Security of Public Watermarking Schemes for Binary Sequences

Qiming Li Ee-Chien Chang

School of Computing
National University of Singapore
[liqm, changec]@comp.nus.edu.sg

Abstract. In this paper, we focus on the security aspect of public watermarking schemes. Specifically, given a watermarked sequence $\widetilde{I}$, we consider smart attackers whose task is to find a non-watermarked sequence I' using as few calls to the publicly available detection routine as possible. We restrict the media to binary sequences and use Hamming distance as the measure. We study a class of watermarking schemes and give an attacker who uses expected $O(d(1+\log(n/k)))$ calls to find such I', where d and k are determined by the false alarm and distortion of the scheme, and n is the length of the sequence. This attacker is optimal when $k = o(n)$. By taking the number of calls required as a measure of the security, we can trade-off the requirements on security, false alarm and distortion.

1 Introduction

We consider the "public watermarking scheme" [7]. Under this setting, the detection routine is a black box accessible to the public, including the attackers. To access the detection routine, the public sends a sequence to a detector, which replies with `1` if the sequence is watermarked, and `0` otherwise. Given a watermarked sequence $\widetilde{I}$, the task of an attacker is to find a non-watermarked sequence I', which is as close to I as possible, using limited number of queries to the detector. Cox et al [7] give a heuristic for general watermarking schemes and an estimated number of queries required. The well-known Stir-mark [10] provides a list of practical attacks, many of which are based on image properties. In this paper, we view the attacks as games between the attacker and the watermarking scheme. We focus on a few schemes for binary sequences, and take the Hamming distance as the measure.

Our problem is related to the Twenty Questions Game proposed by Ulam in 1976 [13]. In the original game, the target is a secretly chosen integer between 1 and 2^{20}, and a player is to guess this integer by asking twenty yes-no questions. There are several variants of the Twenty Questions Game since then. For example, the Twenty Questions Game with Genes[11], and [1, 8]. We give a variant of the game that corresponds to the game between the watermarking scheme and the attacker. In this game, the player corresponds to the attacker of the watermarking scheme, and the player's questions correspond to the queries sent to the

detector. We give a randomized player who uses expected $O(d(1 + \log(n/k)))$ questions, where d, k and n are parameters of the game. The number of calls required by the attacker can serve as a measure of the security. This can be traded-off with the requirements on false alarm and distortion.

Our problem is different, however, from the collusion-secure fingerprint problem [2, 9, 12] in the way the watermarked sequences (queries) are generated. In the collusion-attack setting, each user is assigned a unique fingerprint, and an object watermarked with the unique fingerprint is distributed to each user. Some of the users may collude by comparing the different watermarked copies of the same object, and attempt to remove or modify the fingerprint. In our problem, there is only one attacker. The attacker is free to choose any sequence and the detector (available as a black box) has to disclose whether the chosen sequence is watermarked or not. Due to this flexibility, the attacker can intelligently choose a sequence, based on the outcomes of previously chosen sequences, that will lead to successful watermark removal.

We first give the notations used in this paper (Section 2), and then describe a class of watermarking scheme (Section 3). In Section 4, we focus on the Twenty Questions Game. We first give a lower bound (Section 4.1), followed by the randomized player (Section 4.2), and how the game relates to the original watermarking problem (Section 4.3). In Section 5, we give a few variations of our problem.

2 Notations

A watermarking scheme consists of an *encoder* and a *detector*. The encoder of a watermarking scheme takes a binary sequence $I = \langle a_1, a_2, \ldots, a_n \rangle$ as input and gives an encoded sequence $\widetilde{I}$. Let $\mathcal{K}$, the *kernel*, be the set of all possible encoded sequences. The encoder satisfies the *distortion* constraint, which requires the Hamming distance of $\widetilde{I}$ from I to be bounded by a predefined distortion ϵ. In the other end, the detector takes a sequence as input and outputs a 1 or 0 indicating whether the sequence is watermarked. Let $\mathcal{W}$ be the set of all watermarked sequences. The detector satisfies the constraint on the *false alarm* ratio F, that is, the probability of a randomly selected sequence being watermarked is bounded by F. If the underling distribution is the uniform distribution, then

$$F = 2^{-n}|\mathcal{W}|.$$

Besides the above constraint, the scheme should be resilient in the sense that under the influence of noise, the encoded sequence $\widetilde{I}$ should remain watermarked. There are many different models and requirements for the noises. A scheme that can withstand random noise is usually known as a *robust* scheme. In this paper, we consider security. We say that a scheme meets the security requirement (S, d_0) if, given a watermarked $\widetilde{I} \in \mathcal{K}$, any attacker requires at least expected S number of calls to the detector, so as to find a non-watermarked I', where $\|\widetilde{I} - I'\| \leq d_0$. Note that security implies robustness, because an attacker may wish to act like the random noise.

A lot of works have been done on the robustness of watermarking schemes, for example [6, 4, 5]. Relatively few theoretical works on smart attackers have been reported. This is the focus of this paper.

3 A Watermarking Scheme

This section describes a class of watermarking schemes for binary sequences of length n. This watermarking scheme is analogous to that in [3]. Each scheme is parameterized by the integers d, ϵ and k. The value of d, ϵ and k is made known to the public, including the potential attackers. What are kept secret by the encoder is a secret *key* K and a secret source coding *code-book* C. The code-book C is a collection of *codewords*, which are binary sequences of length k. The code-book satisfies the distortion requirement ϵ in the sense that every sequence is at most ϵ away from its nearest codeword. The secret key $K = \{h_1, h_2, \ldots, h_k\}$ is a set of k indices, where $1 \leq h_i \leq n$ for all $1 \leq i \leq k$. For a sequence I, call the sequence $\langle a_{h_1}, a_{h_2}, \ldots, a_{h_k} \rangle$ the *watermarking coefficients* of I.

Encoder. Given a sequence I to be watermarked, the encoder quantizes the watermarking coefficients of I to the nearest codeword in C. For example, if $\langle a_1, a_2, \ldots, a_k \rangle$ is the watermarking coefficients, and $\langle a'_1, a'_2, \ldots, a'_k \rangle$ is the codeword in C that is nearest to $\langle a_1, a_2, \ldots, a_k \rangle$, then the watermarked sequence $\widetilde{I}$ is the same as I except its watermarking coefficients are replaced by $\langle a'_1, a'_2, \ldots, a'_k \rangle$.

Detector. In the other end, the detector declares a sequence I to be *watermarked* if and only if the watermarking coefficients are within a distance d from a codeword in C. Thus, the kernel $\mathcal{K}$ of this scheme contains sequences whose watermarking coefficients are in C, and the watermarked sequences $\mathcal{W}$ are all the sequences within a distance of d from the kernel.

The false alarm and distortion of this scheme can be easily determined. Define $V_{N,R}$ to be the volume of a sphere in N-dimensional space with radius R, where the distance is measured as Hamming distance. That is,

$$V_{N,R} = \binom{N}{R} + \binom{N}{R-1} + \ldots + \binom{N}{1} + 1.$$

The false alarm F satisfies the following bound,

$$F \geq \frac{V_{k,d}}{V_{k,\epsilon}}. \tag{1}$$

The equality holds if and only if C is an ϵ perfect code. In this case, the distortion D is:

$$D = \epsilon. \tag{2}$$

For $k \gg \epsilon \gg d$, the right-hand-side in (1) is approximately $k^{d-\epsilon}$. Note that the false alarm (1) and distortion (2) do not depend on the size n. The size n plays

an important role in security. To see how the security requirement affects the choice of d and k, let us assume that low false alarm and small distortion are the only desirable properties. Then, with fixed distortion, k should be as large as possible and d should be 0. Since $d = 0$, the watermarked sequences are isolated "points" in $[0, 1]^n$. This amounts to finding a good source code for the binary sequence. By bringing in the security requirement, each sequence in the kernel should be surrounded by watermarked sequences. If not, an attacker can easily find a non-watermarked sequence by random perturbation. Intuitively, d should be as large as possible to enhance security. However, larger d will raise the false alarm (from (1)). Thus an important question is how to choose d and k for given requirements of false alarm, distortion and security. Next section gives an analysis on security that provides a trade-off for the watermarking requirements.

4 Twenty Questions Game with Watermark Attacker

Before we describe a watermark attacker, let us consider this guessing game involving a *player* and a *target*. The target K is a set containing k integers from $U = \{1, 2, \ldots, n\}$. The player knows the size of K and U before the game starts. The goal of the player is to determine at least $d + 1$ elements in K, using as few queries as possible. A query is represented by a set $Q \subseteq U$. The outcome of a query Q, denoted by $\mathcal{Q}(Q)$, is `Yes` if and only if

$$|Q \cap K| > d.$$

This game can be considered as a variant of the Ulam's game [13], and is similar to the Twenty Questions Game with Genes in [11]. In the Twenty Questions Game with Genes, the query is of the form "does a given interval contain an integer from K". The goal is to reconstruct K using as few queries as possible. The lower bound for a deterministic player of the Twenty Questions Game with Genes is $\log \binom{n}{k}$, which is approximately $k \log(n/k)$ for $k \ll n$.

Our game differs from the Twenty Questions Game with Genes in a few ways. Our player has an easier job because he only needs to determine $d + 1$ elements in K. On the other hand, our queries are more general, and thus might provide less information.

4.1 Lower Bound

A lower bound for any deterministic player in our game is

$$\log \left(\binom{n}{d+1} / \binom{k}{d+1} \right). \tag{3}$$

In the guessing game, the player wins if he can identify $d + 1$ elements in the target K. Before the game starts, from the player point of view, all the $\binom{n}{k}$

sets of k elements are possible targets. This class of possible targets reduces as the player asks questions. When all the possible targets contain $d+1$ common elements, the player can confidently outputs these $d+1$ elements and wins the game.

Let us look at the decision tree where each node is a class of possible targets. Thus, the root is the class of size $\binom{n}{k}$. In the best scenario for the player, each leaf is a class with largest possible number of targets, which is $\binom{n-(d+1)}{k-(d+1)}$ (this is the number of possible targets where $d+1$ elements are fixed). Therefore, the height of the tree is at least

$$\log\left(\binom{n}{k} / \binom{n-(d+1)}{k-(d+1)}\right),$$

which is equal to (3). This gives the claimed lower bound. Note that the bound is in $\Omega(d\log(n/k))$, and for small k and d, the bound is approximately $d\log(n/k)$.

By assigning each node with equal probability, and using the Yao's principle[14], we can also show that any randomized player requires expected $\Omega(d\log(n/k))$ questions.

4.2 A Player (Deterministic and Probabilistic)

The job of a player is to identify at least $d+1$ elements in K. Our strategy is to first find a small subset $U_0 \subset U$ that contains at least $d+1$ elements in K. Next, the size of U_0 is gradually reduced in a way similar to binary search, until its size becomes $d+1$, which is what we want. To find the small U_0, the deterministic player uses step §1 in the algorithm below. However, this step requires $(k-1)/d-1$ queries in the worst case. The randomized player improves this step to expected constant number of queries by first shuffling the coefficients (§0).

Deterministic Algorithm. Here we present a deterministic algorithm for the guessing game.

§1. Divide U evenly into $(k-1)/d$ groups, $U_1, U_2, \ldots, U_{(k-1)/d}$. Find an i such that $\mathcal{Q}(U_i)$ gives `Yes`. Let $Q_0 = U_i$.
§2. Divide Q_0 evenly into $2d+2$ groups, $G_1, G_2, \ldots, G_{2d+2}$. Let $L = \phi$ and $G_0 = \phi$, where ϕ is the empty set.
§3. Find the largest $i \in \{0, 1, 2, \ldots, 2d+2\}$ such that $\mathcal{Q}((G_0 \cup G_1 \cup G_2 \cup \ldots \cup G_i) \cup L)$ gives `No`. Update L to be $L \cup G_{i+1}$. Repeat step §3 until no such i exist.
§4. Update Q_0 to be L. If Q_0 contains only $d+1$ elements, Q_0 is the result. Otherwise repeat from step §2.

By the pigeon-hole principle, there exists one group U_i in step §1 that contains at least $d+1$ elements from K, and $\mathcal{Q}(U_i)$ gives `Yes`. Therefore, the number of queries needed for this step is at most $(k-1)/d-1$.

Since each G_{i+1} identified in step §3 contains at least one element from K, the repeat-loop in step §3 repeats for at most $d+1$ rounds. Therefore, step §2 to §3 identify at most $d+1$ groups among $G_1, \ldots, G_{2d+2}$, which in total contain at least $d+1$ elements from K. It follows that the size of L is at most $|Q_0|/2$. Note that step §3 can be completed using a single loop, which uses a total of $2d+2$ queries.

Step §2 to §4 are repeated until $|Q_0|$ is reduced to $d+1$. Thus, the total number of rounds is at most $\max(1, \log(n/k))$ and the total number of queries required to complete the outer-loop is $O\left(d(1+\log(n/k))\right)$.

In the worst case, the number of queries needed by the player is $O(k/d + d(1+\log(n/k)))$.

Randomized Algorithm. When k is small, the above is dominated by the term $d\log(n/k)$. However, if k is large, the term k/d would dominate, which is undesirable. Now we introduce a probabilistic player, who uses expected $O(d(1+\log(n/k)))$ queries.

§0. Permutes the set U uniformly at random.

This probabilistic player performs step §0, and then proceeds from step §1 of the deterministic player.

Recall that the size of a group U_i in step §1 is $dn/(k-1)$. Since the input U is randomly shuffled in step §0, each element in U_i has the probability k/n to be from K. Let Z be the number of elements in U_i that are from K. Then the expected value of Z is $E(Z) = dk/(k-1)$. Since $d < dk/(k-1) < d+1$, the probability $Pr[Z \geq (d+1)] = Pr[Z > E(Z)]$, which is greater than some constant that is approximately $1/2$.

Since we are doing selection without replacement in step §1, if the group we select contains less than $d+1$ elements from K, the following groups would have greater probability to contain at least $d+1$ elements from K.

Thus, step §1 can be completed in expected $O(1)$ queries. This gives expected $O(d(1+\log(n/k)))$ for the randomized algorithm. When $k = o(n)$, we have an optimal $O(d\log(n/k)$ algorithm.

4.3 A Watermark Attacker

For a set X of indices, let I_X be the sequence whose i-th coefficient is 1 if and only if $i \in X$. Given a sequence I and a (n, k, d, ϵ) scheme, the task of the attacker is to find a non-watermarked sequence I' such that $\|I' - I\| \leq d+1$. The attacker knows the values of n, k, d, and ϵ. What he does not know is the code-book and the secret key K. Here, we assume that the code-book is a perfect binary code.

Without loss of generality, we can assume that the given sequence I consists of only 0's, that is $I = \langle 0, 0, \ldots, 0\rangle$, and the code-book contains $\langle 0, 0, \ldots, 0\rangle$. Now, it suffices for the attacker to find a set of indices X such that $|X| = d+1$ and $X \subseteq K$. Since $|X \cap K| = d+1$ and C is a perfect code, I_X is non-watermarked.

The watermark attacker corresponds to the player in the Twenty Questions Game in Section 4, the secret key K corresponds to the target, and the detector corresponds to the query. The sequence I_X is watermarked if $\mathcal{Q}(X)$ gives `No`. Note, however, the two problems are not completely equivalent. Consider a X' where $|X' \cap K| > d$. It is possible that $I_{X'}$ is still watermarked, although $\mathcal{Q}(X')$ gives `Yes`. However, the number of such X' is insignificant comparing to the number of $\tilde{X}$ where $|\tilde{X} \cap K| > d$.

Trade-off with False Alarm and Distortion. For a given false alarm F and distortion D, we want to know how to choose d, k, and ϵ to achieve the highest security. By taking the approximate lower bound on the number of calls to the detector required as a measure of the security S,

$$S = \log \left(\binom{n}{d+1} / \binom{k}{d+1} \right), \tag{4}$$

combining with the equation for false alarm (1) and distortion (2), we can determine the right parameters. For simplicity, use the approximations $S \approx d \log(n/k)$ and $F \approx k^{d-\epsilon}$. Together with (2) and (4), it can be shown that S has the maximum value

$$S_{max} = (\sqrt{D \log n} - \sqrt{\log F^{-1}})^2 \tag{5}$$

when

$$d = D - \sqrt{D \log_n F^{-1}}. \tag{6}$$

5 Variations of the Game

In this section we will examine some variations of the game and the corresponding watermarking schemes. These variations try to confuse the player by introducing a liar and multiple targets into the game. However, as we will see, although these mechanisms make the game more difficult, they degrade the performance on false alarm and distortion. In the overall tradeoff, they do not improve the security.

5.1 Twenty Questions Game Between Watermark Attacker and Liar

The Twenty Questions Game with watermark attacker can be extended to a game with a liar. That is, with some constant probability $p < 1/2$, the answer to the query would be wrong. The error can be two-sided: a type-1 error with probability p_1, when $|Q \cap K| > d$ but the answer is `No`; and a type-2 error with probability p_2, when $|Q \cap K| \leq d$ but the answer is `Yes`.

If $p_2 = 0$, our algorithm will still give a correct solution. However, because of the effect of p_1, the expected number of groups identified in step §2 and §3 will be increased to $(d+1)(1+p_1)$. So the factor by which U_0 is reduced is not

1/2 but $(1+p_1)/2$. Thus the expected cost of our randomized algorithm will be increased by a constant factor $1/(1-\log(1+p_1))$, but is still $O(d\log(n/k))$.

In order to take p_2 into consideration, we need to slightly modify step §3 as the following.

§3. Find the largest $i \in \{0, 1, 2, \ldots, 2d+2\}$ such that $\mathcal{Q}((G_0 \cup G_1 \cup G_2 \cup \ldots \cup G_i) \cup L)$ gives `No`. Update L to be $L \cup G_{i+1}$. Repeat step (3) until no such i exist. If $L = Q_0$, stop with no solution.

Now our algorithm becomes a Monte Carlo algorithm, which gives a correct solution with certain probability. Obviously, if no errors occur in all the queries, the result would be correct. The probability of such cases is $P = (1-p)^{c_1 d \log(n/k)}$, where c_1 is some positive constant.

If we repeat the same query for T times and take the majority answer, the new probability of error $p' < e^{-c_2 T}$, for some positive constant c_2. Now the probability for our algorithm to give a correct solution is $P = (1-p')^{c_1 d \log(n/k)}$, which is approximately $1 - p' c_1 d \log(n/k)$ for small p'. Let $p' c_1 d \log(n/k) < e^{-c_2 T} c_1 d \log(n/k) < 1/2$, then $T > (1/c_2)\ln(2c_1 d \log(n/k))$. Thus for $P > 1/2$, the expected number of queries required by our algorithm is

$$O(d\log(n/k)\log(d\log(n/k))).$$

Therefore, by repeating the algorithm for an expected constant number of times, we will have a correct solution.

The liar in the Twenty Questions Game corresponds to a detector that gives a wrong answer in the watermarking scheme. With probability p_1, the sequence is not watermarked but the detector says that it is; with probability p_2, the sequence is watermarked but the detector says that it is not. We can see that in practice p_2 should be negligible, otherwise we could just randomly select a sequence near the watermarked one, and make the detector say that it is not watermarked by repeatedly sending the sequence to it. Because of p_1, the false alarm F will be increased to $F' = F + p_1$. In order for p_1 to be significant enough to our algorithm, p_1 has to be greater than $c_3/d\log(n/k)$, for some constant c_3. However, since the original false alarm $F \approx k^{d-\epsilon} \ll 1/d\log(n/k)$, it is very difficult, if not impossible, to compensate for the false alarm by adjusting the values of k and d. Even if we want to do so, the number of queries will increase because of the changes to d and k.

5.2 Modified Twenty Questions Game with Multiple Targets

We can also extend the Twenty Questions Game to have two secret sets $\mathcal{K}_1$ and $\mathcal{K}_2$. The answer to the query would be `Yes` if $|Q \cap \mathcal{K}_1| > d$ and $|Q \cap \mathcal{K}_2| > d$, and `No` otherwise. The player is still required to identify more than d elements from $\mathcal{K}_1$.

Interestingly, the algorithm and analysis in Section 5.1 are still applicable, with $p_2 = 0$. Therefore it also can be solved in expected $O(d\log(n/k))$ queries. This variation can be easily extended further to more than two secret sets,

where different secret sets may have different values of d and k. However, those variations will not make the game more difficult.

The corresponding watermarking scheme would have multiple code-books, and only use one of them to watermark a sequence. The choice of the code-book to be used can be random, or based on sequence specific information, such as the nearest distances from the codewords of each code-book. Similar to the watermarking scheme in Section 5.1, the false alarm F increases significantly due to p_1, and the number of calls to the detector increases if we want to compensate for F.

6 Remark and Future Works

We have also explored other watermarking schemes on binary sequences. It turns out that the simple watermarking scheme in Section 3 outperforms them. This leads to a general question: given the requirements on false alarm and distortion, what is the highest security (measured in term of number of calls to the detector) we can achieve. We do not know the solution to this general question. We suspect that the security of a watermarking scheme is closely related to the *critical distance*, that is, the radius of the smallest sphere centered at the kernel, whose surface contains roughly half watermarked sequences. Note that our randomized player given in Section 4.2 uses this distance to obtain the set U_0. We also do not know any non-trivial bound of this distance with a given false alarm and distortion. Many interesting problems remain open.

References

1. Andris Ambainis, Stephen A. Bloch, and David L. Schweizer. Playing twenty questions with a procrastinator. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 844–845. ACM Press, 1999.
2. D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *IEEE Trans. on Information Theory*, 44(5):1897–1905, 1998.
3. E.C. Chang and M. Orchard. Geometric properties of watermarking schemes. In *ICIP*, volume 3, pages 714–717, 2000.
4. B. Chen and G.W. Wornell. Achievable performance of digital watermarking systems. *IEEE Int. Conf. on Multimedia Computing & Systems*, 1:13–18, 1999.
5. J. Chou, S.S. Pradhan, and K. Ramchandran. On the duality between distributed source coding and data hiding. *33rd Asilomar conference on Signals, System and Computers*, pages 1503–1507, 1999.
6. M. Costa. Writing on dirty paper. *IEEE Trans. on Information Theory*, 29(3):439–441, 1983.
7. I.J. Cox and J-.P. Linnartz. Public watermarks and resistance to tampering. *IEEE Int. Conf. on Image Processing*, 3(0_3–0_6), 1997.
8. Aditi Dhagat, Peter Gács, and Peter Winkler. On playing "twenty questions" with a liar. In *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, pages 16–22. ACM Press, 1992.

9. J. Kilian, F.T. Leighton, L.R. Matheson, T.G. Shamoon, R.E. Tarjan, and F. Zane. Resistance of digital watermarks to collusive attacks. In *IEEE International Symposium on Information Theory*, page 271, 1998.
10. Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Attacks on copyright marking systems. In *Information Hiding, Second International Workshop*, number 1525 in LNCS, pages 219–239. Springer-Verlag, 1998.
11. P.A. Pevzner. *Computational Molecular Biology: An Algorithmic Approach*. The MIT Press, 2000.
12. Harold S. Stone. Analysis of attacks on image watermarks with randomized coefficients. Technical report, NEC Research Instititute, 1996.
13. S. Ulam. *Adventures of a mathematician*. Scribner and Sons, 1976.
14. A.C.-C. Yao. Probabilistic computations: Twoard a unified measure of complexity. *18th IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1977.