Improved Algebraic Traitor Tracing Scheme

Chunyan Bai and Guiliang Feng

Center for Advanced Computer Studies,
University of Louisiana at Lafayette,
Lafayette, LA 70504, USA
{cxb7146, glf}@cacs.louisiana.edu
http://www.cacs.louisiana.edu

Abstract. In this paper, we try to use algebraic-geometric codes (AG codes) to solve the traitor tracing problem over the broadcast channel. The scheme is shown by using AG codes to construct the linear space tracing code Γ, which is the base for the distributor to create private keys for each authorized subscribers. The obtained public key tracing scheme is deterministic and can trace all the participated traitors. Compared to the Reed-Solomon code (RS code) based public key traitor tracing scheme, our scheme can accommodate more users and tolerate more colluders given a fixed length of private keys.

Index Terms: algebraic-geometric code, Reed-Solomon code, traitor tracing, broadcast.

1 Introduction

With the rapid development of new IT technologies and electronic commerce, broadcast encryption is used in a wide range of situations such as video-on-demand, multi-party teleconferencing, stock quote distribution and updating softwares. In these cases, everyone can receive the encrypted message, but only a set of registered users are authorized to decrypt and discover the original message. This is implemented by using a public key to encrypt the message and broadcast it into the channel. Also, different decryption keys are generated and distributed to each legitimated user to guarantee him to obtain the subscribed service.

The authorized users, however, may have the temptation to distribute their description keys or the decrypted message further to unauthorized users without the permission of the distributor. This message leakage will cause great loss for the distributor. Such unauthorized access to data is known as *piracy*. Those authorized users who allow other non-authorized users to obtain data are called *traitors* and those unauthorized users are called *pirate users*. This problem can be solved by assigning different decryption keys to different user, then the compromised key could be used to trace back to its origin. While, a new problem will come. If a group of users try to collude and create a new decryption key, which may hide the identity of each colluders. How to prevent such message leakage is the key for the security of broadcast communication.

J. Zhou, M. Yung, Y. Han (Eds.): ACNS 2003, LNCS 2846, pp. 217–227, 2003.

There are two non-exclusive approaches for the distributor to protect himself from the non-authorized redistribution. One option is to deter users from revealing their personal keys to others, which is referred to as *self enforcement*; Another option is to trace the corrupt users back, which is known as *traitor tracing*, and revoke them from further using the service. The self enforcement property is obtained by inserting some sensitive private information, such as bank account or credit card number, of traitors into his personal keys. Then the traitor will feel reluctant to redistribute his personal key to others. Although self enforcement schemes can prevent small scale piracy and can make it harder for pirates to obtain decryption keys, we will concentrate on traitor tracing in this paper since we assume that traitors DO want to reveal their keys to unauthorized users.

Traitor tracing refers to the task of identifying the keys used for generating the pirate key. In most of the available traitor tracing schemes, the keys have some combinatorial properties and tracing is probabilistic[1-7,13,14,18]. These work follows the traitor tracing model proposed by Chor,Fiat and Naor[1]. Each message is encrypted by a public key and sent to the channel together with different private keys for each individual user. Tracing is based on the combinatorial properties of the keys. It is guaranteed from these schemes that at least one of the traitors will be traced with high probability if less than k traitors participate in the collusion, which is called k-collusion resistant. Further work have been done to make the scheme full tracing[10,15,17]. These schemes are deterministic and catches all of the traitors who contributed to the attack based on the number theoretic assumption. Dynamic traitor tracing schemes in [8,9] are designed to combat the less common scenario where a pirate publishes the periodical access control keys on the Internet or simply rebroadcasts the content via an independent pirate network. Authors in [12,13] discuss how to revoke those traitors after they have been discovered. Recent work in [16,18,19] try to break some of the available traitor tracing schemes and propose new algebraic schemes by using error-correcting encoding and decoding. We will review the related work in Section 2.

In this paper, algebraic-geometric codes are used to construct the base for the public key traitor tracing scheme. The obtained scheme accommodate more users and achieve a higher threshold of tracing compared to the existed Reed-Solomon code based scheme.

The rest of this paper is organized as follows: In section 2, we describe work related to traitor tracing. Section 3 outlines an overview of the problem as well as definitions used in our approach. Section 4 characterizes and discusses the new algebraic-geometric codes based traitor tracing scheme. Finally, we summarize the conclusion and future work in section 5.

2 Related Work

The first attempt to deal with the traitor tracing problem was proposed by Chor, Fiat and Naor[1] and generalized by Stinson and Wei in [5]. According to Chor et.al., k-resilient traceability schemes can be implemented, that is, at least one

traitor will be revealed on the confiscation of a pirate decoder if there are at most k traitors. Although Chor's work gives the model for solving the traitor tracing problem, their scheme is inefficient and non-constructive. Stinson and Wei showed some explicit constructions by using combinatorial designs, their work is better for small values of k and n, where k is the number of traitors and n is the number of users. Both of these two schemes are private key encryption scheme, that is, they are symmetric in the sense that legitimate users of the broadcast information share all their secrets with the information provider. Thus they cannot provide non-reputation. Pfitzmann [2] pointed out this problem and introduced asymmetric traceability schemes in which the information provider cannot frame an innocent user and no user can abuse the system without being detected. In [6,11,14,18], asymmetric traitor tracing schemes are further discussed and designed which make traitor tracing designs more practical in the case of disputation between information provider and users.

Traitor tracing scheme can be designed to operate against any pirate decoder with $non - negligible$ success probability, which is called $Fully\ resilient\ schemes$. Although the security quality of the fully resilient schemes are good (they perform better than breaking the underlying encryption system), their complexity costs are too high under some circumstances. Naor and Pinkas introduced threshold tracing schemes in [3] which can trace the source of keys of pirate decoders with probability greater than some threshold. These schemes present a dramatic reduction in the overhead compared to fully resilient schemes while provide quality that are good enough for most cases.

After noticing that all the discussed traceability schemes assume that the data supplier should assign the keys after he has determined whom the authorized users are, which maybe unreasonable because changes between authorized and unauthorized users might be frequent, Stinson and Wei [4] investigated the key preassigned traceability schemes in which the personal keys can be assigned before the authorized users are determined. The schemes have better traceability and are more efficient in the sense of information rate and broadcast information rate. [8] and [9] tried to combat the less common scenario where a pirate publishes the periodically access control keys on the Internet or simply rebroadcasts the content via an independent pirate network. The schemes are accomplished by using the watermarking techniques, which allows the broadcaster to generate different versions of the original content, with no noticeable degradation in the content quality. Watermarks found in the pirate copy are used to trace its supporting traitors. These schemes can deal with not only the case that the private keys are leaked, but also the case that the private messages are rebroadcasted.

In all the above work, traitor tracings are designed in symmetric encryption system, thus requires the information provider be coincident with the administrator of the secure broadcasting infrastructure. However, it is highly desirable to divide these roles, thus the appearance of public-key traitor tracing schemes in [6,7,10,11,18]. Public-key traitor tracing schemes can overcome the problem of non-reputation. The first public-key traceability scheme was shown in [6]. But

this scheme was broken by Stinson and Wei in [4], and Boneh and Franklin in [10].

The next public key tracing scheme came with Boneh and Franklin in 1999[10]. The construction of keys is algebraic, which combines the theory of error correcting codes to discrete logarithm representation problems, and tracing is deterministic. The scheme gains full tracing, that is, if at most k traitors have participated in generating a new key, they can all be traced. Moreover, the tracing algorithm is error free. Innocent users are never blamed. Furthermore, the scheme is efficient in complexity. Following the same idea, some other work [15,17] have been done. [15] presented a general attempt to make the scheme long-lived, that is, the server can adapt its encryption procedure to the presence of pirate decryption keys and modify the transmission to reach the legitimate users only. [17] gave an attack on [10] and proposed efficient modified scheme to make it robust. Our scheme in this paper is also based on the work in [10] because it is the only scheme which can trace all the traitors who participates in the message leakage.

Most recent work on traitor tracing are [16], [18] and [19]. The scheme in [16] is specifically designed for tracing fingerprint media data such as images and video data. Generalized Reed-Solomon codes and their soft-decision decoding algorithms are utilized to present a powerful tracing scheme. [18] broke two of previous tracing schemes and present a new asymmetric public-key traitor tracing procedure with detailed proof of the traceability and security. In [19], a coding theoretic approach is used to produce a fast traitor tracing scheme. It is shown that when suitable error-correcting codes are used to construct traceability schemes, and fast list decoding algorithms are used to trace, the runtime of the tracing algorithm is polynomial in the codeword length. Also, the question of what the information provider should do after finding the traitors is considered. [12] and [13] talk about how to combine tracing scheme with revocation scheme and make the broadcast information distribution more robust. This direction is of great significance for future research.

3 Overview

Traitor tracing schemes help in three aspects of piracy prevention: they deter users from cooperating with pirates, they identify the pirates and enable to take legal actions against them, and they can be used to disable active pirate users. We will address the (k, n) traitor tracing problem in this paper, that is, to identify the source of at least one traitor if there are at most k traitors among n authorized users. a (k, n)-traceability scheme has four components: key generation, an encryption algorithm, a decryption algorithm and a tracing algorithm.

The (k, n) public key traitor tracing scheme proposed by Boneh and Franklin [10] provided a deterministic FULL tracing approach, that is, it can catch ALL traitors without accusing any innocent users as long as the number of traitors is at or below a collusion bound k. The construction of the scheme combines the

theory of error correcting codes to discrete logarithm representation problems. Each private key is a different solution vector for a discrete logarithm problem with respect to a fixed base of field elements. The fixed base is used to construct the public encryption key. This was the first time that the idea of error-correcting code was combined with traitor tracing. The resulting scheme is more efficient than other available schemes in terms of algorithm complexity(the length of the keys) and the tracing performance. Next, we will summarize the idea of the so-called Boneh-Franklin scheme (BF scheme) discussed in [10].

First, let's review some definitions and assumptions that are useful for BF scheme and our own scheme.

Definition 1. (Representation[10]) *If* $y = \prod_{i=1}^{2k} h_i^{\delta_i}$, *we say that* $\boldsymbol{\delta} = (\delta_1, \delta_2, ..., \delta_{2k})$ *is a representation of* y *with respect to the base* $h_1, h_2, ..., h_{2k}$.

Definition 2. (Convex Combination) *We say that a vector* $\boldsymbol{d}$ *is the convex combination of vectors* $\boldsymbol{d_1}, \boldsymbol{d_2}, ..., \boldsymbol{d_m}$ *if* $\boldsymbol{d} = \sum_{i=1}^{m} \alpha_i \boldsymbol{d_i}$, *where* $\alpha_1, ..., \alpha_m$ *are scalars such that* $\sum_{i=1}^{m} \alpha_i = 1$.

According to [10], if $\boldsymbol{d_1}, \boldsymbol{d_2}, ..., \boldsymbol{d_m}$ are representations of y with respect to the same base and $\boldsymbol{d}$ is the convex combination of $\boldsymbol{d_1}, \boldsymbol{d_2}, ..., \boldsymbol{d_m}$, then $\boldsymbol{d}$ is also a representation of y.

The BF scheme works in a multiplicative cyclic group G_q, where q is a prime. The security of the scheme is based on the difficulty of the Decisional Diffie-Hellman problem(DDH) over G_q.

Definition 3. (DDH[20]) *Let* $g \in G_q$ *be a generator. Consider triples of the form R:* $\langle g^a, g^b, g^c \rangle$ *and triples of the form D:*$\langle g^a, g^b, g^{ab} \rangle$, *where* $a, b, c < order(g)$. *A predicate solves the DDH problem if it can distinguish the collection D from the collection R.*

Loosely speaking, the DDH assumption states that no efficient algorithm (polynomial time algorithm) can distinguish between the two distributions $< g^a, g^b, g^{ab} >$ and $< g^a, g^b, g^c >$. The DDH assumption is useful for constructing efficient cryptographic primitives with very strong security guarantees. These include the Diffie-Hellman key agreement protocol, the El Gamal encryption scheme and so on.

Next, let's see the four components in the construction of BF (k, n)-traceability scheme.

Let G be a $(n - 2k) \times n$ matrix as:

$$G = \begin{pmatrix} 1 & 1 & 1 \ldots & 1 \\ 1 & 2 & 3 \ldots & n \\ 1^2 & 2^2 & 3^2 \ldots & n^2 \\ . & . & . \; . & . \\ . & . & . \; . & . \\ 1^{n-2k-1} & 2^{n-2k-1} & 3^{n-2k-1} \ldots & n^{n-2k-1} \end{pmatrix} (mod\ q)\,. \tag{1}$$

Since any vector in the span of the rows of A corresponds to a polynomial of degree at most $n-2k-1$ evaluated at the points $1, 2, ..., n$, G is actually the generator matrix of a $(n, 2k)$ Reed-Solomon Code. Let $w_1, ..., w_{2k}$ be a basis of the linear space of vectors satisfying $G\boldsymbol{x} = 0 \ mod \ q$ and use these $2k$ vectors as the columns of a matrix, an $n \times 2k$ matrix Γ is obtained as

$$\Gamma = \left(\boldsymbol{w_1} \ \boldsymbol{w_2} \ \boldsymbol{w_3} \ ... \ \boldsymbol{w_{2k}} \right) \tag{2}$$

View the line vectors of matrix Γ as a set of codewords, the matrix Γ provides n codewords of length $2k$. Based on this set of codeword, the (n, k)-traceability schemes is designed as:

Key Generation : For $i = 1, ..., 2k$, the data supplier chooses a random $a_i \in Z_q$ and computes $y_i = g^{a_i}$, where g is a generator of G_q and $\sum_{j=1}^{2k} a_j \gamma_j^{(i)} \neq 0$. Then the public key is published as $< z, y_1, ..., y_{2k} >$, where $z = \prod_{i=1}^{2k} y_i^{\beta_i}$ for random chosen $\beta_1, ..., \beta_{2k} \in Z_q$. The personal decryption key of user i is computed as

$$\theta_i = (\sum_{j=1}^{2k} a_i \beta_j)/(\sum_{j=1}^{2k} a_j \gamma_j^{(i)}) \ (mod \ q),$$

where $\gamma^{(i)} = (\gamma_1^{(i)}, ..., \gamma_{2k}^{(i)}) \in \Gamma$ is the i'th codeword of Γ.

Encryption : For a message $M \in G_q$, the information provider computes the ciphertext as $h = (M * z^r, y_1^r, ..., y_{2k}^r)$, where $r \in Z_q$ is a random number.

Decryption : Each user i computes M from h as follows by using θ_i as:

$$M = M * z^r / U^{\theta_i}, \ where \ U = \prod_{j=1}^{2k} (y_j^r)^{\gamma_j^{(i)}}.$$

Traitor Tracing : It is indicated in [10] that if $\boldsymbol{d_1}, ..., \boldsymbol{d_m} \in Z_q^{2k}$ are representations of y, then the convex combinations are the only new representations of y that can be efficiently constructed from $\boldsymbol{d_1}, ..., \boldsymbol{d_m}$. Any representation $(\delta_1, ..., \delta_{2k})$ of y with respect to the base h_i can be used as a decryption key. This is because $\prod_{j=1}^{2k} (h_j^a)^{\delta_j} = y^a$. If the traitors form a new decryption key $\boldsymbol{d}$ by making the convex combination of at most k decryption keys $\boldsymbol{d_1}, ..., \boldsymbol{d_m}$, then those traitors who participated in forming $\boldsymbol{d}$ can be efficiently determined by finding a vector $\boldsymbol{w} \in F_q^n$ of Hamming weight at most k such that $\boldsymbol{w} * \Gamma = \boldsymbol{d}$. Berlekamp's algorithm is used in this procedure to find traitors.

The (k, n)-traceability scheme in [10] is based on the theory of Reed-Solomon codes and the problem of discrete log representation. Traceability follows from the hardness of discrete log. The private key in all cases is just a single element of a finite field and can be as short as 160 bits. The complexity of encryption and decryption is independent of the size of the coalition under the pirate's control. But the key will be long if the finite field is big.

Notice that for a Reed-Solomon code over the finite field $GF(q)$, the codeword length N has to be less than or equal to q, i.e., $N \leq q$. Conversely, for an algebraic-geometric code, the codeword length can be greater than q.

This presents the possibility of using algebraic-geometric codes instead of Reed-Solomon codes to further reduce the bits needed to represent the private keys and consequently the complexity of the scheme.

4 Traitor Tracing Based on Algebraic-Geometric Codes (AG Codes)

4.1 Background on Algebraic-Geometric Codes

We now give the definition of AG codes, also known as geometric Goppa codes following the notation in [21].

Let X be an absolutely irreducible smooth projective algebraic curve of genus g over the finite field $GF(q)$. Consider an ordered set $P = \{P_1, P_2, ..., P_n\}$ of distinct rational points on X and a divisor D on X, rational over $GF(q)$. For simplicity, let us assume that the support of D is disjoint from P.

Definition 4. (Algebraic – Geometric Code) *The linear space $L(D)$ of rational functions on X associated with D yields the linear evaluation map*

$$L(D) \to F_q^n$$

defined by

$$f \to (f(P_1), ..., f(P_n)).$$

The image of this map is a linear code $C_L(P, D)$, which we call an algebraic-geometric code, or a geometric Goppa code.

Guruswami and Sudan give an efficient decoding algorithm of RS code and AG code in [22]. For those who need more discussion about AG code, please refer to [23],[24] and [25].

4.2 AG Codes Based Traitor Tracing Scheme

In this section, we will discuss how to use Algebraic-geometric codes to construct the *linear space tracing code Γ*, which is used to create private keys for each user.

Hermitian curve based construction. An AG code can be constructed from affine plane curves [25]. Let $H \doteq \{h_1, h_2, ..., h_r, ..., h_v\}$ be a sequence of vectors in F_q^n, where $h_r \doteq (h_{r1}, h_{r2}, ..., h_{rn})$, and let $S(r)$ be the linear space over F_q spanned by the first r vectors of H. Let $\hat{H} \doteq \{\hat{h}_1, \hat{h}_2, ..., \hat{h}_\mu, ..., \hat{h}_u\}$ be a supplementary sequence and $S(r, u)$ be the linear space over F_q spanned by only the first r vectors of H and all the vectors of $\hat{H}$. In most cases, the supplementary sequence $\hat{H}$ may be empty, that is, $u = 0$. Let

$$H_r \doteq \begin{bmatrix} h_1 \\ h_2 \\ ... \\ h_r \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & ... & h_{1n} \\ h_{21} & h_{22} & ... & h_{2n} \\ ... & ... & ... & ... \\ h_{r1} & h_{r2} & ... & h_{rn} \end{bmatrix}. \tag{3}$$

We define $H_r^* \doteq \begin{bmatrix} \hat{H} \\ H_r \end{bmatrix}$ to be a parity check matrix of a linear code over F_q, denoted by C_r. For a special case of $u = 0$, H_r^* reduces to H_r. Next we will show how to include the current AG codes to the above H sequence construction.

Let χ be an algebraic geometric curve with the genus g and let $P_1, P_2, ..., P_n$ and P_∞ be the set of rational points over a finite field F_q. Let $H \doteq \{f_1, f_2, ..., f_n\}$ be a sequence of functions and $\hat{H} \doteq \phi$, then the linear code C_r defined by H_r^* is an $(n, n-r)$ AG code $C_\Omega(D, G)$, where $D = P_1+P_2+...+P_n$ and $G = (r+g-1)P_\infty$.

Improved geometric Goppa codes can be constructed from algebraic-geometric curves based on a well-behaving sequence H [23]. The key of the construction lies in the definitions of *weight* $w(x)$ and *total order*. Next, we will show how to construct an improved geometric code from the Hermitian curves.

Let a location set LS be a set of all the rational points of the Hermitian curve $x^{q+1} + y^q + y = 0$ over $GF(q^2)$, there should have a total of q^3 elements in LS. Let $w(x) = q$ and $w(y) = q + 1$, the total ordering of monomials $x^{i_1}y^{i_2}$ can be determined. After deleting all the monomials linearly dependent on their previous monomials in H, a well-behaving sequence $H(|H| = q^3)$ will be obtained, from which a $(n, n-r, d^*) = (q^3, q^3 - r, d^*)$ AG code can be constructed. H_r^* will be treated as the parity check matrix for $(n, n-r, d^*)$ AG code, where r is determined by the given designed minimum distance d^* according to [23]. And this $(n-r) \times n$ matrix will be utilized as the G matrix, which is further used to create the linear space tracing code Γ in the traitor tracing scheme. Let us see an example about the detailed construction of the H sequence.

Consider the Hermitian curve $x^5 + y^4 + y = 0$ over $GF(4^2)$. Let $w(x) = 4$ and $w(y) = 5$, then we have

$$H' = \{1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3, x^4, x^3y,$$
$$x^2y^2, xy^3, y^4, x^5, x^4y, x^3y^2, x^2y^3, xy^4, x^6,$$
$$y^5, x^5y, x^4y^2, x^3y^3, x^2y^4, x^7, xy^5, x^6y, y^6, x^5y^2, ...\}$$

After deleting all the monomials denoted by $y^{i_2}x^{i_1}$, which are linearly dependent on their previous monomials, a well-behaving sequence H is found to be:

$$H = \{1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3, x^4, x^3y, x^2y^2,$$
$$xy^3, y^4, x^4y, x^3y^2, x^2y^3, xy^4, y^5, x^4y^2, x^3y^3,$$
$$x^2y^4, xy^5, y^6, x^4y^3, x^3y^4, x^2y^5, xy^6, y^7, x^4y^4,$$
$$x^3y^5, x^2y^6, xy^7, y^8, x^4y^5, x^3y^6, x^2y^7, xy^8,$$
$$y^9, x^4y^6, x^3y^7, x^2y^8, xy^9, y^{10}, x^4y^7, x^3y^8, x^2y^9,$$
$$xy^{10}, y^{11}, x^4y^8, x^3y^9, x^2y^{10}, xy^{11}, y^{12}, x^4y^9,$$
$$x^3y^{10}, x^2y^{11}, y^{13}, x^4y^{10}, x^3y^{11}, y^{14}, x^4y^{11}, y^{15}\}.$$

$|H| = 64 = 4^3$. The weight sequence of H is:

$$W = \{0, 4, 5, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20,$$

$$21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32,$$

$$33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44,$$

$$45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56,$$

$$57, 58, 59, 60, 61, 62, 63, 65, 66, 67, 70, 71, 75\}.$$

We can see that the weight sequence is not a continuous integer sequence, i.e., some gaps exist. For example, $1, 2, 3$ between 0 and 4, $6, 7$ between 5 and 8. Such gaps provide the Hermitian curve with a genus, which is related to the performance of the constructed AG code. That is, the smaller the genus is, the greater the efficiency of the linear codes C_r.

Suppose we want to construct an improved geometric Goppa code with length $n = 64$ and designed minimum distance $d^* = 6$ over $GF(4^2)$. Using the construction 2.1 from [23], we have $r = 7$ and thus $H_7^* = (1, x, y, x^2, xy, y^2, x^3, y^3, x^4)^T$. This $(n-r) \times n = 9 \times 64$ matrix is the parity check matrix for $(64, 55, 6)$ AG code and will be utilized as the G matrix in the public key traitor tracing scheme.

The tracing algorithm, that is, how to efficiently determine the unique set of vectors in Γ used to construct the pirate key $\boldsymbol{d}$ can be implemented by the decoding algorithm of AG code in [24].

5 Discussion and Comparison

Notice that for a Reed-Solomon code over the finite field $GF(q)$, the codeword length n has to be less than or equal to q, i.e., $n \leq q$. Conversely, for an algebraic-geometric code, the codeword length can be designed with any desired length n, which can be greater than q. Because the codeword length determines the total number of users in the system, algebraic-geometric code based scheme will accommodate more users than Reed-Solomon code based scheme if both the schemes are defined on the same finite field $GF(q)$.

Now, let us see the complexity of the given pubic key traitor tracing scheme.

Suppose n is the total number of users in the system and k is the bound of the number of traitors, that is, the maximum number of traitors in a confiscation. In the presented scheme, the size of the ciphertext is $2k + 1$ elements of the given finite field $GF(q)$ and the size of the user-key is $O(1)$, that is, only one finite field element. The encryption key size is $2k + 1$ elements of the finite field. So the number of bits used to represent the element in a finite field is one of the key factors that will greatly affect the overload of the scheme.

If we use the parity check matrix of $(n, 2k, d)$ Reed-Solomon code over $GF(q^3)$ as the G matrix in the (n, k)-traceability scheme, then the codeword length n is less than or equal to the order of the finite field, q^3. That is, the maximum number of users that the scheme can accommodate is $n = q^3$ (this comes from the construction of the linear space tracing code Γ). In order to accommodate the

same number of users in the system, we can instead use an algebraic-geometric code which is defined on the Hermitian curve

$$x^{q+1} + y^q + y = 0 \ \ over \ \ GF(q^2).$$

Since the given Hermitian curve has a total of q^3 roots pairs $(x_i, y_i), i = 1, 2, ..., q^3$, the codeword length of the corresponding algebraic-geometric code can be designed as q^3 although the code is designed on finite field $GF(q^2)$. That is, the algebraic-geometric code based tracing scheme can also accommodate q^3 users. As we know that in order to represent each element of finite field $GF(q^3)$, a total number of $3log(q)$ bits is needed. On the contrary, $2log(q)$ bits are enough to represent each element of finite field $GF(q^2)$. That is, if we use AG code defined on Hermitian curve $x^{q+1} + y^q + y = 0$ over $GF(q^2)$ to construct the linear space tracing code Γ, $log(q) \times (4k + 2)$ bits will saved compared to RS code based tracing scheme in order to create the cipertext and the encryption key. Also, $n * log(q) = q^3 \times log(q)$ number of bits will be saved for a total of q^3 users in the system. Consequently, the AG code based traitor tracing scheme greatly eliminates the complexity overload compared to the RS code based scheme.

6 Conclusion and Future Work

In this paper, we use algebraic-geometric code to construct the linear space tracing code Γ, which is the key step in public key traitor tracing scheme. The obtained public key tracing scheme is deterministic and can trace all the participated traitors. Compared to the Reed-Solomon code based public key traitor tracing scheme, our scheme can accommodate more users and tolerate more colluders given a fixed length of private keys. Also, the complexity overload of the AG code based scheme has been greatly eliminated compared to the RS code based scheme, which makes the AG code based scheme more feasible in practice.

Although the scheme presented in this paper is an pubic key tracing scheme, it is not an asymmetric scheme. That is, non-reputation is not provided in this scheme. Also, what the information provider should do after figuring out the traitors is not discussed in this paper. So how to make the scheme more robust in the case of the disputation between the information provider and users and how to combine the given traitor tracing scheme with revocation are our future work.

References

1. B.Chor, A.Fiat and M.Naor, "Tracing Traitors", Proceedings of Crypto'94, LNCS 839, Springer-Verlag, Berlin 1994.
2. B.Pfitzmann, "Trials of Traced Traitors", Workshop on Information Hiding, Cambridge, UK, LNCS 1174, Springer-Verlag, 1996.
3. M.Naor and B.Pinkas, "Threshold Traitor Tracing", Proceedings of Crypto'98, LNCS 1462, Springer-Verlag, Berlin 1998.

4. D.R.Stinson and R.Wei, "Key Preassigned Traceability for Broadcast encryption," Selected Area in Cryptology, SAC'98, LNCS 1556, 1999.
5. D.R.Stinson and R.Wei, "Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes", J. on Discrete Mathematics, Vol.11, No.1, 1998.
6. K.Kurusawa and Y.Desmedt, "Optimum Trator Tracing and Asymmetric Schemes", LNCS 1403,1998.
7. K.Kurusawa ,M.Burmester and Y.Desmedt, "A Proven Secure Tracing Algorithm for the Optimun KD Traitor Tracing Scheme", Proceedings of Eurocrypt'98, 1998.
8. A.Fiat and T.Tassa, "Dynamic Traitor Tracing," J. of Cryptology, Vol.4, 2001.
9. O.Berkaman, M.Parnas and J.Sgall, "Efficient Dynamic Traitor Tracing", J. on Computing, vol.30, No.6, 2001.
10. D.Boneh and M.Franklin, "An efficient Public Key Traitor Tracing Scheme", Proc. of Crypto'99, 1999.
11. A.Kiayias and M.Yung, "Self Protecting Pirates and Black-Box Traitot Tracing", Proc. of the 21st Annual International Cryptology Conference, Crypto'01, LNCS 2139, 2001.
12. D.Naor, M.Naor and J.Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers", Proc. of Advances in Cryptoloty, Santa Barbara, CA, Aug., 2001
13. M.Naor and B.Pinkas, "Efficient Trace and Revoke Schemes", Proc. of Financial Cryptography, Anguila, 2000.
14. E.Magkos, P.Kotzanikolaou and V.Chrissikopoulos, "An Asymmetric Traceability Scheme for Copyright Protection Without Trust Assumptions", LCNCS 2115, 2001.
15. S.Maki, "On Long-Lived Public-Key Traitor Tracing. First Steps", Proc.of the Helsinki University of Technology, Semenar on Network Security, Fall, 2000.
16. R.Safavi-Naini and Y.Wang, "Traitor Tracing for Shortened and Corrupted Fingerprints", ACM workshop on Digital Rights Management, 2002.
17. J.J.Yan and Y.Wu, "An Attack on A Traitor Tracing Scheme", Technical Report No.518, Computer Laboratory, Univ. of Cambridge, 2001.
18. A.Kiayias and M.Yung, "Breaking and Repairing Asymmetric Public-Key Traitor Tracing", ACM workshop on Digital Rights Management, DRM'2002, 2002.
19. A.Silverberg, J.Staddon and J.Walker, "Efficient Traitor Tracing Algorithms using List Decoding", Cryptology ePrint Archive: Report 2001/016, 2001.
20. D.Boneh, "The Decision Diffie-Hellman Problem", Proc. of the 3rd Algorithmic Number Theory Symposium, LNCS Vol.1423, Springer, 1998.
21. J.H.van Lint and G.van der Geer, Introduction to coding theory and Algebraic geometry, Birkhauser Verlag publisher, Boston, 1988.
22. V.Guruswami and M.Sudan, "Improved Decoding of Reed-Solomon and Algebraic-Geometric Codes," IEEE Transactions on Information theory, vol.45, 1999.
23. G.L. Feng and T.R.N. Rao, "Improved Geometric Goppa Codes Part I: Basic Theory,", IEEE Transactions on Information Theory, vol.41, No.6, 1995.
24. G.L. Feng, V.K. Wei, T.R.N. Rao, and K.K. Tzeng, "Simplified Understanding and Efficient Decoding of a Class of Algebraic-Geometric Codes". IEEE Transaction on Information Theory, vol.40, No.4, Jul. 1994.
25. G.L.Feng and T.R.N.Rao "A simple Approach for Construction of Algebraic-Geometric Codes from Affine Plane Curves", IEEE Transaction on Information Theory, vol.40, No.4, Jul. 1994.