# Tracing and Revoking Pirate Rebroadcasts

Aggelos Kiayias* and Serdar Pehlivanoglu*

Computer Science and Engineering, University of Connecticut
Storrs, CT, USA
{aggelos,sep05009}@cse.uconn.edu

**Abstract.** All content distribution systems are vulnerable to the attack of rebroadcasting: in a pirate rebroadcast a pirate publishes the content in violation of the licensing agreement. This attack defeats any tracing mechanism that requires interaction with the pirate decoder for identifying compromised keys. Merely tracing pirate rebroadcasts is of little use and one should be also able to revoke the involved traitor keys. The only currently known scheme addressing this issue is implemented as part of the Advanced Access Content System (AACS) used in Blu-Ray and HD-DVD disks. In this paper we perform an analysis of this construction and we find it has serious limitations: the number of revocations is bound by the size of the receiver storage (for the actual parameters reported this is merely 85 keys).

We address the limitations of the state of the art (i) by formally modeling the problem of tracing and revoking pirate rebroadcasts and (ii) by presenting the first efficient constructions of tracing and revoking pirate rebroadcasts that are capable of performing tracing for *unlimited numbers* of traitors and revoking *unlimited numbers* of users. We present three instantiations of our framework: our first construction achieves a linear communication overhead in the number of revoked users and traitors and is capable of eliminating a pirate rebroadcast by any number of traitors in time that depends logarithmically in the number of users and polynomially on the number of revocations and traitors. Our second construction assumes a fixed bound on the number of traitors and improves the elimination time to depend only logarithmically on the number of revocations. Both of these constructions require merely a binary marking alphabet. Our third construction utilizes a larger marking alphabet and achieves even faster pirate rebroadcast elimination; our analysis improves the previously known bound for the same alphabet size due to Fiat and Tassa from Crypto'99 while offering revocation explicitly.

## 1 Introduction

In the broadcast encryption setting a center broadcasts content to a number $N$ of receivers. The center wishes to utilize the broadcast medium in such a way so that it can revoke at will any subset of size $R$ from the population of receivers for any transmission. This requirement makes it impossible to hand the

* Research partly supported by NSF Awards 0447808, 0831304, 0831306.

M. Abdalla et al. (Eds.): ACNS 2009, LNCS 5536, pp. 253–271, 2009.

same key to all receivers. The two trivial solutions to the broadcast encryption problem exhibit the trade-off between the receiver storage requirement and the ciphertext length. In the first trivial solution, each receiver obtains a personal key and subsequently the center can use the broadcast medium to simulate a unicast by transmitting a (vector) ciphertext of length $N-R$. While this solution is optimal from the receiver storage point of view, it wastes a lot of bandwidth. In the second trivial solution the center assigns a different key for any subset of receivers and each receiver is handed the keys for all the subsets it belongs to. In this case the ciphertext has optimal length but each receiver is required to store $2^{N-1}$ keys which is an exponential blow-up. Since the introduction of the first non-trivial scheme in [13] a number of subsequent works gave solutions exhibiting improved trade-offs [11,16,17,28]. Notably, [29] introduced the subset-cover framework that enabled the first schemes with ciphertext length linear in the number of revoked users $R$ that allow unlimited revocations.

While revocation is an operation of critical importance, it is not sufficient for a content distribution system. Malicious receivers may obtain access to their keys (e.g., by reverse engineering their decoders) and then leak their key material to a "pirate." The pirate subsequently can construct a decoder that employs all these keys. The adversarial receivers in this setting are called traitors. A traitor tracing scheme [7] is a scheme that was suggested to deal with this problem: in a traitor tracing scheme, the center possesses the capability to interact with a pirate decoder and recover the identities of the traitors. Presumably after identification such traitors can be revoked. A number of subsequent works in [4,9,12,22,23,24,26,27,30,32,34,35,36,38,39,40] designed improved traitor tracing schemes and related codes dealing with this problem which has also being called the "clone decoder attack".

Combining the two functionalities of tracing and revoking in a single system is not straightforward. This was identified in [31] and Naor, Naor and Lotspiech [29] introduced trace and revoke schemes that are capable of offering a combined functionality that can deal with the problem of disabling pirate decoders. Subsequent work in the subset-cover framework of [29] gave better constructions [2,15,18,19,41] while also limitations were discovered in the form of a type of attacks called pirate evolution in [25].

*Employing Trace and Revoke Schemes in practice.* It should be noted that all trace and revoke schemes rely on multiple encryptions of the same plaintext under different keys something that suggests that they are not suitable for direct encryption of large messages. Given that the intended application scenario is content distribution it is expected that an encryption mechanism would have to handle large messages. The way this is solved is by employing hybrid encryption: the trace and revoke scheme is used to encrypt a one-time content key $K$ and subsequently the content is appended encrypted with the key $K$.

*The Pirate Rebroadcast Attack.* The scenario of pirate rebroadcast attacks in the context of traitor tracing was introduced by Fiat and Tassa, [14]. In this scenario, the traitors first decrypt the content by using their key material and then once

it is in clear text form, they rebroadcast the content[1]. In the hybrid encryption setting described above the attack is even worse: they can simply publish the content key $K$ thus avoiding bulky uploads to online storage systems or other distribution mechanisms for content. Clearly a trace and revoke scheme would be useless against a pirate rebroadcast attack: the center is entirely powerless in handling such an attacker as the output of the rebroadcast itself provides no information about the traitor keys.

A solution to this problem would be feasible by employing watermarking techniques (e.g., such as those of [10]) so that the content itself becomes varied over the user population. As before the trivial solution would be marking the content individually so that each user has its own copy. As it was the case for broadcast encryption this solution wastes too much bandwidth. There are essentially two techniques known in the literature for obtaining non-trivial solutions that relax the bandwidth requirement: one is dynamic traitor tracing [14] and the other is sequential traitor tracing [33,21]. The idea in both cases is similar: the center will induce a marking of content and by observing the feedback from the pirate rebroadcast it will identify the traitors (refer to figure 1 for an illustration). The two methods differ in the following way: in the former, after each transmission the center obtains the feedback and tries to localize the suspect list by reassigning the marks adaptively. The number of traitors is not known beforehand and the system adjusts itself after each feedback. In the latter setting, the assignment of marks to the variations is predetermined (hence the transmission mechanism is not adaptive to the feedback). Note that depending on the parameters used, it may take a number of transmissions till the system converges and identifies one traitor.

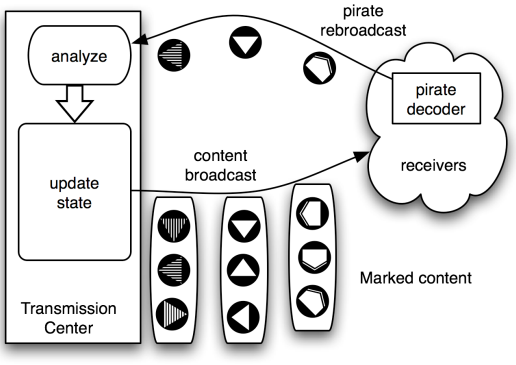


**Fig. 1.** The model for tracing a pirate rebroadcast attacker

[1] This attack has also being called the "anonymous attack" in [21,20]. We opt for calling it pirate rebroadcast instead given that we find it more descriptive of the adversarial action it describes; the same term was also used in the original paper [14]. Moreover, "anonymity" is a heavily overloaded term in the computer security literature and adding another connotation to it perhaps should be best if it is avoided.

*Tracing and Revoking Pirate Rebroadcasts.* In the above techniques that dealt with pirate rebroadcast attacks, the tracing mechanism does not provide a revocation capability. This is an important concern as it is not straightforward how to add revocation on top of either dynamic traitor tracing or sequential traitor tracing. To see why the straightforward approach fails suppose one decides to cascade a broadcast encryption at the decoder level with, say, a sequential traitor tracing scheme by composing the two encryption functions. This means that legitimate decoders will have to possess independent sets of keys from both schemes, i.e., one set of keys for the encryption/decryption involved in the sequential traitor tracing that binds the marked content to a receiver and one set of keys for the encryption/decryption involved in the broadcast encryption. It is easy to see that a pirate possessing the key material of as few as two traitor users can evade revocation at the decoder level by simply employing the keys of one user for decrypting the sequential traitor tracing layer and the keys of the other user for decrypting the broadcast encryption layer. In this attack scenario, the sequential traitor tracing scheme will successfully recover the identity of one of the traitors but subsequently revoking the recovered user will have absolutely no effect in the decryption capability of the pirate decoder (which will continue to operate due to the fact that it is using the broadcast encryption keys of the second uninidentified user).

It follows that any realistic solution for the problem of pirate rebroadcasts would require the design of a scheme that is capable of dealing with both tracing and revocation at the same time; failure of attaining these defensive properties simultaneously would make any scheme unusable in practice. The only known scheme offering both functionalities is implemented as part of the AACS [1] (and it also appears in [20]). The AACS is the current standard for content distribution that is employed in Blu-Ray and HD-DVD's. In this work we observe that the scheme employed in [1,20] has serious limitations (see below).

*Our Results.* In the present work we formally model the primitive of tracing and revoking pirate rebroadcasts and we present the **first constructions** that are capable of tracing and revoking an unlimited number of users. We also present a security and performance analysis of the only previously known scheme [20] and we find that the maximum number of revocations is bounded the receiver storage and the maximum traitor collusion that can be traced without false accusations is similarly bounded. Moreover, we provide a general design framework for tracing and revocation in the pirate rebroadcasting setting that permits a lot of flexibility in the choice of the basic parameters. The basic parameters of trace and revoking scheme for pirate rebroadasts is the *communication overhead* which is the amount of replication necessary in order to transmit a key, the *rebroadcast bound* which is the maximum number of transmissions a rebroadcasting pirate can "survive" before it is being entirely revoked in the system, and the *marking alphabet* which refers to the number of different variants of the content that the distribution center should create.

We present a number of instantiations of our design framework. Our first construction employs merely a binary marking alphabet and can withstand an

unlimited number of traitors and revocations. The communication overhead is additively linear in the number of revoked users $R$ and the number of traitors $t$. It follows that the communication overhead grows linearly in the number of malicious users; the constant hidden in the asymptotic notation is very small (it is $2R + 4t$ in the worst case). The pirate rebroadcast bound on the other hand is quadratic in the communication overhead and depends only logarithmically in the total number of users. This construction can thus keep on tracing and revoking arbitrary number of users with the only penalty of an extended rebroadcast bound as revocations accumulate. Our second construction also employs a binary alphabet and imposes a bound $w$ on the size of the maximum traitor coalition. This enables us to improve the maximum pirate rebroadcast bound to depend on *logarithmically* in the number of revoked users (while being polynomially bounded on $w$). Finally our third construction improves further on the rebroadcast bound by employing a larger marking alphabet size of $2t + 1$ where $t$ is the number of traitors; the bound is $O(t \log(N/t))$ and thus improves on the previously known convergence bound of $O(t \log N)$ for dynamic traitor tracing that was given by Fiat and Tassa [14] for the same alphabet size while offering revocation explicitly (while [14] offered tracing that relied on revocation without specifying how revocation is actually done).

*Comparison with the AACS.* The Advanced Access Content System (AACS) [1] is the current standard for content scrambling of Blu-Ray disks and HD-DVDs. It offers the functionality of a trace and revoke scheme (in fact it employs a variation of the subset-difference method of [29]) and it also offers a trace and revoking mechanism for pirate rebroadcasts (chapter 4 of [1], known as the "sequence key block") that was also detailed in [20]. While the subset-difference tracing and revocation mechanism is better understood much less attention has been given to the tracing and revocation for pirate rebroadcasts component of AACS. We perform an analysis based on the parameters suggested in [20] and we find that a maximum number of 85 revocations can be performed. Moreover, even if the parameters of [20] are variated to accomodate more revocations the scheme will suffer by a bound of the number of revocations that is limited by the number of stored keys in a receiver. Given that key assignment needs to be performed at system setup time this suggests that the approach taken in [20] has inherent limitations in terms of the number of revocations it allows. Moreover, the approach is also limited in terms of the maximum number of traitors that can be identified: the underlying traceability code suggested in [20] can identify at most 9 traitors. In fact, this is true even if a brute-force search is performed to try all possible coalitions (cf. section 4). Potentially this can be improved by a probabilistic analysis as suggested in [20] but in all cases once the number of traitors exceeds 9 the system has the potential to accuse innocent users. In contrast, the set of schemes we present here enjoy unlimited number of revocations and impose no bound on the number of traitors.

In our system, the device stores another layer of key system to process the key used to encrypt the watermarked content. In our setting, it is possible to have two devices with two different device keys, while at the same time, they

obtain the same variant of the content. This has additional advantages as follows: In the AACS setting, the devices should update their key materials after 85 revocations. In many settings such update is impossible, e.g. in a DVD player case where the device is sold as an hardware in the market. In our system there is no need for such an update since a modified transmission block can account for all revocations needed.

Finally, it is possible to implement our construction in the AACS context using only the keys that are already present in the devices due to the implementation of NNL [29] without using any extra key material as it would be required for implementing the sequence key block method of [20].

## 2 Trace and Revoke Schemes for Pirate Rebroadcasts

A trace and revoke scheme $\mathtt{T}$ for pirate rebroadcasts consists of four procedures $\langle \mathsf{Init}, \mathsf{Transmit}, \mathsf{Receive}, \mathsf{Revoke} \rangle$. The parameters of the scheme are $N$, the number of users, and $q$ the number of symbols in the transmission alphabet $\Sigma$. The scheme $\mathtt{T}$ is stateful and is parameterized by a set of states denoted by $\mathsf{States}$. We define the four procedures below.

- $\mathsf{Init}$. It is a probabilistic procedure that given $1^N$ it produces a set system $\langle \mathsf{N}, \mathcal{J}, \{\mathcal{I}_u\}_{u \in \mathsf{N}} \rangle$, where $\mathsf{N} = \{1, \ldots, N\}$, $\mathcal{J} \subseteq 2^{\mathsf{N}}$ and $\mathcal{I}_u \subseteq \mathcal{J}$ for all $u \in \mathsf{N}$ as well as an initial state $\sigma_0 \in \mathsf{States} \times 2^{\mathcal{J}}$.
- $\mathsf{Transmit}$. It is a probabilistic procedure that given a state $\sigma \in \mathsf{States} \times 2^{\mathcal{J}}$ and (optionally) a feedback symbol $f \in \{1, \ldots, q\}$ it produces a new state $\sigma' \in \mathsf{States} \times 2^{\mathcal{J}}$ and a subset of $\mathcal{J} \times \Sigma$.
- $\mathsf{Receive}$. It is a procedure that given $\mathcal{I}_u$ for some $u \in \mathsf{N}$ and a subset of $\mathcal{J} \times \Sigma$, it returns a symbol in $\Sigma$ or $\perp$.
- $\mathsf{Revoke}$. It is a procedure that given a state $\sigma \in \mathsf{States} \times 2^{\mathcal{J}}$ and a set $\mathsf{R} \subseteq \mathsf{N}$ it returns a new state $\sigma' \in \mathsf{States} \times 2^{\mathcal{J}}$.

*Intuition.* The four procedures in a trace and revoke scheme $\mathtt{T}$ for pirate rebroadcasts play the following role in an actual system instantiation. The $\mathsf{Init}$ procedure produces $\mathcal{J}$ which corresponds to the set of keys in the system and the sets $\mathcal{I}_u$ which determine the key assignment for each user $u$, i.e., each user $u \in \mathsf{N}$ will receive a set of keys corresponding to the set $\mathcal{I}_u$. It also produces the initial state $\sigma_0 = \langle \mathsf{state}_0, \mathcal{V}_0 \rangle$. The set $\mathcal{V}_0$ is the set of keys that are initially revoked in the system (typically $\mathcal{V}_0 = \emptyset$). The procedure $\mathsf{Transmit}$ possibly receives some feedback symbol $f \in \Sigma$ (originating from a pirate rebroadcast) and produces a subset $J$ of $\mathcal{J} \times \Sigma$ that determines the way that the content should be transmitted. In particular for each $(j, s) \in J$ the system will transmit the encryption under the key $j \in \mathcal{J}$ of a version of the content marked with symbol $s \in \Sigma$. The $\mathsf{Receive}$ procedure will produce an accessible marking symbol given the content transmission and the keys of the user (note that in an actual implementation this procedure involves the identification of the watermark produced by a decoder). Finally $\mathsf{Revoke}$ updates the state of the system taking into account the set of revoked users $\mathsf{R}$.

Next we define the correctness properties that are required from a trace and revoke scheme for pirate rebroadcasts.

**Definition 1.** ***Correctness.*** *For each of the four procedures of a trace and revoke scheme for pirate rebroadcasts we have a corresponding correctness property:*

Initialization Correctness. *For all* $\mathsf{R} \subseteq \mathsf{N}$ *it holds that there exist* $j_1, \ldots, j_v \in \mathcal{J}$ *for some* $v \in \mathbb{N}$ *such that* $\mathcal{I}_u \cap \{j_1, \ldots, j_v\} \neq \emptyset$ *for all* $u \in \mathsf{N} \setminus \mathsf{R}$.

Transmitting Correctness. *For any* $\sigma = \langle \mathsf{state}, \mathcal{V} \rangle$ *it holds that* $\mathsf{Transmit}(\sigma, f)$ *returns a state* $\sigma' = \langle \mathsf{state}', \mathcal{V}' \rangle$ *and a set* $J = \{(j_\ell, s_\ell)\}_{\ell=1}^{v} \in \mathcal{J} \times \Sigma$, *that satisfies the property* $\mathcal{I}_u \not\subseteq \mathcal{V}'$ *if and only if* $\exists j \in \mathcal{I}_u \setminus \mathcal{V}'$ *such that* $(j, s) \in J$ *for some* $s \in \Sigma$.

Receiving Correctness. *For any* $J = \{(j_\ell, o_\ell)\}_{\ell=1}^{v} \in \mathcal{J} \times \Sigma$, *it holds that* $\mathsf{Receive}(\mathcal{I}_u, J)$ *picks any element of the set* $\{s \in \Sigma \mid \exists (j, s) \in J$ *where* $j \in \mathcal{I}_u\}$, *or returns* $\perp$ *if this set is empty.*

Revocation Correctness. *For any* $\sigma = \langle \mathsf{state}, \mathcal{V} \rangle$ *it holds that* $\mathsf{Revoke}(\sigma, \mathsf{R})$ *returns a state* $\sigma' = \langle \mathsf{state}', \mathcal{V}' \rangle$ *such that* $\mathcal{I}_u \subseteq \mathcal{V}'$ *for all* $u \in \mathsf{R}$.

*Remarks on correctness.* The correctness definition for the initialization property ensures that for any set of revoked users $\mathsf{R}$ it holds that we can find a set of keys $j_1, \ldots, j_v$ such that a user that is not revoked (i.e., $u \in \mathsf{N} \setminus \mathsf{R}$) has at least one key among $j_1, \ldots, j_v$. This requirement can be relaxed to hold only with very high probability over all possible choices for set systems on $N$ users or it may be required to hold only for sets of revoked users that are bounded by some parameter $r$. Such relaxations may impact the system operation as they will hinder the exclusion of certain sets of users or introduce a failure probability in user revocation. Moreover, we note that the recovery of $j_1, \ldots, j_v$ given $\mathsf{R}$ should be done efficiently for a system to be useful in an applied setting.

The transmission correctness definition ensures that the subset $J$ that is selected by $\mathsf{Trasmit}$ will enable a user that holds at least one unrevoked key to recover at least one symbol from the transmission alphabet. We note that an unrevoked user may recover many such symbols. The receiving correctness property specifies that the $\mathsf{Receive}$ function should choose one of the transmission symbols that the user can recover from a transmission or return $\perp$ in case that no symbol is accessible to the user. This would happen in case when all keys $\mathcal{I}_u$ of a certain user $u$ have been included in the set $\mathcal{V}$ that holds the set of revoked keys in the state of the system. Finally, the revocation correctness given a set of users to be revoked it includes all their keys into the set of revoked keys $\mathcal{V}$.

Next we proceed to define the security aspects of a trace and revoke scheme against pirate rebroadcasts. We first define our notion of an adversary which is a pirate rebroadcast:

**Definition 2.** ***Pirate Rebroadcast.*** *A pirate rebroadcast of length* $n$ *for a scheme* $\mathsf{T}$ *starting at state* $\sigma_b$ *with respect to a set* $\mathcal{K} \subseteq \mathcal{J}$ *is a random variable* $\langle f_1, \ldots, f_n \rangle$ *over* $\Sigma^n$ *that is subject to the constraint for all* $i = 1, \ldots, n$, *there exists* $(j, s) \in J$ *such that* $s = f_i$ *and* $j \in \mathcal{K}$ *where* $(\sigma_{b+i}, J)$ *is distributed according to* $\mathsf{Transmit}(\sigma_{b+i-1}, f_{i-1})$ *and* $f_0 = \epsilon$.

In the above definition the set $\mathcal{K}$ is the set of keys that are available to the adversary. This may include the set $\mathcal{I}_u$ of all keys of a user $u$. A pirate rebroadcast consists of those symbols that are accessible to the adversary based on the way the system is choosing to transmit different symbols to subsets of users. Observe that any user in the system may produce pirate rebroadcasts of arbitrary length as long as its keys are not revoked (i.e., become part of the $\mathcal{V}$ set inside the system state).

Now we define the security property that needs to be satisfied by a trace and revoke scheme for pirate rebroadcasts. It states that any coalition of malicious users (or traitors) can only produce pirate rebroadcasts of a *bounded length* with high probability. This effectively means that the scheme is capable of identifying the source of a rebroadcast and over a number of transmissions eliminate it. The security property will be in the form of a bound $\mu$ that will specify the maximum number of transmissions a traitor coalition can withstand. The bound $\mu$ will be a function of the number of users $N$, the number of traitors $t$ and the number of already revoked users $R$.

**Definition 3.** ***Traceability of Pirate Rebroadcasts.*** *We say that a scheme* $\mathtt{T}$ *satisfies* $(\mu, w)$*-traceability against pirate rebroadcasts with probability* $\epsilon$ *provided that the following holds: for any set of* $t$ *traitors* $\mathsf{T} \subseteq \mathsf{N}$ *with* $t \leq w$ *and any set of revoked users* $\mathsf{R}$ *it holds that, if* $B$ *is the length of any pirate rebroadcast that starts at state* $\sigma$ *with respect to the set of keys* $\mathcal{K} = \cup_{u \in \mathsf{T}} \mathcal{K}_u$*, then* $\mathbf{Pro}[B \leq \mu] \geq 1 - \epsilon$*. The probability distribution is taken over all states* $\sigma$ *distributed according to* $\mathsf{Revoke}(\sigma_0, \mathsf{R})$ *where* $\sigma_0$ *is the initial state of the scheme as produced by* $\mathsf{Init}$*. The parameter* $\mu$ *depends on* $N, t, R, \log(\frac{1}{\epsilon})$*.*

We define $\mu$-full-traceability against pirate rebroadcasts when there is no bound $w$ in the number of traitors. We remark that our tracing and revoking schemes for pirate rebroadcasts do not mandate the identification of the traitor users. This is because our aim is to eliminate any pirate rebroadcast that our system is given as feedback even if this rebroadcast does not uniquely identify a traitor. Observe that eventually, if the pirate keeps using the traitor key material available to it, all traitors will be revoked and hence they will be identified (but of course a pirate may stop rebroadcasting before that). This analysis approach is in line with the trace & revoke schemes of [29].

**Communication Overhead.** The communication overhead $\psi$ of a scheme $\mathtt{T}$ is the amount of replication the scheme employs in order to trace the rebroadcasts. In order to measure the impact of both revocation and tracing on the communication overhead we will consider the case of a pirate rebroadcast generated by $t$ users that occurs after the revocation of $R$ users. In particular the communication overhead $\psi$ of the scheme $\mathtt{T}$ will be a function of $N, t, R$ that bounds from above the size of all sets $J$ that are produced by $\mathsf{Transmit}$ following any pirate rebroadcast with respect to the set of the keys of any $t$ traitors that start at state $\sigma$ where $\sigma \leftarrow \mathsf{Revoke}(\sigma_0, \mathsf{R})$ and $\mathsf{R}$ is any set of $R$ users.

## 3 Our Construction

**Preliminaries.** A subset cover scheme $SCS = (\mathsf{N}, \mathcal{J}, \mathsf{Cover}(\cdot), \mathsf{Split}(\cdot,\cdot))$ is a class of combinatorial design introduced in [29] that can be used for constructing key revocation methods. Note that $\mathsf{N} = \{1,\ldots,N\}$, $\mathcal{J} \subseteq 2^{\mathsf{N}}$. $Cover(\cdot)$ is a function that given a set of users $\mathsf{R} \subseteq \mathsf{N}$, it outputs a collection of subsets $\{\mathsf{S}_{i_1},\ldots,\mathsf{S}_{i_v}\} \subseteq \mathcal{J}$, that is called a "broadcast pattern" or simply pattern and denoted by $\mathcal{P}$ such that $\mathsf{N}\setminus\mathsf{R} = \bigcup_{j=1}^{v} \mathsf{S}_{i_j}$. All subsets in $\mathsf{Cover}(\mathsf{N}\setminus\mathsf{R})$ are disjoint. $Split(\cdot,\cdot)$ is a function that given a broadcast pattern $\mathcal{P} = \{\mathsf{S}_{i_1},\ldots,\mathsf{S}_{i_v}\}$ and a set of disjoint subsets $\mathcal{T} \subseteq \mathcal{J}$ it splits each subset of $\mathcal{P} \cap \mathcal{T}$ evenly (based on the "bifurcation property" of [29]) and returns an updated broadcast pattern that is derived from $\mathcal{P}$ by replacing the subsets $\mathcal{P} \cap \mathcal{T}$ with their splittings. If a subset in $\mathcal{P} \cap \mathcal{T}$ cannot be split it will simply be removed by $Split(\cdot,\cdot)$.

A fingerprinting code is a pair of algorithms $(\mathsf{CodeGen}, \mathsf{Tracing})$ that is defined as follows: $CodeGen$ is a probabilistic algorithm that is given input $(n, \nu, w, q)$ where $\nu = \log(\frac{1}{\epsilon})$ and $\epsilon$ is a security parameter, and it outputs a code $\mathcal{C}$ of $n$ codewords over $\Sigma^{\ell}$ where $|\Sigma| = q$ (we refer to such codes as $(\ell, n, q)$-codes) as well as a tracing key $tk$. $\mathsf{Tracing}$ is an algorithm that, informally, if $c$ is constructed by a traitor coalition of size at most $w$ by combining their codewords, it identifies at least one of the traitors with high probability. The fingerprinting code is called "open" if $tk$ is empty.

**The construction.** We first describe our construction at a high level. In the initialization stage, we define the keys of all users based on a subset cover sheme. Upon detecting a pirate rebroadcast we will take advantage of the splitting property of the subset cover scheme and derive a pattern covering the active users that includes two or more subsets. Each subset in the pattern will be assigned potentially different symbols following a fingerprinting code that will be selected on the fly. After a sufficient number of transmissions (that matches the length of the code) the sequence of feedback symbols will form a pirate codeword and by applying the tracing algorithm of the fingerprinting code on it we will identify a subset that contains some traitors. This process will be repeated recursively until all the traitors are identified or the rebroadcast ceases. An illustration of this process is presented in figure 2.

We next describe our construction in more detail. We define the set of states of our scheme first: a state is a pair $(\mathsf{state}, \mathcal{V})$ such that (i) $\mathsf{state} \in \mathsf{States}$ consists of a pattern $\mathcal{P} \subseteq \mathcal{J}$ of keys, an instance of a fingerprinting code $(\mathsf{CodeGen}, \mathsf{Tracing})$ and a message transmission index $m$, and (ii) $\mathcal{V} \subseteq \mathcal{J}$ such that the following holds: $u$ is such that $\mathcal{I}_u \not\subseteq \mathcal{V}$ if and only if $(\mathcal{I}_u \cap \mathcal{P}) \setminus \mathcal{V} \neq \emptyset$. Intuitively, $\mathcal{V}$ contains the keys of all revoked users and $\mathcal{P}$ is a set of disjoint subsets whose corresponding keys enable the transmission of content to the users who are not revoked.

- $\mathsf{Init}$. Given $1^N$, it produces a subset cover scheme $SCS = (\mathsf{N}, \mathcal{J}, \mathsf{Cover}(\cdot), \mathsf{Split}(\cdot,\cdot))$ which defines the set system $\langle \mathsf{N}, \mathcal{J}, \{\mathcal{I}_u\}_{u\in\mathsf{N}}\rangle$ by setting $\mathcal{I}_u$ to contain all $\mathsf{S} \in \mathcal{J}$ such that $u \in \mathsf{S}$. State $\sigma_0 = \langle \mathsf{state}_0, \mathcal{V}_0\rangle$ is initialized as follows: $\mathcal{V}_0 = \emptyset$ and $\mathsf{state}_0$ consists of the triple $(\mathcal{P}, FC, 0)$ that is selected as

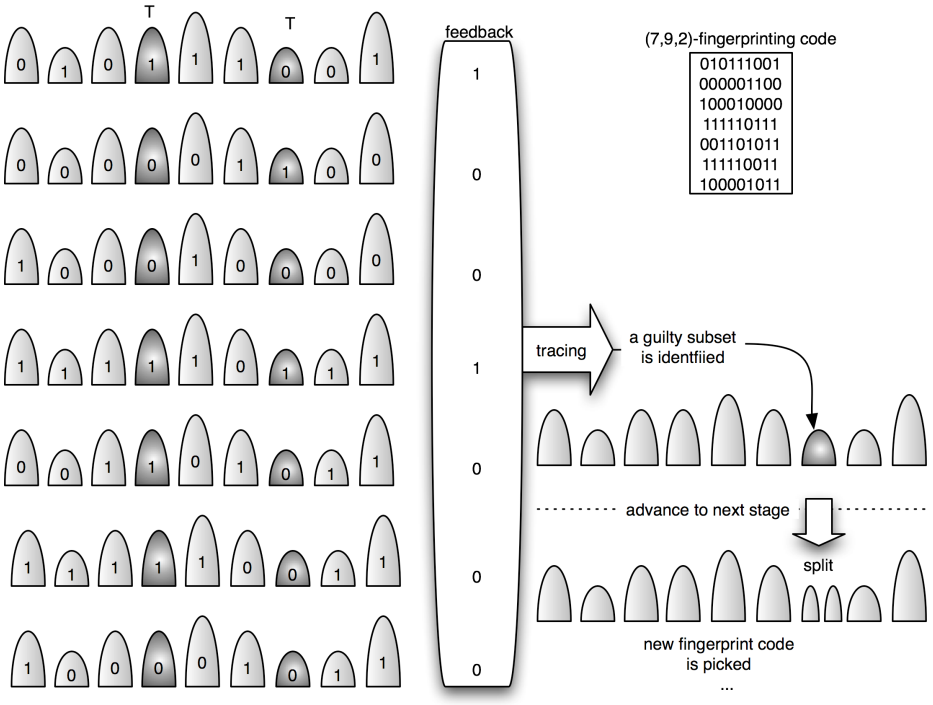


**Fig. 2.** Illustration of tracing and revoking a pirate rebroadcast with our construction

follows (i) $\mathcal{P} = \mathsf{Cover}(\mathsf{N})$, (ii) $FC \leftarrow \mathsf{CodeGen}(|\mathcal{P}|, \nu, w, q)$, i.e., $FC = (\mathcal{C}, tk)$ where $\mathcal{C}$ is a $(\ell, |\mathcal{P}|, q)$-code and $tk$ is the corresponding tracing key. Note that each key index $\mathsf{S}_j \in \mathcal{P}$ is associated to a unique codeword $y_j \in \mathcal{C}$ for $j = 1, \ldots, |\mathcal{P}|$.

– Transmit. Given the current state of the system $\sigma_{p-1} = \langle \mathtt{state}_{p-1}, \mathcal{V}_{p-1} \rangle$ and a feedback symbol $f \in \Sigma$, the system state is first updated to $\sigma_p = \langle \mathtt{state}_p, \mathcal{V}_p \rangle$. The update of the system is done as follows: the previous message transmission index $m$ and the set of keys $\mathcal{P}$ are pulled out from $\sigma_{p-1}$. If $m < \ell$ where $\ell$ is the length of the code $\mathcal{C} = (\ell, |\mathcal{P}|, q)$ in $\mathsf{state}_{p-1}$ then $m$ is increased by one and the feedback symbol $f$ is stored. Otherwise (if $m = \ell$), we need to update the broadcast pattern $\mathcal{P}$. This is done as follows: the feedback values of all the $\ell$ recent transmissions are used to define a codeword $a \in \Sigma^\ell$ and then a set of subsets $T \subseteq \mathcal{P}$ is identified as follows: we compute $B = \mathsf{Tracing}(a, tk)$ and define $T$ as $\mathsf{S}_j \in T$ iff $y_j \in B$ (here we use the 1-1 correspondence between the pattern subsets and codewords in $\mathcal{C}$). Then the broadcast pattern is updated by calling $\mathcal{P}' = \mathit{Split}(\mathcal{P}, T)$. A new fingerprint code $\mathcal{C}'$ should be sampled now to support as many codewords as the size of new broadcast pattern $\mathcal{P}'$ using $\mathsf{CodeGen}$ as described in the initialization step. The message transmission index $m$ is set to 1. $\mathcal{V}_p$ is set to $\mathcal{V}_{p-1} \cup \{\mathcal{I}_u \mid \exists \mathsf{S}_j \in T \text{ where } \mathsf{S}_j = \{u\}\}$. This completes the description of

the state update operation. After the state update the transmission function proceeds to select the set $J \subseteq \mathcal{J} \times \Sigma$.
This is done as follows: the triple $(\mathcal{P}, FC, m)$ is pulled out from $\sigma_p$. Then the subset $J$ is defined to include all pairs $(\mathsf{S}_j, y_j[m])$ for $j = 1, 2, \ldots |\mathcal{P}|$ where $y_j[m]$ denotes the $m$-th symbol of the codeword $y_j \in \mathcal{C}$.

- Receive. Given $\mathcal{I}_u$ for some $u \in \mathsf{N}$ and $J \subseteq \mathcal{J} \times \Sigma$, the procedure finds a pair $(j, s) \in J$ such that $j \in \mathcal{I}_u$ and returns $s$. If no such pair exists it returns $\perp$.
- Revoke. Given the current state $\sigma_{p-1} = \langle \mathsf{state}_{p-1}, \mathcal{V}_{p-1} \rangle$ and a set $\mathsf{R}$, a new pattern $\mathcal{P}$ is selected as $\mathsf{Cover}(\mathsf{N} \setminus (\mathsf{R} \cup \mathsf{R}_{p-1})$ where $\mathsf{R}_{p-1} = \{u \mid \mathcal{I}_u \subseteq \mathcal{V}_{p-1}\}$. Subsequently, a new state $\mathsf{state}_p$ is formed by selecting a new fingerprinting code $FC \leftarrow \mathsf{CodeGen}(|\mathcal{P}|, \nu, w, q)$. The procedure returns $\langle \mathsf{state}_p, \mathcal{V}_p \rangle$ where $\mathsf{state}_p = (\mathcal{P}, FC, 0)$ and $\mathcal{V}_p = \mathcal{V}_{p-1} \cup (\cup_{u \in \mathsf{R}} \mathcal{I}_u)$.

**Proposition 1.** *The construction presented above satisfies correctness according to definition 1.*

**Analysis of the construction.** We next analyze the efficiency and security parameters of our construction. We first should examine in more depth the way the pattern updating algorithm $Split(\cdot, \cdot)$ operates. Without loss of generality we will instantiate our construction using for the underlying subset-cover scheme $SCS$ the Subset-Difference method of [29]. The analysis is similar if another subset-cover scheme is being used such as those of [2,15,18,19,41]. We prove the following regarding the efficiency parameters of our construction:

**Theorem 1.** *The communication overhead $\psi$ of our construction starting at state $\sigma \leftarrow \mathsf{Revoke}(\sigma_0, \mathsf{R})$ where $\sigma_0$ is the initial state of the scheme as produced by* Init *satisfies $\psi \leq 2|\mathsf{R}| + 4t$ where $t$ is the number of traitors.*

For an illustration of how our construction works in combination with the subset-difference method we refer to figure 3 in the appendix. Next we present the analysis of the pirate rebroadcast bound for our construction.

**Picking a Fingerprinting Code.** The choice of the underlying fingerprinting code is flexible. It is possible to pick totally different codes in each stage (after a subset has been identified as containing a traitor) or keep the same code throughout. Moreover, this choice will be reflected in the deciphering process within the content transmission, hence the choice of fingerprinting code is independent from the keys stored in the device. The code is used to simply restructure the marking-assignment logically, by reassigning a subset to a new codeword.

A crucial difference regarding the selection of the fingerprinting code in our setting when compared to previous mechanisms that are employing some traceability or fingerprinting code, is that in our setting we only need codes with a number of codewords proportional to the number of revoked users and active traitors as opposed to the whole population. In contrast, previous works strived to produce codes with a number of codewords equal to the population size given a fixed small number of revoked users or traitors. Due to this important fact, we are able to employ fingerprinting codes that allow for arbitrary traitor collusions such as those presented in [6,40] without hurting the efficiency of our

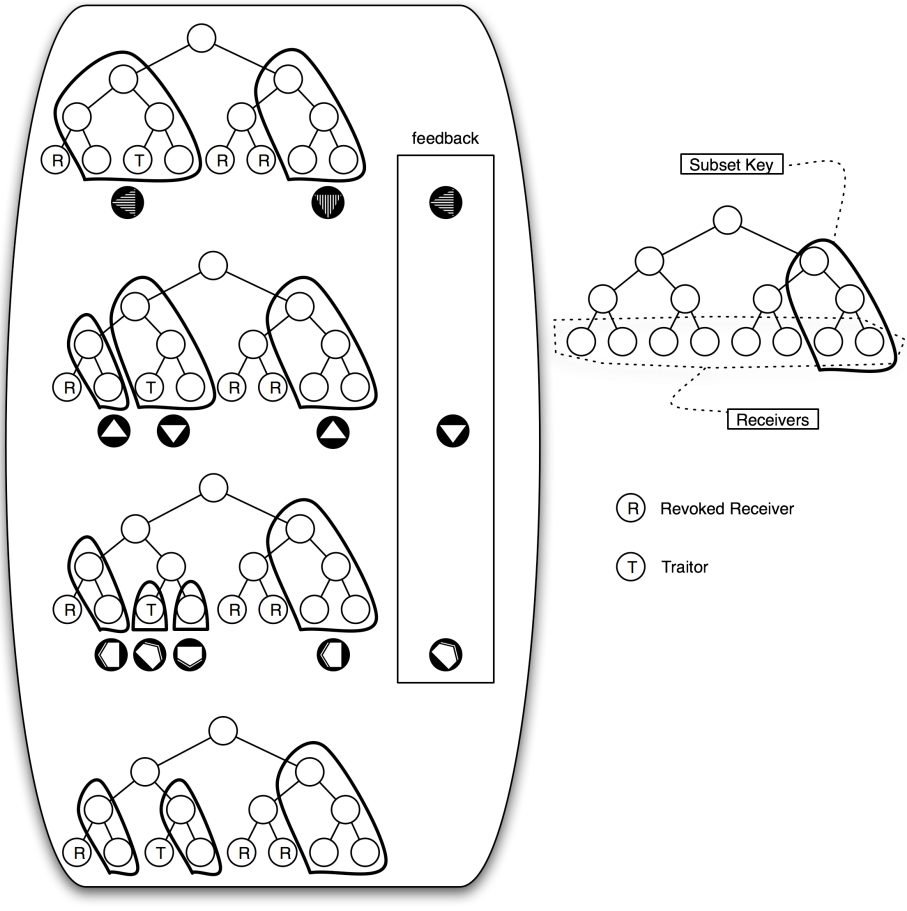


**Fig. 3.** Depiction of tracing a traitor following a pirate rebroadcast it produces using our construction (while employing the Subset-Difference method for key assignment)

construction and thus we can trace and revoke an unlimited number of traitors. We note that "picking a code" is not a computationally intensive operation as the codes we need can be sampled very efficiently or can be available in the form of a codebook and it is an operation that happens at the center and does not affect the complexity of the honest devices in any way.

**Traceability of Pirate Rebroadcasts.** The Tracing algorithm over the code $\mathcal{C}$ that is employed in the Transmit function will identify a subset containing a traitor with high probability. This is because of the fact that the codewords of $\mathcal{C}$ are assigned to subsets of devices, i.e., the detection of a "traitor" from the Tracing algorithm is now equivalent to finding a subset that contains a traitor. Once such a subset is found, this subset will be split into 2 subsets taking advantage of the *Split* function. The updated set of users, i.e the subsets in the new partition, will be reassigned new codewords from possibly a fresh fingerprinting code. Observe that the

Tracing algorithm of the fingerprinting code substitutes the "walking" argument that was employed in previous traitor tracing schemes (cf. [8,4,23,29]) that progressively randomized the pattern ciphertext till a position is identified that the pirate-box fails to decrypt successfully. This mechanism was the the basic procedure which tests a pirate box that is successful in decrypting with a given pattern by using some special tracing ciphertexts to output a subset containing a traitor. We note that such walking arguments cannot be used in our setting since they would require transmitting "garbage" to a subset of legitimate receivers, something that is unacceptable in the content distribution setting. This problem is not present in the previous works mentioned since they assume that they can analyze the pirate decoder in isolation from the transmission system (something impossible in the stronger adversarial setting we consider here).

We next analyze the convergence of our construction, i.e., the number of transmissions required to eliminate a pirate rebroadcast. At any system state $\sigma_p$ with a set of revoked users $\mathsf{R}$ with $R = |\mathsf{R}|$ and a set of $t$ subsets known to contain traitors, the number of subsets in the broadcast pattern covering the enabled set of users will be a function of $R$ and $t$. Assuming that Subset Difference method of [29] is used, the size of broadcast pattern would be at most $2R + 4t$ as shown in Theorem 1. The pirate rebroadcast bound would depend on the tracing algorithm over the code $\mathcal{C}$ of size $O(R + t)$. We observe that it will require $O(\log N)$ stages to identify at least one traitor as this is the height of the Subset-Difference tree. Based on this we show the following:

**Theorem 2.** *Consider a set of traitors* $\mathsf{T}$ *with* $|\mathsf{T}| = t$*, a set of revoked users* $\mathsf{R}$ *with* $|\mathsf{R}| = R$*. If* $\sigma$ *is a state distributed according to* $\mathsf{Revoke}(\sigma_0, \mathsf{R})$ *then the length of any pirate rebroadcast starting at* $\sigma$ *is* $O(\ell \cdot t \cdot \log N)$ *with probability* $1 - t \cdot \log N \cdot \epsilon'$ *where* $\ell$ *is the length of the fingerprinting code used to instantiate the scheme and* $\epsilon'$ *the failure probability of the associated* Tracing *algorithm.*

Note that the dependency of $\mu$ in $R$ is through the fingerprinting code length $\ell$. Moreover, if there is a bound $w$ on the number of traitors (i.e., $t \leq w$) this parameter will also appear as a function of $\ell$. The proof is straight forward, length $\ell$ is required to detect at least one subset that contains a traitor. Identification of a single traitor requires at most $\log N$ new code selections. Identification of all traitors will then yield the bound given in the theorem. The actual pirate rebroadcast bound $\mu$ will depend on the choice of the code.

(**Instantiation 1**). In our first instantiation of the framework we use the optimal codes of [40] in conjunction to the subset-difference subset cover set system of [29]. This provides for a communication overhead $\psi = O(R+t)$ due to theorem 1 and a rebroadcast bound $\mu = O(t(R + t)^2 \log N \cdot \log((R + t)t \log N\epsilon^{-1}))$, where $R$ is the number of revoked users, $t$ the number of traitors, $N$ the number of users and $\epsilon$ the error probability. The bound follows from theorem 2 and the fact that the length of Tardos' codes is $O(n^2 \log(n/\epsilon))$ where $n$ is the number of codewords (that in our setting matches the communication overhead). Note that this scheme tolerates an *unlimited* number of traitors and revocations.

(**Instantiation 2**). Our second instantiation employs again Tardos' codes but assuming an upper bound on the number of traitors $w$. This provides for code length of $O(w^2 \log n/\epsilon)$ and given that in our setting we have that $n$ is the number of codewords that should be equal to the communication overhead using theorem 2 we obtain a rebroadcast bound of $O(tw^2(\log N)\log((R+t)t\log N\epsilon^{-1})))$, i.e., with only logarithmic dependency on the number of revocations in the system.

(**Instantiation 3**). In our third instantiation we will take advantage of an increasing marking alphabet (instead of binary as in the previous two constructions). This will enable a very short rebroadcast bound of $O(t\log(N/t))$. We will use the complete subtree method of [29] to instantiate the subset cover system (that is superior for our purpose compared to the subset difference method). Recall that in the complete subtree method users are aligned as the leaves of a complete binary tree and the set system defines a key for any complete binary subtree of the total tree. Instead of relying on a fixed fingerprinting code to perform tracing as in the previous two constructions we will take advantage of our larger alphabet of $2t+1$ where $t$ is the number of traitors to assign different versions to all subsets that result from the $Split(\cdot,\cdot)$ of the underlying subset cover scheme. We observe that there are at most $2t$ subsets that are formed after splitting in any pattern at any step of the tracing process. This means that we can use a $2t+1$ symbol alphabet. Given that the number of steps required to trace all $t$ traitors equals the number of nodes in the Steiner tree of the $t$ leaves that correspond to the traitors we conclude that the maximum pirate rebroadcast length is $O(t\cdot\log(N/t))$. This analysis not only improves on the previously known construction of Fiat and Tassa [14] that achieves $O(t\cdot\log N)$, but also it has an explicit description about how the revocation is performed.

## 4 Tracing Pirate Rebroadcasts in the AACS

In this section we describe[2] the part of AACS specifications [1] that deals with tracing pirate rebroadcasts (also published in [20]). We recast the description of that scheme in our terminology to facilitate the comparison with our framework. During the initialization of the system a sequence of partitions of the user population is formed. The partitions are determined according to the a fingerprinting code, specifically Reed-Solomon Code in the descriptions. After a sufficient number of transmissions (that matches the length of the code) the sequence of feedback symbols will form a pirate codeword and by applying the tracing algorithm of the fingerprinting code we will identify a traitor and revoke all of its keys. Note that revocation of some keys will effect the future feedbacks since the partitions remain unchanged and hence the traceability of the

[2] We note that this description may not necessarily reflect entirely all the details of the actual implementation of the AACS as many details are hidden, obfuscated or omitted in the references [1,20]. Still, we are confident that all the facts that we present about their construction are valid.

scheme will be harder in future transmissions which we will take advantage in our analysis later.

Formally, the construction of [20] is as follows: we define first a partition space $\mathsf{P}_{S,k}$ as a set of $k$ disjoint subsets of $S$ whose union yields the set $S$, i.e. $\{\mathsf{S}_1, \ldots, \mathsf{S}_k\} \in \mathsf{P}_{S,k}$ iff $\cup_{i=1}^{k} \mathsf{S}_k = S$ and $\mathsf{S}_i \cap \mathsf{S}_j = \emptyset$ for $i \neq j$. The state of the scheme $\mathsf{state} \in \mathsf{States}$ stores the history of feedback values from the pirate rebroadcast.

- Init. Given the set of devices $\mathsf{N} = \{1, 2, \ldots, 256^4\}$, it produces a set system $\langle \mathsf{N}, \mathcal{J}, \{\mathcal{I}_u\}_{u \in \mathsf{N}} \rangle$ where $\mathcal{J}$ consists of the subsets of 255 different partitions in $\mathsf{P}_{\mathsf{N},256}$. Denoting the $j$-th partition by $v_j$, the subsets of $\mathcal{J}$ are represented by $\mathsf{S}_{i,j}$ for $0 < i \leq 256$ and $0 < j < 256$ where $v_j = \{\mathsf{S}_{1,j}, \mathsf{S}_{2,j}, \ldots, \mathsf{S}_{256,j}\}$. The selection of partitions is done by using a Reed-Solomon code $\mathcal{C}$ with an alphabet $\Sigma = \{1, \ldots 256\}$ and of length 255. According to the the specifications, $\mathcal{C}$ is defined over polynomials of degree 3 in the finite field $\mathbb{F}_{256}$. Each receiver $u \in \mathsf{N}$ is assigned a codeword $y \in \mathcal{C}$ so that $\mathcal{I}_u = \{\mathsf{S}_{y[1],1}, \mathsf{S}_{y[2],2}, \ldots, \mathsf{S}_{y[255],255}\}$ where $y[j]$ denotes the $j$-th symbol of the codeword $y$. This will set $\mathsf{S}_{i,j} = \{u \in \mathsf{N} \mid u \text{ is assigned } y \in \mathcal{C}, y[j] = i\}$. $\sigma_0 = \langle \mathsf{state}_0, \mathcal{V}_0 \rangle$ is initialized such that both $\mathsf{state}_0$ and $\mathcal{V}_0$ being emptyset.
- Transmit. Given the current state of the system $\sigma_{p-1} = \langle \mathsf{state}_{p-1}, \mathcal{V}_{p-1} \rangle$ and a feedback value $f \in \{1, \ldots, 256\}$, the system state is first updated to $\sigma_p = \langle \mathtt{state}_p, \mathcal{V}_p \rangle$. The update of the state is done as follows: first $\mathsf{state}$ is updated to include feedback value $f$. If the sequence of feedback values stored in $\mathsf{state}_p$ makes it possible to identify a user $u \in \mathsf{N}$ as a traitor, then its keys are added to $\mathcal{V}_{p-1}$ to obtain $\mathcal{V}_p$, i.e. $\mathcal{V}_p = \mathcal{V}_{p-1} \cup \mathcal{I}_u$. The way the traitor detection is performed using the history of feedback values is explained in [20]. After state update, the transmission function proceeds to select the set $J \subseteq \mathcal{J} \times \Sigma$.

  This is done as follows: A set of partitions $\{v_{j_1}, \ldots v_{j_r}\} \subseteq \{1, \ldots, 255\}$ is chosen so that for any enabled device $u$, $(\mathcal{I}_u \setminus \mathcal{V}_p) \cap \{\mathsf{S}_{i,j_m} \mid 0 < m \leq r, 0 < i \leq 256\} \neq \emptyset$ holds. Note that the way the partitions are selected is not specified in [20] (presumably a heuristic can be used given that it is easy to test whether a given set of partitions satisfies the constraint or not[3]). We also remark that it is possible that such set of partitions does not exist, in which case the Transmit procedure will fail (i.e. the encryption phase put on top of Transmit procedure will exclude some honest devices from the transmission of actual content). We take advantage of this later in our analysis.
  The subset $J$ is defined to include all pairs $(\mathsf{S}_{i,j}, i)$ where $\mathsf{S}_{i,j} \in \cup_{m=1}^{r} v_m \setminus \mathcal{V}_p$
- Receive. Given $\mathcal{I}_u$ for some $u \in \mathsf{N}$ and $J \subseteq \mathcal{J} \times \Sigma$, the procedure finds a pair $(\mathsf{S}_{i,j}, s) \in J$ such that (1) $\mathsf{S}_{i,j} \in \mathcal{I}_u$ and (2) $j$ is minimal among all subsets intersecting with $\mathcal{I}_u$ and the procedure returns $s$. If no such pair exists it returns $\perp$.

[3] Specifically in [20], it is stated(here columns refer to the partitions) "*However, after some number of columns depending on the actual number of compromised keys, the AACS licensing agency will know that only compromised devices would be getting the link key; all innocent would have found the output key in this column or in a previous column.*"

– Revoke. Given the current state $\sigma_{p-1} = \langle \mathsf{state}_{p-1}, \mathcal{V}_{p-1} \rangle$ the procedure returns $\langle \mathsf{state}_p, \mathcal{V}_p \rangle$ where $\mathsf{state}_p = \emptyset$ and $\mathcal{V}_p = \mathcal{V}_{p-1} \cup (\cup_{u \in \mathsf{R}} \mathcal{I}_u)$.

**Plaintext Preprocessing and Marking.** As seen above for a certain movie $m$ the transmission center needs to produce 256 variations $\{m_1, \ldots, m_{256}\}$. A marking scheme of only 16 variations is being used though. For this reason a second (inner) Reed Solomon code $\mathcal{C}'$ is used over an alphabet $Q' = \{1, 2, \ldots 16\}$ that has length 15 and is defined over linear polynomials in the field $\mathbb{F}_{16}$. Thus, there are $16^2 = 256$ codewords in $\mathcal{C}'$, and each codeword in $\mathcal{C}'$ is a vector $\langle b_1, \ldots, b_{15} \rangle$. In order to transmit a movie $m$, the movie is split into 15 segments and the marking process is applied to each segment resulting in 16 different variations; let $s_{e,l}$ denote the $e$-th variation of the $l$-th segment, $l = 1, \ldots, 15$ and $e = 1, \ldots, 16$. Subsequently 256 versions of the movie are formed by employing the code $\mathcal{C}'$. In particular the $i$-th version of the movie $m$ would be the string $s_{w_1,1} \ldots s_{w_{15},15}$ where $\langle w_1, \ldots, w_{15} \rangle$ is the $i$-th codeword of $\mathcal{C}'$. Note that this complicates somewhat the traceability analysis since the inner code is a 3-TA as opposed to a 9-TA that is employed for key assignment. Effectively this violates the marking assumption since it is possible for a coalition of size 4 to produce a movie for which the identification algorithm may not be able to identify any of its members. Without loss of generality we will ignore this issue for the moment and we will assume a "best case" behavior of their scheme for the sake of comparison. We refer to [20] for further discussion on this issue.

**Revocation.** The scheme of [20] as presented above has the capability to revoke a given set of users by selecting appropriately the set of columns that are used in the transmission. Here we observe that the revocation capability of the scheme is very limited. For the suggested parameters as described above we have the following:

**Fact 1.** *The scheme of [20] as presented above can support at most* 85 *revocations.*

To see why the above is true consider that key assignment is based on a Reed-Solomon code and during revocation a set of columns needs to be selected so that no innocent user is covered entirely by the set of keys assigned to the revoked users. It follows that a coalition capable of framing a user in the 9-TA code would cause the system to produce transmissions that disable some innocent users. It is easy to see that a coalition of 85 users may frame an innocent user in the system (this is because $3 \cdot 85 \geq 255$ where the coefficient 3 stems from the fact that this is the maximum number of locations that two codewords can agree in this code).

**Traceability of Pirate rebroadcasts in AACS.** In [20] it is argued based on a probabilistic analysis that a coalition of 9 traitors can be traced in 56 movie transmissions. We note that this analysis is performed in a setting without any revocations. Not only tracing after some number of revocations differs from the case when there is no revocation, it doesn't have a trivial solution. Morever, tracing after revocation will severely hurts the efficiency of the transmission overhead. On top of those limitations we observe the following on coalition bound:

**Fact 2.** *The scheme of [20] as presented may disable innocent users in case of a traitor coalition of a size larger than* 9 *occurs.*

To see why this is the case, consider that in the 9-TA Reed-Solomon code employed, it holds that there exist a coalition $C_1$ of size 10 that can produce a pirate codeword for which there exists an innocent user $u \notin C$ such that $u$'s assigned codeword has maximum overlap with pirate codeword (strictly higher than any of the users in $C_1$). Note that any correct tracing algorithm would accuse the users that have very high overlap with the pirate codeword; hence no matter how the tracing algorithm of [20] operates the user $u$ will be a likely outcome and hence this suggests that the revocation algorithm will be incapable of revoking the correct set of users (i.e., in some cases innocent users may be disabled from the system).

**A Comparison of AACS with Our Constructions.** Even if the above construction is variated over a different parameter selection and different codes the net effect would be the following: given the way the revocation algorithm works, the number of revocations will never exceed the length of the code employed (in fact they would be much less as illustrated above). Given that the code length selected in AACS [1] is 255 this suggests that the number of revocations feasible by this scheme is very limited. Even worse, as it is also pointed out in [20] as revocations accumulate the traceability of the underlying construction gets substantially reduced. In contrast our constructions enable an unlimited number of revocations against an unlimited number of traitors without incurring any degradation in security as the number of revocations accumulate. Moreover, given that the key assignment in our construction is based only on the subset-cover framework our tracing and revoking schemes for pirate rebroadcasts can be applied readily in the context of AACS (that employs a subset-cover key assignment already but used only for regular trace and revoking in the clone-decoder attack setting).

## References

1. AACS Specifications specifications (2006), `http://www.aacsla.com/`
2. Attrapadung, N., Imai, H.: Graph-Decomposition-Based Frameworks for Subset-Cover Broadcast Encryption and Efficient Instantiations. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 100–120. Springer, Heidelberg (2005)
3. Berkman, O., Parnas, M., Sgall, J.: Efficient dynamic traitor tracing. In: SODA 2000, pp. 586–595 (2000)
4. Boneh, D., Franklin, M.: An Efficient Public-Key Traitor Tracing Scheme. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 338–353. Springer, Heidelberg (1999)
5. Boneh, D., Sahai, A., Waters, B.: Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
6. Boneh, D., Shaw, J.: Collusion-Secure Fingerprinting for Digital Data. IEEE Transactions on Information Theory 44(5), 1897–1905 (1998)

7. Chor, B., Fiat, A., Naor, M.: Tracing Traitors. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 257–270. Springer, Heidelberg (1994)
8. Chor, B., Fiat, A., Naor, M., Pinkas, B.: Tracing Traitors. IEEE Transactions on Information Theory 46(3), 893–910 (2000)
9. Chabanne, H., Hieu Phan, D., Pointcheval, D.: Public Traceability in Traitor Tracing Schemes. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 542–558. Springer, Heidelberg (2005)
10. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing 6(12), 1673–1687 (1997)
11. Dodis, Y., Fazio, N.: Public Key Broadcast Encryption for Stateless Receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003)
12. Dodis, Y., Fazio, N., Kiayias, A., Yung, M.: Scalable public-key tracing and revoking. In: PODC 2003, Proceedings of the Twenty-Second ACM Symposium on Principles of Distributed Computing (PODC 2003), Boston, Massachusetts, July 13-16 (2003)
13. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
14. Fiat, A., Tassa, T.: Dynamic Traitor Tracing. Journal of Cryptology 4(3), 211–223 (2001)
15. Gentry, C., Ramzan, Z., Woodruff, D.P.: Explicit Exclusive Set Systems with Applications to Broadcast Encryption. In: FOCS 2006, pp. 27–38 (2006)
16. Gafni, E., Staddon, J., Lisa Yin, Y.: Efficient Methods for Integrating Traceability and Broadcast Encryption. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 372–387. Springer, Heidelberg (1999)
17. Garay, J.A., Staddon, J., Wool, A.: Long-Lived Broadcast Encryption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 333–352. Springer, Heidelberg (2000)
18. Goodrich, M.T., Sun, J.Z., Tamassia, R.: Efficient Tree-Based Revocation in Groups of Low-State Devices. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 511–527. Springer, Heidelberg (2004)
19. Halevy, D., Shamir, A.: The LSD Broadcast Encryption Scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
20. Jin, H., Lotspiech, J.: Renewable Traitor Tracing: A Trace-Revoke-Trace System For Anonymous Attack. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 563–577. Springer, Heidelberg (2007)
21. Jin, H., Lotspiech, J., Nusser, S.: Traitor tracing for prerecorded and recordable media. In: Digital Rights Management Workshop, pp. 83–90 (2004)
22. Kiayias, A., Yung, M.: Self Protecting Pirates and Black-Box Traitor Tracing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 63–79. Springer, Heidelberg (2001)
23. Kiayias, A., Yung, M.: On Crafty Pirates and Foxy Tracers. In: Sander, T. (ed.) DRM 2001. LNCS, vol. 2320, pp. 22–39. Springer, Heidelberg (2002)
24. Kiayias, A., Yung, M.: Traitor tracing with constant transmission rate. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 450–465. Springer, Heidelberg (2002)
25. Kiayias, A., Pehlivanoglu, S.: Pirate Evolution: How to Make the Most of Your Traitor Keys. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 448–465. Springer, Heidelberg (2007)

26. Kurosawa, K., Desmedt, Y.: Optimum Traitor Tracing and Asymmetric Schemes. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 145–157. Springer, Heidelberg (1998)
27. Le, T.V., Burmester, M., Hu, J.: Short c-Secure Fingerprinting Codes. In: Boyd, C., Mao, W. (eds.) ISC 2003. LNCS, vol. 2851, pp. 422–427. Springer, Heidelberg (2003)
28. Micciancio, D., Panjwani, S.: Corrupting One vs. Corrupting Many: The Case of Broadcast and Multicast Encryption. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 70–82. Springer, Heidelberg (2006)
29. Naor, D., Naor, M., Lotspiech, J.B.: Revocation and Tracing Schemes for Stateless Receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
30. Naor, M., Pinkas, B.: Threshold Traitor Tracing. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 502–517. Springer, Heidelberg (1998)
31. Naor, M., Pinkas, B.: Efficient Trace and Revoke Schemes. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 1–20. Springer, Heidelberg (2001)
32. Hieu Phan, D., Safavi-Naini, R., Tonien, D.: Generic Construction of Hybrid Public Key Traitor Tracing with Full- Public-Traceability. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174, pp. 49–63. Springer, Heidelberg (1996)
33. Safavi-Naini, R., Wang, Y.: Sequential Traitor Tracing. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 316–332. Springer, Heidelberg (2000)
34. Safavi-Naini, R., Wang, Y.: Collusion secure $q$-ary fingerprinting for perceptual content. In: Sander, T. (ed.) DRM 2001. LNCS, vol. 2320, pp. 57–75. Springer, Heidelberg (2002)
35. Safavi-Naini, R., Wang, Y.: New Results on Frameproof Codes and Traceability Schemes. IEEE Transactions on Information Theory 47(7), 3029–3033 (2001)
36. Safavi-Naini, R., Wang, Y.: Traitor Tracing for Shortened and Corrupted Fingerprints. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 81–100. Springer, Heidelberg (2003)
37. Silverberg, A., Staddon, J., Walker, J.L.: Efficient Traitor Tracing Algorithms Using List Decoding. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 175–192. Springer, Heidelberg (2001)
38. Staddon, J.N., Stinson, D.R., Wei, R.: Combinatorial Properties of Frameproof and Traceability Codes. IEEE Transactions on Information Theory 47(3), 1042–1049 (2001)
39. Stinson, D.R., Wei, R.: Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes. SIAM Journal on Discrete Math. 11(1), 41–53 (1998)
40. Tardos, G.: Optimal probabilistic fingerprint codes. In: ACM 2003, pp. 116–125 (2003)
41. Jho, N., Hwang, J.Y., Hee Cheon, J., Hwan Kim, M., Hoon Lee, D., Sun Yoo, E.: One-Way Chain Based Broadcast Encryption Schemes. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 559–574. Springer, Heidelberg (2005)