# A New Joint Fingerprinting and Decryption Scheme based on a Lattice Problem

Jia XU
jiaxu2001@gmail.com

Department of Computer Science
National University of Singapore

**Abstract.** We propose a new encryption scheme that supports joint fingerprinting and decryption. The scheme is remarkably resistant to known-plaintext attack and collusion attack (e.g. average attack or other linear combination attack) on keys. Interestingly, the security of our scheme is relied on a lattice problem: Given a collection of random lattice points generated from a short basis of a lattice, find the short basis. The scheme can be used as a traitor-tracing scheme or a buyer-seller watermarking scheme.

## 1 Introduction

With the boom of digital computing and the Internet, digital right protection becomes more and more crucial and challenging. Apart from legal penalties, many techniques are devised to resolve or mitigate the threat of pirate. Watermarking is an important technique among them. A typical watermarking application first embeds watermarks into different copies of the content on the server side, and then distributes the watermarked copies to different users. Some scenarios require client-side watermarking, whereby the watermark is only embedded on the client side. One way to achieve secure client-side embedding is through trusted hardware [1,2]. Each subscribed user will receive a tamper-proof physical token to embed watermark into streamed video on the fly. One potential application is subscribed cable TV service.

However, securing hardware is costly and is not fool-proof. An alternative is joint fingerprinting and decryption. This framework consists of two phases: in the key setup phase, through a secure point-to-point channel, each user $u$ receives a personalized decryption key $\mathcal{K}_u$ and the content distributor receives a master encryption key $\mathcal{K}$, from a trusted key generator. In the service phase, only an unsecure broadcast channel is required. All content will be encrypted under the encryption key $\mathcal{K}$ before being broadcasted to all subscribed users. Then each subscribed user $u$ can decrypt the received data using his/her personalized decryption key $\mathcal{K}_u$, and obtains a watermarked copy which is different from but perceptually similar to the original content. Joint fingerprinting and decryption can be realized using Chameleon encryption scheme, where a decryption key is a noisy version of encryption key and this noise is embedded into the decrypted content during decryption.

In this paper, we propose a new joint fingerprinting and decryption scheme, which is remarkably resistant to known-plaintext attack and collusion attack. The security of our scheme is based on a lattice problem and some techniques we used is related to Dan Boneh [3].

## 2 Related works

The problem that these server-side watermark embedding techniques do not address legitimate customer's right for not being framed by a malicious seller, is first identified by Qiao *et al.* [4]. They proposed a owner-customer to deal with this problem, where the owner embed a bit sequence encrypted by the customer as a watermark. Memon *et al.* [5] proposed a buyer-seller protocol, where the seller does not get to know the exact watermark the buyer receives. Lei *et al.* [6] improve the buyer-seller protocol so that a buyer, after observing a pirate copy, cannot transplant the watermark to a higher value digital content.

Client side watermark embedding allows the distributor to distribute a unique copy of encrypted content, and the personalized watermark will be added into the content on the client side in a secure way. The hardware solution requires each legitimate user to have a dedicated secure hardware to embed the watermark, thus incurs a huge deployment cost. A secure software solution is much more preferable. Anderson *et al.* [7] proposed Chameleon Encryption scheme which binds decryption and watermark embedding together so that adversary cannot separate them. Adelsbach *et al.* [8] and Celik *et al.* [9] improved Chameleon Encryption. Adelsbach *et al.* briefly summarized previous works on joint fingerprinting and decryption [10,11,12]. Lemma *et al.* [13] proposed a similar method buy the content owner distributes personalized help data to each user for every decryption.

We realize that all these joint fingerprinting and decryption schemes are vulnerable to known-plaintext attack, and linear combination attack on the decryption keys. In this paper, we propose an encryption scheme, which supports joint fingerprinting and decryption and is resistant to both above attacks.

A widely considered attack method is collusion attack where $k$ subscribed users want to composite their watermarked document $m_1, \ldots m_k$ and create a new document $m^*$ which cannot identify any of the $k$ users. Boneh et al.[14] introduced a formal model of collusion resistance, where it assumes if some mark is same for all the $k$ users, they are not not able to remove that mark. In a similar scenario, Chor et al.[15] proposed traitors tracing scheme, where the data owner publish the data in encrypted form and distribute the decryption keys to subscribed users. When a pirate decoder is found, we will be able to trace the source of the decryption key it used. Zhao et al. presented an analysis on average collusion attack and nonlinear collusion attacks[16,17]. Ergun et al.[18] gives a upper bound that at most $\mathcal{O}(\sqrt{\frac{n}{\log n}})$ collusive users together can defeat any watermarking scheme, where $n$ is the effective document length. Although theoretically bounded, it is still interesting to construct a watermark scheme that is secure when the number of cooperative collusive users is small. A major category of collusion-resistant fingerprinting employs algebraic coding theory. Silverberg et al. gives an approach using list decoding algorithm to reduce the complexity in tracing traitor[19,20]. He et al.[21] taking advantage of joint coding, propose a scheme to improve the collusion resistance of coded fingerprinting.

## 3 Definitions and Notations

Table 1 summarizes the key notations we are going to use in this paper.

**Table 1.** Table of key notations used.

| | |
|---|---|
| $m$: | A block of data, it is represented as an integer in plaintext space $\mathbb{M}$. |
| $\mathsf{Dis}$: | A distance function for comparing the similarity of two data $m_1$, $m_2$. |
| $\mathbf{H}(A)$: | The shannon-entropy of random variable $A$. |
| $\mathbf{H}_\infty(A)$: | The min-entropy of random variable $A$. |
| $\widetilde{\mathbf{H}}_\infty(A\|B)$: | The average min-entropy of $A$ given $B$. |
| $k_e$: | The encryption key, which is also the master key of the movie owner. |
| $k_d, k_{d_i}$: | The decryption key of user $i$, when the context is clear, we simply write $k_d$. |
| $\mathsf{Enc}_k(m)$: | Encryption of $m$ using key $k_e$. |
| $c$: | The ciphertext output by $\mathsf{Enc}_k(m)$, it is an integer in $\mathbb{Z}_p$. |
| $\mathsf{Dec}_{k_i}(c)$: | Decryption of $c$ using key $k_{d_i}$. |

### 3.1 Model

We rephrase the problem as follows. There are four different roles involved: a key generator $\mathcal{G}$, a content distributor $\mathcal{D}$, a set $\mathcal{U}$ of users, and a dispute arbiter $\mathcal{A}$. At the very beginning, the key generator $\mathcal{G}$ generates and distributes an encryption key $\mathcal{K}_e$ to the content distributor $\mathcal{D}$, and a personalized decryption key $\mathcal{K}_{d,u}$ to each user $u \in \mathcal{U}$. After this setup, $\mathcal{D}$ can distribute his/her content (digital movies, music and so on) to all users (who have paid $\mathcal{D}$ for such contents) in $\mathcal{U}$ in this way:

1. $\mathcal{D}$ encrypts the content $m$ using the encryption key $\mathcal{K}_e$: $c \leftarrow \mathsf{Enc}(m; \mathcal{K}_e)$, and distribute $c$ to all users in $\mathcal{U}$ via broadcast.
2. Each user $u \in \mathcal{U}$ decrypts $c$ using the decryption key $\mathcal{K}_{d,u}$: $\widetilde{m}_u \leftarrow \mathsf{Dec}(c; \mathcal{K}_{d,u})$.

Later on, if a pirate copy of content (key, respectively) is found, the dispute arbiter $\mathcal{A}$ will decide which users are responsible by running a detection algorithm $\mathsf{Detect}$.

Each decrypted content $\widetilde{m}_u$ is a (distinct) watermarked copy of $m$. The watermark is embedded into $\widetilde{m}_u$ in the same process of decryption on the client-side. This framework is called as "Joint Fingerprinting and Decryption", and is more preferable than server-side watermark embedding or client-side watermark embedding using trusted hardware in online video/audio streaming service.

**Trust Model.** In all different applications, both key generator $\mathcal{G}$ and dispute arbiter $\mathcal{A}$ are trusted, and all users in $\mathcal{U}$ are not trusted. In traditional watermarking scheme, the content distributor $\mathcal{D}$ is also trusted. However, in a buyer-seller watermarking scheme, $\mathcal{D}$ is not trusted either. In some applications, $\mathcal{G}, \mathcal{A}$ and $\mathcal{D}$ refers to the same entity.

**Threat and Security.** There may be different kinds of adversaries based on their knowledges and roles: (1) Outside adversary who has no encryption key or decryption key; (2) Collusive subscribed users (or buyers) who possess multiple decryption keys; (3) Dishonest content distributor (or seller) who possesses the encryption key. There are various security concerns against the above adversaries, including but not limited to:

1. Confidentiality of content: Prevent outside adversaries from accessing the content from the ciphertext without decryption key.
2. Confidentiality of long term encryption key: Prevent collusive users from accessing the encryption key, even if they possess multiple decryption keys and ciphertexts.
3. Traceability of pirate copy of decryption key: From a pirate copy of decryption key (or an hardware/software implementation of some decryption algorithm), trace at least one among collusive users who are responsible for creation of this pirate key.
4. Traceability of pirate copy of data content: From a pirate copy of data content, trace at least one among collusive users who are responsible for creation of this pirate copy of content.
5. Frame-Proof: Prevent content distributor from framing (or setting up) users.

**Known-Plaintext Attack.** In this paper, we focus on the confidentiality of long term encryption key against known-plaintext attack: (1) If an adversary has access to the (unwatermarked) plaintext, whether he/she can obtain the encryption key; (2) If an adversary has only partial knowledge of the plaintext, whether he/she can remove his/her watermark or estimate an approximate version of encryption key.

### 3.2 Hard Problem Assumptions

The security of our scheme relies on a lattice related problem: Informally, given $m$ points generated from a short basis of a 2-dimensional lattice, inside a $n$ dimensional space, find the short basis of the lattice.

**Definition 1 (Short Basis from Lattice Points $\mathcal{SBLP}$-$(n, m)$).** *Let $p$ be a large prime, and $q_1, q_2 \in \mathbb{Z}_p$ such that $q_1 q_2 < p$. Let $\overrightarrow{\boldsymbol{x}}, \overrightarrow{\boldsymbol{y}} \in (-q_1, q_1)^n$. For $1 \leq i \leq m$, let $\alpha_{i,1}, \alpha_{i,2} \xleftarrow{\$} [0, q_2)$ and compute the vector $\overrightarrow{\boldsymbol{z}}_i = \alpha_{i,1}\overrightarrow{\boldsymbol{x}} + (\alpha_{i,2} - \alpha_{i,1})\overrightarrow{\boldsymbol{y}} \mod p$. Given the values of $p, q_1, q_2$, and $\{\overrightarrow{\boldsymbol{z}}_i : 1 \leq i \leq m\}$ , find $\overrightarrow{\boldsymbol{x}}$ and $\overrightarrow{\boldsymbol{y}}$.*

Note that there are probably many short bases $\{\overrightarrow{\boldsymbol{x'}}, \overrightarrow{\boldsymbol{y'}}\}$ satisfying the constraints, Problem $\mathcal{SBLP}$ is asking to find the exact one, instead of any one. Finding any short basis could be computationally hard, and finding the exact one should be information-theoretically hard to some extent.

**Definition 2 (Approximate Straight Line 1 $\mathcal{ASL}$1-$(L, m)$).** *Let $p$ be a large prime, and $q_1, q_2 \in \mathbb{Z}_p$ such that $q_1 q_2 < p$. Let $\boldsymbol{l}$ be a random straight line in space $\mathbb{Z}_p^3$. Let $\mathcal{S}$ be the set of points on $\boldsymbol{l}$. For each $1 \leq i \leq L$, let $e_i \xleftarrow{\$} \mathbb{Z}_p$, and for each $1 \leq u \leq m$, let $w_{u,i} \xleftarrow{\$} [-\frac{1}{2}q_1, \frac{1}{2}q_1)$.*

*Given* $\{(e_i + w_{u,i})\overrightarrow{\boldsymbol{x}} : \overrightarrow{\boldsymbol{x}} \xleftarrow{\$} \mathcal{S}, 1 \le i \le L, 1 \le u \le m\}$ *and the normal plane of* $\overrightarrow{\boldsymbol{l}}$*, find the line* $\boldsymbol{l}$*.*

**Definition 3 (Approximate Straight Line 2 $\mathcal{ASL}2$-$(L, m)$).** *Let $p$ be a large prime, and $q_1, q_2 \in \mathbb{Z}_p$ such that $q_1 q_2 < p$. Let $\boldsymbol{l}$ be a random straight line in space $\mathbb{Z}_p^3$. Let $\mathcal{S}$ be the set of points on $\boldsymbol{l}$. For each $1 \le i \le L$, let $e_i \xleftarrow{\$} \mathbb{Z}_p$, and for each $1 \le u \le m$, let $w_{u,i} \xleftarrow{\$} [-\frac{1}{2}q_1, \frac{1}{2}q_1)$. Given $\mathcal{L} = \{(e_i + w_{u,i})\overrightarrow{\boldsymbol{x}} : \overrightarrow{\boldsymbol{x}} \xleftarrow{\$} \mathcal{S}, 1 \le i \le L, 1 \le u \le m\}$ and the normal plane of $\overrightarrow{\boldsymbol{l}}$, find $(\overrightarrow{\boldsymbol{y}}, i)$, $1 \le i \le L$, such that there exists $\overrightarrow{\boldsymbol{x}} \in \mathcal{S}$ and $w \in [-\frac{1}{2}q_1, \frac{1}{2}q_1)$, $\overrightarrow{\boldsymbol{y}} = (e_i + w)\overrightarrow{\boldsymbol{x}} \notin \mathcal{L}$.*

These two problems $\mathcal{ASL}1$ and $\mathcal{ASL}2$ are very similar: informally, given a set of approximate points along a straight line, the former asks to find the hidden straight line , and the latter asks to forge a *new* approxmate point on that line. Note that it is not clear whether $\mathcal{ASL}1$ is harder than $\mathcal{ASL}2$, due to unkonwn $e_i$'s.

# 4 An Encryption Scheme supporting Joint Decryption-Watermarking

In this section, we describe the encryption scheme in an incremental manner: In Section 4.1, we propose Scheme 1, which is resistant to known-plaintext attack, but vulnerable to collusion attack on decryption keys; in Section 4.2, we propose Scheme 2, which is resistant to both known-plaintext attack and collusion attack.

## 4.1 Scheme 1: Resistant to Known-Plaintext Attack

Here we give an watermarking encryption scheme such that for any plaintext $m$, $\mathsf{Dec}_{\mathcal{K}_d}(\mathsf{Enc}_{\mathcal{K}_e}(m)) \in [m + \epsilon, m + \epsilon + Q)$, where $\mathcal{K}_e, \mathcal{K}_d$ are the encryption key and decryption respectively, and $\epsilon, Q$ are parameters about tolerable noise determined by the application domain.

1. Encryption key: Let $p$ be a large prime. For each $1 \le i \le 2$, $e_i \xleftarrow{\$} \mathbb{Z}_p; r_i \xleftarrow{\$} \mathbb{Z}_p^*$. Choose $q_1, q_2$, such that $q_1 q_2 \le Q < p$. The encryption key is $\mathcal{K}_e = (p, e_1, e_2, r_1, r_2)$.
2. Decryption key: For $1 \le i \le 2$, $d_i = (e_i + w_i)r_i \mod p$, where $w_i \xleftarrow{\$} [-\frac{1}{2}q_1, \frac{1}{2}q_1)$. The decryption key is $\mathcal{K}_d = (p, d_1, d_2)$.
3. The encryption of $m \in \mathbb{M} \subset \mathbb{Z}_p$ under key $\mathcal{K}_e$, with tolerable error range $[\epsilon, \epsilon + Q)$:
   (a) Choose $\gamma_1, \gamma_2$ from $[0, q_2)$ at random. Let $\alpha_1 = \gamma_1, \alpha_2 = \gamma_2 - \gamma_1$.
   (b) Choose $\beta$ from $[\beta_{\min}, \beta_{max})$ at random, where $\beta_{\min} = \epsilon + \frac{1}{2}Q$, and $\beta_{max} = \beta_{min} + Q - (q_1 - 1)(\alpha_1 + \alpha_2)$.
   (c) $\mathsf{Enc}(m; \mathcal{K}_e) = (c_1, c_2, c_3)$ where

$$c_1 = \alpha_1 r_1^{-1} \mod p;\ c_2 = \alpha_2 r_2^{-1} \mod p;\ c_3 = m - \alpha_1 e_1 - \alpha_2 e_2 + \beta \mod p.$$

4. Decryption (Joint Fingerprinting and Decryption) of $(c_1, c_2, c_3)$ using key $\mathcal{K}_d$:

$$\begin{aligned}
&\mathsf{Dec}(c_1, c_2, c_3; \mathcal{K}_d) \\
&= c_1 d_1 + c_2 d_2 + c_3 \\
&= \alpha_1 r_1^{-1} \times (e_1 + w_1) r_1 + \alpha_2 r_2^{-1} \times (e_2 + w_2) r_2 + (m - \alpha_1 e_1 - \alpha_2 e_2 + \beta) \\
&= m + (\alpha_1 w_1 + \alpha_2 w_2 + \beta) \mod p \\
&\in [m + \epsilon, m + \epsilon + Q).
\end{aligned}$$

Note that our choice of $\alpha_1, \alpha_2, \beta$ ensure that $(\alpha_1 w_1 + \alpha_2 w_2 + \beta) \mod p$ is within $[\epsilon, \epsilon+Q)$ as desired.

**Theorem 1.** $\mathbf{H}_\infty(\mathbf{m}|\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) = \mathbf{H}_\infty(\mathbf{m})$

*Proof.* We have $\mathbf{H}_\infty(\mathbf{m}|\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \leq \mathbf{H}_\infty(\mathbf{m})$. Furthermore,

$$\begin{aligned}
&\mathbf{H}_\infty(\mathbf{m}|\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \\
&\geq \mathbf{H}_\infty(\mathbf{m}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) - 3 \log p \\
&= \mathbf{H}_\infty(\mathbf{m}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3 - \mathbf{m}) - 3 \log p \\
&= \mathbf{H}_\infty(\mathbf{m}) + \mathbf{H}_\infty(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3 - m) - 3 \log p \\
&= \mathbf{H}_\infty(\mathbf{m}).
\end{aligned}$$

So $\mathbf{H}_\infty(\mathbf{m}|\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) = \mathbf{H}_\infty(\mathbf{m})$.

**Theorem 2.** $\mathbf{H}_\infty(\boldsymbol{\mathcal{K}}_e|\boldsymbol{\mathcal{K}}_d) \geq 2 \log p + 2 \log q_1$

**Lemma 1.** *Let* $\boldsymbol{\alpha} = \boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2$

$$\mathbf{H}(\boldsymbol{\beta}|\boldsymbol{\alpha}) \approx \log Q - 1.$$

*Proof.* At first, it is straightforward to find that $\mathbf{H}(\boldsymbol{\beta}|\boldsymbol{\alpha}) < \log Q$. Next, we will show that $\mathbf{H}(\boldsymbol{\beta}|\boldsymbol{\alpha}) > \log Q - \frac{1}{\ln 2}$.

$$\mathbf{H}(\boldsymbol{\beta}|\boldsymbol{\alpha} = \alpha) = \log(Q - \alpha(q_1 - 1)) > \log q_1(q_2 - \alpha).$$

$$\begin{aligned}
&\mathbf{H}(\boldsymbol{\beta}|\boldsymbol{\alpha}) \\
&= \sum_{\alpha \in [0, q_2)} \mathsf{Pr}(\boldsymbol{\alpha} = \alpha) \mathbf{H}(\boldsymbol{\beta}|\boldsymbol{\alpha} = \alpha) \\
&> \sum_{\alpha \in [0, q_2)} \frac{1}{q_2} \log q_1(q_2 - \alpha) \\
&= \frac{1}{q_2} \sum_{\alpha \in [0, q_2)} (\log q_1 + \log(q_2 - \alpha)) \\
&= \log q_1 + \frac{1}{q_2} \sum_{\alpha \in [0, q_2)} \log(q_2 - \alpha) \\
&= \log q_1 + \frac{1}{q_2} \sum_{N \in (0, q_2]} \log N
\end{aligned}$$

We have

$$\sum_{N\in(0,q_2]} \ln N = \ln q_2! > q_2 \ln q_2 - q_2$$

Hence

$$\sum_{N\in(0,q_2]} \log N = \frac{1}{\ln 2} \sum_{N\in(0,q_2]} \ln N > q_2 \log q_2 - \frac{q_2}{\ln 2}$$

and

$$\mathbf{H}(\boldsymbol{\beta}|\boldsymbol{\alpha}) > \log Q - \frac{1}{\ln 2}.$$

Note that $Q = q_1 q_2$.

**Theorem 3.** *1.* $\mathbf{H}_\infty(\mathbf{w}_1, \mathbf{w}_2, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\beta}|\boldsymbol{\alpha}_1\mathbf{w}_1 + \boldsymbol{\alpha}_2\mathbf{w}_2 + \boldsymbol{\beta}) \geq 3\log Q - \log p - \frac{1}{\ln 2}$.
*2.* $\mathbf{H}_\infty(\mathbf{w}_1, \mathbf{w}_2|\boldsymbol{\alpha}_1\mathbf{w}_1 + \boldsymbol{\alpha}_2\mathbf{w}_2 + \boldsymbol{\beta}) \geq 2\log q_1 + \log Q - \log p - \frac{1}{\ln 2}$.

*Proof.* $\mathbf{H}_\infty(\mathbf{w}_1, \mathbf{w}_2, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\beta}, \boldsymbol{\alpha}_1\mathbf{w}_1 + \boldsymbol{\alpha}_2\mathbf{w}_2 + \boldsymbol{\beta}) \geq 2\log q_1 + 2\log q_2 + \log Q - \frac{1}{\ln 2}$

**Security against Collusive Attackers** The security of the scheme against $(n+1)$ collusive users with $m$ ciphertexts can be reduced to the problem $\mathcal{SBLP}$-$(n, m)$.

**Theorem 4.** *The secret key of Scheme 1 is CPA2 secure against* $(n+1)$ *collusive attackers with* $m$ *ciphertexts, if* $\mathcal{SBLP}$-$(n, m)$ *is hard.*

**Average attack and linear combination of user keys.** Colluded adversaries can forge a new decryption key by linear combination of their user keys. How to use $w_i$ sequence to identify user in a robust manner?

### 4.2 Scheme 2: Resistant to Collusion Attack

1. Setup: Let $p$ be a large prime. Choose $q_1, q_2$ such that $q_1 q_2 \leq Q < p$. Find two linearly independent vectors $\overrightarrow{\boldsymbol{n}}_1 = (A_1, B_1, C_1)$ and $\overrightarrow{\boldsymbol{n}}_2 = (A_2, B_2, C_2)$ from $\mathbb{Z}_p^3$. Let $\mathcal{S}$ be the set[1]

$$\{\overrightarrow{\boldsymbol{x}} : \overrightarrow{\boldsymbol{n}}_1 \cdot \overrightarrow{\boldsymbol{x}} = 1 \mod p,\ \overrightarrow{\boldsymbol{n}}_2 \cdot \overrightarrow{\boldsymbol{x}} = 1 \mod p\}$$

   (a) Encryption key: $e_1, e_2 \xleftarrow{\$} \mathbb{Z}_p$. The encryption key is $\mathcal{K}_e = (p, e_1, e_2, \overrightarrow{\boldsymbol{n}}_1, \overrightarrow{\boldsymbol{n}}_2)$.
   (b) Decryption key: $\overrightarrow{\boldsymbol{d}}_1 = (e_1 + w_1)\overrightarrow{\boldsymbol{x}}_1$ and $\overrightarrow{\boldsymbol{d}}_2 = (e_2 + w_2)\overrightarrow{\boldsymbol{x}}_2$ where $\overrightarrow{\boldsymbol{x}}_1, \overrightarrow{\boldsymbol{x}}_2 \xleftarrow{\$} \mathcal{S}$ and $w_1, w_2 \xleftarrow{\$} [-\frac{1}{2}q_1, \frac{1}{2}q_1)$. The decryption key is $\mathcal{K}_d = (p, \overrightarrow{\boldsymbol{d}}_1, \overrightarrow{\boldsymbol{d}}_2)$.
2. Encryption of $m \in \mathcal{M}$, with tolerable error range $[\epsilon, \epsilon + Q)$
   (a) Choose $\alpha_1, \alpha_3, v_1, v_2$ from $[0, q_2)$ at random. Let $\alpha_2 = (v_1 - \alpha_1) \mod p$ and $\alpha_4 = (v_2 - \alpha_3) \mod p$.
   (b) Choose $\beta$ from $[\beta_{min}, \beta_{max})$ at random, where $\beta_{min} = \epsilon + \frac{1}{2}Q$ and $\beta_{max} = \beta_{min} + Q - (q_1 - 1)(v_1 + v_2)$.

[1] In fact, $\mathcal{S}$ is the set of points in a straight line, determined by $\overrightarrow{\boldsymbol{n}}_1$ and $\overrightarrow{\boldsymbol{n}}_2$, in space $\mathbb{Z}_p^3$.

(c) $\mathsf{Enc}(m; \mathcal{K}_e) = (\overrightarrow{\boldsymbol{c}}_1, \overrightarrow{\boldsymbol{c}}_2, c_3)$, where

$$\begin{aligned}
\overrightarrow{\boldsymbol{c}}_1 &= \alpha_1 \overrightarrow{\boldsymbol{n}}_1 + \alpha_2 \overrightarrow{\boldsymbol{n}}_2 \mod p, \\
\overrightarrow{\boldsymbol{c}}_2 &= \alpha_3 \overrightarrow{\boldsymbol{n}}_1 + \alpha_4 \overrightarrow{\boldsymbol{n}}_2 \mod p, \\
c_3 &= m - e_1(\alpha_1 + \alpha_2) - e_2(\alpha_3 + \alpha_4) + \beta \mod p.
\end{aligned}$$

3. Decryption of $(\overrightarrow{\boldsymbol{c}}_1, \overrightarrow{\boldsymbol{c}}_2, c_3)$

$$\begin{aligned}
\mathsf{Dec}(\overrightarrow{\boldsymbol{c}}_1, \overrightarrow{\boldsymbol{c}}_2, c_3; \mathcal{K}_d) &= \overrightarrow{\boldsymbol{c}}_1 \cdot \overrightarrow{\boldsymbol{d}}_1 + \overrightarrow{\boldsymbol{c}}_2 \cdot \overrightarrow{\boldsymbol{d}}_2 + c_3 \\
&= (e_1 + w_1)(\alpha_1 + \alpha_2) + (e_2 + w_2)(\alpha_3 + \alpha_4) + \\
&\quad m - e_1(\alpha_1 + \alpha_2) - e_2(\alpha_3 + \alpha_4) + \beta \\
&= m + w_1(\alpha_1 + \alpha_2) + w_2(\alpha_3 + \alpha_4) + \beta \pmod p
\end{aligned}$$

**Remark.**

1. Use of random nonces $\alpha$'s and $\beta$.
2. For every user, $e_1$ and $e_2$ (respectively, $w_1$ and $w_2$; $\overrightarrow{\boldsymbol{x}}_1$ and $\overrightarrow{\boldsymbol{x}}_2$) are probably distinct. For any two users, they share the same $e_1, e_2$, but each has different version of $w_1, w_2$, $\overrightarrow{\boldsymbol{x}}_1$, $\overrightarrow{\boldsymbol{x}}_2$.

**Remark.** In fact, a point in $\mathcal{S}$ is essentially the same as a "presentation " as in Dan Boneh [3].

**Theorem 5.** *The secret keys of Scheme 3 is CPA2-secure with at most $m/2$ oracle queries against collusive attackers, if $\mathcal{SBLP}$-$(3, m)$ is hard.*

It is clear that collusive linear combination attackers (e.g. average attack) cannot forge a new decryption key. Furthermore, we have

**Theorem 6.** *$m$ collusive attackers cannot forge a new key $\overrightarrow{\boldsymbol{d}}'_1$ or $\overrightarrow{\boldsymbol{d}}'_2$, if $\mathcal{ASL}$2-$(1, m)$ is hard.*

Therefore, the only way that the collusive attackers (who are legitimate users) can pirate a different key $\mathcal{K}'_d$, is to mix and combine different key components from different legitimate users. Fortunately, such kinds of attackers can be easily caught.

The drawback of this scheme is that ciphertext is 7 times of plaintext. We resolve this problem in Section 5 by reusing $\alpha_i$'s and $\beta$ for several times, at the cost of slightly larger encryption key and decryption key.

## 5 Applications in Tracing Pirate

Let $\mathsf{Enc}$ and $\mathsf{Dec}$ be as in Scheme 3 in Section 4.2.

1. Setup: Choose $p, q_1, q_2, \overrightarrow{\boldsymbol{n}}_1, \overrightarrow{\boldsymbol{n}}_2, \mathcal{S}$ as in Scheme 3 in Section 4.2.

(a) Encryption key: for $1 \le i \le L$, $e_i \xleftarrow{\$} \mathbb{Z}_p$. Let $\mathbf{E}$ be a $L \times 1$ dimensional table, such that $\mathbf{E}[i] = e_i$. The encryption key is $\mathcal{K}_e = (p, \mathbf{E}, \overrightarrow{\boldsymbol{n}}_1, \overrightarrow{\boldsymbol{n}}_2)$.

(b) Decryption key: For any user $u \in \mathcal{U}$, for $1 \le i \le L$, let $\overrightarrow{\boldsymbol{d}}_{u,i} = (e_i + w_{u,i})\overrightarrow{\boldsymbol{x}}_{u,i}$, where $\overrightarrow{\boldsymbol{x}}_{u,i} \xleftarrow{\$} \mathcal{S}$ and $w_{u,i} \xleftarrow{\$} [-\frac{1}{2}q_1, \frac{1}{2}q_1)$. Let $\mathbf{D}_u$ be a $L \times 1$ dimensional table, such that $\mathbf{D}_u[i] = \overrightarrow{\boldsymbol{d}}_{u,i}$. The decryption key for User $u$ is $k_u = (p, \mathbf{D}_u)$.

2. Encryption of $(m_1, m_2, \ldots, m_{\frac{1}{2}L}) \in \mathcal{M}^{\frac{1}{2}L}$ under encryption key $\mathcal{K}_e$ and session key $s$, with tolerable error range $[\epsilon, \epsilon + Q)$

(a) From session key $s$, generate a permutation $t_1, t_2, t_3, \ldots, t_L$ of the set $[1, L]$.

(b) Choose $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ and $\beta$ as in Scheme 3 in Section 4.2.

(c) For each $1 \le i \le \frac{1}{2}L$, let $k_{e,i} = (p, \mathbf{E}[t_{2i-1}], \mathbf{E}[t_{2i}], \overrightarrow{\boldsymbol{n}}_1, \overrightarrow{\boldsymbol{n}}_2)$,

$$(\overrightarrow{\boldsymbol{c}}_1, \overrightarrow{\boldsymbol{c}}_2, c_{i,3}) \leftarrow \mathsf{Enc}(m_i; k_{e,i}).$$

Note that $\overrightarrow{\boldsymbol{c}}_1$ and $\overrightarrow{\boldsymbol{c}}_1$ are common for all $m_i$'s, $1 \le i \le \frac{1}{2}L$.

(d) Output ciphertext $C \leftarrow (s, \overrightarrow{\boldsymbol{c}}_1, \overrightarrow{\boldsymbol{c}}_2, c_{1,3}, c_{2,3}, \ldots, c_{\frac{1}{2}L,3})$.

3. Decryption of $C$ by User $u$ using decryption key $k_u$

(a) From $s$, generate a permutation $t_1, t_2, t_3, \ldots, t_L$ of the set $[1, L]$.

(b) For each $1 \le i \le \frac{1}{2}L$, let $d_i = (p, \mathbf{D}_u[t_{2i-1}], \mathbf{D}_u[t_{2i}])$,

$$\widetilde{m}_i \leftarrow \mathsf{Dec}(\overrightarrow{\boldsymbol{c}}_1, \overrightarrow{\boldsymbol{c}}_2, c_{i,3}; d_i).$$

(c) Output $(\widetilde{m}_1, \widetilde{m}_2, \ldots, \widetilde{m}_{\frac{1}{2}L})$.

The ciphertext size is about $(1 + \frac{14}{L})$ times of plaintext, if ignoreing the difference in size between $m_i$ and $p$.

## 6 Conclusions and Future works

We proposed a joint fingerprinting and decryption scheme which is remarkably resistant to known-plaintext attack and collusion attack. Interestingly, the security of our scheme is related with a Lattice problem.

## References


1. Tomsich, P., Katzenbeisser, S.: Copyright protection protocols for multimedia distribution based on trusted hardware. In: in Protocols for Multimedia Systems (PROMS 2000), Proceedings. (2000) 815–819
2. Tomsich, P., Katzenbeisser, S.: Towards a secure and de-centralized digital watermarking infrastructure for the protection of intellectual property. In: EC-Web. (2000) 38–47
3. Boneh, D., Franklin, M.K.: An efficient public key traitor tracing scheme. In: CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology. (1999) 338–353
4. Qiao, L., Nahrstedt, K.: Watermarking schemes and protocols for protecting rightful ownership and customer's rights. Journal of Visual Communication and Image Representation **9**(3) (1998) 194–210
5. Memon, N., Wong, P.: A buyer-seller watermarking protocol. IEEE Transactions on image processing **10**(4) (2001) 643–649

6. Lei, C., Yu, P., Tsai, P., Chan, M., et al.: An efficient and anonymous buyer-seller watermarking protocol. IEEE Transactions on Image Processing **13**(12) (2004) 1618–1626
7. Anderson, R.J., Manifavas, C.: Chameleon - a new kind of stream cipher. In: FSE '97: Proceedings of the 4th International Workshop on Fast Software Encryption. (1997) 107–113
8. Adelsbach, A., Huber, U., Sadeghi, A.R.: Fingercasting-joint fingerprinting and decryption of broadcast messages. T. Data Hiding and Multimedia Security **2** (2007) 1–34
9. Celik, M.U., Lemma, A.N., Katzenbeisser, S., van der Veen, M.: Lookup-table-based secure client-side embedding for spread-spectrum watermarks. IEEE Transactions on Information Forensics and Security **3**(3) (2008) 475–487
10. Parviainen, R., Parnes, R.: Large scale distributed watermarking of multicast media through encryption. In: Proceedings of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security Issues of the New Century, Deventer, The Netherlands, The Netherlands, Kluwer, B.V. (2001) 17
11. Kundur, D., Karthik, K.: Video fingerprinting and encryption principles for digital rights management. In: Proceedings of the IEEE. Volume 92. (2004) 918–932
12. Luh, W., Kundur, D.: New paradigms for effective multicasting and fingerprinting of entertainment media. IEEE Communications Magazine **43**(6) (2005) 77–84
13. Lemma, A., Katzenbeisser, S., Celik, M., van der Veen, M.: Secure watermark embedding through partial encryption. In: International Workshop on Digital Watermarking. Volume 4283. (2006) 433–445
14. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Transactions on Information Theory **44**(5) (1998) 1897–1905
15. Chor, B., Fiat, A., Naor, M., Pinkas, B.: Tracing traitors. IEEE Transactions on Information Theory **46**(3) (2000) 893–910
16. Zhao, H., Wu, M., Wang, Z., Liu, K.: Nonlinear collusion attacks on independent fingerprints for multimedia. In: IEEE ICASSP. (2003)
17. Zhao, H., Wu, M., Wang, Z., Liu, K.: Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting. IEEE Transactions on Image Processing **14**(5) (2005) 646–661
18. Ergun, F., Kilian, J., Kumar, R.: A note on the limits of collusion-resistant watermarks. Lecture notes in computer science (1999) 140–149
19. Silverberg, A., Staddon, J., Walker, J.: Efficient traitor tracing algorithms using list decoding. Lecture notes in computer science (2002) 175–192
20. Silverberg, A., Staddon, J., Walker, J.: Applications of list decoding to tracing traitors. IEEE Transactions on Information Theory **49**(5) (2003) 1312–1318
21. He, S., Wu, M.: Joint coding and embedding techniques for MultimediaFingerprinting. IEEE Transactions on Information Forensics and Security **1**(2) (2006) 231–247