Dynamic Tardos Traitor Tracing Schemes

Thijs Laarhoven* Jeroen Doumen† Peter Roelse† Boris Škorić* Benne de Weger*

November 15, 2011

Abstract

We construct binary dynamic traitor tracing schemes, where the number of watermark bits needed to trace and disconnect any coalition of pirates is quadratic in the number of pirates, and logarithmic in the total number of users and the error probability. Our results improve upon results of Tassa, and our schemes have several other advantages, such as being able to generate all codewords in advance, a simple accusation method, and flexibility when the feedback from the pirate network is delayed.

1 Introduction

To protect digital content from unauthorized redistribution, distributors embed watermarks in the content such that, if a customer distributes his copy of the content, the distributor can see this copy, extract the watermark and see which user it belongs to. By embedding a unique watermark for each different user, the distributor can always determine from the detected watermark which of the customers is guilty. However, several users could cooperate to form a coalition, and compare their differently watermarked copies to look for the watermark. Assuming that the original data is the same for all users, the differences they detect are differences in their watermarks. The colluders can then distort this watermark, and distribute a copy which matches all their copies on the positions where they detected no differences, and has some possibly non-deterministic output on these detected watermark positions. Since the watermark does not match any user's watermark exactly, finding the guilty users is non-trivial.

In this paper we focus on the problem of constructing efficient collusion-resistant schemes for tracing pirates, which involves finding a way to choose watermark symbols for each user (the traitor tracing code) and a way to trace a detected copy back to the guilty users (an accusation algorithm). In particular, we will focus on the application of such schemes in a dynamic setting, where the pirate output is detected in real-time, before the next watermark symbols are embedded in consecutive segments of the content. We will show that by building upon the Tardos scheme [8], we can construct efficient and flexible dynamic traitor tracing schemes. The number of watermark symbols needed in our schemes is a significant improvement compared to [9], and our schemes can be easily adjusted when the model is slightly different from the standard dynamic traitor tracing model described in [1, 4, 9].

1.1 Model

Let us first formally describe the mathematical model for the problem discussed in this paper. First, some entity called the distributor controls the database of watermarks and distributes the content. The recipients, each receiving a watermarked copy of the content, are referred to as users. We write $U = \{1, \ldots, n\}$ for the set of all users, and we commonly use the symbol j for indexing these users. For the watermarks, we refer to the sequence of watermarking symbols assigned to a user j by the vector $\vec{X}_j$, which is also called a codeword. We write ℓ for the total number of watermark symbols in a codeword, so that each codeword $\vec{X}_j$ has length ℓ, and we commonly use the symbol i to index the watermark positions. In this paper we only focus on watermark symbols from a binary alphabet, so that $(\vec{X}_j)_i \in \{0,1\}$ for all i, j. The traitor tracing code, consisting of all watermark symbols for each user, is denoted by $\mathscr{X} = \{\vec{X}_1, \ldots, \vec{X}_n\}$. A more common way to represent this code is by putting all codewords $\vec{X}_j$ as rows in a matrix X, so that $X_{j,i} = (\vec{X}_j)_i$ is the symbol on position i of user j.

After assigning a codeword to each user, the codewords are embedded in the data as watermarks. The watermarked copies are sent to the users, and some of the users (referred to as the pirates) collude to create a pirate copy. The pirates form a subset $C \subseteq U$, and we use $c = |C|$ for the number of pirates in the coalition. The pirate copy has some distorted watermark, denoted by $\vec{y}$. We assume that if on some position i all pirates see the same symbol, they output this symbol. This assumption is known in the literature as the marking assumption. On other positions we assume pirates simply choose one of the two symbols to output. This choice of pirate symbols can be formalized by denoting a pirate strategy by a function ρ, which maps a code matrix X (or the part of the matrix visible to them) to a forgery $\vec{y}$. After the coalition generates a pirate copy, we assume the distributor detects it and uses some accusation algorithm σ to map the forgery $\vec{y}$ to some subset $\sigma(\vec{y}) = \hat{C} \subseteq U$ of accused users. These users are then disconnected from the system. Ideally $\hat{C} = C$, but this may not always be achievable.

Static schemes. We distinguish between two types of schemes. In static schemes, the process ends after one run of the above algorithm with a fixed codelength ℓ, and the set $\hat{C}$ is the final set of accused users. So the complete codewords are distributed, the pirates generate and distribute a pirate copy, and the distributor detects this output and calculates the set of accused users. In this case an elementary result is that one can never have any certainty of catching all pirates. After all, the

*T. Laarhoven, B. Škorić, and B. de Weger are with the Department of Mathematics and Computer Science, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands.
E-mail: {t.m.m.laarhoven,b.skoric,b.m.m.d.weger}@tue.nl.

†J. Doumen and P. Roelse are with Irdeto BV, 5656 AE Eindhoven, The Netherlands.
E-mail: {jdoumen,peter.roelse}@irdeto.com.

coalition could decide to sacrifice one of its members, so that $\vec{y} = \vec{X}_j$ for some $j \in C$. Then it is impossible to distinguish between other pirates $j' \in C \setminus \{j\}$ and innocent users $j' \in U \setminus C$. However, static schemes do exist that achieve catching at least one guilty user and not accusing any innocent users with high probability. The original Tardos scheme [8] belongs to this class of schemes.

Dynamic schemes. The other type of scheme is the class of dynamic schemes, where the process of sending out symbols, detecting pirate output and running an accusation algorithm is repeated multiple times. In this case, if a user is caught, he is immediately cut off from the system and can no longer access the content. These dynamic scenarios for example apply to live broadcasts, such as pay-tv. The distributor broadcasts the content, while the pirates directly output a pirate copy of the content. The distributor then listens in on this pirate broadcast, extracts the watermarks, and uses this information for the choice of watermarks for the next segment of the content. We assume that the pirates always try to keep their broadcast running, so we assume that if one of the pirates is disconnected, the other pirates will take over. Ideally one demands that the set of accused users always matches the exact coalition, i.e. $\hat{C} = C$, and with dynamic schemes we can also achieve this with high probability, as we will see later. The new schemes we present in this paper belong to this class of schemes.

As mentioned earlier, we call static schemes successful if with high probability, at least one guilty user is caught, and no innocent users are accused. With dynamic schemes one can catch all pirates, so we only call such schemes successful if with high probability, all pirates are caught and no innocent users are accused. This leads to the following definitions of soundness and static/dynamic completeness.

Definition 1 (Soundness and completeness)**.** *Let $(\mathscr{X}, \sigma)$ be a traitor tracing scheme, let $c \geq 2$ and let $\varepsilon_1, \varepsilon_2 \in (0,1)$. Then a scheme has the soundness property with parameter ε_1, or is called ε_1-sound, if for all coalitions $C \subseteq U$ and pirate strategies ρ, the probability of accusing one or more innocent users is bounded from above as*

$$P(\hat{C} \not\subseteq C) \leq \varepsilon_1.$$

A static traitor tracing scheme $(\mathscr{X}, \sigma)$ has the static completeness property with parameters ε_2, c, or is called static (ε_2, c)-complete, if for all coalitions $C \subseteq U$ of size at most c and for all pirate strategies ρ, the probability of not catching any pirates is bounded from above as

$$P(C \cap \hat{C} = \emptyset) \leq \varepsilon_2.$$

Finally, a dynamic traitor tracing scheme $(\mathscr{X}, \sigma)$ has the dynamic completeness property with parameters ε_2, c, or is called dynamic (ε_2, c)-complete, if for all coalitions $C \subseteq U$ of size at most c and for all pirate strategies ρ, the probability of not catching all pirates is bounded from above as

$$P(C \not\subseteq \hat{C}) \leq \varepsilon_2.$$

In the following sections we will omit the c in the completeness property if the parameter is implicit. Similarly, when ε_1 or ε_2 is implicit, we sometimes also simply call a scheme sound or complete. As we will see later, in the schemes discussed in this paper, ε_1/n and ε_2 are closely related. We will use the notation $\eta = \ln(\varepsilon_2)/\ln(\varepsilon_1/n)$ to denote the logratio of these error probabilities. In most practical scenarios we have $\varepsilon_1/n < \varepsilon_2$, so usually $\eta \in (0,1)$.

1.2 Related work

The schemes in this paper all build upon the Tardos scheme, introduced in [8]. This is an efficient static traitor tracing scheme, and it was the first scheme to achieve ε_1-soundness and (ε_2, c)-completeness with a codelength of $\ell = O(c^2 \ln(n/\varepsilon_1))$. In the same paper it was also proved that this asymptotic behaviour is optimal. The original Tardos scheme had a codelength of $\ell = 100c^2 \ln(n/\varepsilon_1)$, and several improvements for the Tardos scheme have been suggested to reduce the constant before the $c^2 \ln(n/\varepsilon_1)$. We mention two in particular: the improved analysis done in [2]; and the introduction of a symmetric score function in [7]. Combining these improvements led to even shorter codelength constants in [6]. For $c \geq 2$ and $\eta \leq 1$ the construction in [6] gives codelengths of $\ell < 24c^2 \ln(n/\varepsilon_1)$, with the constant further decreasing as c increases or η decreases. For asymptotically large c, the construction from [6] gives a codelength of $\ell = [\frac{\pi^2}{2} + O(c^{-1/3})]c^2 \ln(n/\varepsilon_1)$. The scheme from [6] and its properties are discussed in Section 2.

For the dynamic setting, we mention three papers. In [4] a scheme with no probability of error was introduced, which uses a large alphabet size. The number of symbols needed to catch pirates in that scheme is $\ell = O(c \log_2(n))$, but the alphabet size needed is $q = 2c+1$. In [1] several schemes were suggested using a smaller alphabet of size $q = c+1$, with codelengths ranging from $O(c^3 \log_2(n))$ to $O(c^2 + c\log_2(n))$. In [9] the dynamic scheme from [4] was combined with the static scheme from [3], giving a dynamic scheme using a binary alphabet with a total codelength of $\ell = O(c^4 \log_2(n) \ln(c/\varepsilon_1))$. In the same paper the author suggests that using the Tardos scheme instead of the scheme from [3] may decrease the codelength by a factor c, thus possibly giving a codelength of $\ell = O(c^3 \log_2(n) \ln(c/\varepsilon_1))$. As detailed later, the schemes we present here are even more efficient than this suggested improvement.

1.3 Contributions and outline

First we show that the static Tardos scheme can be extended to a dynamic traitor tracing scheme in a very efficient way. This dynamic scheme then has a codelength of $\ell = O(c^2 \ln(n/\varepsilon_1))$, where the constant only slightly increases compared to the scheme from [6], while with this scheme we have certainty about catching all pirates, instead of at least one pirate as in the static Tardos scheme. These adjustments do not influence the codeword generation of the Tardos scheme, so the codewords can still be generated in advance. This is an advantage compared to the schemes from [1, 4, 9].

Since the value of c is usually not known in advance, we then show how to create a c-independent "universal" dynamic scheme that does not require c as input. The property that the codewords can be generated in advance is left unchanged, while the scheme also has several advantages with respect to flexibility, detailed in Section 6. The codelength of this scheme is also $\ell = O(c^2 \ln(n/\varepsilon_1))$, thus improving upon the results from [9]

by roughly a factor $O(c^2)$ and upon the suggested improvement in [9] by a factor $O(c)$.

This paper is organized as follows. First, in Section 2 we present the construction of the static Tardos scheme with the improved analysis from [6]. This scheme and its results will be used as the foundation for the dynamic Tardos scheme, which we present in Section 3. Then, in Section 4 we present a modification of the dynamic Tardos scheme when the setting is not fully dynamic. In Section 5 we then present the universal Tardos scheme, which is an extension of the dynamic Tardos scheme that does not require c as input. In Section 6, we discuss the results and argue that our schemes have several advantages with respect to flexibility as well. Finally, in Section 7 we list some open problems raised by our work.

This paper is mainly based on results from the first author's Master's thesis [5].

2 Preliminaries: The Tardos scheme

The results in the next sections all build upon results from the static Tardos scheme, so we first discuss this scheme here. Since the codeword generation of the schemes discussed in this paper all use (a variant of) the arcsine distribution, we also first describe this distribution below.

2.1 Arcsine distribution

The standard arcsine distribution function $F(p)$, and its associated probability density function $f(p)$, are given by:

$$F(p) = \frac{2}{\pi}\arcsin(\sqrt{p}), \quad f(p) = \frac{1}{\pi\sqrt{p(1-p)}}. \tag{1}$$

This distribution function will be used in Section 5. In Sections 2, 3 and 4 we will use a variant of this distribution function, where the values of p cannot be arbitrarily close to 0 and 1, as this generally leads to a high probability of accusing innocent users. In [8], Tardos therefore used the arcsine distribution with a certain small cutoff parameter $\delta > 0$, such that $p \in [\delta, 1-\delta]$. By scaling F and f appropriately on this interval, this leads to the distribution functions F_δ and associated density functions f_δ:

$$F_\delta(p) = \frac{2\arcsin(\sqrt{p}) - 2\arcsin(\sqrt{\delta})}{\pi - 4\arcsin(\sqrt{\delta})}, \tag{2}$$

$$f_\delta(p) = \frac{1}{(\pi - 4\arcsin(\sqrt{\delta}))\sqrt{p(1-p)}}. \tag{3}$$

Note that taking $\delta = 0$ (i.e. using no cutoff) leads to $F_0(p) \equiv F(p)$.

2.2 Construction

Here we describe the Tardos scheme, with parameters d_ℓ, d_z, d_δ as introduced in [2], and with the symmetric score function introduced in [7]. If certain requirements on these constants are satisfied, one can prove soundness and static completeness, as in [6].

1. **Initialization phase**
   (a) Take the codelength as $\ell = d_\ell c^2 \ln(n/\varepsilon_1)$.
   (b) Take the accusation threshold as $Z = d_z c \ln(n/\varepsilon_1)$.
   (c) Take the cutoff parameter as $\delta = 1/(d_\delta c^{4/3})$. [1]
2. **Codeword generation**
   For each position $1 \le i \le \ell$:
   (a) Select $p_i \in [\delta, 1-\delta]$ from the distribution function $F_\delta(p)$ defined in (2).
   (b) For each user $j \in U$, generate the ith entry of the codeword of user j according to $P(X_{j,i} = 1) = p_i$ and $P(X_{j,i} = 0) = 1 - p_i$.
3. **Distribution of codewords**
   (a) Send to each user $j \in U$ their codeword $\vec{X}_j = (X_{j,1}, \ldots, X_{j,\ell})$, embedded as a watermark in the content.
4. **Detection of pirate output**
   (a) Detect the pirate output, and extract the watermark $\vec{y} = (y_1, \ldots, y_\ell)$.
5. **Accusation phase**
   For each user $j \in U$:
   (a) For each position $1 \le i \le \ell$, calculate the user's score S_{ji} for this position according to:

$$S_{j,i} = \begin{cases} +\sqrt{(1-p_i)/p_i} & \text{if } X_{ji} = 1, y_i = 1, \\ -\sqrt{(1-p_i)/p_i} & \text{if } X_{ji} = 1, y_i = 0, \\ -\sqrt{p_i/(1-p_i)} & \text{if } X_{ji} = 0, y_i = 1, \\ +\sqrt{p_i/(1-p_i)} & \text{if } X_{ji} = 0, y_i = 0. \end{cases} \tag{4}$$

   (b) Calculate the user's total score $S_j(\ell) = \sum_{i=1}^{\ell} S_{ji}$.
   (c) User j is accused (i.e. $j \in \hat{C}$) if $S_j(\ell) > Z$.

2.3 Soundness

For the above construction, one can prove soundness and static completeness, provided the constants d_ℓ, d_z, d_δ satisfy certain requirements. For soundness, in [6] the following lemma was proved. Here $h(x) = (e^x - 1 - x)/x^2$, which is a strictly increasing function from $(0, \infty)$ to $(\frac{1}{2}, \infty)$.

Lemma 1. *[6, Lemma 1] Let the Tardos scheme be constructed as in Subsection 2.2. Let j be some arbitrary innocent user, and let $a > 0$. Then*

$$E\left(e^{aS_j(\ell)c^{-1}}\right) \le \left(\frac{\varepsilon_1}{n}\right)^{-a\lambda_a d_\ell},$$

where $\lambda_a = ah(a\sqrt{d_\delta}c^{-1/3})$.

Now if the following condition of soundness is satisfied,

$$\exists\, a > 0: \quad a(d_z - \lambda_a d_\ell) \ge 1, \tag{S}$$

[1] Previously, in [2, 6–8], it was common to parametrize the offset δ as $\delta = 1/(d_\delta c)$. However, in [6] it was shown that to get optimal results, δ should scale as $c^{-4/3}$ rather than c^{-1}. Therefore we now use $\delta = 1/(d_\delta c^{4/3})$, with d_δ converging to a non-zero constant for asymptotically large c.

then using the Markov inequality and Lemma 1 with this a, for innocent users j we get

$$P(j \in \hat{C}) = P(S_j(\ell) > Z) \le \frac{E\left(e^{aS_j(\ell)c^{-1}}\right)}{e^{aZc^{-1}}} \le \left(\frac{\varepsilon_1}{n}\right)^{a(d_z-\lambda_a d_\ell)} \le \frac{\varepsilon_1}{n}.$$

So the probability that no innocent user is accused is at least $(1-\frac{\varepsilon_1}{n})^n \ge 1-\varepsilon_1$, as was also shown in [6, Theorem 3].

2.4 Static completeness

To prove static completeness, in [6] the following lemma was used. Below, and throughout the rest of this paper, $S(\ell) = \sum_{j\in C} S_j(\ell)$ represents the total coalition score, i.e. the sum of the scores of all pirates $j \in C$.

Lemma 2. *[6, Lemma 2] Let the Tardos scheme be constructed as in Subsection 2.2, and let $b > 0$. Then*

$$E\left(e^{-bS(\ell)c^{-5/3}}\right) \le \left(\frac{\varepsilon_1}{n}\right)^{b\lambda_b d_\ell c^{1/3}},$$

where $\lambda_b = \frac{2}{\pi} - \frac{4}{d_\delta \pi} c^{-1/3} - bh(b\sqrt{d_\delta})c^{-2/3}$.

If the following condition of completeness is satisfied,

$$\exists\, b > 0: \quad b(\lambda_b d_\ell - d_z) \ge \eta c^{-1/3}, \tag{C}$$

then using the pigeonhole principle, the Markov inequality and Lemma 2 with this b we get

$$P(C \cap \hat{C} = \emptyset) \le P(S(\ell) < cZ) \le \frac{E\left(e^{-bS(\ell)c^{-5/3}}\right)}{e^{-bZc^{-2/3}}} \le \left(\frac{\varepsilon_1}{n}\right)^{b(\lambda_b d_\ell - d_z)c^{1/3}} \le \left(\frac{\varepsilon_1}{n}\right)^{\eta} = \varepsilon_2.$$

So static completeness follows from Lemma 2 and condition (C), as was also shown in [6, Theorem 4].

2.5 Codelengths

In [2] and [6] a detailed analysis is given to go from requirements (S) and (C) to the asymptotically optimal set of parameters that satisfies the constraints and minimizes d_ℓ. Recall that $\ell = d_\ell c^2 \ln(n/\varepsilon_1)$, so a smaller d_ℓ gives shorter codelengths, whereas the parameters d_z and d_δ affect only Z and δ, which have no influence on the efficiency of the scheme. In [6] the following result was obtained.

Lemma 3. *[6, Theorem 6] Let $\gamma = \left(\frac{2}{3\pi}\right)^{2/3} \approx 0.36$. The optimal asymptotic value for d_ℓ is*

$$d_\ell = \frac{\pi^2}{2} + O(c^{-1/3}),$$

the associated values for d_z and d_δ are

$$d_z = \pi + O(c^{-1/3}), \quad d_\delta = \frac{4}{\gamma} - O\left(\frac{\eta}{\ln c}\right),$$

and the corresponding values for $a, b, \lambda_a, \lambda_b$ are

$$a = \frac{2}{\pi} - O(c^{-1/3}), \quad b = \frac{\ln c}{9\pi\gamma} - O\left(\ln\left(\frac{\ln c}{\eta}\right)\right),$$
$$\lambda_a = \frac{1}{\pi} + O(c^{-1/3}), \quad \lambda_b = \frac{2}{\pi} - O(c^{-1/3}).$$

A direct consequence of Lemma 3 is the following corollary, which gives the optimal asymptotic scheme parameters for $c \to \infty$.

Corollary 1. *[6, Corollary 1] The construction from Subsection 2.2 gives an ε_1-sound and static (ε_2, c)-complete scheme with asymptotic scheme parameters*

$$\ell \to \frac{\pi^2}{2} c^2 \ln(n/\varepsilon_1), \quad Z \to \pi c \ln(n/\varepsilon_1), \quad \delta \to \frac{\gamma}{4} c^{-4/3}.$$

For further details on the optimal first order constants, see [6].

2.6 Example

Let the scheme parameters be given by $c = 25$ pirates, $n = 10^6$ users, and error probabilities $\varepsilon_1 = \varepsilon_2 = 10^{-3}$. Then $\eta = \frac{1}{3}$, and the optimal values of d_ℓ, d_z, d_δ can be calculated numerically as

$$d_\ell = 8.46, \quad d_z = 4.53, \quad d_\delta = 14.36.$$

This leads to the scheme parameters

$$\ell = 109\,585, \quad Z = 2345, \quad \delta = 5.09 \cdot 10^{-4}.$$

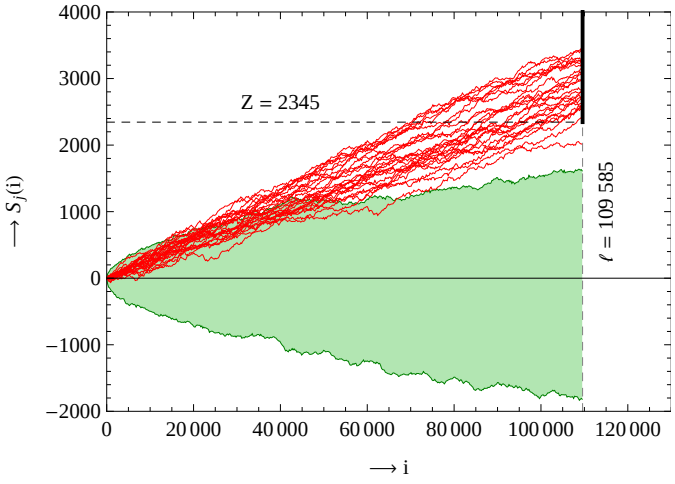
So using these scheme parameters, we know that after 109 585 symbols, with probability at least $1-\varepsilon_1$ there are no false accusations, and with probability at least $1-\varepsilon_2$ at least one pirate is accused. In Fig. 1 we show simulation results for these parameters. The curves in the figure are the pirate scores $S_j(i)$ for each pirate $j \in C$, while the shaded area is bounded from above by the highest score of an innocent user, and bounded from below by the lowest score of an innocent user in this simulation. In Fig. 1a we simulated pirates using the interleaving attack (i.e. for each position, they choose a random pirate and output his symbol), and in Fig. 1b they used the scapegoat strategy (i.e. one pirate, the scapegoat, always outputs his symbol, until he is caught and another pirate is picked as the scapegoat). With the scapegoat strategy, only one pirate is caught, while using the interleaving attack leads to many accused pirates.

3 The dynamic Tardos scheme

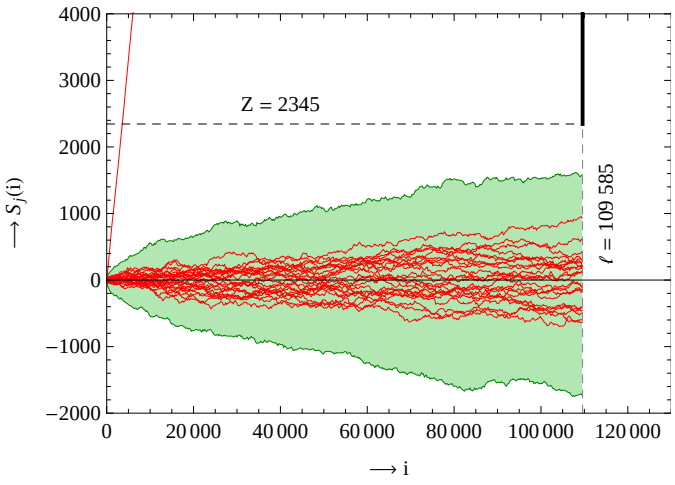
Let us now explain how we create a dynamic scheme from the static Tardos scheme, such that with high probability we catch all colluders, instead of at least one colluder. The change we make is the following. Instead of only comparing scores with Z after ℓ symbols, we now compare the scores with Z after every position i. If a user's score exceeds Z at any point in time, then he is disconnected immediately and can no longer access the content. His score is then necessarily between Z and $\tilde{Z} := Z + \sqrt{d_\delta} c^{2/3} > Z + \max_{p_i, X_{ji}, y_i} S_{ji}$. The rest of the construction remains the same, except for the values of d_ℓ, d_z, d_δ, which now have to be chosen differently.

3.1 Construction

The scheme again depends on three constants d_ℓ, d_z, d_δ. We show in Subsections 3.2 and 3.3 that if certain requirements on these constants are satisfied, we can prove soundness and dynamic completeness. Below we say a user is active if he is



(a) Interleaving attack



(b) Scapegoat strategy

Figure 1: Simulations of the Tardos scheme, with $c = 25$ colluders, $n = 10^6$ users, and error probabilities $\varepsilon_1 = \varepsilon_2 = 10^{-3}$. In Fig. 1a the pirates used the interleaving attack, whereas in Fig. 1b they used the scapegoat strategy. In both cases, the total coalition score $S(\ell)$ at time ℓ is approximately 72 000, but while in the first case the score is evenly divided among the pirates, in the second case one pirate takes all the blame.

not disconnected in the scheme. We assume that the pirates always output some watermarked data, unless all of the pirates are disconnected. Then the traitor tracing scheme terminates.

1. **Initialization phase**
   (a) Take the codelength as $\ell = d_\ell c^2 \ln(n/\varepsilon_1)$.
   (b) Take the accusation threshold as $Z = d_z c \ln(n/\varepsilon_1)$.
   (c) Take the cutoff parameter as $\delta = 1/(d_\delta c^{4/3})$.
   (d) Set initial user scores at $S_j(0) = 0$.
2. **Codeword generation**
   For each position $1 \le i \le \ell$:
   (a) Select $p_i \in [\delta, 1-\delta]$ from $F_\delta(p)$ defined in (2).
   (b) Generate $X_{j,i} \in \{0,1\}$ using $P(X_{j,i} = 1) = p_i$.
3. **Distribution/Detection/Accusation**
   For each position $1 \le i \le \ell$:
   (a) Send to each active user j symbol $X_{j,i}$.
   (b) Detect the pirate output y_i.
   (If there is no pirate output, terminate.)
   (c) Calculate scores $S_{j,i}$ using (4).
   (d) For active users j, set $S_j(i) = S_j(i-1) + S_{j,i}$.
   (For inactive users j, set $S_j(i) = S_j(i-1)$.)
   (e) Disconnect all active users j with $S_j(i) > Z$.

In the construction above, we separated the codeword generation from the distribution, detection and accusation. These phases can also be merged by generating p_i and X_{ji} once we need them. However, we present the scheme as above to emphasize the fact that these phases can indeed be executed sequentially instead of simultaneously, and that the codeword generation can thus be done before the traitor tracing process begins.

3.2 Soundness

For the dynamic Tardos scheme as given above, we can prove the following result regarding soundness.

Theorem 1. *Consider the dynamic Tardos scheme in Subsection 3.1. If the following condition is satisfied,*

$$\exists\, a > 0: \quad a(d_z - \lambda_a d_\ell) \ge 1 + \frac{\ln(2)}{\ln(n/\varepsilon_1)}, \qquad \text{(S')}$$

then the scheme is ε_1-sound.

To prove the theorem, we first prove a relative upper bound on the probability that a single innocent user is accused and disconnected. This bound relates the error probability in the dynamic Tardos scheme to the probability that the user score at time ℓ is above Z. We then use the proof of the original Tardos scheme to get an absolute upper bound on the soundness error probability, and to prove Theorem 1. Since the relative upper bound gives us an extra factor 2, and since the terms in (S') appear as exponents in the proof, we get an additional term $\ln(2)/\ln(n/\varepsilon_1)$ compared to (S). Note that this term is small for reasonable values of n and ε_1, so this only has a small impact on the right hand side of (S'), compared to (S).

In the following we write $\tilde{S}_j(i) = \sum_{k=1}^{i} S_{jk}$. If user j is still active at time i, then $\tilde{S}_j(i) = S_j(i)$, but whereas $S_j(i)$ does not change anymore once user j is disconnected, the score $\tilde{S}_j(i)$ does change on every position even if the user has already been disconnected, and calculates the user's score as if he had not been disconnected. Similarly, we write $\tilde{S}(i) = \sum_{j \in C} \tilde{S}_j(i)$ for coalitions C. Note that if the last pirate is disconnected at position $i_0 < \ell$, then S_{ji} and $S_j(i)$ are not defined for $i_0 < i \le \ell$.

Lemma 4. *Let $j \in U$ be an arbitrary innocent user, let $C \subseteq U \setminus \{j\}$ be a pirate coalition and let ρ be some pirate strategy employed by this coalition. Then*

$$P(j \in \hat{C}) \le 2 \cdot P\left(\tilde{S}_j(\ell) > Z\right).$$

Proof. Let us define events A and B as

$$A := \{j \in \hat{C}\} = \{S_j(\ell) > Z\} = \bigcup_{i=1}^{\ell} \{\tilde{S}_j(i) > Z\},$$

$$B := \{\tilde{S}_j(\ell) > Z\}.$$

We trivially have $P(A \mid B) = 1$. For $P(B \mid A)$, note that under the assumption that A holds, the process $\{\tilde{S}_j(i)\}_{i=i_0}^{\infty}$ starting at position $i_0 = \min\{i : S_j(i) > Z\} \leq \ell$ describes a symmetric random walk with no drift. So we then have $P(\tilde{S}_j(\ell) \geq \tilde{S}_j(i_0)) = 1/2$, and since $S_j(i_0) > Z$ it follows that $P(B \mid A) \geq 1/2$. Finally we apply Bayes' Theorem to A and B to get

$$P(A) = \frac{P(A \mid B)}{P(B \mid A)} \cdot P(B) \leq 2 \cdot P(B).$$

This completes the proof. □

Proof of Theorem 1. First, we remark that the distribution of $\tilde{S}_j(\ell)$ is the same as the distribution of the scores $S_j(\ell)$ in the original Tardos scheme, for the same parameters ℓ, Z, δ. From the Markov inequality, Lemma 1 and condition (S') it thus follows that

$$P(\tilde{S}_j(\ell) > Z) \leq \frac{E\left(e^{a\tilde{S}_j(\ell)c^{-1}}\right)}{e^{aZc^{-1}}} \leq \left(\frac{\varepsilon_1}{n}\right)^{a(d_z - \lambda_a d_\ell)} \leq \frac{\varepsilon_1}{2n}.$$

Using Lemma 4 the result follows. □

3.3 Dynamic completeness

With the dynamic Tardos scheme, we get the following result regarding dynamic completeness. Recall that here we require that *all* pirates are caught, instead of at least one, as was the case in the original Tardos scheme.

Theorem 2. *Consider the dynamic Tardos scheme in Subsection 3.1. If the following condition is satisfied,*

$$\exists\, b > 0 : b(\lambda_b d_\ell - d_z) \geq \left(\eta + \frac{\ln(2) + b\sqrt{d_\delta}}{\ln(n/\varepsilon_1)}\right) c^{-1/3}, \quad \text{(C')}$$

then the scheme is dynamic (ε_2, c)-complete.

Similar to the proof of soundness, we prove dynamic completeness by relating the error probability to the static completeness error probability of the static scheme from Section 2. Then we use the results from the static scheme to complete the proof. We again see a factor 2 in the relative upper bound, which again explains the additional term $\ln(2)/\ln(n/\varepsilon_1)$ compared to (C). The other term $b\sqrt{d_\delta}/\ln(n/\varepsilon_1)$ is a consequence of using $\tilde{Z}$ instead of Z in the proofs. Note that these two terms are small, compared to the first term η.

Lemma 5. *Let C be a coalition of size at most c, and let ρ be any pirate strategy employed by this coalition. Then*

$$P(C \not\subseteq \hat{C}) \leq 2 \cdot P\left(\tilde{S}(\ell) < c\tilde{Z}\right).$$

Proof. First we remark that $P(\tilde{S}(\ell) < c\tilde{Z} \mid C \not\subseteq \hat{C}) \geq 1/2$. This is because if $C \not\subseteq \hat{C}$, then $S(\ell) < c\tilde{Z}$, and since $\tilde{S}(\ell) - S(\ell) = R(\ell)$ is a symmetrical, unbiased random walk, with probability at least $1/2$ we have $R(\ell) < 0$ and $\tilde{S}(\ell) < c\tilde{Z}$. Next, we use the definition of the conditional probability to get

$$\begin{aligned} P(C \not\subseteq \hat{C}) &\leq 2 \cdot P(C \not\subseteq \hat{C}) \cdot P(\tilde{S}(\ell) < c\tilde{Z} \mid C \not\subseteq \hat{C}) \\ &= 2 \cdot P(\tilde{S}(\ell) < c\tilde{Z}, C \not\subseteq \hat{C}) \leq 2 \cdot P(\tilde{S}(\ell) < c\tilde{Z}). \end{aligned}$$

This proves the result. □

Proof of Theorem 2. First, note that in the dynamic Tardos scheme, the only extra information pirates receive compared to the static Tardos scheme is the fact whether some of them are disonnected. This information is certainly covered by the information contained in the values of p_i (if pirates know p_i, then they can calculate their scores $S_j(i)$ themselves and calculate whether they will be disconnected or not). Also note that $\tilde{S}(\ell)$ behaves the same as $S(\ell)$ in the static Tardos scheme, where the total coalition score is calculated for all pirates and all positions, regardless of whether they contributed on that position or not. So if we can prove that even in the static Tardos scheme, and even if coalitions get information about the values of p_i, the probability of keeping the coalition score $S(\ell)$ below $c\tilde{Z}$ is bounded by $\varepsilon_2/2$, then it follows that also $P(\tilde{S}(\ell) < c\tilde{Z}) \leq \varepsilon_2/2$.

For the static Tardos scheme, note that the proof method for the completeness property does not rely on p_i being secret. The only assumption that is used in that proof is that the Marking Assumption applies, which does apply here. So here we can also use the proof method of the static Tardos scheme. From the Markov inequality, Lemma 2 and condition (C'), it thus follows that

$$\begin{aligned} P\left(\tilde{S}(\ell) < c\tilde{Z}\right) &\leq \frac{E\left(e^{-b\tilde{S}(\ell)c^{-5/3}}\right)}{e^{-b(Z+\sqrt{d_\delta}c^{2/3})c^{-2/3}}} \\ &\leq \left(\frac{\varepsilon_1}{n}\right)^{b\left(\lambda_b d_\ell - d_z - \frac{\sqrt{d_\delta}}{\ln(n/\varepsilon_1)} c^{-1/3}\right) c^{1/3}} \\ &\leq \left(\frac{\varepsilon_1}{n}\right)^{\eta + \frac{\ln 2}{\ln(n/\varepsilon_1)}} = \frac{\varepsilon_2}{2}. \end{aligned}$$

Using Lemma 5 the result then follows. □

3.4 Codelengths

The requirements (S') and (C') are only slightly different from requirements (S) and (C). For asymptotically large c, these differences disappear, and thus the optimal asymptotic codelength is the same as in the static Tardos scheme. In Fig. 2 we show the optimal values of d_ℓ in the dynamic Tardos scheme for $\eta = 1$ and $\eta = 0.01$. The different curves correspond to different values of n/ε_1, ranging from $n/\varepsilon_1 = 10^3$ (the highest values of d_ℓ) to $n/\varepsilon_1 = 10^{15}$ (the lowest values of d_ℓ).

Note that these values of d_ℓ correspond to the theoretical codelengths such that probability at least $1 - \varepsilon_1$, by time ℓ all of the pirates have been disconnected. This however does not mean that the last pirate is caught exactly at time ℓ, only that he is caught before or at time ℓ. So in practice the number of symbols needed to disconnect all traitors may be below this theoretical codelength ℓ, and may even decrease compared to the static Tardos scheme.

Furthermore, if the coalition size is not known, but only an upper bound on the coalition size is known, i.e. $c \leq c_0$ for some c_0, then one generally uses a traitor tracing scheme that is resistant against up to c_0 colluders. In [7] it was shown that for the Tardos scheme, the total coalition score $S(i) = \sum_{j \in C} S_j(i)$ always increases linearly in i with approximately the same slope, regardless of the coalition size. More precisely, the score $S(i)$ behaves as $S(i) \approx i\tilde{\mu}$, with $\tilde{\mu} \approx \frac{2}{\pi}$ only slightly depending on the coalition size c and the pirate strategy ρ. Since one chooses ℓ and Z such that $S(\ell) \approx \ell\tilde{\mu} \approx c_0 Z$, it follows that $S(\frac{c}{c_0}\ell) \approx cZ$. In other words, to catch a coalition of size

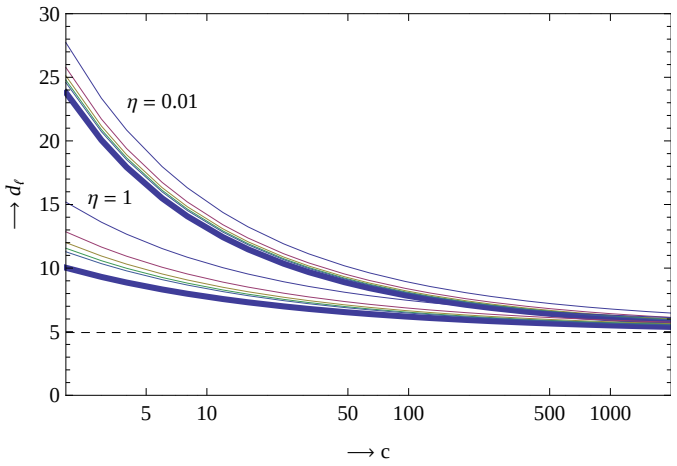


Figure 2: Optimal values of d_ℓ in the dynamic Tardos scheme. The dotted line corresponds to the asymptotic optimal value $d_\ell = \frac{\pi^2}{2} \approx 4.93$. The bold curves show the values of d_ℓ in the static Tardos scheme for $\eta = 1$ (top) and $\eta = 0.01$ (bottom) respectively. The five curves slightly above each of the bold curves show the optimal values of d_ℓ in the dynamic Tardos scheme for $n/\varepsilon_1 = 10^{3k}$, for $k = 1$ up to 5. Higher values of k correspond to lower values of d_ℓ.

$c \le c_0$, the expected number of symbols needed is approximately $\ell = O(\frac{c}{c_0}\ell) = O(cc_0 \ln(n/\varepsilon_1))$. So compared to the static Tardos scheme, where the codelength is fixed in advance at $O(c_0^2 \ln(n/\varepsilon_1))$, the codelength is reduced by a factor $\frac{c}{c_0}$. In particular, small coalitions of few pirates are caught up to $O(c_0)$ times faster.

3.5 Example

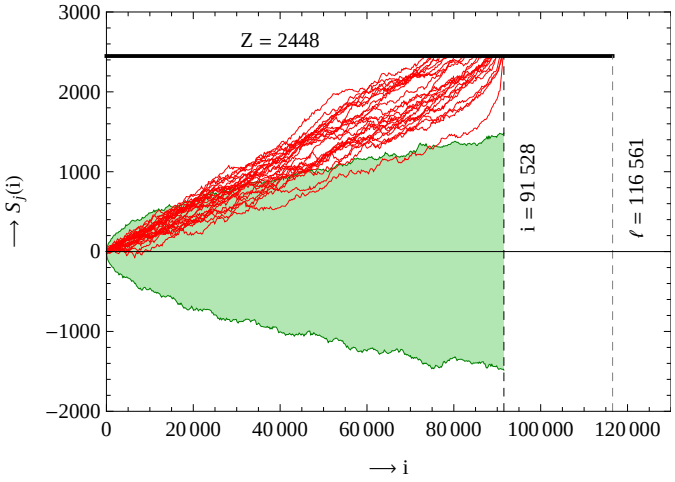
Let the scheme parameters be the same as in Section 2.6, i.e. $c = 25$, $n = 10^6$ and $\varepsilon_1 = \varepsilon_2 = 10^{-3}$, so that $\eta = \frac{1}{3}$. The optimal values of d_ℓ, d_z, d_δ satisfying (S') and (C') can be calculated numerically as

$$d_\ell = 9.00, \quad d_z = 4.73, \quad d_\delta = 13.44$$

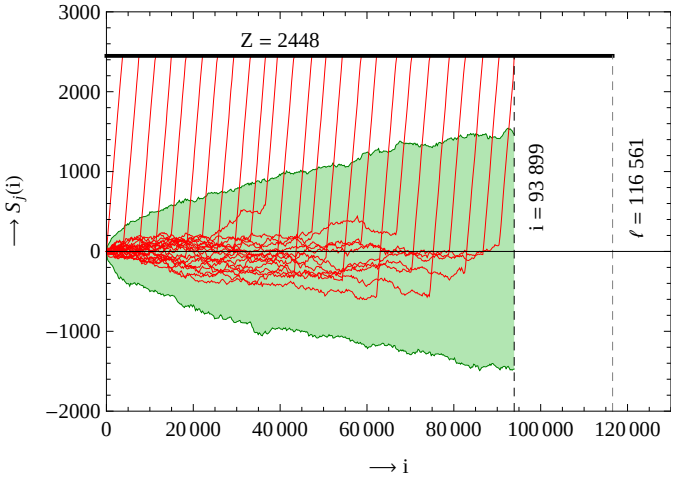
This leads to the scheme parameters

$$\ell = 116\,561, \quad Z = 2448, \quad \delta = 1.02 \cdot 10^{-3}$$

In Fig. 3 we show some simulation results for these parameters. In Fig. 3a the pirates used the interleaving attack, and in Fig. 3b they used the scapegoat strategy.



(a) Interleaving attack



(b) Scapegoat strategy

Figure 3: Simulations of the dynamic Tardos scheme, with parameters $c = 25$, $n = 10^6$ and $\varepsilon_1 = \varepsilon_2 = 10^{-3}$. In Fig. 3a the pirates used the interleaving attack, whereas in Fig. 3b they used the scapegoat strategy. In both cases, after less than 95 000 symbols all pirates have been caught, which is less than the theoretical codelength $\ell = 116\,561$, and less than the codelength of the static Tardos scheme with the same parameters, $\ell = 109\,585$.

4 The semi-dynamic Tardos scheme

In the dynamic Tardos scheme, we need to disconnect users as soon as their scores exceed the threshold Z. In some scenarios this may not be feasible, and we may only be able to disconnect users several positions later, say after B more symbols. This can have several reasons:

- The pirates transmit their content with a small delay, so that there is a delay of B symbols between original broadcast and the corresponding pirate output.
- Detecting the pirate output, extracting the watermark, updating the scores and disconnecting users takes so much time that a user is only disconnected after B more symbols have already been distributed.

For these cases, we present two different solutions. First, in Section 4.1 we present a scheme that achieves a codelength of at most $\ell = d_\ell c^2 \ln(n/\varepsilon_1) + Bc$, where d_ℓ is the same as in the dynamic Tardos scheme for the same parameters. For small values of B, this means that with codelength which is only slightly higher than in the dynamic Tardos scheme, we can also catch all pirates in a semi-dynamic setting. Then, in Section 4.2 we present a scheme that achieves a codelength of $\ell = d_{\ell,B} c^2 \ln(n/\varepsilon_1)$, where $d_{\ell,B}$ increases with B. Since a small increase in d_ℓ can already lead to a big increase in the codelength, the second scheme generally has a larger codelength than the first scheme. Only for values B with $B > (d_{\ell,B} - d_\ell)c \ln(n/\varepsilon_1) = \Omega(c \ln(n/\varepsilon_1))$, the second scheme achieves a shorter codelength than the first scheme.

4.1 First scheme: $\ell = d_\ell c^2 \ln(n/\varepsilon_1) + Bc$

The first scheme is based on the following modification to the accusation algorithm of the dynamic Tardos scheme. Suppose a user's score exceeds Z after i_0 positions. At position i_0 we

now disconnect this user. Since this user may have contributed to the next B symbols of the pirate output $\vec{y}$, we disregard the following B positions of the watermark, and do not update the scores for positions $i \in \{i_0+1,\dots,i_0+B\}$. After those positions we continue the traitor tracing process as in the dynamic Tardos scheme, and we repeat the above procedure for each time a user's score exceeds Z.

With this modification, the traitor tracing process on those positions that were used for calculating scores is identical to the traitor tracing process of the dynamic Tardos scheme. We can therefore use the analysis from Section 3 and conclude that with $d_\ell c^2 \ln(n/\varepsilon_1)$ positions for which we calculate scores, we catch any coalition of size ℓ. Since we disregarded Bc positions, the pirate broadcast will not last longer than $\ell = d_\ell c^2 \ln(n/\varepsilon_1) + Bc$ positions in total, where d_ℓ, d_z and d_δ are as in the dynamic Tardos scheme for the same parameters. This means that with at most Bc more symbols than in the dynamic Tardos scheme, we can also catch coalitions in this semi-dynamic traitor tracing setting.

4.2 Second scheme: $\ell = d_{\ell,B} c^2 \ln(n/\varepsilon_1)$

Instead of using Bc more symbols, we can also try to adjust the analysis of the dynamic Tardos scheme to the semi-dynamic traitor tracing scenario. We can do this by following the proof methods of the dynamic Tardos scheme, and by making one small adjustment. The change we make in the analysis is to use $\tilde{Z}_B := Z + B\sqrt{d_\delta}c^{2/3} > Z + B\max_p S_{ji}(p)$ instead of $\tilde{Z} = Z + \sqrt{d_\delta}c^{2/3}$ as our new upper bound for the scores of users in the proofs. This results in the following, slightly different condition for dynamic completeness:

$$\exists\, b > 0 : b(\lambda_b d_\ell - d_z) \geq \left(\eta + \frac{\ln(2) + Bb\sqrt{d_\delta}}{\ln(n/\varepsilon_1)}\right) c^{-1/3}. \quad \text{(C")}$$

If some parameters $d_{\ell,B}, d_{z,B}, d_{\delta,B}$ satisfy (S') and (C"), then using these constants as our scheme parameters, we obtain a ε_1-sound and dynamic (ε_2, c)-complete scheme with a codelength of $\ell = d_{\ell,B} c^2 \ln(n/\varepsilon_1)$. In Fig. 4 we show the values of $d_{\ell,B}$ for the parameters $n = 10^6$, $\varepsilon_1 = \varepsilon_2 = 10^{-3}$, and $\eta = \frac{1}{3}$, for several values of B. As the value of B increases, the value of $d_{\ell,B}$ increases as well.

4.3 Example

As before, let the scheme parameters be given by $c = 25$, $n = 10^6$ and $\varepsilon_1 = \varepsilon_2 = 10^{-3}$, so that $\eta = \frac{1}{3}$, and let us use $B = 8$. For the first scheme, the codelength then increases by $Bc = 200$ compared to the dynamic Tardos scheme, giving scheme parameters:

$$\ell = 116\,761, \quad Z = 2448, \quad \delta = 1.02 \cdot 10^{-3}$$

For the second scheme, the optimal values of $d_{\ell,B}, d_{z,B}, d_{\delta,B}$ satisfying (S') and (C") for $B = 8$ can be calculated numerically as

$$d_{\ell,B} = 10.16, \quad d_{z,B} = 4.94, \quad d_{\delta,B} = 10.07.$$

This leads to the scheme parameters

$$\ell = 131\,587, \quad Z = 2561, \quad \delta = 1.36 \cdot 10^{-3}.$$

So in this case, using the first scheme leads to a shorter codelength.

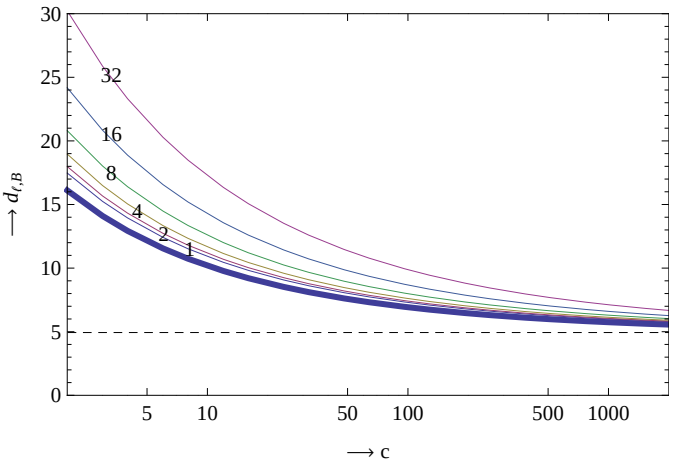


Figure 4: Optimal values of $d_{\ell,B}$ in the semi-dynamic Tardos scheme from Section 4.2, for the parameters $n = 10^6$, $\varepsilon_1 = \varepsilon_2 = 10^{-3}$, and $\eta = \frac{1}{3}$. The bold curve corresponds to the values of d_ℓ in the static Tardos scheme with the same parameters, while the six curves above this curve correspond to the optimal values of $d_{\ell,B}$ for $B = 1, 2, 4, 8, 16, 32$ respectively. The dotted line corresponds to the asymptotic optimal value $d_\ell = \frac{\pi^2}{2} \approx 4.93$. For $B = 1$ we get exactly the codelengths of the dynamic Tardos scheme.

5 The universal Tardos scheme

In this section we present a dynamic scheme that does not require c as input. The name "universal" comes from the fact that for the codeword generation, we now use a distribution function F which does not depend on c, but is a universal distribution function that can be used for all values of c. Because of this, we can use the same codewords for different values of c. In particular, we will use the first $\ell^{(c)} = O(c^2 \ln(n/\varepsilon_1))$ symbols to catch coalitions of size c, for $c \geq 2$. We do this in such a way that if a coalition has some unknown size c, then after $\ell^{(c)}$ symbols, the probability of not having caught all members of this coalition is at most ε_2. Since we do this for each value of c, we now only need $O(c^2 \ln(n/\varepsilon_1))$ symbols to catch a coalition of size c.

The only problem with this new codeword generation method is that a completely universal distribution function which is optimally efficient for all values of c does not exist. More precisely, the proof of soundness of the Tardos scheme requires that the cutoff parameter δ is sufficiently large in terms of c, whereas for completeness we need that δ approaches 0 as $c \to \infty$. One way to avoid this problem is the following. For generating the values of p_i, we use the standard arcsine distribution function from (1). Then for any value of c we simply disregard those values p_i which are not between the corresponding cutoff parameter δ and $1-\delta$. The fraction of values of p_i that is disregarded can be estimated easily as follows:

$$1 - \int_{\delta^{(c)}}^{1-\delta^{(c)}} f(p)\,\mathrm{d}p = \frac{4}{\pi} \arcsin\left(\sqrt{\delta^{(c)}}\right).$$

Since $\frac{4}{\pi}\arcsin(\sqrt{\delta^{(c)}}) = \frac{4}{\pi\sqrt{d_\delta}} c^{-2/3} + O(c^{-2})$, the right hand side is small and decreasing in c, so the fraction of disregarded data is small.

5.1 Construction

Basically we run several dynamic Tardos schemes simultaneously with shared codewords, so scheme parameters and scores

now have to be calculated for each of these schemes, i.e. for each of the values of c. We introduce counters $t^{(c)}$ to keep track of the number of positions that are not disregarded, for each value of c. For each c we then basically run a dynamic Tardos scheme using the same code X until $t^{(c)} = \ell^{(c)}$.

1. **Initialization phase**
   For each $c \geq 2$:
   (a) Take the codelength as $\ell^{(c)} = d_\ell^{(c)} c^2 \ln(n/\varepsilon_1^{(c)})$.
   (b) Take the threshold as $Z^{(c)} = d_z^{(c)} c \ln(n/\varepsilon_1^{(c)})$.
   (c) Take the cutoff parameter as $\delta^{(c)} = 1/(d_\delta^{(c)} c^{4/3})$.
   (d) Initialize the user scores at $S_j^{(c)}(0) = 0$.
   (e) Initialize the counters $t^{(c)}$ at $t^{(c)}(0) = 0$.
2. **Codeword generation**
   For each position $i \geq 1$:
   (a) Select $p_i \in [0,1]$ from $F(p)$ as defined in (1).
   (b) Generate $X_{ji} \in \{0,1\}$ using $P(X_{ji} = 1) = p_i$.
3. **Distribution/Detection/Accusation**
   For each position $i \geq 1$:
   (a) Send to each active user j symbol X_{ji}.
   (b) Detect the pirate output y_i.
   (If there is no pirate output, terminate.)
   (c) Calculate scores S_{ji} using (4).
   (d) For active users j and values c such that $p_i \in [\delta^{(c)}, 1-\delta^{(c)}]$, set $S_j^{(c)}(i) = S_j^{(c)}(i-1) + S_{ji}$.
   (Otherwise set $S_j^{(c)}(i) = S_j^{(c)}(i-1)$.)
   (e) For values of c such that $p_i \in [\delta^{(c)}, 1-\delta^{(c)}]$, set $t^{(c)}(i) = t^{(c)}(i-1) + 1$.
   (Otherwise set $t^{(c)}(i) = t^{(c)}(i-1)$.)
   (f) Disconnect all active users j with $S_j^{(c)}(i) > Z^{(c)}$ and $t^{(c)}(i) \leq \ell^{(c)}$ for some c.

As was already mentioned in Section 3.1, if desired the codeword generation can be merged with the distribution/detection/accusation phase. This depends on the scenario and the exact implementation of the scheme.

Also note that in the above construction, no bounds on i and c are given. If we run the above scheme forever, we will eventually catch any coalition of any size. In practice one may choose to use some upper bound c_0 on the size of any coalition (in the worst case: $c_0 \leq n$), and only keep scores for $2 \leq c \leq c_0$. In that case, drawing values p_i from $F_{\delta^{(c_0)}}$ also makes more sense, as values $p_i \notin [\delta^{(c_0)}, 1-\delta^{(c_0)}]$ would be disregarded for all $2 \leq c \leq c_0$. So while the construction given above is the theoretical solution to catch arbitrary large coalitions, we can also construct an efficient scheme to catch coalitions of size at most c_0 from the above scheme by making these small adjustments.

5.2 Soundness

For the universal Tardos scheme we prove the following result regarding soundness.

Theorem 3. *Consider the universal Tardos scheme in Subsection 5.1. If* (S') *is satisfied for each set of parameters* $d_z^{(c)}, d_\ell^{(c)}, d_\delta^{(c)}, \varepsilon_1^{(c)}$*, and if the* $\varepsilon_1^{(c)}$ *satisfy the following requirement:*

$$\sum_{c=2}^{\infty} \varepsilon_1^{(c)} \leq \varepsilon_1, \tag{E}$$

then the scheme is ε_1*-sound.*

Proof. For each $c \geq 2$, let $\hat{C}^{(c)}$ be the set of users that are accused because their scores $S_j^{(c)}$ exceeded $Z^{(c)}$ before $t^{(c)} > \ell^{(c)}$. Then $\hat{C} = \bigcup_{c=2}^{\infty} \hat{C}^{(c)}$. For any fixed c, we can then apply Theorem 1 to the parameters $d_\ell^{(c)}, d_z^{(c)}, d_\delta^{(c)}, a^{(c)}$ and $\varepsilon_1^{(c)}$ so that we know that the probability that $j \in \hat{C}^{(c)}$ for innocent users j is at most $\varepsilon_1^{(c)}/n$. So the probability that an innocent user is disconnected is then bounded from above by

$$P(j \in \hat{C}) \leq \sum_{c=2}^{\infty} P(j \in \hat{C}^{(c)}) \leq \sum_{c=2}^{\infty} \frac{\varepsilon_1^{(c)}}{n} \leq \frac{\varepsilon_1}{n}.$$

The result then follows. □

Note that one can choose values $\varepsilon_1^{(c)}$ satisfying (E) such that $O(c^2 \ln(n/\varepsilon_1^{(c)})) = O(c^2 \ln(n/\varepsilon_1))$, e.g. by taking $\varepsilon_1^{(c)} = 6\varepsilon_1/(\pi^2 c^2)$. If furthermore $c^2 = o(n)$, then it also follows that $d_\ell c^2 \ln(n/\varepsilon_1^{(c)}) = d_\ell c^2 \ln(n/\varepsilon_1)(1+o(1))$ and we achieve the same asymptotic codelength as in the static and dynamic Tardos schemes.

5.3 Dynamic completeness

Theorem 4. *Consider the universal Tardos scheme in Subsection 5.1. If* (C') *is satisfied for each set of parameters* $d_z^{(c)}, d_\ell^{(c)}, d_\delta^{(c)}, \varepsilon_1^{(c)}, \eta^{(c)}$*, where* $\eta^{(c)} = \ln(1/\varepsilon_2)/\ln(n/\varepsilon_1^{(c)})$*, then for each* $c \geq 2$ *the scheme is dynamic* (ε_2, c)*-complete.*

Proof. This follows directly from applying Theorem 2 to $d_\ell^{(c)}, d_z^{(c)}, d_\delta^{(c)}, a^{(c)}$ and $\varepsilon_1^{(c)}$, where c is the actual (unknown) coalition size. □

To prove that the scheme catches a coalition of size c, we only argue that the coalition's score $S^{(c)}(i)$ will exceed $cZ^{(c)}$ before we have seen $\ell^{(c)}$ positions with $\delta^{(c)} \leq p_i \leq 1-\delta^{(c)}$. In reality, the probability of catching the coalition is much larger than this, since e.g. with high probability their score $S^{(c-1)}$ will also exceed $Z^{(c-1)}$ before we have seen $\ell^{(c-1)}$ positions with $p_i \in [\delta^{(c-1)}, 1-\delta^{(c-1)}]$. And if a pirate is disconnected because his score exceeded some threshold $Z^{(k)}$, then we do not have to wait until $S(i) > c\tilde{Z}^{(c)}$ but only until $S(i) > (c-1)Z^{(c)} + Z^{(k)}$. And since $S(i)/c$ has a constant slope, as soon as a pirate is caught, the other pirates' scores will increase even faster. In practice we therefore also see that we usually need fewer than $\ell^{(c)}$ useful positions to catch a coalition of size c.

5.4 Codelengths

The theoretical results from the previous subsections are not for exactly $\ell^{(c)}$ watermark positions, but for some number of symbols $T^{(c)}$ such that there are $\ell^{(c)}$ positions i with $p_i \in [\delta^{(c)}, 1-\delta^{(c)}]$ by then. The difference $T^{(c)} - \ell^{(c)}$ is a random variable, and is distributed according to a negative binomial distribution with parameters $r = \ell^{(c)}$ (the number of successes we are waiting for) and $p = 1 - P(p_i \in [\delta^{(c)}, 1-\delta^{(c)}]) = \frac{4}{\pi}\arcsin(\sqrt{\delta^{(c)}})$ (the probability of a success). Because the parameter $p = O(c^{-2/3})$ is very small for large c, the difference between $T^{(c)}$ and $\ell^{(c)}$ will also be small. More precisely, $T^{(c)}$ has mean $\ell^{(c)}/(1-p) = \ell^{(c)}(1+O(c^{-2/3}))$ and variance $\sigma^2 = \ell^{(c)}p/(1-p)^2 = O(\ell^{(c)}c^{-2/3})$, and the probability that $T^{(c)}$ exceeds its mean by $m > 0$ decreases exponentially in m.

Also note that if some upper bound $c_0 \geq c$ is used for constructing the scheme as described earlier, then since the values of p_i are drawn from $F_{\delta^{(c_0)}}$ we have $T^{(c_0)} = \ell^{(c_0)}$, as no values of p_i are disregarded for $c = c_0$. So then the maximum codelength is fixed in advance, at the cost of possibly not catching coalitions of size $c > c_0$.

Finally, note that this scheme is constructed in such a way that coalitions of any (small) size can be caught more efficiently. To catch a coalition of size c we now only use $O(c^2 \ln(n/\varepsilon_1))$ symbols. This in comparison to the static and dynamic Tardos scheme, where we need $O(c_0^2 \ln(n/\varepsilon_1))$ and $O(cc_0 \ln(n/\varepsilon_1))$ symbols respectively, where c_0 is again some upper bound on the coalition size used to construct the schemes. So while using the dynamic Tardos scheme already reduces the codelength by a factor $\frac{c}{c_0}$, the universal Tardos scheme further reduces the codelength by another factor $\frac{c}{c_0}$.

5.5 Example

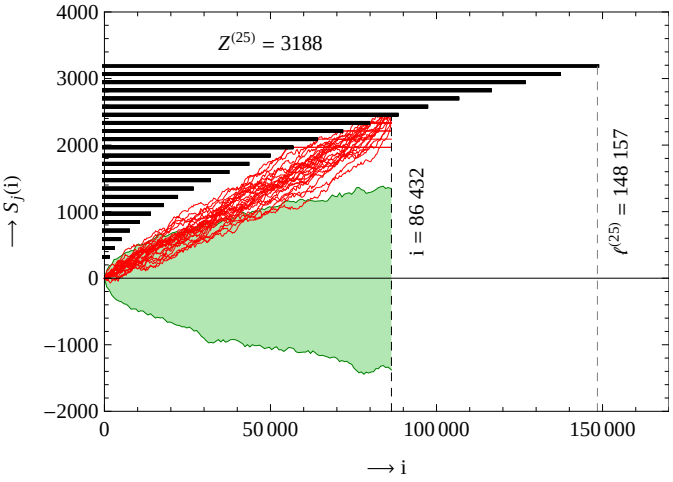
As before, let the scheme parameters be given by $c = 25$, $n = 10^6$ and $\varepsilon_1 = \varepsilon_2 = 10^{-3}$. Let us take $c_0 = 25$, and use $F_{\delta^{(c_0)}}$ for generating values of p_i. Let us also use $\varepsilon_1^{(c)} = 6\varepsilon_1/(\pi^2 c^2)$, so that $\sum_{c=2}^{c_0} \varepsilon_1^{(c)} \leq \sum_{c=2}^{\infty} \varepsilon_1^{(c)} \leq \varepsilon_1$. The optimal values of $d_\ell^{(25)}, d_z^{(25)}, d_\delta^{(25)}$ satisfying (S') and (C') can be calculated numerically as

$$d_\ell^{(25)} = 8.59, \quad d_z^{(25)} = 4.61, \quad d_\delta^{(25)} = 13.83.$$

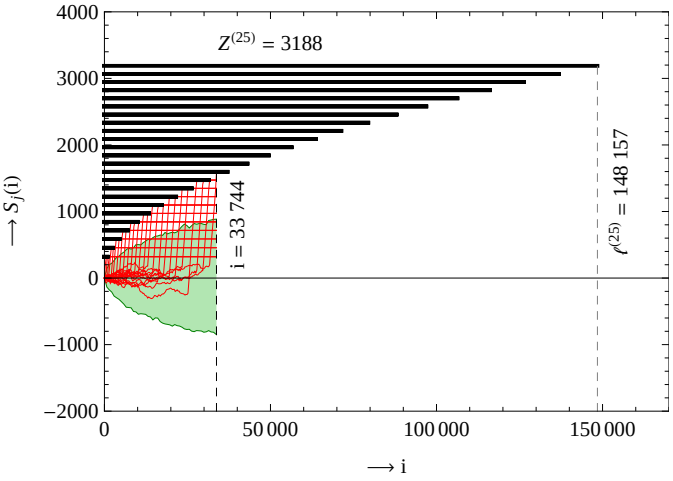
For $c = 25$, this leads to the scheme parameters

$$\ell^{(25)} = 148\,457,\ Z^{(25)} = 3188,\ \delta^{(25)} = 9.89 \cdot 10^{-4}.$$

In Fig. 5 we show some simulation results for these parameters. In Fig. 5a we simulated pirates using the interleaving attack, and in Fig. 5b the pirates used the scapegoat strategy. As one can see, in the universal Tardos scheme the scapegoat strategy is not a good strategy, as the whole coalition is caught sooner.



(a) Interleaving attack



(b) Scapegoat strategy

Figure 5: Simulations of the universal Tardos scheme, with parameters $c = 25$, $n = 10^6$ and $\varepsilon_1 = \varepsilon_2 = 10^{-3}$. In Fig. 5a the pirates used the interleaving attack, whereas in Fig. 5b they used the scapegoat strategy. For each pirate j we only show the score $S_j^{(c)}(i)$ which got him disconnected.

6 Discussion

Comparing the universal scheme to the static scheme, we see that the main advantages are that (a) we now have certainty about catching the whole coalition instead of at least one pirate, and (b) we no longer need the coalition size c, or a sharp upper bound on the coalition size as input. We do need to keep multiple scores per user, namely one for each possible coalition size c, so therefore in practice one will in fact use some upper bound c_0 on the coalition size, and keep scores for values c with $2 \leq c \leq c_0$ for some c_0. But since the only disadvantage of a large c_0 is a large number of scores per user (which may not be an issue), c_0 can easily be much higher than the expected coalition size. This in contrast to the static and dynamic Tardos schemes, where an increase in c_0 means an increase in ℓ as well.

In Table 1 we list some of the differences between the static, dynamic, semi-dynamic and universal Tardos schemes. Here we assume that the upper bound c_0 on the number of colluders is the same for each scheme. The actual coalition size is denoted by c. The example referred to in the table is the example used throughout this paper, with $c = c_0 = 25$, $n = 10^6$, and $\varepsilon_1 = \varepsilon_2 = 10^{-3}$. The practical codelengths are based on 1000 simulations for each scheme, where the pirates used the interleaving attack in all cases. For the semi-dynamic Tardos scheme we used $B = 8$ in our example.

Since the scheme is a dynamic traitor tracing scheme, it also makes sense to compare it to other dynamic traitor tracing schemes from literature. The deterministic scheme of Fiat

Table 1: A comparison of the Tardos schemes discussed in this paper.

	static (Section 2)	dynamic (Section 3)	semi-dynamic (Section 4.1)	semi-dynamic (Section 4.2)	universal (Section 5)
scores per user	1	1	1	1	c_0
density function	$f^{(c_0)}$	$f^{(c_0)}$	$f^{(c_0)}$	$f^{(c_0)}$	$f^{(c_0)}$
blocks	1 of size ℓ	ℓ of size 1	ℓ/B of size B	ℓ/B of size B	ℓ of size 1
guilty caught	at least 1	all c	all c	all c	all c
expected codelength	$O(c_0^2\ln(n/\varepsilon_1))$	$O(cc_0\ln(n/\varepsilon_1))$	$O(cc_0\ln(n/\varepsilon_1))$	$O(cc_0\ln(n/\varepsilon_1))$	$O(c^2\ln(n/\varepsilon_1))$
asymptotic codelength	$\frac{\pi^2}{2}c^2\ln(n/\varepsilon_1)$	$\frac{\pi^2}{2}c^2\ln(n/\varepsilon_1)$	$\frac{\pi^2}{2}c^2\ln(n/\varepsilon_1)$	$\frac{\pi^2}{2}c^2\ln(n/\varepsilon_1)$	$\frac{\pi^2}{2}c^2\ln(n/\varepsilon_1)$
example, theoretical codelength	109 585	116 561	116 761	131 587	148 457
example, practical codelength	109 585	92 000	92 000	96 000	89 000

and Tassa from [4] has a codelength of $\ell = O(c\log_2(n))$ but an alphabet size of $q = 2c+1$. In [1], Berkman et al. studied deterministic schemes with slightly smaller alphabet sizes ($q = c+1$) leading to an algorithm using $\ell = O(c^2 + c\log_2(n))$ symbols, with "large hidden constants". Finally, in [9] Tassa investigated a probabilistic binary ($q = 2$) dynamic traitor tracing scheme, but his codelengths $\ell = O(c^4\log_2(n)\ln(n/\varepsilon_1))$ are more than a factor $O(c^2)$ larger than the codelengths of the universal Tardos scheme. Furthermore, if we look at other properties of these dynamic schemes, we see that the universal Tardos scheme has some nice properties. Most of these properties are inherited from the static Tardos scheme, and we list some of them below.

Symbols are always taken from a small alphabet. This is an advantage compared to the schemes of Fiat and Tassa [4] and Berkman et al. [1], which used alphabets of size $q = 2c+1$ and $q = c+1$ respectively.

Codewords of users are independent. This means that framing a specific user is basically impossible, as the codewords of the pirates and the pirate output are independent of the innocent users' codewords. Also, a new user can be added to the system easily after the codewords of other users have already been generated, since the codewords of other users then do not have to be updated.

Codewords positions are independent. In other words, the scheme does not make use of the information obtained from previous pirate output for generating new symbols for each user. Therefore the codewords can all be generated in advance, and can even be stored at the client side. The security solely depends on the marking assumption, so pirates are even allowed to learn the values of p_i. Furthermore this also allows us to effectively tackle semi-dynamic traitor tracing scenarios, as described in Section 4, which is impossible for Fiat and Tassa's scheme and Berkman et al.'s schemes. Without making changes to their schemes, their codelengths would simply increase by a factor B.

The distribution of watermark symbols is identical for each position. This property offers new options, like tracing several coalitions simultaneously, using the same traitor tracing code. This also means that multiple watermarks from several broadcasts can be concatenated and viewed as one long watermark from one longer broadcast, allowing one to catch large coalitions with multiple watermarked broadcasts.

The codeword generation and accusation algorithm are computationally and memory-wise efficient. Our scheme does not require any complicated data structures, and the only memory needed during the broadcast is the scores for each user at that time, and the counters $t^{(c)}$. During the broadcast only simple calculations are needed: computing S_{ji} (which has to be calculated only once), adding S_{ji} to those scores $S_j^{(c)}$ where c satisfies a certain condition, and comparing $S_j^{(c)}$ against $Z^{(c)}$.

Several instances of the scheme with different parameters ε_1 can be run simultaneously. For example, by using parameters $\{\varepsilon_1^{(c)}\}$ with $\sum\varepsilon_1^{(c)} \le 0.01$ and $\{\overline{\varepsilon}_1^{(c)}\}$ with $\sum\overline{\varepsilon}_1^{(c)} \le 0.05$ for two different instances of the universal Tardos scheme (using the same codewords), a pirate will first cross one of the thresholds associated to $\{\overline{\varepsilon}_1\}$, and only later cross one of the thresholds associated to the $\{\varepsilon_1\}$. If we use the $\{\varepsilon_1\}$ for disconnecting users, then even before a user is disconnected, we can give some sort of statistic to indicate the 'suspiciousness of this user. If a user then does not cross the highest thresholds, one could still decide whether to disconnect him or not. After all, the choice of ε_1 may be a bit arbitrary, and a user which almost but not quite crosses the thresholds $Z^{(c)}$ is probably guilty as well.

7 Open problems

The universal Tardos scheme has two minor drawbacks. First, we have to keep multiple scores for each user, namely for each possible coalition size c, and secondly, part of the data is disregarded for each value of c. To address these issues, one could for example try adjusting the universal Tardos scheme, or start from the dynamic Tardos scheme and build a new, c-independent traitor tracing scheme. Below we present some ideas, which seem to work in practice, but for which rigorously proving soundness and completeness seems hard.

7.1 The queue Tardos scheme

One possible modification is the following. Instead of using "bad" values of p_i early on in the broadcast, i.e. using values of p_i which are not between $\delta^{(c)}$ and $1-\delta^{(c)}$ at positions $i \in \{1,\dots,\ell^{(c)}\}$, we temporarily store those values of p_i in queues $Q_3, Q_4, \dots$, where p_i gets stored in queue Q_c iff $p_i \in (\delta^{(c)}, \delta^{(c-1)}]$ or $p_i \in [1-\delta^{(c-1)}, 1-\delta^{(c)})$. After $\ell^{(c)}$ positions, for $c \ge 2$ we then empty the queue Q_{c+1} by using these values

as the next values of p_i for $i \in \{\ell^{(c)}+1,\ldots,\ell^{(c)}+|Q_{c+1}|\}$. Doing this for all values of c, we get that all p_i, for $i \in \{0,\ldots,\ell^{(c)}\}$, are between $\delta^{(c)}$ and $1-\delta^{(c)}$. So instead of disregarding data and using multiple scores, we can just use the traditional dynamic Tardos scoring system on all positions sequentially. This scheme seems to work well in practice, but proving soundness and completeness seems hard. A further possible drawback is the fact that the p_i are now no longer universally distributed according to F for each $i \geq 1$, which makes it harder to efficiently concatenate multiple codes or catch multiple coalitions simultaneously.

7.2 The staircase Tardos scheme

Similar to the above, one could try abolishing the whole queue-system, and simply generate p_i according to density functions in such a way that the overall distribution on $[1,\ell^{(c)}]$ is $f^{(c)}$ for each c. First, we simply use $f^{(1)}$ for $i \in \{1,\ldots,\ell^{(2)}\}$. Now to get an "average" distribution of $f^{(c)}$ on the whole interval $i \in \{1,\ldots,\ell^{(c)}\}$ for each $c \geq 2$, we use $\tilde{f}^{(c)} = (\ell^{(c)} f^{(c)} - \ell_{(c-1)} f^{(c-1)})/(\ell^{(c)} - \ell^{(c-1)})$ as the density function for positions $i \in \{\ell^{(c-1)}+1,\ldots,\ell^{(c)}\}$, so that $\ell^{(c)} f^{(c)} = \ell^{(c-1)} f^{(c-1)} + (\ell^{(c)} - \ell^{(c-1)})\tilde{f}^{(c)}$. For $c \geq 3$ we see that $\tilde{f}^{(c)}(p) \geq 0$ for all p, so that these functions are indeed proper probability density functions. This scheme seems to work in practice as well, but to prove completeness seems hard. Using the standard proof method does not work here, as for that we need that all p_i are generated using the density function $f^{(c)}$. And again, a possible drawback of this scheme is that the distribution of the values of p_i is not identical for each i, and so concatenating multiple codes may not be possible anymore.

7.3 The continuous Tardos scheme

Looking at Fig. 5 suggests that a continuous threshold function $Z(i)$ might also be an option, with Z depending on the position i instead of on the possible coalition size c. However, for the proof of soundness of the universal Tardos scheme we simply added up the error probabilities for each threshold, and by scaling each error probability with a factor $1/c^2$, the total soundness error probability could be bounded by ε_1. If we use a continuous $Z(i)$ and use this same proof method, we would have to add up $O(c^2)$ of these error probabilities, leading to even smaller values of $\varepsilon^{(i)}$ and slightly longer codelengths. Theoretically it would be interesting to see if such a continuous threshold function is feasible.

7.4 The q-ary dynamic Tardos scheme

Finally, one can try to generalize these results to higher alphabet sizes $q > 2$. In this paper we only used the binary alphabet, since we could then use the results from [6]. One could try to give a similar analysis for higher alphabet sizes, possibly by using results from [7]. This could lead to even shorter codelengths with smaller constants d_ℓ, although the codelength will still satisfy $\ell = O(c^2 \ln(n/\varepsilon_1))$.

References

[1] O. Berkman *et al.*, "Efficient Dynamic Traitor Tracing," *SIAM J. Comput.*, vol. 30, no. 6, pp. 1802–1828, 2001.

[2] O. Blayer and T. Tassa, "Improved Versions of Tardos' Fingerprinting Scheme," *Des. Codes Cryptogr.*, vol. 48, no. 1, pp. 79–103, 2008.

[3] D. Boneh and J. Shaw, "Collusion-Secure Fingerprinting for Digital Data," *IEEE Trans. Inform. Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[4] A. Fiat and T. Tassa, "Dynamic Traitor Tracing," *J. Cryptology*, vol. 14, no. 3, pp. 211–223, 2001.

[5] T. Laarhoven, "Collusion-Resistant Traitor Tracing Schemes," M.Sc. thesis, Dept. Math. Comp. Sc., Eindhoven Univ. Techn., Eindhoven, The Netherlands, July 2011.

[6] T. Laarhoven and B. de Weger, "Optimal Symmetric Tardos Traitor Tracing Schemes," submitted for publication. Available: http://arxiv.org/abs/1107.3441.

[7] B. Škorić *et al.*, "Symmetric Tardos Fingerprinting Codes for Arbitrary Alphabet Sizes," *Des. Codes Cryptogr.*, vol. 46, no. 2, pp. 137–166, 2008.

[8] G. Tardos, "Optimal Probabilistic Fingerprint Codes," *Proc. 35th ACM Symp. on Theory of Computing*, 2003, pp. 116–125.

[9] T. Tassa, "Low Bandwidth Dynamic Traitor Tracing Schemes," *J. Cryptology*, vol. 18, no. 2, pp. 167–183, 2005.