TDDC03 Projects, Spring 2006

Traitor Tracing

v1.4 final

Gustav Nordvall and Oscar Nordvall

Supervisor: Tina Lindgren

Traitor Tracing

Gustav Nordvall Oscar Nordvall
Linköpings universitetet, Sweden
Email: {gusno334,oscno178}@student.liu.se

Abstract

Distribution of digital content consists of a major issue, copyright protection. Traitor tracing is a technique to prevent this illegal copying of distributed material such as software, broadcast of pay-tv and multimedia.

In this paper we will provide a general overview of traitor tracing, its techniques and methods. Furthermore, the methods are discussed briefly with the intention to get a clear picture of what models exists today and their weaknesses we have identified.

To extend the overview, we will discuss what legal problems exist and what has been done in the field of research.

1. Introduction

Traitor tracing is about preventing people from creating illegal copies. The main task is to make every copy unique so that we are able to trace traitors. We do not focus on tracing *pirates*. A pirate is referring to a person that has a copy from a *traitor* and shares this copy with others illegally. In addition, a *traitor* is an authorized person giving their legal copy to an unauthorized person, namely pirates, which spreads their copies illegally.

To make unique copies we need identification implemented in the digital media by marking the copies. This is done by techniques such as *fingerprinting*. Fingerprinting hides identification or license information while *watermarking* hides information such as copyright or ownership. In other words fingerprinting identifies users and watermarking identifies the owner. [5] In traitor tracing, the fingerprinting is done through different keys. The keys are personalized and do not always exist in the copies as we will see in the different models. To trace traitors we examine the media content together with the use of personal keys to bind the media content to a traitor.

Today traitor tracing is important due to the file sharing community that has grown because of Internet and the simplyness of copying digital media.

The interest in this area has brought us several *models* and *schemes* for solving this problem, with the use of traitor tracing. These models and schemes include algorithms that are able to trace the traitors. If a traitor is found, it is possible for the distributors of the digital media to make accusations.

To get a better understanding of where traitor tracing is used, figure 1 from [2] shows all the way from the source object to accusation of the pirates.

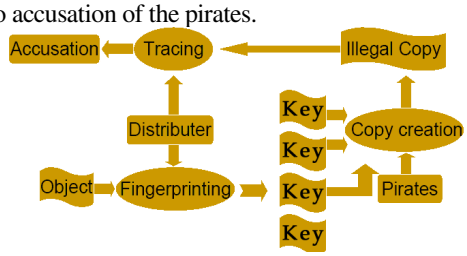


Figure 1. Tracing pirates

When traitor tracing is used in practice there are legal problems that will be discussed later in the report due to the fact that legal problems can make traitor tracing difficult to perform using the provided methods.

We will also discuss the research that has been done in this area.

2. Models and schemes

This chapter contains some brief descriptions of available models and schemes that are used in traitor tracing.

Every scheme has different properties and they use different techniques but they have three main components in common.

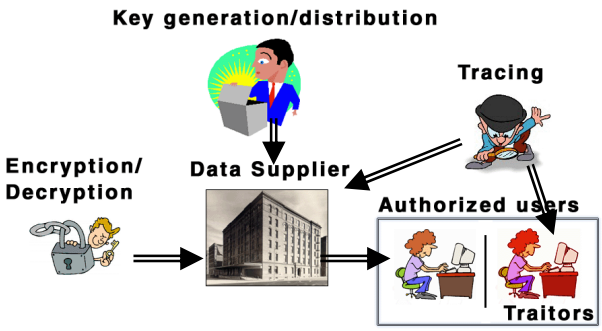


Figure 2. Scheme components

In figure 2 we can see the connection between the scheme components. The first one is *key generation-distribution* which is used by the data supplier (also referred to provider) to generate and distribute unique keys for each user. The second one is *encryption/decryption* where the data supplier use an encryption scheme to encrypt the session key (described in section 2.1) and all authorized users uses a decryption scheme to decrypt their session key. The third component is a *tracing algorithm* which is used to identify one or more traitors with the help of the personal keys or the personalized content. [1]

Every model and scheme have some tracing goals, these goals are from [1]:

- Tracing the source of the piracy.
- Not harm legitimate users.
- Any unauthorized user should be disconnected.
- Provide legal evidence of pirate's identity.
- Deterring potential traitors.

The following models are discussed below:

- Static
- Dynamic
- Threshold
- Sequential
- Asymmetric and symmetric

There exist more schemes but they often are combinations of the ones that we present.

2.1 How the schemes are built

Before we go any deeper into the different schemes we need to understand how they are constructed.

The provider starts by generating a number of sets containing several keys. The user gets one key from each set. These keys make the *codeword* or the users personal key. [1, 3]

Then the provider sends the information in encrypted segments. The segments are divided into enabling and cipher blocks. For the users to view the content they need to decrypt the enabling blocks with their personal key to get a part of the session key that they use to decrypt one of the cipher blocks. This behavior repeats through the whole session, and then the session keys are no longer valid.

As an example of traitor tracing, we can examine the personal keys from a decoder and determine if it is a traitor or not. [3]

The following figure 3, shows how traitor tracing works in practice.

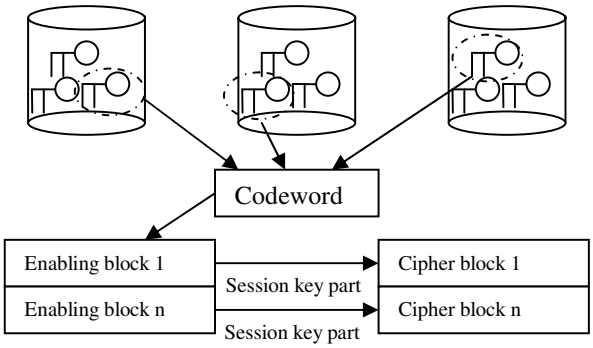


Figure 3. Traitor tracing in practice

For example, in a broadcast of a movie, the provider has sent a number of keys to the user. These keys are used to create a codeword. The user will not be able to see the movie without using the codeword in his decoder. The movie is splitted into segments. These segments are sent in different variants by having different enabling and cipher blocks. The enabling blocks does contain the session key, which is used to decrypt the ciphers blocks, that contains the movie content. Since the content is not marked we can not identify a traitor by examine the content. This problem is solved in a dynamic traitor tracing, which is discussed in section 2.2, by watermarking the content.

2.2 Static

One of the most appropriate scheme for DVD:s, CD:s or other electronic data distribution systems is the static traitor tracing scheme. The difference between this scheme and a dynamic scheme is that the codewords are static. Here, each copy is marked only one time. The basic idea is that we add enough marks with enough variations so that the pirates need c distinct copies in order to detect the entire fingerprint. [1, 19]

m_1	=	0	1	1	1	0	1
m_2	=	0	1	1	0	1	1
m_3	=	0	0	1	1	1	1

Figure 4. Marked content

Figure 4 from [19] shows that in this case the content is marked in three positions. The binary numbers represent the content and as we can see, the content includes a fingerprint in the gray positions. We have in this case three different fingerprints attached to the content for each user {1,1,0}, {1,0,1} and {0,1,1}. If two users that have content m_1 and m_2 collude they can create {1,X,X}. Since the first bit is not changed, it means that

the content must be from one of m_1 or m_2, but we do not know which one. [19]

2.3 Dynamic

In a dynamic scheme the users get some kind of interaction with a *center* so that it is able to change the key in real-time. [1] The content is dynamically changed to different set of users and makes it possible to trace traitors even if he is rebroadcast the content itself and not only the keys, solving the problem we mentioned in section 2.1. [3] This makes it useful in TV broadcast. The center is where the sources and the *watermarked content* are stored. In this model the content are splitted into segments which the center sends. A watermarked content is a content that consist of different variants of each segment.

Watermarked content

Segment 1	Segment 1 variant 1	…	Segment 1 variant *r*
Segment 2	Segment 2 variant 1	…	Segment 2 variant *r*
…	…	…	…
Segment *m*	Segment *m* variant 1	…	Segment *m* variant *r*

Figure 5. Watermarked content

In figure 5 there are *r* different variants of each segment and the content is splitted into *m* segments. The segment could contain for example one minute video.

For each user a specific variant of each segment is chosen, this is the codeword. For example, a user can get variant "segment1variant2" and then "segment2variant3". The codeword is chosen along the way since it is a dynamic scheme.

The tracing algorithm that is running in the center could simply disconnect a user if the same sequence of variants of the segment is found in another user. In this way it is possible for a dynamic scheme to trace all traitors and it is more powerful than the static one because it can prevent further content in a broadcasting channel. [5]

The static method is *unreliable* due to the fact that we can not know how many active traitors there are because we do not get any feedback from the pirate network. By feedback we mean that users are connected online to the center. To trace all traitors in the static scheme we need to know a priori bound on the number of traitors because we do not get any information on the fly (like the dynamic method). [5]

In both static and dynamic, the size of the alphabet used to generate keys should be at least the number of traitors plus one to trace the traitors and not accusing innocent in other words, *deterministic*. [5] In the dynamic case this may require a large bandwidth and the result of that is that the center may not handle it. [7]

2.3.1 Threshold

The threshold scheme is a dynamic scheme, since the codewords are chosen along the way, as in the dynamic scheme.

All the schemes we have introduced are *fully resilient* schemes that are able to trace any *pirate decoder* that decrypts successfully with a probability that is non-negligible. Since the decoders (used in for example pay tv systems) that decrypts only part of the content is considered useless, threshold schemes were introduced. These schemes only trace the source of the keys of the decoders where the probability of successful decryption is greater than some threshold q. [4] On the other hand, there is no guarantee that they can trace decoders where this probability is smaller than q. [6]

To use threshold traitor tracing the communication needs to be divided into blocks which are encrypted independently. To make a legitimate decoder work, it needs to contain all keys to decrypt every block. In the case of a pirate decoder, we notice that if the decoder contains enough keys to decrypt more than a q fraction of the blocks, we can, with the use of threshold traitor tracing, trace at least one of the traitors. [6]

If a pirate decoder is not able to decrypt more than a q fraction of the blocks, it is not very useful and therefore not important to trace. [6]

The threshold scheme is built as we have seen in section 2.1. We model all of the keys as a matrix. A user then picks a key from each of the rows to create the codeword. In the matrix we have a number of rows, that are marked.

We are only concentrated on those decoders that use keys from the marked rows, as the decoder that uses the keys from these rows, have the probability q of a successful decoding. With other words, decoders that do not use keys from the marked rows is considered useless.

The tracing is done in the same way as dynamic schemes but we only concentrate in the use of the keys from the marked rows. That makes us able to reduce the enabling blocks and therefore get a dramatic reduction in the *data redundancy overhead* (the increased size of the data to allow traitor tracing [4]) compared to the fully resilient schemes.

2.3.2 Sequential

The sequential scheme is placed between static and dynamic schemes. This is because the codewords are

predefined, however the attacks are the same as dynamic schemes since it is used in a distributed pay tv system.

As with the dynamic scheme, it has a drawback against *delayed rebroadcast attack*. A delayed rebroadcast attack is where attackers rebroadcast the content with some delay. Sequential traitor tracing has solved this problem by using a *mark allocation table*. The mark allocation table contains individual marks for each user and is predefined which means it does not require any real-time computation. [8] The marks used in the table provides as in dynamic tracing a way to trace the source of the rebroadcast by examining these marks in the rebroadcast. But in opposite to dynamic tracing, the allocation table is predefined which makes it possible to compute the allocation table before the transmission starts. [11] This is the main difference between the traditional dynamic traitor tracing scheme as that scheme require a lot of computation to keep the allocation table updated in every interval with new marks.

For an example of how marks are used to trace the traitors in sequential tracing we can see in figure 7 the mark allocation table. This table contains a row for each user with the columns as blocks containing marks.

$$M = \begin{pmatrix} M_0 & \phi_{11}(M_0) & \phi_{12}(M_0) & \cdots & \phi_{1m}(M_0) \\ M_0 & \phi_{21}(M_0) & \phi_{22}(M_0) & \cdots & \phi_{2m}(M_0) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M_0 & \phi_{b1}(M_0) & \phi_{b2}(M_0) & \cdots & \phi_{bm}(M_0) \end{pmatrix}$$

Figure 7. Mark allocation table

To be able to trace the traitors we need a feedback sequence $F = (f_1, f_2, ..., f_j)$ containing the marks from the users content (that we get in return from the decoder), where j is the numbers of the columns in the mark allocation table. To be able to identify traitors, we compare this feedback sequence with the rows in the mark allocation table. If a user is identified as a traitor he or she will be disconnected.

Also with this scheme, if a traitor is found it will be disconnected and the algorithm will continue to find the next traitor. Because of this sequential identification of traitors, this scheme is called sequential traitor tracing. [8]

2.4 Asymmetric and symmetric

If the information provider and the user share the same keys for encrypting and decrypting of sessions it is a symmetric traitor tracing scheme. [1] In this scheme both the user and the information provider share the same secrets. So that means that we have to trust the information provider because we do not know if the redistributed information comes from the provider or a traitor. [12] In this case we cannot provide a solid proof against the implication of any traitor and this is even if the scheme is *fully frameproof*. [10, 12] Fully frameproof means that it is impossible for colluding traitors to frame another user. [12]

Furthermore the symmetric schemes are either open or secret depending on whether the generation/distribution scheme is made public or not. [1] The problems with secret schemes are that the codewords must remain secret from the users and the traitors might choose a codeword for a key that belong to an honest user. [12]

In an asymmetric traitor tracing scheme the problem with thrusting the information provider is solved. We can't blame the system-manager because he doesn't know all the key information, just a portion of the user key. He needs the users to participate in order to trace a traitor. This is usually referred to as *non-repudiation*. [9]

In a public-key asymmetric traitor tracing scheme there is three components: system manager (an authority responsible for broadcasting infrastructure), channel-provider (who distribute the data to the users) and a judge. The judge is an arbitrary third party that the information provider tries to convince of a traced traitor. [9, 12] See figure 6 for better understanding.

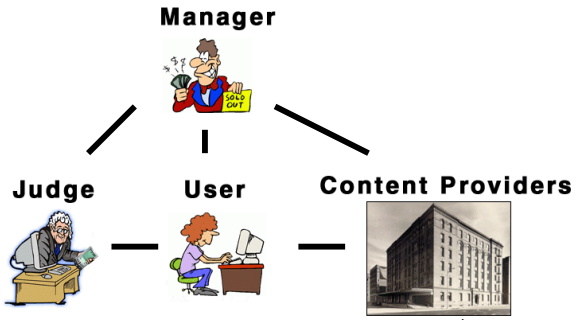


Figure 6. Public key asymmetric traitor tracing

In this scheme the content is encrypted using a public-encryption procedure which the system-manager combines with the user keys. [10]

3. Attacks

The mentioned schemes above are all subject to the collusion attack which is the most common attack. This type of attack is a coalition of traitors where they create pirate decoders that hide their identity. The traitors

observe their sequence of codewords, and combine them to create new fake keys. [20]

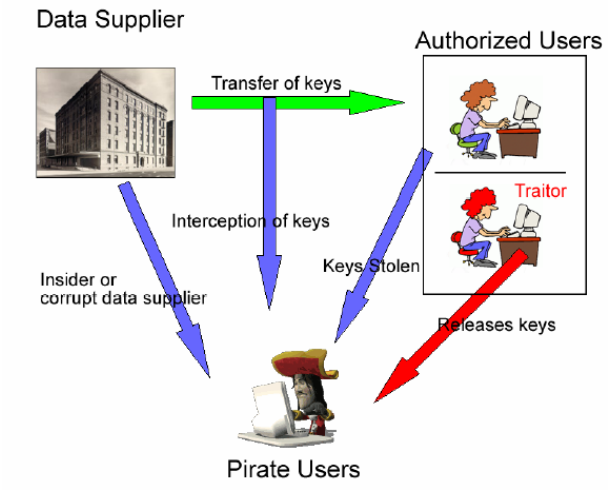
In schemes such as sequential traitor tracing these new fake codewords are discovered by the feedback channel discussed in section 2.4. If we consider the static schemes this is difficult in the sense that these schemes don't get any feedback. The dynamic schemes do get the feedback but in return requires a lot of bandwidth.

In the asymmetric and symmetric case, sharing of codewords is about trust. In the symmetric case we need to trust the information provider (that they are not the traitors). In the asymmetric case we can be more sure about who is being the traitors since the information providers is sharing the keys with more involved parties.

Another attack we have identified is the delayed rebroadcast attack that is discussed in section 2.4. This attack can also be a collusion attack if more attackers are involved.

More attacks are identified in figure 8:

- **Interception of keys by pirate users.** Probably a man-in-the-middle attack where a user is claiming to be an authorized user.
- **Keys stolen from authorized users.** User keys stored on a CD or a computer could be stolen.
- **An insider from the data supplier.** Someone from the inside could leak information. Typically pirated movies.
- **A corrupt data supplier.** [18]



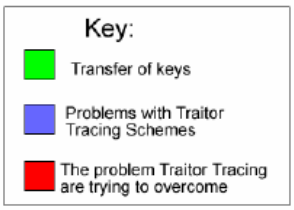


Figure 8. Problems with traitor tracing

4. Legal problems

As we have seen, in the symmetric traitor tracing schemes they are not able to convince a third party, like a judge, of the implication of any traitor. This problem is tried to be taken care of with an asymmetric scheme by trying to convince an arbitrary third party, a judge, by providing some input that is reliable linked to the accused user. A problem here is that this input might not be reliable and therefore could not detect accidental sharing. [18]

More problems with traitor tracing schemes are that it generates *false positives*. This is because it might not be the owner of the keys, found with a traitor tracing algorithm, that are the guilty party. To make false accusation would damage both the data supplier and the traitor and will cost a lot of money for both parties. [18] It is even worse than an undetected redistribution. [12]

There are also problems with mapping a key to a user. If we assume it is only one user that knows a secret to a safe and that secret comes out in public it is very likely to accuse this user. But if there are several users it is harder to know which one. [18]

5. Research

The research in traitor tracing is divided into several areas such as fingerprinting, protecting against collusion, improved tracing algorithms, linear and bilinear traitor tracing and broadcasting of messages. [13]

According to [14] the first traitor tracing scheme was introduced by Chor, Fiat and Naor in 1994. The development of new schemes has continued by the work of many researchers [14]. As an example, Boney and Shaw has suggested a solution for collusion secure fingerprinting for digital data. [15]

Model/technique	Year
Traitor tracing introduced	1994
Asymmetric/Symmetric traitor tracing	1997
Threshold traitor tracing	1998
Dynamic traitor tracing	1999
Sequential traitor tracing	2003

Table 1. History of models and techniques

If we consider [15] and table 1 we can see the history of the schemes arrivals. The research in traitor tracing started in 1994 and the first attempt to new schemes, called asymmetric and symmetric and was developed 1997 by B. Pfitzmann. M. Naor and B. Pinkas.

They introduced in 1998 the beginning of dynamic schemes, the threshold tracing scheme and a year after A. Fiat and T. Tassa presented dynamic traitor tracing. The last scheme we have concentrated on, sequential traitor tracing is only a couple of years old, 2003 and presented by R. Safavi-Naini and Y. Wang.

1994-2003 we can in [15] notice that the algorithms that the schemes use are improved and solutions to various attacks has been proposed.

Today as said by [16] that due to the development of internet and digital media the protection is becoming more and more important. Big companies like the Hollywood film industry seek for traitor tracing techniques and we believe the demand increases the motivation for further research. Another reason why companies are investigating in this area is because of the serious collusion attack threat where traitors cooperate [17].

6. Conclusions

In this article traitor tracing is about finding the source of the piracy, namely the traitors. We had to introduce various terms such as pirates and fingerprinting to understand what the traitor tracing is about.

We have found several models and techniques that tries to solve the problem of identifying the source of the piracy. All of the models have different behaviors and purposes but they all have the same goals.

The problems we have identified in the models are the use of bandwidth, memory and real-time computation which increases the demand of capacity to be able to use these models in practice. For example in a dynamic scheme where the content are splitted into segments with different variants it is almost impossible for a sender to send out all of these variants because of the limitations in bandwidth. The increase in bandwidth is also due to the enabling blocks that are added to each segment. Another example would be that the number of keys is limited due to memory, as we have seen the schemes are using a lot of keys to generate different personal keys for each user.

And there are in conjunction to these problems, legal aspects as well. This is due to the fact that is difficult to gather evidence that holds in court. As for example, in the symmetric schemes it is impossible to convince any third party because we do not know if the redistributed content comes from the provider or pirates.

Traitor tracing is a subject that contains many areas and as we can see the history of traitor tracing is rather new. It seems to be in development but never used in practice yet. However nowadays, big companies are cooperating with researchers to get traitor tracing that can be used in practice due to the fact that they loose very much capital when traitors and pirates are involved. But on the other hand, to accuse an innocent is much more expensive than an undetected redistribution.

References

[1] J. Trevathan, H. Ghodasi,"Overview of Traitor Tracing Schemes", CiteSeer, 2003.

[2] J. Löfvenberg, "An overview over technical copyright protection", Linköpings universitet, Copyright protection lecture in TDDC03, 2006.

[3] A. Dembczynska, P. Kaniewski, "Copyright Protection with Traitor Tracing schemes and supplying undeniable proof of the traitor's treachery by means of Asymmetric Traitor Tracing", Linköpings universitet, 2003

[4] B. Chor, A. Fiat, M. Naor, B. Pinkas, "Tracing Traitors", IEEE Trans. Inform. Theory, vol. 46, pp 893-909, May 2000

[5] A. Fiat, T. Tamir, "Dynamic Traitor Tracing", J. Cryptology, 2001.

[6] M. Noar, B. Pinkas, "Threshold traitor tracing", "Lecture Notes in Computer Science", vol. 1462, pp 502--??, 1998

[7] T. Tamir, "Dynamic Traitor Tracing", Lecture 8, 2003

[8] R. Safavi-Naini, Y. Wang, "Sequential Traitor Tracing" IEEE Trans. Inform. Theory, vol. 49, pp 1319-1326, May 2003

[9] B. Pfitzmann, M. Waidner "Asymmetric Fingerprinting for Larger Collusions", Conference on Computer and Communications Security Proceedings of the 4th ACM conference on Computer and communications security 1997

[10] A. Kiayias, M. Yung "Breaking and Repairing Asymmetric Public-Key Traitor Tracing", "Lecture Notes in Computer Science", Volume 2696, pp. 32-50, 2003

[11] R. Safavi-Naini and Y. Wang, "Sequential Traitor Tracing", "Advances in Cryptology - CRYPTO 2000 (Lecture notes in computer science)", Berlin, Germany: Springer-verlag, vol. 1880, pp 316-332, 2000

[12] Birgit Pfitzmann, "Trials of Traced Traitors", "Information Hiding Workshop", Spring LNCS 1174, pp. 49-63, 1996

[13] A. Hindle, "Exposing Traitors: Traitor Tracing, Watermarks and DRM", "CSC482A/582B University Of Victoria", 2003

[14] J. Chen, "A Survey on Traitor Tracing Schemes", "University of Waterloo", 2000

[15] H. Lipmaa, "Traitor Tracing and Broadcast Encryption",1997-2005

<http://www.cs.ut.ee/~lipmaa/crypto/link/protocols/tracing.php>
[16] M. Wu "Digital Fingerprinting for Multimedia Security and Forensics", 2005, <http://www.enee.umd.edu/~minwu/research/ACC_fingerprint.html>
[17] "Tokyo Institute of Technology | NEWS", 2005, < http://www.titech.ac.jp/news/e/news050610-2.html>
[18] Jesse Wu, "Innocence of Traced Traitors", "Department of Computer Science", "University of Auckland", 2005
[19] Sophie Engle, "Fingerprinting and the Marking Assumption", 2005, <http://www.node99.org/hosted/marking/>
[20] M. Fernandez, M. Soriano, "Identification Algorithms for Sequential Traitor", INDOCRYPT, LNCS 3348, pp. 414–429, 2004