A CLOSER LOOK AT BROADCAST ENCRYPTION AND TRAITOR TRACING FOR CONTENT PROTECTION

Hongxia Jin
IBM Almaden Research Center
San Jose, CA, USA

Keywords: Content Protection, Broadcast Encryption, Traitor Tracing, Anti-piracy.

Abstract: In this paper we take a closer look at broadcast encryption and traitor tracing in the context of content protection. In current state-of-art, these are viewed as two separate and orthogonal problems. In this paper we challenge this separation. We presented example that shows it can be insecure if a broadcast encryption scheme offers no traceability. We also show it is insufficient to have a traitor tracing scheme that does not have revocation capability and does not support multi-time tracing. Furthermore we show supporting multi-time tracing may actually mean a traitor tracing scheme also needs to have broadcast capability. We hope the evidences we presented in this paper can raise the awareness of the connections between these two problems and shed new insights on future research directions in this important area.

1 INTRODUCTION

In this paper we are concerned with content protection for copyrighted materials. In particular we are concerned with the distribution channel that is one-way. For example, pay-TV system or massively distributing physical media, like DVDs. It is essential for a broadcast encryption scheme to be able to revoke non-compliant users. The goal can be achieved by a solution called *broadcast encryption* (Fiat and Naor, 1993) to emphasize its one-way nature.

When a broadcast encryption scheme is used for content protection, the enabling block is sometimes called MKB (media key block), where the media key is indirectly used to encrypt the content. MKB is a structure that gets put together with the content. It is basically the media key encrypted with compliant private keys. Each compliant device can use his private key to process MKB differently but get the same valid media key. If some devices are compromised and their private keys need to be excluded (revoked), processing MKB will give them garbage. That is how revocation works.

One of the risks for the broadcast encryption scheme is that some of the legitimate users maybe collude to forge a pirate decoder (or a software program) that can decrypt the encrypted content. So pirate decoders enable illegitimate users to watch pay-TV free. To defend against this type of pirate attack, a traitor tracing scheme (Chor et al., 1994; Boneh et al., 2006b) is available to identify at least one colluder (traitor).

Another pirate attack for the broadcast encryption scheme is that some of the legitimate users maybe collude and redistribute the decrypted content or the per-movie encryption key, i.e., the media key. The hackers in this pirate attack can stay anonymous. It is called ”anonymous attack”. A traitor tracing scheme (Safani-Naini and Wang, 2003; H. Jin and Nusser, 2004) is available to identify traitors involved in anonymous attack.

Even though it seems natural that a broadcast encryption scheme should go hand-in-hand with a traitor tracing scheme in any real content protection system, these two has been considered as two separate and orthogonal problems. In this paper we challenge the current separation of considering these two problems.

We will first show in Section 2 the potential serious consequence of designing a broadcast encryption system that does not have traceability. We show a pirated decrypting key may become a global secret that has no way to revoke, thus effectively break the system. In Section 3 we will discuss the insufficiency of designing a traitor tracing system that does not have

Jin H. (2007).
A CLOSER LOOK AT BROADCAST ENCRYPTION AND TRAITOR TRACING FOR CONTENT PROTECTION.
In *Proceedings of the Second International Conference on Security and Cryptography*, pages 295-298
DOI: 10.5220/0002128502950298

broadcast capability so as to revoke traitors. We also believe it is essential to provide continued traceability throughout lifetime. In Section 4 we show providing continued traceability may simply mean adding broadcast capability to a traitor tracing scheme. We conclude in Section 5 for future work.

2 BROADCAST ENCRYPTION SCHEME WITHOUT TRACEABILITY?

In this section, we will discuss the broadcast encryption system, the BGW scheme, recently presented in (Boneh et al., 2005). Let S denote the designated receiving set, and let $\mathbb{G}$ be a bilinear group of prime order q. The BGW scheme for $n-1$ users works as follows:

Setup $(n-1) \Rightarrow ((g, g_1, \cdots, g_n, g_{n+2}, \cdots, g_{2n}, v), (d_1, \cdots, d_{n-1}))$

(1) Pick a random generator $g \in \mathbb{G}$ and random $\alpha, \gamma \in \mathbb{Z}_q$.

(2) Compute $g_i = g^{\alpha^i}$ for $i = 1, \cdots, n, n+2, \cdots, 2n$ and $v = g^{\gamma}$. The public key is

$$PK = (g, g_1, \cdots, g_n, g_{n+2}, \cdots, g_{2n}, v).$$

(3) Compute the private key for user i as $d_i = g_i^{\gamma}, i = 1, \cdots, n-1$.

Encrypt $(S, PK) \Rightarrow (header, K)$

(1) Run the *SigkeyGen* algorithm to obtain a signing key K_{SIG} and a verification key $V_{SIG} \in \mathbb{Z}_q$.

(2) Pick a random $t \in \mathbb{Z}_q$ and set $K = e(g_{n+1}, g)^t$.

(3) Set $C = \left(g^t, (v \cdot g_1^{V_{SIG}} \cdot \prod_{j \in S} g_{n+1-j})^t\right)$ and $header = (C, Sign(C, K_{SIG}), V_{SIG})$.

Decrypt $(S, i, d_i, header, PK) \Rightarrow K$

(1) Let $header = ((C_0, C_1), \sigma, V_{SIG})$. Verify that σ is a valid signature of (C_0, C_1) under the key V_{SIG}. If invalid, output '?'.

(2) Otherwise, pick a random $w \in \mathbb{Z}_q$ and compute

$$\widehat{d_0} = \left(d_i \cdot g_{i+1}^{V_{SIG}} \cdot \prod_{\substack{j \in S \\ j \neq i}} g_{n+1-j+i}\right) \cdot \left(v \cdot g_1^{V_{VIG}} \cdot \prod_{j \in S} g_{n+1-j}\right)^w, \quad \widehat{d_1} = g_i g^w.$$

(3) Output $K = e(\widehat{d_1}, C_1) / e(\widehat{d_0}, C_0)$.

Authors in (Jian Weng and Chen, 2007) showed a way to construct a pirate key and proved the BGW scheme has no traceability. Suppose $k (\geq 2)$ traitors, say, $\{i_1, \cdots, i_k\}$, are involved in the pirate decoding. They forge a private key in the following way.

(1) Let $S^* \supset \{i_1, \cdots, i_k\}$. Here $S^* \backslash \{i_1, \cdots, i_k\}$ is the set of innocent users.

(2) Each traitor i_l chooses a random $\beta_l \in Z_q^*$, and computes $g_{i_l}^{\beta_l}$, $g_{i_l+1}^{\beta_l}$, $d_{i_l}^{\beta_l}$, $\left(\prod_{\substack{j \in S^* \\ j \neq i_l}} g_{n+1-j+i_l}\right)^{\beta_l}$ and $\left\{g_{n+1-j+i_l}^{\beta_l}, \quad j \in \{1, \cdots, n\} \backslash S^*\right\}$.

(3) The k traitors use secure multi-party computation to get $\beta = \sum_{l=1}^{k} \beta_l \mod q$, and compute $\beta^{-1} \mod q$.

(4) The forged private key consists of the following $n + 4 - |S^*|$ components

$$\left(\prod_{l=1}^{k} g_{i_l}^{\beta_l}\right)^{\beta^{-1}}, \left(\prod_{l=1}^{k} g_{i_l+1}^{\beta_l}\right)^{\beta^{-1}},$$
$$\left(\prod_{l=1}^{k} d_{i_l}^{\beta_l}\right)^{\beta^{-1}},$$
$$\left(\prod_{l=1}^{k} (\prod_{\substack{j \in S^* \\ j \neq i_l}} g_{n+1-j+i_l})^{\beta_l}\right)^{\beta^{-1}},$$
$$\left\{\left(\prod_{l=1}^{k} g_{n+1-j+i_l}^{\beta_l}\right)^{\beta^{-1}}, j \in \{1, \cdots, n\} \backslash S^*\right\}.$$

Note. When $k = 2$, one traitor is able to derive the other's private key. When $k > 2$, none of private keys could be exposed unless $k-1$ traitors collude, which is ensured by secure multi-party computation.

Theorem 1 *(Jian Weng and Chen, 2007) Let S^* be defined as above. With the above forged pirate key, a pirate decoder can be built to recover the session key $K = e(g_{n+1}, g)^t$ from the broadcast aiming to any set S with $S \supseteq S^*$.*

Theorem 2 *(Jian Weng and Chen, 2007) The above pirate decoder is untraceable, even if the forged private key is retrieved.*

Basically it can be shown that given the above forged private key, any subset S' of users, with $S' \subseteq S^*$ and $|S'| > 2$, may have colluded to forge the private key. Therefore, no tracing algorithm can distinguish the subsets and tell the exact set of traitors, even if the forged private key is obtained.

While the authors in (Jian Weng and Chen, 2007) proved the BGW scheme has no traceability, we want to point out the consequence is a lot more serious than simply no traceability. In fact, in some cases one does not need to trace. For example, in a subscription-based system, a subscribers subscription may expire. However, regardless of traceability, a broadcast encryption must be able to revoke pirated key. A system that cannot revoke keys is useless for content protection.

It is fine if one can revoke all the keys in a coalition that constructed that pirate key, assuming it effectively revokes the forged key. Unfortunately in the above shown scenario, one cannot do that. Indeed any $S' \subseteq S^*$ and $|S'| > 2$ users may have colluded to forge that private key. The broadcast encryption scheme has no way to revoke this forged key. It can become a global secret and the broadcast encryption scheme is therefore broken. This broadcast encryption scheme is not sound.

3 TRAITOR TRACING WITHOUT REVOCATION CAPABILITY?

We want to argue that a traitor tracing scheme without revocation capability is of little value if not useless in reality. However, while there exist some trace-and-revoke systems (Boneh et al., 2006a) for clone pirate attack, it is considered optional rather than a must-have. As a result, for some existing schemes, it is impossible to add revocation capability on top of tracing.

As an example, the same authors for the above broadcast encryption scheme came up with a separate traitor tracing scheme which appeared in Eurocrypt 2006 (Boneh et al., 2006b). Without going to much detail, we point out that scheme is impossible to revoke. In fact, if one needs to revoke a detected hacking device i, one would also have to revoke devices $i+1 \ldots N$. Of course this is unacceptable.

4 TRAITOR TRACING DOES NOT SUPPORT MULTI-TIME TRACING?

To be practically useful, a traitor tracing system must be able to trace again responding to new attack after the previous traitors are identified and revoked. However it is not always easy to achieve continued tracing after revocation. We show this using a traitor tracing scheme, the JLN scheme (H. Jin and Nusser, 2004), that was designed to defend against anonymous attack.

Recall that in anonymous attack the attackers redistribute the media key or the decrypted content itself just to stay anonymous and avoid being identified. To defend against anonymous attack, different versions of the content encrypting key as well as the content are needed. To do that, the content is divided into multiple segments of which n segments are chosen, each to have q variations. These variations are not only differently watermarked, but also differently encrypted. Each device can only decrypt one of the variations for each segment. In other words, each device plays back the content through a different path. This effectively builds different versions of the content. Each version of the content contains one variation for each segment. In order to avoid having large number of variations at any single point and still be able to support large number of users, the JLN scheme adopted two levels of codes. The ”inner code” is used to assign variations within a movie to effectively create multiple versions of a movie, and the ”outer code” is used to assign movie versions over a sequence of movies.

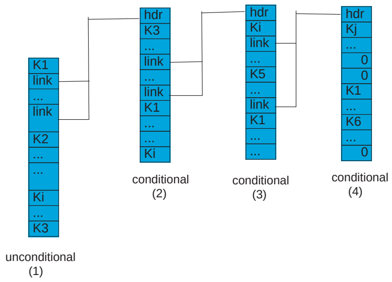

Figure 1: Sample SKB.

The traitor tracing keys for the above scheme are assigned from a large matrix based on the outer code. The columns correspond to the movies in the sequence, the rows correspond to different versions for each movie. Each device is assigned exactly one key from each column. The tracing keys are called ”sequence keys”.

Similar as Media Key Block (MKB), one can build a structure called Sequence Key Block (SKB) to revoke sequence keys. The idea is to revoke the entire set of sequence keys owned by a traitor. Of course, many devices might share a single compromised key. The purpose of the Sequence Key Block is to give all innocent devices a column they can use to calculate a correct answer, while at the same time preventing compromised devices (who have compromised keys in all columns) from getting to a correct answer. Keep in mind in an SKB there are actually many correct answers, one for each variation in the content.

Unfortunately, the above combined scheme cannot support multi-time continued tracing. If the attackers combine the revoked keys with the keys that have not been detected, there are multiple paths to obtain the same valid answer. In other words, it is not always possible to know from which column the SKB processing ends to get a valid key, thus it hurts tracing.

To force the undetected traitors to reveal the keys they use when processing SKB, one must make sure each column gets different variations so that when recovering a key/variation, the scheme knows from which column it comes from. That puts challenges on how to design SKB to broadcast the new content so that the SKB not only revokes the identified compromised sequence keys but also continues to provide tracing information to the license agency to enable continued tracing. It essentially demands a traitor tracing scheme to have special broadcast capability.

5 CONCLUSION

In this paper we challenge the current definition of the security of a broadcast encryption and the separation between the broadcast encryption problem and traitor tracing problem.

Firstly, we pointed out that a broadcast encryption scheme without traceability can be insecure and impractical in reality. We do this by showing a broadcast encryption system recently presented in (Boneh et al., 2005) can be broken due to the impossibility of revoking the pirate key constructed by the attacker.

Second, we show that a traitor tracing system must be able to revoke and broadcast again to be useful. We show that current researches donot realize this problem, As a result, it is impossible to add revocation capability on top of some existing traitor tracing schemes. We also believe a traitor tracing system must be able to trace again to support multi-time tracing in order to be useful in real world. Unfortunately we show in some cases supporting multi-time tracing poses some new challenges on broadcasting.

The presented evidences in this paper show these two areas are much more closely connected and related. We hope the results in this paper can shed some insights on new directions for future work. We believe the relationship between these two topics needs to be further studied. Furthermore, is it possible to formally prove these two problems are actually equivalent, at least for some type of attacks like the one shown in Section 4?

REFERENCES

Boneh, D., Gentry, C., and Waters, B. (2005). Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Crypto 2005, Lecture Notes in computer science*, volume 3621, pages 258–275.

Boneh, D., Gentry, C., and Waters, B. (2006a). A fully collusion resistant broadcast, trace and revoke system. In *ACM conference on Computer and Communication Security*, pages 211–220.

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

Chor, B., Fiat, A., and Naor, M. (1994). Tracing traitors. In *Crypto 1994, Lecture Notes in computer science*, volume 839, pages 480–491.

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

Jian Weng, S. L. and Chen, K. (2007). Pirate decoder for the broadcast encryption schemes from crypto 2005. In *Series F: Information Science special issue on Information Security, June issue*.

Safani-Naini, R. and Wang, Y. (2003). Sequential traitor tracing. In *IEEE Transactions on Information Theory*, volume 49, No.5, pages 1319–1326.