Design und Evaluierung von Steganographie für Voice-over-IP

Thomas Vogel, Jana Dittmann, Reyk Hillert, Christian Krätzer

Fakultät für Informatik
Arbeitsgruppe Multimedia and Security
Institut für Technische und Betriebliche Informationssysteme
Otto-von-Guericke Universität Magdeburg

Abstract: Aufbauend auf den Ergebnissen vorangegangener Arbeiten wird in diesem Beitrag die Erweiterung eines Voice-over-IP-Frameworks um die Möglichkeit, geheime Kommunikation auf Basis von Steganographie durchzuführen, fortgesetzt. Kommunikationspartner einer VoIP-Verbindung sind durch Nutzung des erweiterten Frameworks in der Lage, während einer VoIP-Sitzung geheime Nachrichten auszutauschen. Diese Arbeit greift auf bekannte Verfahren zum Verschlüsseln und Einbringen von steganographischen Nachrichten zurück. Dabei stellt jedoch einerseits das Einbetten von Informationen in Streaming-Daten eine neue Herausforderung dar, da beispielsweise Start und Ende der Übertragung flexibel synchronisiert werden müssen, andererseits muss die Verwendung von Sprachdaten über einen fehlerbehafteten Kanal mit eingeschränkter Qualität berücksichtigt werden. Erste Tests sollen Ergebnisse hinsichtlich der Bestimmung von Nutz- und Paketkapazität, sowie der Fehlerrate der Übertragung liefern, um die Anwendbarkeit von Steganographie auf Sprachdaten innerhalb einer paketbasierten Streaming-Applikation zu evaluieren. Des Weiteren erfolgt eine erste Transparenzuntersuchung des unterschiedlichen Verhaltens von Sprach- und Musikdaten hinsichtlich der Einbettung eines LSB-Wasserzeichens in PCM kodierte Audiodaten. Diese Untersuchung wird sowohl für die Transparenz bezüglich der Hörbarkeit als auch der einer Steganalyse (Detektierbarkeit) vorgenommen.

1 Einführung und Motivation

Der Begriff Steganographie bezeichnet die Wissenschaft des Versteckens von Informationen in einem Trägermedium. Dabei wird eine beliebige Information in einem Trägermedium (Cover) versteckt, so dass diese nicht wahrnehmbar als auch nicht detektierbar ist. Für den Bild- und Audiobereich existiert eine Vielzahl von Applikationen, siehe z.B. [1][2][3][4], wobei eine ebenso große Zahl von Algorithmen existiert, welche das Ziel haben, die Verwendung von Steganographie in Bildern nachzuweisen, z.B. [5][6][7]. Insgesamt sind jedoch nur wenige steganographische Entwicklungen zu finden, welche als Trägermedium gezielt Sprachdaten benutzen, siehe zum Beispiel in [17], obwohl sich, zum Beispiel durch die verstärkte Nutzung von Voice-over-IP (VoIP) als Ablösung von Telefonanlagen, hier hervorragende Ansätze für Audio-Steganographie bieten. Ziel unserer Arbeit ist es deshalb zu zeigen, wie bekannte steganographische Ansätze auf VoIP-Applikationen angewendet werden können, und welche Ergebnisse bezüglich Fehlertoleranz bei der Übertragung der Nachricht erzielt werden. Hinter VoIP verbirgt sich die Digitalisierung und Kompression eines analogen Audio-Signals (vorzugsweise Sprache), welches in Paketen über eine Netzwerkverbindung zu einem eindeutigen Ziel (IP-Adresse) übermittelt wird. Dort erfolgt der Prozess in umgekehrter Reihenfolge und aus den digitalen Daten wird ein

hörbares, analoges Signal, die Sprache, dekodiert. Die Größe des Netzwerkes spielt dabei keine Rolle, somit ist auch ein weltweites Telefonieren möglich, daher wird VoIP auch oftmals mit Internet-Telefonie bezeichnet. Die effektive Nutzung von VoIP ist relativ neu und damit ist auch eine stetig größer werdende Zahl von Software- und Hardware-VoIP-Applikationen nicht überraschend. Aufgrund der stetig wachsenden Verbreitung ist in einem VoIP-Kanal ein äußerst interessantes Trägermedium zur steganographischen Kommunikation zu sehen, was allerdings neue Herausforderungen für die Anwendung klassischer steganographischer Techniken bedeutet. Im Gegensatz zu konventionellen Verfahren für Audio- und Bilddateien stellt sich bei steganographischen Anwendungen für VoIP das Problem der Übertragung von kontinuierlichen Datenströmen (Streaming). Die einzubettende Nachricht muss mit Start- und Endmustern versehen werden, um sie im Datenstrom beim Empfänger lokalisieren zu können. Die maximale Kapazität ist bei stream-basierter Steganographie kein fester Wert, sondern resultiert unmittelbar aus der Dauer der Kommunikation und ist somit zeitabhängig. Zusätzlich müssen Techniken zur Sicherung der Integrität der Nachricht verwendet werden, um die vollständige und korrekte Übertragung der Daten erkennen zu können. Für unsere Untersuchungen haben wir uns für das VoIP-Framework JVOIPLIB 1.3.0 [8] entschieden, weil diese Laufzeitbibliothek plattformunabhängig ist und unter der GPL (Gnu Public License) genutzt werden kann. Die Erweiterbarkeit der Bibliothek und die Funktionstüchtigkeit des implementierten Steganographie-Verfahrens auf Sprachdaten ist bereits in [9] mittels bekannter LSB-Steganographie gezeigt worden. Nach einer Überarbeitung des Designs der VoIP-Applikation folgen für die implementierte Software Tests, die Aussagen über die Nicht-Detektierbarkeit und das Fehlerverhalten zulassen. Das Paper ist wie folgt gegliedert: In Kapitel 2 wird zunächst in die Grundlagen anhand einer Formalisierung des Themas eingeführt, wobei auf bekannte Verfahren aufgesetzt und deren Anwendbarkeit auf VoIP beschrieben wird. Kapitel 3 beschreibt das Design der VoIP-Software auf Basis der verwendeten Komponenten. Das vierte Kapitel behandelt die Testdurchführung und –auswertung von ersten Tests. Kapitel 5 liefert eine Zusammenfassung und gibt einen kurzen Ausblick auf noch offene Herausforderungen.

2 Formalisierung von VoIP-Software mit Steganographie-Kanal

Der von uns entwickelte schematische Aufbau einer aktiven VoIP-Verbindung mit Steganographie-Erweiterung wurde bereits in [9] detailliert beschrieben. Dabei nutzen zwei Kommunikationspartner (A und B) eine VoIP-Verbindung um eine nicht-verdächtige Kommunikation aufzubauen. Zusätzlich wird von A gleichzeitig eine geheime Nachricht an B übermittelt. Das Szenario der beiden Kommunikationspartner lässt sich durch je zwei Parameter beschreiben. Sowohl Sender als auch Empfänger wählen aus einer Menge von Codecs $SC = \{sc_1,sc_2,sc_3,..,sc_i \mid i \in N\}$ (set of codecs) je einen Codec für die Sprachkompression aus. Aus der Menge aller existierenden steganographischen Einbettungsalgorithmen $SE = \{se_1,se_2,se_3,...,se_i \mid i \in N\}$ (embedding) legt der Sender sich auf einen Algorithmus - in diesem hier betrachteten Fall Einbettung in die LSBs (Least Significant Bits) - fest. Ebenso muss der Empfänger einen Algorithmus aus der Menge aller Auslesealgorithmen $SR = \{sr_1,sr_2,sr_3,...,sr_i \mid i \in N\}$ (retrieval) spezifizieren. Sind auf beiden Seiten die Codecs und Algorithmen zum Einbettung/Auslesen passend gewählt, so kann B die von A gesendete Nachricht wiederherstellen. Damit bei einem gebrochenem Einbettungs-/Auslesealgorithmus die geheime Nachricht für Dritte nicht lesbar ist, sollte aus der Menge aller kryptographischen Verschlüsselungsverfahren $CM = \{cm_1,cm_2,cm_3,...,cm_i \mid i \in N\}$ (cryptographic method) ein zuverlässiges Verfahren ausgewählt werden, mit dem die Nachricht verschlüsselt wird. Zur Verschlüsselung ist ein Schlüssel K notwendig der in eindeutiger Abhängigkeit zu einer Passworteingabe pw stehen sollte. Damit sowohl kurze als auch (beliebig) lange Passwörter eindeutig auf eine feste Schlüssellänge abgebildet werden, wird vor Implementierung aus der Menge aller kryptographischen Hashfunktionen $SH = \{sh_1,sh_2,sh_3,...,sh_i \mid i \in N\}$ (set of cryptophic hash functions) eine Hashfunktion ausgewählt werden. Nach vorangegangener Verschlüsselung oder Kompression liegen die Audiodaten annähernd gleich verteilt vor. Damit folgen die LSBs der Audio-Samples nach der Einbettung einer annähernden Gleichverteilung. Im Gegensatz dazu sind die Werte der Samples

während eines VoIP-Gesprächs im Allgemeinen nicht gleich verteilt. Zwischen Worten und Sätzen folgen teils längere Intervalle von Stille, idealisiert also Amplitude 0 bei verwendeter PCM Kodierung. Damit die steganographische Kommunikation nicht von unbekannten Dritten detektiert werden kann, ist es sinnvoll die verschlüsselte Nachricht nicht linear, sondern durch Mischen oder Partitionieren verteilt im Audiostrom einzubetten. Ein Algorithmus aus der Menge der Misch-Algorithmen $SM = \{sm_1,sm_2,sm_3,...,sm_i \mid i \in N\}$ (mixing method) übernimmt die Aufgabe das Signal zu partitionieren und zeitlich zu streuen. Für die Zerstreuung der Bits sind Zufallszahlen notwendig, welche mit einem Zufallszahlengenerator (pseudo random number generator, PRNG) gebildet werden. Diese sollten in jedem Fall einer Gleichverteilung unterliegen, um eine gleichmäßige Verteilung der Nutzdaten (Bits) über die Menge von Samples zu garantieren. Aus der Menge der Zufallszahlengeneratoren $SP = \{sp_1,sp_2,sp_3,...,sp_i \mid i \in N\}$ (set of pseudo random number generators) ist ein solcher auszuwählen, der eine Gleichverteilung und hohe Geschwindigkeit gewährleisten kann. Man kann von einem Angriff auf das Steganogramm sprechen, wenn dieses von einem potentiellen Angreifer auf Auffälligkeiten hin untersucht wird. Entsprechend versucht ein Angreifer C in der VoIP-Kommunikation zwischen A und B, die aus einzelnen Paketen mit komprimierter Sprache besteht, den Verdacht einer geheimen Kommunikation zu bestätigen oder abzuweisen. Gegebenenfalls kann er bei Verdacht auf versteckte Kommunikation versuchen, das Steganogramm zu stören oder zu löschen. Um den Verdacht überprüfen zu können, müssen dem Angreifer C die Menge der Codecs sowie eine Teilmenge aus der Menge aller Angriffe $A = \{a_1,a_2,a_3,...,a_i \mid i \in N\}$ (attacks) zur Verfügung stehen. Damit C in der Lage ist, Angriffe durchzuführen muss er eine Methode aus der Menge alle Detektionsverfahren für VoIP-Verbindungen $SD = \{sd_1,sd_2,sd_3,...,sd_i \mid i \in N\}$ (set of detectors for voip connections) auswählen. Diese kann dann auf die Menge aller VoIP-Verbindungen $SV = \{sv_1,sv_2,sv_3,...,sv_i \mid i \in N\}$ (set of voip connections) angewendet werden.

3 Design der VoIP-Szenario-Komponenten

Zur besseren Übersicht haben wir das Design der einzelnen VoIP-Szenario-Komponenten in die Module Sender, Empfänger und Angreifer unterteilt. Diese Module werden im Folgenden jeweils separat erörtert.

Der Sender: Da es sich um eine prototypische Entwicklung für erste Tests handelt, wird die zu unterstützende Menge von Codecs zunächst auf einen Codec beschränkt $SC = \{sc_1\}$. Bei dem unterstützten Codec sc_1 handelt es sich RAW PCM (8.000 Hz, 8 Bit). Als Einbettungsalgorithmus $se_1 \in SE$ wird wie bereits erwähnt ein LSB-Ortsraum-Verfahren gewählt (angelehnt an [1]), da sowohl PCM als auch LSB weit verbreitete Verfahren mit hoher Leistung und geringer Komplexität darstellen. Das Einbetten/Auslesen muss ohne Verzögerung und erhöhte Systemauslastung erfolgen, während die Kommunikationspartner A und B ohne Wartepausen und Störungen in der Übertragung miteinander sprechen können. Passend zur Einbettung wird ein Auslesealgorithmus definiert, der es ermöglicht, die auf der Senderseite eingebrachte Nachricht wiederherzustellen. Als kryptographische Verschlüsselungsmenge wird $cm_1 = \{Twofish\}$ gewählt [10] und als Hashfunktion Tiger $sh_1 = \{Tiger\}$ [11]. Diese europäische Entwicklung von Eli Braham und Ross Anderson wird relativ selten verwendet, erzeugt aber einen Hashwert mit 192 Bit Länge und gilt derzeit als sicher. Der Algorithmus wurde speziell für 64-Bit-Prozessoren (Alpha) entwickelt und erreicht dort die höchste Performanz von 390 Takten pro zu verschlüsselndem Byte.

Er liefert jedoch auch auf 32-Bit-Maschinen die höchste Leistung seiner Klasse, so dass er allein aus Gründen der Performance eine geeignete Wahl darstellt, um den Anforderungen an echtzeitfähiges Streaming zu genügen. Der aus dem Passwort *pw* erzeugte Hashwert mit fixer Länge (192 Bit) ist zugleich der Schlüssel K, welcher für die Twofish-Verschlüsselung benutzt wird. Ein Nachteil der CTR-Modus-Verschlüsselung (counter mode) ist, dass Fehler oder Manipulationen an dem Chiffrat während der Entschlüsselung nicht zu Fehlern führen und daher nicht erkannt werden. Generell besteht bei Übertragungen über ein Netzwerk die Gefahr, dass Pakete auf Grund eines überlasteten Netzwerkverkehrs nicht ankommen bzw. vom Router nicht bearbeitet werden können, d.h. es kann zu Paketverlusten (packet loss) kommen. Zudem ist es möglich, dass ein Angreifer den Netzwerkverkehr abhört (sniffing) und Manipulationen an den Daten vornimmt. Diese Tatsachen machen es daher erforderlich einen Kontrollmechanismus zu integrieren. Über die verschlüsselte Nachricht wird ein Hashwert gebildet, der zur Integritätsprüfung der Nachricht (CHK) genutzt und mit der eigentlichen Nachricht verknüpft wird. Für die Prüfsumme wird die Hashfunktion MD5 verwendet, da mit einer Länge von 128 Bit die zusätzlich übertragenen Daten relativ gering bleiben. Auf der Empfängerseite wird ebenfalls ein Hashwert über die empfangenen Daten gebildet und mit dem Übertragenen verglichen. Sind beide identisch kann davon ausgegangen werden, dass die Nachricht korrekt übertragen worden ist.

Da neben der direkten Kommunikation von A und B durchaus auch eine VoIP-Konferenz (multicasting) mit mehreren Teilnehmern möglich sein könnte, besteht eine notwendige Bedingung darin, dass nur der/die Empfänger welche das Wissen um das korrekte Passwort haben, auch die Nachricht empfangen und entschlüsseln können. Das bedeutet: Sender und Empfänger müssen sich auf Synchronisationsmuster einigen, so dass der Empfänger feststellen kann, ob es sich um Stego-Daten handelt bzw. an welcher Position die Nachricht beginnt und endet. Als Synchronisation verwenden wir zwei mit BOM (Begin Of Message) und EOM (End Of Message) bezeichnete Bitmuster. Diese voneinander verschiedenen Bitmuster werden in Abhängigkeit zu dem Schlüssel K gebildet, der wiederum aus dem Passwort *pw* erzeugt wird. Die Synchronisationsmuster BOM und EOM, die geheime Nachricht und der Hash-Wert (CHK) werden zur Integritätsprüfung zu einer Nachricht M^* zusammengefasst werden. Die Nachrichtenbits werden in den LSBs der Audio-Samples eines VoIP-Paketes eingebettet, wobei das Paketintervall (frame) I_S mit 50 Hz (20 ms) als übliche Frame-Länge in [12] definiert ist. ITU-Studien haben ergeben, dass bei der Übertragung von Sprachdiensten die Verzögerungszeit und das Echo eine besondere Bedeutung für die Qualität der Übertragung haben. Unter der Annahme, dass die zugrunde liegende Paketgröße nicht fix, sondern variabel ist, errechnet sich die maximale Kapazität eines Paketes C_P in Bit, d.h. bei Nutzung aller möglichen LSB-Werte, für eine Abtastrate (sampling rate) P_{samp} mit $C_P = P_{samp} / I_S$.

In Tabelle 1 sind für den Codec sc_1 typische Abtastraten und die daraus resultierenden maximalen Kapazitäten, sowie die rein rechnerisch ermittelte Dauer T_t der Übertragung einer 1 Mbyte großen Nachricht dargestellt. Dabei wird vorausgesetzt, dass weder Hardware- (z.B. Netzwerk, Netzwerkkarte) noch Software-Komponenten (VoIP-Bibliothek, Netzwerkkartentreiber) die Kapazität und Dauer der Übertragung limitieren.

P_{samp} in Hz	C_P = 100% in Bits/s	C_P = 100% in Kbytes/h	T_t in min
8.000	160	562	~109
11.025	222	780	~79
22.050	441	1.550	~40
44.100	882	3.100	~20

Tabelle 1: Hypothetische Betrachtung der Paketkapazität für variable UDP-Paketgrößen in Abhängigkeit von der Abtastrate.

Im Allgemeinen ist es nicht ratsam, die gesamte Paketkapazität C_P = 100% zur Einbettung zu nutzen. Aufgrund der Gleichverteilung der Bits wäre der Verdacht auf Steganographie nachweisbar, vgl. auch die Ergebnisse aus [9]. Deshalb sollte nur eine Teilmenge (einzelne Samples) des Covers verändert werden. Die Mächtigkeit dieser Nutzkapazität C_P^* (payload) wird durch C_P^* = round(*packet_usage* * C_P / 100) bestimmt. Dabei bestimmt *packet_usage* (in %) als Nutzlast-Faktor die prozentuale Verwendung von Samples eines Paketes. Für *packet_usage* = 1% ergibt sich bei einer Abtastrate von 8.000 Hz eine Nutzkapazität C_P^* = 2 Bits/Paket. Wie in Kapitel 2 dargelegt, wird die geheime Nachricht mit Twofish verschlüsselt, und mittels des Vermischens werden für jedes Paket individuelle Positionen zur Einbettung der Nachrichtenbits berechnet. Die Positionen sind zufällig gewählt und für jedes Paket einzigartig. Dazu wird als Zufallszahlengenerator sp_1 = {MT19937} [13] verwendet. Dieser Generator ist unter dem Namen Mersenne-Twister bekannt, der 1997 veröffentlicht wurde. Mersenne-Twister hat eine Periode von 2^{19937}-1 bei einer Geschwindigkeit von mehr als 16 Millionen gleichverteilten Zufallszahlen pro Sekunde und ist somit auch für performance-kritische Anwendungen geeignet. Die Samples selbst bleiben in ihrer Reihenfolge unverändert. Die Initialisierung erfolgt mit einem Startwert (seed) welcher in Abhängigkeit zum Schlüssel *K* steht. Anschließend werden für C_P^* Positionen die LSBs der Samples so verändert, dass sie den Bits der Nachricht M^* entsprechen. Der Prozess des Partitionierens und der zufälligen Positionswahl der LSBs erfolgt für jedes gesendete Paket separat. Damit ist nur derjenige, der das Wissen um den Schlüssel *K* und den verwendeten Algorithmus des Zufallszahlengenerators hat in der Lage, die Bits der Nachricht M^* korrekt zu rekonstruieren. Als Cover stehen die Samples des Paketes S_P mit beispielhaften Werten zur Verfügung. Die einzubettende Nachricht M^* sei $(100)_2$. Mit den berechneten Positonen idx_P^* = {6,2,5,...} werden die LSBs der Samples aus S_P so manipuliert, dass sie der Nachricht M^* entsprechen.

Der Empfänger: Damit die Nachricht aus den ankommenden VoIP-Paketen rekonstruiert werden kann, müssen Sender und Empfänger über die selben Parameter verfügen. Das heißt, beide müssen denselben Schlüssel *K* und Nutzlast-Faktor *packet_usage* kennen und verwenden. Daraufhin ist die Initialisierungskette des Empfängers mit der des Senders identisch, damit sowohl Sender als auch Empfänger denselben Schlüssel und davon abhängig auch dieselben Zufallszahlen konstruieren können. Der schematische Ablauf des Ausleseprozesses ist in Abbildung 1 zusammengefasst.

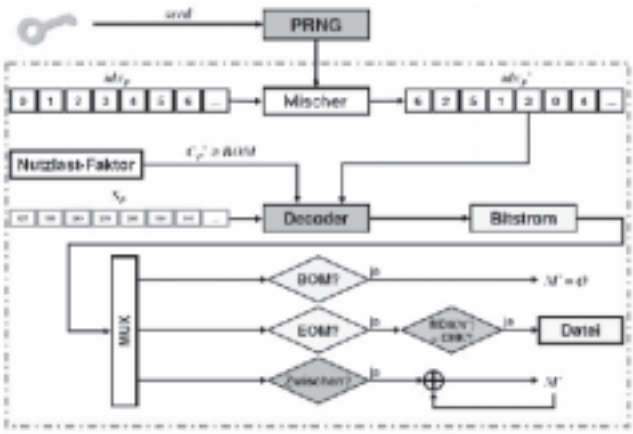

Abbildung 1: Decoder Schema.

Der Auslesealgorithmus schreibt die rekonstruierten Bits aus den jeweilig berechneten Positionen von S_P in einen schieberegister-ähnlichen Paketpuffer M_P^*. Für M_P^* wird nach jedem geschriebe-

nen Bit geprüft, ob es eines der Synchronisationsmuster BOM/EOM enthält. Wird ein BOM erkannt, so werden alle folgenden Nachrichtenbits M_P^* an den Nachrichtpuffer M^* angehängt. Wird nach einem bereits gefundenen BOM ein EOM erkannt, so wird der Hashwert (CHK) aus der erweiterten Nachricht extrahiert und mit dem über die Nachricht berechneten verglichen. Sind beide Hashwerte identisch, so ist die Übertragung erfolgreich abgeschlossen und die Nachricht kann nach erfolgter Entschlüsselung in eine Datei geschrieben werden.

Der Angreifer: Um Aussagen zur Nicht-Detektierbarkeit des entworfenen Verfahrens geben zu können, wird die Existenz des Angreifers C in dem Szenario simuliert. Dieser muss die Netzwerkkommunikation abhören und eine VoIP-Verbindung erkennen, um anschließend Angriffe aus der Menge A durchführen zu können. Die Datenpakete können so auf Steganographie in einem Netzwerk untersucht und als potentiell unautorisierte Kommunikation erkannt werden. Zum Testen wird auf ein in [14] vorgestelltes IDS (Intrusion Detection System) / IPS (Intrusion Prevention System) zurückgegriffen. Speziell wird das darin enthaltene Steganalyse-Modul, welches eine Teilmenge A^* mit bekannten $|A| = 13$ enthält, eingesetzt, um den Verdacht einer Kommunikation mit Steganographie zu bestätigen oder abzuweisen. In [9] sind die 13 implementierten Angriffe und deren mathematische Beschreibung ausführlich dargestellt. Zusätzlich soll die Detektierbarkeit aufgrund der Transparenz durch Abhören des Sprachsignals analysiert werden.

4 Testziele und Testergebnisse

In erster Linie ist zu untersuchen, ob für eine große Menge an Daten (Nutzkapazität) unter Laborbedingungen Paketverluste auftreten, d.h. zu welchen Fehlerraten es bei der Übertragung von geheimen Nachrichten über einen VoIP-Kanal als Trägermedium kommt. Es kann im (weltweiten) Netzwerkverkehr auf Grund verschiedener Routing-Prokolle oder einer Überlastung zu Paketverlust (packet loss) kommen. Mit der Bestimmung der Fehlerrate erhalten wir eine Aussage darüber, in wie vielen Fällen die Nachricht korrekt, und damit vollständig, empfangen wird. Die Transparenz oder auch Nicht-Wahrnehmbarkeit als Haupteigenschaft von Steganographie, werden wir subjektiv und objektiv für verschiedene Nutzkapazitäten C_P^* bestimmen. Die statistischen Eigenschaften der durch die eingebetteten Nachrichten veränderten Audiodateien werden wir mit Hilfe bekannter und schon erwähnter Steganalyse analysieren. Zuvor stellen wir jedoch die folgenden Testhypothesen auf.

4.1 Testhypothesen

Aufgrund unserer Erwartungen haben wir folgende Testhypothesen aufgestellt, die es zu überprüfen gilt:

1. (Angreifer.) Zunächst wird die Wahrnehmbarkeit der eingebetteten Nachricht bei der Einbettung beurteilt, ob ein potentieller Angreifer die Präsenz der steganographischen Kommunikation durch Qualitätsverlust feststellen kann. Testhypothesen werden sowohl für die subjektive wie objektive Evaluation aufgestellt. Dabei wird auch die Kontextabhängigkeit von der Art des unterliegenden Audiomaterials (Sprache und Musik) untersucht.

   A. Subjektive Evaluation: Bei minimalem Nutzlast-Faktor *package_usage* = 1%, das entspricht zwei Bits pro Paket (á 20 ms Audio), ist die Transparenz am größten. Dabei handelt es sich jedoch nur um einen theoretisch erzielbaren Nutzlast-Faktor, weil dabei das BOM/EOM-Muster ignoriert wird. Praktisch können in der aktuellen Implementierung Nutzlast-Faktoren verwendet werden, für die *package_usage* $\geq$ 30% gilt. Insgesamt gilt es als unwahrscheinlich, dass die menschliche Wahrnehmung eine blinde Unterscheidung zwischen Stego-Audio und Cover-Audio vornehmen kann. In subjektiven Hörtests wird dies analysiert werden.

   B. Objektive Evaluation: Eine objektive Bestimmung zur Verschiedenheit, und damit indirekt zur Wahrnehmbarkeit, von Cover und Steganogramm wird mit Hilfe von ODG-Werten

(Objective Difference Grade) erfolgen. Die Differenzmessung bezieht sich auf das menschliche Psychoakustikmodell. Es ist zu erwarten, dass der objektive Test die erste Testhypothese unterstützen wird, d.h. dass eine zuverlässige Erkennung der Steganogramme aufgrund wahrnehmbarer, akustischer Veränderungen bei minimalem Nutzlast-Faktor nicht möglich ist.

2. (Angreifer.) Zur Bestimmung der Nicht-Detektierbarkeit der Nachricht durch einen potentiellen Angreifer werden die statistischen Verfahren verwendet, die in [9] vorgestellt wurden. Es steht zu erwarten, dass die Steganalyse mit Hilfe der 13 Angriffe nicht in der Lage ist, eine zuverlässige Detektion von Steganographie in den abgegriffenen VoIP-Paketen durchzuführen.

3. (Sender/Empfänger.) Auf Sender- bzw. Empfängerseite wird geprüft, ob bei der Übertragung steganographischer Nachrichten über einen fehlerbehafteten Kanal mit Verlusten gerechnet werden muss. Die Fehlerrate in Bezug auf Verluste in den übertragenen Daten hängt in starkem Maße vom Netzwerkverkehr und den verwendeten Routing-Protokollen aber auch mit der Leistungsfähigkeit der Clients und Router ab. Da in dem Netzwerk des Testaufbaus keine Verbindungen über Router untersucht werden, ist mit einer Fehlerrate $\eta < 1\%$ zu rechnen.

4.2 Testaufbau

Um das Szenario der VoIP-Steganographie nachzustellen und alle im vorangegangenen Abschnitt definierten Testhypothesen verifizieren zu können, wurden die folgenden Tests vorgesehen:

1. Der subjektive und objektive Hörtest wird auf Basis von 14 ausgewählten WAV-Dateien aus der zur Verfügung stehenden Audiodatenbank durchgeführt.

A. Für den subjektiven Hörtest zur Transparenzuntersuchung wird eine Nachricht mit einer Größe von 100 KBytes in die 14 ausgewählten Audiodateien eingebettet. Als Nutzlast-Faktor werden jeweils 1% (2 Bits), 25% (40 Bits), 50% (80 Bits), 75% (120 Bits) und 100% (160 Bits) pro Paket für die Einbettung gewählt, so dass die insgesamt 84 Audiodateien (70 Steganogramme plus 14 Originale) in den Hörtest einfließen. Die 14 ausgewählten Audiodaten bestehen aus Aufnahmen von menschlicher und computergenerierter Sprache, Musik und Naturgeräuschen und besitzen eine für VoIP übliche Abtastrate von 8.000 Hz bei einer Sample-Auflösung von 8 Bit. Die zehn Probanden (P1-P10) haben dabei folgende Aufgaben:

a. Offline-Test: Aus den jeweils 6 sich in unbekannter Reihenfolge wiederholenden Audiodateien sollen durch Abhören des Signals das Cover bzw. die Steganogramme herausgefunden werden. Zur Auswertung wird die erfolgreiche Erkennungsrate (Positive Detection Ratio, PDR) und nicht-erfolgreiche Erkennungsrate (Negative Detection Ratio, NDR) bestimmt.

b. Online-Test: Darüber hinaus wird in für das ausgewählte Audiomaterial dieselbe Aufgabe live/online durchgeführt. Dabei wird das Audiosignal aus einer echten bestehenden VoIP-Verbindung auf der Seite des Empfängers abgegriffen und der Testperson über einen Kopfhörer (Headset) zugeleitet. Dieser Test lässt Rückschlüsse auf eventuelle Verstärkungsoperationen oder andere Auswirkungen der Netzwerkübertragung zu. Wichtig hierbei ist, dass die Aktivierung der Übertragung erst nach einer gewissen Laufzeit erfolgt, d.h. die steganographische Nachricht wird variable in das VoIP-Streaming eingebettet. Damit wird feststellbar, ob die plötzlich einsetzende Manipulation der LSBs zu hören ist. Möglicherweise ist eine abrupte Veränderung in der Signalqualität durch die Differenzen von Cover und Steganogramm nachweisbar. Auch hier erfolgt die Bestimmung der PDR und NDR.

B. Eine objektive Messung der Transparenz wird für die 14 ausgewählten Audiodateien durch die Bestimmung von ODG-Werten mit Hilfe von EAQUAL (Evaluating of Audio QUALity) vorgenommen. Es handelt sich dabei um eine Softwareentwicklung des Heinrich-Hertz-Instituts für Nachrichtentechnik GmbH um objektive Aussagen über die Verschiedenheit zweier Audiodateien treffen zu können (vgl. [15]). Die Berechnung von Verschiedenheitsmaßen dient als Qualitätsmaß für Audio-Codecs, dabei wird die menschliche Psychoakustik simuliert und unter anderem eine objektive Differenz (ODG) bestimmt. Ein ODG-Wert $|\omega| < 1$ bedeutet, dass die Differenz des Testsignals zum Referenzsignal gering und damit vernachlässigbar ist. Ein noch

akzeptabler ODG-Wert liegt zwischen $1 \leq |\omega| < 3$. Mit $|\omega| \geq 3$ wäre die Differenz sehr groß und damit die Veränderung sehr stark hörbar. Für die verschiedenen Steganogramme aus dem Offline-Test (1.A.a) mit unterschiedlicher Kapazitätsnutzung wird der ODG-Wert in Bezug zum Cover (Original) berechnet.

2. Die Steganalyse wird für die 14 ausgewählten Audiodateien des Offline-Tests (1.A.a) in Form der 13 Angriffe aus dem Steganalyse-Modul (SDM) durchgeführt. Prinzipiell werden zur Analyse dieselben Audiodaten genutzt, die für die subjektiven Hörtests verwendet werden. Der Unterschied ist: Die Steganogramme werden in zwei Hälften geteilt und die erste Hälfte ersetzt durch die des Covers. Man erhält dadurch jeweils ein Audiosignal, das zur ersten Hälfte aus unveränderten und danach aus manipulierten Daten besteht. Simuliert wird damit eine plötzliche Aktivierung der Stego-Übertragung in einer VoIP-Sitzung, wie es im subjektiven Online-Hörtest der Fall ist. Damit eine plötzlich auftretende Signal-Veränderung sofort in einer Verschiebung der Analysewerte resultiert, erfolgt die Steganalyse gefenstert (windowed). Das heißt, jeweils N Samples werden zu einem Fenster (window) zusammengefasst, welches untersucht wird. Als optimale Fenstergröße wurde $N = 1.024$ bestimmt. In der Datenauswertung und Visualisierung wird anschließend untersucht, ob das Stego-Signal signifikante Veränderungen hervor ruft und ob diese für eine Detektion genutzt werden könnten.

3. Zur Prüfung der Fehlerrate während der Übertragung wird ein Testaufbau eingesetzt, in dem drei Rechner zu einem lokalen Netz zusammen geschlossen sind. Ein 100 Mbit Ethernet-Hub sorgt dabei für eine Verbindung zwischen den Computern. Während zwei Rechner Sender (A) und Empfänger (B) repräsentieren, ist ein dritter Rechner als Angreifer (C) und zum Abspielen der Sprachdaten vorgesehen. Für Tests, welche sich über einen Zeitraum von mehreren Tagen erstrecken, ist es kaum realisierbar, diese durch echte Sprecher durchführen zu lassen. Stattdessen wird auf eine Audiodatenbank des MIT (Masachusetts Institute of Technology) mit 2.402 aufgenommenen Sprachsamples (englisch) zurückgegriffen. Ergänzt wird diese durch 320 Audiodateien, die von Mitarbeitern der Forschungsgruppe bereitgestellt wurden. Insgesamt stehen somit 2.722 Audiodateien zur Verfügung. Neben menschlicher sowie computergenerierter Sprache enthält das Testset unter anderem auch Musik sowie Naturgeräusche. Die im WAV-Format vorliegenden Audiodateien werden in zufälliger Reihenfolge von einem Abspielprogramm abgespielt. Die VoIP-Kommunikation wird durch Ausgabe der Audiodaten über einen Lautsprecher und erneutes Sampling innerhalb einer schalldichten Kabine möglichst originalgetreu nachgestellt. Der Angreifer C wird mittels des in [9] vorgestellten Steganalyzer Network-based Intrusion Detection System (SNIDS) simuliert, das die übertragenen Pakete zur späteren Analyse in eine Datenbank schreibt. Die Synchronisierung der Fehlerprotokolle von Sender und Empfänger übernimmt ein Timeserver verwendet.

4.3 Ergebnisse

In diesem Abschnitt werden die Ergebnisse der beschriebenen Tests dargestellt und erörtert.

1. Analog zur Beschreibung des Testaufbaus in Abschnitt 4.2 werden auch hier die Testergebnisse des subjektiven und objektiven Hörtest getrennt dargestellt.

A. Im subjektiven Hörtest konnten die Probanden für den Online-Test (in Tabelle 2 kontextabhängig beschrieben für die 10 Probanden P1 bis P10 und 6 verschiede Einbettstärken) nur eine relativ geringe durchschnittliche Erkennungsrate über alle Tests von weniger als 16% erreichen (für Einbettstärken von 1%, 25% und 50% wurden die Einbettung fast überhaupt nicht erkannt, erst ab Einbettstärken von 75% auf Sprache und 100% auf Musik wurden Einbettungen in stärkerem Maß detektiert). Erwartungsgemäß nahm mit zunehmendem Nutzlast-Faktor die Erkennungsrate zu. Während bei einer Nutzlast von 2 Bits pro Paket die PDR unter 3% lag, konnten die Probanden bei 160 Bits pro Paket das Steganogramm in Sprache mit einer Wahrscheinlichkeit von über 55% erfolgreich detektieren. Die FAR (False Acceptance Rate) für eine Einbettstärke von 0% (keine Steganogramm eingebettet) liegt bei diesem Test bei 0%. Insgesamt schneidet bei diesem

Test die Transparenz der Einbettung in Sprachdaten für jede Einbettstärke jeweils schlechter ab als für die Einbettung in Musik.

	0%	1%	25%	50%	75%	100%	0%	1%	25%	50%	75%	100%
P1	0,00	0,00	0,00	0,00	54,55	54,55	0,00	0,00	0,00	33,33	0,00	33,33
P2	0,00	9,09	9,09	9,09	36,36	45,45	0,00	0,00	0,00	0,00	0,00	33,33
P3	0,00	0,00	0,00	9,09	27,27	63,64	0,00	0,00	0,00	0,00	0,00	33,33
P4	0,00	9,09	0,00	0,00	18,18	54,55	0,00	0,00	0,00	0,00	33,33	33,33
P5	0,00	0,00	9,09	0,00	9,09	72,73	0,00	0,00	0,00	0,00	0,00	66,67
P6	0,00	0,00	0,00	9,09	9,09	54,55	0,00	0,00	0,00	0,00	0,00	66,67
P7	0,00	0,00	9,09	0,00	9,09	72,73	0,00	0,00	0,00	0,00	0,00	33,33
P8	0,00	0,00	9,09	9,09	0,00	36,36	0,00	0,00	0,00	0,00	0,00	66,67
P9	0,00	9,09	0,00	0,00	18,18	45,45	0,00	0,00	0,00	0,00	0,00	33,33
P10	0,00	0,00	0,00	9,09	9,09	54,55	0,00	0,00	0,00	0,00	0,00	66,67
Ø	0,00	2,73	3,64	4,55	19,09	55,45	0,00	0,00	0,00	3,33	3,33	46,67

Tabelle 2: Resultate für den Online-Test (links Sprache, rechts Musik)

Der Offline-Test (in Tabelle 3 kontextabhängig beschrieben für die 10 Probanden P1 bis P10 und 6 verschiedene Einbettstärken – Tabelle 2 und 3 sind nicht direkt vergleichbar, da in beiden Fällen mit unterschiedlich Gruppen von Probanten gearbeitet wurde) zeigt ein ganz ähnliches Bild, wobei die durchschnittliche Erkennungsrate insgesamt mit 49% deutlich höher liegt, was darauf zurückzuführen ist, dass beim Online-Test die Einflüsse der VoIP-Übertragung, wie Abtastfehler, Verstärkung oder Verzögerungen, die zuverlässige Detektion der Steganogramme durch die Probanden behinderte und zusätzlich der Offline-Test keinen Begrenzungen bezüglich der Wiederholung des Abspielens unterlag. Zu beachten ist, dass sich in diesem Test im Durchschnitt eine FAR von 46,36% bei Sprachdaten und von 30% bei Musik für eine Einbettstärke von 0% erkennen lässt.

	0%	1%	25%	50%	75%	100%	0%	1%	25%	50%	75%	100%
P1	36,36	36,36	63,64	72,73	36,36	54,55	33,33	33,33	66,67	0,00	33,33	33,33
P2	45,45	45,45	72,73	72,73	45,45	90,91	33,33	33,33	66,67	33,33	100,00	66,67
P3	81,82	63,64	81,82	63,64	45,45	72,73	33,33	0,00	66,67	66,67	33,33	33,33
P4	27,27	72,73	27,27	45,45	81,82	63,64	33,33	33,33	33,33	100,00	33,33	66,67
P5	36,36	27,27	36,36	36,36	27,27	54,55	0,00	0,00	66,67	0,00	0,00	0,00
P6	54,55	36,36	18,18	27,27	45,45	54,55	66,67	66,67	33,33	0,00	33,33	66,67
P7	54,55	63,64	54,55	54,55	54,55	72,73	0,00	33,33	66,67	33,33	33,33	33,33
P8	36,36	36,36	36,36	36,36	63,64	54,55	0,00	33,33	33,33	66,67	66,67	100,00
P9	63,64	54,55	63,64	81,82	45,45	54,55	66,67	33,33	33,33	0,00	66,67	0,00
P10	27,27	63,64	36,36	36,36	27,27	81,82	33,33	66,67	66,67	0,00	0,00	66,67
Ø	46,36	50,00	49,09	52,73	47,27	65,45	30,00	33,33	53,33	30,00	40,00	46,67

Tabelle 3: Resultate für den Offline-Test (links Sprache, rechts Musik)

Sowohl der Offline-Test, als auch das realistischere Szenario des Online-Tests bestätigen die erste Testhypothese, dass eine zuverlässige Erkennung des Steganogramms bei einem Nutzlast-Faktor von 1% kaum möglich ist. Die Nutzlast ist dabei allerdings so minimal, dass für die Übertragung einer Nachricht der Länge eines KByte bei einer Abtastrate von 8.000 Hz knapp 1:22 Minuten

benötigt werden, die Übertragung eines MBytes würde mehr als 23:18 Stunden dauern. Eine wichtige erste Erkenntnis aus diesen Testresultaten ist der Umstand, dass die Einbettung in Sprachdaten eine geringere Transparenz als die Einbettung in Musik zu haben scheint.

B. Die Ergebnisse des objektiven Transparenztests für die 14 getesteten Audiodateien sind in Abbildung 2, sowie Tabelle 4 und 5 zusammengefasst. Dabei wurde für jeden gewählten Nutzlast-Faktor eine eigene Kurve in der Abbildung eingezeichnet. Wie in Abschnitt 4.2 beschrieben, bedeutet ein ODG-Wert $|\omega| < 1$, dass die Veränderungen zur Originaldatei so gering ausfallen, dass sie durch den Menschen kaum bzw. nicht wahrnehmbar sind. Für alle getesteten Nutzlast-Faktoren liegen die gemessenen ODG-Werte unter diesem Schwellwert, selbst für die schlechtesten Werte gilt $|\omega| < 0{,}7$ (wobei auffällig ist, das die Werte für 75% und 100% sehr ähnlich sind). Die Resultate der objektiven Transparenzmessung konnten demnach die zweite Testhypothese bestätigen, die eine sehr geringe akustische Veränderung des Audiosignals durch Einbringen des Steganogramms vorausgesagt hat.

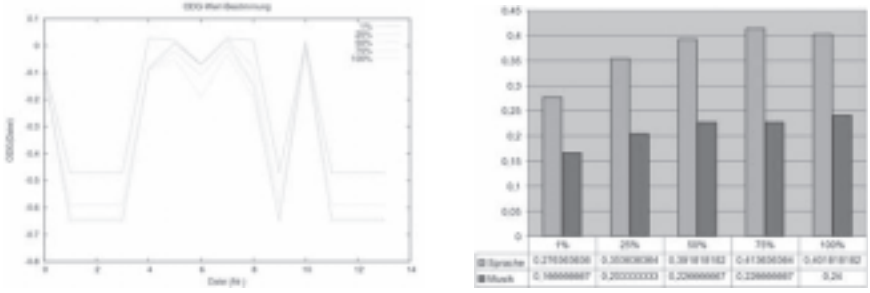

Abbildung 2: Durchschnittlicher ODG ω (y-Achse) für die getesteten Kapazitäten aller 14 Audiodateien (links), sowie der Absolutwert |ω| aufgeschlüsselt in Musik und Sprache (rechts).

In Tabelle 4 und 5 ist gut zu erkennen, dass auch für die objektiven Transparenztests dieselbe Erkenntnis wie in den subjektiven Tests gezogen werden kann. Die Einbettung in Sprachdaten hat stärkere Auswirkungen auf die Transparenz als die Einbettung in Musik. Diese ersten, hier durchgeführten, Tests implizieren, dass die Transparenz der Einbettung in Musik mit einem Nutzlastfaktor von 100% (Durchschnittlich 0,24) immer noch besser ist als die Einbettung in Sprachdaten mit 1% (Durchschnittlich 0,276). D.h. bei gleicher angestrebter Transparenz hätten Sprachdaten eine wesentlich geringere Einbettkapazität als Musikdaten.

	1%	25%	50%	75%	100%
Track2	0,07	0,07	0,07	0,11	0,19
Track3	0,03	0,03	0,02	0,00	0,03
Track4	0,02	0,08	0,14	0,15	0,19
Track5	0,47	0,59	0,65	0,75	0,64
Track8	0,47	0,59	0,65	0,65	0,64
Track9	0,47	0,59	0,65	0,65	0,64
Track10	0,07	0,09	0,09	0,10	0,08
Track11	0,47	0,59	0,65	0,75	0,64
Track12	0,47	0,59	0,65	0,65	0,64
Track13	0,47	0,59	0,65	0,65	0,64
Track14	0,03	0,08	0,09	0,09	0,09
Ø	0,2764	0,354	0,392	0,414	0,402

Tabelle 4: Resultate für den objektiven Transparenztest auf Sprache.

	1%	25%	50%	75%	100%
Track1	0,02	0,01	0,01	0,02	0,05
Track6	0,01	0,01	0,02	0,01	0,03
Track7	0,47	0,59	0,65	0,65	0,64
Ø	0,1667	0,203	0,227	0,227	0,24

Tabelle 5: Resultate für den objektiven Transparenztest auf Musik.

2. Für die Evaluierung der statistischen Eigenschaften wurden 13 Angriffe des Steganalyzer-Frameworks auf die veränderten Audiodaten angewendet. Von den 13 ausgewählten Angriffen sind in Abbildung 3 nur zwei Ergebnisse für eine Sprachdatei und eine Musikdatei für jeweils fünf unterschiedliche Nutzlastfaktoren beispielhaft hervorgehoben, welche auch sichtbare Unterschiede zwischen Cover und Steganogramm erkennen lassen. In den meisten Fällen der getesteten Audiodaten ist der Kurvenverlauf der verschiedenen Nutzlast-Faktoren identisch mit dem des Covers. An den ausgewählten Beispielen wird aber erkennbar, dass sich ab dem Zeitpunkt des Einbettens der steganographischen Botschaft zumindest für das dargestellte statistische Merkmal ein geringfügig abweichender Kurvenverlauf feststellen lässt. Allerdings sind auch hier die Unterschiede so marginal, dass ohne Kenntnis des Originalsignals eine zuverlässige Detektion des Steganogramms anhand der hier untersuchten statistischen Verfahren ausgeschlossen werden kann. Zusammenfassend bleibt festzuhalten, dass sich auch die dritte Testhypothese für die untersuchten bekannten 13 Steganalyseverfahren als zutreffend herausgestellt hat.

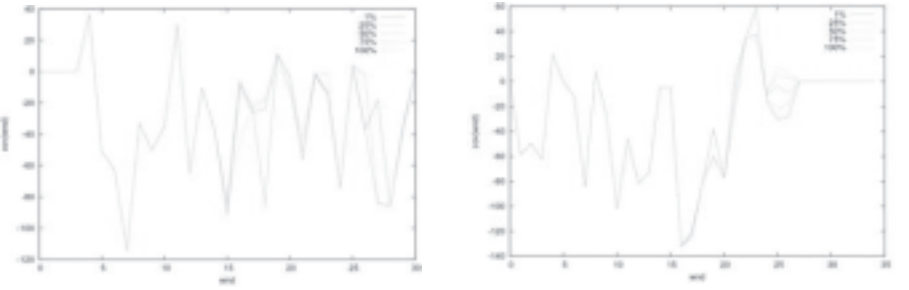

Abbildung 3: Statistische Auswertung für Sprache (Track 13, links) und Musik (Track 1, rechts) durch das Steganalyse-Modul am Beispiel der Kovarianz (vgl. [9]).

3. Die Testhypothese 3 wurde durch die während der Tests gesammelten Erfahrungen ebenfalls gestützt. Die eingesetzte VoIP-Software funktionierte zuverlässig und fehlerfrei über die gesamte Testphase hinweg. Für die gesamte Testdurchführung waren rund 240 Stunden erforderlich. Es kam weder zu Paketverlusten noch zu Einschränkungen der Funktionalität während des Betriebs. Auch die Übertragung der Nachrichten erfolgte stets zuverlässig, was auf das abgeschlossene Szenario ohne Netzverkehr und Störungen von Dritten zurückgeführt werden kann.

5 Zusammenfassung und Ausblick

Diese ersten Tests haben gezeigt, dass VoIP-Kommunikation unter Laborbedingungen zuverlässig für steganographische Anwendungen genutzt werden kann. Zu untersuchen bleibt, welche Fehlerraten unter realistischen Bedingungen auftreten und wie mit Paketverlusten umgegangen werden kann. Die aufgestellten Testhypothesen haben sich als richtig herausgestellt, d.h. die Detektion der eingebetteten Nachricht durch den Menschen selbst, sowie durch Untersuchung statistischer Merkmale ist bei geringern Nutzlastfaktoren nicht zuverlässig möglich. Eine wichtige Erkenntnis aus den ersten hier beschriebenen Tests ist der Umstand, dass die LSB-Einbettung von steganographischen Nachrichten in Sprachdaten möglicherweise eine deutlich schlechtere Transparenz hat als die Einbettung in Musik. Dieser Umstand muss Gegenstand weiterer Untersuchungen sein, um diesen Fakt zu bestätigen oder zu widerlegen. Als mögliche Weiterentwicklung soll hier nur auf die Einbindung weiterer Audio-Streaming-Formate kurz eingegangen werden. Ein relativ weit verbreiteter Codec neben der PCM-Kodierung stellt der Global System for Mobile Communications-(kurz GSM)-Sprachcodec dar, welcher auch speziell auf die Kodierung von Sprachdaten abgestimmt ist. In [16][17] werden Verfahren beschrieben, in GSM-kodierte Audiodaten steganographische Informationen einzubetten, da das dort beschriebene Verfahren durch ein Patent

geschützt ist konnte es bisher nicht mit getestet werden. Eine Erweiterung der eingesetzten VoIP-Software um die Funktionalität der GSM-Kodierung und eine darauf angepasste Einbettung der steganographischen Nachrichten ist vorgesehen.

Acknowledgements: The work about transparency evaluation described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. Effort for implementing and evaluating the VoIP scenario described in this paper is 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.

Literaturverzeichnis

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

[2] Fabien Ptitcolas: MP3Stego - Homepage. http://www.petitcolas.net/fabien/steganography/mp3stego/index.html, 2005.

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

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

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

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

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

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

[9] 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.

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

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

[12] H. Schuzrinne, S. Casner, R. Frederick, V. Jacobson: RFC 1889: RTP: A Transport Protocol for Real-Time Applications. 1996.

[13] 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.

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

[15] EAQUAL - Evaluating of Audio Quality. http://sourceforge.net/projects/eaqual/index.html, 2004.

[16] Gopalan, Wenndt, Noga: Covert Speech Communication Via Cover Speech by Tone Insertion, Proc. of the 2003 IEEE Aerospace Conference, Big Sky, 2003.

[17] 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.