# Improved Versions of Tardos' Fingerprinting Scheme

Oded Blayer and Tamir Tassa*

March 9, 2008

**Abstract**

We study the Tardos' probabilistic fingerprinting scheme and show that its codeword length may be shortened by a factor of approximately 4. We achieve this by retracing Tardos' analysis of the scheme and extracting from it all constants that were arbitrarily selected. We replace those constants with parameters and derive a set of inequalities that those parameters must satisfy so that the desired security properties of the scheme still hold. Then we look for a solution of those inequalities in which the parameter that governs the codeword length is minimal. A further reduction in the codeword length is achieved by decoupling the error probability of falsely accusing innocent users from the error probability of missing all colluding pirates. Finally, we simulate Tardos scheme and show that, in practice, one may use codewords that are shorter than those in the original Tardos scheme by a factor of at least 16.

## 1 Introduction

A data distribution system is any setting in which a data provider is providing data to a large group of paying users. Piracy occurs when one or few legal users redistribute the data for which they paid to illegal users, thus rendering financial losses to the legitimate data provider. The problem of protecting copyrighted data became acute in the digital era, as digital data can be duplicated perfectly, without quality degradation, easily stored on optic and magnetic media, and seamlessly distributed using the Web. Hence, tracing techniques are required in order to find the source of such information leakage, disconnect it and press charges against those traitorous users.

Digital fingerprinting (or watermarking) is a method that aims at protecting copyrighted data against piracy. The idea is to personalize each copy of the data prior to distribution, by embedding in it some unique mark (called *the fingerprint*), so that two properties hold:

- The fingerprint does not affect the intended use of the data in which it was embedded. For instance, if the data is a software, the fingerprinted software must run correctly. As another example, if the data is a digital movie, the fingerprinted movie must be indistinguishable from the original movie by a human viewer.
- The users must be incapable of removing the fingerprint without the risk of changing or removing other parts of the data that are relevant for its functionality.

Such fingerprinting is a good solution against piracy, as long as pirates do not collude. However, if several pirates collude, they may combine their personalized copies in order to create a new copy that is still fully functional, but is different from each of the personalized copies that they possess. This way, it may be hard or even impossible to link the distributed copy to any of the pirates that collaborated in generating it. Moreover, such a copy may even be identical or similar to a personalized copy that was given to an innocent user that has nothing to do with the coalition of pirates. Fingerprinting schemes are mechanisms that were devised in order to thwart such piracy. Such schemes consist of a fingerprinting algorithm, coupled with a tracing algorithm, as we proceed to define.

*Department of Mathematics and Computer Science, The Open University, Ra'anana, Israel.

### 1.1 The model

Let $U = \{u_1, \ldots, u_n\}$ denote the set of users of some data distribution system. As described above, each user gets a personalized fingerprinted copy of the data. The fingerprinting algorithm consists of the following ingredients:

- A marking alphabet, $\Sigma$, where $|\Sigma| = r$.
- A codeword length $m$.
- A one-to-one personalization function $P : U \to \Sigma^m$ that determines how to mark the data that is provided to a particular user with a codeword in $\Sigma^m$.

The technique in which the selected codeword is inserted into the data so that the two properties that we described earlier hold, depends on the type of data. E.g., Cox et. al. [4] described such techniques for fingerprinting images.

The fingerprinting algorithm is coupled with a tracing algorithm. Assuming that the coalition of pirates is $T = \{t_1, \ldots, t_c\} \subset U$, the tracing algorithm consists of the following ingredients:

- A generation assumption: Letting $P(T) = \{P(t_1), \ldots, P(t_c)\}$ be the set of codewords in the copies that are owned by the pirates, $\langle P(T) \rangle$ denotes the set of codewords that could be generated by the pirates and be placed in the pirate copy. $\langle P(T) \rangle \subset (\Sigma \cup \{?\})^m$, where "?" denotes an unreadable mark. Different assumptions were made in different studies about the strength of the generation operation $\langle \cdot \rangle$, depending on the underlying application. The minimal generation assumption is

$$\langle P(T) \rangle = \{y \in \Sigma^m \ \text{ s.t. } y_j \in \{P(t_1)_j, \cdots, P(t_c)_j\} \text{ for all } 1 \leq j \leq m\} \ . \tag{1}$$

  Namely, pirates with minimal capabilities can detect positions where their codewords differ and place there any one of the marks that they hold in that position. The maximal generation assumption, on the other hand, is

$$\langle P(T) \rangle = \{y \in (\Sigma \cup \{?\})^m \ \text{ s.t. } y = P(t_1)|_R\} \ , \tag{2}$$

  where $R = \{j : P(t_1)_j = \cdots = P(t_c)_j\}$. Namely, in marking positions that are detectable by the pirates, they may place any marking symbol from $\Sigma$ or completely remove that mark.

- A tracing algorithm that, given a pirate copy, aims at tracing back (at least) one pirate that collaborated in producing that copy. This algorithm may be therefore viewed as a function $\sigma : \langle P(T) \rangle \to 2^U$.

The following definition extends the notion of $\varepsilon$-security, as appears in [1, 7].

**Definition 1.1** *A fingerprinting scheme is called $(\varepsilon, \hat{\varepsilon})$-secure against coalitions of size $c$ if for any $T \subset U$ of size $|T| \leq c$, and for every $y \in \langle P(T) \rangle$, the following two properties hold:*

$$Pr\left(\sigma(y) \cap (U \setminus T) \neq \emptyset\right) \leq \varepsilon \, , \tag{3}$$

*and*

$$Pr\left(\sigma(y) \cap T = \emptyset\right) \leq \hat{\varepsilon} \, . \tag{4}$$

*In case $\hat{\varepsilon} = \varepsilon$, the scheme is called $\varepsilon$-secure.*

As an example, consider a data provider that sells viewing rights of the latest digital features. Since digital material may be cloned many times without experiencing quality degradation, immoral users that paid for their copies might redistribute such copies for a bargain price in the black market. In order to deter such users and to be able to trace them in case that they do

exercise such piracy, Boneh and Shaw [1] suggested a fingerprinting technique. As in our framework, the original data is personalized prior to distribution. The movie $V$ is broken up to $m$ short segments, $V = V_1||\dots||V_m$, where $||$ denotes concatenation, and to each segment, $V_j$, $r$ almost-identical variants are generated, $V_j \mapsto \{V_j^1, \dots, V_j^r\}$. Letting $\Sigma = \{1, \dots, r\}$, if user $u \in U$ was assigned the codeword $P(u) \in \Sigma^m$, he will get the following version of the movie:

$$V(u) = V_1^{P(u)_1}||\cdots||V_m^{P(u)_m} \ .$$

Boneh and Shaw [1] concentrated on the case $r = 2$ and designed fingerprinting schemes that are capable of tracing at least one of the colluding pirates that participated in producing the pirate version, with an error probability as small as desired. In the binary case, the minimal and maximal generation assumptions, (1) and (2), are equivalent: While the minimal assumption forces the pirates to choose in every position a mark that they have, the maximal assumption allows them to put in detectable positions any mark from $\Sigma$ or nothing at all ("?"). But in the binary case, a detectable mark occurs in segments where the pirates have all $r = 2$ variants. In that case there is not much point in the pirates removing the mark (an operation that usually damages the quality of the produced copy); instead, they could pick any of the $r = 2$ variants that they have.

Schemes that are $\varepsilon$-secure with $\varepsilon = 0$ are called deterministic. A necessary condition for a scheme to be deterministic is that $r > c$, [5, Theorem 1]. Hence, binary schemes ($r = 2$) are never deterministic and their output is always accompanied by a small error probability.

### 1.2 Related work

Fingerprinting schemes with the identifiable parent property (IPP) were introduced by Chor et al. under the name *traitor tracing schemes* [2]. They considered the setting of a Pay-TV system, where Conditional Access techniques are implemented in order to deny access to content from non-paying users. In such systems, the stream of content that is broadcast from the center is encrypted, and each paying user is given a decoder with embedded keys that are capable of decrypting the transmission. In order to thwart potential users from cloning their decoders and selling them to illegal users, each decoder is personalized by a unique selection of decryption keys that enables the decryption of the entitlement messages (the encrypted messages that hold the periodical keys that are used for the encryption of the broadcast). Their fingerprinting scheme falls under the model of Section 1.1 and the relevant generation assumption is the minimal one, (1). In their simplest scheme, [2, Section 3.3], $r = 4c$, $m = 4c \log_2(n/\varepsilon)/3$, and the algorithm is simply the majority algorithm: Given a pirate decoder that is marked by a codeword $y \in \Sigma^m$, the tracing algorithm $\sigma$ outputs a user $u_i \in U$ whose codeword $P(u_i)$ has the maximal number of matches with $y$. The above selection of parameters, $r$ and $m$, ensures that the scheme is $\varepsilon$-secure.

Boneh and Shaw [1] described binary fingerprinting codes, $r = 2$. The price that is incurred by using the smallest possible marking alphabet is in the length of the codewords. Their basic code has length $m = O(n^3 \log(n/\varepsilon))$[1] and it is $\varepsilon$-secure for any size of coalition. By combining this basic code with the above described scheme of Chor et al., they devised another $\varepsilon$-secure fingerprinting code of length $m = O(c^4 \log(1/\varepsilon) \log(n/\varepsilon))$ for coalitions of size at most $c$ for some $c < n$.

Boneh and Shaw used restricted randomization in constructing their codewords. In their basic scheme, they first fix some large parameter $d$ and set $m = (n-1)d$. Then they construct the codewords $w_i = 0^{(i-1)d}1^{(n-i)d}$, $1 \le i \le n$. Finally, they select a random permutation $\pi \in S_m$ and the codeword of user $u_i$ is then $P(u_i) = \pi(w_i)$. Namely, randomization kicks in only in permuting the bits of the codewords, but not in selecting the values of those bits. In other words, the bits in any given position $1 \le j \le m$ – $(P(u_1)_j, \dots, P(u_n)_j)$ – obey some deterministic pattern;

[1] Hereinafter, log denotes the natural logarithm.

specifically, they take the form $(1,\ldots,1,0,\ldots,0)$, where the number of leading 1s is anywhere between 0 and $n$.

Tardos [7] took Boneh and Shaw's ideas one step further and introduced a full randomization in determining the codewords. In his fingerprinting code, there are $m$ binary distributions and then $P(u_i)_j$ is selected to be 0 or 1 according to the $j$th distribution. By carefully selecting those distributions and designing the tracing algorithm, the resulting codeword length is almost the square root of that in the Boneh-Shaw scheme – $m = O(c^2 \log(n/\varepsilon))$. This code length is optimal, within a constant factor, for sufficiently small (yet reasonable) values of $\varepsilon$.

## 1.3 The Tardos fingerprinting code

### 1.3.1 Phase 1: Creating the codebook

The codewords are generated as follows:

1. Set the length of the codewords to $m = d_m c^2 k$, for some integral constant $d_m$, where hereinafter $k = \lceil \log \frac{1}{\varepsilon} \rceil$.
2. Select probabilities $p_j$ for all $1 \le j \le m$, in a manner that is explained below.
3. For all $1 \le i \le n$ (a loop over all users) and for all $1 \le j \le m$ (a loop over the bits of the codeword of that user) set $P(u_i)_j = 1$ with probability $p_j$ and $P(u_i)_j = 0$ otherwise.

The resulting codebook, $\{P(u_1),\ldots,P(u_n)\} \subset \{0,1\}^m$, is denoted $F_{n,c,\varepsilon}$.

Next, we describe the critical part of selecting the probabilities. Let $d_t \ge 1$ be some constant and set

$$t = \frac{1}{d_t c}, \quad \sin^2 t' = t.$$

Since we assume hereinafter that $c > 2$, then $t \in (0, \frac{1}{2})$ and, consequently, $t' \in (0, \frac{\pi}{4})$. With this choice of parameters, we select $r_j$ uniformly at random from the interval $[t', \frac{\pi}{2} - t']$, and then set $p_j = \sin^2 r_j$, $1 \le j \le m$.

### 1.3.2 Phase 2: Accusation

Define the array

$$V_{i,j} = \begin{cases} \sqrt{\frac{1-p_j}{p_j}} & \text{if } P(u_i)_j = 1 \\ -\sqrt{\frac{p_j}{1-p_j}} & \text{if } P(u_i)_j = 0 \end{cases}, \quad 1 \le i \le n, \quad 1 \le j \le m\ . \tag{5}$$

Assume that the codeword that was extracted from the pirate copy is $y \in \{0,1\}^m$. Then user $u_i$, $1 \le i \le n$, will be accused if

$$\sum_{j=1}^{m} y_j V_{i,j} > z, \tag{6}$$

where $z = d_z ck$ and $d_z$ is some constant. The set of all users that are accused by this algorithm when the pirate codeword is $y$ is denoted $\sigma(y)$.

### 1.3.3 Properties of the code

The code that we described above uses several undetermined constants. Those constants are $d_m$ and $d_t$ from Phase 1, and $d_z$ from Phase 2. The values that were set by Tardos for those constants were:

$$d_m = 100, \quad d_t = 300, \quad d_z = 20\,. \tag{7}$$

With this choice of constants, and under generation assumption (1), Tardos proved the following claims.

**Theorem 1.1** *Let $u_i \in U$ be an arbitrary user, and $T \subseteq U \setminus \{u_i\}$ be a coalition of arbitrary size. Then for any codeword $y \in \langle P(T) \rangle$,*

$$\Pr[u_i \in \sigma(y)] < \varepsilon.$$

**Theorem 1.2** *Let $T \subset U$ be a coalition of size $|T| \leq c$. Then for any codeword $y \in \langle P(T) \rangle$*

$$\Pr[T \cap \sigma(y) = \emptyset] < \varepsilon^{c/4}.$$

Theorem 1.1 bounds the probability of falsely accusing just one innocent user when using the codebook $F_{n,c,\varepsilon}$. Hence, in order to achieve $\varepsilon$-security, one needs to use the codebook $F_{n,c,\varepsilon/n}$. In view of Theorem 1.2, that code is in fact $(\varepsilon, \hat{\varepsilon})$-secure (in the sense of Definition 1.1) with $\hat{\varepsilon} = (\varepsilon/n)^{c/4}$. Note that $\hat{\varepsilon} \ll \varepsilon$ because $n$ is typically a very large number and $c$, being an assumed upper bound on the size of the coalition of pirates (and not the actual size of the active coalition), is usually set to values for which $c/4 \gg 1$.

### 1.4 Our contribution

The choice of constants made by Tardos, (7), seems, at first, arbitrary. As this code has a minimal length to within a constant factor, a natural question arises: does there exist a different choice of constants with a smaller value of $d_m$ - the constant that determines the codeword length?

In this study we retrace Tardos' analysis and extract from it all constants that were arbitrarily selected. We identify a set of seven constants that appear in Tardos' analysis that could be tuned differently in order to decrease the codeword length. Those constants include the above mentioned $d_m$, $d_t$ and $d_z$, and four other constants that pop up in the proofs of Theorems 1.1 and 1.2. We replace those constants with parameters and derive a set of inequalities that those parameters must satisfy in order for Theorems 1.1 and 1.2 to hold. Then, we look for a solution of those inequalities in which $d_m$ is minimal.

Another way to reduce the codeword length is through the error probabilities in Theorems 1.1 and 1.2. As noted at the end of Section 1.3, the Tardos fingerprinting scheme is $(\varepsilon, \hat{\varepsilon})$-secure with $\hat{\varepsilon} \ll \varepsilon$. However, a better setting of error probabilities is one in which $\hat{\varepsilon} \gg \varepsilon$, because accusing an innocent user of forgery is much worse than allowing a pirate to act undetected. Since pirates tend to repeat their actions, the probability they have for acting undetected for a long period of time is slim. For example, while $\varepsilon$ would be typically set to a small value such as $\varepsilon = 10^{-3}$ or $\varepsilon = 10^{-4}$, $\hat{\varepsilon}$ could be set to $\hat{\varepsilon} = \frac{1}{2}$ or even larger, because the expected number of piracy actions until one pirate is finally traced would not exceed $\frac{1}{1-\hat{\varepsilon}}$. For this reason, we decouple the two error probabilities in Theorems 1.1 and 1.2. This decoupling allows us to further decrease the value of $d_m$.

We keep denoting the bound on the error probability in Theorem 1.1 by $\varepsilon$, while the bound on the error probability in Theorem 1.2 will be denoted by $\hat{\varepsilon}$. Finally, we set $\eta = \log_{\varepsilon}(\hat{\varepsilon})$ so that $\hat{\varepsilon} = \varepsilon^{\eta}$.

**Example.** Assume that $n = 10^6$ and it is desired to achieve $(10^{-3}, 3/4)$-security (namely, the probability of accusing any innocent user is bounded by $10^{-3}$ while the expected number of piracy acts until the scheme traces a true pirate is 4). Then we set in Theorem 1.1 $\varepsilon = 10^{-3}/n = 10^{-9}$, while in Theorem 1.2 we take the bound on the error probability to be $\hat{\varepsilon} = 3/4$. In this case, $\eta = \frac{1}{9}\log_{10}\frac{4}{3} \approx 0.014$.

The paper is organized as follows. In Section 2 we carry out the above described analysis of the proofs of Theorems 1.1 and 1.2. Given $n$, $c$, $\varepsilon$ and $\eta$, we determine the domain $\Omega = \{(d_m, d_t, d_z) \in$

$\mathbb{R}^3\}$ of all triplets $(d_m, d_t, d_z)$ for which both Theorems 1.1 and 1.2 still hold (where the error probability in Theorem 1.2 is replaced with $\hat{\varepsilon} = \varepsilon^\eta$). Then, we find the minimal $d_m$ in $\Omega$. It turns out that the domain $\Omega$ depends only on $c$ and $\eta$. This analysis enables us to decrease $d_m$ by a factor of approximately 4. In Section 3 we simulate the Tardos' fingerprinting scheme and show that in practice one can use much smaller values of $d_m$ than what is allowed by the theory, even when the pirate exercises his best strategy. Specifically, we show that in practice one may usually take $d_m < 8$ (as opposed to $d_m = 100$ in the original Tardos scheme). This offers a significant improvement in the codeword length.

# 2 Analysis of Tardos' Scheme

In the following subsections we will establish the main properties of the code. In Section 2.2 we derive the requirements that the parameters $d_m$, $d_t$ and $d_z$ need to satisfy so that, with almost certainty, no innocent user is accused (Theorem 1.1). Here, two new parameters join the optimization game – $d_\alpha$ and $r$. The bottom line of Section 2.2 is two requirements that the five parameters $(d_m, d_t, d_z, d_\alpha, r)$ need to satisfy so that Theorem 1.1 holds. (The values that were set by Tardos for those parameters satisfy those two requirements.)

Then, in Section 2.3, we derive the requirements that those five parameters need to satisfy so that at least one of the pirates is accused by the algorithm with a given threshold probability (Theorem 1.2). Here, two new parameters enter the game – $s$ and $g$. The final conclusion of Section 2.3 is two additional requirements that the seven parameters $(d_m, d_t, d_z, d_\alpha, r, s, g)$ need to satisfy so that Theorem 1.2 holds.

Those four requirements define a domain of parameters $(d_m, d_t, d_z, d_\alpha, r, s, g)$ in $\mathbb{R}^7$. The projection of that domain into $\mathbb{R}_{d_m} \times \mathbb{R}_{d_t} \times \mathbb{R}_{d_z}$ gives the domain $\Omega$ of all triplets $(d_m, d_t, d_z)$ for which the resulting scheme is $(\varepsilon, \hat{\varepsilon})$-secure. (The four other parameters are used only in the proofs and have no manifestation in the actual scheme.) We analyze those requirements in Section 2.4 in order to find a minimal $d_m$ – the constant that determines the length of the codewords – in that domain.

But first, we derive in Section 2.1 some simple inequalities that we shall need later on.

## 2.1 Quadratic upper bounds on the exponential function

**Lemma 2.1** *For every $r > \frac{1}{2}$, there exists $h(r) > 0$ such that*

$$e^x \leq 1 + x + rx^2, \quad \forall x \leq h(r)\,. \tag{8}$$

*Furthermore, the function $h(r)$ is monotonically increasing from 0 to $\infty$ for $\frac{1}{2} < r < \infty$.*

**Proof.** Let $y(x) = 1 + x + rx^2 - e^x$. Differentiating with respect to $x$ we get that $y' = 1 + 2rx - e^x$and $y'' = 2r - e^x$. The latter function is monotonically decreasing. Additionally, it is positive when $x < \ln(2r)$ and negative when $x > \ln(2r)$. Hence, $x = \ln(2r) > 0$ is a global maximum of $y'$. Since $y'(0) = 0$, we infer that $y'(\ln(2r)) > 0$. Moreover, as $\lim_{x\to\infty} y' = -\infty$, there exists a point $f(r) > \ln(2r)$ for which $y'(f(r)) = 0$. This point marks the global maximum of $y$ for $x > 0$. As $y(0) = 0$, we conclude that $y(f(r)) > 0$. Additionally, since $\lim_{x\to\infty} y = -\infty$, there exists a point $h(r) > f(r)$ for which $y(h(r)) = 0$.

To conclude, for every $r > \frac{1}{2}$ there exists a point $h(r) > 0$ such that $y(0) = y(h(r)) = 0$, $y > 0$ for all $0 < x < h(r)$ and $y < 0$ for $x > h(r)$. Moreover, the function $h(r)$ is monotonically increasing. Indeed, denote $y(x; r) = 1 + x + rx^2 - e^x$. Then if $r_2 > r_1 > \frac{1}{2}$, we have $y(x; r_2) > y(x; r_1)$ for every $x > 0$. Therefore $y(h(r_1), r_2) > y(h(r_1), r_1) = 0$. Since $y(x, r_2) > 0$ only when $x < h(r_2)$ we conclude that $h(r_2) > h(r_1)$. Hence, $h(r)$ is monotonically increasing. Furthermore, it is easy to see that $\lim_{r\to\frac{1}{2}^+} h(r) = 0$ and $\lim_{r\to\infty} h(r) = \infty$, since for every fixed $x > 0$ there exists $r$ sufficiently large such that $y(x; r) > 0$. □

As a consequence of Lemma 2.1, the function $h : (\frac{1}{2}, \infty) \to (0, \infty)$ has an inverse $h^{-1}$. Denoting $s = h(r)$ and $r = h^{-1}(s)$, we get another version of inequality (8):

$$e^x \leq 1 + x + h^{-1}(s)x^2, \quad \forall x \leq s\,. \tag{9}$$

This version is explicit since by substituting in the inequality the value $x = s$, where it holds with equality, we get that

$$h^{-1}(s) = \frac{e^s - 1 - s}{s^2}\,. \tag{10}$$

## 2.2 Why the innocent is not accused

**Proof of Theorem 1.1.** Let $y$ be the pirate codeword and let $u_i \in U \setminus T$ be an innocent user. Letting $V_{i,j}$ be as in (5), we denote, for simplicity, $V_j = V_{i,j}$. User $u_i$ will be accused if $S = \sum_{j=1}^{m} y_j V_j = \sum_{j:y_j=1} V_j > z$. We proceed to derive conditions on the scheme's parameters so that $\Pr[S > z] < \varepsilon$.

Let $\alpha = \frac{1}{d_\alpha c}$ for some constant $d_\alpha$. Tardos set the value of this constant to $d_\alpha = 10$. However, as we aim to optimize the selection of constants, we consider $d_\alpha$ as a parameter of an undetermined value, and keep track of the requirements that it needs to satisfy (together with the other parameters – $d_m$, $d_t$ and $d_z$). Consider the expected value $E\left[e^{\alpha S}\right]$. Using the independence of the random variables $V_j$ we have

$$E\left[e^{\alpha S}\right] = E\left[\prod_{j:y_j=1} e^{\alpha V_j}\right] = \prod_{j:y_j=1} E\left[e^{\alpha V_j}\right].$$

Next, Tardos used the fact that

$$1 + x \leq e^x \leq 1 + x + x^2,$$

where the second inequality holds for $x < 1.7$. Here, we take that approach one step further and use the bound

$$1 + x \leq e^x \leq 1 + x + rx^2,$$

for some $r > \frac{1}{2}$, where, in view of Lemma 2.1, the second inequality holds for all $x < h(r)$. Applying that bound for $e^{\alpha V_j}$ we get:

$$1 + \alpha V_j \leq e^{\alpha V_j} \leq 1 + \alpha V_j + r(\alpha V_j)^2, \tag{11}$$

provided that $\alpha V_j < h(r)$. Since $p_j = \sin^2 r_j \geq \sin^2 t' = t$, we infer that

$$V_j \leq \sqrt{\frac{1-p_j}{p_j}} \leq \sqrt{\frac{1-t}{t}} < \sqrt{\frac{1}{t}} = \sqrt{d_t c}\,.$$

Consequently, $\alpha V_j \leq \frac{1}{d_\alpha c}\sqrt{d_t c} = \frac{\sqrt{d_t}}{d_\alpha \sqrt{c}}$. Hence, (11) holds if $\frac{\sqrt{d_t}}{d_\alpha \sqrt{c}} < h(r)$, or, after rearrangement, if the following requirement is satisfied:

| **Requirement 1:** $d_\alpha > \frac{\sqrt{d_t}}{h(r)\sqrt{c}}$. |
|---|

We note that the values that were set by Tardos, $d_t = 300$, $d_\alpha = 10$, and $h(r = 1) = 1.7$, indeed satisfy this requirement for all $c \geq 2$.

Next, we observe that

$$E[V_j] = 0 \text{ and } E[V_j^2] = 1. \tag{12}$$

Indeed,

$$E[V_j] = p_j \cdot \sqrt{\frac{1-p_j}{p_j}} + (1-p_j) \cdot \left(-\sqrt{\frac{p_j}{1-p_j}}\right) = 0,$$

and

$$E\left[V_j^2\right] = p_j \cdot \frac{1-p_j}{p_j} + (1-p_j) \cdot \frac{p_j}{1-p_j} = 1.$$

Therefore, by (11) and (12),

$$E\left[e^{\alpha V_j}\right] \leq E\left[1 + \alpha V_j + r\alpha^2 V_j^2\right] = 1 + \alpha E[V_j] + r\alpha^2 E\left[V_j^2\right] = 1 + r\alpha^2 \leq e^{r\alpha^2},$$

and

$$E\left[e^{\alpha S}\right] = \prod_{j:y_j=1} E\left[e^{\alpha V_j}\right] \leq \left(e^{r\alpha^2}\right)^{|\{j:y_j=1\}|} \leq e^{r\alpha^2 m}.$$

By Markov's inequality, if $Y$ is a nonnegative random variable, then for all $t > 0$, $\Pr[Y \geq t] \leq \frac{E[Y]}{t}$. Applying this inequality to $e^{\alpha S}$ we infer that

$$\Pr[S > z] = \Pr\left[e^{\alpha S} > e^{\alpha z}\right] \leq \frac{E\left[e^{\alpha S}\right]}{e^{\alpha z}} \leq e^{r\alpha^2 m - \alpha z}.$$

We would like to show that $e^{r\alpha^2 m - \alpha z} \leq e^{-k} \leq \varepsilon$. This will be true if

$$r\alpha^2 m - \alpha z \leq -k,$$

or

$$\frac{r d_m c^2 k}{d_\alpha^2 c^2} - \frac{d_z c k}{d_\alpha c} = k\left(\frac{r d_m}{d_\alpha^2} - \frac{d_z}{d_\alpha}\right) \leq -k.$$

This inequality holds when the following requirement is satisfied:

| **Requirement 2:** $\frac{d_z}{d_\alpha} - \frac{r d_m}{d_\alpha^2} \geq 1$. |
|---|

Note that if we pick $d_z = 20$, $d_m = 100$, $d_\alpha = 10$ and $r = 1$, as Tardos did, we get $\frac{20}{10} - \frac{100}{100} = 1$.

To summarize, in the case where both Requirements 1 and 2 are fulfilled, we get $\Pr[S > z] \leq \varepsilon$ as needed. □

### 2.3 Why some guilty is accused

**Proof of Theorem 1.2.** We assume without loss of generality that $T = U$ and that $n = c$ since the codewords of the users outside $T$ are irrelevant. Let $X$ be the $n \times m$ matrix representing the codebook (namely, the $i$th row of $X$ is $P(u_i)$). For simplicity, we introduce $q_j = \sqrt{\frac{1-p_j}{p_j}}$, and recall that by (5),

$$V_{i,j} = \begin{cases} q_j & \text{if } P(u_i)_j = 1 \\ -\frac{1}{q_j} & \text{if } P(u_i)_j = 0 \end{cases}.$$

Let $y$ be the pirate codeword and let $S_i = \sum_{j=1}^m y_j V_{i,j}$ for $u_i \in T$. Define

$$S = \sum_{i=1}^c S_i = \sum_{j=1}^m y_j \left(x_j q_j - \frac{n - x_j}{q_j}\right), \tag{13}$$

where $x_j = \sum_{i=1}^{n} P(u_i)_j$ denotes the number of ones in column $j$ of $X$. Recall that $u_i \in T$ is accused if $S_i \geq z$. Thus, if $S \geq nz$, at least one of the pirates in $T$ must be accused. Hence, it is sufficient to bound the probability that $S < nz$ because

$$\Pr\left[T \cap \sigma\left(y\right) = \emptyset\right] \leq \Pr\left[S < nz\right].$$

Let $\beta = \frac{s\sqrt{t}}{c}$, where $s$ is a parameter that will be determined later. Tardos used $s = 1$. Using the rules of the second phase of the code generation, and letting $\overline{p} = (p_1, \ldots, p_m)$, we have

$$\begin{aligned} E_{\overline{p},X}\left[e^{-\beta S}\right] &= E_{\overline{p}}\left[\sum_{X}\left(e^{-\beta S}\prod_{j=1}^{m}\left(p_j^{x_j}\left(1-p_j\right)^{n-x_j}\right)\right)\right] = \\ &= \sum_{X} E_{\overline{p}}\left[e^{-\beta S}\prod_{j=1}^{m}\left(p_j^{x_j}\left(1-p_j\right)^{n-x_j}\right)\right]; \end{aligned}$$

here, $E_{\overline{p},X}$ stands for the expectation with respect to all random choices of the $m$ probabilities in $\overline{p}$ and all $nm$ random choices of the entries of $X$, while $E_{\overline{p}}$ stands for the expectation with respect only to selections of $\overline{p}$. The summation is for all $n \times m$ binary matrices $X$. Using equation (13) we have

$$E_{\overline{p},X}\left[e^{-\beta S}\right] = \sum_{X}\left[\prod_{j=1}^{m} E_{p_j}\left(p_j^{x_j}\left(1-p_j\right)^{n-x_j} e^{-\beta y_j\left(x_j q_j - \frac{n-x_j}{q_j}\right)}\right)\right].$$

Here $x_j$ and $y \in \langle P\left(T\right)\rangle$ are determined by $X$, while $q_j = \sqrt{\frac{1-p_j}{p_j}}$ is determined by $\overline{p}$. Notice that for a fixed matrix $X$, term $j$ of the product depends solely on $p_j$, whence these terms are independent. As each $p_j$ is identically distributed we write hereinafter $p$ instead, and $q$ instead of $q_j$.

Recall that each $y_j$ is either 0 or 1. Furthermore, by the generation assumption, if $x_j = 0$ then $y_j = 0$ and if $x_j = n$ then $y_j = 1$. Thus we have

$$E_{\overline{p},X}\left[e^{-\beta S}\right] \leq \sum_{X}\prod_{j=1}^{m}\max{}^{*}\left(N_{0,j}, N_{1,j}\right),$$

where

$$N_{0,j} = E_p\left[p^{x_j}\left(1-p\right)^{n-x_j}\right],$$

$$N_{1,j} = E_p\left[p^{x_j}\left(1-p\right)^{n-x_j} e^{-\beta\left(x_j q - \frac{n-x_j}{q}\right)}\right],$$

and

$$\max{}^{*} = \begin{cases} N_{0,j} & \text{if } x_j = 0 \\ N_{1,j} & \text{if } x_j = n \\ \max(N_{0,j}, N_{1,j}) & \text{otherwise} \end{cases}.$$

As the $j$th term in the product depends only on $x_j$ and the summation is for all 0-1 matrices $X$, we may switch the summation and the product to get

$$E_{\overline{p},X}\left[e^{-\beta S}\right] \leq \prod_{j=1}^{m}\sum_{x_j=0}^{n}\binom{n}{x_j}\max{}^{*}\left(N_{0,j}, N_{1,j}\right).$$

Hence,

$$E_{\overline{p},X}\left[e^{-\beta S}\right] \leq \left(\sum_{x=0}^{n}\binom{n}{x}M_x\right)^{m}, \tag{14}$$

where

$$M_0 = E_{0,0}, \quad M_n = E_{1,n}, \quad \text{and } M_x = \max\left(E_{0,x}, E_{1,x}\right) \text{ for } 1 \le x \le n-1, \tag{15}$$

and

$$E_{0,x} := E_p\left[p^x\left(1-p\right)^{n-x}\right], \tag{16}$$

$$E_{1,x} := E_p\left[p^x\left(1-p\right)^{n-x} e^{-\beta\left(xq-\frac{n-x}{q}\right)}\right], \tag{17}$$

for all $0 \le x \le n$. Since $p \le 1-t$ and $\beta = \frac{s\sqrt{t}}{c}$, the value in the exponent in (17) may be bounded as follows:

$$-\beta\left(xq - \frac{n-x}{q}\right) \le \frac{\beta n}{q} = \beta n\sqrt{\frac{p}{1-p}} \le \frac{\beta n}{\sqrt{1-p}} \le \frac{s\sqrt{t}\cdot n}{c\cdot\sqrt{t}} = \frac{sn}{c} = s. \tag{18}$$

With this, we proceed to bound the exponent in (17). Tardos used the value $s=1$ and then used the bound $e^u \le 1+u+u^2$, that holds for $u < 1.7$. However, since inequality (18) guarantees that the value in the exponent in (17) does not exceed $s$, we may use equation (9) as a tighter bound to get

$$E_{1,x} \le E_p\left[p^x\left(1-p\right)^{n-x}\left(1-\beta\left(xq-\frac{n-x}{q}\right)+h^{-1}(s)\beta^2\left(xq-\frac{n-x}{q}\right)^2\right)\right].$$

Using the notation $E_{0,x}$, that was defined in (16),

$$E_{2,x} := E_p\left[p^x\left(1-p\right)^{n-x}\left(xq-\frac{n-x}{q}\right)\right],$$

and

$$E_{3,x} := E_p\left[p^x\left(1-p\right)^{n-x}\left(xq-\frac{n-x}{q}\right)^2\right] \ge 0\,,$$

we have

$$E_{1,x} \le E_{0,x} - \beta E_{2,x} + h^{-1}(s)\beta^2 E_{3,x}\,. \tag{19}$$

The term $E_{2,x}$ is the most important one. Recall that $p = \sin^2 r$ with a uniform random $r \in \left[t', \frac{\pi}{2}-t'\right]$, where $\sin^2 t' = t$. Hence,

$$1-p = 1-\sin^2 r = \cos^2 r,$$

and

$$q = \sqrt{\frac{1-p}{p}} = \sqrt{\frac{\cos^2 r}{\sin^2 r}} = \frac{\cos r}{\sin r} = \cot r.$$

Next, we can calculate $E_{2,x}$ using the following integral

$$E_{2,x} = \frac{\int_{t'}^{\frac{\pi}{2}-t'} \sin^{2x} r\cos^{2n-2x} r\left(x\cot r - (n-x)\tan r\right)dr}{\frac{\pi}{2}-2t'}.$$

The primitive function of the integrand is

$$f\left(r\right) = \frac{1}{2}\sin^{2x} r\cos^{2n-2x} r.$$

Therefore,

$$\begin{aligned} E_{2,x} &= \frac{f\left(\frac{\pi}{2}-t'\right)-f\left(t'\right)}{\frac{\pi}{2}-2t'} = \\ &= \frac{\sin^{2x}\left(\frac{\pi}{2}-t'\right)\cos^{2n-2x}\left(\frac{\pi}{2}-t'\right)-\sin^{2x}\left(t'\right)\cos^{2n-2x}\left(t'\right)}{\pi-4t'} = \\ &= \frac{\cos^{2x}\left(t'\right)\sin^{2n-2x}\left(t'\right)-\sin^{2x}\left(t'\right)\cos^{2n-2x}\left(t'\right)}{\pi-4t'} = \\ &= \frac{(1-t)^x t^{n-x}-t^x(1-t)^{n-x}}{\pi-4t'}. \end{aligned} \tag{20}$$

Hence,

$$E_{2,x} \geq -\frac{t^x(1-t)^{n-x}}{\pi-4t'} < 0. \tag{21}$$

Going back to (15), we use equations (19) and (21) to infer that

$$\begin{aligned} M_x &= \max\left(E_{0,x},E_{1,x}\right) \leq \max\left(E_{0,x},E_{0,x}-\beta E_{2,x}+h^{-1}(s)\beta^2 E_{3,x}\right) \leq \\ &\leq \max\left(E_{0,x},E_{0,x}+\beta\cdot\frac{t^x(1-t)^{n-x}}{\pi-4t'}+h^{-1}(s)\beta^2 E_{3,x}\right) \leq \\ &\leq E_{0,x}+\beta\cdot\frac{t^x(1-t)^{n-x}}{\pi-4t'}+h^{-1}(s)\beta^2 E_{3,x}\,, \end{aligned}$$

for all $1 \leq x \leq n-1$. We also have $M_0 = E_{0,0}$. Finally, as by (20) we have $E_{2,n} = \frac{(1-t)^n - t^n}{\pi-4t'}$, we conclude, in view of (19), that

$$M_n = E_{1,n} \leq E_{0,n} - \beta\frac{(1-t)^n-t^n}{\pi-4t'} + h^{-1}(s)\beta^2 E_{3,n}.$$

Next, we use the above estimates on $M_x$, $0 \leq x \leq n$, in order to bound the sum in equation (14):

$$\begin{aligned} \sum_{x=0}^{n}\binom{n}{x}M_x &\leq M_0+M_n+\sum_{x=1}^{n-1}\binom{n}{x}M_x \leq \\ &\leq E_{0,0}+E_{0,n}-\beta\frac{(1-t)^n-t^n}{\pi-4t'}+h^{-1}(s)\beta^2E_{3,n}+ \\ &+ \sum_{x=1}^{n-1}\binom{n}{x}E_{0,x}+\frac{\beta}{\pi-4t'}\sum_{x=1}^{n-1}\binom{n}{x}t^x(1-t)^{n-x}+h^{-1}(s)\beta^2\sum_{x=1}^{n-1}\binom{n}{x}E_{3,x} \leq \\ &\leq \sum_{x=0}^{n}\binom{n}{x}E_{0,x}+h^{-1}(s)\beta^2\sum_{x=1}^{n}\binom{n}{x}E_{3,x}+ \\ &+ \frac{\beta\left(\left(\sum_{x=1}^{n-1}\binom{n}{x}t^x(1-t)^{n-x}\right)-(1-t)^n+t^n\right)}{\pi-4t'} \end{aligned}$$

Rearranging the terms on the right hand side we get that

$$\sum_{x=0}^{n}\binom{n}{x}M_x \leq \sum_{x=0}^{n}\binom{n}{x}E_{0,x}-\frac{\beta\left((1-t)^n-\sum_{x=1}^{n}\binom{n}{x}t^x(1-t)^{n-x}\right)}{\pi-4t'}+h^{-1}(s)\beta^2\sum_{x=1}^{n}\binom{n}{x}E_{3,x}. \tag{22}$$

We proceed to bound separately each of the addends on the right hand side of (22). First, using the binomial theorem, we have

$$\sum_{x=0}^{n}\binom{n}{x}E_{0,x} = \sum_{x=0}^{n}\binom{n}{x}E_p\left[p^x(1-p)^{n-x}\right] = E_p\left[\sum_{x=0}^{n}\binom{n}{x}p^x(1-p)^{n-x}\right] = E_p\left[(p+1-p)^n\right] = 1.$$

Next, using the same tool we get that

$$-\sum_{x=1}^{n}\binom{n}{x}t^x\,(1-t)^{n-x}=(1-t)^n-\sum_{x=0}^{n}\binom{n}{x}t^x\,(1-t)^{n-x}=(1-t)^n-(t+1-t)^n=(1-t)^n-1.$$

Consequently,

$$(1-t)^n-\sum_{x=1}^{n}\binom{n}{x}t^x\,(1-t)^{n-x}=2\,(1-t)^n-1\,.$$

As $0\le t\le 1$, we get that

$$(1-t)^n=1-nt+O\left(t^2\right)\ge 1-nt,$$

and

$$2\,(1-t)^n-1\ge 2-2nt-1=1-2nt.$$

Finally, we estimate the third and last sum on the right hand side of (22):

$$\begin{aligned}\sum_{x=1}^{n}\binom{n}{x}E_{3,x} &= \sum_{x=0}^{n}\binom{n}{x}E_{3,x}-E_{3,0}=\sum_{x=0}^{n}\binom{n}{x}E_p\left[p^x\,(1-p)^{n-x}\left(xq-\frac{n-x}{q}\right)^2\right]-E_{3,0}=\\ &= E_p\left[\sum_{x=0}^{n}\binom{n}{x}p^x\,(1-p)^{n-x}\left(xq-\frac{n-x}{q}\right)^2\right]-E_{3,0}.\end{aligned}$$

In order to proceed and estimate this last sum, consider a fixed $0<p<1$ and then define the sequence of independent and identically distributed random variables $H_i$, $1\le i\le n$, with the following distribution:

$$H_i=\begin{cases}q & \text{with probability } p\,,\\ -\frac{1}{q} & \text{with probability } 1-p\,.\end{cases}$$

The expectation and variance of $H_i$ are:

$$E\left[H_i\right]=pq+(1-p)\left(-\frac{1}{q}\right)=p\cdot\sqrt{\frac{1-p}{p}}-(1-p)\sqrt{\frac{p}{1-p}}=0\,;$$

$$E\left[H_i^2\right]=p\cdot q^2+(1-p)\left(-\frac{1}{q}\right)^2=(1-p)+p=1\,.$$

As $H_i$ are independent, we conclude, in view of the above equalities, that

$$E\left[\left(\sum_{i=1}^{n}H_i\right)^2\right]=\sum_{i=1}^{n}E\left[H_i^2\right]+\sum_{1\le i\ne j\le n}E\left[H_iH_j\right]=\sum_{i=1}^{n}1+\sum_{1\le i\ne j\le n}E[H_i]E[H_j]=n.$$

However, examining this expectancy we discover that

$$E\left[\left(\sum_{i=1}^{n}H_i\right)^2\right]=\sum_{x=0}^{n}\binom{n}{x}p^x\,(1-p)^{n-x}\left(x\cdot q-(n-x)\,\frac{1}{q}\right)^2.$$

As this is the same expression that appears in the bound on $\sum_{x=1}^{n}\binom{n}{x}E_{3,x}$, we conclude that:

$$\sum_{x=1}^{n}\binom{n}{x}E_{3,n}=E_p\left[n\right]-E_{3,0}=n-E_p\left[(1-p)^n\left(-\frac{n}{q}\right)^2\right]\le n.$$

We may now return to equation (22). As we derived bounds on all the addends on its right hand side, we find out that:

$$\sum_{x=0}^{n}\binom{n}{x}M_x < 1-\beta\frac{1-2nt}{\pi-4t'}+h^{-1}(s)\beta^2 n.$$

We would like to arrive at the estimate

$$\sum_{x=0}^{n}\binom{n}{x}M_x < 1-g\beta\,, \tag{23}$$

for some parameter $g$ that is yet to be determined. (Tardos picked $g=\frac{1}{4}$.) The inequality holds when

$$1-\beta\frac{1-2nt}{\pi-4t'}+h^{-1}(s)\beta^2 n < 1-g\beta\,,$$

or, after substituting $\beta=\frac{s\sqrt{t}}{c}$, when

$$\frac{s\sqrt{t}}{c}\cdot\frac{1-2nt}{\pi-4t'}-h^{-1}(s)\frac{s^2tn}{c^2} > \frac{gs\sqrt{t}}{c}\,.$$

Dividing by $\frac{s\sqrt{t}}{c}$ we arrive at the following inequality

$$\frac{1-2nt}{\pi-4t'}-\frac{h^{-1}(s)s\sqrt{t}n}{c} > g\,.$$

Since $n=c$ and $t=\frac{1}{d_t c}$ we can lower bound the left hand side of the inequality above as follows:

$$\frac{1-2nt}{\pi-4t'}-\frac{h^{-1}(s)s\sqrt{t}n}{c}=\frac{1-2c\frac{1}{d_t c}}{\pi-4t'}-\frac{h^{-1}(s)s}{\sqrt{d_t c}}=\frac{1-\frac{2}{d_t}}{\pi-4t'}-\frac{h^{-1}(s)s}{\sqrt{d_t c}}>\frac{1-\frac{2}{d_t}}{\pi}-\frac{h^{-1}(s)s}{\sqrt{d_t c}}.$$

This leads us to the third requirement:

**Requirement 3:** $\frac{1-\frac{2}{d_t}}{\pi}-\frac{h^{-1}(s)s}{\sqrt{d_t c}} > g.$

Note that if we pick $s=1$, $g=\frac{1}{4}$ and $d_t=300$, as Tardos did, we get

$$\frac{1-\frac{2}{300}}{\pi}-\frac{h^{-1}(1)}{\sqrt{300c}} > 0.316-\frac{0.719}{\sqrt{300\cdot 1}}=0.275>\frac{1}{4}\,.$$

We further notice that Requirement 3 can hold only when $g<\frac{1}{\pi}$.

Finally, using equations (14) and (23), we may bound the expectancy of $e^{-\beta S}$ as follows

$$E_{\overline{p},X}\left[e^{-\beta S}\right]\le\left(\sum_{x=0}^{n}\binom{n}{x}M_x\right)^m<(1-g\beta)^m<e^{-g\beta m}.$$

Using Markov's inequality, we get that

$$\Pr\left[S\le nz\right]=\Pr\left[S\le cz\right]=\Pr\left[e^{-\beta S}\ge e^{-\beta cz}\right]\le\frac{e^{-g\beta m}}{e^{-\beta cz}}=e^{-\beta(gm-cz)}.$$

Rearranging the exponent in the last expression yields that:

$$-\beta\left(gm-cz\right)=-\frac{s\sqrt{t}}{c}\left(gd_m c^2k-cd_z ck\right)=-\sqrt{\frac{1}{d_t c}}\cdot\frac{sc^2k}{c}\left(gd_m-d_z\right)=\frac{d_z-gd_m}{\sqrt{d_t}}\cdot s\sqrt{c}k.$$

Since we aim at having $\Pr[S \leq nz] \leq \hat{\varepsilon} = \varepsilon^{\eta}$, we require that $\Pr[S \leq nz] \leq e^{-k\eta}$. This yields the following requirement.

**Requirement 4:** $gd_m - d_z \geq \eta\sqrt{\frac{d_t}{s^2 c}}$.

By setting $d_m = 100$, $d_z = 20$, $d_t = 300$, $s = 1$, and $g = \frac{1}{4}$, like Tardos did in [7], the claim of Theorem 1.2 holds with $\eta = \sqrt{c}/4$. That result was improved by Tardos in [8] to hold with $\eta = c/4$ (as stated here in Theorem 1.2). □

### 2.4 Optimizing the constants

Our goal, in this section, is to examine the domain of parameters $(d_m, d_t, d_z, d_\alpha, r, s, g)$ in $\mathbb{R}^7$ where the four requirements that we introduced in the previous subsections are satisfied, in order to find within that domain the minimal value of $d_m$. Let us begin by a brief review of those parameters: Given a bound $c$ on the size of the coalition of pirates, a maximum tolerated error probability, $\varepsilon$, for falsely accusing any innocent user, and a maximum tolerated error probability, $\hat{\varepsilon}$, for falsely acquitting all pirates, we have the following positive parameters:

- $d_m$ is used in defining the codeword length, $m = d_m c^2 k$, where $k = \lceil \log \frac{1}{\varepsilon} \rceil$.
- $d_z$ is used in defining the threshold for accusation, $z = d_z c k$.
- $d_t$ is used in defining $t = \frac{1}{d_t c}$, a parameter that lower bounds the value of the probabilities $p_j$ that govern the selection of bits in the random codewords. Recall that $d_t \geq 1$.
- $d_\alpha$ is used in defining $\alpha = \frac{1}{d_\alpha c}$, a parameter that plays role in bounding the probability that an innocent user is accused by the algorithm.
- $r$ is used for the bound $e^x < 1 + x + rx^2$. This bound holds for all $x < h(r)$, as described in Lemma 2.1. Recall that $r > \frac{1}{2}$.
- $s > 0$ is used for the definition of $\beta = \frac{s\sqrt{t}}{c}$, a parameter that plays a key role in the proof of Theorem 1.2.
- $g$ is used in inequality (23), that is used later on in bounding the error probability in Theorem 1.2. Requirement 3 dictates that $g < \frac{1}{\pi}$.

The set of requirements that all of those parameters have to satisfy is as follows:

**Requirement 1:** $d_\alpha \geq \frac{\sqrt{d_t}}{h(r)\sqrt{c}}$.
**Requirement 2:** $\frac{d_z}{d_\alpha} - \frac{r d_m}{d_\alpha^2} \geq 1$.
**Requirement 3:** $\frac{1-\frac{2}{d_t}}{\pi} - \frac{h^{-1}(s)s}{\sqrt{d_t c}} \geq g$.
**Requirement 4:** $gd_m - d_z \geq \eta\sqrt{\frac{d_t}{s^2 c}}$.

**Example.** The following parameters values meet Requirements 1 through 4 for any $c \geq 2$ and $\eta = 1$:

$$d_\alpha = 8, \quad d_t = 40, \quad d_z = 15, \quad d_m = 85, \quad r = 0.611, \quad s = 0.757, \quad g = 0.2461\,.$$

(For the above values of $r$ and $s$, $h(r) = 0.571$ and $h^{-1}(s) = 0.6543$.) The scheme with those parameter values improves the scheme proposed by Tardos in two ways:

1. The constant $d_m$ is smaller than the one used by Tardos (100), whence it offers shorter codewords.

2. It applies to all $c \geq 2$, as opposed to Tardos' scheme that applies to all $c \geq 4$. (The analysis that was carried out by Tardos in [7] showed that the scheme works for all $c \geq 16$; the journal version [8], on the other hand, contained a proof that the scheme works for all $c \geq 7$, and Tardos remarks there that the proof may be extended to include all $c \geq 4$.)

We proceed to carry out an analysis that finds the optimal settings of the first four parameters, under the assumption that $r$, $s$ and $g$ are constants satisfying $r > \frac{1}{2}$, $s > 0$, and $0 < g < \frac{1}{\pi}$. Then we shall use numerical optimization procedures to find $r$, $s$ and $g$ that yield a minimal value for $d_m$.

### 2.4.1 Minimizing $d_m$

Our goal is to minimize the codeword length $m = d_m c^2 k$. As Tardos proved that the optimal codeword length is $\Omega\left(c^2 k\right)$, the best that we can hope for is the reduction of $d_m$. We aim at finding herein the minimal $d_m$ for which there exist constants $d_t$, $d_z$, $d_\alpha$, $r > \frac{1}{2}$, $s > 0$, and $0 < g < \frac{1}{\pi}$ such that all seven constants satisfy Requirements 1 through 4.

Since we aim at achieving minimal $d_m$, we conclude, by Requirement 4, that

$$d_m \geq \frac{\eta\sqrt{\frac{d_t}{s^2 c}} + d_z}{g}\,. \tag{24}$$

In other words, the minimal value of $d_m$ is:

$$\hat{d}_m = \frac{\eta\sqrt{\frac{d_t}{s^2 c}} + d_z}{g}\,. \tag{25}$$

We proceed to show that the minimum of the sum $\eta\sqrt{\frac{d_t}{s^2 c}} + d_z$ is obtained when both $d_t$ and $d_z$ are minimized, and then find the minimal values of those two parameters.

### 2.4.2 Minimizing $d_t$

Requirement 3 translates into the following quadratic inequality in $x = \sqrt{\frac{1}{d_t}}$,

$$\frac{2}{\pi} \cdot x^2 + \frac{h^{-1}(s)s}{\sqrt{c}} \cdot x - \left(\frac{1}{\pi} - g\right) \leq 0. \tag{26}$$

Since we are looking to minimize $d_t$, we would like to maximize $x$. The quadratic expression on the left hand side of (26) has two roots,

$$x_\pm = \frac{-\frac{h^{-1}(s)s}{\sqrt{c}} \pm \sqrt{\frac{(h^{-1}(s)s)^2}{c} + \frac{8}{\pi}\left(\frac{1}{\pi} - g\right)}}{\frac{4}{\pi}}\,.$$

Therefore, the maximal value of $x$ that satisfies (26) is $x_+$. Hence, as $d_t = \left(\frac{1}{x}\right)^2$, we infer that the minimal value of $d_t$ that satisfies Requirement 3 is

$$\hat{d}_t = \left(\frac{\frac{4}{\pi}}{\sqrt{\frac{(h^{-1}(s)s)^2}{c} + \frac{8}{\pi}\left(\frac{1}{\pi} - g\right)} - \frac{h^{-1}(s)s}{\sqrt{c}}}\right)^2\,,$$

or after rearrangement,

$$\hat{d}_t = \left( \frac{1}{\frac{2}{\pi} - 2g} \cdot \left( \sqrt{\frac{(h^{-1}(s)s)^2}{c} + \frac{8}{\pi}\left(\frac{1}{\pi} - g\right)} + \frac{h^{-1}(s)s}{\sqrt{c}} \right) \right)^2 . \tag{27}$$

### 2.4.3 Minimizing $d_z$

According to Requirement 2,

$$\frac{d_z}{d_\alpha} - \frac{r d_m}{d_\alpha^2} \geq 1 .$$

Substituting into that inequality the optimal value of $d_m$, i.e. $\hat{d}_m = \frac{\eta\sqrt{\frac{d_t}{s^2 c}} + d_z}{g}$, we get

$$\frac{d_z}{d_\alpha} - \frac{r d_z + r\eta\sqrt{\frac{d_t}{s^2 c}}}{g d_\alpha^2} = \frac{g d_\alpha d_z - r d_z - r\eta\sqrt{\frac{d_t}{s^2 c}}}{g d_\alpha^2} = \frac{(g d_\alpha - r)\, d_z - r\eta\sqrt{\frac{d_t}{s^2 c}}}{g d_\alpha^2} \geq 1,$$

or, after rearrangement,

$$(g d_\alpha - r) d_z \geq g d_\alpha^2 + r\eta\sqrt{\frac{d_t}{s^2 c}} .$$

First, we conclude that $(g d_\alpha - r)$ must be positive. Hence, assuming hereinafter that

$$d_\alpha > \frac{r}{g} , \tag{28}$$

we find out that

$$d_z \geq \frac{g d_\alpha^2 + r\eta\sqrt{\frac{d_t}{s^2 c}}}{g d_\alpha - r} . \tag{29}$$

We observe that the lower bound on $d_z$ decreases when $d_t$ decreases. Hence, in order to minimize the sum in (25), namely $\hat{d}_m = \frac{\eta\sqrt{\frac{d_t}{s^2 c}} + d_z}{g}$, we can first minimize $d_t$ and then minimize $d_z$. Therefore, we replace $d_t$ with $\hat{d}_t$ in (29) to obtain the minimal value of $d_z$,

$$\hat{d}_z = \frac{g d_\alpha^2 + r\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}{g d_\alpha - r} . \tag{30}$$

### 2.4.4 Optimizing $d_\alpha$

So far we saw that in order to minimize $d_m$ (for given $c$ and $\eta$ and a fixed selection of $r$, $s$ and $g$), we have to set $d_t = \hat{d}_t$ as in (27), set $d_z = \hat{d}_z$ as in (30), and then set $d_m = \hat{d}_m = \frac{\eta\sqrt{\frac{\hat{d}_t}{s^2 c}} + \hat{d}_z}{g}$. The remaining undetermined parameter is $d_\alpha$. That parameter effects the value of $\hat{d}_z$. We proceed to find the value of $d_\alpha$ for which $\hat{d}_z$, and consequently $\hat{d}_m$, are minimized.

$$\begin{aligned}
\hat{d}_z &= f(d_\alpha) = \frac{g d_\alpha^2 + r\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}{g d_\alpha - r} = \frac{d_\alpha^2 + \frac{r}{g}\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}{d_\alpha - \frac{r}{g}} = \\
&= \frac{d_\alpha^2 - \left(\frac{r}{g}\right)^2}{d_\alpha - \frac{r}{g}} + \frac{\left(\frac{r}{g}\right)^2 + \frac{r}{g}\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}{d_\alpha - \frac{r}{g}} = \\
&= d_\alpha + \frac{r}{g} + \frac{\left(\frac{r}{g}\right)^2 + \frac{r}{g}\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}{d_\alpha - \frac{r}{g}} .
\end{aligned}$$

Differentiating with respect to $d_\alpha$, we find that

$$f'(d_\alpha) = 1 - \frac{\left(\frac{r}{g}\right)^2 + \frac{r}{g}\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}{\left(d_\alpha - \frac{r}{g}\right)^2}.$$

We see that $f'(d_\alpha)$ is negative, namely, $f(d_\alpha)$ is decreasing, in the interval

$$\left(\frac{r}{g} - \sqrt{\left(\frac{r}{g}\right)^2 + \frac{r}{g}\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}\ ,\ \frac{r}{g} + \sqrt{\left(\frac{r}{g}\right)^2 + \frac{r}{g}\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}\right),$$

and increasing elsewhere. Therefore, the global minimum of $f(d_\alpha)$ in the domain $d_\alpha > 0$ is obtained at

$$d_\alpha = \frac{r}{g} + \sqrt{\left(\frac{r}{g}\right)^2 + \frac{r}{g}\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}.$$

However, the parameter $d_\alpha$ is restricted by two constraints. First, it is restricted by (28), namely, $d_\alpha > \frac{r}{g}$. Moreover, it is also restricted by Requirement 1,

$$d_\alpha > \frac{\sqrt{\hat{d}_t}}{h(r)\sqrt{c}}.$$

We conclude that within the constraints that $d_\alpha$ has to satisfy, the minimum of $f(d_\alpha)$ is obtained at

$$\hat{d}_\alpha = \max\left\{\frac{\sqrt{\hat{d}_t}}{h(r)\sqrt{c}}\ ,\ \frac{r}{g} + \sqrt{\left(\frac{r}{g}\right)^2 + \frac{r}{g}\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}\right\}. \tag{31}$$

### 2.4.5 Conclusion

We summarize herein the results of our analysis. Given $c$, $\eta = \log_\varepsilon \hat{\varepsilon}$, and constant values $r > \frac{1}{2}$, $s > 0$ and $0 < g < \frac{1}{\pi}$, the best parameter values for $d_m$, $d_t$, $d_z$, and $d_\alpha$ are computed as follows:

1. Set $\hat{d}_t$ according to equation (27), namely,

$$\hat{d}_t = \left(\frac{1}{\frac{2}{\pi} - 2g} \cdot \left(\sqrt{\frac{(h^{-1}(s)s)^2}{c} + \frac{8}{\pi}\left(\frac{1}{\pi} - g\right)} + \frac{h^{-1}(s)s}{\sqrt{c}}\right)\right)^2.$$

2. Set $\hat{d}_\alpha$ according to equation (31), namely,

$$\hat{d}_\alpha = \max\left\{\frac{\sqrt{\hat{d}_t}}{h(r)\sqrt{c}}\ ,\ \frac{r}{g} + \sqrt{\left(\frac{r}{g}\right)^2 + \frac{r}{g}\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}\right\}.$$

3. Set $\hat{d}_z$ according to equation (30), namely,

$$\hat{d}_z = \frac{g\hat{d}_\alpha^2 + r\eta\sqrt{\frac{\hat{d}_t}{s^2 c}}}{g\hat{d}_\alpha - r}.$$

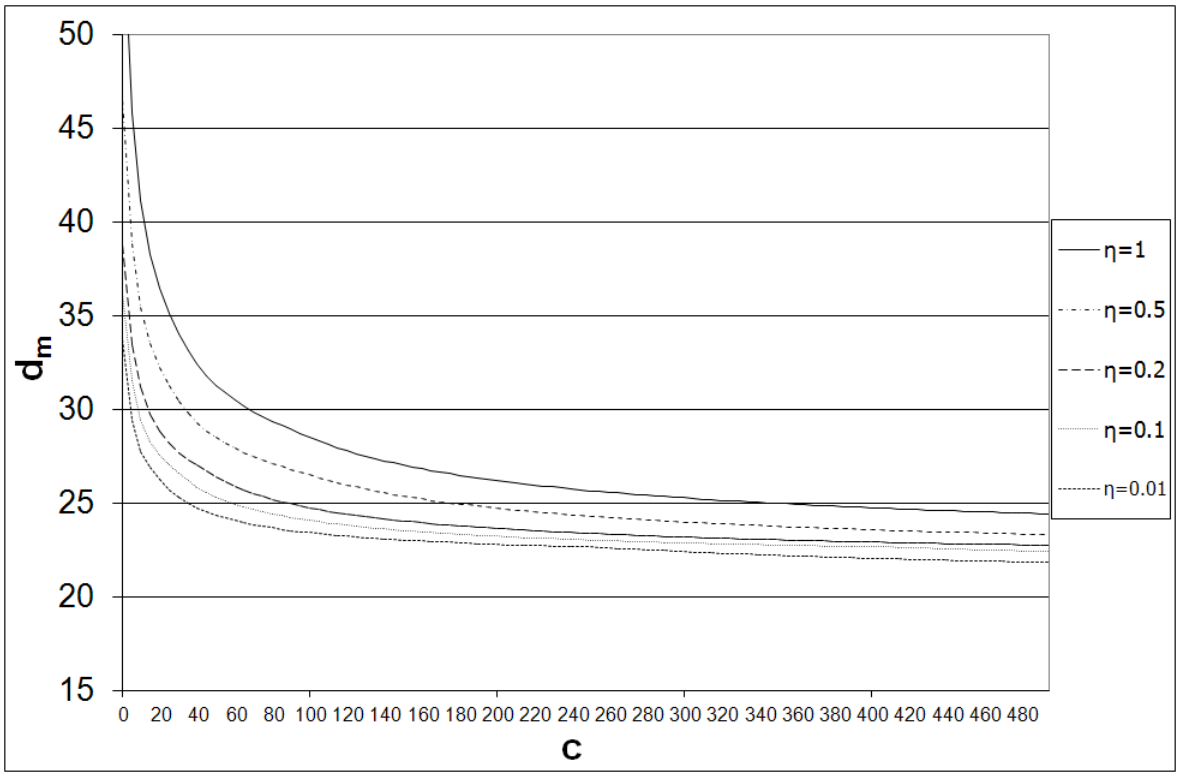


Figure 1: $\hat{d}_m$ versus $c$ for various values of $\eta$.

4. Set $\hat{d}_m$ according to (25),

$$\hat{d}_m = \frac{\eta\sqrt{\frac{\hat{d}_t}{s^2 c}} + \hat{d}_z}{g}\,.$$

In view of the above, given $c$ and $\eta$, the value of $\hat{d}_m$ – the minimal $d_m$ – depends on our selection of $r$, $s$ and $g$. That dependency may be examined by numerical means in order to find the values of $r > \frac{1}{2}$, $s > 0$ and $0 < g < \frac{1}{\pi}$ that minimize $\hat{d}_m$. Figure 1 presents the correlation between $c$ and $\hat{d}_m$ for various values of $\eta$. We note that when $\eta$ decreases, so does $\hat{d}_m$. Hence, the decoupling of $\varepsilon$ and $\hat{\varepsilon} = \varepsilon^\eta$ by allowing $\hat{\varepsilon}$ to be much larger than $\varepsilon$ does help to further reduce the codeword length. We see that our analysis enables a reduction of the codeword length by a factor of approximately 4.

# 3 Simulation

In this section we describe simulations that we ran in order to find out the essential efficiency of Tardos scheme. The simulator was written in Java and ran on a desktop machine.

## 3.1 The setup

The input to each experiment is composed of the following parameters:

1. $n$ – the number of users.
2. $\varepsilon$ – the maximal probability of accusing an innocent user.
3. $\hat{\varepsilon}$ – the maximal probability of not accusing any of the traitors ($\hat{\varepsilon} \geq \varepsilon$).
4. $c$ – the assumed bound on the number of traitors.
5. $d_m$ – the parameter that determines the codeword length, $m = d_m c^2 k$, where hereinafter $k = \lceil \log(n/\varepsilon) \rceil$.
6. $d_t$ – the parameter that determines the probabilities according to which the codeword bits are generated.

Given such an input, we perform $r$ experiments where $r = \lceil 100/\varepsilon \rceil$ at the least. In each experiment we generate $n$ codewords according to the given parameters $d_m$ and $d_t$. The first $c$ codewords are then used in order to generate a pirate codeword $y \in \{0,1\}^m$. We examine several possible strategies that the pirate may adopt in generating the pirate codeword $y$ out of the $c$ codewords that he possesses; we describe those strategies in Section 3.2.

We recall (see Section 1.3.2) that a user with codeword $x \in \{0,1\}^m$ is accused if its accusation score $X = \sum_{j=1}^{m} y_j V_j$ is greater than $z = d_z ck$, where

$$V_j = \begin{cases} \sqrt{\frac{1-p_j}{p_j}} & \text{if } x_j = 1 \\ -\sqrt{\frac{p_j}{1-p_j}} & \text{if } x_j = 0 \end{cases}, \quad 1 \leq j \leq m\,.$$

Namely, the higher $d_z$ is, the less users are accused.

Given the pirate codeword $y$, we compute two threshold values of $d_z$. The first one is the minimal value, $d_z^0$, for which no innocent user is accused. Namely, if we use lower values of $d_z$ at least one innocent user will pass the accusation test. Letting $X_0$ be the maximal accusation score of an innocent user, then

$$d_z^0 = \frac{X_0}{ck}\,.$$

The second one is $d_z^1$ – this is the minimal value of $d_z$ that will result in missing all traitors. Letting $X_1$ be the maximal accusation score of a traitor, then

$$d_z^1 = \frac{X_1}{ck}\,.$$

Let $d_z^0(i)$ and $d_z^1(i)$ denote the two threshold values in experiment number $i$, $1 \leq i \leq r$. Then, for every value of $d_z$ in the interval $\mathrm{Conv}\{d_z^0(i), d_z^1(i) : 1 \leq i \leq r\}$, we define

$$E(d_z, 0) = |\{i : d_z^0(i) > d_z\}| \quad \text{and} \quad E(d_z, 1) = |\{i : d_z^1(i) \leq d_z\}|\,.$$

Namely, $E(d_z, 0)$ is the number of experiments in which we shall have failures of the first kind (accusing an innocent user) assuming that we used the value $d_z$ in the accusation stage. $E(d_z, 1)$,

on the other hand, is the number of experiments in which we shall have failures of the second kind (missing all traitors), assuming that we used $d_z$ in the accusation stage. Clearly, $E(d_z, 0)$ is monotonically non-increasing in $d_z$ while $E(d_z, 1)$ is monotonically non-decreasing in $d_z$.

We refer to a value of $d_z$ as *good*, if it satisfies the following two inequalities:

$$\frac{E(d_z, 0)}{r} \leq \varepsilon \,, \tag{32}$$

and

$$\frac{E(d_z, 1)}{r} \leq \hat{\varepsilon} \,. \tag{33}$$

For such good values of $d_z$, the scheme satisfies both security goals as dictated by the threshold parameters $\varepsilon$ and $\hat{\varepsilon}$. Next, we aim at finding out whether the given $d_m$ allows us to tune $d_z$ to a good value. To that end, we look for the minimal $d_z$ for which (32) holds. Let $d_{z,\min}$ denote that value. Next, we check whether $d_{z,\min}$ satisfies (33). If not, then there exists no good $d_z$, as implied by the monotonicity properties of $E(d_z, 0)$ and $E(d_z, 1)$, and by the natural assumption that $\hat{\varepsilon} \geq \varepsilon$. In that case, the value of $d_m$ is too small and we have no choice but to increase $d_m$ in order to allow the required separation between the two curves $E(d_z, 0)$ and $E(d_z, 1)$. If, on the other hand, $d_{z,\min}$ satisfies (33), then there exists a whole interval of good $d_z$s, which is

$$[d_{z,\min}, d_{z,\max}] \tag{34}$$

where $d_{z,\max}$ is the maximal value of $d_z$ for which (33) is satisfied. In that case, $d_m$ is sufficiently large.

Having defined which values of $d_m$ are too small and which ones are sufficiently large, we find an approximation of the optimal $d_m$ by means of binary search. The stopping criterion is as follows: if we find a sufficiently large $d_m$ where the interval of good $d_z$s, (34), is such that $d_{z,\min} > 0.85 d_{z,\max}$, we stop the binary search and consider the current value of $d_m$ to be "near-optimal".

### 3.2 Pirate codeword generation strategies

We examined four different pirate strategies in generating the pirate codeword. We used the generation assumption which allows the pirate to manipulate only *detectable* bit positions, i.e., positions in which the pirate has both 0 and 1. The strategies that we examined were:

- All 0s - The pirate sets 0s in all detectable positions. This decreases the accusation score for all users – innocent as well as traitors.
- All 1s - The pirate sets 1s in all detectable positions. This increases the accusation score for all users.
- Coin Flip - The pirate randomly sets 0 or 1 in each detectable position by flipping a fair coin.
- Random - The pirate randomly sets 0 or 1 in each detectable position $j$, where the probability in which he selects 1 is $\hat{p}_j$ – the percentage of 1s that the pirate sees in position $j$. (Note that $\hat{p}_j$ are the maximum likelihood estimation of the actual probabilities $p_j$ that were used in generating the codebook, given the codewords that the pirate sees.)

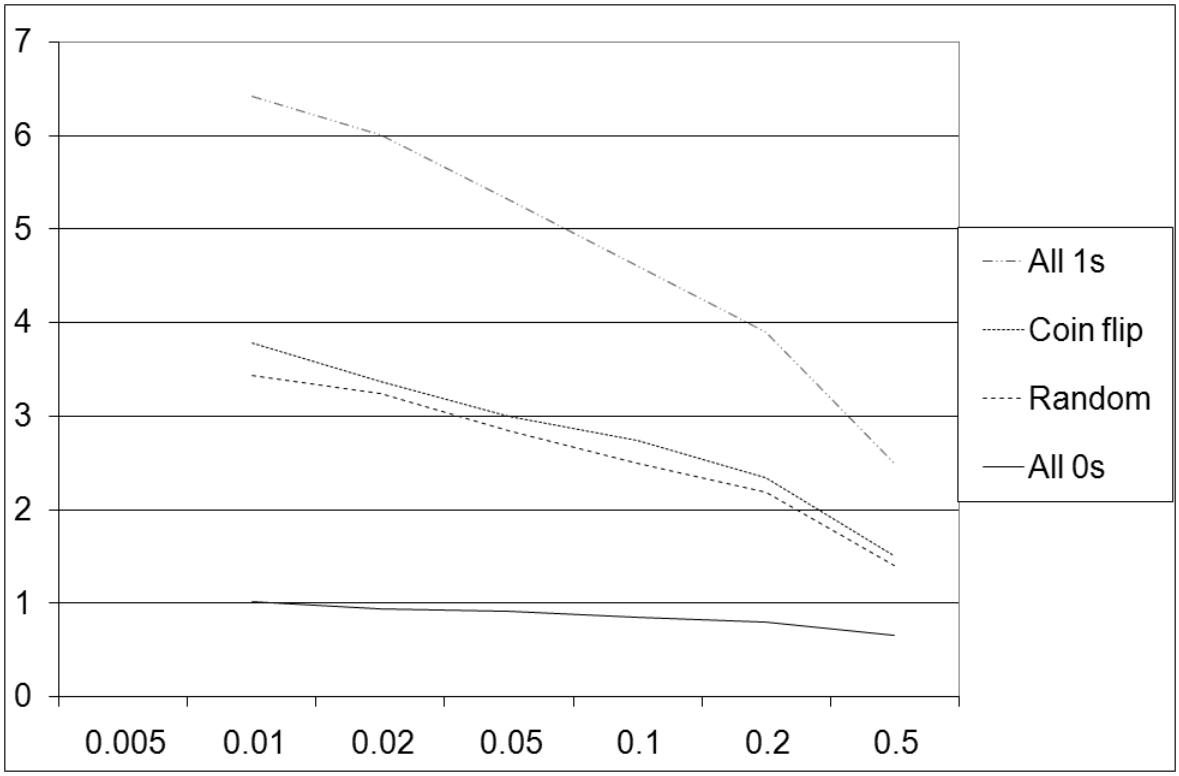


Figure 2: $d_m$ versus $\hat{\varepsilon}$ for different pirate strategies

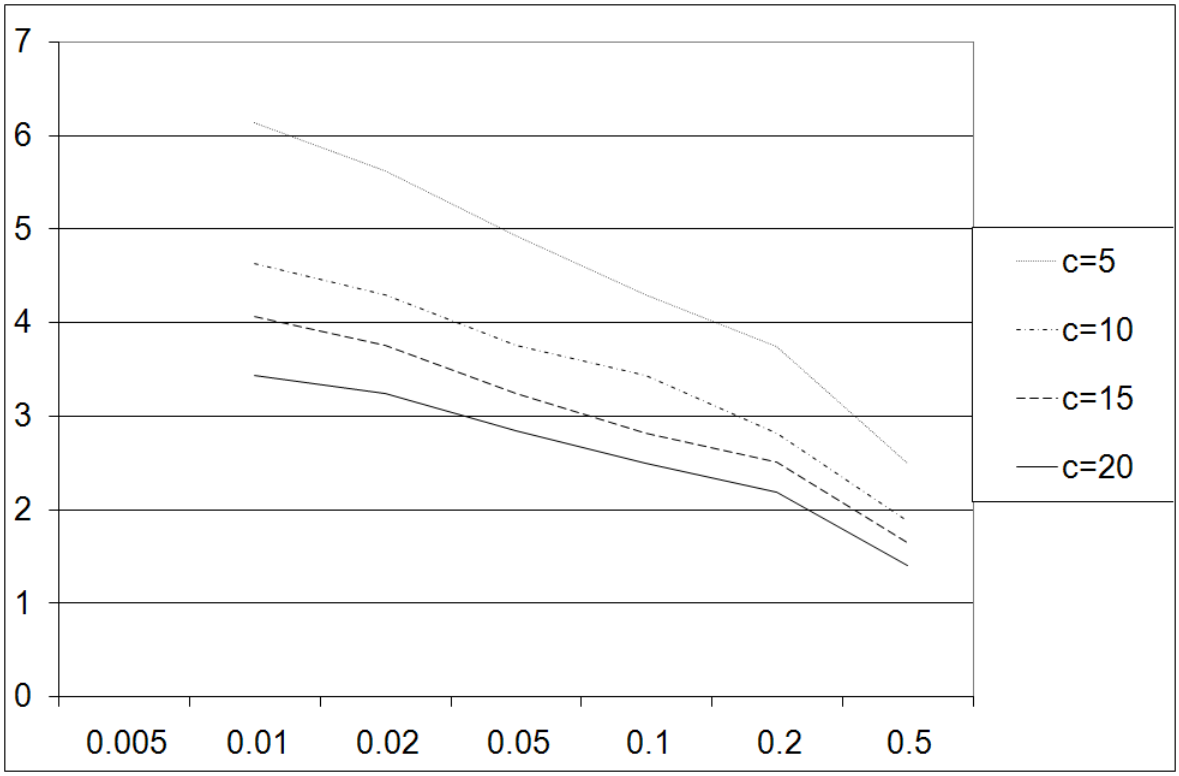


Figure 3: $d_m$ versus $\hat{\varepsilon}$ for different values of $c$

### 3.3 Results

We ran the test for each of the four generation strategies on the following inputs:

$$\begin{array}{l} n = 100, \quad d_t = 50 \\ \varepsilon = \frac{1}{200}, \frac{1}{100}, \frac{1}{50} \\ \hat{\varepsilon} = \frac{1}{200}, \frac{1}{100}, \frac{1}{50}, \frac{1}{20}, \frac{1}{10}, \frac{1}{2} \\ c = 5, 10, 15, 20\,. \end{array} \tag{35}$$

The results show that in "real world" applications of Tardos scheme, we can use $d_m < 8$ for most cases. Namely, in practice one may use codewords that are shorter by a factor of about 16 than the codewords in the original Tardos scheme (where $d_m = 100$). The near-optimal $d_m$ decreases as the maximum coalition size $c$ increases (as we also saw in our analysis in Section 2, see Figure 1 there). Our simulation shows that the best strategy for the pirate is the ALL 1S strategy. That strategy forces us to use the largest value of $d_m$ in order to meet given security requirements.

The following figures visualize some of the results. Figure 2 shows the near-optimal value of $d_m$ as a function of $\hat{\varepsilon}$, when $\varepsilon = 0.01$, $c = 20$, and $n$ and $d_t$ are as in (35). It contains four graphs, one for each of the four pirate strategies, as described in Section 3.2. We can see that the ALL 1S is the best pirate strategy, while the ALL 0S is the worst one. The COIN FLIP and the RANDOM strategies, on the other hand, behave similarly.

Figure 3 shows the near-optimal value of $d_m$ as a function of $\hat{\varepsilon}$, when $\varepsilon = 0.01$ and $n$ and $d_t$ are as in (35). It contains four graphs, one for each value of $c$ in (35). The data is taken from the experiments with the RANDOM strategy. A similar dependence of $d_m$ on $\hat{\varepsilon}$ and $c$ occurs also if we apply each of the other pirate strategies.

Figure 4 shows the near-optimal value of $d_m$ as a function of $\hat{\varepsilon}$, where $n$ and $d_t$ are as in (35), and $c = 20$, for three values of $\varepsilon$ as given in (35). The pirate strategy that we used here is the ALL 1S strategy.

Judging by the simulation results we see that the theoretic bounds are quite distant from the empiric values. For instance, for $n = 100$, $c = 20$, and $\varepsilon = \hat{\varepsilon} = 0.01$, the best $d_m$ that provided a provably $(\varepsilon, \hat{\varepsilon})$-secure scheme was $d_m \approx 38$; empirically, on the other hand, we see that using $d_m = 6.426$ yields a scheme that is also $(\varepsilon, \hat{\varepsilon})$-secure, even in the face of the best pirate strategy.

## References


[1] D. Boneh and J. Shaw, Collusion-Secure Fingerprinting for Digital Data, *IEEE Transactions on Information Theory*, vol. 44, no. 5 (1998), pp. 1897–1905. See also *Proc. Crypto 95*, Springer LNCS 963 (1995), pp. 452-465.

[2] B. Chor, A. Fiat, and M. Naor, Tracing Traitors, *Proc. Crypto 94*, Springer LNCS 839 (1994), pp. 257–270. For a full version see [3].

[3] B. Chor, A. Fiat, M. Naor and B. Pinkas, Tracing Traitors, *IEEE Transactions on Information Theory*, vol. 46, no. 3 (2000), pp. 893-910.

[4] I.J. Cox, J. Kilian, T. Leighton and T. Shamoon, A Secure, Robust Watermark for Multimedia, *Information Hiding*, Springer LNCS 174 (1996), pp. 185–226.

[5] A. Fiat and T. Tassa, Dynamic Traitor Tracing, *Journal of Cryptology*, vol. 14, no. 3 (2001), pp. 211-223. See also *Proc. Crypto 99*, Springer LNCS 1666 (1999), pp. 537-554.

[6] C. Peikert, A. Shelat and A. Smith, Lower Bounds for Collusion-Secure Fingerprinting, in *Proceedings of the 40th Symposium on Discrete Algorithms*, SODA'03, pp. 472-479.

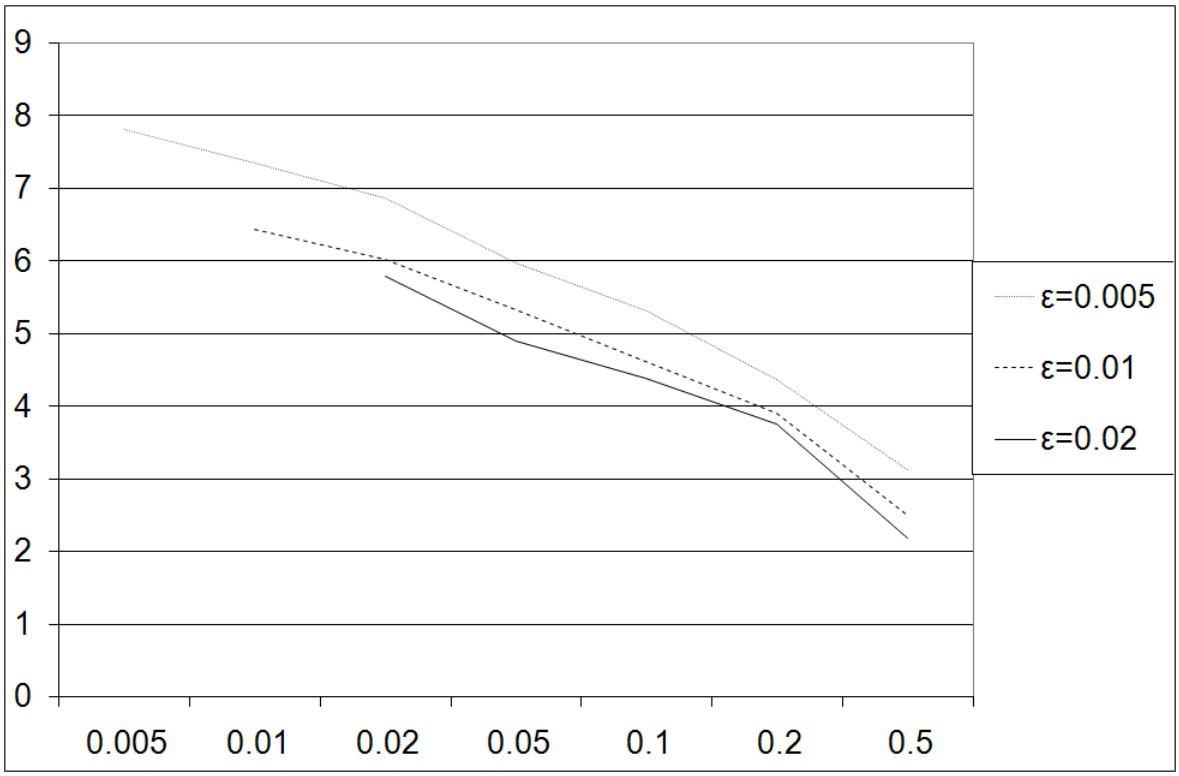


Figure 4: $d_m$ versus $\hat{\varepsilon}$ for different values of $\varepsilon$

[7] G. Tardos, Optimal Probabilistic Fingerprint Codes, In *Proceedings of the 35th ACM Symposium on Theory of Computing*, STOC'03, pp. 116-125.

[8] G. Tardos, Optimal probabilistic fingerprint codes., *Journal of the ACM*, to appear. Available also in http://www.renyi.hu/ tardos/fingerprint.ps.