Design and Evaluation of Steganography for Voice-over-IP

Jana Dittmann, Thomas Vogel and Reyk Hillert
Advanced Multimedia and Security Lab (AMSL)
Otto-von-Guericke-Universität
Magdeburg, Germany
Email: {jdittman, tvogel, hillert}@iti.cs.uni-magdeburg.de

***Abstract*—According to former results from [1] in this paper we introduce design principles and first experimental test results of a Voice-over-IP (VoIP) framework including a steganographic channel. We show that using this framework it is largely secure to transmit hidden messages during a VoIP session and demonstrate first results with respect to perceptibility, detection probability and reliability.**

I. INTRODUCTION

For digital images as wells as digital audio there exist many steganographic techniques [3][4][5], and furthermore there exist a lot of approaches to detect steganography in digital images [6][7][8]. However, for detection of hidden messages within audio data there are only few methods published even though novel technologies like Voice-over-IP (VoIP) provide a new field for applied steganography. The term "VoIP" describes the digitalization compression and transmission of analogue audio signals (in the majority of cases speech) from a sender to a receiver using IP packets. The receiver applies the reverse process and gets the reconstructed audio signal. After that he can act as the sender. For transmission the size of the used network and the distance between communication partners are of little relevance which means VoIP can be and already is used for worldwide telephony. Many applications of VoIP technology have been developed and are currently under development. For that reason embedding hidden messages in VoIP communication is a very interesting task and may become subject of further studies. In this paper we use the JVOIPLIB 1.3.0 [2] as VoIP framework as basis for our analysis, since this framework is platform-independent and can be used free of charge. Details about the software and our specific implementation are given in [1]. Beyond the work described in former publications we present an extended design and extensive test results. The tests we performed aim at perceptibility, probability of detection, as well as possible malfunction while handling large amounts of data over a long time. The paper is organized as follows: In chapter II an abstract overview of the analyzed VoIP scenario is introduced. Chapter III gives an overview of the design concept including the participating components. In chapter IV test performance and test results are presented. The paper concludes with a short summary and future work.

II. OVERVIEW

The environment of an active VoIP communication using a steganographic channel is illustrated in Figure 1. Alice (A) on the left and Bob (B) on the right are talking over an unsuspicious VoIP connection. Let us assume, Alice wants to send a hidden message to Bob, that means Alice acts as sender and Bob as receiver. She embeds her message in the VoIP stream by using secret side information which is known by Bob, too, but no one else has it.

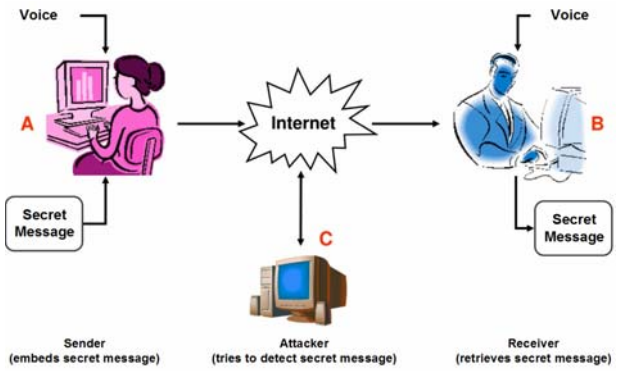

Figure 1. VoIP scenario with steganographic channel.

For describing that communication in a more formal manner we use the following statements. First sender and receiver choose from a set of codecs denoted by $SC=\{sc_1,sc_2,sc_3,..,sc_i \mid i \in N\}$ one audio codec for their communication. From the set of steganographic embedding techniques $SE=\{se_1,se_2,se_3,...,se_i \mid i \in N\}$ Alice chooses one algorithm while Bob selects an according retrieving technique from $SR=\{sr_1,sr_2,sr_3,...,sr_i \mid i \in N\}$. If both techniques match Bob is able to reconstruct the message from Alice. In order to increase security the hidden message is encrypted by

using a symmetric cryptographic scheme from the set of all cryptographic methods CM={cm_1,cm_2,cm_3,...,cm_i | i∈N}. For applying a cryptographic scheme a secret key K is necessary which is generated from a secret password. The mapping of a password to a secret key of a fixed length is applied by choosing a cryptographic hash function. After encrypting the hidden message the message bits are uniformly distributed and spread over the whole audio stream by using a scrambling algorithm. As input the scrambling algorithm gets a pseudo random number which is generated by a pseudo random number generator (PRNG).

III. Design of VoIP Scenario

For our prototypical scenario we concentrate on one codec $sc_1 \in SC$, i.e. we use RAW PCM (8,000 Hz, 8 Bit). As a main requirement we postulate embedding and retrieving must be possible without causing delays or interferences during audible communication. Hence, for embedding the hidden message we choose for $se_1 \in SE$ a Least Significant Bit (LSB) scheme, providing a high capacity and low complexity. According to this the retrieving scheme is chosen. For encryption we use *Twofish* cipher [9] and for the cryptogaphic hash function *Tiger* [12], since for the well-known cryptographic hash functions as MD5 (128 Bit), SHA-1 (160 Bit) and RIPEMD (160 Bit) collisions have been found (see [10] and [11]). *Tiger* has been developed by Eli Biham and Ross Anderson and is rather seldom used. Its benefits are the 192 Bit hash value and the comparatively low complexity. The algorithm has been especially developed for 64 bit CPUs (Alpha) and encryption needs only 390 clocks per byte on that platform, but even on 32 bit platforms *Tiger* is the fastest algorithm. Therefore we estimate Tiger as good choice for our scenario. In addition we use MD5 which produces a shorter hash value to calculate a checksum over the hidden message which is only used for detecting errors during transmission.

A. Sender

Using the above techniques Alice starts the embedding process which is illustrated in Figure 2. She uses a secret password from which a cryptographic hash value is calculated referred to as secret key *K*. The secret key is used for encrypting the hidden message with *Twofish* cipher. After encryption a hash value (*CHK*) is calculated, which helps the receiver to detect errors caused by packet loss or intense network traffic. Furthermore two binary patterns are generated referred to as *Begin Of Message* (*BOM*) and *End Of Message* (*EOM*). *BOM* indicates the start of the hidden message in the audio stream while *EOM* represents the end of the embedded data. Combining the *BOM* pattern, the encrypted message, the hash value *CHK* and *EOM* results in the data which is embedded on sender side. After constructing the message the data is embedded by a spatial domain LSB scheme. The message bits are stored in the least significant bits of the audio samples taken from the VoIP stream. The packet interval I_S is set to 20ms which means 50Hz. Therefore the maximum capacity C_P of one VoIP packet is given by (1) depending on the chosen sample rate P_{samp}.

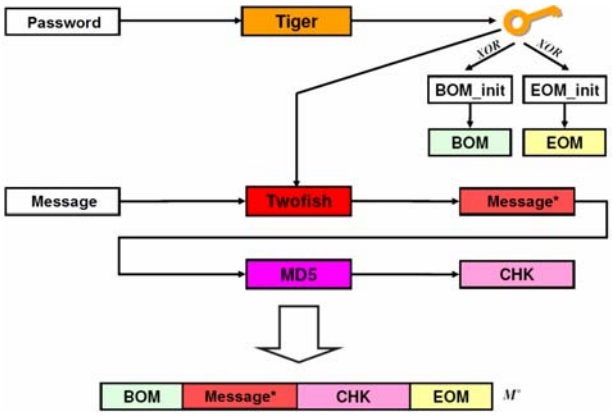

Figure 2. Construction of embedded message.

In Table 1 typical sample rates and the resulting maximum capacity are listed. Moreover, the time T_t needed for transmitting a 1 Mbyte hidden message using the according sample rate is given.

P_{samp} *in Hz*	C_P *in Bits/s*	C_P *in Kbytes/h*	*Time* T_t *for 1 Mbyte*
8,000	160	562	~109 min
11,025	222	780	~79 min
22,050	441	1.550	~40 min
44,100	882	3.100	~20 min

Table 1. Maximum capacity depending on the used sample rate.

In general it is not advisable to use the complete capacity C_P for embedding a secret message. Using maximum capacity may introduce perceptible distortions, especially in silent parts of audio data. Furthermore statistical detection of modifications becomes easier since each least significant bit is changed and contains a part of the embedded message. The test results in chapter IV demonstrate that fact. For that reason only a subset of the possible embedding positions should be used for embedding. Introducing the payload factor *packet_usage* we define the payload C_P* as illustrated in (1).

$$C_P^* = round\left(\frac{packet\_usage \cdot C_P}{100}\right), \quad C_P = \frac{P_{samp}}{I_S} \qquad (1)$$

For example, choosing *packet_usage*=1% we get for the sample rate P_{samp}=8,000 a payload C_P*=2 bits/packet. Beside encryption we try to achieve better security by using a scrambling algorithm in order to distribute the hidden message over the VoIP stream. For each packet we determine individual positions for embedding the message bits by using pseudo random numbers. As PRNG we use sp_1=*MT19937* which is described in [15]. This generator has a period of 2^{19937}-1 and outputs more than 16 million uniformly distributed pseudo random numbers per second. It is initialized by

using the secret key K and its output indicates the position for embedding the next message bit. For a better understanding the process of scrambling is illustrated in Figure 3. The cover data is represented by exemplary sample values of a VoIP packet S_P. Let 100_2 be the hidden message M^*. Applying the scrambling algorithm the positions idx_P={0,1,2,...} are shuffled and changed to idx_P^*={6,2,5,...}. By using the new sequence the encoder embeds the message M^* at the defined positions by adjusting the least significant bit and output the modified VoIP packet S_P^*.

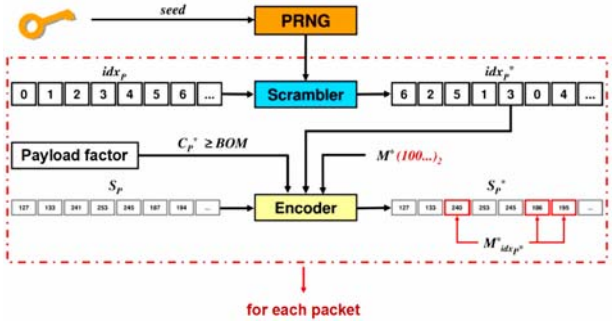

Figure 3. Encoder scheme.

B. Receiver

Sender and Receiver must share the same knowledge of the used parameters which means both of them must know the secret key *K* and the applied payload factor *packet_usage*. After receiving the VoIP packets the recipient uses its key *K* to reconstruct the pseudo random numbers that have been used by the sender. These numbers are used to find the positions within the audio stream *SP* and stores them in a shift register alike packet buffer *MP**. After each writing to *MP** it is checked if the packet buffer contains the synchronization pattern *BOM* or *EOM*. If this is true for *BOM* all following bits of *MP** will be appended to a message buffer *M**. If the pattern *EOM* is recognized after *BOM* has been detected the following bits are interpreted as a hash value (*CHK*) which is compared to the hash value calculated over the message in *M**. If both hash values are equal the transmission of hidden message has been finished successfully. Then the message is decrypted and saved to disk. The overall process is shown in Figure 4.

C. Attacker

In order to make statements about security concerns and non-perceptibility of the described scenario a third person Carol (C) acts as an attacker which is interested in detecting the hidden message of Alice. Let us assume, Carol is capable of accessing the network and detecting VoIP communication. Using their abilities she finds the communication between Alice and Bob and tries to detect the hidden message by analyzing the transmitted VoIP packets. For analysis she uses the Intrusion Detection/Prevention System (IDS/IPS) introduced in [13] and the included module for steganalysis. In [1] the 13 implemented attacks are listed and described in detail.

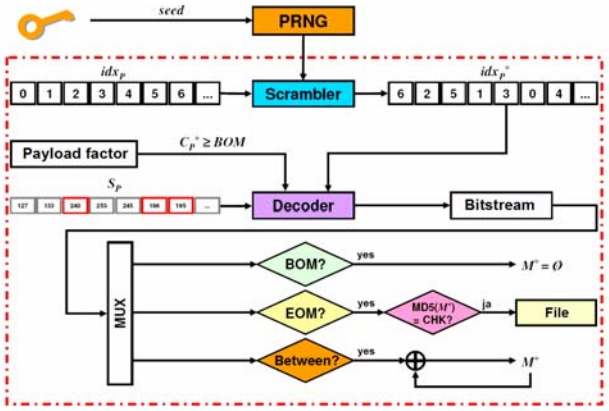

Figure 4. Decoder scheme.

IV. Evaluation and Test Results

First of all we define our test hypotheses. After that we describe the test environments, the test set and finally our first experimental results.

A. Test Hypotheses

For our tests we defined the following three major hypotheses:

(**H1**) Using low *package_usage* results in non-perceptible manipulation of VoIP stream. We expect objective measures will approve that.

(**H2**) Statistical detection of hidden message is not reliably possible. We expect the steganalysis module will approve that.

(**H3**) The error rate of transmitted messages is not significant. We expect error rates near null.

B. Test Set and Test Environments

For our tests we used a soundproof chamber to record audio playback for transmission. Overall, our test set consisted of 2.722 audio files which have been used for simulating VoIP communication. All transmitted data have been stored in a database for offline analysis. The sender and receiver PCs had been synchronized using a central timeserver.

C. Test Results

In Figure 5 the results of our objective quality tests are displayed. We used ODG values *w* to indicate the quality of 14 randomly chosen audio files, which have already been used for a prior subjective hearing test, regarding non-perceptibility of our manipulated VoIP packets. All values $|w| < 1$ represent non-perceptible distortions, which corresponds with results of our subjective hearing test, approving hypothesis (**H1**).

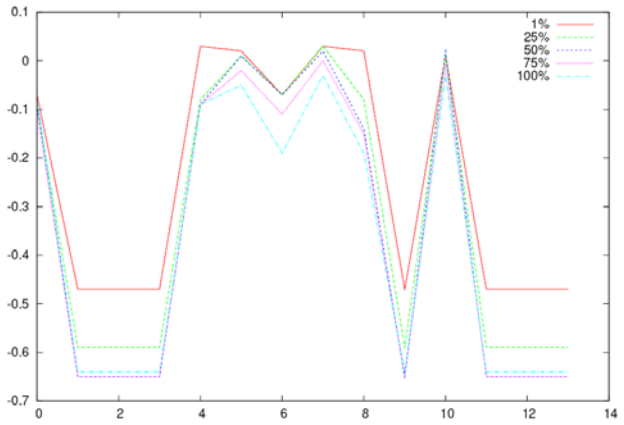

Figure 5. Objective Difference Grade (ODG).

In Figure 6 results of one of our statistical attacks (entropy) are shown. We applied 13 statistical attacks to the transmitted audio data at all. Figure 6 illustrates the windowed entropy of a part of a single audio stream that has been stored using our IDS/IPS referred to as SNIDS. After half of the time message embedding starts. It is obvious that the differences between the chosen payloads are very small and not significant in any case. Without having the original data it is almost impossible to decide if a steganographic message has been embedded, approving hypothesis (**H2**). During evaluation there had been no errors at all, i.e. error rate was in all cases equal to 0, approving hypothesis (**H3**).

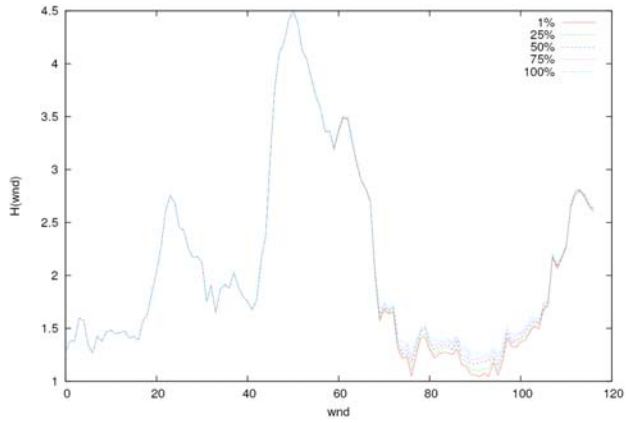

Figure 6. Test results of statistical feature 'windowed entropy'.

SUMMARY AND FUTURE WORK

Our tests have demonstrated that VoIP communication can be practically used for steganographic applications. All test hypotheses introduced in chapter IV have been approved, i.e. the detection of the embedded message by applying statistical methods as well as the perception by human hearer of introduced modifications was not reliably possible. To further minimize the possibility of hearing the manipulation introduced by the hidden message natural sounds, music or continuous audio signals are especially suitable for choosing the cover. In doing so the modifications are superimposed by the cover data completely in all probability. For future work we will concentrate on applying other audio streaming formats, e.g. the Global System for Mobile Communications (GSM) codec. In [14] steganographic techniques are described which embed hidden messages in GSM-coded audio data. Accordingly we plan to extend our scenario to encode and decode GSM-compressed VoIP streams.

ACKNOWLEDGMENT

Effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number FA8655-04-1-3010. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

REFERENCES

[1] Jana Dittmann, Danny Hesse, Reyk Hillert: Steganography and steganalysis in voice-over IP scenarios: operational aspects and first experiences with a new steganalysis tool set, Proc. of SPIE, Vol. 5681, Security, Steganography, and Watermarking of Multimedia Contents VII, San Jose, 2005, pp. 607-618.

[2] JVOIPLIB, Jori's Voice over IP library, Website: http://research.edm.luc.ac.be/jori/jvoiplib/jvoiplib.html, 2004.

[3] StegHide - Homepage: http://steghide.sourceforge.net/, 2005.

[4] Steganos - Homepage: http://www.steganos.de/, 2005.

[5] Westfeld, F5 - Homepage: http://wwwrn.inf.tu-dresden.de/~westfeld/f5.html, 2005.

[6] J. Fridrich. M. Goljan, D. Hogea, Steganalysis of JPEG Images: Breaking the F5 Algorithm, 5th Information Hiding Workshop 2002.

[7] Nils Provos, Steganography Detection with Stegdetect, Website: http://www.outguess.org/detection.php, 2004.

[8] Andreas Westfeld: Detecting Low Embedding Rates, in Fabien A. P. Petitcolas (Ed.): Information Hiding, 5th International Workshop, IH 2002, Noordwijkerhout, The Netherlands, October 7-9, 2002, Lecture Notes in Computer Science 2578 Springer 2003, pp. 324-339.

[9] Schneier: Twofish Cipher. http://www.schneier.com/twofish.html, 2005.

[10] Wang, Feng, Lai: Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. Proc. of the Eurocrypt, 2004.

[11] Wang, Yin, Yu: Collision in the Full SHA1. Crypto, Santa Barbera (USA), 2005.

[12] Biham, Anderson: Tiger - A Fast New Cryptographic Hash Function, http://www.cs.technion.ac.il/biham/, 1995.

[13] Dittmann, Hesse: Network based Intrusion Detection to Detect Steganographic Communication Channels - on the Example of Audio Data. Proc. of the 6th IEEE Workshop, 2004.

[14] K. Gopalan: Audio Steganography by Amplitude or Phase Modification, Proc. Of 15th Annual Symposium on Electronic Imaging -- Security, Steganography, and Watermarking of Multimedia Contents V, San Jose, 2003.

[15] Matsumoto, Nishimura: Mersenne-Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. Proc. of ACM Transactions on Modelling and Computer Simulations: Special Issue on Uniform Random Number Generation, 1998.