfloor \omega(1 + p_{supset})\tau \rfloor$ embedded bits that correspond to the watermark are not as expected. Therefore, the false miss rate is $fm_{supset} = B(\omega(1 + p_{supset}) - \lfloor \omega(1 + p_{supset})\tau \rfloor - 1; \omega \cdot p_{supset}, 0.5)$ Note that $fm_{supset} = 0$ if $\lfloor \omega(1 + p_{supset})\tau \rfloor < \omega$, or $p_{supset} < \frac{1}{\tau} - 1$.

3. SELECT FEASIBLE WATERMARKING PARAMETERS

Before we study how to raise the watermarking errors to a predetermined threshold, we first investigate how to select feasible watermarking parameters under some error and robustness constraints.

The watermarking algorithm inserts watermarks by introducing a small number of errors in data. We wish to keep these errors within certain bounds and ensure that embedded watermarks are robust against attacks. However, more robust watermarks may introduce larger watermarking errors; thus, watermarking parameters should be chosen so as to balance between the watermarking errors and robustness. In this paper, we focus on the error and robustness measures given in the previous section, although our analysis framework can be extended to handle other measures. The purpose of this paper is not to give a complete study on the tradeoffs between the watermarking errors and robustness, but rather to demonstrate how to defend against additive attacks.

Error constraint. Consider watermarking error measures $V[\mu_e]$, $E[\sigma_e^2]$, and $V[\sigma_e^2]$ (see table 1). An *error matrix* Err is a triple $(V[\mu_e], E[\sigma_e^2], V[\sigma_e^2])$. We exclude $E[\mu_e]$ from the error matrix since it is always zero. Given an error threshold $Err^0 = (V^0[\mu_e], E^0[\sigma_e^2], V^0[\sigma_e^2])$, we call $Err \leq Err^0$ *error constraint.*

Three watermarking parameters γ, ν and ξ affect the error matrix. The *error constraint is satisfied* if there is some assignment of γ, ν, ξ such that $Err \leq Err^0$, where $\gamma \in [1, \eta], \nu \in [1, \nu^{max}]$ and $\xi \in [1, \xi^{max}]$. The upper bounds ν^{max} and ξ^{max} are finite integers that depend on data applications.

Table 1. Watermarking error matrix Err

Err	$(V[\mu_e], E[\sigma_e^2], V[\sigma_e^2])$
$V[\mu_e]$	$\frac{p\cdot(2^{2\xi}-1)}{3\eta}$
$E[\sigma_e^2]$	$\frac{p\cdot(\eta-1)\cdot(2^{2\xi}-1)}{3\eta}$
$V[\sigma_e^2]$	$\frac{1}{\eta}\cdot[\frac{p\cdot(2^{4\xi}-1)}{15} - \frac{p^2\cdot(2^{2\xi}-1)^2}{9} + \frac{4p\cdot(2^{2\xi}-1)\sigma^2}{3}] + O(\frac{1}{\eta^2})$

Robustness requirement. Consider the robustness measures given in the previous section. A *robustness matrix* Fhm is a triple (fh, p_{invert}, fm) where $fm = (fm_{flip}, fm_{subset}, fm_{supset})$ (see table 2). Note that the larger the robustness matrix, the less robust the watermarking scheme. Given a predetermined threshold $Fhm^0 = (fh^0, p^0_{invert}, fm^0)$, where $fm^0 = (fm^0_{flip}, fm^0_{subset}, fm^0_{supset})$, we call $Fhm \leq Fhm^0$ *robustness requirement.*

The robustness matrix depends on three watermarking parameters γ, τ, $|\mathcal{K}|$ and three attack parameters $p_{flip}, p_{subset}, p_{supset}$. The attack parameters are chosen for the worst case scenario where $p_{flip} = p^{max}_{flip}$, $p_{subset} = p^{max}_{subset}$, and $p_{supset} = p^{max}_{supset}$ and where $p^{max}_{flip} \in [0, 0.5]$, $p^{max}_{subset} \in [0, 1]$, and $p^{max}_{supset} \geq 0$ are the maximal values expected in the attacks that are to be tolerated in watermark detection. Because the robustness measures are monotonic increasing with the attack parameters, if the robustness requirement is satisfied for the worst case scenario, then it is satisfied for all other scenarios. The *robustness requirement is satisfied* if there is some assignment of watermarking parameters $\gamma, \tau, |\mathcal{K}|$ such that $Fhm \leq Fhm^0$ for the worst case scenario.

Problem 1. We consider the following *feasibility problem* in watermarking: Given a database relation *R,* decide whether the set of watermarking parameters $\gamma, \nu, \xi, \tau, |\mathcal{K}|$ has an assignment that satisfies both watermarking error constraint and robustness requirement.

Table 2. Watermarking robustness matrix Fhm

Fhm	(fh, p_{invert}, fm)	
fh	$B(\lfloor \tau\omega \rfloor; \omega, 0.5)$	
p_{invert}	$\max(\frac{1}{2^{\lvert\mathcal{K}\rvert}}, B(\lfloor \tau\omega \rfloor; \omega, 0.5))$	
fm	fm_{flip}	$B(\omega - \lfloor \tau\omega \rfloor - 1; \omega, p_{flip})$
	fm_{subset}	$(1 - p_{subset})^{\omega}$
	fm_{supset}	$B(\omega(1 + p_{supset}) - \lfloor \omega\,(1 + p_{supset})\tau \rfloor - 1; \omega \cdot p_{supset}, 0.5)$

There are five parameters γ, ν, ξ, τ and $|\mathcal{K}|$ that affect the evaluation of error constraint and robustness requirement. Parameter $|\mathcal{K}|$ can be determined first, independent of other parameters.

LEMMA 1 *The robustness requirement is satisfied if only* $|\mathcal{K}| \geq |\mathcal{K}|^{min}$, *where* $|\mathcal{K}|^{min} = \lceil \log_2 \frac{1}{p^0_{invert}} \rceil$.

Proof sketch The robustness requirement being satisfied requires $p_{invert} \leq p^0_{invert}$, which implies $\frac{1}{2^{|\mathcal{K}|}} \leq p^0_{invert}$, and further implies $|\mathcal{K}| \geq \lceil \log_2 \frac{1}{p^0_{invert}} \rceil$.

We can thus fix $|\mathcal{K}|$ in our analysis for any $|\mathcal{K}| \geq |\mathcal{K}|^{min}$. Then $p_{invert} \leq p^0_{invert}$ implies $B(\lfloor \tau\omega \rfloor; \omega, 0.5)) \leq p^0_{invert}$. Therefore, we can use $fh \leq \min(fh^0, p^0_{invert})$ to replace $p_{invert} \leq p^0_{invert}$ and $fh \leq fh^0$ in the robustness requirement. In this sense, the robustness requirement is examined for false hit and false miss only.

Then we have four watermarking parameters left for solving the feasibility problem, where γ decides how many times the watermark is embedded, ν and ξ determine where a watermark bit could be embedded, which depends on applications, and τ is used to balance between false miss and false hit.

Parameters $\gamma \in [1, \eta], \nu \in [1, \nu^{max}], \xi \in [1, \xi^{max}]$ are integers from finite space, while $\tau \in [0.5, 1)$ is real. We now illustrate that only finite τ's should be examined for the evaluation of the robustness requirement, including the computation of $fh, p_{invert}, fm_{flip}, fm_{supset}$. To compute fh, p_{invert} and fm_{flip}, only those τ's that correspond to different $\lfloor \tau\omega \rfloor$ need to be considered (otherwise they produce the same values). Those τ's could be $\tau = \frac{\lfloor 0.5\omega \rfloor}{\omega}, \frac{\lfloor 0.5\omega \rfloor + 1}{\omega}, \ldots, \frac{\omega - 1}{\omega}$, where $\omega = \lfloor \frac{\eta}{\gamma} \rfloor$. At most $\frac{\omega}{2} \leq \frac{\eta}{2}$ τ's need to be considered for each γ (i.e., for each $\omega = \lfloor \frac{\eta}{\gamma} \rfloor$). Similarly, to compute fm_{supset}, at most $\frac{\omega(1+p_{supset})}{2}$ τ's need to be considered. We have the following

LEMMA 2 *The watermarking parameters can be chosen from a finite domain for the feasibility problem.*

Let $S_\tau = \{\frac{\lfloor 0.5\omega \rfloor}{\omega}, \ldots, \frac{\omega-1}{\omega}\} \cup \{\frac{\lfloor 0.5\omega(1+p_{supset}) \rfloor}{\omega(1+p_{supset})}, \ldots, \frac{\omega(1+p_{supset})-1}{\omega(1+p_{supset})}\}$. We have the following

Proposition 1. A generic solution for the feasibility problem that examines the error constraint first:

for each $\gamma \in [1, \eta], \nu \in [1, \nu^{max}], \xi \in [1, \xi^{max}]$ and if $Err \leq Err^0$ do
  for each $\tau \in S_\tau$ do
    if $Fhm \leq Fhm^0$ then return true to the feasibility problem
return false to the feasibility problem

Proposition 2. Another solution for the feasibility problem that examines the robustness requirement first:

for each $\gamma \in [1, \eta], \tau \in S_\tau$ and if $Fhm \leq Fhm^0$ do
  for each $\nu \in [1, \nu^{max}], \xi \in [1, \xi^{max}]$ do
    if $Err \leq Err^0$ then return true to the feasibility problem
return false to the feasibility problem

In the worst case, we need to evaluate the error constraint and robustness requirement $\eta \cdot \nu^{max} \cdot \xi^{max} \cdot \omega(2 + p_{supset}^{max})/2 = O(\eta^2)$ times in both propositions.

Monotonicity. The monotonicity of the error and robustness measures can be used to develop more efficient algorithms for solving the feasibility problem. We have the following

LEMMA 3 *Consider the error matrix* $Err = (V[\mu_e], E[\sigma_e^2], V[\sigma_e^2])$ *as a function of* ξ, γ, ν. *(1)* Err *is monotonic increasing with* ξ. *(2) Let* $p = \frac{1}{2\gamma\nu\xi}$. $V[\mu_e]$ *and* $E[\sigma_e^2]$ *are monotonic increasing with* p. *(3) Let* $p_0 = \frac{3(2^{4\xi}-1)}{10(2^{2\xi}-1)^2} + \frac{6\sigma^2}{2^{2\xi}-1}$. *If* $p \leq p_0$, *then* $V[\sigma_e^2]$ *is monotonic increasing with* p, *otherwise* $V[\sigma_e^2]$ *is monotonic decreasing with* p *(note that we ignore the high order term* $O(\frac{1}{\eta^2})$ *in* $V[\sigma_e^2]$*).*

If $p_0 \geq 0.5$, then $V[\sigma_e^2]$ never decreases because $p \leq 0.5$. We will show that even if $p_0 < 0.5$, $V[\sigma_e^2]$ is monotonic increasing with p as long as $V[\sigma_e^2] \leq V^0[\sigma_e^2]$. To be consistent with the error tolerance assumption (see section 2), the threshold $V^0[\sigma_e^2]$ should be chosen such that it is less than $V[\sigma_e^2]$ at $p = 0.5$, which is the case that all least significant bits are randomized. Therefore, for all $p_0 < p \leq 0.5$, $V^0[\sigma_e^2]$ is less than $V[\sigma_e^2]$ since $V[\sigma_e^2]$ is monotonic decreasing with p. In a summary, all error measures are monotonic increasing with p as long as the error constraint is satisfied.

LEMMA 4 *Consider the robustness matrix* $Fhm = (fh, fm)$ *as a function of* τ, γ, *where* $fm = (fm_{flip}, fm_{subset}, fm_{supset})$. *(1)* fh *is monotonic decreasing while* fm *is monotonic increasing with* τ. *(2) Let* $\omega = \lfloor \frac{\eta}{\gamma} \rfloor$. fm_{subset} *is monotonic decreasing with* ω. *(3) If there is no rounding in computation, or the rounding effect is not significant enough to change the monotonicity of* Fhm *as a function of* γ, *then (3.1)* fh *is monotonic decreasing with* ω; *(3.2)* fm_{flip} *is monotonic decreasing with* ω *if* $p_{flip} + \tau \leq 1$ *and monotonic increasing with* ω *otherwise; and (3.3)* fm_{supset} *is monotonic decreasing with* ω *if* $p_{supset} \in [\frac{1}{\tau} - 1, \frac{2(1-\tau)}{2\tau-1}]$ *and monotonic increasing with* ω *otherwise (note that* $fm_{supset} = 0$ *if* $p_{supset} < \frac{1}{\tau} - 1$*).*

We note that Fhm is not strict monotonic with γ. One can easily prove that the monotonicity of fh, fm_{flip} and fm_{supset} may change frequently due to the rounding effect on $\tau\omega$ or $\tau\omega(1 + p_{supset})$ if ω is relatively small. Further, the monotonicity may depend on p_{flip} and p_{supset}.

The above monotonicity results can be incorporated into propositions 1 and 2 for more efficient search of feasible parameters. For example, we can use binary search for feasible $\tau \in S_\tau$, reducing the overall evaluation times from $O(\eta^2)$ to $O(\eta \log \eta)$.

Monotonic watermarking. A watermarking process is called *monotonic* if the error matrix Err is monotonic decreasing with γ and the robustness matrix Fhm is monotonic increasing with γ. Intuitively, monotonic watermarking means that the more bits in data are used to embed watermark, the more errors are introduced, and the more robust the embedded watermark. Most of watermarking schemes in the literature (e.g., [2–1, 10, 9]) are monotonic.

In monotonic watermarking, we can compute a minimal γ by binary search in step 1 of proposition 1. For this minimal γ, if there exist a τ (step 2) such that $Fhm \leq Fhm^0$ (step 3) then the answer to the feasibility problem is true, otherwise false. Similarly, in proposition 2, we can compute a maximal γ in step 1. By doing so, we can further reduce the evaluation times to $O(\log \eta \log \eta)$.

4. RAISING WATERMARKING ERRORS TO THRESHOLDS

Our strategy to defend against additive attacks is to raise the watermarking error to a threshold threshold such that any additive attack will introduce more errors than the threshold. In such a case, the pirated data is less useful or less competitive. Since we cannot bring multiple error components to different thresholds at the same time, we need to define a unique error metric.

Watermarking error metric. A *watermarking error metric* (or simply error metric) is a real function f of error matrix Err and f is monotonically decreasing with γ.

It is natural to require that $f(Err)$ be monotonic decreasing with γ; that is, the more the underlying data is modified, the larger the error metric. We know that all error measures $V[\mu_e], E[\sigma_e^2], V[\sigma_e^2]$ are monotonic increasing with p as long as the error constraint is satisfied. Therefore, if the error constraint is satisfied, any error measure, or the maximum of them, can be used as the error metric.

Let f^0 be a predetermined constant, which we call *watermarking error limit* (or simply error limit). The error limit represents the threshold of database usefulness. We hope to raise the watermarking error metric to the error limit such that the errors introduced by additive attack will exceed the error limit. We first consider how to maximize the error metric $f(Err)$ under error constraint and robustness requirement.

Problem 2. We consider the following *maximal error problem* in watermarking: Given a database relation *R,* maximize the error metric $f(Err)$ subject to $Err \leq Err^0$ and $Fhm \leq Fhm^0$.

Proposition 3. A generic solution for the maximal error problem could be:

for each $\gamma = 1, \ldots, \eta$ do
  if there exists $\tau \in S_\tau$ such that $Fhm \leq Fhm^0$ is satisfied do
    find $\nu \in [1, \nu^{max}], \xi \in [1, \xi^{max}]$ to maximize $f(Err)$ s.t. $Err \leq Err^0$
      if solution exists *return* $f(Err)$ to the maximal error problem
return no solution to the maximal error problem

The algorithm examines γ from small to large. Once a smallest γ is found such that $Fhm \leq Fhm^0$ and $Err) \leq Err^0$ are satisfied, the maximal $f(Err)$ is returned because $f(Err)$ is monotonic decreasing with γ. We can use the monotonicity of Fhm as a function of τ in step 2 and the monotonicity of Err as a function of ν, ξ in step 3. In the worst case, we need to evaluate the error metric and robustness requirement $O(\eta \log \eta)$ times.

Let f^{max} denote the maximal error metric returned by proposition 3. It is possible that $f^{max} < f^0$ (i.e., $f^{max} \neq f^0$) because: (i) the integer valued parameters γ, ν, ξ do not yield $f^{max} = f^0$; (ii) f^0 is larger than the error metric for any watermarking parameters. To raise the error metric to its threshold, we need the following watermark padding technique.

Watermark padding. Before watermark insertion, we flip each least significant bit with probability p_{pad} where $0 \leq p_{pad} \leq 0.5$. We call this process *watermark padding.* Watermark padding is done before any watermark is embedded.

The advantage of using watermark padding is that while watermark padding introduces errors, it does not affect the robustness. One problem remains unsolved: how to select p_{pad} such that the error limit is reached assuming that the maximal error metric f^{max} is less than f^0 before padding.

The padding error has the same form as Err as long as p (recall that $p = \frac{1}{2\gamma\nu\xi}$ is the probability that a least significant bit is flipped in watermarking process) is replaced by p_{pad}. Similarly, the error introduced by both padding and the subsequent watermarking has the same form as Err as long as p is replaced by p', where

$$p' = p_{pad} \cdot (1 - p) + (1 - p_{pad}) \cdot p$$

is the probability that a least significant bit is flipped either by padding or by watermarking but not by both. The error is denoted by Err'.

After padding and watermarking, the error metric $f(Err')$ is defined the same way as $f(Err)$ as long as p is replaced by p'. Function $f(Err')$ is monotonic increasing with p' and p_{pad}.

Let f^{max} be the maximal error metric without watermark padding. Assuming $f^{max} < f^0$, we study how to choose p_{pdd} such that $f(Err') = f^0$. Note $f^{max} = f(Err')$ when $p_{pad} = 0$. Because $f(Err')$ is monotonic increasing with p_{pad}, we have $f(Err') \geq f^{max}$. If we can prove that $f(Err') \geq f^0$ at $p' = 0.5$ ($p' = 0.5$ corresponds to $p_{pad} < 0.5$), then we can always find appropriate p_{pad} (e.g., by binary search) such that $f(Err') = f^0$. Recall that the error tolerance assumption does not allow all least significant bits to be randomized. Requiring $f(Err') \geq f^0$ at $p' = 0.5$ is consistent with the assumption because when $p' = 0.5$, every least significant bit is flipped randomly.

5. DEFENDING AGAINST ADDITIVE ATTACKS

Now consider how to defend against additive attacks using the techniques presented above. In an additive attack, a pirate inserts an additional watermark such that the ownership proof is ambiguous (both the owner and the attacker can detect a watermark from pirated data).

Our method will be based on the observation that an additive attack inevitably introduces additional error to the underlying data while rendering the original watermark less robust. By padding and watermarking, we have presented how to bring the watermark error metric to a predetermined limit. The limit represents the threshold of database usefulness. If the error exceeds the limit significantly, the underlying database is considered to be less useful or less competitive. We show that the additional errors introduced by additive attack will exceed the error limit.

Let $\gamma_{add}, \nu_{add}, \xi_{add}$ be the watermarking parameters in additive attack. Let $p_{add} = \frac{1}{2\gamma_{add}\nu_{add}\xi_{add}}$ be the probability that a least significant bit is flipped in additive attack. To prevent additive attack from embedding too few bits

(i.e., small p_{add}), a requirement (e.g., by convention) should be placed that a watermark cannot be used as ownership proof unless its robustness requirement is satisfied at a certain level (e.g., 10^{-9} for false hit rate). Under the same level of robustness requirement, we can assume that $p_{add} \simeq p$.

Let p'' be the probability that each least significant bit is flipped after padding, watermarking, and additive attack. We compare p'' with corresponding p' (after both padding and watermarking) and p (after watermarking only) in table 3. Except these probabilities, the overall error metrics for different cases are of the same form. For simplicity, we assume that the error metric is a linear function of these probabilities; thus, we only need to compare these probabilities for the purpose of comparing the corresponding error metrics.

Table 3. Error comparison (probability that each least significant bit is flipped)

watermarking only	$p = \frac{1}{2\gamma\nu\xi}$
padding and watermarking	$p' = p_{pad} \cdot (1-p) + (1-p_{pad}) \cdot p$
padding, watermarking, and additive attack	$p'' = p' \cdot (1-p_{add}) + (1-p') \cdot p_{add}$

No watermarking padding. First consider the case that the error error limit is reached without watermark padding; that is $p' = p$. In this case, $p'' \simeq 2p - 2p^2 \simeq 2p$ as $p_{add} \simeq p$. The error metric after the additive attack is almost double the error after watermarking only, which is already at the limit. Therefore, the error metric after the additive attack is well beyond the error limit. The owner of data may simply ignore such pirated data (after additive attack) since such data is less useful and competitive.

With large watermark padding. Now consider the case that watermark padding has been used to increase the watermarking error significantly in order to reach the error limit. Since the watermarking error is comparably small, one may expect that the additional error caused by an additive attack (under the same level of robustness requirement) is also small. Such pirated data may not be ignored by the owner even though its error is over the error limit.

In this case, one can resort to the previous methods for defending against additive attack (see Section 1). Those methods involve a trusted third party to either check the original copies, or register secret keys. However, our technique still plays an important role to thwart malicious attacks that may destroy the original watermark before the additive attack. This is important due to the following reasons.

- The essential prerequisite of previous methods for dispute resolution after an additive attack is that the original watermark is not destroyed by

other attacks (e.g., bit-flipping attack launched before the additive attack). The watermark padding technique makes it much harder to destroy the original watermark because the padding, which significantly increases watermarking errors to the error limit, also significantly restricts the errors that other attacks can introduce. For instance, if a malicious attack has to bring the error metric to the limit so as to destroy the original watermark, then after padding, such an attack has to introduce almost double the error beyond the limit.

- A pirate benefits more from a combination attack which first destroys the original watermark and then performs an additive attack. As observed by [1], "the benefit to the attacker from successfully establishing a false ownership claim is not as great as the benefit from destroying the watermark by means of a successful malicious attack. Indeed, if the ownership claims cannot be resolved, then customers may be wary of using contested data, thereby reducing the value of the pirated data to the attacker." Our padding technique helps to thwart such combination attacks.

6. CONCLUSION

Additive attacks are a severe threat to watermarking relational databases but have not been formally investigated before. This paper presents an effective solution that raises watermarking errors to a predetermined threshold such that the additional errors introduced by additive attacks will render the pirated data less competitive. In particular, we have solved the following problems:

- Feasibility problem: Is there some assignment of watermarking parameters such that both the error constraint and the robustness requirement are satisfied?
- Maximal error problem: How can we choose watermarking parameters to maximize a watermarking error metric?
- Watermark padding problem: How can we raise the watermarking error to a predetermined threshold if the maximal watermarking error metric is less than the threshold?

We are considering two directions in which to extend this work: (i) Enhance the underlying watermarking scheme such that it does not depend on primary key and attribute order. (ii) Investigate specific forms of watermarking errors in different application scenarios.

References

[1] R. Agrawal, P. J. Haas, and J. Kiernan. Watermarking relational data: framework, algorithms and analysis. *The VLDB Journal,* 12(2): 157–169, 2003.

[2] R. Agrawal and J. Kiernan. Watermarking relational databases. In *Proceedings of VLDB,* pages 155–166, 2002.

[3] Harald Cramer. *Mathematical Methods of Statistics.* Princeton University Press, 1946.

[4] S. Craver, N. Memon, B.L. Yeo, and M.M. Yeung. Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications. *IEEE Journal on Selected Areas in Communications,* 16(4):573–586, 1998.

[5] D. Gross- Amblard. Query-preserving watermarking of relational databases and xml documents. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS),* pages 191–201, 2003.

[6] Y. Li, V. Swarup, and S. Jajodia. Constructing a virtual primary key for fingerprinting relational data. In *Proceedings of ACM Workshop on Digital Rights Management (DRM),* October 2003.

[7] Y. Li, V. Swarup, and S. Jajodia. A robust watermarking scheme for relational data. In *the Thirteenth Annual Workshop on Information Technologies and Systems (WITS),* pages 195–200, December, 2003.

[8] SIIA. Database protection: Making the case for a new federal database protection law. http://www.siia.net/sharecontent/govt/issues/ip/dbbrief.html.

[9] R. Sion. Proving ownership over categorical data. In *Proc. IEEE International Conference on Data Engineering,* pages 584–596, 2004.

[10] R. Sion, M. Atallah, and S. Prabhakar. Rights protection for relational data. In *Proceedings of ACM SIGMOD International Conference on Management of Data,* pages 98–108, 2003.