Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2008, Article ID 803217, 15 pages
doi:10.1155/2008/803217

Research Article

Novel Attacks on Spread-Spectrum Fingerprinting

Hans Georg Schaathun

Department of Computing, University of Surrey, Guildford, Surrey GU2 7XH, UK

Correspondence should be addressed to Hans Georg Schaathun, h.schaathun@surrey.ac.uk

Received 9 May 2008; Accepted 7 August 2008

Recommended by Stefan Katzenbeisser

Spread-spectrum watermarking is generally considered to be robust against collusion attacks, and thereby suitable for digital fingerprinting. We have previously introduced the minority extreme attack (IWDW '07), and showed that it is effective against orthogonal fingerprints. In this paper, we show that it is also effective against random Gaussian fingerprint. Furthermore, we develop new randomised attacks which counter the effect of the decoder preprocessing of Zhao et al.

1. INTRODUCTION

Unauthorised copying is a major worry for many copyright holders. As digital equipment enables perfect copies to be created on amateur equipment, many are worried about lost revenues, and steps are introduced to reduce the problem. Technology to prevent copying has been along for a long time, but it is often controversial because it not only prevents unauthorised copying, but also a lot of the legal and fair use.

A different approach to the problem is to deter potential offenders using technology to allow identification after the crime. Thus, the crime is not prevented, but the guilty users can be prosecuted. If penalties are sufficiently high, potential pirates are unlikely to accept the risk of being caught.

One such solution is digital fingerprinting, first proposed by Wagner [1]. Each copy of the copyrighted file is marked by hiding a fingerprint identifying the buyer. Illegal copies can then be traced back to one of the legitimate copies and the guilty user be identified. Obviously, the marking must be made such that the user cannot remove the fingerprint without ruining the file. Techniques to hide data in a file in such a way are known as robust watermarking. All references to watermarking (WM) in this paper refer to robust watermarking.

A group of users can compare their individual copies and observe differences caused by the different fingerprints embedded. By exploiting this information they can mount so-called *collusive attacks*. There is a growing literature on collusion-secure fingerprinting, both from mathematical and abstract and from practical view-points.

In this paper, we focus on Gaussian, spread-spectrum fingerprinting, where each user is identified by a random, Gaussian signal which is added to the copyrighted file (host signal). Our main purpose is to demonstrate that there are collusion attacks which are more effective than the ones studied by Zhao et al. [2]. We make extensive experiments to compare the various attacks. Our starting point is the minority extreme attack introduced in [3] in a context of non-Gaussian fingerprints.

The outline of the paper is as follows. We will introduce our model for fingerprinting in general and spread spectrum fingerprinting in particular in Section 2. We introduce our new collusion attacks in Section 3, and consider noise attacks in Section 4. In Section 5, we make a further evaluation, testing the attacks under different conditions. Finally, there is a conclusion in Section 6.

2. FINGERPRINTING MODELS

There are several different approaches to fingerprinting. It is often viewed as a layered system. In the fingerprinting (FP) layer, each user is identified by a codeword $\mathbf{c}$, that is, an n-tuple of symbols from a discrete q-ary alphabet. If there are M codewords (users), we say that they form an $(n, M)_q$ code.

In the watermarking (WM) layer, the copyrighted file is divided into n segments. When a codeword $\mathbf{c}$ is embedded, each symbol of $\mathbf{c}$ is embedded independently in one segment.

The layered model allows independent solutions for each layer. Coding for the FP layer is known as collusion-secure

Table 1: Overview of notation used throughout.

Symbol	Name
$\mathbf{x}$	Host signal (original, copyrighted file)
$\mathbf{w}^{(u)}$	Watermark of user u
$\mathbf{y}^{(u)} = \mathbf{x} + \mathbf{w}^{(u)}$	Watermarked file distributed to user u
$\mathbf{z}$	Hybrid copy produced by the collusion
$\mathbf{r} = \mathbf{z} - \mathbf{x}$	Received watermark
$\mathbf{r}'$	Received watermark after preprocessing

codes and was introduced in [4]. A number of competing abstract models have been suggested, and mathematically secure solutions exist for most of the models.

In principle, any robust watermarking scheme can be used in the WM layer. However, there has been little research into WM systems which supports the abstract models assumed for the collusion-secure codes, thus it is not known whether existing collusion-secure is applicable to a practical system. Recent studies of this interface are found in [5, 6], but they rely on experimental studies with few selected attacks, and the mathematical model has not been validated.

In this paper, we will consider a simpler class of solutions, exploiting some inherent collusion resistance in spread-spectrum watermarking. We focus on the solution suggested in [2].

2.1. Spread-spectrum fingerprinting

We view the copyrighted file as a signal $\mathbf{x} = (x_1, \ldots, x_N)$, called the *host signal*, of real or floating-point values x_i. Each user u is identified by a *watermark signal* $\mathbf{w}^{(u)} = (w_1^{(u)}, \ldots, w_N^{(u)})$ over the same domain as the host signal. The encoder simply adds the two signals to produce a *watermarked copy* $\mathbf{y}^{(u)} = (y_1^{(u)}, \ldots, y_N^{(u)})$ for distribution.

A goal is to design the watermark $\mathbf{w}$ so that $\mathbf{y}$ and $\mathbf{x}$ are perceptually as similar as possible. No perfect measure is known to evaluate perceptual similarity. He and Wu [5] use the peak signal-to-noise ration (PSNR). Zhao et al. [2] consider the *just noticeable difference* (JND) as the smallest, perceptible change which can be made to a single sample, and they measure distortion as the mean square error (MSE) ignoring samples with distortion less than some threshold (called JND). This heuristic is called MSE_{JND}.

In the system of [2], which we study, the watermark signals $\mathbf{w}^{(u)}$ are drawn independently at random from a normal distribution with variance $\sigma^2 = 1/9$ and mean $\mu = 0$.

It is commonly argued that in most fingerprinting applications, the original file will be known by the decoder, so that nonblind detection can be used [2, 5]. Let $\mathbf{z} = (z_1, \ldots, z_N)$ denote the received signal, such as an intercepted unauthorised copy. Knowing $\mathbf{x}$, the receiver can compute the *received watermark* $\mathbf{r} = (r_1, \ldots, r_N) = \mathbf{z} - \mathbf{x}$, which is the input to the decoder.

The adversary, the copyright pirates in the case of fingerprinting, will try to disable the watermark by creating an attacked copy $\mathbf{z}$ which is perceptually equivalent to $\mathbf{y}$, but where the watermark cannot be correctly interpreted. In the case of a collusion attack, there is a group of pirates each possessing one watermarked copy $\mathbf{y}_i$.

An overview of the symbols introduced can be seen in Table 1.

2.2. Fingerprint decoding

For any signal $\mathbf{s}$, let $\bar{s}$ denote its average, that is

$$\bar{s} = \sum_{i=1}^{N} \frac{s_i}{N}. \tag{1}$$

The Euclidean norm is denoted by

$$\|\mathbf{s}\| = \sqrt{\sum_{i=1}^{N} s_i^2}. \tag{2}$$

The correlation of two signals is denoted by

$$\langle \mathbf{s}, \mathbf{s}' \rangle = \sum_{i=1}^{N} s_i s_i'. \tag{3}$$

The simplest decoding algorithm would return the user solving $\max_u \langle \mathbf{w}^{(u)}, \mathbf{r} \rangle$. This is sometimes used, but more often some kind of normalisation is recommended.

2.2.1. The general decoder

Following [2], we study three heuristics which assign a numerical value $h(\mathbf{r}, \mathbf{w})$ to any pair of signals $\mathbf{r}$ and $\mathbf{w}$. Each heuristic h can be used either for list decoding or for *maximum heuristic decoding*. The latter returns the user u solving $\max h(\mathbf{r}, \mathbf{w}^{(u)})$. A *list decoder* would return all users u such that $h(\mathbf{r}, \mathbf{w}^{(u)}) \geq \tau$ for some threshold τ.

The performance measure for a maximum heuristic decoder is simply the error rate. Only one user is output, who is either guilty (correct) or not (error). List decoder performance cannot be described by a single parameter. The output may be empty (false negative); it may include innocent users (false positive); or it may be a nonempty set of guilty users only (correct decoding). The trade-off between false positive and false negative error rates is controlled by the threshold τ.

One may also want to consider the number of guilty users returned by the list decoder. If two decoders have identical error rates, one would clearly prefer one which tends to return two guilty users instead of just one.

It should be noted that a list decoder can never have a higher probability of correct decoding than a maximum heuristic decoder for the same heuristic. When the list decoder decodes correctly, the user with the maximum heuristic will clearly be in the output set and also be correctly returned by the maximum heuristic decoder.

We will mainly consider the maximum heuristic decoder. This does provide a bound on the performance of a list decoder, and we avoid any potential controversies in the choice of τ.

2.2.2. Decoding heuristics

The so-called T statistic is simply normalised correlation, defined as follows:

$$T(u) = \frac{\langle \mathbf{r}, \mathbf{w}^{(u)} \rangle}{\|\mathbf{w}^{(u)}\|}. \tag{4}$$

From the attacker's point of view, this is the easiest heuristic to analyse, as it is linear in each sample of $\mathbf{r}$.

The most effective heuristic according to the experiments of [2] is the so-called Z statistic, defined as

$$Z(u) = \frac{1}{2}\sqrt{N-3}\log\frac{1+\rho_u}{1-\rho_u}, \tag{5}$$

where

$$\rho_u = \frac{(1/N)\langle \mathbf{r}, \mathbf{w}^{(u)} \rangle - \bar{r}\,\bar{w}^{(u)}}{\hat{\sigma}_{\mathbf{r}}\hat{\sigma}_{\mathbf{w}^{(u)}}}, \tag{6}$$

where $\bar{s}$ is the mean of $\mathbf{s}$ and $\hat{\sigma}_{\mathbf{s}}$ is the empirical standard deviation, that is,

$$\hat{\sigma}_{\mathbf{s}}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(s_i - \bar{s})^2. \tag{7}$$

The final statistic is the q statistic, which is based on the mean M_u and standard deviation V_u of the signal $(r_i w_i^{(u)} \mid i = 1, \ldots, N)$. It is defined as

$$q(u) = \frac{\sqrt{N}M_u}{V_u}. \tag{8}$$

Observe that $M_u = \langle \mathbf{r}, \mathbf{w}^{(u)} \rangle / N$. Thus, all the three heuristics are based on correlation.

2.2.3. Preprocessing

Zhao et al. [2] point out that the three decoding heuristics presented have not been designed for collusion-resistance in particular. In order to improve the performance, they introduce a preprocessing step. The theoretical foundation is not very clear in their paper, but it works well experimentally. Our simulations have confirmed this.

They considered the histogram of the received watermark $\mathbf{r}$ at the decoder for various attacks presented in Section 2.3.

The median, average, and midpoint attacks roughly produce normal distribution with zero mean. The Min and Max attacks give normal distributions with nonzero means (negative and positive means, resp.). The RandNeg attacks give a histogram with two peaks, one positive and one negative. Very few samples are close to zero.

In the case of the single peak, the preprocessor subtracts the mean, to return $\mathbf{r}' = \mathbf{r} - \bar{r}$. In the case of a double peak, the samples are divided into two subsets, one for negative values and one for positive ones. The mean is calculated and subtracted independently for each subset.

Zhao et al. gave no definition of a peak in the histogram, and no algorithm to identify them automatically. As long as we are restricted to the known attacks, this is only a minor problem. It is obvious from visual inspection which case we are in.

We will, however, introduce attacks where it is not clear which preprocessor mode to use. In these cases we will test both modes, so Preproc(1) denotes the preprocessor assuming two peaks, and Preproc(2) is the preprocessor assuming a single peak.

2.3. Spread spectrum collusion attacks

The collusion attack is mounted by a collusion of pirates, each of whom has a watermarked copy $\mathbf{y}^{(u)}$ perceptually equivalent to the (unknown) host $\mathbf{x}$. The most commonly studied attacks are functions working independently on each sample i, that is, $z_i = A(y_i^{(u_1)}, \ldots, y_i^{(u_t)})$, where $P = \{\mathbf{y}^{(u_1)}, \ldots, \mathbf{y}^{(u_t)}\}$ is the set of colluder watermarks.

Both randomised and deterministic attack functions A have been studied. In principle, A could depend on the entire signal, and not only on the samples corresponding to the output sample, but this possibility has received little attention in the literature. Our starting point is the following range of attacks which were analysed in [2].

$$\begin{aligned}
\text{Average:} \quad & \bar{z}_i = \frac{1}{t}\sum_{\mathbf{y}\in P} y_i. \\
\text{Minimum:} \quad & z_i^{\min} = \min_{\mathbf{y}\in P} y_i. \\
\text{Maximum:} \quad & z_i^{\max} = \max_{\mathbf{y}\in P} y_i. \\
\text{Median:} \quad & z_i^{\mathrm{med}} = \operatorname*{median}_{\mathbf{y}\in P} y_i. \\
\text{Midpoint (MinMax):} \quad & z_i^{\mathrm{mid}} = (z_i^{\min} + z_i^{\max})/2. \\
\text{Modified negative:} \quad & z_i^{\mathrm{modneg}} = z_i^{\min} + z_i^{\max} - z_i^{\mathrm{mid}}. \\
\text{Randomised negative:} \quad & z_i^{\mathrm{rndneg}} = \begin{cases} z_i^{\min} & \text{with probability } p, \\ z_i^{\max} & \text{with probability } 1-p, \end{cases}
\end{aligned} \tag{9}$$

It was assumed in [2], that p for the randomised negative attack be independent of the signals $\{y_i\}$.

The analysis of [2] demonstrated that the randomised negative attack gave the highest error rate against decoders without preprocessing. None of the attacks were effective against decoders with preprocessing for the parameters studied. The average attack gives the lowest distortion of all the attacks. This is obvious as it is known as a good estimate for the original host $\mathbf{x}$.

2.4. Collusion attacks and collusion-secure codes

It is instructive to consider attacks commonly considered in the literature on collusion-secure codes. Recall that the fingerprint $\mathbf{w}$ in the context of collusion-secure codes is not a numerical signal, but rather a word (vector) over a discrete alphabet Q. The basic operations of average, minimum, and maximum are not defined on this alphabet.

The so-called marking assumption defines which attacks are possible in the model. In the original scenario of [4],

the pirates can produce an output symbol z_i, if and only if $z_i \in \{y_i^{(u_1)}, \ldots, y_i^{(u_t)}\}$. In a more realistic scenario [6, 7], the pirates can produce a symbol $z_i \notin \{y_i^{(u_1)}, \ldots, y_i^{(u_t)}\}$ with probability p. However, with probability $1 - p$, we have $z_i \in \{y_i^{(u_1)}, \ldots, y_i^{(u_t)}\}$.

It is generally known that the so-called *minority choice attack* is very effective if correlation decoding (or, equivalently, closest neighbour decoding) is used. In this attack the output is the symbol $z_i \in \{y_i^{(u_j)} \mid j = 1, \ldots, t\}$ minimising the number of colluders u with $y_i^{(u)} = z_i$.

The rationale for this attack is straight forward. All the colluders u with $y_i^{(u)} = z_i$ gets a positive contribution to the correlation from sample i; all the other users get a negative contribution. Hence, the minority choice minimises the average correlation of the colluders.

The minority choice attack does not apply directly to Gaussian fingerprints. With each watermark drawn randomly from a continuous set, one would expect all the samples $y_i^{(u)}$ seen by the pirates to be distinct. However, we will see that we can construct an effective attack based on the same idea.

2.5. Evaluation methodology

There are two important characteristics for the evaluation of fingerprinting attacks.

Success rate: The attack succeeds when an error occurs at the watermark decoder.

Distortion: The unauthorised copy has to pass in place of the original, so it should be as close as possible to the unknown signal $\mathbf{x}$ perceptually.

The success rate of the attack is the resulting error rate at the decoder/detector. As long as we use a maximum heuristic decoder, this is a single figure. In the event of list decoding, it is more complex as explained in Section 2.2.1.

Distortion is, following [2], measured by the MSE_{JND} as defined below.

Definition 1 (just notable difference). Given a signal $\mathbf{x} = (x_1, \ldots, x_N)$, the *just noticeable difference*, JND_i, is the smallest positive real number, such that $\mathbf{x}' = (x_1, \ldots, x_{i-1}, x_i \pm \text{JND}_i, x_{i+1}, \ldots, x_N)$ is perceptually different from $\mathbf{x}$.

In our simulations we have assumed, without loss of generality, that $\text{JND}_i = 1$ for all i. The general case is achieved by scaling each sample of the fingerprint signal by factor of JND_i^{-1} before embedding, and rescale before decoding.

Definition 2. The MSE_{JND} between to signal $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\text{MSE}_{\text{JND}} = \sum_{i=1}^{N} [\max\{0, (|x_i - y_i| - \text{JND}_i)\}]^2. \tag{10}$$

It is natural to expect low distortion from the average, median, and midpoint attacks. The pirate collusion is likely to include both positive and negative fingerprint signals. Consequently, these attacks are likely to produce a hybrid which is closer to the original sample than any of the colluder fingerprints. On the contrary, the maximum, minimum, and randomised negative attacks would tend to give a very distorted hybrid, by using the most distorted version of each sample. This is experimentally confirmed in [2, 8].

Not surprisingly, the most effective attacks are the most distorting. The most effective attack according to [8] is the randomised negative, but the authors raise some doubt that it be practical due to the distortion.

The performance of existing fingerprinting schemes and joint WM/FP schemes have been analysed experimentally or theoretically. Very few systems have been studied both experimentally and theoretically. In the cases where both theoretical and experimental analyses exist, there is a huge discrepancy between the two.

It is not surprising that theoretical analyses are more pessimistic than experimental ones. An experimental simulation (e.g., [5]) has to assume one (or a few) specific attack(s). An adversary who is smarter (or more patient) than the author and analyst may very well find an attack which is more effective than any attack analysed. Thus, the experimental analyses give lower bounds on the error rate of the decoder, by identifying an attack which achieves the bound.

The theoretical analyses of the collusion-secure codes of [4, 9, 10] give mathematical upper bounds on the error rate under any attack provided that the appropriate marking assumption holds. Of course, attacks on the WM layer (which is not considered by those authors) may very well break the assumptions and thereby the system. Unfortunately, little work has been done on theoretical upper bounds for practical fingerprints embedded in real data.

In any security application, including WM/FP schemes, the designer has a much harder task than the attacker. The attacker only needs to find one attack which is good enough to break the system, and this can be confirmed experimentally. The designer has to find a system which can resist every attack, and this is likely to require a complex argument to be assuring.

This paper will improve the lower bounds (experimental bounds) for Gaussian spread spectrum fingerprinting, by identifying more complex nonlinear attacks, which are more effective than those originally studied. These attacks are likely to be effective against other joint schemes as well.

3. THE NOVEL ATTACKS

In this section, we will consider four new classes of attacks. The minority extreme attack was introduced in a different model in [3], and the uniform attack is introduced in this paper. The last two classes of attacks are hybrid attacks, behaving as different pure attacks either at random or depending on the collusion signals. We introduce each attack separately with its rationale and simulation results. In the next section we will consider noise attacks.

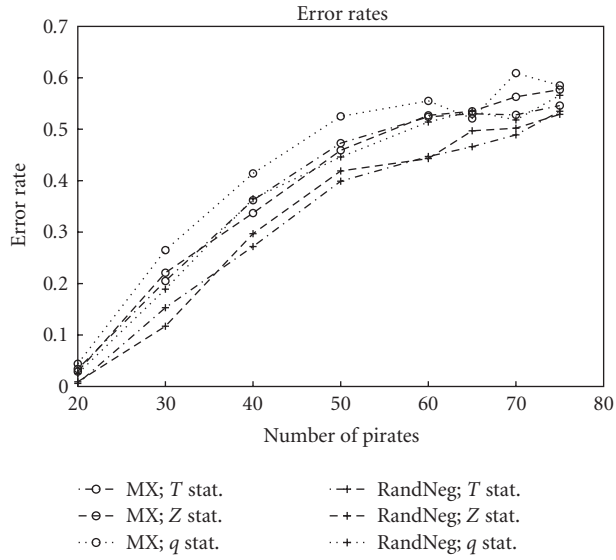


Figure 1: Comparing MX against RandNeg. Decoding with preprocessing gives zero errors throughout.

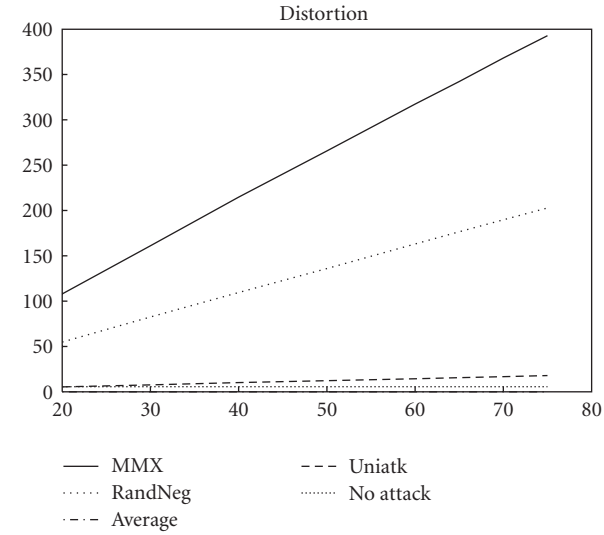


Figure 2: Distortion of pure attacks.

Let $\mathbf{w}_u$ be the watermark identifying user u, and let $\mathbf{r} = \mathbf{z} - \mathbf{x}$ be the hybrid watermark generated by the collusion. All the heuristics we consider include the correlation

$$h_u = \mathbf{r} \cdot \mathbf{w}^{(u)} = \sum_{i=1}^{N} r_i \cdot w_i^{(u)}. \tag{11}$$

In order to avoid detection, the pirates should attempt to minimise $\max_{u \in P} h_u$. Without complete knowledge of the original host $\mathbf{x}$ and the watermark signals used, an accurate minimisation is intractable. However, attempting to minimise $\overline{h} = \text{avg}_{u \in P} h_u$ is a reasonable approximation, and this can be done by minimising sample by sample, $\text{avg}_{u \in P} r_i \cdot w_i^{(u)}$.

All the simulations in this section use sequences of length $n = 10\,000$ with $M = 512$ users. The sequences are drawn from a normal distribution of mean $\mu = 0$ and variance $\sigma^2 = 1/9$.

With the exception of the code size (i.e., the number of users), these are the same parameters as used in [2]. There are two reasons for using larger codes. Firstly, it is hard to come up with plausible applications for small codes. Secondly, and more importantly, larger codes give higher error rates which can be estimated more accurately.

For each simulation, 1000 different codes are created, and one hybrid fingerprint is generated and decoded for each code. Although this is a smaller sample size than the 2000 tests used in [2], it is appropriate for tuning the attack parameters. In the next section we will run larger simulations for a more significant comparison to previous work.

3.1. The minority extreme attack

We introduced the moderated minority extreme (MMX) attack in [3] in order to break the joint scheme of [5]. Consider the difference $D = z_i^{\text{avg}} - z_i^{\text{mid}}$. Since z_i^{mid} is an unbiased estimate for the unknown host x_i, a positive D indicates that $\overline{w}_i$ is probably positive. In this case, the minimum attack is good for the pirates.

If $D \approx 0$, we expect that the choice for z_i makes little difference to the decoding. In this case, we output $z_i = \overline{z}_i$ to minimise the distortion in the hybrid copy.

Definition 3 (moderated minority extreme attack). Let $D_i = z_i^{\text{avg}} - z_i^{\text{mid}}$. The MMX attack for a given threshold θ outputs the hybrid signal $\mathbf{z}^{\text{MMX}(\theta)}$, where

$$z_i^{\text{MMX}(\theta)} = \begin{cases} z_i^{\min} & \text{if } D_i \geq \theta, \\ z_i^{\text{avg}} & \text{if } \theta > D_i > -\theta, \\ z_i^{\max} & \text{if } D_i \leq -\theta. \end{cases} \tag{12}$$

The MMX attack with $\theta = 0$ was called the minority extreme (MX) attack [3]. Figure 1 shows a simulation of the MX and RandNeg attack. We observe that the MX attack causes a slightly higher error rate, confirming that the criterion that $D > 0$ is better than a random choice. However, with preprocessing, the error rate is zero for both attacks. The average attack was tested as well, but it gave zero errors with all of the tested decoders. These results are consistent with those reported in [2].

Figure 2 shows, unfortunately, that the MX attack also causes about twice the distortion of RandNeg. Given the very modest increase in error rate, the MX attack is unlikely to be useful in itself.

3.2. The uniform attack

So far we have seen that the preprocessor of Zhao et al. is very effective against the attacks considered to date. Somehow we need to break the preprocessing scheme.

Remember that the preprocessor considers the histogram and split the samples into two classes around each histogram

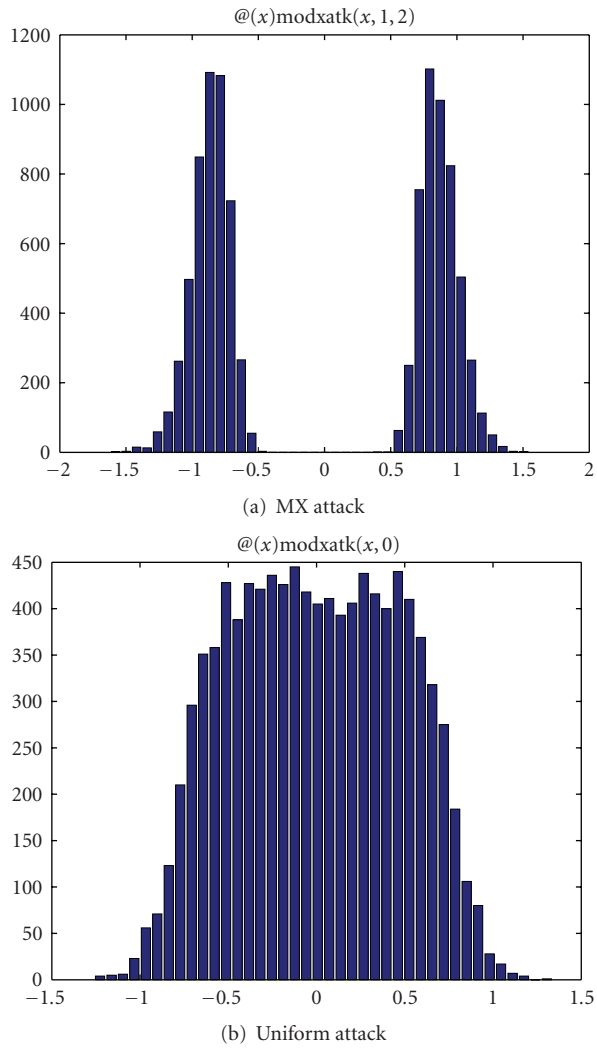


Figure 3: Histogram of a hybrid copies.

peak. An attack which produces a near-flat histogram seems the natural choice. Our proposal is to draw each hybrid sample a uniformly at random between the minimum and maximum observed. This is defined formally as follows.

Definition 4 (the uniform attack). The uniform attack ("uniatk") takes t watermarked signals $\mathbf{w}^{(u)}$, and produces a hybrid copy $\mathbf{z}$ where each sample z_i^{uni} is drawn independently and uniformly at random on the interval $[z_i^{\min}, z_i^{\max}]$.

Figure 3 shows example histograms of the MX and uniform attacks. We can clearly see how the MX attack gives a histogram resembling that of the RandNeg attack, while the uniform attack achieves the flatness sought.

Figure 4 shows simulations of the uniform attack compared to the MX attack. The important feature to note is that the behaviour is very similar for all the decoding options. The error rate is lower than for the MX decoder without preprocessor, but for the uniform attack the preprocessor does not help. Furthermore, as seen in Figure 2, the uniform

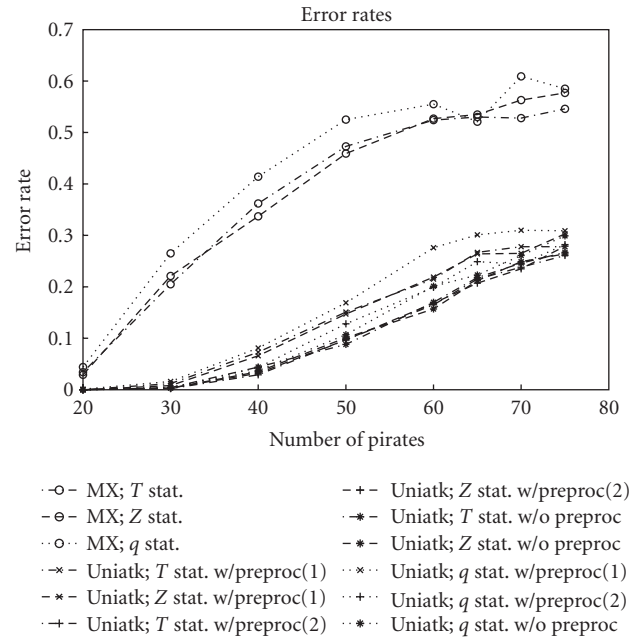


Figure 4: Comparing the uniform attack against MMX and the classics.

attack causes very little distortion. For large collusions it seems to have an excellent potential.

3.3. Hybrid attacks

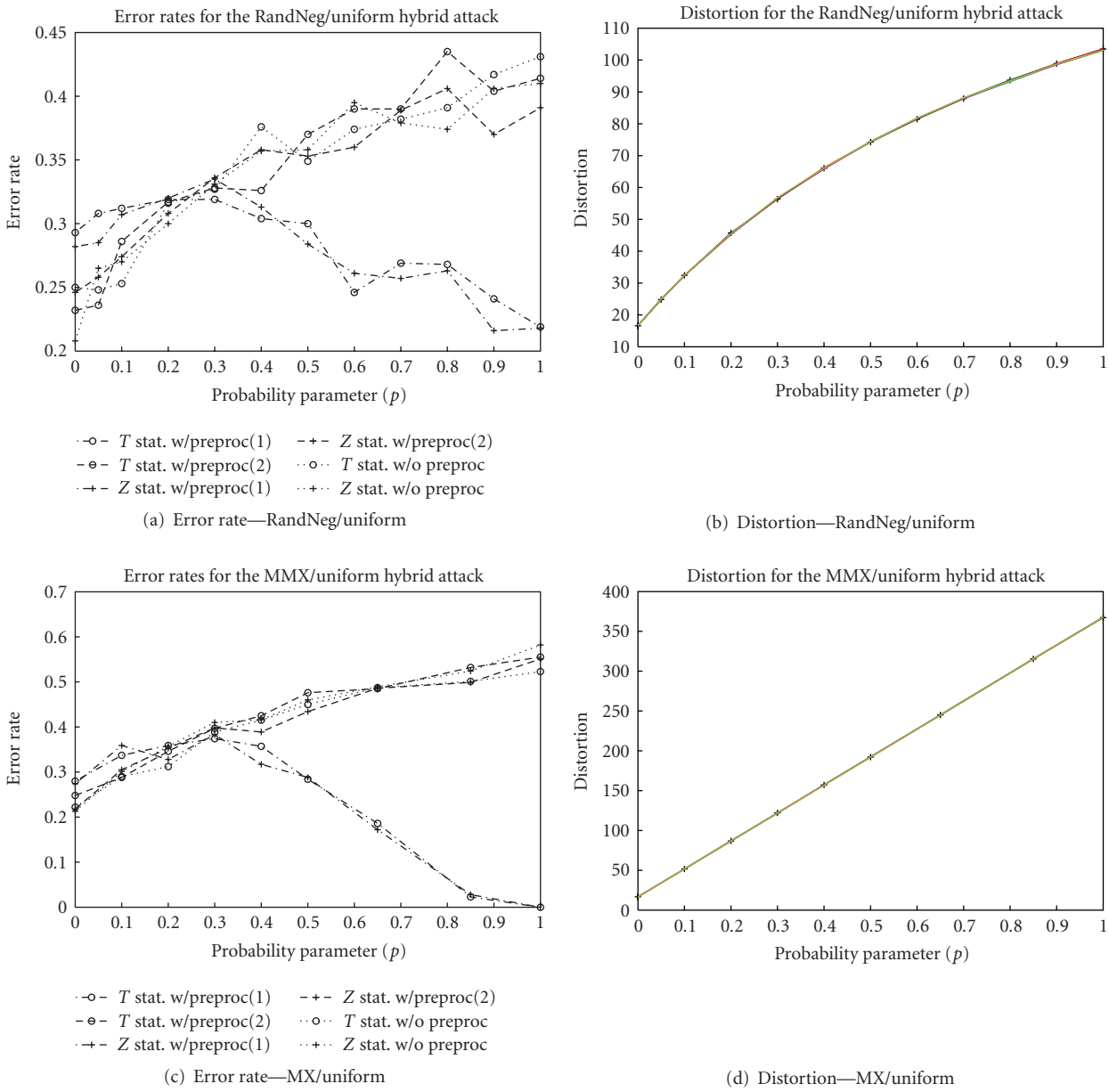
The uniform attack is the bluntest way to produce a flat histogram, and as we see, it breaks the preprocessing. An interesting question is if better attacks can be developed by combining the basic attacks already introduced. We introduce *hybrid attacks* as the attack is chosen independently for each sample according to some probability distribution.

In Figure 5, we have compared hybrid attacks which use the uniform attack with probability $1 - p$, and, respectively, the MMX or the RandNeg attacks with probability p. As expected there is a significant difference between one-peak and two-peak preprocessing, but the most interesting feature is that different decoding strategies are optimal for different p. The curves cross around $p = 0.3$. Typical histograms at for $p = 0.3$ are shown in Figure 6.

At the expense of increased distortion, these hybrid attacks allows us to increase the error rates compared to the pure uniform attack. This is true up to the point, where the histogram gets a distinctive two-peak shape and Preproc(1) becomes effective.

3.4. Hybrid attacks with MMX threshold

An alternative to the randomised hybrid attacks just described is to base the choice on a threshold. This is already part of the idea in the MMX attack. If the heuristic D_i is close to zero, an average attack is used, and otherwise the MX attack (minimum or maximum) is used. Obviously, other combinations are also possible, and we also introduce the



(a) Error rate—RandNeg/uniform

(b) Distortion—RandNeg/uniform

(c) Error rate—MX/uniform

(d) Distortion—MX/uniform

Figure 5: Comparing hybrid attacks for $t = 70$ colluders.

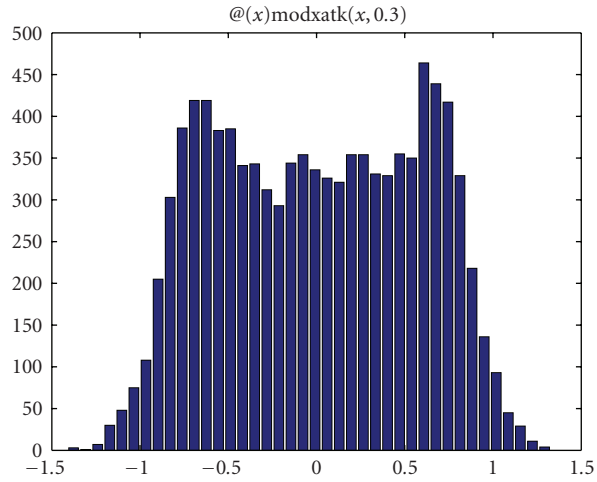
MMX-2 attack, where the average is replaced by the uniform attack.

In Figure 7, we have simulated the MMX with different thresholds. The result is similar to what we saw for the previous hybrid attacks, but even more pronounced. The single-peak preprocessor has no significant effect and has been excluded from the figure. The two-peak preprocessor is effective for small thresholds. The curves cross around $\theta = 0.08$.

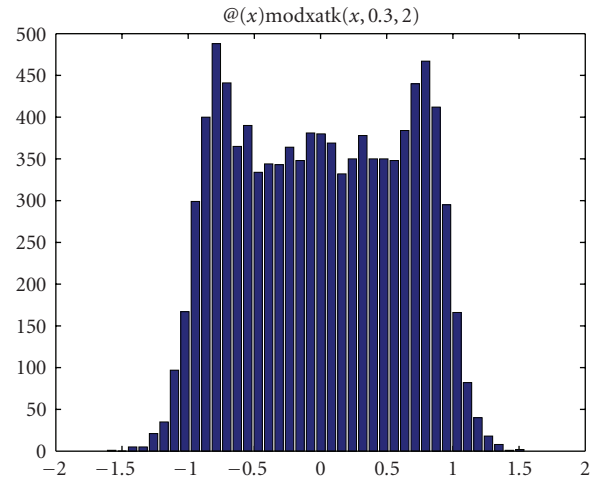
Typical histograms are shown in Figure 8 at $\theta = 0.1$. For the MMX-2 attack, we have the same flattish histogram as before, and no obvious approach preprocessing can be seen. However, for the regular MMX(-1) attack, we see a new pattern, with three peaks. It seems plausible that a preprocessor can be developed to decode correctly in this scenario, but, unless manual interference is acceptable, a strict definition of a peak would have to be developed.

4. THE NOISE ATTACK

He and Wu [5], citing [2], claim that "a number of nonlinear collusions can be well approximated by an averaging collusion plus additive noise." We did not find any explicit details on this claim in either paper, neither on the recommended noise distribution, nor on which nonlinear attacks can be so approximated. However, it is an interesting claim to explore.



(a) Uniform/RandNeg



(b) Uniform/MX

Figure 6: Histogram of hybrid copies from hybrid attacks with $p = 0.3$.

We consider the following two attacks:

$$\begin{aligned} &\text{averaging with Gaussian noise:} \quad z_i^{\mathrm{NG}} = \overline{z}_i + \alpha N_G, \\ &\text{averaging with Uniform noise:} \quad z_i^{\mathrm{NU}} = \overline{z}_i + \alpha N_U, \end{aligned} \tag{13}$$

where N_G is drawn from a standard normal distribution, and N_U is uniformly distributed on $[-1/2, 1/2]$. The first simulation, for $t = 70$ pirates, is shown in Figure 9. As we can see, both attacks are effective, but Gaussian noise causes enormous distortion.

To get a better picture, we plot the noise attacks against distortion in Figures 10 and 11. We have shown decoding of the noise attacks without preprocessor only; decoding with preprocessing is less effective. Supported by Figure 7, we decode the MMX attack without preprocessor only and MMX-2 without and with Preproc(1).

Three observations stand out as significant in this comparison:

(i) attacks with uniform noise are very effective for given distortion compared to other attacks,

(ii) attacks with Gaussian noise are considerably less effective than Uniform noise, and inferior to several other attacks studied,

(iii) for few pirates ($t = 35$) the distortion/error rate trade-off is much steeper for MMX-1 than for the noise attack, and it outperforms it at high distortion (150–200).

Now, if a three-peak Zhao et al. type preprocessor is used, the MMX-1 attack is likely to become ineffective.

We conclude that there may be some truth in the claims that averaging attacks with added noise are the most efficient attacks known to date. However, two important points have to be noted in this context. Firstly, the noise should not be Gaussian. We do not know if Uniform noise is optimal, or if an even better distribution can be found. Secondly, the preprocessor of Zhao et al. has to be developed further to be able to cope, automatically, with all the various attacks we have studied.

5. EVALUATION

In this section, we report additional simulations of the attacks which have proved most effective so far, to see how they compare under different conditions, that is, varying t, M, and n.

We have not include simulations with real images, because all the processes studied are oblivious to any added host signal. The detector is nonblind so any host added would be subtracted before detection. Also the attacks would be unaffected by the added host signal. Hence, simulations with real hosts would not give us any additional information.

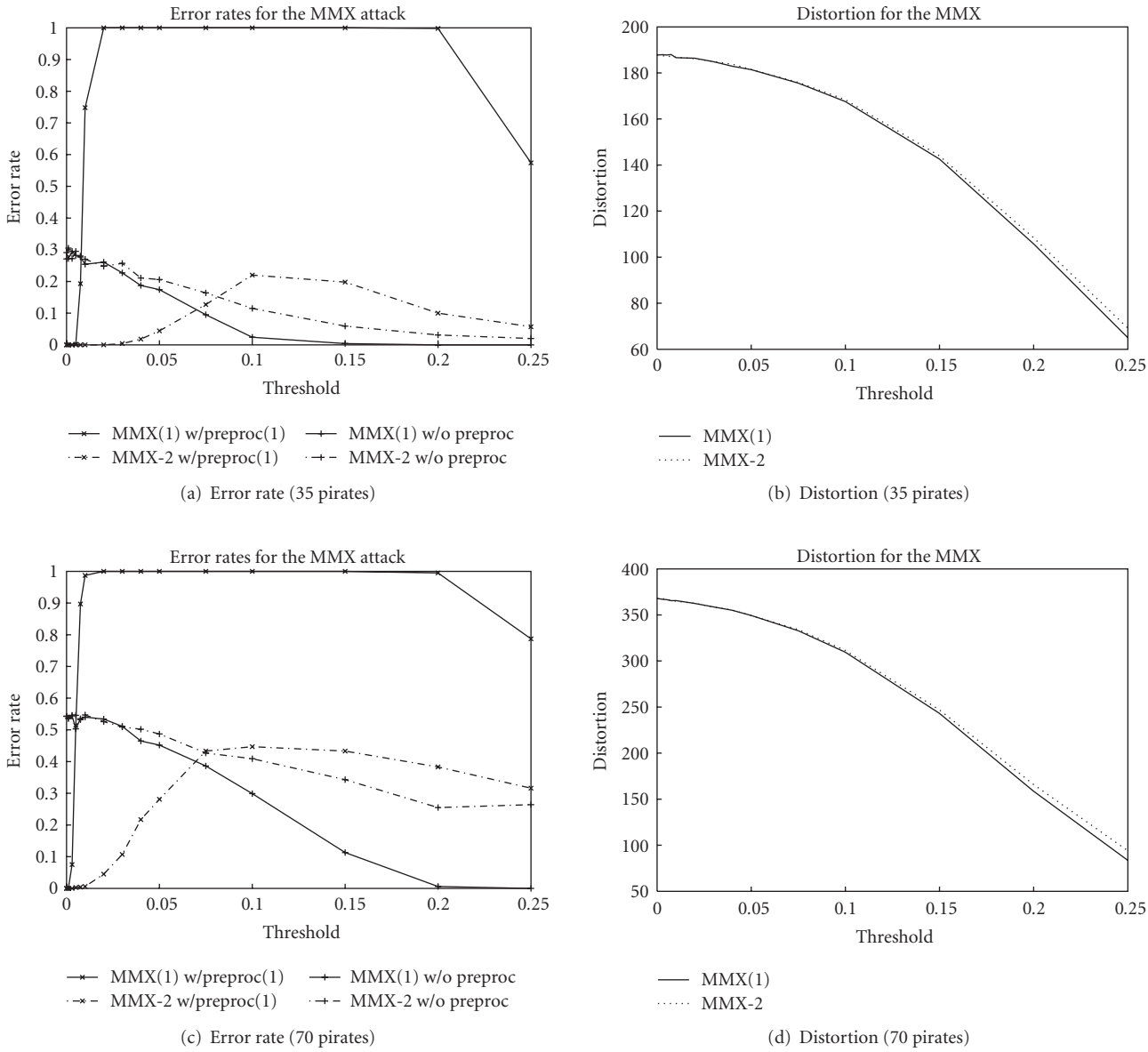
The constants, namely, the power of the fingerprint and the value of the Just Noticeable Difference would be scaled by the same factor according to perceptibility constraints in the same image. As stated, we have used the values suggested in [2], and a further study of these parameters is outside the scope of this paper.

None of the attacks discussed in Section 2.3, nor the MX attack, are effective against the preprocessor. Hence, the interesting attacks for further study are the hybrid attacks, the MMX attack with nonzero threshold, and the Uniform noise attack. The Uniform attack is a special case of the hybrid attack.

5.1. The Zhao et al. parameters

In this section, following [2], we assume $M = 100$ users. We have used Uniform noise with scaling factor 2.2, and Gaussian noise with power 0.47. The MMX-1 attack is with $\theta = 0.05$, and MMX-2 with $\theta = 0.08$. The hybrid attacks are with $p = 0.25$.

The results, shown in Figures 12 and 13, confirm what we have seen before. There is little difference between the



(a) Error rate (35 pirates)

(b) Distortion (35 pirates)

(c) Error rate (70 pirates)

(d) Distortion (70 pirates)

FIGURE 7: The MMX attack with different thresholds.

different decoders, and the best attacks achieve error rates $p_e \approx 6\%$ against the best decoder. It seems that the parameters of [2] suffice to ensure reasonable robustness against known nondesynchronising attacks. However, we have also confirmed that with our novel attacks, properly tuned, the preprocessing algorithm does not improve detection.

It is also confirmed that averaging with uniform noise is among the most efficient attacks. It is not feasible to run enough simulations to determine the optimal noise power or MMX thresholds for every number t of pirates. Thus, this simulation is insufficient to determine if one attack is strictly better under any given conditions.

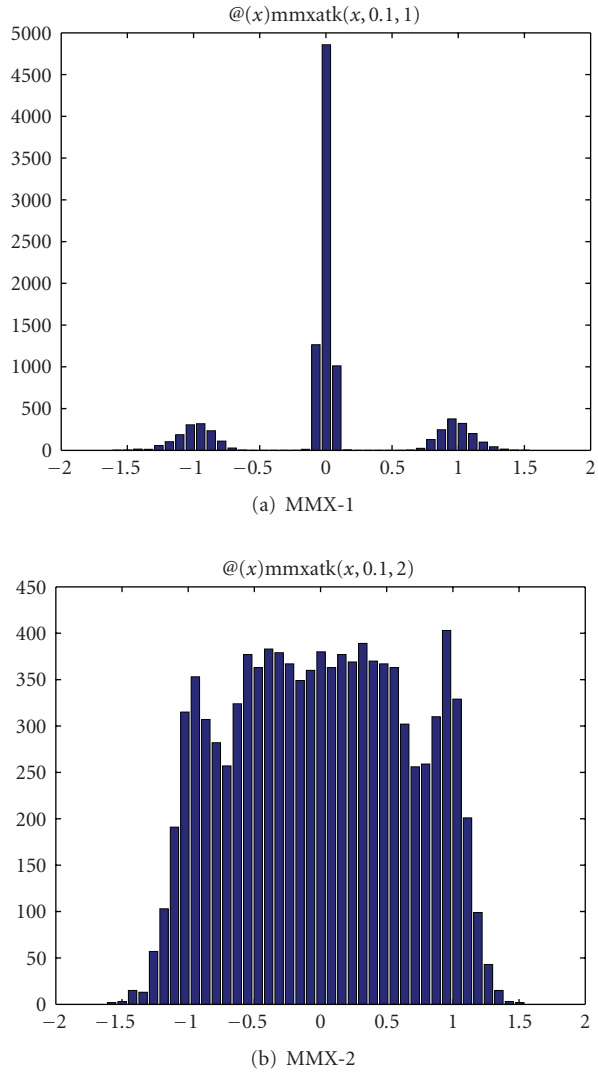
The choice of decoding heuristic seems to matter very little, although the q statistic is consistently outperformed. No clear distinction can be made between the Z and T statistics. In Figure 13 we show only Z decoding.

5.2. List decoding

Since list decoding is more popular than maximum heuristic decoding in the fingerprinting literature, we will have a brief look at this as well, for comparison.

We have seen that the Uniform noise attack (scale 2.2) gives an error rate of about 5% with $t = 70$ colluders using maximum heuristic decoding (3% at $t = 35$). The resulting $\mathrm{MSD}_{\mathrm{JND}}$ distortion (not normalised) is about 100–150. This is slightly less distortion than the RandNeg attack at $t = 70$ and slightly more at $t = 35$. Simulations are shown in Figure 14.

The experiment is conducted as follows. We generate a set $\mathcal{G}$ of t "guilty" codewords and a set $\mathcal{I}$ of $100 - t$ "innocent" codewords. The average of the "guilty" codewords is calculated and noise added, to give the received fingerprint $\mathbf{r}$.



(a) MMX-1

(b) MMX-2

Figure 8: Histogram of a hybrid copies with MMX attacks at threshold $\theta = 0.1$.

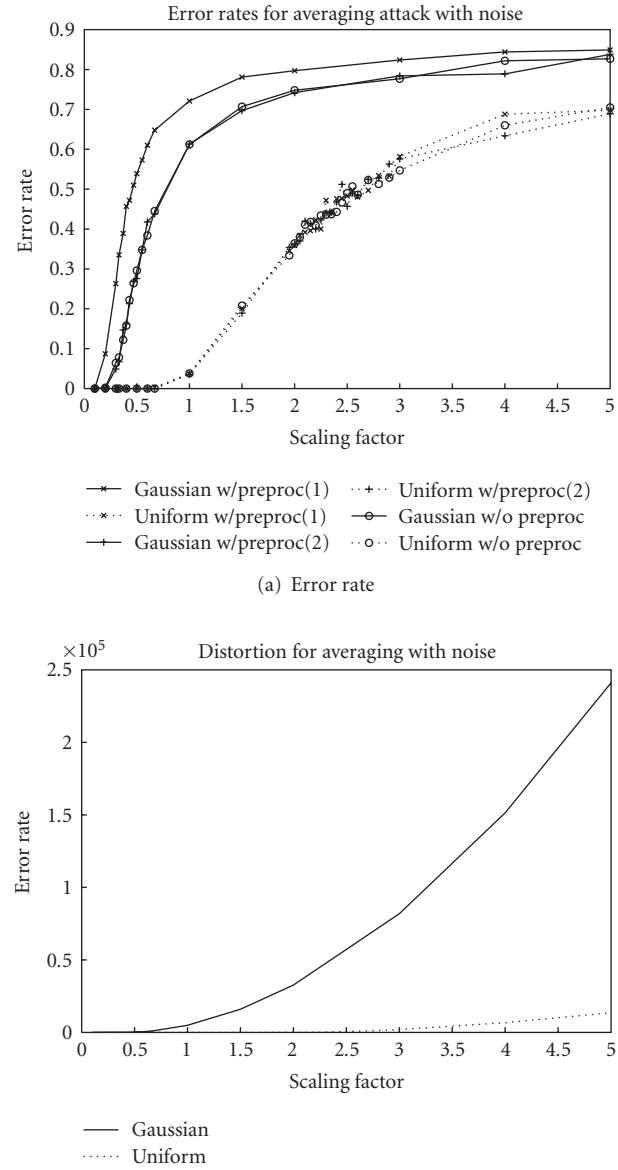
The Z statistic $Z(u)$ is calculated for every user $u \in \mathcal{G} \cup \mathcal{I}$. This experiment is repeated 2000 times, and for each iteration j we keep the following data:

$$G_j = \{Z(u) : u \in \mathcal{G}\}, \qquad I_j = \{Z(u) : u \in \mathcal{I}\},$$
$$g_j = \max_{u \in \mathcal{G}} Z(u), \qquad i_j = \max_{u \in \mathcal{I}} Z(u). \tag{14}$$

We estimate the expected number of false positives $E(P_F)$ and true positives $E(P_T)$ at a given threshold τ, as

$$\hat{E}(P_F) = \#\left\{h \in \bigcup_j G_j : h \geq \tau\right\},$$
$$\hat{E}(P_T) = \#\left\{h \in \bigcup_j I_j : h \geq \tau\right\}. \tag{15}$$

We have plotted $\hat{E}(P_F)$ against $\hat{E}(P_T)$ for varying threshold τ in Figure 14 (left-hand side).



(a) Error rate

(b) Distortion

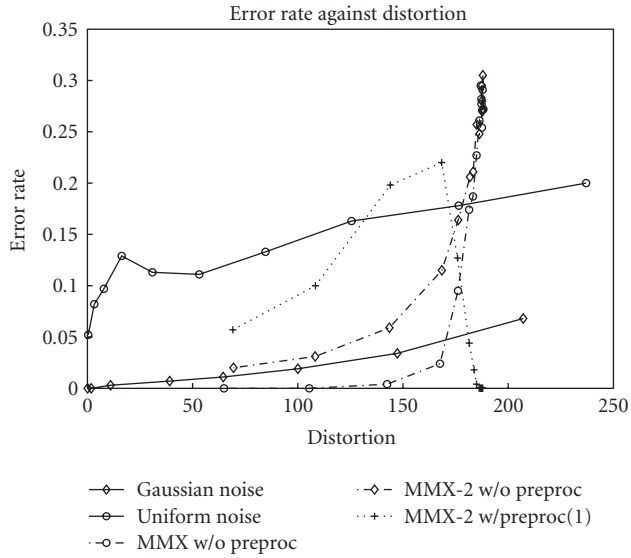
Figure 9: The averaging with noise attack by 70 pirates with different thresholds.

The probability p_c of at least one correct output and the probability p_f of at least one false negative are estimated as

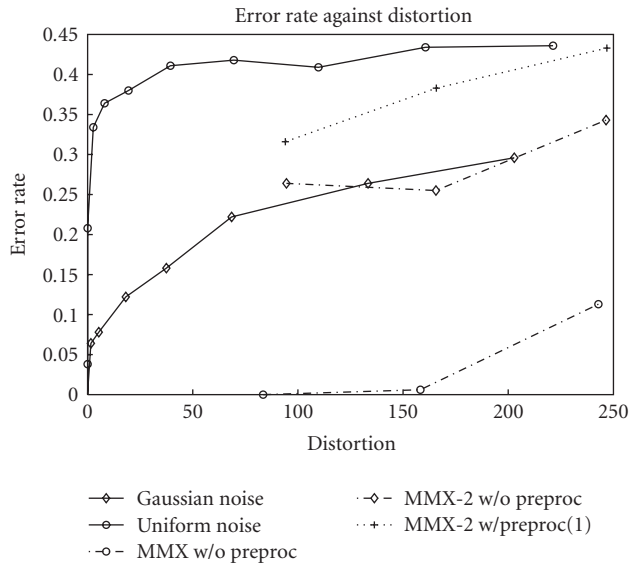
$$\hat{p}_c = \#\{j \mid i_j \geq \tau\}, \qquad \hat{p}_f = \#\{j \mid g_j \geq \tau\}. \tag{16}$$

Figure 14 (right-hand side) shows $\hat{p}_c$ plotted against $\hat{p}_f$ for varying thresholds.

As we can see, the different attacks have similar performances. We observe that with $t = 70$ and $p_f = 5\%$, we get only $p_c \approx 80\%$, even in the best case for the decoder. The noise attack gives $p_c \approx 70\%$. For $t = 35$ colluders and $p_f = 3\%$ we have $p_c \approx 80\%$ against the noise attack. It follows that the total error rate in the list decoding scenario is considerably worse than it is with maximum heuristic decoding.

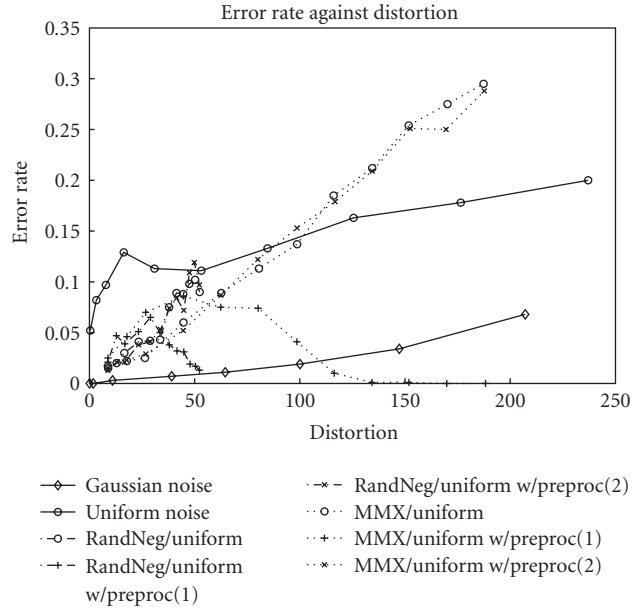


(a) 35 pirates

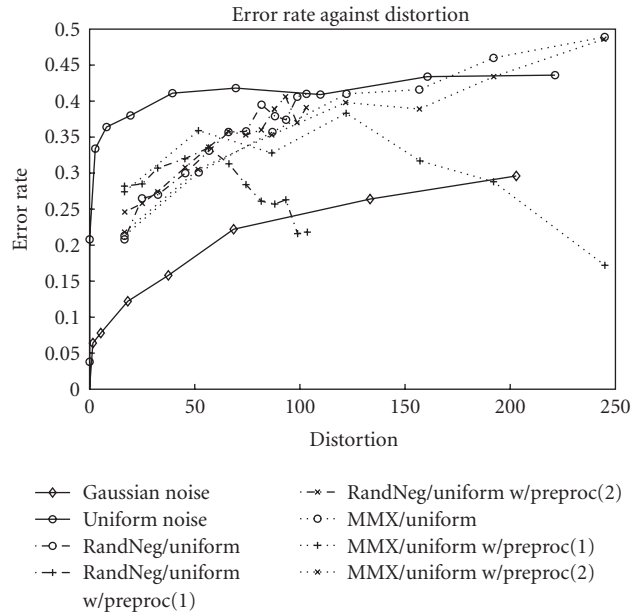


(b) 70 pirates

Figure 10: Noise versus the MMX attacks.



(a) 35 pirates



(b) 70 pirates

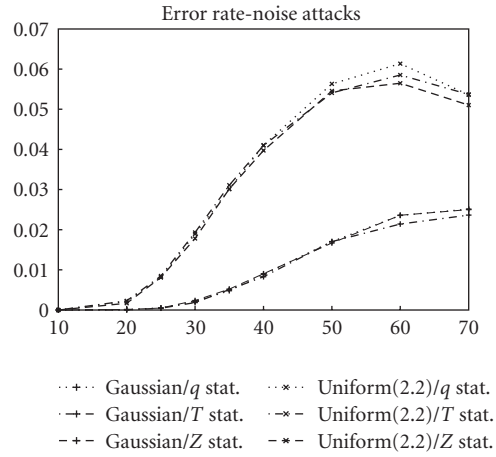
Figure 11: Noise versus hybrid attacks.

If we require $p_f \leq 1\%$ as assumed in [2], the detection rate for the noise attack at $t = 70$ is little more than 20%, and at $t = 35$ it is about 50%.

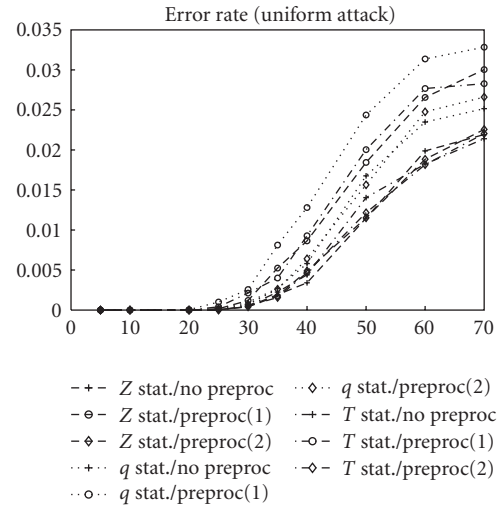
5.3. Scalability

So far we have considered very small codes, which are unlikely to be of practical use. One real application of fingerprinting is for the issue of screening copies for the academy awards ("Oscar"). (See, e.g., http://www.msnbc.msn.com/id/4037016.) In this scenario the number of users is in the order of 5000. It is hard to come up with real applications with fewer users, so we run one set of simulations for $M = 5000$. We assumed an averaging attack with uniform additive noise on the interval $[-1.1, 1.1]$.

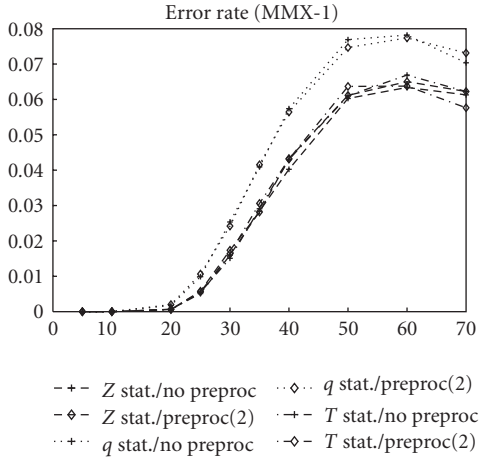
In coding theory and communications, it is normally expected that a well-designed code can scale freely keeping the rate $R = (\log M)/n$ constant. With $M = 5000$ users and the rate of the $(10\,000, 100)$ code, we get $n = 18\,500$. The result was an error rate of 71.2%, so evidently Gaussian fingerprinting does not scale well.



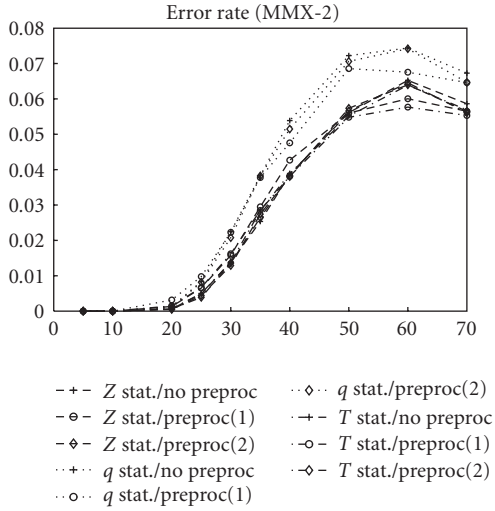
(a) Noise attacks



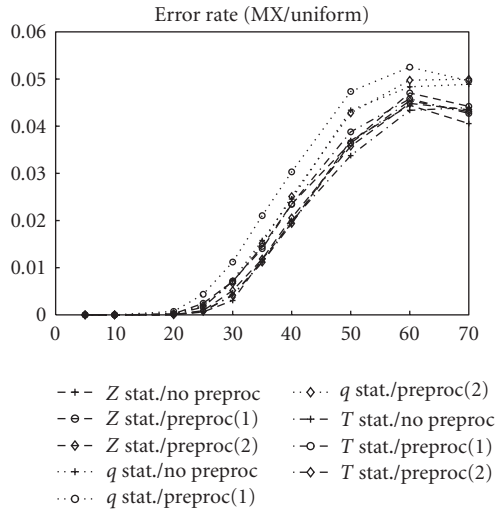
(b) Uniform attack



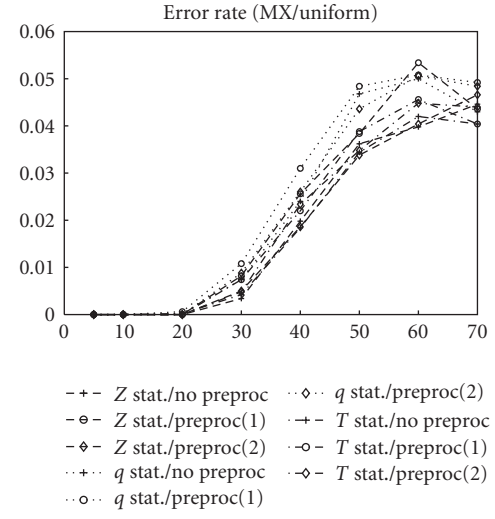
(c) MMX-1



(d) MMX-2

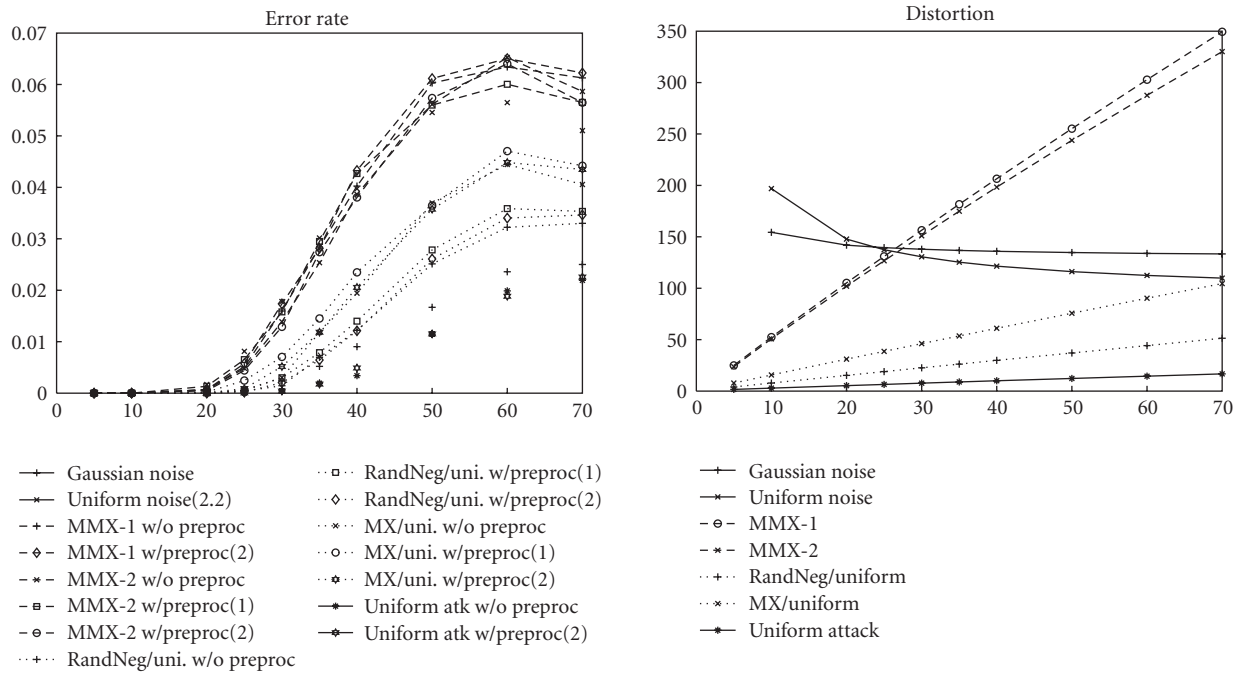


(e) Uniform/RandNeg hybrid



(f) Uniform/MX hybrid

Figure 12: Large simulation with $M = 100$ users and 25 000 tests.



(a) Error rate (b) Distortion

FIGURE 13: Large simulation with $M = 100$ users and 25 000 tests, with Z statistic decoding only.

TABLE 2: Error rates after the Uniform noise attack with $t = 70$ pirates, for various sequence lengths n and numbers M of users. Simulation based on 1000 samples.

(n, M)	R	P_e
$(10000, 100)$	$6.64 \cdot 10^{-4}$	6.6
$(18500, 5000)$	$6.64 \cdot 10^{-4}$	71.2
$(23500, 3000)$	$4.92 \cdot 10^{-4}$	52.1
$(21500, 1500)$	$4.91 \cdot 10^{-4}$	44.7
$(19500, 750)$	$4.90 \cdot 10^{-4}$	27.2
$(20000, 200)$	$3.82 \cdot 10^{-4}$	7.9
$(25000, 500)$	$3.59 \cdot 10^{-4}$	16.5
$(50000, 2000)$	$2.19 \cdot 10^{-4}$	11.6
$(50000, 1000)$	$1.99 \cdot 10^{-4}$	6.4
$(50000, 500)$	$1.79 \cdot 10^{-4}$	3.2

A larger range of code parameters are shown in Table 2. Admittedly, a small sample has been used, to get results in reasonable time, but the tendency is clear and consistent. Keeping constant rate, the error rate increases dramatically when the code size increases.

Codebooks of $nM \approx 10^8$ is close to the limits of what we can simulate with our current crude Matlab implementation on a 32-bit system. Codebook storage may very well be the limiting factor also for practical applications, even though somewhat larger codebooks could be made possible by a more efficient implementation. The largest codebooks that we have tried would use 400 Mb at single precision. Significantly larger codebooks would probably have to be generated on the fly by a pseudorandom number generator, so that only the seed has to be stored.

6. CONCLUSION

We have performed an extensive experimental analysis of collusive and noise attacks on Gaussian spread-spectrum fingerprinting, and introduced a couple of novel attacks. Below, we will itemise what we consider the main outcomes of our study, as well as the key questions left open.

6.1. Observations made

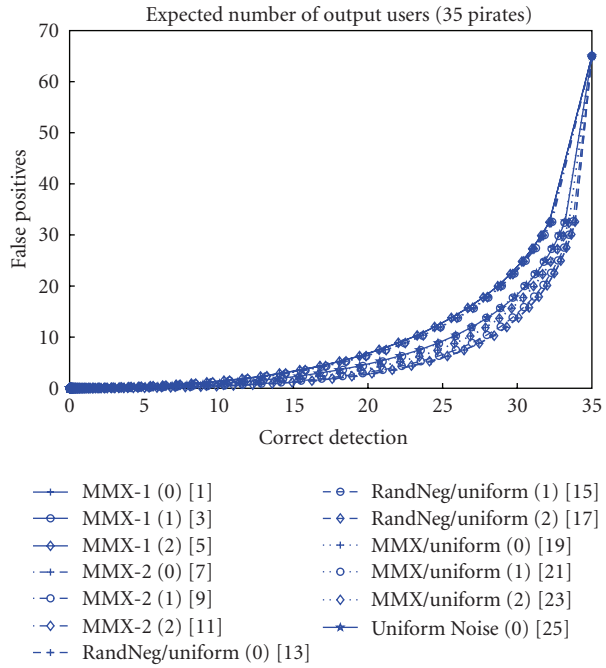
(i) The MX attack introduced in [3] is effective against common decoders with Z, T, or q statistics. However, it is not effective against the preprocessor of Zhao et al.

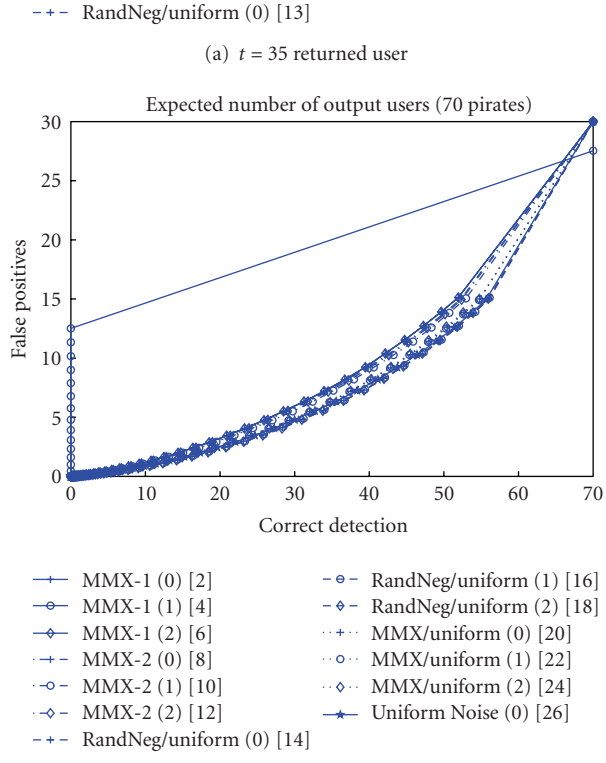
(ii) The parameters suggested by Zhao et al. appear to give a fairly robust system against known (nondesynchronising) attacks.

(iii) The uniform attack, as well as hybrid attacks based thereon, break the Zhao et al. preprocessor.

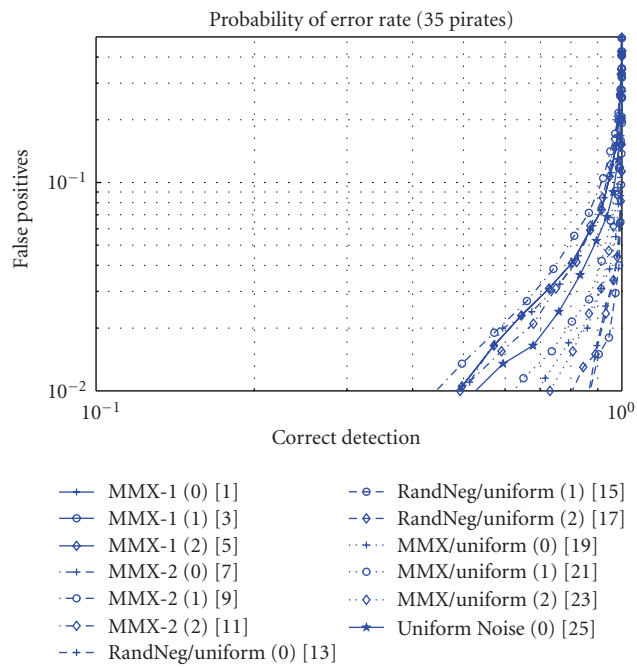
(iv) Averaging combined with Gaussian noise is not an effective attack compared with the other attacks studied.

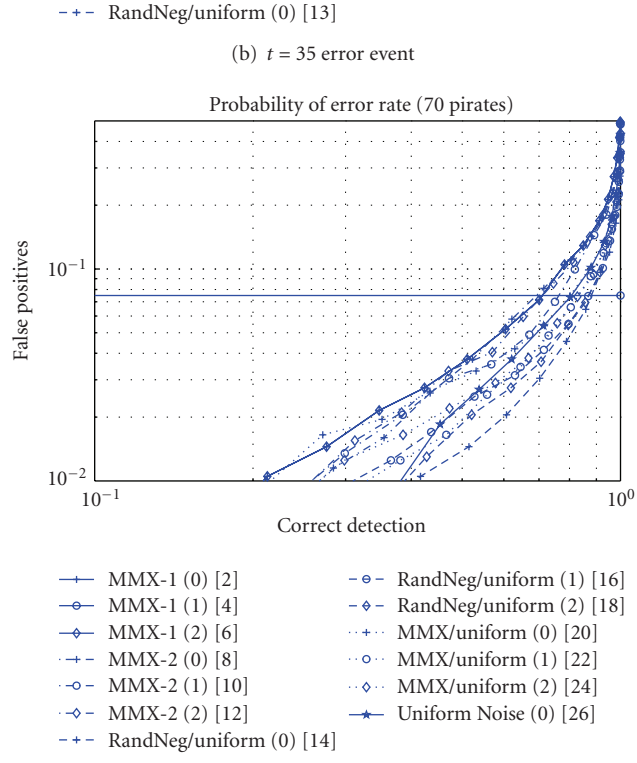
(v) Averaging combined with Uniform noise is very effective. It seems to outperform the other attacks considered under most, if not all, conditions.





(a) $t = 35$ returned user





(b) $t = 35$ error event

(c) $t = 70$ returned user

(d) $t = 70$ error event

Figure 14: List decoding performance. The left-hand figures show the probability of at least one true positive against the false positive rate. The right-hand figures show the average number of true and false positives for different thresholds.

(vi) Gaussian fingerprinting does not scale well from an information theoretic perspective.

Based on these observations, we conclude that the analysis of new fingerprinting schemes requires attention to a wider range of possible attacks than those considered in the literature. We have introduced a number of attacks worth mentioning, but we do not claim to have found them all.

6.2. Questions left open

The most interesting question left open by this study is a theoretical analysis of the attacks presented. This is expected

to be slightly harder than the analysis of previous attacks [2], leading to more complicated formulae.

From an applied viewpoint, a more important question is how a complete fingerprint system can be designed. Very little research exists on watermarking robust against desynchronising attacks, and nobody has yet considered a combination of collusion attacks and desynchronisation. In a real scenario, the attackers will have such attacks at their disposal in addition to what we have studied.

In our analysis we have been assuming that the correct preprocessor mode can be easily determined, and we also supposed that it can be extended for a three-peak histogram. At present this is, at best, true using manual inspection. Further research is needed to implement automatic histogram analysis and application of the optimal preprocessor. It is also an open question if sufficient information is available from the histogram.

Another open direction in research is the application of collusion-secure codes (e.g., [4, 6, 9]) in a practical watermark/fingerprint system. Since Gaussian fingerprints do not scale well, they may have to be combined with an outer, q-ary, collusion-secure code. In this case, q Gaussian sequences would be used to represent the q-ary symbols of the outer codewords.

REFERENCES

[1] N. R. Wagner, "Fingerprinting," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '83)*, pp. 18–22, Oakland, Calif, USA, April 1983.

[2] H. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu, "Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting," *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 646–661, 2005.

[3] H. G. Schaathun, "Attack analysis for He&Wu's joint watermarking/fingerprinting scheme," in *Proceedings of the 6th International Workshop on Digital Watermarking (IWDW '07)*, vol. 3304 of *Lecture Notes in Computer Science*, pp. 134–145, Springer, Guangzhou, China, December 2007.

[4] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[5] S. He and M. Wu, "Joint coding and embedding techniques for multimedia fingerprinting," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 231–247, 2006.

[6] H. G. Schaathun, "On error-correcting fingerprinting codes for use with watermarking," *Multimedia Systems*, vol. 13, no. 5-6, pp. 331–344, 2008.

[7] H.-J. Guth and B. Pfitzmann, "Error- and collusion-secure fingerprinting for digital data," in *Proceedings of the 3rd International Workshop on Information Hiding (IH '99)*, vol. 1768 of *Lecture Notes in Computer Science*, pp. 134–145, Springer, Dresden, Germany, September-October 1999.

[8] H. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu, "Nonlinear collusion attacks on independent fingerprints for multimedia," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '03)*, vol. 5, pp. 664–667, Hong Kong, April 2003.

[9] G. Tardos, "Optimal probabilistic fingerprint codes," *Journal of the ACM*, vol. 55, no. 2, article 10, pp. 1–24, 2008.

[10] H. G. Schaathun and M. Fernandez, "Boneh-Shaw fingerprinting and soft decision decoding," in *Proceedings of the IEEE Information Theory Workshop (ITW '05)*, pp. 183–186, Rotorua, New Zealand, August-September 2005.