Pit Stop for an Audio Steganography Algorithm

Andreas Westfeld, Jürgen Wurzer, Christian Fabian, Ernst Piller

Pit Stop for an Audio Steganography Algorithm

Andreas Westfeld[1], Jürgen Wurzer[2], Christian Fabian[2], and Ernst Piller[2]

[1] Dresden University of Applied Sciences, Germany
[2] St. Pölten University of Applied Sciences, Austria

Abstract. Steganography plays an important role in the field of secret communication. The security of such communication lies in the impossibility of proving that secret communication is taking place.
We evaluate the implementation of a previously published spread spectrum technique for steganography in auditive media. We have unveiled and solved several weaknesses that compromise undetectability.
The spread-spectrum approach of the technique under evaluation is rather unusual for steganography and makes the secret message fit to survive A/D and D/A conversions of analogue audio telephony, re-encoded speech channels of GSM/UMTS, or VoIP. Its impact to signal statistics, which is at least concealed by the lossy channel, is reduced. There is little published on robust audio steganography, its steganalysis, and evaluation, with the possible exception of audio watermarking, where undetectability is not as important.

Keywords: information hiding, steganalysis, spread spectrum BPSK, VoIP steganography

1 Introduction

Steganography is the art and science of invisible communication. Its aim is the transmission of information embedded invisibly into cover data. Secure watermarking methods embed short messages protected against modifying attackers (robustness, watermarking security) while the existence of steganographically embedded information cannot be proven by a third party (indiscernibility, steganographic security).

In general, steganographic communication uses an error-free channel, hence messages are received unmodified. Digitised image or audio files reach the recipient virtually without errors when sent, e.g., as an e-mail attachment. The data link layer ensures a safe, i.e., mostly error-free, transmission. If every bit of the cover medium is received straight from the source, then the recipient can extract a possibly embedded message without any problem. However, analogue audio telephony with A/D and D/A conversions, re-encoded speech channels of GSM/UMTS, and VoIP telephony use lossy compression or even do without a data link layer. This is because emerging errors have little influence on the (auditive) quality and can therefore be tolerated.

Without error correction, distortions are acceptable only in irrelevant parts of the cover signal. However, typical steganographic methods prefer these locations for hiding payload. The hidden message would experience the most interference in error-prone channels. Therefore, robust embedding functions have to add redundancy and change only locations that are carefully selected w.r.t. the proportion between unobtrusiveness and probability of error. This increases the risk of detection and permits only a small payload.

Information hiding techniques can be described in the classical triangle, i.e., a set of three characteristics: capacity, robustness, and undetectability. There are highly robust watermarking methods that offer small capacities and achieve perceptual transparency. Some watermarking methods are even robust against distortions in the time and frequency domains. Tachibana et al. introduced an algorithm that embeds a watermark by changing the power difference between the consecutive DFT frames [1]. It embeds 64 bits in a 30-second music sample. Compared to the proposed steganographic method this is a quarter of the payload in a host signal (cover) occupying 50 times the bandwith. It is robust against radio transmission. However, it was not designed to be steganographically secure and the presence of a watermark is likely to be detected by calculating the statistics of the power difference without knowing the pseudo random pattern. Van der Veen et al. published an audio watermarking technology that survives air transmission on an acoustical path and numerous other robustness tests while being perceptionally transparent [2]. The algorithm of Kirovski and Malvar [3] embeds about 1 bit per second (half as much as the one in [1]) and is even more robust (against the Stirmark Benchmark [4]). Arnold et al. presented an adaptive spread phase modulation (ASPM) that embeds an inaudible watermark with good robustness [5]. Although watermarking algorithms are perceptually transparent, they are not intended to be steganographically secure.

Examples for robust steganography are rather rare and, in most cases, embed into images. Marvel et al. [6] developed a robust steganographic method for images based on spread spectrum modulation [7]. This technique enables the transmission of information below the noise or cover signal level (signal to noise ratio below 0 dB). Likewise it is difficult to jam, as long as transmitter and receiver are synchronised. Therefore, successful attacks de-synchronise the modulated signal [8]. Further examples robustly embed messages using DSSS in slow scan television signals [9] or in auditive media.

This paper evaluates a particular implementation of spread spectrum technique for steganography in auditive media, introduced by Nutzinger et al. in 2010 [10] and implemented by Nutzinger and Wurzer in 2011 [11]. This technique survived several robustness tests, such as noise addition, variable time delay, frequency shifting, GSM coding, air transmission, cropping, and resampling. It also did not show significant changes of perceived distortion level in hearing tests comparing original and modified signals. Finally, the phase spectrum and the time and frequency representation did not show significant changes [11].

What is the goal of this paper? As the title suggests, it is not a description of an implementation of an audio embedding method that is claimed to be secure, just an evaluation of a previously known method from the literature. It might well be a bit more secure than before, under particular assumptions. We can set some of these assumptions as long as we want to play the attacker, but it would not say much about the security during a real application of the embedding method. It is probable that some of the attacks that we describe in this paper will be effective under certain conditions, and even successful through *other* steganographic (audio?) techniques. We unveil some weaknesses using rather simple methods that the implementors have not been aware of in their own validation of the embedding method. An evaluation at the given level of security—defined by the embedding method—does not require a universally working detector that is aware of all possible sources of stego signals, even if the steganographer could conceal some of the weaknesses using these sources. It is always advisable to identify the source of the weakness and revise the responsible part of the embedding method.

The paper is organised as follows. In the next section, the algorithm of the spread spectrum technique is described. Section 3 scrutinises the implementation of the spread spectrum technique. We found several weaknesses with proposed fixes in Sect. 4. Finally, Sect. 5 concludes the paper and gives an overview on our further work.

2 Spread spectrum algorithm

The steganographic algorithm of the StegIT-3 research project uses the audio signal of voice calls as its cover media. The voice call can be either a VoIP call or a mobile call over GSM or UMTS. The steganographic modulation for embedding the secret is applied at the decoded audio signal. The sample values of the uncompressed audio signal S_{float} are between the floating point values -1.0 and 1.0. If the encoded audio signal uses the PCM16 codec, it will be converted as shown below:

$$S_{\text{float}} = S_{\text{PCM16}}/32768.0 \tag{1a}$$

$$S_{\text{PCM16}} = \lfloor S_{\text{float}} \cdot 32768.0 + 0.5 \rfloor \qquad \text{if } S_{\text{float}} \geq 0 \tag{1b}$$

$$S_{\text{PCM16}} = \lceil S_{\text{float}} \cdot 32768.0 - 0.5 \rceil \qquad \text{if } S_{\text{float}} < 0 \tag{1c}$$

The implementation of the StegIT-3 framework had a rounding bug. For more information see section 4.1.

For embedding, the original unchanged decoded voice signal (cover signal) $c(t)$ is used. By default, the sample rate f_{s} of a phone call is 8000 Hz, but the algorithm implementation would also work with any higher sample rate. At the sender, each secret bit is embedded as a chip sequence. One pseudo-noise chip sequence represents the bit value *false* while the other represents the bit value for *true*. A chip is represented by the value -1 or 1 ($V_{\text{chip}}(t)$). These sequences

are generated by a linear feedback shift register (LFSR). Each chip of the chip sequence is embedded into the cover signal by the binary phase-shift keying (BPSK) modulation. The count of chips for one bit can be configured. It is a part of the stego key and also determines the transmission time for one embedded secret bit. The following equations show parameters for embedding a chip.

$$500.0\ \text{Hz} \leq f \leq 3000.0\ \text{Hz} \quad \text{BPSK modulation freq.} \tag{2a}$$
$$T = 1/f \quad \text{period time} \tag{2b}$$
$$C_{\text{opc}} = 3 \cdots 12 \quad \text{oscillations per chip} \tag{2c}$$
$$t_{\text{c}} = C_{\text{opc}} \cdot T \quad \text{chip period, chip time} \tag{2d}$$
$$V_{\text{chip}}(t) \quad \text{chip value } \{-1, 1\} \tag{2e}$$
$$t_{\text{start}} \quad \text{chip start offset} \tag{2f}$$
$$0 \leq \varphi \leq 2\pi \quad \text{phase for BSPK} \tag{2g}$$

For embedding the chip value $V_{\text{chip}}(t)$, the cover signal $c(t)$ is BPSK modulated according to Eq. 3, creating the modified (stego) audio signal

$$s(t) = c(t) + A_{\text{embed}}(t) \cdot V_{\text{chip}}(t) \cdot \cos(2\pi \cdot f \cdot t + \varphi). \tag{3}$$

$A_{\text{embed}}(t)$ and $V_{\text{chip}}(t)$ are constant for the embedding of one chip (chip time). The challenge of the algorithm is to find the perfect value for A_{embed}. The value of A_{embed} represents the amplitude for the BPSK modulation of one chip and affects the ability of the receiver to successfully extract the chip. A higher amplitude—while enhancing the quality of extraction—has negative impacts on the security of the steganographic algorithm. This algorithm uses a constant modulation frequency. An advanced version of this algorithm is described by Nutzinger [10]. Figure 1 shows the BPSK modulation of chips and Fig. 2 the cover and the stego signal with the added BPSK modulation chip signal.

The receiver has to extract the chips from the audio signal. The demodulated chip value at the receiver side is given by

$$V_{\text{chip,ext.}} = \begin{cases} 1 & \text{if } d \geq 0 \\ -1 & \text{if } d < 0 \end{cases}, \tag{4}$$

$$\text{with} \quad d = \int_{t_{\text{start}}}^{t_{\text{start}}+t_{\text{c}}} s(t) \cdot \cos(2\pi \cdot f \cdot t + \varphi)\, \mathrm{d}t. \tag{5}$$

3 Attacks

The goal of this project was to judge the security of the embedding algorithm. It is sensible to study all information before trying to detect traces of the embedding process in the output signal, however, we (involuntarily) played the attacker in two different setups. Due to an intellectual property issue, neither the C++ source code, nor the binary of the implementation was available in the first phase. We could only get a small number of WAV files (recorded phone calls),

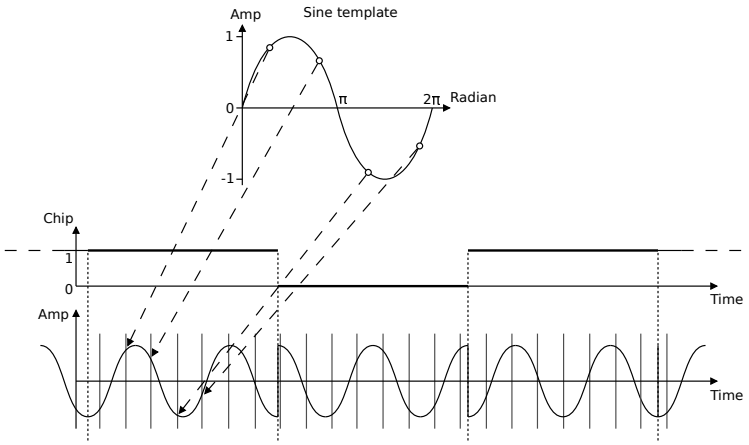

Fig. 1. Generation of the chips

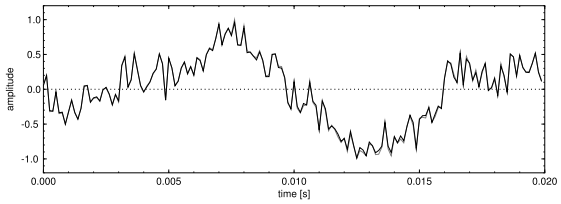

Fig. 2. Cover signal (dark line) and stego signal (light line)

each file in several versions (without any message, with an embedded message in three different embedding configurations: "amp," "bpsk," and "phase," in order to find out which one is the best). We also knew that the messages had been embedded with some kind of spread spectrum modulation.

3.1 Twin Peaks?

We started our research with the simplest attack we could think of, namely a histogram attack à la *Twin Peaks* [12], since the embedding method uses spread spectrum modulation (SS). Not knowing the exact implementation of the SS method, we assumed a simple direct sequence SS (DSSS) algorithm. We hoped that at least one of the three configurations ("amp," "bpsk," and "phase") is close enough to our vague assumption. Figure 3 gives a concrete example of the assumed simple SS embedding[3] with a (zero mean) Gaussian cover signal. A PN sequence, consisting of random samples -1 and 1 only, is used to spread one symbol (e.g. a bit) over a longer time period. If this spreading sequence is added

[3] which is indeed hard to match to the real implementation described in Sect. 2

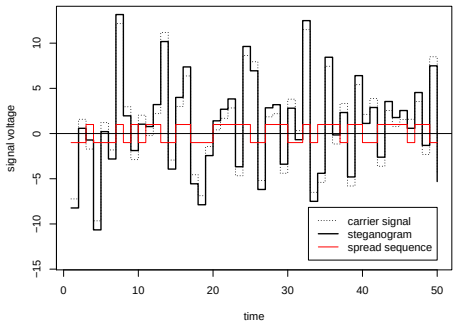

Fig. 3. A spreading PN sequence is added to the cover signal (dotted line), resulting in the stego signal (bold line)

to the cover signal, the symbol can be decoded as the scalar product of the spreading sequence and the stego signal. If the histogram of the cover signal has one peak, there might be two peaks in the histogram of the stego signal. (Hence the name of the attack.) Since the cover signal is Gaussian, the stego signal is the sum of two Gaussian distributions, with a mean distance determined by the spreading sequence $(1 - (-1) = 2)$. If the variance of the cover signal is large (cf. Fig. 4, left), e.g. a louder part of a phone call, the resulting distribution is

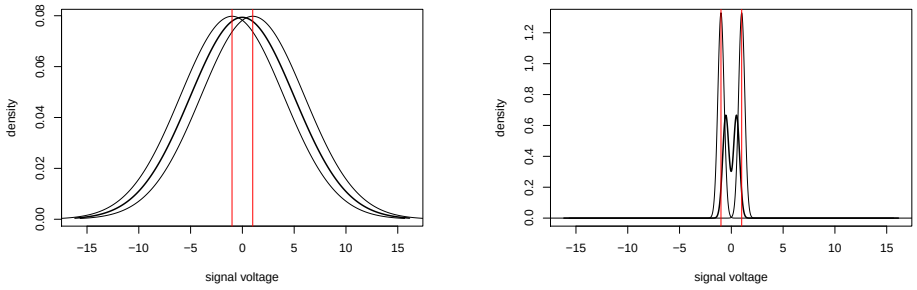

Fig. 4. If the cover signal is strong enough, the resulting composite distribution (bold line) still seems to be Gaussian (left), however, quiet passages of the stego signal might show twin peaks (right)

hard to distinguish from the original Gaussian. However, in more quiet passages of the phone call, the resulting distribution might show twin peaks (cf. Fig. 4, right), or a noticable change of the distribution exploitable for the detection of the steganographic method. We expected at least a kind of automatic volume control of the spread sequence, which reduces the amplitude according to the

cover signal. However, since the embedding method should be useful for phone calls, real-time properties are a concern. It may be that the control is delayed, in order not to delay the speech signal too much.

Surprisingly, our rather blind attack separated (even parts of) cover and stego signal in our test database perfectly. The detector worked in two steps. The first step selected quiet parts of the signal (a dispensable step as we will see later), the second created a histogram, which showed a single peak at zero for stego signals, but not for cover signals. While the cover signal contained about the same number of zeros as ones (maybe up to 30 % more), in stego signals we found twice as many zeros. Interestingly, this worked for the configurations "amp," "bpsk," and "phase" with the same threshold. If there are more than 1.5 as many zeros than ones, the signal is a detected stego signal.

To understand the reason, we had to wait until we finally received the source code (cf. Sect. 4.1).

3.2 "Steps"

A cover–stego attack is possible here, i.e., the synchronous confrontation of cover samples c_i and their corresponding stego samples s_i. The closer the samples to the diagonal $s_i = c_i$, the smaller the change caused by the embedding. Figure 5 opposes cover and stego samples, resulting in an overlay of diagonal stripes.

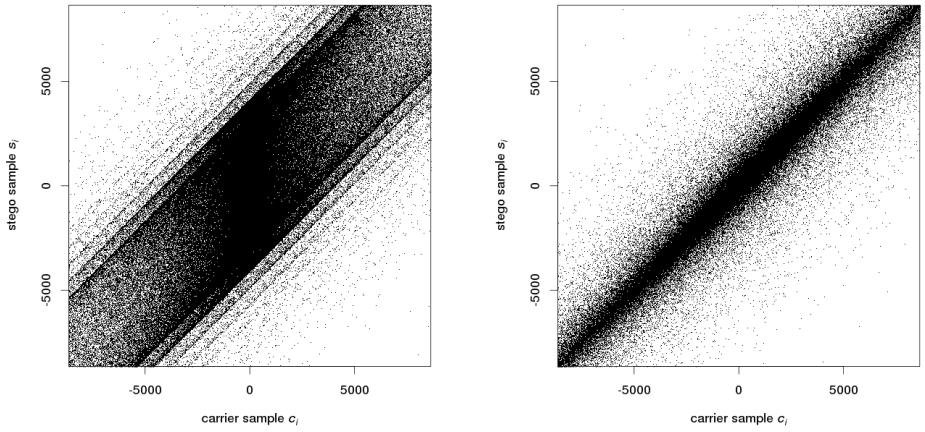

Fig. 5. Synchronous confrontation of cover and stego samples shows steps of a controller (left), and no steps after correction (right)

Obviously, there is a mechanism that controls the embedding intensity in discrete steps. However, under more realistic conditions (stego only attack), an attacker has to estimate the cover from the stego signal. This is usually called "denoising."

3.3 Saturn Sighted

We could model the cover sample from other, but temporally close stego samples:

$$c_i \sim s_{i-2} + s_{i-1} + s_{i+1} + s_{i+2}. \tag{6}$$

A similar approach was used during the BOWS-2 contest to estimate the unmarked magnitude of wavelet coefficients from its surrounding [13]. This has been successful, because the piece of watermark in s_i was independent of the watermark in the samples of the surrounding. Unfortunately, we cannot be sure or even expect this property in the case of the attacked audio signal here, since a chip time could be longer than a sample time. Nevertheless, we simply predicted the next cover sample by the current stego sample

$$\hat{c}_{i+1} = s_i \tag{7}$$

leading to the impressive, "astronomic" constellation in Fig. 6.

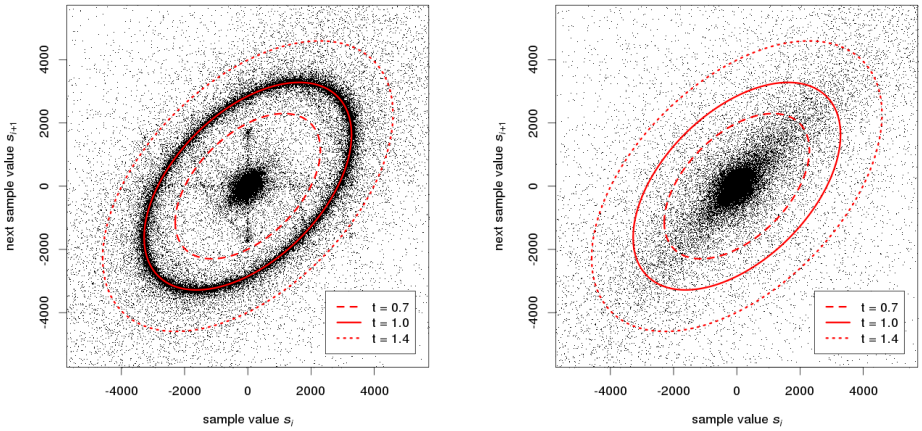

Fig. 6. Confrontation of consecutive stego samples (next sample value as an estimate of cover) shows a Lissajous curve (left), and after correction (right)

Although resembling the planet Saturn, it is technically a Lissajous curve that can be estimated using the following ellipse:

$$\begin{aligned} t^2 &= \left(\frac{y - mx}{b}\right)^2 - \frac{x^2}{a^2}, \quad \text{with} \\ a &= 3280, \\ b &= 2850, \\ m &= 0.496, \end{aligned} \tag{8}$$

where t is a threshold parameter. Lissajous curves appear on an oscilloscope in X–Y mode, when the two inputs are sinusoidal signals. Here the two inputs come from the same, but time shifted signal, i.e. have the same frequency but different phase, resulting in an ellipse. The phase shift is determined by the time between two consecutive samples, i.e., depends on the sample rate. A detector can be constructed, which assumes a stego signal if a suspicious amount of samples occurs between the dashed ellipse ($t = 0.7$) and the dotted one ($t = 1.4$), compared to the total number of samples.

The strength of the effect might be surprising. It is *not* the usual parameterisation, but one for testing the algorithm's behaviour on a GSM channel. In this test case it is of course audible. However, even with the "production" parameters, a small ring ($a \approx b \approx 30$: ;-) can be isolated in some quiet passages of the stego signal (and is covered by the dark centre of Fig. 6 otherwise). We admit that we could have missed this fact in the rest of the test cases if the accidental GSM test case would not have been included in our test database setup.

4 Countermeasures

4.1 Solitary Peak

The cover signal is read from an audio source, usually a telephone, sound card or WAV file (e.g., 8000 PCM samples per second, 16 bits per sample, 1 channel). The samples are signed integers. To support different rates and precisions, the embedding algorithm internally maps the raw data to a series of normalised `double` floating point samples in the range $-1 \ldots 1$.

The conversion is implemented by a type cast from `double` to integer, followed by scaling down to the desired interval $[-1, 1]$. Finally, the `double` values are scaled up (and clipped, if necessary) to the original range, and casted back to integer. If the values are not changed in between by the embedding step, the final cast from `double` to integer is one-to-one, because the fractional part is zero.

However, if something is embedded, the final `double` values will also have non-zero fractional parts. The obvious, but careless, use of type cast takes revenge here. The cast operator (`y = (int)x;`) takes a numeric argument $x \in \mathbb{R}$ and returns an integer $y \in \mathbb{Z}$, formed by truncating the values in x toward 0 (cf. Fig. 7). Possible repair: `y = (int)(x + ((x < 0) ? -0.5 : 0.5));`

After the problem of spotty rounding around zero was solved, another problem was detected that occurs with some cover signals only, because of the asymmetry of the integer domain. There is an even number of 16 bit integers, one is neutral (0), 32767 are positive, 32768 are negative. If we negated -32768 (`0x8000`) there would be a sign overflow, resulting in the same (negative) value. The implemented mapping to $[-1.0, 1.0]$ divided all samples by 32767 and clipped -32768 to -1.0. If the signal is sufficiently saturated, there will be peaks in the histogram of cover samples for the saturated values $(-32768, 32767)$. The conversion routines of the mapping shifted the peak at -32768 to -32767. In case

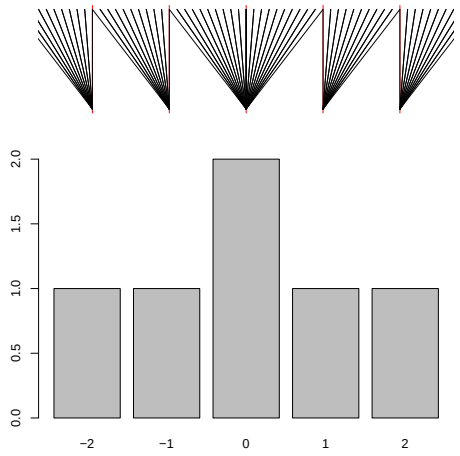

Fig. 7. Bin 0 collects from a double-width interval due to truncation towards zero, resulting in a histogram peak

of saturation this provides a rather safe feature for detection. The obvious repair is to change the divisor to 32768, and to clip positive values above 32767 when mapping back to integer.

4.2 Better amplitude adjustment control for BPSK modulation

At the sender's side, the embedding algorithm determines whether the original cover signal is suitable for embedding the secret chip. In case decoding the original audio signal would result in the secret chip's value, it is not necessary to modify the cover signal to embed the chip. This is shown in Eq. 9. The value of c_{embed} decides if a modification of the cover signal is necessary (cf. Eq. 10).

$$c_{\text{embed}} = \int_{t_{\text{start}}}^{t_{\text{start}}+t_{\text{c}}} c(t) \cdot V_{\text{chip}}(t) \cdot \cos(2\pi \cdot f \cdot t + \varphi)\,\mathrm{d}t \tag{9}$$

$$\text{steganographic modification is } \begin{cases} \text{not necessary} & \text{if } c_{\text{embed}} > 0 \\ \text{necessary} & \text{if } c_{\text{embed}} \leq 0 \end{cases} \tag{10}$$

In case of embedding, the value of A_{embed} is increased step-by-step until a chip can be extracted correctly at the receiver's side. However, this approach shows steps when plotting the cover against the stego signal (cf. Fig. 5). Also a fix minimum amplitude was added. The fix minimum amplitude increases the effect of "Saturn Rings" (cf. Sect. 3.3). In order to avoid "Saturn Rings" and "steps," the amplitude A_{embed} is determined by a modified control mechanism (cf. Eq. 11 . . . 15).

$$c_{\text{avg}} = \frac{1}{t_{\text{c}}} \int_{t_{\text{start}}}^{t_{\text{start}}+t_{\text{c}}} c(t) \cdot V_{\text{chip}}(t) \cdot \cos(2\pi \cdot f \cdot t + \varphi)\,\mathrm{d}t \tag{11}$$

$$\text{steganographic modification is} \begin{cases} \text{not necessary} & \text{if } c_{\text{avg}} > 0 \\ \text{necessary} & \text{if } c_{\text{avg}} \leq 0 \end{cases} \tag{12}$$

$$\int_{t_{\text{start}}}^{t_{\text{start}}+t_{\text{c}}} c(t) - 2 \cdot c_{\text{avg}} \cdot V_{\text{chip}}(t) \cdot \cos(2\pi \cdot f \cdot t + \varphi)\,\mathrm{d}t = 0 \tag{13}$$

$$s_{\text{ARV}} = \frac{1}{t_{\text{c}}} \int_{t_{\text{start}}}^{t_{\text{start}}+t_{\text{c}}} |c(t)|\,\mathrm{d}t \tag{14}$$

$$A_{\text{embed}} = 2 \cdot |c_{\text{avg}}| + s_{\text{ARV}} \cdot A_{\text{add}} \tag{15}$$

The signal mean c_{avg} (cf. Eq. 11) determines whether embedding is necessary. This value is the basis in the new definition of the embedding amplitude A_{embed} (cf. Eq. 15). For a successful chip extraction at the receiver side, A_{embed} must be greater then the double of c_{avg}. The embedding amplitude A_{embed} is increased by an additional part A_{add} that is scaled with the averaged rectified values in the signal segment s_{ARV} to provide the receiver the necessary margin for extraction of the correct chip value. The scaling relieves the "Saturn" effect (cf. Sect. 3.3) compared to the initial choice of a constant margin (e.g., an unscaled $A_{\text{add}} = 0.1$).

5 Conclusions

We found several weaknesses in the implementation of a spread spectrum technique for steganography in auditive media. Some of them did not result from the embedding technique itself, but the mapping from the external cover representation to the internal working representation. It seems to be important to carefully check conversion and normalisation functions, their homogeneity around special values like 0, and their properties in case of saturation.

However, also the embedding algorithms itself showed weaknesses. It seems to be important to consider the difference between cover signal and stego signal during the design of the algorithm. Although an average attacker cannot access this difference signal, obtrusive properties might radiate through the cover's shielding guard. It is also advisable to use pathologic signals, like rhythmic audio pulses, to test the integrity, for instance, of control mechanisms.

Be aware of correlations within the cover's values. Such correlations will also occur in the stego signal. Often, such correlations can be used to "denoise" the signal or to "calibrate" statistics [14, 15], even in audio streams.

Acknowledgments This work was supported in the KIRAS programme for security research by the Austrian Federal Ministry for Transport, Innovation and Technology.

References

1. Tachibana, R., Shimizu, S., Nakamura, T., Kobayashi, S.: An audio watermarking method robust against time- and frequency-fluctuation. In Delp III, E.J., Wong, P.W., eds.: Security, Steganography and Watermarking of Multimedia Contents III (Proc. of SPIE), San Jose, CA (2001) 104–115
2. van der Veen, M., Bruekers, F., Haitsma, J., Klaker, T., Lemma, A.N., Oomen, W.: Robust multi-functional and high-quality audio watermarking technology. In: 110th Audio Engineering Society Convention. Volume Convention Paper 5345. (2001)
3. Kirovski, D., Malvar, H.S.: Spread-spectrum watermarking of audio signals. IEEE Trans. on Signal Processing **51** (2003) 1020–1033
4. Steinebach, M., Petitcolas, F., Raynal, F., Dittmann, J., Fontaine, C., Seibel, S., Fates, N., Ferri, L.: StirMark benchmark: audio watermarking attacks. In: International Conference on Information Technology: Coding and Computing. (2001) 49–54
5. Arnold, M., Baum, P.G., Voeßing, W.: A phase modulation audio watermarking technique. In Katzenbeisser, S., Sadeghi, A.R., eds.: Information Hiding (11th International Workshop). Volume 5806 of LNCS., Berlin Heidelberg, Springer-Verlag (2009) 102–116
6. Marvel, L.M., Boncelet, C.G., Retter, C.T.: Spread spectrum image steganography. IEEE Transactions on Image Processing **8** (1999) 1075–1083
7. Pickholtz, R.L., Schilling, D.L., Milstein, L.B.: Theory of spread-spectrum communications—a tutorial. IEEE Transactions on Communications **30** (1982) 855–884
8. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Attacks on copyright marking systems. In Aucsmith, D., ed.: Information Hiding (2nd International Workshop). Volume 1525 of LNCS., Berlin Heidelberg, Springer-Verlag (1998) 219–239
9. Westfeld, A.: Steganography for radio amateurs—a DSSS based approach for slow scan television. In Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P., eds.: Information Hiding (8th International Workshop). Volume 4437 of LNCS., Berlin Heidelberg, Springer-Verlag (2007) 201–215
10. Nutzinger, M., Fabian, C., Marschalek, M.: Secure hybrid spread spectrum system for stegnanography in auditive media. 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP) (2010) 78–81
11. Nutzinger, M., Wurzer, J.: A novel phase coding technique for steganography in auditive media. 6th International Conference on Availability, Reliability and Security (ARES) (2011) 91–98
12. Maes, M.: Twin Peaks: The histogram attack to fixed depth image watermarks. In Aucsmith, D., ed.: Information Hiding (2nd International Workshop). Volume 1525 of LNCS., Berlin Heidelberg, Springer-Verlag (1998) 290–305
13. Bas, P., Westfeld, A.: Two key estimation techniques for the broken arrows watermarking scheme. In: Proc. of ACM Multimedia and Security Workshop, Princeton, NJ, USA (2009) 1–8
14. Fridrich, J., Goljan, M., Hogea, D.: Steganalysis of JPEG images: Breaking the F5 algorithm. In Petitcolas, F.A.P., ed.: Information Hiding (5th International Workshop). Volume 2578 of LNCS., Berlin Heidelberg, Springer-Verlag (2003) 310–323
15. Kodovský, J., Fridrich, J.: Calibration revisited. In: Proc. of ACM Multimedia and Security Workshop, Princeton, NJ, USA (2009) 63–73