Making Collusion-Secure Codes (More) Robust against Bit Erasure

Koji Nuida

Research Center for Information Security (RCIS), National Institute of Advanced Industrial Science and Technology (AIST), Akihabara-Daibiru Room 1003, 1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan

E-mail: k.nuida@aist.go.jp

Abstract

A collusion-secure code is called robust if it is secure against erasure of a limited number of undetectable bits, in addition to collusion attacks under Marking Assumption. In this article, we propose the first general conversion method of (non-robust) c-secure codes to robust c-secure codes. Also, the same method amplifies robustness of given robust c-secure codes. By applying our conversion to c-secure codes given by Nuida et al. (AAECC 2007), we present robust c-secure codes with code length of order $\Theta(c^2 \log^2 c)$ with respect to c. This code length improves preceding results by Sirvent (WCC 2007) and by Boneh and Naor (ACM CCS 2008) and is close to the one by Billet and Phan (ICITS 2008), although our construction is based on a weaker assumption than those preceding results. As an application, applying our resulting code to construction by Boneh and Naor also improves their traitor tracing scheme against imperfect decoders in efficiency of both key sizes and pirate tracing procedure.

Keywords: Collusion-secure code, bit erasure tolerance, general conversion, digital fingerprinting, traitor tracing scheme

1 Introduction

1.1 Background

Recently, digital content distribution services have been widespread by virtue of progress of information technology. Digitization of content distribution has improved convenience for ordinary people. However, the digitization also enables malicious persons to perform stronger attacks, therefore the number of illegal content redistribution is increasing very rapidly. Hence technical countermeasures for such illegal activities are strongly desired.

Digital fingerprinting is a possible solution for such problems. Among several possible approaches, in this article we focus on code-based digital fingerprinting schemes. Namely, we suppose that a content provider

first encodes each user's identification data and then embeds each codeword as a fingerprint into a copied content that will be sent to the corresponding user. Such a scheme aims at tracing, when unauthorized copies of the content are distributed, the adversarial user (called *pirate*) from the fingerprint word embedded in the pirated content (called *pirated word*). However, it is known that a coalition of two or more pirates would be able to perform a strong attack to the fingerprint word, called *collusion attack* [5]. Hence any fingerprint code should be equipped with traceability of pirates (with overwhelming probability) against collusion attacks.

For traceability evaluation of fingerprint codes, the security assumption that is most standard so far is the one proposed by Boneh and Shaw [5], called *Marking Assumption*. The motivation of the assumption is as follows. Suppose that in embedding process of fingerprints, first each content is divided into several segments, and then each bit of a fingerprint codeword is embedded into one of the segments (without overlapping). Suppose further that the choices of segments are common to all users' contents. In the situation, pirates can compare their copies of the content with each other to find differences. If their contents contain different bits at some position, then some difference will be detected there, which tells the pirates that the segment contains a bit of the fingerprint (such a position is referred to as *detectable*). On the other hand, if their contents contain the same bit at some position, then no differences will be detected there and pirates will obtain no information on this bit of the fingerprint word (such a position is referred to as *undetectable*). Now the Marking Assumption states that any bit in an undetectable position will be left unchanged in the pirated word, while any bit in a detectable position can be freely selected or even erased. A fingerprint code is called *c-secure* [5] if at least one of the pirates is traceable under Marking Assumption with overwhelming probability provided the number of pirates does not exceed a given constant c. See Section 2 for precise definitions. The first concrete instance of c-secure codes was given by Boneh and Shaw themselves [5], and several construction of c-secure codes have been proposed so far, such as [3, 7, 8, 11, 15, 16, 17].

However, one may feel that the Marking Assumption is somewhat impractical, as the requirement of perfect protection of undetectable positions would be far beyond the ability of state-of-the-art fingerprint embedding schemes. Thus various relaxation of Marking Assumption to allow some attacks to undetectable positions have been introduced, and several c-secure codes under those assumptions, called *robust c-secure codes*, have been proposed, for example, in [9, 12, 13]. Recently, such robust c-secure codes are also studied in connection with traitor tracing schemes against pirates with powerful decoders [14] or imperfect decoders [4]. Hence robust c-secure codes are important in both theoretical and practical viewpoints.

In general, the existing constructions and security proofs of robust c-secure codes tend to require further intricate and scheme-specific arguments, even if the construction is based on some preceding (non-robust) c-secure codes. This tendency seems stronger for c-secure codes with combinatorial construction (such as [7, 8, 17]) or with a flavor of "joint decoding" [1] (such as [3, 11]). Hence novel ideas of reducing such

difficulty in studies of robust c-secure codes are desired.

1.2 Our Contribution and Some Relevant Remarks

In this article, we present the first general conversion method of any c-secure code to a robust c-secure code. The same method can also amplify robustness of a given robust c-secure code. Our conversion is very simple and is applicable to c-secure q-ary codes (such as [13, 15]) as well as binary ones. Our conversion is a "black-box" algorithm, in the sense that our conversion requires *no* knowledge of specific properties for the target c-secure code, except a relation between code length and error probability.

The essential idea of our conversion method is explained as follows. Here we focus on binary codes for simplicity. First, our relaxation of Marking Assumption says that at most a limited number (that is δ-fraction of the total code length, where δ is a parameter) of undetectable bits may be erased. The case $\delta = 0$ coincides with the original Marking Assumption. For simplicity, here we assume that the target c-secure code $\mathcal{C}$ is not robust. To resist erasure of some undetectable bits, our conversion method first expands each bit in $\mathcal{C}$ to a block of b identical bits, and then appends L common dummy bits to every codeword. Moreover, the distribution of the "undetectable blocks" (that is, blocks originated from undetectable bits in $\mathcal{C}$) is concealed from the pirates by using a (common and secret) random permutation and (common and secret) random bit flippings. In the situation, when a pirated word for the expanded code is given, even if a part of an undetectable block was erased, the original undetectable bit can be still recovered provided at least one bit in the block survives. Moreover, by choosing sufficiently large block size b and number L of dummy bits, it becomes unlikely that all bits in an undetectable block are erased. Thus a valid pirated word for the original code is obtained with overwhelming probability, therefore the resulting code is equipped with the desired robustness (see Theorem 1 for details).

We also investigate appropriate parameters b and L (see Theorem 2). By using the result, we describe the asymptotic behavior of code lengths of the resulting robust c-secure codes in terms of those of the original c-secure codes. Moreover, by adopting the (less) robust c-secure codes proposed by Nuida et al. [12] as the target code, we obtain robust c-secure binary codes (for arbitrary parameters $0 < \delta < 1$) with code lengths m satisfying

$$m \sim 21.41244 \left(\frac{c \log c}{1 - \delta} \right)^2 \log(N/\varepsilon) \ , \tag{1}$$

where N is the number of users, ε is the error probability and log denotes the natural logarithm (see Theorem 3). Comparing with the tight lower bound $\Omega(c^2 \log(N/\varepsilon))$ of code lengths of c-secure codes given by Tardos [16], it seems that our code length is of "nearly optimal" order. The constant factor ≈ 21.4 is also not huge; for example, the factor is 100 for Tardos codes [16].

We give a remark on "efficiency" of our conversion. For example, for a certain choice of parameters, the original robust 2-secure code with parameter $\delta = 0.005$ has about 400-bit length, while the robust 2-secure

code with $\delta = 0.5$ generated by our conversion has about $177,000$-bit length, which is about 440 times as long as the original (see Section 5 for details). However, when we implement those codes by embedding them into a digital content, embedding one bit with bit erasure rate (in undetectable positions) 0.5% is very likely to need much more redundancy than embedding one bit with the bit erasure rate 50%. Comparison of efficiency of two robust c-secure codes with different robustness parameters δ *in implementation* would be a challenging research topic, which is beyond the scope of this article.

To be honest, there remain some rooms for improving the results of this work. First, our conversion method can be thought of as concatenating the target c-secure code with a repetition code (together with some dummy digits); hence it is likely that the use of more sophisticated erasure codes can improve the efficiency, and that the use of better error-correcting codes can extend our method to a further relaxed situation (discussed in, for example, [12]) where some bit flipping (as well as bit erasure) may occur in undetectable positions. Secondly, the security evaluation in this article is not fully optimized, therefore a more detailed analysis would be able to improve the choice of parameters (such as the code length). However, this article aims at pioneering, by exhibiting simple but concrete ideas, the study of general conversion methods from c-secure codes to (more) robust ones. Therefore, we leave the above-mentioned issues as open problems, which would need further complicated arguments.

1.3 Related Works

As mentioned in Section 1.1, there have been proposed various relaxation of the Marking Assumption. Guth and Pfitzmann [9] considered the situation that each undetectable bit is erased *independently* with a certain probability, and they extended Boneh-Shaw codes [5] to their assumption. (Safavi-Naini and Wang [13] considered the same assumption for q-ary codes.) Though there seems no overall implication from one of our assumption and theirs to the other, our assumption would look weaker due to the lack of bitwise independence condition in their assumption.

Recently, Sirvent [14], Billet and Phan [2], and Boneh and Naor [4] considered another assumption that is more relevant to ours. Their assumption does not require the bitwise independence of erasure as well as ours, but their assumption restricts the number of erasure in *arbitrary* positions (see, for example, Section 4.1 of [4]), not only the number in *undetectable* positions as our assumption does. Hence our assumption is readily weaker than theirs. In [4, 14], they extended Boneh-Shaw codes to their assumption, with resulting code lengths $m = \Theta(c^4 \log(N/\varepsilon) \log(c^2 \log(N/\varepsilon)/\varepsilon))$ in [14] (where its dependence on parameter δ seems not clarified) and $m = \Theta((N^3/(1-\delta)^2) \log(2N/\varepsilon))$ in [4] for the full-collusion case $c = N$. On the other hand, in [2], they extended Tardos codes [16] to their assumption, with resulting code lengths $m = \Theta((c^2/(1-\delta)) \log(N/\varepsilon))$. Despite that our assumption is weaker than their assumption, our code length in (1) (with $c = N$ when comparing with [4]) is significantly more efficient than [4, 14] and is even close to [2]. Hence

by replacing the extended Boneh-Shaw code used in the traitor tracing scheme in [4] with our code, their scheme is improved in efficiency of both key size and pirate tracing procedure.

On the other hand, Nuida et al. [12] considered another relaxation of Marking Assumption, which bounds the number of undetectable bits *that are either erased or flipped* by δ fraction of the total code length. Their assumption is thus weaker than ours, and their δ-robust c-secure codes have code lengths $m = \Theta(c^2 \log(N/\varepsilon))$ that are shorter than (1). However, in their scheme the parameter δ is restricted to be far from 1, namely $\delta = O(c^{-2})$ (see Section 6.1 of [12]), while in our scheme δ can be arbitrarily close to 1. To extend our conversion method to their weaker assumption would be an interesting future research topic.

1.4 Notations and Terminology

Here we clarify some notations and terminology. In this article, log denotes the natural logarithm. The expression "$x \to x_0$" means "x converges to x_0" (or "x diverges to x_0", when $x_0 = \pm\infty$). For $i, j \in \mathbb{Z}$, the lower factorial $(i)_j$ is defined by $(i)_j = i(i-1)\cdots(i-j+1)$. The symbols $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the largest $M \in \mathbb{Z}$ with $M \leq x$ and the smallest $M \in \mathbb{Z}$ with $M \geq x$, respectively. Moreover, $\Sigma_q = \{s_0, s_1, \ldots, s_{q-1}\}$ denotes a q-ary alphabet (where $\Sigma_2 = \{0, 1\}$), and for $s \in \Sigma_q$ and $j \in \mathbb{Z}$, the expression "rotate s by j" means to convert $s = s_h \in \Sigma_q$ into $s_i \in \Sigma_q$, where $i \equiv h + j \pmod{q}$. We put $[n] = \{1, 2, \ldots, n\}$ for an integer n, and define $2^{[n]}$ and $\binom{[n]}{k}$ to be the sets of all subsets of $[n]$ and of all k-element subsets of $[n]$, respectively.

1.5 Organization of the Article

This article is organized as follows. Section 2 summarizes the notion of fingerprint codes and our relaxed Marking Assumption, called δ-Marking Assumption. Section 3 presents the main theorems of this article. Section 4 gives the proofs of main theorems, where some intricate part will be postponed until the appendix. Finally, Section 5 shows some numerical examples of our results.

2 Robust Fingerprint Codes

In this section, we summarize formal definitions regarding fingerprint codes, including security assumptions. One may feel that our formulation looks different from the conventional one due to the slight generality, however the conversion between the two is easy.

We start with the following definition:

Definition 1. *Given the q-ary alphabet Σ_q, integer parameters $1 \leq c \leq N$, and an "evaluation function" $\mathsf{ev} : 2^{[N]} \times \bigcup_{\ell=1}^{c} \binom{[N]}{\ell} \to \{0, 1\}$ (see Section 1.4 for notations), we define the following game, which we refer*

to as pirate tracing game. *The players of the game is a* provider *and* pirates*, and the game is proceeded as follows:*

1. Provider *generates an* $N \times m$ *q-ary matrix* $W = (w_{i,j})_{1 \le i \le N, 1 \le j \le m}$ *and an element* st *called* state information.
2. Pirates *generate* $C \subset [N]$*,* $1 \le |C| \le c$*, without knowing* W *and* st*.*
3. Pirates *receive the codewords* $w_i = (w_{i,1}, \ldots, w_{i,m})$*,* $i \in C$*.*
4. Pirates *generate a word* $y = (y_1, \ldots, y_m)$ *on* $\Sigma_q \cup \{?\}$ *under a restriction specified below, and send* y *to* provider*.*
5. Provider *generates* $\mathsf{Acc} \subset [N]$ *from* y*,* W*, and* st *(without knowing* C*).*
6. *Then* pirates *win if* $\mathsf{ev}(\mathsf{Acc}, C) = 1$*, and* provider *wins if* $\mathsf{ev}(\mathsf{Acc}, C) = 0$*.*

In the above formulation, the symbol '?' signify erasure of a digit in the word y. Let Gen, Reg, ρ, and Tr denote the algorithms used in the steps 1, 2, 4, and 5 in the above game, respectively. We call the algorithms Gen, Reg, ρ, and Tr *codeword generation algorithm*, *registration algorithm*, *pirate strategy*, and *tracing algorithm*, respectively. An intuitive meaning of the function ev is that ev returns 0 if the tracing of the provider "succeeded" and it returns 1 if the tracing "failed". In the usual setting, ev should be defined to return 0 if and only if $\emptyset \neq \mathsf{Acc} \subset C$. However, the above formulation enables us to consider more general situations as well. For example, in some preceding schemes [10, 16] false accusation of innocent users does not occur even under a very weak assumption. In our formulation, we can deal with such false accusation alone by defining ev to return 0 if and only if $\mathsf{Acc} \subset C$. On the other hand, an example of state information st in our formulation is the collection of bias parameters p_i in Tardos codes [16]. The pair $\mathcal{C} = (\mathsf{Gen}, \mathsf{Tr})$ of two algorithms Gen and Tr is called a *fingerprint code*, and the following quantity

$$Pr[(W, \mathsf{st}) \leftarrow \mathsf{Gen}(); C \leftarrow \mathsf{Reg}(); y \leftarrow \rho(C, (w_i)_{i \in C}); \mathsf{Acc} \leftarrow \mathsf{Tr}(y, W, \mathsf{st}) : \mathsf{ev}(\mathsf{Acc}, C) = 1]$$

is called *error probability* of $\mathcal{C}$.

To specify the restriction mentioned in the step 4 above, we introduce some terminology. We call the word y output by ρ a *pirated word*. For $1 \le j \le m$, j-th column in a codeword is called *undetectable* if the j-th digits $w_{i,j}$ of codewords w_i coincide for all $i \in C$; and called *detectable* otherwise. Now we introduce the following two variants of the assumption on the pirate strategy ρ, where $0 \le \delta < 1$ is a parameter (the classification is according to [15]):

Definition 2. *In the above setting, the* δ-Marking Assumption (in unreadable digit model) *states the following: We have* $y_j \in \{w_{i,j} \mid i \in C\} \cup \{?\}$ *for every column, and the number of undetectable columns with* $y_j = ?$ *is not larger than* δm*.*

Definition 3. *In the above setting, the* δ-Marking Assumption (in general digit model) *states the following: We have* $y_j \in \Sigma_q \cup \{?\}$ *for every detectable column, while we have* $y_j \in \{w_{i,j} \mid i \in C\} \cup \{?\}$ *for every undetectable column; and the number of undetectable columns with* $y_j = ?$ *is not larger than* δm.

Our following argument is based on either of the above two assumptions, where which of the two assumptions is adopted is fixed throughout the argument. Note that these two assumptions are identical for binary ($q = 2$) case. Note also that any of the two assumptions with parameter $\delta = 0$ coincides with the conventional Marking Assumption [5]. We say that a fingerprint code $\mathcal{C}$ is *δ-robust c-secure (with ε-error)* if, for any registration algorithm Reg and any pirate strategy ρ satisfying δ-Marking Assumption, the error probability of $\mathcal{C}$ is lower than or equal to ε. When $\delta = 0$, such a $\mathcal{C}$ is called *c-secure (with ε-error)* [5]. The aim of this article is to propose the first general conversion method from given δ_0-robust c-secure codes to δ-robust c-secure codes, where $0 \leq \delta_0 < \delta < 1$.

3 Main Results

This section summarizes the main results of this article. In Section 3.1, we describe our general conversion method and state its validity. In Section 3.2, we propose an appropriate choice of code lengths and relevant parameters for our conversion, and describe the asymptotic behavior of the resulting δ-robust c-secure codes. Proofs of the results will be given in Section 4.

3.1 The Conversion Method

Let $\mathcal{C} = (\mathsf{Gen}, \mathsf{Tr})$ be an arbitrary δ_0-robust c-secure q-ary code with ε_0-error, where $0 \leq \delta_0 < 1$ and $0 \leq \varepsilon_0 < 1$. Let m_0 be the code length of $\mathcal{C}$. We show construction (from $\mathcal{C}$) of a δ-robust c-secure q-ary code $\overline{\mathcal{C}} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Tr}})$ with ε-error, where $\delta_0 < \delta < 1$ and $\varepsilon_0 < \varepsilon < 1$. The value $\varepsilon - \varepsilon_0$ signifies the loss of security through our conversion. The resulting code $\overline{\mathcal{C}}$ has code length $m = bm_0 + L$, where $b \geq 1$ and $L \geq 0$ are integer parameters. First, the new codeword generation algorithm $\overline{\mathsf{Gen}}$ proceeds in the following manner:

1. Perform Gen and obtain the outputs $W = (w_{i,j})_{i,j}$ and st.
2. For every digit $w_{i,j}$ in W, replace it with a block of b digits each of which is identical with $w_{i,j}$.
3. Append L identical digits $s_0 \in \Sigma_q$, called *dummy digits*, to the tail of every word obtained by the previous step. Hence the resulting word has length $m = bm_0 + L$.
4. Choose a secret word $\mathsf{fl} = (\mathsf{fl}_1, \ldots, \mathsf{fl}_m)$, where $\mathsf{fl}_j \in \{0, 1, \ldots, q-1\}$, uniformly at random. Then for every word obtained by the previous step and for every $1 \leq j \leq m$, rotate j-th digit of the word by fl_j (see Section 1.4 for the terminology).

5. Choose a secret permutation perm of m letters $1, \ldots, m$ uniformly at random, and permute the digits of every word obtained by the previous step according to perm (i.e., j-th digit of the word becomes $\mathsf{perm}(j)$-th digit of the resulting word).

6. Output the matrix $\overline{W} = (\overline{w}_{i,j})_{1 \le i \le N, 1 \le j \le m}$ and the corresponding state information $\overline{\mathsf{st}} = (\mathsf{st}, \mathsf{fl}, \mathsf{perm})$, where $\overline{w}_i = (\overline{w}_{i,1}, \ldots, \overline{w}_{i,m})$ is the word obtained from w_i by the steps 2–5.

Intuitively, the steps 2–3 supply sufficient redundancy to the codewords, while the steps 4–5 conceal the internal structure of codewords from the pirates. On the other hand, the new tracing algorithm $\overline{\mathsf{Tr}}$ proceeds in the following manner, where it takes as inputs a pirated word $\overline{y}$ of length m, an $N \times m$ q-ary matrix $\overline{W}$, and a triple $\overline{\mathsf{st}} = (\mathsf{st}, \mathsf{fl}, \mathsf{perm})$:

1. Permute the digits in $\overline{y}$ according to the inverse of perm, and for every $1 \le j \le m$, rotate j-th digit of the resulting word by $-\mathsf{fl}_j$ if and only if the digit is not '?'. Let $\overline{y}^{(1)}$ denote the resulting word.

2. Generate a word $y = (y_1, \ldots, y_{m_0})$ in the following way: For each $1 \le j \le m_0$, put

   - $y_j = x \in \Sigma_q$, if j-th block of $\overline{y}^{(1)}$ (that is, from the $((j-1)b+1)$-th to the (jb)-th digits) contains at least one digit x and no digits different from both x and '?'.
   - $y_j = ?$, otherwise.

3. For every $1 \le i \le N$, permute the digits in $\overline{w}_i$ according to the inverse of perm; remove the last L digits (i.e., the dummy digits) of the resulting word; rotate j-th digit of the resulting word by $-\mathsf{fl}_j$ for every $1 \le j \le bm_0$; and replace the j-th block of the resulting word with its first digit for every $1 \le j \le m_0$. Let $w_i = (w_{i,1}, \ldots, w_{i,m_0})$ denote the resulting word.

4. Perform Tr, with y, $W = (w_{i,j})_{1 \le i \le N, 1 \le j \le m_0}$ and st as inputs, and output what the Tr outputs.

Intuitively, the algorithm $\overline{\mathsf{Tr}}$ first converts the given codewords and pirated word for $\overline{\mathcal{C}}$ to those for $\mathcal{C}$ by reversing the conversion process in $\overline{\mathsf{Gen}}$ and then perform the tracing algorithm for $\mathcal{C}$. If suitable parameters are chosen, the obtained pirated word y for $\mathcal{C}$ will be valid except a sufficiently small probability, hence the overall error probability of $\overline{\mathcal{C}}$ will be bounded by the specified value ε.

In order to prove the security of our conversion, we assume that the parameters satisfy the following condition:

$$\begin{aligned} &L \ge \nu_1 \text{ and } \binom{a}{\nu_2} \frac{\binom{ba+L-b\nu_2}{\nu_1-b\nu_2}}{\binom{ba+L}{\nu_1}} \le \varepsilon - \varepsilon_0 \text{ for every integer } 0 \le a \le m_0 \ , \\ &\text{where } \nu_1 = \lfloor \delta(bm_0 + L) \rfloor \text{ and } \nu_2 = \lfloor \delta_0 m_0 \rfloor + 1 \ . \end{aligned} \tag{2}$$

Some concrete parameters satisfying this condition will be proposed in Section 3.2. Then we have the following result, which will be proven in Section 4.1:

Theorem 1. *In the above situation, the resulting fingerprint code* $\overline{\mathcal{C}} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Tr}})$ *is* δ*-robust* c*-secure with* ε*-error.*

3.2 Code Lengths and Parameters

In this subsection, first we propose an appropriate choice of parameters that satisfy the condition (2). Here we assume for simplicity that

$$\varepsilon_0 = \varepsilon/2$$

for bounds of error probabilities. Now we set

$$b = \left\lceil \frac{\log(m_0/\nu_2) + 1 + \nu_2^{-1}\log(1/\varepsilon_0)}{\log(1/\delta)} \right\rceil \ , \tag{3}$$

$$L = \max\left\{ \left\lceil \frac{b\nu_2}{1-(1-\nu_2/m_0)^{1/b}} \right\rceil - bm_0 + b - 1, \left\lceil \frac{\delta b m_0}{1-\delta} \right\rceil \right\} \ , \tag{4}$$

where $\nu_1 = \lfloor \delta(bm_0 + L) \rfloor$ and $\nu_2 = \lfloor \delta_0 m_0 \rfloor + 1$ as specified in (2). Moreover, in fact when the value of ν_2 is small, some part of our analysis below can be made sharper, therefore the choice of parameters can be improved. More precisely, in the case $\nu_2 = 1$, which means that $\delta_0 = 0$ (i.e., the original c-secure code $\mathcal{C}$ is not robust), we can define the parameter b by

$$b = \left\lceil \frac{\log(m_0/\varepsilon_0)}{\log(1/\delta)} \right\rceil \quad (\text{when } \delta_0 = 0) \ , \tag{5}$$

which is smaller than the parameter in (3) for the case $\nu_2 = 1$. Now we have the following result, which will be proven in Section 4.2:

Theorem 2. *In the above situation, the parameters* b *and* L *satisfy the condition (2) for Theorem 1.*

Hence by Theorem 1, the resulting fingerprint code $\overline{\mathcal{C}}$ of length $m = bm_0 + L$, with the above parameters, obtained by our conversion method indeed becomes δ-robust c-secure with ε-error.

For the above choice of parameters, we study the asymptotic behavior of the code length m of $\overline{\mathcal{C}}$. In what follows, we consider (even implicitly) sequences of δ_0-robust c-secure codes $\mathcal{C}$ with ε_0-error and of the corresponding δ-robust c-secure codes $\overline{\mathcal{C}}$ with ε-error (with various parameters), rather than an individual instance of them. We consider the asymptotic behavior in the limit case $c \to \infty$, $N/\varepsilon \to \infty$, and $\delta \to 1$. We may assume without loss of generality that the parameter δ_0 converges to a constant d with $0 \le d \le 1$, by applying Bolzano-Weierstrass Theorem (which implies that any infinite sequence of real numbers in a finite interval has a convergent subsequence) to the sequence of the parameter δ_0, $0 \le \delta_0 < 1$. Moreover, we assume further that $d < 1$ to simplify our argument. Note that $m_0 = \Omega(c^2 \log(N/\varepsilon_0))$ by the celebrated lower bound of code lengths of c-secure codes given by Tardos [16]. Now we have the following result, which will be proven in Section 4.3:

Theorem 3. *In the above situation, we have the followings:*

1. *We have $m = \Theta(b^2 m_0)$ for arbitrary δ_0-robust c-secure codes $\mathcal{C}$.*

2. *If $\delta_0 = \Omega(c^{-2})$ and $m_0 = \Theta(c^2 \log(N/\varepsilon_0))$, then the lengths m of the corresponding δ-robust c-secure codes $\overline{\mathcal{C}}$ satisfy*
$$m = \Theta\left(\left(\frac{c\log(1/\delta_0)}{1-\delta}\right)^2 \log(N/\varepsilon)\right) \ .$$
More precisely, if $m_0 \sim Kc^2\log(N/\varepsilon_0)$ for a constant $K > 0$, and

   (a) *if $\delta_0 = \Theta(g(c)^{-1})$ for an eventually positive function $g(c)$ such that $g(c) = O(c^2)$ and $g(c) = \omega(1)$, then we have*
$$m \sim K\left(\frac{c\log g(c)}{1-\delta}\right)^2 \log(N/\varepsilon) \ ;$$

   (b) *if $0 < d < 1$, then we have*
$$m \sim DK\left(\frac{c}{1-\delta}\right)^2 \log(N/\varepsilon) \ ,$$
where
$$D = \max\left\{\frac{-d(1-\log d)^2}{\log(1-d)},\ 1-\log d\right\} < \infty \ .$$

3. *There exist δ-robust c-secure binary codes $\overline{\mathcal{C}}$ with ε-error of length*
$$m = \Theta\left(\left(\frac{c\log c}{1-\delta}\right)^2 \log(N/\varepsilon)\right) \ .$$
Moreover, the constant factor for the above asymptotic expression can be approximately 21.41244*; i.e., we have*
$$m \sim K\left(\frac{c\log c}{1-\delta}\right)^2 \log(N/\varepsilon)\,,\ K \lessapprox 21.41244 \ .$$

The third part of Theorem 3 will be proven by applying the part 2(a) to the robust c-secure codes given in [12]. The part 2(b) says that to obtain δ-robust c-secure codes for arbitrary $0 < \delta < 1$ with code lengths of quadratic order with respect to c, which is theoretically the lowest, it suffices to find such codes for *constant* $0 < \delta < 1$.

4 Proofs of Main Theorems

In this section, we prove the three main theorems presented in Section 3. Sections 4.1, 4.2, and 4.3 deal with the proofs of Theorems 1, 2, and 3, respectively. Some parts of the proofs will be supplied in the appendix.

4.1 Proof of Theorem 1

To prove Theorem 1, let $\overline{\rho}$ be an arbitrary pirate strategy satisfying δ-Marking Assumption for the fingerprint code $\overline{\mathcal{C}}$ obtained by our conversion. By using this $\overline{\rho}$, we construct a pirate strategy ρ for the original δ_0-robust c-secure code $\mathcal{C}$ in the following manner. The algorithm ρ takes as input a subset $C \subset [N]$ and a collection W_C of codewords w_i for $\mathcal{C}$ with $i \in C$. Then ρ proceeds as follows:

1. Choose a sequence fl and a permutation perm as in the definition of $\overline{\mathsf{Gen}}$ uniformly at random, and convert W_C to a collection $\widehat{W}_C$ of codewords for $\overline{\mathcal{C}}$ in the same way as the steps 2–5 in the definition of $\overline{\mathsf{Gen}}$ by using those fl and perm.

2. Execute $\overline{\rho}$ with inputs C and $\widehat{W}_C$, and obtain the output word $\widehat{y}$ (of length m).

3. Convert $\widehat{y}$ to a word $\widehat{y}^{(1)}$ of length m_0 in the same way as the steps 1–2 in the definition of the algorithm $\overline{\mathsf{Tr}}$ by using the same fl and perm as the first step above.

4. If the number of undetectable columns in $\widehat{y}^{(1)}$ (with respect to W_C) marked with '?' is larger than $\delta_0 m_0$, then replace the '?' in every such column with the common digit of the same column in W_C. Otherwise, leave $\widehat{y}^{(1)}$ unchanged. Then output the resulting word $\widehat{y}^{(2)}$.

By definition, the pirate strategy ρ satisfies δ_0-Marking Assumption with respect to $\mathcal{C}$. An Intuitive idea of the proof is to show that the distributions of $\widehat{y}^{(2)}$ and of the word y constructed in the step 2 of $\overline{\mathsf{Tr}}$ are sufficiently close to each other, therefore the security of $\mathcal{C}$ implies the security of $\overline{\mathcal{C}}$.

To give a formal security proof, we may assume without loss of generality that the registration algorithm Reg outputs an arbitrary *fixed* subset $C \subset [N]$. In the proof, we use the following notations:

- $(\overline{W}, \overline{\mathsf{st}})$: The output of the algorithm $\overline{\mathsf{Gen}}$
- (W, st): The output of the algorithm Gen performed in the step 1 of $\overline{\mathsf{Gen}}$
- $\widehat{y}^{(1)}$: The word of length m_0 generated by the step 3 of the above pirate strategy ρ, with inputs being C and the collection W_C of codewords of pirates in W
- $\widehat{y}^{(2)}$: The word of length m_0 generated from $\widehat{y}^{(1)}$ in the step 4 of ρ, with the same inputs as above
- $\overline{y}$: The word of length m output by the pirate strategy $\overline{\rho}$ for $\overline{\mathcal{C}}$, with inputs being C and the collection $\overline{W}_C$ of codewords of pirates in $\overline{W}$
- $\overline{y}^{(1)}$: The word of length m generated by the step 1 of the algorithm $\overline{\mathsf{Tr}}$, with inputs $\overline{y}$, $\overline{W}$, and $\overline{\mathsf{st}}$
- y: The word of length m_0 generated from $\overline{y}^{(1)}$ in the step 2 of $\overline{\mathsf{Tr}}$
- $W^{(1)}$: The collection of words $w_i^{(1)} = w_i$ generated by the step 3 of $\overline{\mathsf{Tr}}$

Then the following property holds, which will be proven in Appendix A:

Lemma 1. *Let* E *denote the event that more than* $\delta_0 m_0$ *undetectable blocks in* $\overline{y}^{(1)}$ *(with respect to* C, $\overline{W}$ *and* perm*) are entirely marked with '?'. Then we have* $Pr[\mathsf{E}] \leq \varepsilon - \varepsilon_0$.

Let $\mathcal{X}$ be the set of all triples (y', W', st') of a word y' of length m_0 over $\Sigma_q \cup \{?\}$, an $N \times m_0$ q-ary matrix W', and state information st' for $\mathcal{C}$. Let $\mathcal{X}_{\mathrm{MA}}$ be the subset of $\mathcal{X}$ of all $(y', W', \mathsf{st}') \in \mathcal{X}$ such that y' satisfies δ_0-Marking Assumption with respect to C and W', and put $\mathcal{X}_{\neg\mathrm{MA}} = \mathcal{X} \setminus \mathcal{X}_{\mathrm{MA}}$. By virtue of Lemma 1, the definition of $\overline{\mathsf{Tr}}$ implies that

$$Pr[(y, W^{(1)}, \mathsf{st}) \in \mathcal{X}_{\neg\mathrm{MA}}] = Pr[\mathsf{E}] \leq \varepsilon - \varepsilon_0 \ . \tag{6}$$

On the other hand, the definition of ρ implies that

$$Pr[(\widehat{y}^{(1)}, W, \mathsf{st}) = \Xi] = Pr[(y, W^{(1)}, \mathsf{st}) = \Xi] \text{ for any } \Xi \in \mathcal{X} \ . \tag{7}$$

In what follows, we write $\Xi^{(1)} = (\widehat{y}^{(1)}, W, \mathsf{st})$ and $\Xi^{(2)} = (\widehat{y}^{(2)}, W, \mathsf{st})$. Then by (7) and the definition of $\overline{\mathsf{Tr}}$, the error probability of $\overline{\mathcal{C}}$ against $\overline{\rho}$ is

$$\sum_{\Xi \in \mathcal{X}} Pr[(y, W^{(1)}, \mathsf{st}) = \Xi] Pr[\mathsf{ev}(\mathsf{Tr}(\Xi), C) = 1] = \sum_{\Xi \in \mathcal{X}} Pr[\Xi^{(1)} = \Xi] Pr[\mathsf{ev}(\mathsf{Tr}(\Xi), C) = 1] \ .$$

On the other hand, by the definition of ρ, the error probability of $\mathcal{C}$ against ρ is

$$\begin{aligned} &\sum_{\Xi \in \mathcal{X}_{\mathrm{MA}}} Pr[\Xi^{(2)} = \Xi] Pr[\mathsf{ev}(\mathsf{Tr}(\Xi), C) = 1] \\ &= \sum_{\Xi \in \mathcal{X}_{\mathrm{MA}}} \sum_{\Xi' \in \mathcal{X}} Pr[\Xi^{(1)} = \Xi'] Pr[\Xi^{(2)} = \Xi \mid \Xi^{(1)} = \Xi'] Pr[\mathsf{ev}(\mathsf{Tr}(\Xi), C) = 1] \end{aligned}$$

(note that ρ satisfies δ_0-Marking Assumption as mentioned above). Moreover, by the definition of ρ, we have $\Xi^{(2)} = \Xi^{(1)}$ if and only if $\Xi^{(1)} \in \mathcal{X}_{\mathrm{MA}}$. This implies that the error probability of $\mathcal{C}$ against ρ is

$$\begin{aligned} &\sum_{\Xi \in \mathcal{X}_{\mathrm{MA}}} Pr[\Xi^{(1)} = \Xi] Pr[\mathsf{ev}(\mathsf{Tr}(\Xi), C) = 1] \\ &+ \sum_{\Xi \in \mathcal{X}_{\mathrm{MA}}} \sum_{\Xi' \in \mathcal{X}_{\neg\mathrm{MA}}} Pr[\Xi^{(1)} = \Xi'] Pr[\Xi^{(2)} = \Xi \mid \Xi^{(1)} = \Xi'] Pr[\mathsf{ev}(\mathsf{Tr}(\Xi), C) = 1] \ . \end{aligned}$$

Hence the difference of the error probability of $\overline{\mathcal{C}}$ against $\overline{\rho}$ from the error probability of $\mathcal{C}$ against ρ is

$$\begin{aligned} &\sum_{\Xi \in \mathcal{X}_{\neg\mathrm{MA}}} Pr[\Xi^{(1)} = \Xi] Pr[\mathsf{ev}(\mathsf{Tr}(\Xi), C) = 1] \\ &- \sum_{\Xi \in \mathcal{X}_{\mathrm{MA}}} \sum_{\Xi' \in \mathcal{X}_{\neg\mathrm{MA}}} Pr[\Xi^{(1)} = \Xi'] Pr[\Xi^{(2)} = \Xi \mid \Xi^{(1)} = \Xi'] Pr[\mathsf{ev}(\mathsf{Tr}(\Xi), C) = 1] \\ &\leq \sum_{\Xi \in \mathcal{X}_{\neg\mathrm{MA}}} Pr[\Xi^{(1)} = \Xi] Pr[\mathsf{ev}(\mathsf{Tr}(\Xi), C) = 1] \leq \sum_{\Xi \in \mathcal{X}_{\neg\mathrm{MA}}} Pr[\Xi^{(1)} = \Xi] = Pr[\Xi^{(1)} \in \mathcal{X}_{\neg\mathrm{MA}}] \leq \varepsilon - \varepsilon_0 \end{aligned}$$

(we used the properties (6) and (7) in the last inequality). As $\mathcal{C}$ is δ_0-robust c-secure with ε_0-error and ρ satisfies δ_0-Marking Assumption, the error probability of $\overline{\mathcal{C}}$ against $\overline{\rho}$ is bounded by $\varepsilon_0 + (\varepsilon - \varepsilon_0) = \varepsilon$. Hence $\overline{\mathcal{C}}$ is δ-robust c-secure with ε-error, concluding the proof of Theorem 1.

4.2 Proof of Theorem 2

To prove Theorem 2, we suppose that the parameter L satisfies (4) and the parameter b satisfies (3). However, in the case $\nu_2 = 1$, we suppose instead that b satisfies (5) which is better than (3). Our goal is to show the property (2). First, it follows from (4) that $L \geq \delta b m_0/(1-\delta)$, therefore $L \geq \delta(bm_0 + L)$ and $L \geq \nu_1$ by the definition of ν_1. The main part of the claim is thus the second inequality in (2).

For the purpose, we may assume that $\nu_1 \geq b\nu_2$, as otherwise the desired inequality is obvious. First note that for any integer $0 \leq a \leq m_0$, we have

$$\frac{\binom{ba+L-b\nu_2}{\nu_1-b\nu_2}}{\binom{ba+L}{\nu_1}} = \frac{(ba+L-b\nu_2)!\nu_1!(ba+L-\nu_1)!}{(\nu_1-b\nu_2)!(ba+L-\nu_1)!(ba+L)!} = \frac{(ba+L-b\nu_2)!\nu_1!}{(\nu_1-b\nu_2)!(ba+L)!} = \frac{(\nu_1)_{b\nu_2}}{(ba+L)_{b\nu_2}}$$

(see Section 1.4 for the notation). Now we present the following lemma, which will be proven in Appendix B:

Lemma 2. *In the above setting, $\binom{a}{\nu_2}(\nu_1)_{b\nu_2}/(ba+L)_{b\nu_2}$ is increasing for integer $0 \leq a \leq m_0$.*

By virtue of Lemma 2, it suffices to prove that $\binom{m_0}{\nu_2}(\nu_1)_{b\nu_2}/(bm_0+L)_{b\nu_2} \leq \varepsilon - \varepsilon_0$. To prove this, we use the following two inequalities:

Lemma 3 (cf., [6])**.** *For integers $0 \leq k \leq n$, we have $\binom{n}{k} \leq (ne/k)^k$.*

Proof. First, we have $\binom{n}{k} = (n)_k/k! \leq n^k/k!$, therefore it suffices to show that $1/k! \leq (e/k)^k$, or equivalently $k! \geq (k/e)^k$. As $\log x$ is an increasing function, we have

$$\log(k!) = \sum_{j=1}^{k} \log j \geq \int_0^k \log x\, dx = \lim_{\xi \to +0} (x\log x - x)|_{x=k} - (x\log x - x)|_{x=\xi} = k\log k - k \ ,$$

therefore $k! \geq e^{k\log k - k} = k^k/e^k$. Hence Lemma 3 holds. □

Lemma 4. *For integers $h \geq i \geq j \geq 1$, we have $(i)_j/(h)_j \leq (i/h)^j$.*

Proof. Apply the inequality $(i-x)/(h-x) \leq i/h$ for every $0 \leq x \leq j$. □

First we consider the case that b satisfies (3). By Lemma 3 and Lemma 4, we have

$$\binom{m_0}{\nu_2}\frac{(\nu_1)_{b\nu_2}}{(bm_0+L)_{b\nu_2}} \leq \left(\frac{m_0 e}{\nu_2}\right)^{\nu_2}\left(\frac{\nu_1}{bm_0+L}\right)^{b\nu_2} = \left(\frac{m_0 e}{\nu_2}\left(\frac{\nu_1}{bm_0+L}\right)^{b}\right)^{\nu_2} \leq \left(\frac{m_0 e}{\nu_2}\delta^b\right)^{\nu_2} \ , \qquad (8)$$

where we used the fact $\nu_1 \leq \delta(bm_0+L)$ (that follows from the definition of ν_1) in the last inequality. By the condition (3), we have

$$b\nu_2 \log(1/\delta) \geq \nu_2 \log(m_0/\nu_2) + \nu_2 + \log(1/\varepsilon_0) \ ,$$

therefore we have $\delta^{-b\nu_2} \geq (m_0 e/\nu_2)^{\nu_2}\varepsilon_0^{-1}$. This implies that the right-hand side of (8) is not larger than $\varepsilon_0 = \varepsilon - \varepsilon_0$ (note that now $\varepsilon_0 = \varepsilon/2$), therefore the claim holds in this case.

Secondly, we consider the case that $\nu_2 = 1$, therefore b satisfies (5) instead of (3). In this case, we use the precise value m_0 of the binomial coefficient $\binom{m_0}{\nu_2}$ instead of the bound given in Lemma 3 to sharpen the analysis. Now we have

$$\binom{m_0}{\nu_2} \frac{(\nu_1)_{b\nu_2}}{(bm_0 + L)_{b\nu_2}} \leq m_0 \left(\frac{\nu_1}{bm_0 + L} \right)^b \leq m_0 \delta^b \ , \tag{9}$$

where we used the fact $\nu_1 \leq \delta(bm_0 + L)$ in the last inequality. By the condition (5), we have $b \log(1/\delta) \geq \log(m_0/\varepsilon_0)$, therefore the right-hand side of (9) is not larger than $\varepsilon_0 = \varepsilon - \varepsilon_0$. Hence the claim also holds in this case, concluding the proof of Theorem 2.

4.3 Proof of Theorem 3

We give proofs of the first part, the second part, and the third part of Theorem 3 in Sections 4.3.1, 4.3.2, and 4.3.3, respectively.

4.3.1 Proof of the First Part

To prove the first part of Theorem 3, put

$$L_1 = \left\lceil \frac{b\nu_2}{1 - (1 - \nu_2/m_0)^{1/b}} \right\rceil - bm_0 + b - 1 \, , \, L_2 = \left\lceil \frac{\delta b m_0}{1 - \delta} \right\rceil \ ,$$

therefore we have $L = \max\{L_1, L_2\}$ (see (4)). First we present the following lemma, which will be proven in Appendix C:

Lemma 5. *In the above setting, we have*

$$1 - \left(1 - \frac{\nu_2}{m_0}\right)^{1/b} \sim \begin{cases} \dfrac{\nu_2}{m_0 b} & \text{if } d = 0 \ , \\ -\dfrac{\nu_2 \log(1-d)}{m_0 b d} & \text{if } 0 < d < 1 \ . \end{cases}$$

By virtue of Lemma 5, we have

$$bm_0 + L_1 \sim \begin{cases} b^2 m_0 & \text{if } d = 0 \ , \\ \dfrac{-d}{\log(1-d)} b^2 m_0 & \text{if } 0 < d < 1 \end{cases} \tag{10}$$

(note that $m_0 = \omega(1)$, as we are assuming that $c \to \infty$, $N/\varepsilon \to \infty$, and $\varepsilon_0 = \varepsilon/2$). On the other hand, we give the following lemma to analyze L_2:

Lemma 6. *We have* $\log(1/\delta) \sim 1 - \delta$ *when* $\delta \to 1$.

Proof. Apply l'Hôpital's Rule to derive $\lim_{\delta \to 1} \log(1/\delta)/(1-\delta) = 1$. □

We put

$$B = \begin{cases} \log(m_0/\nu_2) + 1 + \nu_2^{-1}\log(1/\varepsilon_0) & \text{if } b \text{ is defined by (3)}, \\ \log(m_0/\varepsilon_0) & \text{if } b \text{ is defined by (5)}, \end{cases} \tag{11}$$

therefore $b = \lceil B/\log(1/\delta)\rceil$. As $B = \Omega(1)$ and $\log(1/\delta) \to 0$ (note that now $\delta \to 1$), we have $B/\log(1/\delta) = \omega(1)$ and $b \sim B/\log(1/\delta) \sim B/(1-\delta)$ (see Lemma 6), therefore $(1-\delta)^{-1} \sim b/B = O(b)$. Hence we have $L_2 = O(b^2 m_0)$ and $bm_0 + L_2 = O(b^2 m_0)$. This implies that $bm_0 + L_1$ is eventually dominant over $bm_0 + L_2$, therefore we have $m = \max\{bm_0 + L_1, bm_0 + L_2\} = \Theta(b^2 m_0)$. Hence the first part of Theorem 3 holds.

4.3.2 Proof of the Second Part

To prove the second part of Theorem 2, we use the following lemma, which will be proven in Appendix D:

Lemma 7. *Let x_1 and x_2 be eventually positive functions. If either $x_1 = \Theta(x_2)$ and $x_2 = \omega(1)$, or $x_1 \sim x_2$ and $\log x_2 = \Omega(1)$, then we have $\log x_1 \sim \log x_2$.*

In the following proof, we define the parameter b by (3). We use the value B given in (11). Then we have $b \sim B/(1-\delta)$ as mentioned in Section 4.3.1. On the other hand, as $\varepsilon_0 = \varepsilon/2$ and $N/\varepsilon \to \infty$, we have $\log(N/\varepsilon_0) = \log 2 + \log(N/\varepsilon)$ and $\log(N/\varepsilon_0) \sim \log(N/\varepsilon)$.

First, we have $\delta_0 m_0 = \Omega(\log(N/\varepsilon_0)) = \Omega(\log(N/\varepsilon))$ by the properties of m_0 and δ_0 specified in the statement. Secondly, as $\nu_2 = \lfloor \delta_0 m_0 \rfloor + 1$, we have $\nu_2^{-1}\log(1/\varepsilon_0) \le (\delta_0 m_0)^{-1}\log(1/\varepsilon_0) = O(1)$. Thirdly, we have $\nu_2 \sim \delta_0 m_0$ as $\delta_0 m_0 = \Omega(\log(N/\varepsilon)) = \omega(1)$ (note that $N/\varepsilon \to \infty$), hence $m_0/\nu_2 \sim 1/\delta_0$. Moreover, we have $\log(1/\delta_0) = \Omega(1)$ as we assumed that $\delta_0 \to d < 1$. Now we have $\log(m_0/\nu_2) \sim \log(1/\delta_0)$ by the last two properties and the second part of Lemma 7. By these results, we have $b \sim B/(1-\delta) \sim (1-\delta)^{-1}\log(1/\delta_0)$, therefore

$$m = \Theta\left(\left(\frac{c\log(1/\delta_0)}{1-\delta}\right)^2 \log(N/\varepsilon)\right)$$

by the first part of Theorem 3.

From now, we prove claims (a) and (b). First we consider the claim (a). In this case, we have $d = 0$ as $\delta_0 = \Theta(g(c)^{-1})$ and $g(c) = \omega(1)$, therefore $bm_0 + L_1 \sim b^2 m_0$ by (10). Moreover, we have $\log(1/\delta_0) \sim \log g(c)$ by the first part of Lemma 7. Now $b \sim (1-\delta)^{-1}\log(1/\delta_0)$ as shown above, therefore $b \sim (1-\delta)^{-1}\log g(c)$. Hence we have

$$bm_0 + L_1 \sim \left(\frac{\log g(c)}{1-\delta}\right)^2 m_0$$

and

$$bm_0 + L_2 \sim bm_0 + \frac{\delta b m_0}{1-\delta} = \frac{bm_0}{1-\delta} \sim \frac{\log g(c)}{(1-\delta)^2} m_0 \ .$$

As $\log g(c) = \omega(1)$, this implies that we have eventually $m = bm_0 + L_1$. Hence the claim (a) follows, as $m_0 \sim Kc^2 \log(N/\varepsilon_0)$ and $\log(N/\varepsilon_0) \sim \log(N/\varepsilon)$.

Secondly, we consider the claim (b). Recall from the second last paragraph that $\log(m_0/\nu_2) \sim \log(1/\delta_0)$, $\nu_2^{-1}\log(1/\varepsilon_0) \leq (\delta_0 m_0)^{-1}\log(1/\varepsilon_0)$, and $b \sim B/(1-\delta)$. On the other hand, as $\delta_0 \to d > 0$, we have $\log(1/\delta_0) \sim -\log d$ and

$$\frac{\log(1/\varepsilon_0)}{\nu_2} \leq \frac{\log(1/\varepsilon_0)}{\delta_0 m_0} = \Theta\left(\frac{\log(1/\varepsilon_0)}{\delta_0 c^2 \log(N/\varepsilon_0)}\right) = o(1) \ ,$$

therefore $B \sim 1 - \log d$. Thus we have $b \sim (1-\log d)/(1-\delta)$, and the property (10) implies that

$$bm_0 + L_1 \sim \frac{-db^2 m_0}{\log(1-d)} \sim \frac{-d(1-\log d)^2}{(1-\delta)^2 \log(1-d)} m_0$$

and

$$bm_0 + L_2 \sim bm_0 + \frac{\delta b m_0}{1-\delta} = \frac{bm_0}{1-\delta} \sim \frac{1-\log d}{(1-\delta)^2} m_0 \ .$$

Hence we have $m = \max\{bm_0 + L_1, bm_0 + L_2\} \sim Dm_0/(1-\delta)^2$, therefore the claim (b) holds as $m_0 \sim Kc^2 \log(N/\varepsilon_0) \sim Kc^2 \log(N/\varepsilon)$, concluding the proof of the second part of Theorem 3.

4.3.3 Proof of the Third Part

To prove the third part of Theorem 3, we apply the part 2(a) of the theorem to the δ_0-robust c-secure binary codes $\mathcal{C}$ given by Nuida et al. in [12]. It follows immediately from the argument in Section 6.1 of [12] that their δ_0-robust c-secure codes with ε_0-error have code lengths m_0 satisfying that $m_0 \sim Kc^2 \log(N/\varepsilon_0)$, where $K = (j_1^2(A_0 \log A_0 - A_0 + 1))^{-1}$, $j_1 = 2.40482\cdots$ (see Section 4 of [12] for the precise definition of j_1), and $A_0 = 1 + 2(\pi^{-1} - \Delta_0)/j_1$ for a certain parameter Δ_0, provided $0 \leq \Delta_0 \leq (2\pi)^{-1}$ and $2c^2\delta_0/j_1 \sim \Delta_0$. As K is a continuous function of Δ_0, and we have $K \lessapprox 5.35311$ when $\Delta_0 = 0$ (see Theorem 2 of [12]), it follows that there exists a constant $0 < \Delta_0 < (2\pi)^{-1}$ such that $K \lessapprox 5.35311$. Now by putting $\delta_0 = j_1 \Delta_0 c^{-2}/2 = \Theta(c^{-2})$ to satisfy the above requirement, the part 2(a) of Theorem 3 implies that $m \sim 4K(c \log c/(1-\delta))^2 \log(N/\varepsilon)$ with $4K \lessapprox 21.41244$. Hence the third part of Theorem 3 holds, concluding the proof of Theorem 3.

5 Examples

We have seen in Theorem 3 the asymptotic behavior of code lengths of δ-robust c-secure codes obtained by our conversion method. In this section, we give some numerical examples for the case of smaller c. Here we use the δ_0-robust c-secure binary codes $\mathcal{C}$ with ε_0-error given in [12] as the target of our conversion method. We choose c as $c \in \{2, 3, 4, 6, 8\}$, and we consider the following three choices of parameters N and ε_0:

- Case 1: $N = 100c$ and $\varepsilon_0 = 10^{-11}$;
- Case 2: $N = 10^9$ and $\varepsilon_0 = 10^{-6}$;
- Case 3: $N = 10^6$ and $\varepsilon_0 = 10^{-3}$.

Let $\mathcal{C}_k$ ($k \in \{1,2,3\}$) denote the above δ_0-robust c-secure code $\mathcal{C}$ with various parameter δ_0 listed in Table 1. Namely, the first instance $\mathcal{C}_1$ is not robust at all. The second instance $\mathcal{C}_2$ is a slightly robust one, which is the one appeared in Section 5 of [12]. The third instance $\mathcal{C}_3$ is the most robust one, in the sense that the values δ_0 for $\mathcal{C}_3$ are maximal subject to the conditions for δ_0 specified in [12]. The code lengths m_0 for these three families are shown in Tables 2 and 3. For the tables, the lengths of $\mathcal{C}_1$ and $\mathcal{C}_2$ are quoted from Tables 4 and 5 in [12]. On the other hand, for $\mathcal{C}_3$, we chose the parameters β appeared in the formula of error probability (see Theorem 1 in [12]) as in Table 4, which are optimized by numerical calculation.

Table 1: Parameter δ_0 for the original codes $\mathcal{C} = \mathcal{C}_k$ in [12]

c	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$
2	0	0.005	0.0625
3	0	2.58556×10^{-3}	1.76067×10^{-2}
4	0	2.58556×10^{-3}	1.32044×10^{-2}
6	0	1.78017×10^{-3}	5.61077×10^{-3}
8	0	1.36437×10^{-3}	3.09638×10^{-3}

Then we apply our conversion method to these codes $\mathcal{C}_k$, $k \in \{1,2,3\}$, obtaining δ-robust c-secure codes $\overline{\mathcal{C}_k}$ with ε-error. Here we set $\varepsilon = 2\varepsilon_0$ (as in Section 3) and $\delta = 0.5$ for the parameters, hence the resulting codes $\overline{\mathcal{C}_k}$ are much more robust than the original codes $\mathcal{C}_k$. The code lengths $m = bm_0 + L$ for $\overline{\mathcal{C}_k}$ are also shown in Tables 2 and 3, where we determined the parameters L by (4) and we determined the parameters b for $\overline{\mathcal{C}_1}$ by (5) and for $\overline{\mathcal{C}_2}$ and $\overline{\mathcal{C}_3}$ by (3), respectively. The values of b are also included in Tables 2 and 3.

Tables 2 and 3 show that both $\overline{\mathcal{C}_2}$ and $\overline{\mathcal{C}_3}$ are always more efficient than $\overline{\mathcal{C}_1}$ under our choices of parameters. This suggests that the (even slight) robustness of the original c-secure codes is significant to improve efficiency of (more) robust c-secure codes obtained by our conversion method. On the other hand, these two tables also show that there does not exist overall superiority or inferiority of any of $\overline{\mathcal{C}_2}$ and $\overline{\mathcal{C}_3}$ to the other, hence it is *not* always a good strategy to apply our conversion to more robust c-secure codes. An intuitive explanation of the phenomenon would be possible, as follows. If the original code for our conversion becomes more robust, then the efficiency of our conversion itself is improved (indeed, in Tables 2 and 3, the ratio m/m_0 of code lengths for the pair $(\overline{\mathcal{C}_3}, \mathcal{C}_3)$ is always better than the ratio for the pair $(\overline{\mathcal{C}_2}, \mathcal{C}_2)$), while the code length m_0 of the original code increases, which makes the efficiency of the resulting code worse. Hence there exists a trade-off between those two effects both caused by increase of robustness of the original code. A study of the optimal point for the trade-off would be an interesting future research topic.

Table 2: Code lengths for c-secure codes, with parameter $\delta = 0.5$, $c \in \{2, 3, 4\}$

		Case 1		Case 2		Case 3	
c	code	length	b	length	b	length	b
2	$\mathcal{C}_1$	373	—	410	—	253	—
	$\mathcal{C}_2$	403	—	444	—	273	—
	$\mathcal{C}_3$	1,429	—	1,572	—	969	—
	$\overline{\mathcal{C}_1}$	788,278	46	344,432	29	81,836	18
	$\overline{\mathcal{C}_2}$	177,113	21	113,319	16	53,339	14
	$\overline{\mathcal{C}_3}$	50,082	6	55,094	6	33,963	6
3	$\mathcal{C}_1$	1,309	—	1,423	—	877	—
	$\mathcal{C}_2$	1,514	—	1,646	—	1,014	—
	$\mathcal{C}_3$	4,973	—	5,404	—	3,330	—
	$\overline{\mathcal{C}_1}$	2,890,548	47	1,367,068	31	350,630	20
	$\overline{\mathcal{C}_2}$	604,859	20	322,174	14	198,484	14
	$\overline{\mathcal{C}_3}$	315,807	8	343,166	8	211,470	8
4	$\mathcal{C}_1$	2,190	—	2,360	—	1,454	—
	$\mathcal{C}_2$	2,671	—	2,879	—	1,774	—
	$\mathcal{C}_3$	8,420	—	9,074	—	5,591	—
	$\overline{\mathcal{C}_1}$	5,044,682	48	2,416,177	32	641,024	21
	$\overline{\mathcal{C}_2}$	682,951	16	485,939	13	255,137	12
	$\overline{\mathcal{C}_3}$	677,987	9	577,375	8	355,754	8

Table 3: Code lengths for c-secure codes, with parameter $\delta = 0.5$, $c \in \{6, 8\}$

		Case 1		Case 2		Case 3	
c	code	length	b	length	b	length	b
6	$\mathcal{C}_1$	5,546	—	5,909	—	3,640	—
	$\mathcal{C}_2$	7,738	—	8,244	—	5,079	—
	$\mathcal{C}_3$	21,300	—	22,691	—	13,980	—
	$\overline{\mathcal{C}_1}$	13,314,843	49	6,434,407	33	1,761,551	22
	$\overline{\mathcal{C}_2}$	1,515,387	14	1,186,157	12	730,727	12
	$\overline{\mathcal{C}_3}$	2,124,604	10	2,263,344	10	1,394,451	10
8	$\mathcal{C}_1$	10,469	—	11,062	—	6,815	—
	$\mathcal{C}_2$	16,920	—	17,879	—	11,015	—
	$\mathcal{C}_3$	40,185	—	42,463	—	26,161	—
	$\overline{\mathcal{C}_1}$	26,171,387	50	12,787,166	34	3,604,908	23
	$\overline{\mathcal{C}_2}$	2,857,620	13	2,572,937	12	1,585,115	12
	$\overline{\mathcal{C}_3}$	4,855,517	11	4,240,366	10	2,612,417	10

Table 4: Parameter β for error probability formula of $\mathcal{C}_3$

c	2	3	4	6	8
β	0.093099	0.032980	0.019780	0.0085396	0.0047522

6 Conclusion

In this article, we proposed the first general conversion method of non-robust c-secure fingerprint codes to robust c-secure codes, and of less robust c-secure codes to more robust ones. Our conversion deals with the original c-secure code as a black-box, namely we need no knowledge of properties of the original code except code length and error probability. We estimated appropriate values of parameters for our conversion theoretically, deriving a closed-form formula of the resulting code length. By using the formula, we described the asymptotic behavior of the resulting code length. Moreover, by applying our conversion to some existing (less robust) c-secure codes, we obtained robust c-secure codes with code lengths of order $(c \log c)^2$ with respect to c, which improves some preceding construction based on a stronger assumption than ours.

Acknowledgments

A preliminary version of this article is to be presented at The 4th International Conference on Information Theoretic Security (ICITS 2009), Shizuoka, Japan, December 3–6, 2009. The author would like to thank

Hideki Imai and the anonymous referees of ICITS 2009 for their precious comments.

A Proof of Lemma 1

Recall the condition (2) for the parameters. To prove Lemma 1, it suffices to consider the worst case that pirates always mark as many undetectable columns in the pirated word $\overline{y}$ with '?' as δ-Marking Assumption allows, i.e., they mark $\lfloor \delta m \rfloor = \nu_1$ undetectable columns with '?' in total (note that there are indeed at least ν_1 undetectable columns in $\overline{y}$ by the condition $L \geq \nu_1$; see (2)). For an integer a and each $J \in \binom{[a]}{\nu_2}$, where $\nu_2 = \lfloor \delta_0 m_0 \rfloor + 1$ as defined in (2), let $\mathsf{E}'(a, J)$ denote the event that the number a_{u} of undetectable columns in W (or equivalently, the number of undetectable blocks in $\overline{W}$) with respect to C is a and that for every $j \in J$, the j-th undetectable block in $\overline{y}^{(1)}$ is entirely marked with '?'. (Note that $\binom{[a]}{\nu_2} = \emptyset$ when $a < \nu_2$.) Then, whenever the event E specified in the statement of Lemma 1 occurs, some of the events $\mathsf{E}'(a, J)$ with $0 \leq a \leq m_0$ and $J \in \binom{[a]}{\nu_2}$ also occurs. This implies that

$$Pr[\mathsf{E}] \leq \sum_{a=0}^{m_0} Pr[a_{\mathrm{u}} = a] \sum_{J \in \binom{[a]}{\nu_2}} Pr[\mathsf{E}'(a, J) \mid a_{\mathrm{u}} = a] \ . \tag{12}$$

As the undetectable digits in codewords of pirates are completely shuffled by the steps 4–5 in $\overline{\mathsf{Gen}}$, it follows that every ν_1-element subset of the $ba_{\mathrm{u}} + L$ undetectable columns in $\overline{y}^{(1)}$ is chosen by pirates with equal probability to be marked with '?'. Hence for each a, the probabilities $Pr[\mathsf{E}'(a, J) \mid a_{\mathrm{u}} = a]$ for $J \in \binom{[a]}{\nu_2}$ coincide with each other. As $|\binom{[a]}{\nu_2}| = \binom{a}{\nu_2}$, it follows from (12) that

$$Pr[\mathsf{E}] \leq \sum_{a=0}^{m_0} Pr[a_{\mathrm{u}} = a] \binom{a}{\nu_2} Pr[\mathsf{E}'(a, [\nu_2]) \mid a_{\mathrm{u}} = a] \ . \tag{13}$$

Now suppose that $a_{\mathrm{u}} = a$ and the first ν_2 undetectable blocks in $\overline{y}^{(1)}$ (containing $b\nu_2$ digits in total) are entirely marked with '?'. Then there are $\binom{ba+L-b\nu_2}{\nu_1-b\nu_2}$ choices of the remaining $\nu_1 - b\nu_2$ digits out of the remaining $ba + L - b\nu_2$ undetectable columns to be marked with '?'. On the other hand, there are in total $\binom{ba+L}{\nu_1}$ choices of the ν_1 undetectable columns to be marked with '?'. This implies that $Pr[\mathsf{E}'(a, [\nu_2]) \mid a_{\mathrm{u}} = a] = \binom{ba+L-b\nu_2}{\nu_1-b\nu_2} / \binom{ba+L}{\nu_1}$, therefore it follows from (13) that

$$Pr[\mathsf{E}] \leq \sum_{a=0}^{m_0} Pr[a_{\mathrm{u}} = a] \binom{a}{\nu_2} \frac{\binom{ba+L-b\nu_2}{\nu_1-b\nu_2}}{\binom{ba+L}{\nu_1}} \leq \sum_{a=0}^{m_0} Pr[a_{\mathrm{u}} = a](\varepsilon - \varepsilon_0) = \varepsilon - \varepsilon_0 \ ,$$

where we used the condition (2) in the second inequality. Hence we have $Pr[\mathsf{E}] \leq \varepsilon - \varepsilon_0$ as desired, therefore the proof of Lemma 1 is concluded.

B Proof of Lemma 2

To prove Lemma 2, it suffices to show that

$$\binom{a}{\nu_2}/(ba+L)_{b\nu_2} \le \binom{a+1}{\nu_2}/(ba+L+b)_{b\nu_2}$$

for any integer $0 \le a \le m_0 - 1$. As the claim is obvious when $a < \nu_2$, we may assume that $a \ge \nu_2$. As $\binom{a}{\nu_2}/\binom{a+1}{\nu_2} = 1 - \nu_2/(a+1)$ and

$$\frac{(ba+L)_{b\nu_2}}{(ba+L+b)_{b\nu_2}} = \frac{(ba+L+b-b\nu_2)_b}{(ba+L+b)_b}$$

(note that $b \le b\nu_2 \in \mathbb{Z}$), the claim is equivalent to that

$$1 - \frac{\nu_2}{a+1} \le \frac{(ba+L+b-b\nu_2)_b}{(ba+L+b)_b} \text{ for any integer } \nu_2 \le a \le m_0 - 1 \ . \tag{14}$$

To prove this, we use the following inequality:

Lemma 8. *For integers $h \ge i \ge j \ge 1$, we have*

$$\frac{(i)_j}{(h)_j} \ge \left(\frac{i-j+1}{h-j+1}\right)^j \ .$$

Proof. Apply the inequality $(i-j+x)/(h-j+x) \ge (i-j+1)/(h-j+1)$ for every $1 \le x \le j$. □

By this lemma, the right-hand side of (14) is larger than or equal to

$$\left(\frac{ba+L+1-b\nu_2}{ba+L+1}\right)^b = \left(1 - \frac{b\nu_2}{ba+L+1}\right)^b \ ,$$

therefore it suffices to show that

$$1 - \frac{\nu_2}{a+1} \le \left(1 - \frac{b\nu_2}{ba+L+1}\right)^b \text{ for any } \nu_2 \le a \le m_0 - 1 \ ,$$

or equivalently,

$$\frac{b\nu_2}{ba+L+1} \le 1 - \left(1 - \frac{\nu_2}{a+1}\right)^{1/b} \text{ for any } \nu_2 \le a \le m_0 - 1$$

(note that $0 < \nu_2/(a+1) < 1$). This is also equivalent to $L \ge f(a)$, where

$$f(a) = \frac{b\nu_2}{1 - \left(1 - \frac{\nu_2}{a+1}\right)^{1/b}} - ba - 1 \text{ for } \nu_2 \le a \le m_0 - 1 \ .$$

Now we present a lemma, which will be proven below:

Lemma 9. *The function $f(a)$ is increasing for $\nu_2 \le a \le m_0 - 1$.*

This lemma implies that $f(a) \le f(m_0 - 1)$, while $L \ge f(m_0 - 1)$ by the condition (4) for L, therefore we have $L \ge f(a)$ provided Lemma 9 holds.

Finally, we prove Lemma 9. The claim is obvious if $b = 1$, therefore we assume that $b \ge 2$. Put $x = (1-\nu_2/(a+1))^{1/b}$. Then x is an increasing function of a, and $0 < x < 1$. Now we have $a+1 = \nu_2/(1-x^b)$, therefore $\frac{d}{dx}a = b\nu_2 x^{b-1}(1-x^b)^{-2}$. As $f(a) = b\nu_2/(1-x) - ba - 1$, this implies that

$$\frac{d}{dx}f(a) = \frac{b\nu_2}{(1-x)^2} - \frac{b^2\nu_2 x^{b-1}}{(1-x^b)^2} = \frac{b\nu_2}{(1-x)^2(1-x^b)^2}f_1(x) \ ,$$

where $f_1(x) = (1-x^b)^2 - bx^{b-1}(1-x)^2$. As x is an increasing function of a, it suffices for our purpose to prove that $\frac{d}{dx}f(a) > 0$, or equivalently $f_1(x) > 0$. As $0 < x < 1$, we have $f_1(x) > 0$ if and only if $1 - x^b > \sqrt{b}x^{(b-1)/2}(1-x)$. Put $f_2(x) = 1 - x^b - \sqrt{b}x^{(b-1)/2}(1-x)$, therefore our goal is to show that $f_2(x) > 0$. Now we have

$$\frac{d}{dx}f_2(x) = -bx^{b-1} - \sqrt{b}\frac{b-1}{2}x^{(b-3)/2} + \sqrt{b}\frac{b+1}{2}x^{(b-1)/2} = \sqrt{b}x^{(b-3)/2}f_3(x) \ ,$$

where $f_3(x) = -\sqrt{b}x^{(b+1)/2} - (b-1)/2 + (b+1)x/2$. Moreover, we have

$$\frac{d}{dx}f_3(x) = \frac{b+1}{2}(1 - \sqrt{b}x^{(b-1)/2}) \ .$$

This implies that the function $f_3(x)$ for $0 < x < 1$ is maximized at $x = x_0 = b^{-(b-1)^{-1}}$. We have

$$f_3(x_0) = -\sqrt{b}x_0 \cdot x_0^{(b-1)/2} - \frac{b-1}{2} + \frac{b+1}{2}x_0 = -x_0 - \frac{b-1}{2} + \frac{b+1}{2}x_0 = \frac{b-1}{2}(x_0 - 1) < 0 \ ,$$

therefore $f_3(x) < 0$ for $0 < x < 1$. This implies that $\frac{d}{dx}f_2(x) < 0$ for $0 < x < 1$, therefore $f_2(x) > f_2(1) = 0$ for $0 < x < 1$, as desired. Hence Lemma 9 holds, concluding the proof of Lemma 2.

C Proof of Lemma 5

To prove Lemma 5, first we give the following lemma:

Lemma 10. *In the current situation, we have $\nu_2/m_0 \to d$.*

Proof. First, note that $\nu_2 = \lfloor \delta_0 m_0 \rfloor + 1 \le \delta_0 m_0 + 1$, therefore $\nu_2/m_0 \le \delta_0 + 1/m_0$. As we are assuming that $c \to \infty$, $N/\varepsilon \to \infty$, and $\varepsilon_0 = \varepsilon/2$, it follows that $m_0 = \omega(1)$. Hence we have $\nu_2/m_0 \to 0$ if $\delta_0 \to 0$, or equivalently, if $d = 0$. On the other hand, if $d > 0$, then we have $\delta_0 m_0 = \omega(1)$, therefore $\nu_2 = \lfloor \delta_0 m_0 \rfloor + 1 = \omega(1)$ and $\nu_2 \sim \delta_0 m_0 \sim d m_0$. This implies that we have $\nu_2/m_0 \to d$ in any case. □

By Lemma 10, we have $(1 - \nu_2/m_0)^{m_0/\nu_2} \to A$, where we put $A = e^{-1}$ if $d = 0$ and $A = (1-d)^{1/d}$ if $0 < d < 1$. Note that $0 < A < 1$ in any case. Put $\xi = \nu_2/(bm_0)$. Now for any $0 < \eta < \min\{A, 1-A\}$, we have eventually

$$0 < A - \eta < (1 - \nu_2/m_0)^{m_0/\nu_2} < A + \eta < 1 \ ,$$

therefore we have eventually $0 < (A-\eta)^{\xi} < (1-b\xi)^{1/b} < (A+\eta)^{\xi} < 1$ and

$$\frac{1-(A+\eta)^{\xi}}{\xi} < \frac{1-(1-b\xi)^{1/b}}{\xi} < \frac{1-(A-\eta)^{\xi}}{\xi} \ . \tag{15}$$

Note that $b = \omega(1)$ by the definition of b (note that $\delta \to 1$), therefore we have $\xi \to 0$. Now by l'Hôpital's Rule, the left-hand side and the right-hand side of (15) converge to $-\log(A+\eta)$ and $-\log(A-\eta)$, respectively. Hence we have

$$-\log(A+\eta) \leq \liminf \frac{1-(1-b\xi)^{1/b}}{\xi} \leq \limsup \frac{1-(1-b\xi)^{1/b}}{\xi} \leq -\log(A-\eta) \ .$$

As $\eta > 0$ can be arbitrarily small, by taking the limit $\eta \to 0$ we have

$$\liminf \frac{1-(1-b\xi)^{1/b}}{\xi} = \limsup \frac{1-(1-b\xi)^{1/b}}{\xi} = -\log A \ ,$$

therefore $(1-(1-b\xi)^{1/b})/\xi \to -\log A$. As $-\log A = 1$ if $d = 0$ and $-\log A = -d^{-1}\log(1-d)$ if $0 < d < 1$, the proof of Lemma 5 is concluded.

D Proof of Lemma 7

To prove Lemma 7, first we consider the case that $x_1 = \Theta(x_2)$ and $x_2 = \omega(1)$. Then the first condition implies that there exist an $M_0 > 0$ and an $M_1 < \infty$ such that we have eventually $M_0 < x_1/x_2 < M_1$. Hence it holds eventually that $\log M_0 < \log x_1 - \log x_2 < \log M_1$, or equivalently

$$\frac{\log M_0}{\log x_2} < \frac{\log x_1}{\log x_2} - 1 < \frac{\log M_1}{\log x_2}$$

(note that by the condition $x_2 = \omega(1)$, $\log x_2$ is eventually positive). As $x_2 = \omega(1)$, both the left-hand side and the right-hand side converge to 0, therefore $\log x_1 / \log x_2 \to 1$. Hence $\log x_1 \sim \log x_2$ in this case.

Secondly, we consider the other case that $x_1 \sim x_2$ and $\log x_2 = \Omega(1)$. Then the first condition implies that $x_1/x_2 \to 1$, therefore $\log x_1 - \log x_2 \to 0$. As $\log x_2 = \Omega(1)$, it follows that $\log x_1 / \log x_2 - 1 \to 0$, therefore we have $\log x_1 \sim \log x_2$ in this case. Hence the proof of Lemma 7 is concluded.

References

[1] E. Amiri and G. Tardos, "High rate fingerprinting codes and the fingerprinting capacity," in Proc. SODA 2009, 2009, pp. 336–345.

[2] O. Billet and D. H. Phan, "Efficient traitor tracing from collusion secure codes," in Proc. ICITS 2008, 2008, pp. 171–182.

[3] G. R. Blakley and G. Kabatiansky, "Random coding technique for digital fingerprinting codes," in Proc. IEEE ISIT 2004, 2004, p. 202.

[4] D. Boneh and M. Naor, "Traitor tracing with constant size ciphertext," in Proc. ACM CCS 2008, 2008, pp. 501–510.

[5] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Information Theory*, vol. 44, pp. 1897–1905, 1998.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, 2001.

[7] J. Cotrina-Navau, M. Fernandez and M. Soriano, "A family of collusion 2-secure codes," in Proc. IH 2005, 2005, pp. 387–397.

[8] M. Fernandez and M. Soriano, "Fingerprinting concatenated codes with efficient identification," in Proc. ISC 2002, 2002, pp. 459–470.

[9] H.-J. Guth and B. Pfitzmann, "Error- and collusion-secure fingerprinting for digital data," in Proc. IH 1999, 2000, pp. 134–145.

[10] K. Nuida, "An improvement of short 2-secure fingerprint codes strongly avoiding false-positive," in Proc. IH 2009, 2009, pp. 161–175.

[11] K. Nuida, S. Fujitsu, M. Hagiwara, H. Imai, T. Kitagawa, K. Ogawa and H. Watanabe, "An efficient 2-secure and short random fingerprint code and its security evaluation," IEICE Trans. Fundamentals, vol. E92-A, pp. 197–206, 2009.

[12] K. Nuida, S. Fujitsu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa and H. Imai, "An improvement of discrete Tardos fingerprinting codes," Des. Codes Cryptogr., vol. 52, pp. 339–362, 2009.

[13] R. Safavi-Naini and Y. Wang, "Collusion secure q-ary fingerprinting for perceptual content," in Proc. DRM 2001, 2002, pp. 57–75.

[14] T. Sirvent, "Traitor tracing scheme with constant ciphertext rate against powerful pirates," in Proc. WCC 2007, 2007, pp. 379–388.

[15] B. Škorić, S. Katzenbeisser and M. U. Celik, "Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes," Des. Codes Cryptogr., vol. 46, pp. 137–166, 2008.

[16] G. Tardos, "Optimal probabilistic fingerprint codes," J. ACM, vol. 55, pp. 1–24, 2008.

[17] V. D. Tô, R. Safavi-Naini and Y. Wang, "A 2-secure code with efficient tracing algorithm," in Proc. INDOCRYPT 2002, 2002, pp. 149–163.