EFFICIENT TRAITOR TRACING FOR CONTENT PROTECTION

Hongxia Jin
IBM Almaden Research Center
San Jose, CA, U.S.A.

Keywords: Traitor tracing, Content protection, Anti-piracy, Broadcast encryption.

Abstract: In this paper we study the traitor tracing problem in the context of content protection. Traitor tracing is a forensic technology that attempts to detect the users who have involved in the pirate attacks when pirate evidences are recovered. There are different types of pirate attacks and each requires a different traitor tracing mechanism. We studied different types of attacks, surveyed various traitor tracing schemes and analyzed spectrum of traceabilities of different schemes using two representative schemes. We shall present some observations on the designs and their impact on the efficiency of the schemes. We shall also present various future directions that can lead to simpler and more efficient traitor tracing schemes for various pirate attacks.

1 INTRODUCTION

The goal of a content protection system for copyrighted materials is to make sure the materials are only accessible to user who are authorized to access. Of course pirated attackers want to circumvent all the protections and get access to the content illegally. Broadcast encryption and traitor tracing are two primary technologies used in a content protection system. In a broadcast encryption (Fiat and Naor, 1993) based content protection system, each user (also called decoders, devices) is assigned a unique set of secret keys (called device keys). Oftentimes the content is encrypted by a randomly chosen session key (sometimes termed as "media key") once and only once. When a broadcast encryption scheme is used for content protection, it uses a revocation structure (termed as "Media Key Block") that contains the session key encrypted by and only by privileged users' secret device keys. MKB is distributed together with the encrypted content. During playback, all privileged users can use their secret device keys to decrypt the structure although differently but can obtain the same valid session key to access the content; while all other excluding users can only decrypt the structure to garbage strings. There can exist different types of pirate attacks in this broadcast encryption based content protection system.

1. Pirates disclose secret device keys by building a clone pirate decoder.
2. Pirates disclose content encrypting key.
3. Pirates disclose (redistribute) decrypted content.

When the pirate evidences are found, traitor tracing is a forensic technology that can defend against piracy. The source devices (users) that involved in piracy are called traitors.

In the first pirate attack, attackers compromise several legitimate devices (decoders) and extract the secret device keys from the compromised devices to build a clone device which can be used to decrypt and access the encrypted content. In second and third type of attack, the attackers decrypt the MKB to get the valid media key. They can choose to re-broadcast the decrypted content or the media key. Because the decrypted content and media key are same for every user, this type of attack help attackers hide their identities, thus sometimes called anonymous attack.

In current state-of-art and state-of-practice, different types of traitor tracing schemes have been designed for different types of pirate attacks and different principles are used in the design in order to achieve efficiency. We believe a more systematic studying on those existing schemes are needed in order to understand the pros and cons of each type of design principles. This type of study can help shed insights on new design principles that can help design simpler, more practical and efficient traitor tracing schemes in future. This is what this paper is about.

In rest of the paper, in Section 2 we will give more background details on existing traitor tracing schemes. We believe designing traitor tracing schemes follow some general steps. In Section 3 we will use those general design steps to categorize ex-

Jin H. (2009).
EFFICIENT TRAITOR TRACING FOR CONTENT PROTECTION.
In *Proceedings of the International Conference on Security and Cryptography*, pages 270-273
DOI: 10.5220/0002235002700273

isting tracing schemes and analyze its pros and cons of those schemes. Based on our analysis, in Section 4 we will present new research directions that can improve the efficiency, practicality and simplicity of the traitor tracing scheme design.

2 TRACING SCHEMES FOR PIRATE DEVICE ATTACK AND ANONYMOUS ATTACK

There exist many broadcast encryption and traitor tracing schemes (Naor et al., 2001; Fiat and Naor, 1993; Boneh et al., 2006) targeting on the "pirate decoder attack". For pirate device attack, forensic testing materials, including forensic MKBs, are fed into the clone. When constructing a forensic MKB at frontier $\mathcal{F}$, one intentionally *enables* certain keys by using them to encrypt a valid media key and *disables* certain keys (not necessary traitorous) by encrypting a random bit string instead of the media key. Based on the keys inside the clone, the clone may or may not decrypt/play the content. Observing a series of response from the clone, the tracing procedure can identify a compromised key in current frontier. The tracing algorithm starts from an initial frontier $\mathcal{F}$ and proceeds by repeatedly using the subset tracing procedure to identify a compromised key $k \in \mathcal{F}$, removing it, and adding to $\mathcal{F}$ k_1 and k_2 such that we can replace k with k_1 and k_2 and still cover the same set of devices. This process is reiterated until the detected compromised key is at the lowest level or the clone box is unable to play the MKB associated with $\mathcal{F}$.

The traceability is defined to be the number of testings needed to detect traitors. The state-of-art and practice is the tree-based NNL scheme in (Naor et al., 2001). Each node in the tree is associated with a key. Each device is associated with a leaf node of the tree. Each device is assigned a set of keys based on the path from the leaf to the tree root. The NNL tracing takes $O(T^3 logT)$ number of tests to detect traitors in a coalition of size T.

Schemes in (H. Jin and Nusser, 2004; J. N. Staddon and Wei, 2001) are designed to defend against anonymous attack. In these schemes, content is differently watermarked and encrypted for different users. Readers refer to (H. Jin and Nusser, 2004) for efficiently prepare different versions. What is relevant in this paper is that every device has only one key to decrypt one version for each content.

The current state-of-art and practice traitor tracing scheme for anonymous attack is the JL scheme shown in (H. Jin, 2007) and deployed in AACS (AACS, 2006). In this scheme, each device is assigned a set of tracing keys from a large matrix. The columns correspond to the movie content in the sequence; the rows correspond to different versions for each movie. For example, the matrix might be 255 by 256. In a sequence of 255 movies, each movie has 256 movie versions. Each device is assigned exactly one key from each column, 255 in totals. Each key is one of the 256 versions. The assignment is based on an error correcting code, making any two devices as far apart as possible to enhance its collusion resistance.

In the process of forensic analysis and traitor detection, JL scheme employs a very different philosophy. All other schemes focused on detecting one traitor each time they incriminate the user who can explain most of the recovered forensic evidences. On contrast, JL traitor detection algorithm focused on finding the entire coalition that can explain ALL recovered forensic evidences. With this detection philosophy, JL scheme can detect traitors with much fewer number of forensic evidences. In fact they achieved a super-linear traceability. For details readers are referred to (Jin et al., 2008).

JL scheme also allows revocation of a set of compromised tracing keys and supports multi-time tracing when new attacks arise. Without needing to update the tracing keys that are burned into devices during manufacture time, the JL scheme employs a TKB (Tracing Key Block) mechanism to revoke tracing keys. It is similar to MKB but the JL scheme has more than one correct K, (called variant data in AACS), one for each version of the content. However, as shown in (H. Jin, 2007), the traceability degrades with revocations with TKB. That puts a limit on the revocation capability of the scheme. Indeed it has a finite revocation capability and traceability degrades with revocations.

3 DESIGNING A TRAITOR TRACING SCHEME

Traditionally a traitor tracing scheme consists of two basic steps. First, assign different keys/content versions to devices. Second, based on the recovered pirated content/keys, trace to the traitors. As we have seen, the spectrum of the traceability ranges from $O(T)$ to $O(T^3)$.

While NNL traceability does not change after revocation, JL scheme indicates the traceability is possible to degrade. In other words in the lifetime of a traitor tracing system, one must also consider the continuous traceability after revocations. In light of that, the design of a complete traitor tracing system

should consist of the following three steps instead of two steps.

1. *Assignment step:* Assign versions of the content/key to currently known innocent devices
2. *Forensic Analysis step:* Based on the recovered forensic evidences, trace to the traitors
3. *Revocation step:* loop to step 1 but exclude the currently discovered traitors.

The newly added step 3 brings in different requirements on the design. For the assignment task, now one must consider new assignment after revocations. For the second task, now one must consider traceability after revocation, and the overall traceability over the lifetime of the tracing system. We have studied carefully the NNL tracing and JL tracing scheme and compare their differences on the design principles when carrying out the above two tasks. Our studies reveal fruitful insights on how to design a more efficient traitor tracing system in the future.

3.1 Assignment: Tree vs. Matrix

Even though they were designed for two different attacks, we believe there are other underlying reasons why the tree-based NNL scheme has traceability of $O(T^3)$ while the matrix-based JL scheme achieves superlinear $O(T)$ traceability.

As one can imagine, in a tree-based system, any two devices may share many keys. For example, any two neighboring devices share all the keys except their leaf keys. During traitor detection process, it takes many forensic testings in order to distinguish two neighboring subsets (or devices). On contrast, in a matrix-based system like JL scheme, any two devices may share much fewer keys. For example, suppose a Reed-Solomon code $< n,k,d >$ is used to assign the keys to devices, any two devices have at least d different keys where d is the Hamming distance of the Reed-Solomon code and d is made as big as possible. In this type of design, any two devices are assigned maximally apart. This contributes to the superior traceability achieved in JL scheme. As a design principle, it seems an efficient tracing scheme needs to assign the keys to devices in a way that makes any two devices share as few key as possible.

3.2 Detection: Dynamic vs. Static

From traitor detection process point of view, it is easy to see that NNL tracing process is dynamic in nature while JL scheme is static in nature. In NNL tracing, when it identifies the traitorous subset at the current level, tracing moves down to the next lower level. New forensic MKBs will be constructed based on the new partition at the new level. This process is repeated until it reaches the leaf level and an actual traitor can be identified. As one can see, the tracing reacts to the previous testing results.

On contrast, the matrix-based JL scheme is static. In the matrix, each column corresponds to a movie content. Different columns clump different devices together. The tracing agency recovers a sequence of pirated evidences from different columns, each providing to license agency some forensic information. It is not required to react to the previous forensic results. As a result, MKBs can be produced way ahead of time. All those MKBs are guaranteed to provide forensic information. This provides some advantage for operation in real world.

As to traitor detection at each step, NNL tracing attempts to find one suspect subset and further split into two smaller subsets. On contrast, JL scheme employs a detection algorithm which tries to detect a coalition of suspects all together. As shown in (Jin et al., 2008), it is a much more efficient detection approach than detecting traitors one by one.

3.3 Continuous Traceability and Revocation Capacity

In a matrix-based tracing system, when there are revocations, a licensing agency producing a multi-column key block must spread the variant keys across all the columns. For example, in a 256 X 255 matrix, suppose the licensing agency has 256 variant keys (q = 256) that has to spread across 4 columns. So it would encrypt only 64 unique movie variant keys in the 255 un-compromised key cells in the first column. In other words, more than one cell (4 in this example) would encrypt the same variant key. In effect, this reduces the original q; the effective q is q/c, where c is the number of columns. In this example, the effective q is reduced from 256 to 64. As shown in (H. Jin and Nusser, 2004), in reality the extra bandwidth restrict the number of variants. Here that number is reduced even more by revocation. And our example has been the minimal case; the situation gets much worse as revocation continues over the life of the system and the number of columns in the key blocks gets larger and larger.

However, if the licensing agency can react to results from previously recovered movies, some of the inefficiencies of multi-column key blocks can be removed. For example, suppose the tracing agency has recovered a pirated key (or content version) corresponding to one media key variant and has deduced that attackers have at least one tracing key in a four-

key-cell clump that were assigned that variant. Then, in a subsequent key block, those four keys could each encrypt a unique media key variant. Of course, this would mean other clumps would have to become larger in that key block, but the attackers have a limited number of keys and eventually they will have to identify an individual key. So, if the licensing agency can react, the q/c problem, while not completely eliminated, can be greatly reduced.

As shown in (H. Jin, 2007), in order to relieve the traceability degradation problem, even the static matrix-based system would require at least two phases to be effective under a reasonable amount of revocation. On the other hand, for a same size MKB, a tree-based system can use many more subsets in the MKB. In other words, it can go down to a much deeper level of the tree to speed up the tracing. With the above observations, we realize, given revocation which is a fact of life in the real world, it is no longer clear that matrix-based systems provide better lifelong traceabilities.

4 FUTURE RESEARCH DIRECTIONS AND CONCLUSIONS

We know revocation-efficient tree-based NNL scheme is not efficient on tracing while tracing efficient matrix-based JL scheme is not efficient on revocation. As shown above, faster tracing requires any two devices be maximally apart, i.e., sharing minimal number of keys. In this way it is easier to distinguish devices. On contrast, efficient revocation (i.e., small MKB) requires any key to be shared by many devices, so that one encryption in MKB enables many devices. However, our analysis above also reveals when taking into consideration of revocation over lifetime of a trace-revoke system, the matrix-based system traceability is not necessary better than the tree-based system over its lifetime.

Our analysis and observations have led us to believe that a future simpler and more efficient trace-revoke system design will need to combine the advantage of the tree-based and matrix-based systems. For example, adding some dynamics into the static tracing in a matrix-based system can greatly reduce the q/c problem, and thus improve revocation capability and alleviate the traceability degradation problem. On the other hand, we also believe it is possible to add some statics into the mostly dynamic tree-based system to improve its traceability. For example, when the tracing reaches to any level of the tree, multiple MKBs at the same level in the tree can be produced ahead of time and each could provide forensic information. It seems in either tree-based or matrix-based system, a semi-static-dynamic tracing can achieve better traceability but still balance off the degradation problem.

To further improve tree-based system NNL tracing, at any level identifying a coalition of traitorous subsets can make it much more efficient. When going to next level, it provides an option to further split not only one suspect subset, but rather multiple suspect subsets.

In this paper, we have studied and compared different traitor tracing schemes that have followed different design principles. With our comparison and analysis, we propose future research directions that can lead to simpler and more efficient traitor tracing schemes. As future work, we are interested in defining new traitor tracing model that takes advantage of our observations and that can lead to series of more efficient traitor tracing schemes.

REFERENCES

AACS (2006). Advanced access content system. http://www.aacsla.com.

Boneh, D., Sahai, A., and Waters, B. (2006). Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EuroCrypto 2006, Lecture Notes in computer science*, volume 4004, pages 573–592.

Fiat, A. and Naor, M. (1993). Broadcast encryption. In *Crypto 1993, Lecture Notes in computer science*, volume 773, pages 480–491.

H. Jin, J. L. (2007). Renewable traitor tracing: A trace-revoke-trace system for anonymous attack. In *ESORICS 2007*, pages 563–577.

H. Jin, J. L. and Nusser, S. (2004). Traitor tracing for prerecorded and recordabe media. In *ACM workshop on Digital Rights Management*, pages 83–90.

J. N. Staddon, D. S. and Wei, R. (2001). Combinatorial properties of frameproof and traceability codes. In *IEEE Transactions on Information Theory*, volume 47, pages 1042–1049.

Jin, H., Lotspiech, J., and Megiddo, N. (2008). Efficient coalition detection for traitor tracing. In *IFIP Sec 2008*, pages 365–380.

Naor, D., Naor, M., and Lotspiech, J. B. (2001). Revocation and tracing schemes for stateless receivers. In *CRYPTO '01, Lecture Notes in Computer Science*, pages 41–62.