COMBINED FINGERPRINTING ATTACKS AGAINST DIGITAL AUDIO WATERMARKING: METHODS, RESULTS AND SOLUTIONS

Martin Steinebach, Jana Dittmann, Eva Saar
Fraunhofer IPSI Germany, martin.steinebach@ipsi.fhg.de
Platanista Germany, HTWK Leipzig University of Applied Sciences,
jana.dittmann@platanita.de
T-Systems, Germany, Eva.Saar@t-systems.de

Abstract: While a reliable protection against illegal copies does not exists today, tracking of illegal copies and prove of ownership are important detection functions, which can be realized by using passive security mechanisms of digital watermarking. Recent research has identified many watermarking algorithms for all common media types ranging from printed matter to multimedia files. The main topics of interest concentrates on transparency and robustness: The watermark must not reduce media quality and should be detectable after most common media operations and attacks. Algorithm security is discussed with regards only to key space most of the times, while especially for customer identification known as active fingerprinting specialized attacks like coalition attacks are known. Digital fingerprinting raises the additional problem that we produce different copies for each customer. Attackers can compare several fingerprinted copies to find and destroy the embedded identification string by altering the data in those places where a difference was detected. Few approaches have been introduced for image and video watermarking schemes, but there are no observations for audio fingerprinting techniques. In our paper we discuss methods for secure customer identification by digital fingerprinting for audio data. We describe first two algorithms by Boneh et al. [BoSh95] and Schwenk et al. [DBS+99] and then combine these schemes with an audio watermarking algorithm for practical evaluation of their coalition resistance to detect illegal copies. We provide test results and evaluate the security against different types of coalition attacks.

Key words: watermarking, customer authentication, fingerprinting algorithms, coalition attacks

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35612-9_23

B. Jerman-Blaži et al. (eds.), *Advanced Communications and Multimedia Security*

1. BACKGROUND AND MOTIVATION

The expansion of digital networks all over the world allows extensive access on, and reuse of, visual material. Problems include unauthorised taping, reading, manipulating or removing of data, which might lead to financial loss or legal problems of the producers and creators. Thus, designers, producers and publishers of digital data like images, video, audio or multimedia material are seeking technical solutions to the problems associated with copyright protection of multimedia data. Therefore the Internet has become in many cases a trading place for illegal copies of movies, music and software. Thus, systems are required which provide environments where digital data can be signed by authors or producers as their intellectual property, i.e. by embedding private or public information into the video data, to ensure and proof ownership rights on the produced video and audio material during its distribution. Digital watermarking in combination with active fingerprinting algorithms offers a solution to trace illegal copies. We introduce now the essential watermarking parameter for authentication and show customer-tracking concepts. In chapter two we discuss our general design of an collusion resistant audio fingerprint watermarking algorithms by using two general fingerprint construction schemes from [BoSh99] and [DBS+01]. In chapter three we introduce our test environment and the tested coalition attacks. Furthermore we describe the test procedure, the test tool and our test results with our implementation. We summarize our paper with a conclusion and directions of further work.

1.1 Customer authentication watermarking

For tracking the source of the copy an identification method is required. The copies must be personalized to decide who is responsible for the copyright violation. The goal of active digital fingerprinting is to embed customers IDs in the digital media. Personalization is of course only possible if the customer or user can be identified like in web shop environments, copies of e.g. songs ripped from CDs bought in stores cannot be marked this way. Embedding a customer ID brings along a number of requirements to the watermarking algorithms.

- Transparency is a common requirement for marking digital media in e-commerce environments as the quality of the content acting as a cover for the watermark must not be reduced.
- Robustness is necessary against common media operations like lossy compression and format change.

- Payload must be high enough to include the customer ID. This can become a critical requirement as we see in section [2]. In our paper we do not focus on that aspect in detail.
- Security is of special importance in this case as the existence of several copies of the same cover with different embedded customer Ids enables a number of specialized attacks commonly called coalition attacks. In our paper we want to evaluate this parameter in more detail for coalition attacks.
- Complexity has to be low enough to enable online and real time marking. A customer who wants to download a song is not willing to stay online for a long time until his personalized copy is available. As there will be multiple customers at the same time and media data may have a playing time of an hour or more, either streaming concepts or multiple real time embedding speed will be necessary.
- Verification should be performed in a secret environment. The shop owner holds a secret watermarking key to embed and read the watermarks. Customers do not know this key as attackers could easily verify their success with it.

1.2 Tracking of illegal copies

With the concept of customer authentication watermarking tracking of illegal copies becomes possible. It is common to have a user ID in web shops for services like customer accounts, tracking of orders and easy login. This ID can be used either directly or indirectly for personalization of digital data.

The direct method uses a watermarking algorithm to embed the ID in the downloaded content. For added security, the ID may be encrypted together with extracted content features to disable copy attacks. The indirect method uses a database. The download of a specific media are given a serial number consisting of a media ID and a running number. When a customer downloads a file, the actual serial number is embedded and stored in the database together with the customer ID. This allows tracking the copy without personalizing it. It would also allow to create pre-marked copies and to store them for use in times of high activity.

Both methods are used in the same way regarding the security framework. They are embedded when a user who authenticated himself at the time he logged in the web shop starts to download some digital content. Based on the user ID, the seed of the personalization process is as secure as the shop system.

Now every user receives an individual copy of the digital content and is responsible for guarantee legal distribution. A secure distribution channel may be necessary to ensure the individual copy is not duplicated while web delivery. E.g. a session key protocol could be applied based on a PKI framework also used for user authentication.

If a copy of the web shop offers is found in Internet trading places, on illegal CDR copies or similar places, the shop owner can use his secret key to retrieve the watermark and identify the origin of the illegal copy. This requires specialized search engines or equal mechanisms to track copies in the Internet. Without a way to find illegal copies, the watermark-based security mechanism cannot help to track illegal usage. An additional watermark with an owner ID could help to identify content coming from the specific web shop in combination with firewall concepts like the ones we describe in [DKL+02]. As an alternative, audio hashing concepts together with search engines and databases of all content sold by the web shop could be applied.

2. DESIGN OF FINGERPRINT WATERMARKING

To solve the problem of the coalition attack, we use the Boneh-Shaw fingerprint and the Schwenk-Ueberberg fingerprint algorithm [BoSh95], [DBS+99]. Both algorithms offer the possibility to find the customers, which have committed the coalition attack. In our last work in [DEV+01] we have introduced for both schemes a video fingerprinting solution and the coalition resistance. To mark the video, we generate positions within the frame to embed the watermark information (in the video the positions stand for scenes). Each customer has his own fingerprint, which contains a number of "1" and "0". Each fingerprint vector is assigned to marking positions in the document to prevent the coalition attack. The only marking positions the pirates cannot detect are those positions, which contain the same letter in all the compared documents. We call the set of these marking positions the intersection of the different fingerprints. In the following two subchapters we summarize the two fingerprinting schemes which are used and show how we apply these schemes to digital audio watermarking.

2.1 Fingerprinting concepts

A digital fingerprinting scheme consists of a number of marking positions in the document, a watermarking algorithm to embed letters from a certain alphabet at the marking positions, a fingerprinting algorithm which selects the letters to be embedded for each marking position depending on the number i of the copy and a pirate tracing algorithm which, on input of a modified document, outputs at least one number i of a copy that was used in constructing the modified document.

2.1.1 Schwenk Fingerprint Scheme

The Schwenk et al. approach [DBS+99] approach puts the information to trace the pirates into the intersection of up to d fingerprints. In the best case (e.g. automated attacks like computing the average of fingerprinted images) this allows us to detect all pirates, in the worst case (removal of individually selected marks) we can detect the pirates with a negligibly small one-sided error probability, i.e. we will never accuse innocent customers.

The fingerprint vector is spread over the marking positions. The marking positions for each customer are the same in every customer copy and the intersection of different fingerprints can therefore not be detected. With the remaining marked points, the intersection of all used copies, it is possible to follow up all customers, which have worked together. Another important parameter is the number n of copies that can be generated with such a scheme. The scheme uses techniques from finite projective geometry [Hir98], [BeRo98] to construct d-detecting fingerprinting schemes with q+1 possible copies. These scheme needs $n=q^d+q^{d-1}+...+q+1$ marking positions in the document. As we see this can be a huge length and can cause problems with the capacity of the watermarking scheme. The idea to built the customer vector is based on finite geometries and the detailed mathematical background ´can be found in[DBS+99].

2.1.2 Boneh-Shaw Fingerprint Scheme

The scheme of Boneh and Shaw [BoSh95] was designed to recognize coalition attacks, with a different approach. The method generates fingerprints for each customer containing a different number of zeros and ones. After the coalition attack the detection function we do not necessarily find all pirates. Furthermore with a (any arbitrary small) probability ε we get the wrong customer based on the different number of zeros in the detected fingerprint.

The number of customers is q and with q and ε we calculate the repeats d. The fingerprint vector consists of (q-1) blocks of the length d ("d-blocks"), the total length of the embedded fingerprint computes as $d*(q-1)$. Depending on the repeats the customer vector can be very long and cause problems with the capacity of the watermarking algorithm. The idea to built the fingerprinting vector for each customer is simple: The first customer has the value one in all marked points, for the second customer all marked points without the first "d-block" are ones, in the third all marked points without the first two "d-blocks" are ones etc. The last customer has the value 0 in all marked points. With a permutation of the fingerprint vector we get a higher security, because the pirates can find differences between the copies, but they can't assign it to a special d-block. Detailed mathematical background can be found in [BoSh95].

2.2 Fingerprinting algorithm combined with audio watermarking

To embed the customer information generated by the fingerprinting algorithm to trace illegal audio copies we use a digital watermarking algorithm. Current digital watermarking techniques usually would embed the generated fingerprinting information FP randomly all over the audio sequences with the disadvantage, that the intersection of the proposed fingerprints cannot be used to find attackers after comparing attacks of different customer copies. To use the excellent properties of the fingerprint to conclude to the customers which attacked the watermark we build a watermarking scheme as introduced for image and video data in [DBS+99] and [DHV+01] with a fixed number of **marking positions** in each copy of the audio. The **fingerprinting algorithm** selects the letters, the FP vector over the binary alphabet {0,1}. The **watermarking algorithm** embeds this binary FP vector at the chosen marking positions.

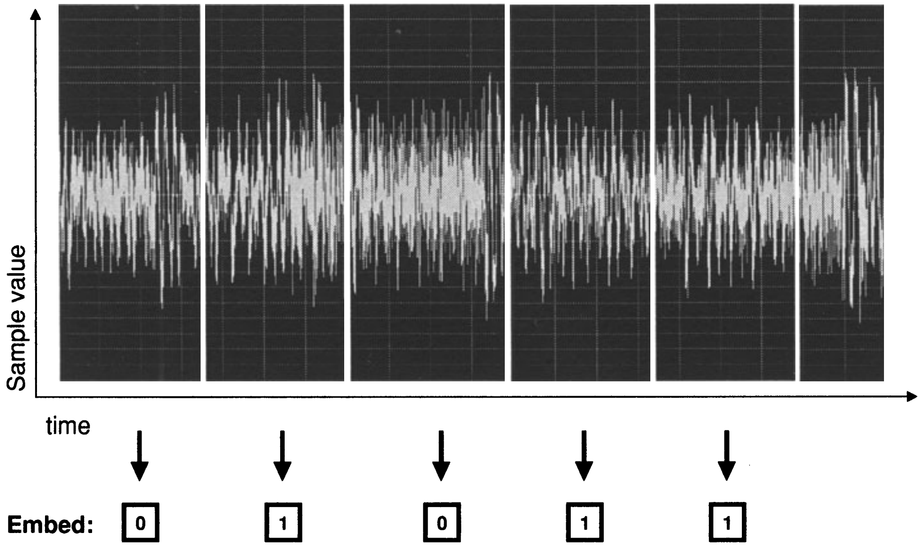


Figure 1: Audio watermarking over time

Generally watermarking algorithms use different methods to embed a message M into a cover C. In this paper, we use an audio watermarking algorithm to embed a fingerprint bit vector FP as M. The way M is embedded in C is relevant for the security of the watermarking and fingerprinting combination: An PCM audio stream consists of a sequence of audio samples over time. Most multi-bit message audio watermarking algorithms use a group of successive samples, e.g. 2048, to embed a single bit of the complete message. Figure 1 illustrates this: The bit sequence 01011 is embedded in a 1 second audio segment by separating the audio into groups of samples and embedding one bit in each of the segments.

This leads to the following situation: If two different bit vectors are embedded in two copies of the same cover with the same key, the two copies differ exactly in those segments where different bits have been embedded as information. Figure 2 shows two embedded bit vectors "01011" and "00001". Both have been embedded in a copy of C. If A and B compare their copies, they find equal segments at position 1,3 and 5 and different segments at position 2 and 4. This enables the attacks we describe in section 3.

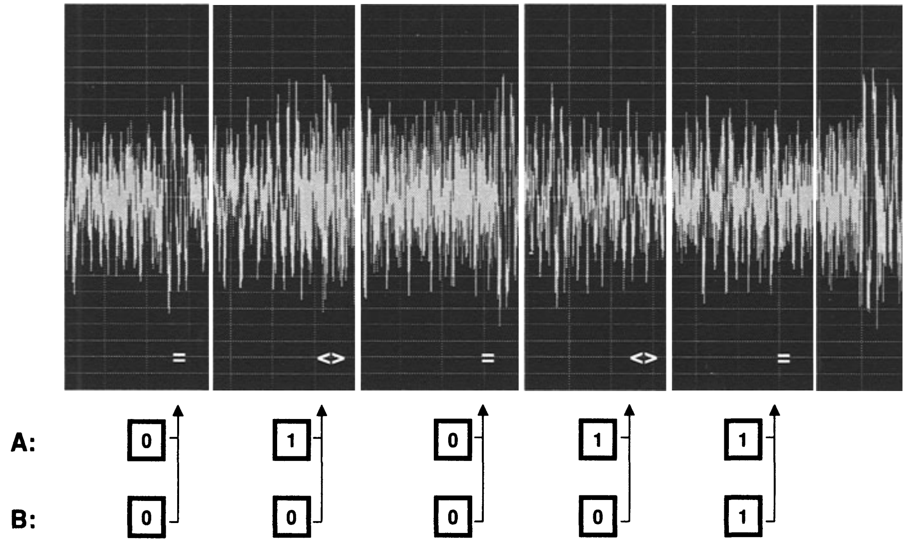


Figure 2: Different embedded bit vectors lead to diffent segments in the copies

3. SECURITY EVALUATION

In this section we describe our evaluation method for security against combined attacks. First we describe a typical attack scenario where two or more clients work together to remove an embedded watermark. Then we introduce our implemented test tool and its features. We show how this tool is used for simulating attacks and define different evaluation situations.

Only coalition attacks on fingerprint sequences are the subject of our tests in this work. Additionally, all attacks capable of removing the complete watermarking information (the client ID in this case) can be seen as successful attacks. In comparison to the attacks described here, these attacks are oriented on the watermarking algorithm, not against the fingerprinting scheme. For those interested in basic audio watermarking robustness, the Stirmark Benchmark Audio Suite [SMBM] and the corresponding papers [PSR+2001], [SPR+2001] and [SLD02] can be a valuable source of information.

3.1 Attack scenario

For coalition attacks against audio fingerprinting watermarks, we can expect that two or more customers will work together to identify differences between their individual copies received from a common source. One can assume they know that a) the media is protected by a customer identification mechanism and b) to identify customers, differences between the individual copies can be detected. Now they can design attacks based on this knowledge. They will either use an audio editor or a tool to compare their copies and create a new version of the media file where the customer Ids have be obscured.

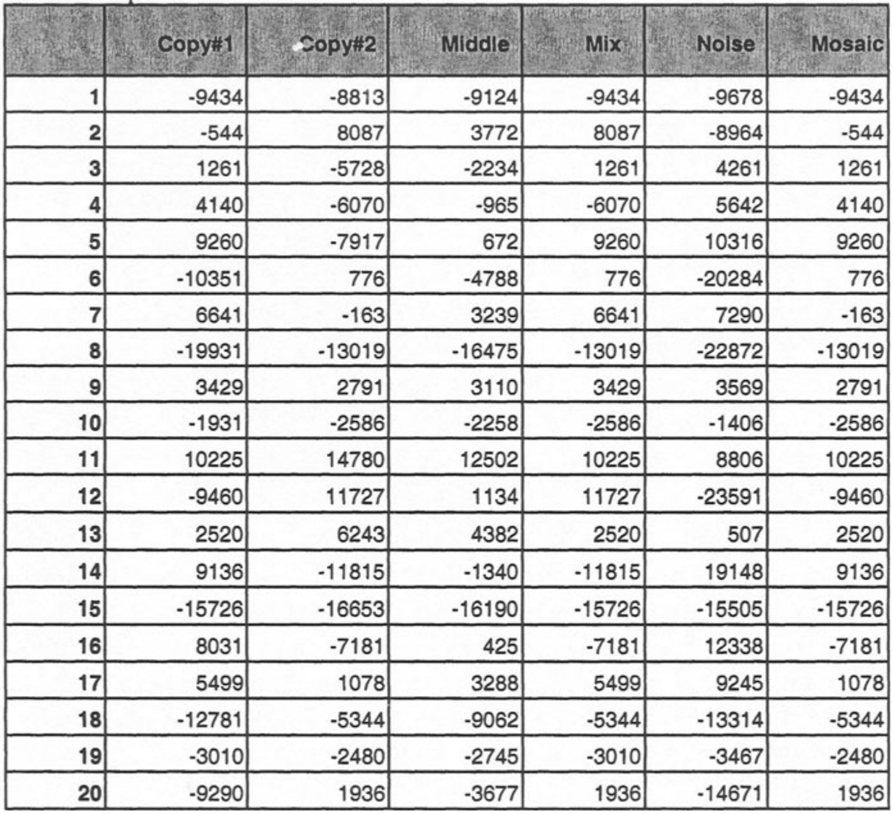
3.2 Test tool

We have implemented a test tool for coalition attacks which can run several types of attacks on a group of marked copies. The number of individual copies which will be used during the attack can be up to 5, which we think is sufficient for most scenarios.

The basic idea of the tool is to stream the marked copies in parallel and detect differences between the samples. If such differences are detected, out of varying attack strategies we have designed the following difference attacks:

- **Middle**: At detected differences the sample values are added and divided by the number of copies resulting in a middle sample value of all contributing files.
- **Switch**: Out of the detected differences one of the samples is selected. The choice is based on a loop running through the available source files. This creates a mixed sequence of the marked sources.
- **Noise**: The difference positions between the samples are calculated. The difference values at these positions is used as a maximum value for a random change in the first source file. This adds a kind of difference-triggered noise to the source file. A parameter is used to control the amount of added noise.

Table 1: Example results of the different attack modi



	Copy#1	Copy#2	Middle	Mix	Noise	Mosaic
1	-9434	-8813	-9124	-9434	-9678	-9434
2	-544	8087	3772	8087	-8964	-544
3	1261	-5728	-2234	1261	4261	1261
4	4140	-6070	-965	-6070	5642	4140
5	9260	-7917	672	9260	10316	9260
6	-10351	776	-4788	776	-20284	776
7	6641	-163	3239	6641	7290	-163
8	-19931	-13019	-16475	-13019	-22872	-13019
9	3429	2791	3110	3429	3569	2791
10	-1931	-2586	-2258	-2586	-1406	-2586
11	10225	14780	12502	10225	8806	10225
12	-9460	11727	1134	11727	-23591	-9460
13	2520	6243	4382	2520	507	2520
14	9136	-11815	-1340	-11815	19148	9136
15	-15726	-16653	-16190	-15726	-15505	-15726
16	8031	-7181	425	-7181	12338	-7181
17	5499	1078	3288	5499	9245	1078
18	-12781	-5344	-9062	-5344	-13314	-5344
19	-3010	-2480	-2745	-3010	-3467	-2480
20	-9290	1936	-3677	1936	-14671	1936

Furthermore we have included an attack, which is not difference-triggered. It is an audio mosaic attack where first a number of samples of the first source, then of the second source and so on are chosen in a loop. The number of consecutive samples from one source is given as a parameter.

Table 1 provides a short example of the different attack types. 20 samples are taken from two copies (copy#1 and copy#2). The column 'Middle' gives the resulting sample value of the middle attack. For the first samples, this is (-9434 + -8813) / 2 = -9124. 'Mix' is an alternating selection from the samples of copy#1 and #2. For the first sample, copy#1 is chosen. The second sample is taken from copy#2. 'Noise' induces changes in the samples of copy#1 controlled by the difference of the sample values between copy#1 and copy#2. 'Mosaic' is basically a 'mix' attack with a bigger step size, the first 5 samples are taken from copy#1, samples 6 to 10 are taken from copy#2 and so on.

3.3 Test procedure

The test procedure is similar for both evaluated fingerprinting schemes:

1. Create n fingerprint sequences depending on the number of customers.
2. Embed each fingerprint sequence in a cover creating n different copies.
3. Attack the fingerprint by using the n sources with the test tool and creating an attacked copy.
4. Detect the watermark in the attacked copy.
5. Identify the customer described by the detected fingerprinting sequences, verify the correctness, verify transparency.

In figure 3 we see a fingerprint attack scheme with two attackers.

While our tool is able to handle up to five differently marked copies, our tests only include coalition attacks of two or three customers. The applied audio watermarking algorithm is our own prototypic implementation. Tests with available demo versions of commercial products did show no different behaviour, so we use only one algorithm for our tests. Four audio test files have been chosen for evaluation: Classical music, pop, rock and speech to provide the typical range of audio material to be protected. While the performance of watermarking algorithm is not independent from the audio material, it has no direct influence to the results of the fingerprinting tests.

4. TEST RESULTS

In this section we summarize the test results of the procedure described in section 3. We do not consider transparency tests after embedding, attack and detection. While Middle, Mix and Mosaic attacks did not produce audible artefacts in some initial listening tests, Noise becomes audible at high levels. As Noise is also the least effective attack in our tests, one can assume that an attacker will not chose this method to destroy a fingerprint vector.

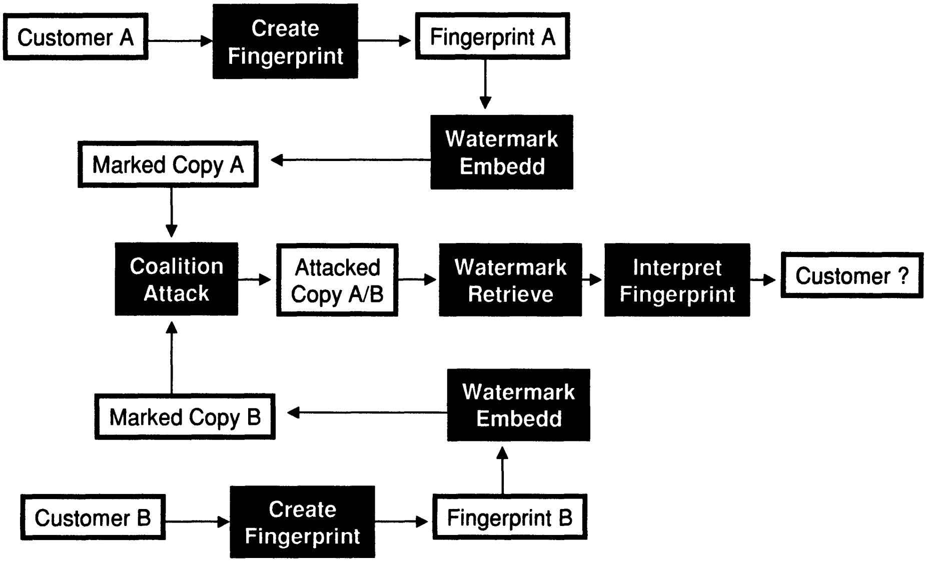


Figure 3: Fingerprint attack scheme with two attackers.

4.1 Schwenk

Table 2 provides an overview of our test results. Line 2 and 3 give the original fingerprint bit vectors, line 4 the result of an AND operation on both bit vectors to identify the common "1" (coalition identifier) bits. We embedded each bit vector twice in one example, the resulting bit vectors for all examples are listed from line 6. From the retrieved fingerprints generated by the Schwenk algorithm we can retrieve the coalition bit (the intersection of bits) successfully to identify the attackers. Most changes in the whole bit vector occur with the middle and switch attacks. The noise attack does not influence the watermark strong enough to result in detecting a wrong bit vector. Therefore if customer A's copy is modulated by the one of customer B, customer A is still identified after the attack. Most problems occur if additional ones are added so that other coalition bits are created. To enable detection, the watermarking algorithm has to ensure that only "0"-values can be created when a "0" and a "1" position are compared.

4.2 Boneh

The Boneh and Shaw algorithm performs similarly to the one by Schwenk when applied with additional bit vector encryption: Middle and Switch attacks result in non-interpretable bit vectors, noise attacks do not affect it in most cases. The most alarming test result is that in some cases a customer not involved in the attack was identified.

Table 2: Test results for client 1 and 2 with a Schwenk et al. fingerprinting scheme

	Bit #		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Client	*1*	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
		2	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0
	AND		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
classic	middle	*1*	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0
	middle	*2*	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0
	switch	*1*	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0
	switch	*2*	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0
	noise	*1*	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
	noise	*2*	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
pop	middle	*1*	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
	middle	*2*	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
	switch	*1*	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
	switch	*2*	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
	noise	*1*	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
	noise	*2*	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
rock	middle	*1*	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
	middle	*2*	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
	switch	*1*	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
	switch	*2*	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
	noise	*1*	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
	noise	*2*	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
speech	middle	*1*	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0
	middle	*2*	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0
	switch	*1*	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0
	switch	*2*	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0
	noise	*1*	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
	noise	*2*	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0

Table 3 shows an example result without bit vector encryption. Here we could identify one correct attacker after the attacks. Lines 1 and 2 show the bit vectors after their generation, line 3 to 5 the resulting vectors after the attack. The last column provides the identified customer.

Table 3: Boneh test results without encryption

Process	**Bit Vector**	**Identified**
ID 2:	0000000000000000000011111111111111111111 1111111111111111111111111111111111111111	2
ID 4:	0000000000000000000000000000000000000000 0000000000000000000011111111111111111111	4
switch	0000000000000000000001101010001011010010 1111000010110000001111111111111111111111	4
middle	0000000000000000000001101010001011010010 1111000010110000001111111111111111111111	4
noise	0000000000000000000011111111111111111111 1111111111111111111111111111111111111111	2

We also simulated an attack of three clients against the Boneh fingerprint. Table 4 shows an excerpt of the results. The complete bit vector has a length of 117 bits. In no case one of the clients could be identified. An interesting difference to the attacks with two clients can be observed in bit column 23: Bit #23 has the value 0 in all three client vectors. The detected post-attack bit value became 1 in some cases. These changes occurred several times in the vector, and both from value 1 to 0 and from 0 to 1.

This leads to an additional challenge in client identification compared to two-client attacks as there equal bit position have not been changed. Redundancy and error correction can help to lessen this thread, but also leads to even higher payload requirements.

Table 4: Attack of three clients vs. Boneh

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Client	1	0	1	1	1	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1
Client	2	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	1	1	1	0	0
Client	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
AND		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
Middle	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	0	0
	2	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1
	3	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	1	0	1	1	1	0	0
	4	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	1	0	0	0	0	1	1	1	0	0
Mix	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	1	1	0	0	0	1	1	0	0
	2	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1
	3	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	1	0	1	1	1	0	0
	4	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	1	0	0	0	0	1	1	1	0	0
Noise	1	0	1	1	1	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
	2	0	1	1	0	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1
	3	0	1	1	1	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1
	4	0	1	1	1	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1

4.3 Client identification without fingerprinting

To verify the necessity of fingerprinting methods, we also examined the performance of different alternative client identification methods. Three different strategies have been evaluated:

- Same key, different watermark: A customer ID is embedded without a fingerprinting strategy using always the same key. The customer ID could be a real name or an ID number.
- Same watermark, different key: Instead of using the watermark as the carrier, the customer could also be identified by a key. This key depends on the client ID, and if a detector is able to retrieve a watermark from the cover, the client is identified. Searching a customer would require brute force scanning through the key space.
- Different key, different watermark: A combination of both strategies. A key is generated based on the client ID, and the watermarking message is the same client ID. This adds security regarding brute force attacks of third parties as the relation of key and ID may be secret.

We used the same attack tool to evaluate the performance of the three methods and got similar results for all of them. This time 10 test files including the 4 of tests 4.1 und 4.1 have been chosen. Two attackers were simulated. In all cases, a correct detection of the client was possible in less then 50% of the attacks. As in tests 4.1 and 4.2 the results were best with the noise attack. In some attacks and files a correct identification was possible in less then 20% of the cases.

This shows that a strategy not using fingerprinting concepts is very vulnerable to coalition attacks. While the results of both fingerprinting and non-fingerprinting approaches are not satisfying, optimisation potential for fingerprinting-based solutions is much greater. In section 5 we describe one possible optimisation.

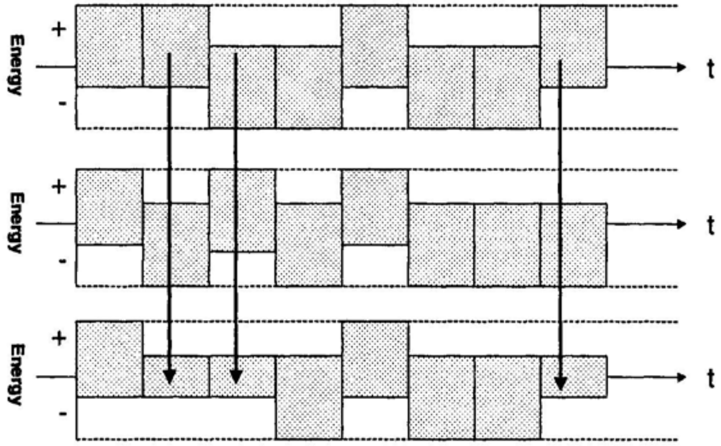
5. CONCLUSION AND FUTURE WORK

In comparison to the collusion resistant results of the image and video fingerprinting watermarking algorithms introduced in [DBS+99] and [DHV+01] the audio results show that the coalition bits of the Schwenk et al. approach can be found successfully. Problems occur if "1"s are created by an attack and can cause problems with other possible coalition bits with other customers. Further tests are necessary for a more comprehensive study. Furthermore the Boneh et al. approach has problems with addional ones and it is possible that innocent customer can be identified.

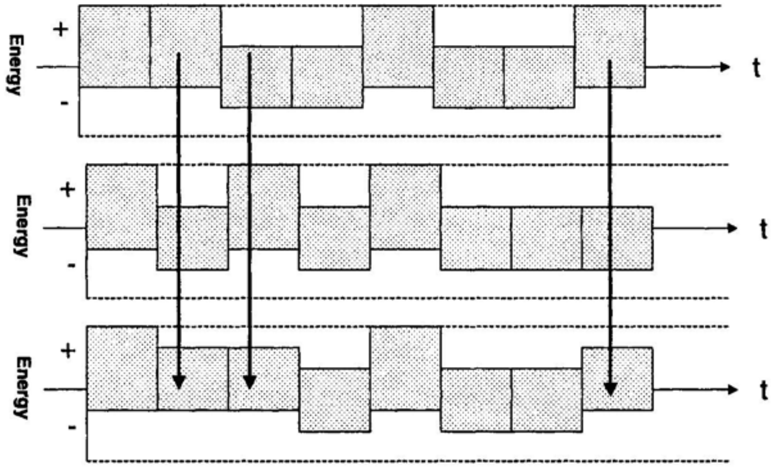
For identifying users that took part in a coalition attack, it could be helpful to change the embedding algorithm so that a rule could be set for mixing two fingerprints. When every time an embedded "0" and "1" are mixed, one specific bit occurs, we would receive a bit vector much more easy to interpret. In the case of the Schwenk algorithm mixing a "0" and a "1" should always result in a "0" as the "1"s are used to identify the group of attackers. E.g. the sequences "0010101" and "0110001" could now result in "0110101" or "0011101" among other possibilities. After an optimisation the only possible result should be "0010001" identifying both attackers by the shared "1"s at bit#2 and bit#6.

To use different embedding strengths for both bits can be a solution and should be take into considerations of further evaluations. In the case of middle or mix attacks this would result in the bit embedded with more strength surviving the coalition attacks. Figure 4 illustrates this concept.

Furthermore generally for a high number of customers the length of the fingerprint vectors for both fingerprint schemes is very high. The optimization of the fingerprint algorithms is an important point for the future research. The problem is to embed the customer vector in material with restricted size.



In a common statistical watermarking algorithm, 0 and 1 Bits are embedded by increasing and decreasing energy of frequency bands at the same amount. When the watermarks of line 1 and 2 are combined, the resulting sequence of energy changes becomes hard to interpret.



We suggest a stronger increase in energy in one direction, + in this case. Now if line 1 and 2 are combined, the resulting watermark still shows a certain tendency to "+", which is weaker then a original "+" position.

Figure 4: Fingerprinting-optimised watermarking

REFERENCES

[DBS+99] Dittmann, Jana; Behr, Alexander; Stabenau, Mark; Schmitt, Peter; Schwenk, Joerg; Ueberberg, Johannes (1999), ***Combining digital Watermarks and collusion secure Fingerprints for digital Images***, Proceedings of SPIE Vol. 3657, [3657-51], Electronic Imaging '99, San Jose USA, 24-29 January 1999.

[BoSh95] D. Boneh and J. Shaw, *Collusion-Secure Fingerprinting for Digital Data.* Proc. CRYPTO'95, Springer LNCS 963, S. 452-465, 1995

[BeRo98] A. Beutelspacher and U. Rosenbaum, *Projective Geometry.* Cambridge University Press 1998.

[DEV+01] Dittmann, Jana; Hauer, Enrico; Vielhauer, Claus; Schwenk, Jörg; Saar, Eva: Customer Identification for MPEG Video based on Digital Fingerprints. In: Proceedings of Advances in Multimedia Information Processing - PCM 2001, The Second IEEE Pacific Rim Conference on Multimedia, Beijing, China, Springer Verlag, Berlin, pp. 383 - 390, ISBN 3-540-42680-9, 2001.

[Hir98] J.W.P. Hirschfeld, *Projective Geometries over Finite Fields.* Oxford University Press, 2nd Edition 1998.

[DKL+02] Jana Dittmann, Stephan Klink, Andreas Lang, Martin Steinebach: Wasserzeichen unterstützte Firewalls, to appear in Proc. Of Enterprise Security, Paderborn, Germany, 2002

[PSR+2001] Petitcolas, F. A. P.; Steinebach, Martin; Raynal, F.; Dittmann, Jana; Fontaine, C.; Fates, N. (2001). Public automated web-based evaluation service for watermarking schemes: StirMark Benchmark. In: Security and Watermarking of Multimedia Contents III, Ping Wah Wong, Edward J. Delp III, Editors, Proceedings of SPIE Vol. 4314, pp. 575 - 584, ISBN 0-8194-3992-4, 2001.

[SPR+2001] Steinebach, Martin; Petitcolas, Fabien A. P.; Raynal, Frederic; Dittmann, Jana; Fontaine, Caroline; Seibel, Christian; Fates, Nazim; Croce Ferri, Lucilla (2001). StirMark Benchmark: Audio watermarking attacks. In: Int. Conference on Information Technology: Coding and Computing (ITCC 2001), April 2 - 4, Las Vegas, Nevada, pp. 49 - 54, ISBN 0-7695-1062-0, 2001.

[SLD02] Steinebach, Martin; Lang, Andreas; Dittmann, Jana, "StirMark Benchmark: Audio watermarking attacks based on lossy compression", Photonics West 2002, 19 - 25 January 2002, Electronic Imaging 2002: Science and Technology; Multimedia Processing and Applications, Security and Watermarking of Multimedia Contents IV, San Jose, CA, USA

[SMBM] http://ms-smb.ipsi.fhg.de/stirmark/index.php