Efficient Decentralized Tracing Protocol for Fingerprinting System with Index Table

Minoru Kuribayashi and Nobuo Funabiki
Okayama University, Okayama, Japan
E-mail: kminoru@okayama-u.ac.jp

***Abstract*—Due to the burden at a trusted center, a decentralized fingerprinting system has been proposed by delegating authority to an authorized server so that the center does not participate in the tracing protocol. As a fingerprinting code is used to retain a collusion resistance, the calculation of correlation score for each user is required to identify illegal users from a pirated copy. Considering the secrecy of code parameters, the computation must be executed by a seller in an encrypted domain to realize the decentralized tracing protocol. It requires much computational costs as well as the communication costs between the center and a seller because encrypted database (DB) is necessary for the computation. In this paper, we propose a method to reduce such costs by using the ElGamal cryptosystem over elliptic curve instead of the Paillier cryptosystem used in the conventional scheme. Our experimental results indicate that the time consumption becomes almost 100 times shorter and the size of encrypted DB reduced by a factor of 7/32 under 112-bit security level. The encrypted DB is further compressed by introducing an index table.**

I. Introduction

In order to trace illegal users from a pirated version of multimedia content, identification (ID) information of a user regarded as fingerprint is inserted into the copy when it is delivered to the user. A fingerprinting system involves the distribution of multimedia content to legitimate users, embedding of user-specific identity information, and identification of illegal users. It requires some primitive techniques such as cryptographic protocol, watermarking, collusion-resistant code, multimedia signal processing and so on.

From the perspective of cryptographic protocol, one of the important issues is the dispute between buyer(user) and seller. If both party obtain a fingerprinted content after a transaction protocol, the seller cannot prove to the other party about an illegal action of the user even if his/her fingerprint is correctly extracted from a pirated copy. The reason for this is that a malicious seller may distribute the copy by himself/herself to frame an innocent user. In [1], an idea of asymmetric protocol was presented so that only a user can obtain his/her fingerprinted copy by exploiting the homomorphism of a public-key cryptosystem. It enables a seller to embed fingerprint into multimedia content in an encrypted domain. Since the ciphertext is computed using a user's public key, only the user can decrypt it; hence, only he can obtain the fingerprinted content. There are many studies for such an asymmetric fingerprinting protocol to improve the performance [2], [3], [4], [5], [6], and to add some functionalities [7], [8], [9], [10], [11].

From the different point of view, as differently fingerprinted versions of same content are delivered to users, a coalition of users will be able to modify/delete the fingerprint, which is called collusion attack. In order to tolerate for the collusion attack, fingerprinting codes [12], [13], [14], [15] enable a seller to catch at least one illegal user from a pirated copy. Among the codes, the Tardos code [14] and its variant [15] are attractive because its code length can be a theoretically minimum order. The codeword is produced by a probabilistic algorithm based on bias probabilities as secret parameter.

Different from the study of asymmetric protocol at the time of content delivery, the tracing protocol was investigated in [16] by introducing an idea of delegated server. The server helps a seller to identify illegal users when a pirated copy is found while a trusted center works only at the time of registration phase. The center selects secret parameters of fingerprinting code and issues each codeword to each user. The trusted center allows the server to check a correlation score whether it exceeds a threshold which is determined by the center. The server's task is to decrypt a ciphertext received from a seller, and return a binary decision. The cryptographic protocol was implemented by using the Paillier cryptosystem [17] and the computational costs and the communication costs are measured for the tracing protocol. Although its protocol can be executed, the costs required for the seller are considerably high. As the encrypted database (DB) is linearly increased with the number of users in a system, reducing the size is advisable.

In this paper, we propose a method to reduce the costs by using the ElGamal cryptosystem [18] over elliptic curve cryptography (ECC). It is well-known that the lifted version of ElGamal cryptosystem retains the property of additive homomorphism though a discrete logarithm problem must be solved at decryption. If the size of plaintext is small, it is not difficult by referring a look-up table that can be prepared in advance. As the correlation score is relatively small, the decryption algorithm can run within a reasonable time and storage. We estimate the time and storage under the same security level in the conventional scheme. As the size of ciphertext of lifted ElGamal cryptosystem over EC is smaller than that of Paillier cryptosystem, the encrypted DB is reduced as shown in the results. In order to further compress the size of encrypted DB, we introduce an index table. It allows us to use a constant number of ciphertexts for the DB. Even though the size of the table is still linearly increased with the number

of users, it is revealed from our estimation that the size of storage required in our method is reasonably small.

II. Preliminaries

A. Additive Homomorphic Encryption

An additive homomorphic encryption scheme allows addition under encryption. It defines an addition "+" on plaintexts and a corresponding operation "$\odot$" on ciphertexts. Let m_1 and m_2 be plaintexts. Then, an enciphering function $Enc()$ satisfies the following property:

$$Enc(m_1) \odot Enc(m_2) = Enc(m_1 + m_2) \tag{1}$$

The popular additive homomorphic encryption system is Paillier cryptosystem [17]. The public key is an RSA modulus N, which is a composite of two large primes, and the secret key is the factorization of the modulus. It provides a $\log_2 N$-bit plaintext space and $2\log_2 N$-bit ciphertext space. A plaintext m is encrypted into a ciphertext $Enc(m)$ by mapping $\mathbb{Z}_N$ to $\mathbb{Z}^*_{N^2}$. Due to the security reason, the size of RSA modulus N is recommended to be $\log_2 N \geq 2048$ bits. In case of $\log_2 N = 2048$, the ciphertext size of Paillier cryptosystem is 4096 bits though the plaintext size is 2048 bits. The operation $\odot$ of the Paillier cryptosystem is multiplication under the modulus N^2. This additive homomorphism makes it possible to perform the following operation:

$$Enc(m_1)^{m_2} = Enc(m_1 \cdot m_2) \tag{2}$$

A generalization of Paillier cryptosystem was presented by Damgård and Jurik [19], [20], which can provide a larger plaintext space. The disadvantage of the Paillier cryptosystem and its generalized version is the heavy computational complexity because the modular multiplication is computationally expensive. The large modulus N^2 also increases the complexity.

B. Fingerprinting Code

A coalition of users may try to prevent from being caught by forming a pirated copy from their individual copies of a same content. To identify at least one of such users from a pirated copy, collusion-resistant fingerprinting schemes are studied. A fingerprinting code is one of the methods to realize the traceability of illegal users called colluders. Among other fingerprinting codes, Tardos [14] proposed an efficient code whose code length is theoretically minimum order.

Let N_u be the number of users in a system. The code length ℓ can be determined both by the number of users N_u in a system and maximum number c_{max} of colluders assumed at the setting of code. A binary codeword of j-th user is denoted by $\boldsymbol{X_j} = (X_{j,1}, \ldots, X_{j,\ell})$, $(X_{j,i} \in \{0,1\}, 1 \leq j \leq N_u, 1 \leq i \leq \ell)$, where $X_{j,i}$ is generated from an independently and identically distributed (i.i.d.) random number with a probability p_i such that $\Pr[X_{j,i} = 1] = p_i$ and $\Pr[X_{j,i} = 0] = 1 - p_i$.

In Tardos's construction, the probability p_i, $(1 \leq i \leq \ell)$ follows a certain continuous distribution, called the bias distribution. Nuida et al. [15] investigated the optimal bias distribution for a given maximum number c_{max} of colluders and gives a discrete version of bias distribution. The number of candidates n_c for the probability p_i is finite and each emerging probability q_ξ, $(1 \leq \xi \leq n_c)$ is also quantitatively calculated. In case of $c_{max} = \{7, 8\}$, the probability p_i has $n_c = 4$ candidates ρ_ξ as shown in Table I.

TABLE I
Parameters of Nuida's code in case of $c_{max} = \{7, 8\}$.

ρ_ξ	q_ξ
0.06943	0.24833
0.33001	0.25167
0.66999	0.25167
0.93057	0.24833

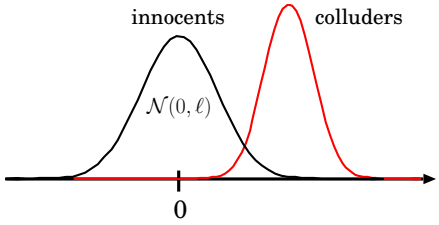

Fig. 1. Probability density function of score S_j.

Suppose that a pirated codeword $\boldsymbol{y} = (y_1, y_2, \ldots, y_\ell)$ is produced by colluders with a certain collusion strategy. The tracing algorithm of Tardos code calculates a similarity of codeword extracted from a pirated copy with candidates. The scoring function proposed by Škorić et al. [21] calculates the similarity score S_j between a pirated codeword and each suspicious codeword.

$$S_j = \sum_{i=1}^{\ell} S_{j,i} = (2y_i - 1)U_{j,i}, \tag{3}$$

where

$$U_{j,i} = \begin{cases} -\sqrt{\frac{p_i}{1-p_i}} & \text{if } X_{j,i} = 0 \\ \sqrt{\frac{1-p_i}{p_i}} & \text{if } X_{j,i} = 1 \end{cases}. \tag{4}$$

A user is detected as guilty if his score S_j exceeds a predefined threshold Z. As the probability catching innocents by mistake must be very small, the design of threshold Z is important. It is reported in [22] that the value of S_j for innocent users can be approximated by Gaussian distribution as shown in Fig. 1, because of the central limit theorem. Its mean and variance are zero and ℓ, respectively. Namely, the probability density function (PDF) is $PDF(S_j) \approx \mathcal{N}(0, \ell)$. As the precision of its tail probability is not assured by such a Gaussian approximation, the threshold Z must not be calculated by using the approximation. Instead, Furon et al. [23] proposed an efficient method to measure a tiny probability with a reasonable computational complexity. The proper threshold Z for a given false-positive probability can be calculated by using the method.

C. Decentralized Fingerprinting System[16]

One of the requirements for fingerprinting system is the asymmetric transaction between buyer (user) and seller. If both parties obtain the fingerprinted copy after a transaction, the seller will be able to distribute the copy by himself to frame an innocent user and an illegal redistributor will repudiate by claiming that the copy is created by the seller. The asymmetric transaction has been studied by introducing cryptographic protocols [1], [2], [3], [4], [5], [7] based on homomorphic encryption schemes. It enables a seller to embed fingerprint in multimedia content in an encrypted domain, and it assures an asymmetric property such that only the user can obtain uniquely fingerprinted content.

There are three parties in many fingerprinting systems, trusted center, user, and seller. A user and seller register at a trusted center and receive a certificate and items required for the cryptographic protocol of asymmetric transaction. In addition to the cryptographic technique, a fingerprinting code is required for collusion resistance. Thus, a trusted center selects secret parameters of fingerprinting code, and assigns each codeword to each user. When the cryptographic protocol is finished, the user obtains the fingerprinted copy in which his codeword is inserted. Once a pirated copy is found anywhere, the seller first extracts the fingerprinting codeword, and then requests a trusted center for identifying colluders. As correlation scores for all users must be calculated, the burden at the center is heavy.

In order to reduce the burden, a delegated server is introduced in [16]. The correlation scores are calculated in an encrypted domain by the seller, and the server checks which users are guilty by examining the scores after decryption of the ciphertexts received from the seller. As the parameters in a fingerprinting code must be kept secret, the trusted center encrypts the weighting parameters $U_{j,i}$, $(1 \le j \le N_u, 1 \le i \le \ell)$, and sends them to the seller at the time of registration. The advantage of the fingerprinting system is that a trusted center does not have to participate in a tracing protocol.

1) Registration: A trusted center selects a security parameter to generate parameters such as c_{max} and p_i, $(1 \le i \le \ell)$ of a fingerprinting code, and issues a codeword $\boldsymbol{X_j}$ to j-th user. Then, the weighting parameters $U_{j,i}$ are calculated for the codeword $\boldsymbol{X_j}$. As $U_{j,i}$ is not an integer, the center first multiplies a scaling parameter α to scale up its small number, and then, rounds the value into a nearest integer.

$$\tilde{U}_{j,i} = \text{round}\big(\alpha U_{j,i}\big), \tag{5}$$

where round() is a round function. Finally, the center encrypts $\tilde{U}_{j,i}$ to create an encrypted DB $Enc(\tilde{\boldsymbol{U}}_{\boldsymbol{j}})$:

$$Enc(\tilde{\boldsymbol{U}}_{\boldsymbol{j}}) = \big(Enc(\tilde{U}_{j,1}), \ldots, Enc(\tilde{U}_{j,\ell})\big). \tag{6}$$

The encrypted DB and the corresponding ID information are sent from a trusted center to a seller. It is noted that the number of ciphertexts in the encrypted DB is $N_u\ell$. A threshold $\tilde{Z}$ is calculated from $\tilde{U}_{j,i}$ by using the probabilistic algorithm [23]. The trusted center informs N_u, c_{max} and $\tilde{Z}$ to the server.

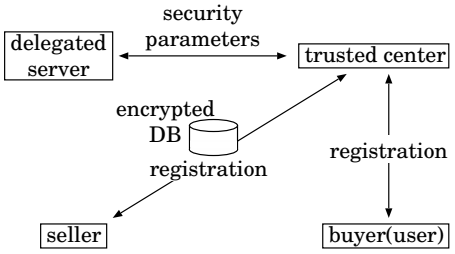

Fig. 2. Illustration of registration protocol.

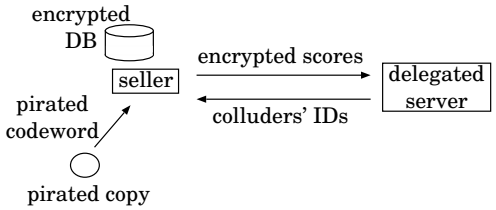

Fig. 3. Illustration of tracing protocol.

2) Tracing Protocol: A seller first extracts each element $y_i \in \{0, 1\}$ of a pirated codeword $\boldsymbol{y}$ from a pirated copy. Then, correlation scores $\tilde{S}_j^{sym}$ for users are calculated in an encrypted domain.

$$\begin{aligned} \bigodot_{i=1}^{\ell} Enc(\tilde{U}_{j,i})^{2y_i-1} &= Enc\Big(\sum_{i=1}^{\ell}(2y_i-1)\tilde{U}_{j,i}\Big) \\ &= Enc(\tilde{S}_j). \end{aligned} \tag{7}$$

Upon a tracing request from a seller, a delegated server checks the following three conditions.

1) The number of ciphertexts $Enc(\tilde{S}_j)$ is N_u.
2) The number of the scores which satisfy $\tilde{S}_j > \tilde{Z}$ is equal to or less than c_{max}.
3) The PDF of $\tilde{S}_j/\alpha$ is approximated by $\mathcal{N}(0, \ell)$.

Only when the above three conditions are satisfied, the server sends the decryption results back to the seller. According the results, the seller identifies the illegal users by checking the IDs corresponding to the index j.

Fig. 2 and Fig.3 illustrate the registration protocol and tracing protocol, respectively.

III. Proposed Fingerprinting System

A. Lifted ElGamal Cryptosystem

Different from the Paillier cryptosystem, the ElGamal cryptosystem [18] has been paid much attention. The original ElGamal cryptosystem is homomorphic with respect to multiplication. By encoding messages as exponents, the additive homomorphism can be satisfied. Such a variant is called "lifted ElGamal". The lifted ElGamal cryptosystem consists of the probabilistic polynomial time algorithms.

Let g be a generator of $\mathbb{G}$. A key generation algorithm takes a security parameter, and outputs a pair of public and secret keys. An encryption algorithm outputs $Enc(m)$ for a

given plaintext m while a decryption algorithm outputs m. For instance, we suppose that a secret key is k and a public key is $h = g^k$. Then, a ciphertext is calculated by using a random number r, $Enc(m) = (c_1 = g^r, c_2 = g^m h^r)$. A decryption algorithm calculates $c_2/c_1^k = g^m$ and solve the discrete logarithm of m. It is noticed that the encryption algorithm satisfy the additive homomorphism as follows:

$$\begin{aligned} Enc(m_1) \odot Enc(m_2) &= (g^{r_1} \cdot g^{r_2}, g^{m_1} h^{r_1} \cdot g^{m_2} h^{r_2}) \\ &= (g^{r_1+r_2}, g^{m_1+m_2} h^{r_1+r_2}) \\ &= Enc(m_1 + m_2). \end{aligned} \tag{8}$$

The decryption is only possible if the plaintext is known to be in a small subset of the plaintext space because the discrete logarithm of a generator with large order has to be solved.

The above cryptosystem can be implemented over EC in a finite field. An EC $E(\mathbb{F}_p)$ consists of element (x, y) of the form:

$$y^2 = x^3 + ax + b \bmod p, \tag{9}$$

where $x, y, a, b \in \mathbb{F}_p$, and an element of infinity point. Then, the addition and multiplication operations over EC are defined, which are different from ordinary arithmetic operations.

Let P is a base point on EC. From a secret key k, its public key is calculated by $Q = kP$, where the operation is EC multiplication. The ciphertext is calculated as $c_1 = rP$ and $c_2 = mP + rQ$ by using EC addition and EC multiplication. For the pair of ciphertext in EC-ElGamal cryptosystem, c_1 and c_2 are the points on EC, hence the size of ciphertext is 4 times larger than that of p.

The operation $\odot$ of EC-ElGamal cryptosystem is EC addition while the message space is an ordinary arithmetic addition.

$$\begin{aligned} &Enc(m_1) \odot Enc(m_2) \\ &\quad = \big(r_1 P + r_2 P, m_1 P + r_2 Q + m_2 P + r_2 Q\big) \\ &\quad = \big((r_1 + r_2)P, (m_1 + m_2)P + (r_1 + r_2)Q\big) \\ &\quad = Enc(m_1 + m_2). \end{aligned} \tag{10}$$

B. Look-Up Table

The message of Paillier cryptosystem can be chosen from positive integers less than N. Even if the scaling parameter α is set to be large, the decryption can be executed. On the other hand, the message of lifted EC-ElGamal cryptosystem must be small so that solving the discrete logarithm problem over EC is feasible. After the decryption of ciphertext $Enc(m)$, we obtain $mP = c_2 - kc_1$. Thus, we need to solve m from mP, which is discrete logarithm problem in general. If m is small, it is possible to use a look-up table. We stress that it is difficult to obtain mP from the ciphertext without a secret key.

The size of look-up table is strongly dependent on the scaling parameter α. Although a seller calculates correlation scores in an encrypted domain from a pirated codeword y_i by using encrypted weighting parameters $Enc(\tilde{U}_{j,i})$, the decryption is only performed at server's side. Hence, the size of score $\tilde{S}_j$ is important. As explained in Section II-B, the PDF of score S_j is shown in Fig. 1. After scaling-up by using α, the Gaussian distribution becomes $\mathcal{N}(0, \alpha\ell)$. From the statistical point of view, the lower bound for the score $\tilde{U}_{j,i}$ can be calculated for a given false probability. The upper bound is also calculated from the statistical distribution of colluders' score though its mean and variance are varied for the number of possible colluders. In case of single colluder, the value become maximum, which PDF can be approximated as $\mathcal{N}(2\alpha\ell/\pi, \alpha\ell(1-4/\pi^2))$ from the analysis in [22]. According to the lower and upper bounds, the look-up table can be calculated for a given α and false probability.

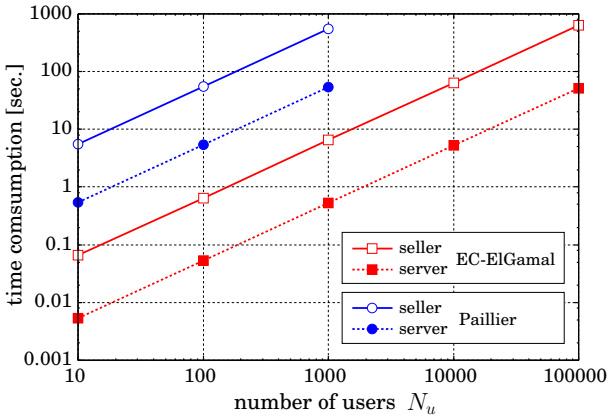

Fig. 4. Comparison of time consumption for EC-ElGamal and Paillier cryptosystems for 112-bit security level.

C. Time Consumption

As referring to the report in [24], the security level of RSA modulus N with size $\log_2 N = 2048$ is equivalent to 224-bit EC parameters, which is 112-bit security level. In such a case, the size of ciphertext of EC-ElGamal cryptsystem is 896 bits while the size of Paillier cryptosystem is 4096 bits.

In the proposed method, we use a scaling parameter α to ensure the precision of weighting parameters $U_{j,i}$. The degradation of traceability is measured under the following conditions. We use a Nuida code [15] with $c_{max} = 8$ and $\ell = 1024$.

We implemented the protocol and measure the time consumption under the following computer environment. The CPU is AMD Ryzen 7 2700X and the RAM memory is 32 GBytes. We use the GNU C compiler with version 7.4.0 and GNU multiple precision (GMP) library with version 6.1.2, at Ubuntu Linux 18.04 LTS. The Paillier cryptosystem is implemented by using the modular multiplication and exponentiation functions of the GMP library. For the implementation of lifted EC-ElGamal cryptosystem, we use the library[1] with the EC defined by "secp224k1" in [24]. The experimental results in [16] show that the traceability of fingerprinting code is not seriously degraded when the scaling parameter is $\alpha \geq 100$. Hence, we compare the time consumption by setting $\alpha = 100$, which results are plotted in Fig. 4. It is noted that the size

[1] https://github.com/aistcrypt/Lifted-ElGamal

TABLE II
COMPARISON OF TIME CONSUMPTION [SEC.] FOR DIFFERENT SECURITY LEVEL WHEN $\ell = 1024$ AND $N_u = 100$.

security level		96	112	128	192
Paillier	$\log_2 N$	1536	2048	3072	7680
	seller	34.033	54.539	103.492	414.914
	server	2.515	5.348	15.100	149.708
EC-ElGamal	$\log_2 p$	192	224	256	384
	seller	0.508	0.626	0.605	0.879
	server	0.036	0.051	0.056	0.112

of look-up table of candidates mP is less than 20MB under the above experimental condition. By using the EC-ElGamal cryptosystem, the computational complexity becomes about 100 times faster. The additive homomorphic operation in the encrypted domain is performed by the EC addition and EC multiplication in the EC-ElGamal cryptosystem while it is performed by the modular multiplication and exponentiation with modulus $\log_2 N^2 = 4096$ in the Paillier cryptosystem. Since the modulus p of the finite field of EC is 224-bit prime, the computational costs becomes much smaller.

Table II enumerates the time consumption at seller and server sides for different security levels. It is noticed that the lifted EC-ElGamal cryptosytem can supress the increase of computational costs more than the Paillier cryptosystem. Due to the difference of EC, the time consumption at $\log_2 = 256$ is slightly smaller than the time at $\log_2 = 224$.

From the above results, we can say that the use of lifted EC-ElGamal cryptosystem in the fingerprinting system efficiently reduces the computational costs and makes it practical.

IV. COMPRESSION OF DB

It is remarkable that p_i must be kept secret from a seller. Otherwise, a malicious seller can produce a codeword to frame an innocent user. If the plain values of the weighting parameters $U_{j,i}$ are observed, the seller can analyze the value of p_i and will be able to guess users' codewords.

Due to the secrecy of parameters p_i, $(1 \le i \le \ell)$, the weighting parameters $U_{j,i}$, $(1 \le i \le \ell,\ 1 \le j \le N_u)$ are encrypted in the conventional scheme. The number of ciphertexts in the encrypted DB is ℓN_u as shown in Fig. 5. By using the lifted EC-ElGamal cryptosystem, the size of ciphertext is 896 bits in 112-bit security level, while it is 4096 bits for the Paillier cryptosystem. The size is reduced by a factor of $7/32 = (894/4096)$. Then, the size is more than 1 GByte in case of $N_u = 10000$ and $\ell = 1024$, which is still heavy, while the size is more than 5 GBytes for the Paillier cryptosystem.

We focus on the discrete bias distribution of Nuida's code [15], and propose an efficient method to compress the DB. Because of the probabilistic construction of Nuida's code, each symbol $X_{j,i}$ of a codeword as well as the bias probability p_i are regarded as i.i.d. variables. As shown in Table I, the candidates for p_i are 4 when $c_{max} = \{7, 8\}$. Due to the rounding operation in Eq.(5), the precision of the parameters p_ξ and q_ξ is sacrificed in the conventional scheme.

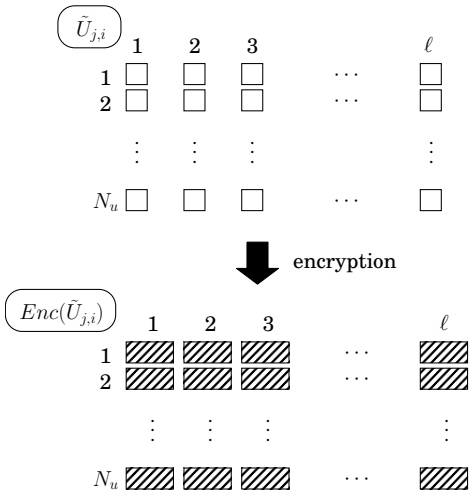

Fig. 5. Encrypted DB.

There are eight candidates for $\tilde{U}_{j,i}$ because $X_{j,i} \in \{0, 1\}$. For convenience, when the scaling parameter is $\alpha = 100$, we define the following four vectors $\boldsymbol{u_\xi}$, $1 \le \xi \le 4$:

$$\boldsymbol{u_1} = (u_{1,0}, u_{1,1}) = (-27, 366) \tag{11}$$

$$\boldsymbol{u_2} = (u_{2,0}, u_{2,1}) = (-70, 142) \tag{12}$$

$$\boldsymbol{u_3} = (u_{3,0}, u_{3,1}) = (-142, 70) \tag{13}$$

$$\boldsymbol{u_4} = (u_{4,0}, u_{4,1}) = (-366, 27) \tag{14}$$

It can be said from Table I that the number of each candidate ρ_ξ, $(1 \le \xi \le 4)$ is approximated as $\ell/4$. Thus, the candidates of $\tilde{U}_{j,i}$ can be classified as one of $\boldsymbol{u_\xi}$ at the i-th element.

It is known that the ciphertext of the ElGamal cryptosystem is indistinguishable under chosen plaintext attack. For given two ciphertexts of messages chosen from two-element plaintexts, an attacker can not distinguish whether their plaintexts are equal. As a random r is used at the encryption, there are several ciphertexts of a same plaintext.

Here, we consider to make 2^γ ciphertexts at i-th candidate of $\tilde{U}_{j,i}$, $(1 \le i \le \ell)$. Namely, the total number of ciphertexts is $\ell 2^\gamma$. Remember that the bias probability is $p_i = \Pr[X_{j,i} = 1]$ and its value is selected from one of four candidates ρ_ξ, $(1 \le \xi \le 4)$ in Table I. Due to the probabilistic construction of codewords, the number of elements $X_{j,i} = 1$ is expected to be $\rho_\xi 2^\gamma$ when $p_i = \rho_\xi$. In order not to leak the information about p_i, $\rho_\xi 2^\gamma$ ciphertexts are calculated from a plaintext $u_{\xi,1}$ using different random numbers, and the others are calculated from $u_{\xi,0}$.

For simplicity, we assume the order of the bias probability as follows:

$$p_i = \begin{cases} \rho_1 & 1 \le i \le \ell/4 \\ \rho_2 & \ell/4 + 1 \le i \le \ell/2 \\ \rho_3 & \ell/2 + 1 \le i \le 3\ell/4 \\ \rho_4 & 3\ell/4 + 1 \le i \le \ell \end{cases}, \tag{15}$$

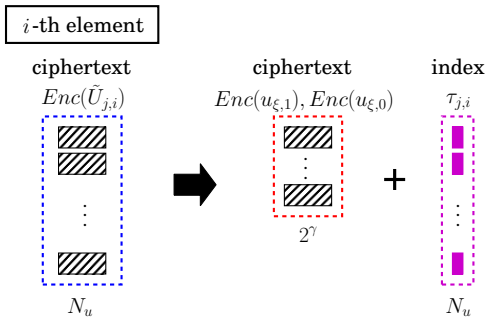

Fig. 6. Proposed DB.

TABLE III
COMPARISON OF REQUIRED STORAGE [MB] WHEN $\gamma = 8$.

ℓ	N_u	Paillier[16]	EC-ElGamal	Compressed DB
1024	100	52.4	11.5	23.5
	1000	524.3	114.7	23.6
	10000	5,242.9	1,146.8	24.5
	100000	52,428.8	11.468.8	33.7
2048	100	104.8	22.9	47.0
	1000	1,048.6	229.4	47.1
	10000	10,485.7	2,293.8	48.0
	100000	104,857.6	22,937.6	57.2

and assume that the first $\rho_\xi 2^\gamma$ ciphertexts are $Enc(u_{\xi,1})$ and the others are $Enc(u_{\xi,0})$. Then, for each $X_{j,i}$, we assign an index number $\tau_{j,i} \in [1, \rho_\xi 2^\gamma]$ randomly if $X_{j,i} = 1$; otherwise $\tau_{j,i} \in [\rho_\xi 2^\gamma + 1, 2^\gamma]$. The index number $\tau_{j,i}$ indicates one ciphertext associated with $X_{j,i}$. When a pirated copy is found and its codeword $\boldsymbol{y}$ is extracted, the encrypted score $Enc(\tilde{S}_{j,i})$ is calculated from the associated ciphertexts by referring to $\tau_{j,i}$ and 2^γ ciphertexts. Therefore, N_u ciphertexs in the conventional scheme is reduced to 2^γ ciphertexts and an index table at i-th element, which is illustrated in Fig. 6. In a real situation, it is difficult to guess p_i by observing the list of ciphertexts and index table as the order of bias probability is random in general.

It is remarkable that the number of ciphertexts are constant with respect to the number of users N_u. In case of of $N_u = 10000$ and $\ell = 1024$ same as the previous example, the total size of ciphertexts is less than 24MBytes when $\gamma = 8$. It is noticed that $\tau_{j,i}$ can be represented by γ bits. Thus, we create a two dimensional index table $\tau_{j,i}$, $(1 \le i \le \ell, 1 \le j \le N_u)$, which total size is $\gamma \ell N_u / 8$ Bytes. Then, the size of index table becomes 10MBytes in the above condition. In total, the size of compressed DB is 34MBytes, which is much smaller than the original size and is realistic size. The detailed quantitative comparisons are shown in Table III. As a consequence, we can say that the efficiency on the implementation is drastically improved in the proposed method.

V. CONCLUSIONS

In this paper, we proposed an efficient method for decentralized fingerprinting system by introducing the lifted EC-ElGamal cryptosystem. As the bit length of the size of correlation score is relatively small, it is possible to obtain the encrypted score value within reasonable computational resources. Under 112-bit security level, the time consumption at both seller and delegated server become about 100 times faster than the conventional scheme. We stress that the implementation over EC enables us to reduce the size of the ciphertext as well as the computational complexity. The size of DB can be further compressed by using the index table assigning to all symbols of users' codewords.

ACKNOWLEDGMENT

This research has been supported by the Kayamori Foundation of Information Science Advancement.

REFERENCES

[1] B. Pfitzmann and M. Schunter, "Asymmetric fingerprinting," in *EUROCRYPT1996*. 1996, vol. 1070 of *LNCS*, pp. 84–95, Springer, Heidelberg.
[2] B. Pfitzmann and A. Sadeghi, "Coin-based anonymous fingerprinting," in *EUROCRYPT1999*. 1999, vol. 1592 of *LNCS*, pp. 150–164, Springer-Verlag.
[3] B. Pfitzmann and A. Sadeghi, "Anonymous fingerprinting with direct non-repudiation," in *ASIACRYPT2000*. 2000, vol. 1976 of *LNCS*, pp. 401–414, Springer-Verlag.
[4] J. Camenisch, "Efficient anonymous fingerprinting with group signatures," in *ASIACRYPT2000*. 2000, vol. 1976 of *LNCS*, pp. 415–428, Springer-Verlag.
[5] N. Memon and P. W. Wong, "A buyer-seller watermarking protocol," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 643–649, 2001.
[6] M. Kuribayashi and H. Tanaka, "Fingerprinting protocol for images based on additive homomorphic property," *IEEE Trans. Image Processing*, vol. 14, no. 12, pp. 2129–2139, 2005.
[7] C. Lei, P. Yu, P. Tsai, and M. Chan, "An efficient and anonymous buyer-seller watermarking protocol," *IEEE Trans. Image Process.*, vol. 13, no. 12, pp. 1618–1626, 2004.
[8] M. Deng, T. Bianchi, A. Piva, and B. Preneel, "An efficient buyer-seller watermarking protocol based on composite signal representation," in *MM& Sec2009*, 2009, pp. 9–18.
[9] J. D.-Ferrer and D. Megías, "Distributed multicast of fingerprinted content based on a rational peer-to-peer community," *Computer Communications*, vol. 36, no. 5, pp. 542–550, 2013.
[10] D. Megías and J. D.-Ferrer, "Privacy-aware peer-to-peer content distribution using automatically recombined fingerprints," *Multimedia Systems*, vol. 20, no. 2, pp. 105–125, 2014.
[11] D. Megías and A. Qureshi, "Collusion-resistant and privacy-preserving P2P multimedia distribution based on recombined fingerprinting," *Expert Systems with Applications*, vol. 71, pp. 147–172, 2017.
[12] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inform. Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.
[13] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1069–1087, 2003.
[14] Gábor Tardos, "Optimal probabilistic fingerprint codes," *J. ACM*, vol. 55, no. 2, pp. 1–24, 2008.
[15] K. Nuida, S. Fujitu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa, and H. Imai, "An improvement of discrete Tardos fingerprinting codes," *Designs, Codes and Cryptography*, vol. 52, no. 3, pp. 339–362, 2009.
[16] M. Kuribayashi and N. Funabiki, "Decentralized tracing protocol for fingerprinting system," *APSIPA Transactions on Signal and Information Processing*, vol. 8, pp. e2, 2019.
[17] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT1999*. 1999, vol. 1592 of *LNCS*, pp. 223–238, Springer, Heidelberg.
[18] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proceedings of CRYPTO 84 on Advances in Cryptology*, New York, NY, USA, 1985, pp. 10–18, Springer-Verlag New York, Inc.

[19] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of Paillier's probabilistic public-key system," in *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, London, UK, UK, 2001, PKC2001, pp. 119–136, Springer-Verlag.

[20] I. Damgård, M. Jurik, and J. B. Nielsen, "A generalization of Paillier's public-key system with applications to electronic voting," *Int. J. Inf. Sec.*, vol. 9, no. 6, pp. 371–385, 2010.

[21] B. Škorić, S. Katzenbeisser, and M. Celik, "Binary and q-ary Tardos codes, revisited," *Designs, Codes and Cryptography*, vol. 74, no. 1, pp. 75–111, 2015.

[22] B. Škorić, S. Katzenbeisser, and M. Celik, "Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes," *Designs, Codes and Cryptography*, vol. 46, no. 2, pp. 137–166, 2008.

[23] F. Cérou, T. Furon, and A. Guyader, "Experimental assessment of the reliability for watermarking and fingerprinting schemes," *EURASIP J. Information Security*, vol. 2008, 2008.

[24] Certicom Research, "Recommended elliptic curve domain parameters," Standards for Efficient Cryptography (SEC) 2, September 2000, https://www.secg.org/sec2-v2.pdf.