Recent Developments in Tracing Traitors

UPI: cche211

Student ID: 3373590

Name: Chang Chen

Abstract:

In 2000 and 2001, Chor and Fiat etc. published two papers, [2] and [4], which can trace traitors who redistribute keys and cleartext respectively. These schemes, however, have some vulnerable points such as high real-time computation in [4]. In this five year, the two kinds of schemes have been published to improve performance, which are *sequential* and *RSA based*. This paper would focus on comparing later two kinds of scheme with Chor and Fiat's schemes, using the latest version of each sort, to figure out the recent improvements in tracing traitors, and discuss the further possible direction of development.

1. Introduction:

Nowadays, downloading via internet is developing in rapid speed along with the rapid increase of bandwidth of internet. How to eliminate illegal redistribution of digital objects such as images, videos and music through internet becomes a more and more important issue to protect ownership rights of intellectual property.

In 1994, traitor was introduced by Chor and Fiat etc. in [1] in the first time, and some tracing schemes for tracing users' keys also were provided. Then in [2], Chor and Fiat etc. provided the formal definition of Traitor: "The *traitor* or *traitors* is the (set of) authorized user(s) who allow other, non-authorized parties, to obtain the data. These non-authorized parties are called *pirate users*."

From this definition, either a user leaking key or cleartext could be call as traitor. And in [4] in 1999, Fiat and Tassa published schemes that could trace the tracing who leaks cleartext ([3] was previous version of [4]).

This paper concentrates on the current developments of schemes in this area by comparing [2], [4] with [5], [6]. And the vulnerability of current schemes will be discussed, and the further direction of development as well.

There is a problem about terminology. The all of current tracing schemes are called "tracing traitor scheme", and researchers would add some attribution word to show what algorithm they used or some other features of schemes. However, there is, no tracing schemes can trace both of key and cleartext. To not make the current terminology more confusing, this paper follows the current terminology and calls the whole set of schemes that includes key and cleartext as *Tracing Schemes*.

The rest of this report is organized as following. In section 2, we will present two basic scenarios of traitors and the principles of [2] and [4]. And compared with [2] and [4], major improvements in [5] and [6] will be presented in section 3. Section 4 will be discussion to analysis some problems existing in tracing schemes, from the author's opinion. The conclusion and reference will be in section 5 and section 6.

2. Related works

2.1 Two scenarios:

All of the current tracing schemes are basing on one of the two scenarios below:

Scenario 1:

The traitors redistribute the clear text to pirates directly, and then the pirates simply rebroadcast the content via network. The best example is that of pay TV systems where subscribers may access specific channels or programs by purchasing their viewing rights. In such systems, the content could be distributed by various ways like cable and terrestrial. This scenario base on a *conditional access* system which can guarantee that only paying subscribers can gain authorization to access the particular content for which they have paid. And there are others example for this scenario as well: conditional access systems are also used to protect pay services on the Web [4]. This piracy scenario may become especially attractive (to pirates) in the context of broadband multicast over the Internet [4]. The typical countermeasure for this scenario is that data supplier inserts some fingerprint (watermark) in every segment of broadcasting content, and then analyzes the fingerprints contained in the segments from pirate to find traitors.

Scenario 2:

But sometime, activities of scenario 1 may be much expensive such as online databases or online newspapers. In these cases, the online content is changed frequently and the unauthorized users only have interest in the latest content; if the pirates still use method in scenario 1, the cost would definitely rise to be unacceptable for them. Therefore, traitors could distribute the part or all decryption keys to pirates and the pirates construct a pirate decoder to decrypt ciphertext from data supplier, and then broadcast the content via network. The obviously method is data supplier assign different decrypt keys to each user and trace them.

2.2 Schemes by Chor and Fiat etc

In [2], Chor etc. presented six tracing traitor schemes based on scenario 2. The goal of the system designer is to assign keys to the users such that when a pirate decoder is captured it should be possible to detect at least one traitor, subject to the limitation that the number of traitors is, at most, *k* [2].

In these schemes, data supplier prepares a base set of keys, set *A*, and distributes a subset of them, to each user like Broadcast Encryption. And the messages that are sent to users consist of pairs of the form (*enabling block, cipher block*). The cipher block is the symmetric encryption of the actual data, encrypted by some secret random key s. The actual data could be the paid content provided by data supplier or some information that need to transfer to user client. The enabling block allows authorized users to obtain s. The enabling block consists of encrypted values under some or all of the keys of the base set *A*. Each authorized user will be able to compute by decrypting the values for which he has keys and then computing the actual key from these values [2].

If the data supplier captures a pirate decoder, then they can test the pirate decoder as black box, which means they can only concern the output by inputting different keys.

All of schemes presented in [2] are symmetric, which mean the content provider shares all secret information with the set of authorized users. The major disadvantage of symmetric scheme is that in a symmetric scheme, data supplier can construct a dummy pirate decoder that contains a particular user's decryption key, to frame an innocent authorized user.

In [4], Fiat and Tassa presented dynamic tracing traitor schemes based on scenario 1. The goal of the schemes is to disconnect all traitors in one time, who collude with one pirate. The content from data supplier is divided into consecutive segments, for example, one minute interval in an audio track. To create *q versions* of the segment, watermarking algorithm is used to embed one of the *q* marks that are in a preset mark set *W* in each segment. In each interval, the user group is divided into *q* subsets and each subset receives one version of the segment, and the subsets are varied in each interval. It is assumed that data supplier can regroup their users in each interval efficiently and the content will be delivered to users securely. In [4], Fiat and Tassa also proved that for tracing p traitors at least (p+1) versions must be used. And these schemes can prevent framing because they use the concept of frameproof codes and secure codes in [7].

These schemes have two major shortcomings from [5], “The first shortcoming is that regrouping of the users and mark allocation to users in each interval depends on the rebroadcasted content, also called *feedback* from the channel. This means that if there is no feedback from the channel no regrouping will occur and so the system is vulnerable to a *delayed rebroadcast attack*.” and “The second shortcoming of the system is high real-time computation for regrouping the users and allocating marks to subsets. This means that the length of a segment cannot be very short.”

3. Recent improvement

3.1 Improvement in Sequential Tracing Traitors Scheme

Sequential Tracing Traitors scheme consider the same scenario as dynamic tracing. In sequential tracing, the channel feedback is only used for tracing and not for allocation of marks to users [5]. Similar to the schemes [4], the system can trace all colluders. And the scheme use the mark allocation table that is predefined, which means there is no need for real-time computation to determine the mark allocation of the next interval. “Other computations related to key management of the group can be all performed as precomputation and so the need for real-time computation will be minimized” [5]. Therefore, the scheme can obviously overcome the shortcoming of high real-time computation in [4]. And actually, the performance of sequential tracing is much better than dynamic tracing.

In Sequential tracing, mark allocation in each interval is depending on the predefined table irrespective of the channel feedback. And the scheme identify traitors sequentially, which means they keep on identifying traitor one by one without waiting feedback until all of traitors identified. Thus, it can easily overcome the shortcoming of delayed rebroadcast attack in [4].

2.4 Improvement in Tracing Traitors Schemes Based on RSA

RSA is a public-key cryptosystem for both encryption and authentication, invented in 1977. In [6], the authors presented a new tracing traitors scheme for scenario 2 based upon RSA encryption algorithm, named Traitor Tracing using RSA (TTR). TTR apply RSA as a secret-key cryptosystem rather than as a public-key cryptosystem, and adopt the same message form as the schemes in [1]. TTR has both clear-box and black-box traitor tracing algorithms. The efficient clear-box algorithm can always identify at least one of the traitors in a collusion of size k or fewer, which they assumed that the pirate decoder contains easily recognizable representations of one or more valid decryption keys. "The efficient black-box algorithm can identify all of the contributing traitors in a collusion of size k or fewer, even when keys cannot be explicitly extracted from the pirate decoder, but only for a limited and special class of pirate decoders" [6]. Furthermore, TTR can prevent traitor collusions from framing innocent users.

Compared with schemes in [1], the major improvement in [6] is improving the performance of scheme, as shown in **table 1**. But TTR still a symmetric scheme, which means it still cannot prevent data supplier from fabricating evidence to frame honest users.

Traitor Tracing scheme	Communication Overhead	Decryption Complexity (Dominant Component) per User	Number of Decryption key per User
One-level in [1]	O($k^4 \log n$)	O($k^2 \log n$)	O($k^2 \log n$)
Based on RSA	O(max($k \ \log n$), $k \ \log\log M / \log k$))	~ 1	1

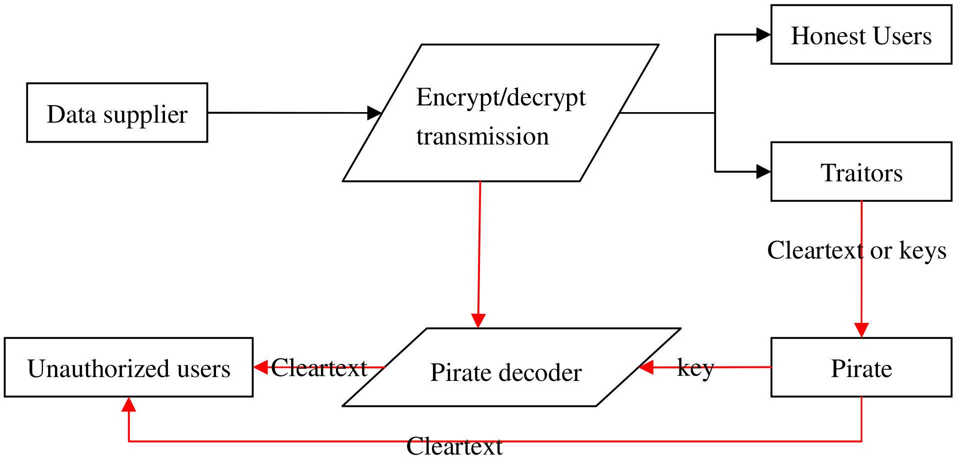
Table 1: Comparison between One-level scheme and TTR

4. Discussion:

It seems that the current research interests concentrate on how to improve the performance of tracing schemes based upon both scenarios. But there is no improvement to solve the problem of fabrication based on scenario 2, except for Pfitzmann. In [8], Pfitzmann first figured out the fabrication problem and gave an asymmetric scheme (Trials), but the overall complexity of the scheme is very high. In the rest of this section, author will discuss the whole system concerned with traitors and try to figure out some other vulnerable point in this scheme.

4.1 Four Ways for Pirates to Obtain Information

The **picture 1** is the sketch map of whole broadcast system that relate to traitors. All of tracing schemes try to analyze fingerprints (watermarks) or keys, to find who participate in redistribution activities, and then data supplier can delete traitors to stop unauthorized users obtaining valuable information.



Picture 1

Note that the illegal activities in Picture 1 are the lines in read in picture 1.

From the picture 1, there are at least four possible ways for a pirate to obtain cleartext or keys:

1. Spy transmission, and break encryption: if pirate have enough technical ability, they can do so, but the cost is also very expensive.

2. Hack some honest users' device: Pirate could spy or hack some innocent user's device to steal cleartext or keys which they want to have.

3. Conspire with traitors: Pirate can buy the keys or cleartext from some authorized users who want to sell.

4. Conspire with insiders within data supplier: Similar to point 3.

4.2 Necessary Components of Tracing Schemes

From the author point of view, tracing schemes on both scenarios should involve following basic components, to keep the honest users away from:

1. Initialization scheme: Data supplier can use to assign keys or watermark to each new user.
2. Encryption and Decryption scheme: Data suppliers encrypt messages and users decrypt those messages respectively.
3. Tracing traitor algorithms: Data suppliers can use to find source, if they find any illegal activities.
4. Strategy inside user device: This component should include where to store the decrypt keys or content and which form the keys or content store in. Users can protect their secret properly.

Some of components in the list should cover some strategy for preventing fabrication by data supplier, which is a function and not an independent component in the whole broadcast system. All of current tracing schemes have three components in the list above, but to author's knowledge, no scheme has the fourth component. Author insists that if the tracing schemes are wildly used one day, the fourth component is necessary.

In 4.1, we have presented the four ways for pirate to obtain data or key they want. The purpose of tracing schemes should be to find the dishonest users in the system; what is more, to prevent pirate from obtaining key or content easily via other three ways.

As for the first way, nearly all of tracing schemes assume that the encryption/decryption transmission is secure; and to the fourth way, all of them ignored insiders within the data supplier. This assumption is acceptable because encrypt and decrypt technology is relative mature and insiders of data supplier are extremely hard to be found and they can provide content without watermark, which means they could make themselves untraceable. And we have asymmetric scheme to prevent fabrication evidence by data suppliers or insiders, which can make harder to frame innocent users and keep harm away from legitimate users at least.

But in current tracing schemes, there is no strategy for prevent hacker from the user devices. May be all the researchers assume that the pirate cannot hack user device with basic protection provided by Operating Systems, or the researchers think that the users should take responsibility of keeping the keys and content secure by themselves, because the contract for asking users keeping secret will be signed when users apply services of data supplier.

While, if the tracing schemes have not the strategy, it seems the data supplier intentionally leave a vulnerable point for pirates and the traitors captured also can shirk their responsibility and shift the blame onto some hackers. Thus, the result of tracing is not convincible to determine a traitor; it only can find keys or watermark that should have some problems somewhere. And in such situation, innocent users must be involved in.

If we want to really keep piracy away from honest users and make the tracing result convincible, data suppliers still need to build strategy for user device, which should be transparent to users. After all, we cannot assume all of the users are professional for computer and they exactly know well about how to protect their secret on their computer.

5. Conclusion

Compared the works in 1999 and 2000, the recent improvements in tracing schemes have been presented, in this paper. After discussed the whole system, there are still some other vulnerable points in the whole system, which were not be figured out by previous researchers. To make up these vulnerable points, the researches should not be only focused on improving the performance of schemes, which has not actually hindered the implementation of tracing schemes.

6. Reference:

[1] B. Chor, A. Fiat, and M. Naor, "Tracing traitors", *Proc. Advances in Cryptology—Crypto '94*: Springr-Verlag, 1994, LNCS 839, pp. 257–270.

[2] B. Chor, A. Fiat, M. Naor, and B. Pinkas, "Tracing Traitors", *IEEE Transactions on Information Theory*, vol. 46, Issue 3, May 2000 Page(s):893 - 910.

[3] A. Fiat and T. Tassa, "Dynamic traitor tracing,", *Proc. Advances in Cryptology—Crypto '99*, LNCS 1666, 1999, pp. 388–397.

[4] A. Fiat and T. Tassa, "Dynamic traitor tracing," *J. Cryptol.*, vol. 14, no. 3, pp. 211–223, 2001.

[5] Safavi-Naini, R. and Yejing Wang; "Sequential traitor tracing", *IEEE Transactions on Information Theory*, Vol. 49, Issue 5, May 2003 Page(s):1319 - 1326

[6] John Patrick McGregor, Yiqun Lisa Yin, and Ruby B. Lee, "A Traitor Tracing Scheme Based on RSA for Fast Decryption", Springer-Verlag, LNCS 3531, 2005, pp. 56–74.

[7] D. Boneh and J. Shaw, Collusion-Secure Fingerprinting for Digital Data, *IEEE Transactions on Information Theory*, vol. 44, no. 5 (1998), pp. 1897–1905 (see also *Proc. Crypto* 95, LNCS 963, Springer-Verlag, Berlin, 1995, pp. 452–465).

[8] B. Pfitzmann, "Trials of traced traitors," in *Workshop on Information Hiding*, LNCS 1174, Cambridge, U.K., 1996, pp. 49–64.