TRAITOR TRACING IN CONTENT DISTRIBUTION: STATE OF THE ART

Peter Meerwald-Stadler

Schießstandstr. 3a, 5061 Elsbethen, AT
pmeerw@pmeerw.net; https://pmeerw.net

Keywords: ***Broadcast encryption, fingerprinting, copyright protection***

Abstract: *Traitor tracing algorithms employed by the content provider allow to trace back content piracy to the source, thus discouraging redistribution of decryption keys and decrypted content. Due to recent advances, e.g. Tardos codes, and research results on cryptographic primitives, e.g. indistinguishability obfuscation, technological barriers are reduced and we might see more of these systems employed. We simply review the state-of-the-art in traitor tracing algorithms, and highlight their capabilities and limitations in plain language.*

1. Introduction

Broadcast encryption [Fiat/Naor 1993] aims to deliver content to legitimate users and protect the copyright of the content owner. Such systems are in widespread use, for example AACS [Noar/Noar/Lotspiech 2001, Henry/Sui/Zhong 2007] used on Blu-ray discs or for Pay-TV as there is considerable commercial interest to control content distribution [Danaher/Smith/Telang 2017]. Nevertheless, from the angle of privacy and anonymity as well as fair use [Qureshi 2013], broadcast encryption as the foundation of digital rights management (DRM) is problematic [Anderson 2008]. Technological means of protection are always intertwined with legal protections (legal circumvention prohibition, protection by contract, protection by technology licenses) [Bechtold 2002].

In this work, we focus on technical measures to protect the content and keep consumers honest by implementing a tracing ability to identify pirates who may have leaked protected material. There is no romantic notion attached to «pirates» (a coalition of dishonest consumers) that may perform attacks on the system. Our view is in terms of a game between a tracing authority and the pirates: false accusations must be avoided and pirates aim to disguise their identity when redistributing protected material. Identified traitors can be disconnected from further content transmissions and legal evidence of a pirate's identity may deter potential traitors. Further notes on legal aspects can be found in the discussion section.

2. Problem Statement

Broadcast encryption delivers content in encrypted form over a broadcast channel, e.g. via satellite or an optical disc. Qualified consumers may use a conforming decoder to decrypt the content. There are several challenges: First, only a subset of consumers (or decoders) should be able to decrypt – those that are paying and compliant; it should be possible to revoke access for non-compliant parties. Second, a group of consumers may share their decryption keys or their decrypted content and fabricate a pirate decoder or pirate content; in this case it should be possible to identify members of the «traitor» group by means of a tracing algorithm. Third, the overhead in terms of transmission and key length must be minimized, as well as the risk of «accusing» an innocent consumer. Traitor tracing is a forensic tool addressing the second and third challenge. It can be the basis to take further legal or business actions against identified individuals. Note that traitor tracing is an active forensic technique, it prepares and manipulates the keys or content upfront to enable tracing. In this section, we look at traitor tracing in broadcast encryption in more detail.

In Figure 1, we illustrate a basic broadcast encryption scenario where the content provider prepares encrypted content consisting of enabling blocks (EB) and cipher blocks (CB) and broadcasts it to its consumers, cf. [Prihandok 2013]. A subset of legitimate consumers will be able to decrypt the content protected (CB) by a session key which they derive by applying a set of personal keys [Chor/Fiat/Naor 1993]. To keep the session key s secret, it is split into shares $s_1, s_2, ..., s_{\text{ld N}}$ such that $s=s_1 \oplus s_2 \oplus ... \oplus s_{\text{ld N}}$ where N is the number of possible consumers and *ld* denotes the base 2 logarithm. Each share s_j is encrypted using two keys $k_{\text{j},0}$ and $k_{\text{j},1}$ so that the enabling block consists of $2 \cdot ld\ N$ subblocks. Each consumer is assigned a personal key K_i comprising a set of *ld N* keys: $K_\text{i}=\{k_{\text{j},b(\text{i},\text{j})}\}$ where *b(i,j)* is the *j*th bit of the binary representation of *i*. By using the personal key, a consumer can decrypt each share, either using the *0* or *1* key depending on the assignment, and then combine the shares to obtain the session key in order to unlock the cipher blocks. Note that only a subset of the $k_{\text{j},0}$ and $k_{\text{j},1}$ are revealed to each customer. Hence, a pirate decoder can only make use of the combined keys available to the traitors.

In this work, we focus on static (i.e. the key allocation is determined upfront) and symmetric broadcast encryption schemes. Since in symmetric schemes the secret is generated by the content provider and shared with the consumer, there is the problem of trust in the arbitration of disputes. This led to asymmetric broadcast encryption schemes that provide non-repudiation [Pfitzmann 1996].

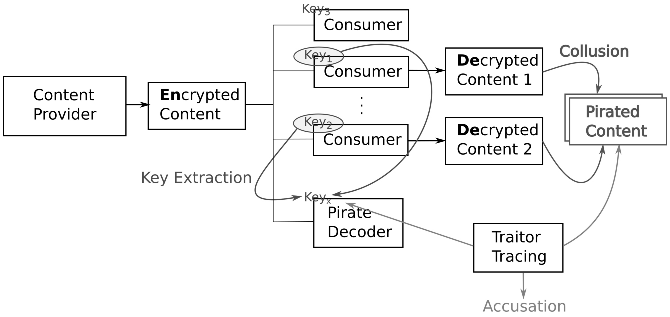

Figure 1: Attack scenarios in content distribution

Any coalition of revoked users should not be able to decrypt the content. Furthermore, it should be possible to disable a pirate decoder from decrypting further transmissions [Naor/Naor/Lotspiech 2001].

A pirate may hack a decoder device receiving the encrypted broadcast and extract its personal key. Using the extracted key, a pirate decoder could then be built which outputs the decrypted content, ignoring any restrictions imposed by DRM. Even worse, the pirate decoder could be made widely available as happened in the case of the DeCSS software [González 2002], for decrypting DVD content.

To discourage such piracy, once a pirate decoder is found, the content provider can run a tracing algorithm to recover the identity of at least one consumer (turned into a traitor) that collaborated in the pirate construction. Note that the number of the pirate collaborators will typically be small; a traitor tracing system is called *t*-resistant if it can withstand collusions of up to *t* consumers. For *t=N*, the system is called fully collusion resistant.

After decryption of the content, a pirate may choose to capture and redistribute it. Technologies such as HDCP [Crosby et al. 2002] aim to protect content transmission to compliant playback devices, limiting capture devices. Here, data fingerprinting, also called transactional watermarking, can add a second layer of protection by embedding the identity of the consumer in the decrypted content. If the copy protection mechanism fails, there still is the deterrent that a particular consumer can be associated with the pirate content. The system comprises two layers: a watermarking layer, modulating and imperceptibly adding a signal to the content in a robust way, and a coding layer to encode the consumer identifier in a collusion resilient way, i.e. the fingerprinting code [Schaathun 2006]. Again, pirates may try to combine their content copies in order to produce pirate content which cannot be traced back to the consumers taking part in the collusion. In addition, they may also target the watermarking layer, i.e. aim to render the embedded signal undetectable; having several copies of the same content can be exploited to estimate the watermark signal.

There are two important performance parameters – the size of the consumer key and the size of the ciphertext, the EB. Fingerprinting codes are closely related to traitor tracing [Boneh/Sahai/Waters, 2006]: a fingerprinting code can be turned into a fully-collusion resistant traitor tracing scheme with constant EB size and consumer key size in $O(N^3)$ [Boneh/Shaw 1998] and $O(N^2)$ [Tardos 2008].

3. Countering the pirate decoder

Implementing cryptography (such as for the decoder in a broadcast encryption scheme) or content fingerprinting in an untrusted environment (such as the set-top box owned by a consumer) is hard. Secret key material may be extracted and DRM measurements may be bypassed after opening «the box» (which is mostly software) by means of reverse engineering and analysis. In this section, we review very recent results giving hope that tamper-resistant software may be possible and leading to reduced overhead for broadcast encryption. We caution that the discoveries do not immediately lead to practical code (yet), a lot more research is needed.

Encryption secures the communication between two endpoints. The problem in the broadcast encryption scenario is that the consumer endpoint is not an ideal black box, but its operation can be observed by a curious and sometimes malicious party. To improve security, tamper-resistant hardware can be employed, and physically unclonable functions (PUFs) [Maes/Verbauwhede 2010, Katzenbeisser et al. 2012] as a hardware-based one-way function can be used. Another approach is to turn to white-box cryptography [Wyseur 2009] in order to create a tamper-resistant program that can be safely executed in such an untrusted environment. A white-box cryptographic primitive can be observed, probed and executed but will not reveal the key embedded in the implementation. Careful definition of the security properties and the underlying assumptions are paramount to the understanding. This field of research is called software obfuscation and deals with «encrypting functionalities» [Horváth 2015, Barak 2016]. Applications range from hiding a secret algorithm, secure software patching without disclosing the vulnerability being addressed, watermarking of software, backdoor functions, etc. Today, often heuristical obfuscators are used, relying on security by obscurity and secrecy, violating Kerckhoffs's principle.

Software can be understood as the compression of the truth table of a function f. The ideal encryption of f is a lookup table of its input-output pairs [Sahai 2014]. Unfortunately, such a software representation has exponential size while we want the obfuscated program's size to be limited by a polynomial. Can the lookup table of f be compressed without revealing its structure to a computationally bounded adversary?

In 2001, [Barak/Goldreich/Impagliazzo/Rudich/Sahai/Vadhan/Yang 2012] gave a rigorous definition of the security of an obfuscator; they required that an obfuscated version $\bar{O}(P)$ of a program P to behave like a «virtual black box» (VBB) in the sense that anything one can compute from it, can also be computed from the input-output behaviour of P. They showed that such an obfuscator cannot be realized in general by pointing out a family of programs where the VBB property does not hold. Interestingly, they also suggested a useful weaker definition: indistinguishability obfuscation (iO) requires that if two programs of similar size compute the same function, than their obfuscations should be indistinguishable. This does not state any guarantee that the obfuscated version actually hides any information. However, [Goldwasser/Rothblum 2007] proved that for efficient obfuscators, the definitions of indistinguishability and best-possible obfuscation are equivalent. [Garg/Gentry/Halevi/Raykova/Sahai/Waters 2013] proposed the first candidate construction following the notion of iO. This breakthrough discovery was prepared by another landmark result: fully homomorphic encryption (FHE) [Gentry 2009].

FHE allows to evaluate a function on ciphertext, such that the result is equivalent to the encryption of the function's evaluation on the original plaintext. Partial homomorphic encryption, i.e. only supporting either addition **or** multiplication but not both, was known before. For privacy in cloud services, FHE is a crucial prerequisite as computation can be performed on the server side in an encrypted domain, without disclosing sensitive plaintext data. For the construction of iO, a primitive named cryptographic multilinear maps is used, also named «homomorphic quasi encryption» [Barak 2016]. In addition to the standard operations *Enc, Dec,* $+$, $\times$ there is a «comparison» operation $Enc(x) \overset{?}{=} Enc(y)$ which is true if $x{=}y$.

Modern cryptography reduces the security of cryptographic primitives into the conjectured hardness of a few well-studied mathematical problems (factoring integer, for example). For the indistinguishability obfuscator, no such security argument can be made (yet?). Nevertheless, iO is emerging as a central tool in cryptography, from which primitives can be derived, such as deniable encryption, public key encryption, etc. [Barak 2016]. The initial construction of iO may be secure against quantum cryptographic attack [Sahai 2014]. A lot of research effort will be needed to turn the theoretical promises of iO into practical code [Horváth 2015].

Returning to a fully collusion resistant traitor tracing scenario, the ciphertext length can be reduced from $O(N)$ [Chor/Fiat/Naor 1994], to $O(\sqrt{N})$ [Boneh/Sahai/Waters 2006], to $(\log N)^{O(1)}$ assuming iO [Tang/Zhang 2015].

4. Tracing pirate content

Digital watermarking has been proposed in the 1990s for copy control and copyright protection purposes [Cox et al. 2002]. Data fingerprinting or transactional watermarking aims to embed information identifying the legitimate consumer into the content in a robust way. The embedded signal must resist common media manipulation, such as recompression, resampling, etc. Different from watermark robustness is the notion of watermark security [Furon 2016], where the attacker is assumed to possess all knownledge about the watermarking method except a secret key. Embedding a fingerprinting code in the broadcast encryption scenario at a consumer's decoder will result in the decrypted content being fingerprinted (see Figure 1). Pirates who choose to redistribute the content run the risk of being traced. A coaliton of pirates may combine their content copies in order to lower their risk. By comparing their copies, pirates can identify different segments of the content that encode different symbols of the fingerprinting code. For simplicity, we assume a binary code here, but there may be q-ary symbols. Besides attacks on the watermarking layer, pirates can perform powerful collusion attacks on the fingerprinting code, often subject to the marking assumption or more relaxed assumptions (basically, the attacker must work with fingerprint symbols received and cannot create content with arbitrary fingerprinting symbols).

[Tardos 2003, Tardos 2008] introduced a probabilistic fingerprinting code and showed that it is optimal in the sense that the code length necessary to fulfill the following requirements (N users, t colluders, probability of accusing at least one innocent below P_{fp}) has the minimum scaling in $O(t^2 \cdot \log N \cdot P_{\mathrm{fp}}{}^{-1})$, improving significantly upon previous codes [Boneh/Shaw 1998]. Many results have been obtained since then: the worst case attack a collusion of size t can produce, and also the best counter attack – a game between pirate and traitor tracer [Huang/Moulin 2012]. An open problem is the construction of an efficient fingerprint decoder [Meerwald/Furon 2012] under real-world constraints. The decoder can be improved if the size of the collusion or the collusion strategy is known or can be estimated. Further decoding benefits can be obtained by tracing tuples of colluders instead of single pirates being part of a collusion – at the cost of decoding complexity.

Before a fingerprint can be traced, a pirate copy of the content has to be obtained which is not without risk as automated search may target also content of third party copyright holders [Steinebach 2010]. This is especially problematic in peer-to-peer networks with simultaneous upload.

5. Discussion

Technology has advanced at a rapid pace in recent years and research on obfuscation may change our understanding of cryptography. Yet it appears that copyright protection in the «real world» so far is more based on legal intimidation, control of redistribution, obsolescence. Demand-side and supply-side anti-piracy strategies [Danaher/Smith/Telang 2017] target consumers looking for pirate content and sites offering means to share copyright-infringing files on the Internet. Anti-piracy actions include per-site take-down notices, blocking access to the Internet, content filters, manipulating search results, data retention, evidence based on logged IP addresses and other collateral damage. Traitor tracing has the potential to trace piracy to the source, in order to police the terms of usage imposed by a content provider and accepted by a consumer. Revoking device keys has worked for AACS to some extent, yet it is unclear if the content tracing ability has been used. Little is known if and how tracing algorithms are employed; it is unlikely that fully collusion secure schemes are a requirement, large coalitions are hard to form unnoticed in practice – and not everyone is a pirate.

Ownership of a device should allow reverse engineering in order to understand its functioning, check compliance, achieve interoperability, etc. Software protection by obfuscation might not only pose a legal barrier to reverse engineering, but also a technical one in the future.

Is an imperceptible watermark providing proof? Can a consumer being identified in a pirate collusion be held liable? Is it a marked copy of a document that I am about to leak... and what precautions should be taken?

Rightsholders have many options to control content distribution and discourage redistribution of pirated material. Research literature frames traitor tracing as a «game» between content owner and pirate, we got a lot more sophisticated in playing it in recent years. It remains to be seen if rightsholders pick up and implement the technology and how this affects our handling of digital media.

6. References

Anderson, Ross, Copyright and DRM, Security Engineering, chapter 22, Wiley, 2nd edition, 2008.

Barak, Boaz/Goldreich, Oded/Impagliazzo, Russell/Rudich, Steven/Sahai, Amit/Vadhan, Salil/Yang, Ke, On the (im)possibility of obfuscating programs, Journal of the ACM, 59(2), 1–48, April 2012.

Barak, Boaz, Hopes, fears, and software obfuscation: What does it mean to be secure? Communication of the ACM, 59(3):88–96, March 2016.

Bechtold, Stefan, From copyright to information law – implications of digital rights management. T. Sander, editor, Security and Privacy in Digital Rights Management, DRM '01, Lecture Notes in Computer Science, 2320, 213–232. Springer-Verlag, 2002.

Boneh, Dan/Shaw, James, Collusion-secure fingerprinting for digital data, IEEE Transaction on Information Theory, 44(5):1897–1905, September 1998.

Boneh, Dan/Shaw, James, Collusion-secure fingerprinting for digital data (extended abstract), Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '95, 452–465, London, UK, 1995. Springer-Verlag.

Boneh, Dan/Sahai, Amit/Waters, Brent, Fully collusion resistant traitor tracing with short ciphertexts and private keys, Cryptology ePrint Archive, Report 2006/045, 2006. http://eprint.iacr.org/2006/045 (all Internet sources accessed on 31 January 2017).

Chor, Benny/Fiat, Amos/Naor, Moni, Tracing traitors, Proceedings of the International Cryptology Conference on Advances in Cryptology, Crypto '94, Lecture Notes in Computer Science, 839, 257–270. Springer-Verlag, 1994.

Cox, Ingemar/Miller, Mathew/Bloom, Jeffrey/Fridrich, Jessica/Kalker, Ton, Digital Watermarking and Steganography, Morgan-Kaufmann, 2nd edition, 2002.

Crosby, Scott/Goldberg, Ian/Johnson, Robert/Song, Dawn/Wagnerl, David, A cryptanalysis of the high-bandwidth digital content protection system. Security and Privacy in Digital Rights Management, Lecture Notes in Computer Science, 2320, 192–200, Springer, May 2002.

Danaher, Brett/Smith, Michael/Telang, Rahul, Copyright enforcement in the digital age: Empirical evidence and policy implications, Communications of the ACM, 60(2):68–75, February 2017.

Fiat, Amos/Naor, Moni, Broadcast encryption, Proceedings of the International Cryptology Conference on Advances in Cryptology, Crypto ’93, Lecture Notes in Computer Science, 773, 480–491. Springer-Verlag, 1993.

Furon, Teddy, Watermarking security, Information hiding, chapter 6, Artech House, 2016.

Garg, Sanjam/Gentry, Craig/Halevi, Shai/Raykova, Marianna/Sahai, Amit/Waters, Brent, Candidate indistinguishability obfuscation and functional encryption for all circuit, Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS ’13, 40–49, Berkeley, CA, USA, October 2013. IEEE.

Gentry, Craig, Fully homomorphic encryption using ideal lattices, Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC ’09, 169–178, Bethesa, MD, USA, May 2009.

Goldwasser, Shafi/Rothblum, Guy, On best-possible obfuscation, Proceedings of the 4th Conference on Theory of Cryptography, TCC ’07, 194–213, Amsterdam, The Netherlands, February 2007. Springer.

González, Andés Guadamuz, Trouble with prime numbers: DeCSS, DVD and the protection of proprietary encryption tools, The Journal of Information, Law and Technology, 3, December 2002.

Henry, Kevin/Sui, Jiayuan/Zhong, Ge, An overview of the advanced access content system (AACS). Technical Report CACR 2007-25, Centre for Applied Cryptographic Research, University of Waterloo, Waterloo, Canada, 2007.

Horváth, Máté, Survey on cryptographic obfuscation. Cryptology ePrint Archive, Report 2015/412, 2015. http://eprint.iacr.org/2015/412.

Huang, Yen-Wei/Moulin, Pierre, One the saddle-point solution and the large-coalition asymptotics of fingerprinting games, IEEE Transactions on Information Forensics and Security, 7(1):160–175, February 2012.

Katzenbeisser, Stefan/Kocabaş, Ünal/Rožić, Vladimir/Sadeghi, Ahmad-Reza/Verbauwhede, Ingrid/Wachsmann, Christian. PUFs: Myth, fact or busted? a security evaluation of physically unclonable functions (PUFs) cast in silicon, Cryptology ePrint Archive, Report 2012/557, 2012. http://eprint.iacr.org/2012/557.

Maes, Roel/Verbauwhede, Ingrid, Physically unclonable functions: A study on the state of the art and future research directions, pages 3–37. Information Security and Cryptography. Springer, October 2010 .

Meerwald, Peter/Furon, Teddy, Toward practical joint decoding of binary Tardos fingerprinting codes, IEEE Transactions on Information Forensics and Security, 7(4):1168–1180, April 2012.

Naor, Dalit/Naor, Moni/Lotspiech, Jeffrey, Revocation and tracing schemes for stateless receivers. Advances in Cryptology, CRYPTO ‘01, Lecture Notes in Computer Science, 2139, 41–62. Springer, 2001.

Pfitzmann, Birgit, Trials of traced traitors, Proceedings of the First International Workshop on Information Hiding, Lecture Notes in Computer Science, 1174, 49–64, Cambridge, UK, May 1996. Springer.

Prihandok, Antonius/Ghodosi, Hossein/Litow, Bruce, A review on traitor tracing schemes, Technical report, James Cook University, Australia, September 2013. Available: https://www.researchgate.net/publication/256498106.

Qureshi, Amna/Rifà-Pous, Helena/Megías Jiménez, David, A survey on security, privacy and anonymity in legal distribution of copyrighted multimedia content over peer-to-peer networks, Technical report, Internet Interdisciplinary Institute, Universitat Oberta de Catalunya, Barcelona, Spain, September 2013. Working Paper Series DWP13-001.

Sahai, Amit, How to encrypt a functionality, Talk at the Quantum Games and Protocols Workshop, Simons Institute, University of California at Berkeley, February 2014, http://simons.berkeley.edu/talks/amit-sahai-2014-02-25.

Schaathun, Hans Georg, On watermarking/fingerprinting for copyright protection, Proceedings of the First International Conference on Innovative Computing, Information and Control, ICICIC ‘06, 3, pages 50–53, Beijing, China, August 2006.

Steinebach, Martin/Wolf, Patrick/Kim, Jee-Un/Engelhardt, Jens, Legal aspects of watermarking search services, A. Arnab, editor, Proceedings of the 8th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods: Incorporating the 6th International ODRL Workshop, 153–159, Namur, Belgium, September 2010.

Tang, Bo/Zhang, Jiapeng, Barriers to black-box constructions of traitor tracing system, Cryptology ePrint Archive, Report 2015/1070, 2015, http://eprint.iacr.org/2015/1070.

Tardos, Gabor, Optimal probabilistic fingerprint codes, Proceedings of the 35th Annual ACM Symposium on Theory of Computing, 116–125, San Diego, CA, USA, June 2003.

Tardos, Gabor, Optimal probabilistic fingerprint codes, Journal of the ACM, 55(2):1–24, May 2008.

Wyseur, Brecht, White-Box Cryptography, PhD thesis, Katholieke Universiteit Leuven, Belgium, March 2009.