Non-repudiation of Voice-over-IP conversations with chained digital signatures

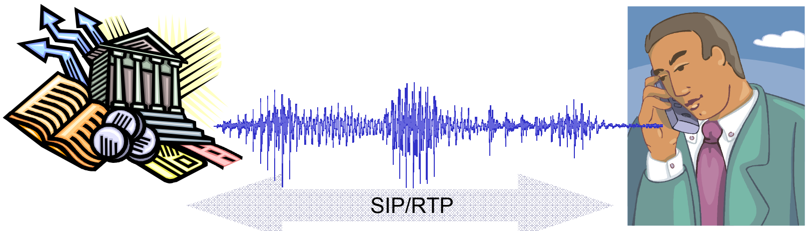
C. Hett, N. Kuntze, A. U. Schmidt
Fraunhofer SIT, Darmstadt, Germany

issa information security south africa conference2006 from insight to foresight

Fraunhofer SIT Institut Sichere Informations-Technologie

Voice communication in business contexts is moving to VoIP

The latest successful example for the ever ongoing convergence of information technologies is internet based telephony, transporting voice over the internet protocol (VoIP). Analysts estimate a rate of growth in a range of 20% to 45% per annual, expecting that VoIP will carry more the fifty percent of business voice traffic (UK) in a few years. Success of VoIP will not be limited to cable networks, convergent speech and data transmission will affect next generation mobile networks as well.



Some security issues of VoIP are tackled

Secure VoIP protocols, using cryptographic protection of a call, would even be at an advantage compared to traditional telephony systems. Protocols like SRTP can provide end-to-end security to phone calls, making them independent from the security of the transport medium. SPIT and DoS are future threats.

Digital voice data needs protection of **integrity**

VoIP communication needs integrity protection for **higher security functions**, just like any other digital data

Goal: Non-repudiation of conversations by caller and callee

Voice has inherent **evidentiary value** due to the possibility of forensic evaluation (e.g. speaker identification). This yields to recorded voice communication a rather **high probative force**, e.g., in a court of law. Specific features of voice communication contribute to non-repudiation. The medium consists in a **linearly time-based full duplex channel**. In particular, **interactivity** allows to make further enquiries in case of insufficient understanding.

State of the art: Analog recording

In **telephony banking and brokerage**, the state of the art to ensure non-repudiation and in turn ascribe to calls the status of **binding contracts**, is still analog recording. The **consent to the recording** is usually embodied in service agreements. A few digital recording solutions exist, but provide only end-to-end security, and **do not provide integrity**

Therefore we need

➢ Protection of the integrity of VoIP conversations

Protecting voice from falsification differs from the protection of other data due to the relevance of temporal context**. Packet ordering/loss** need consideration; **Creation time** must be assigned to conversation.

➢ Speaker authentication

An initial authentication needs to be carried over to the whole call.

➢ Electronic signatures over VoIP conversations

Building on integrity protection and authentication it is possible to achieve, for voice conversations, the level of non-repudiation provided by **electronic signatures** over digital documents, i.e., an **expression of will**.

The Idea: Continuous signing of voice communication

Use digital signatures to sign VoIP streams continuously

Continuous communication requires **continuous application of security methods**. Digital signatures provide the highest level of security and non-repudiation. A signer has to sign a complete call for that.

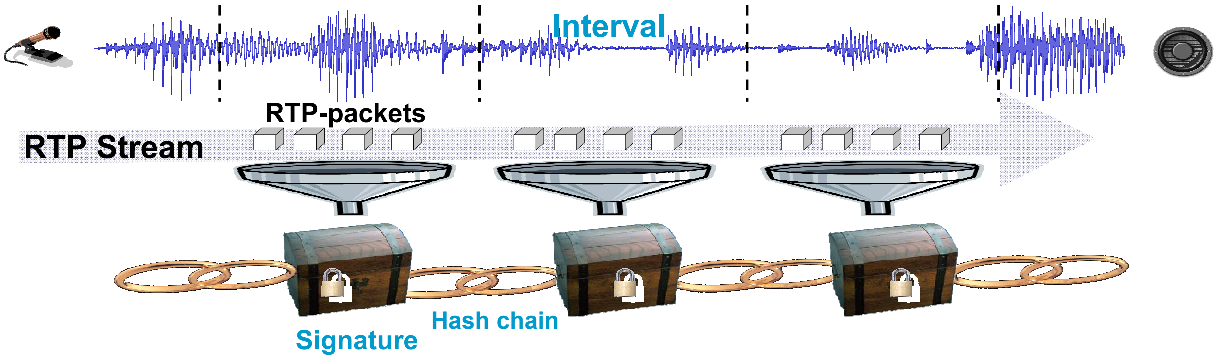
Take full-duplex nature of communication into account

In a **bidirectional, full-duplex** interactive conversation, only both channels together provide the necessary context for full comprehension. To ensure that parts of the talk are not exchanged with other parts, replaced by injections, or cut out, one needs to ensure **cohesion**. That is, the **temporal sequencing** of the communication and its **direction** is data which needs to be protected from tampering

Combine security with efficiency

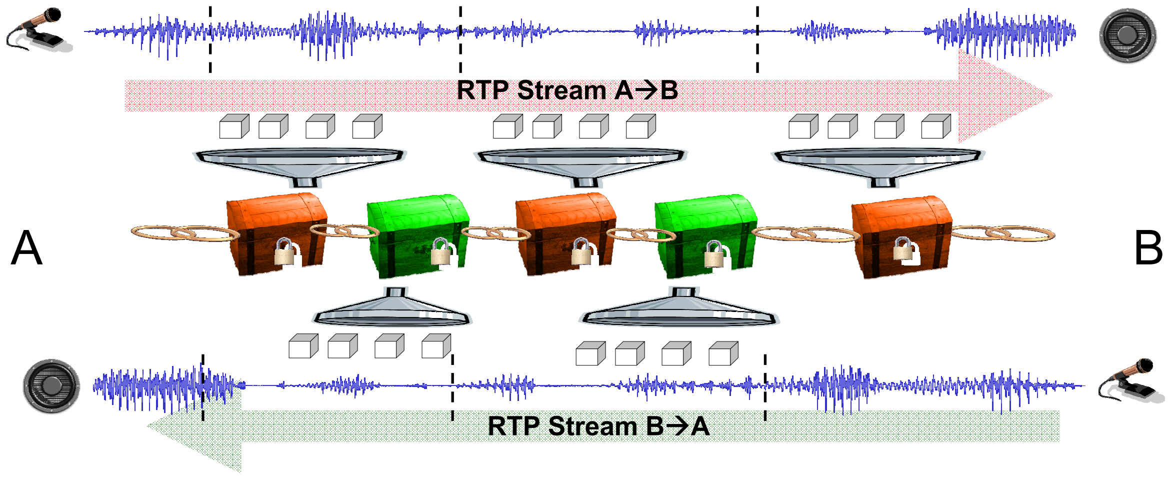
Signing a conversation as a blob is not a good idea in terms of efficiency and scalability (A large number of concurrent calls must be handled in real business environments.). Digitally signing individual packets in an RTP-Stream is excluded for computational load. It is desirable, both from a security as well as an efficiency viewpoint, to sign and **secure the VoIP conversation as "close" as possible to its transmission**, and conceptually close to the actual VoIP stream. Client-side requirements should be minimised.

Central Concept: Split stream in Intervals, build a cryptographic chain



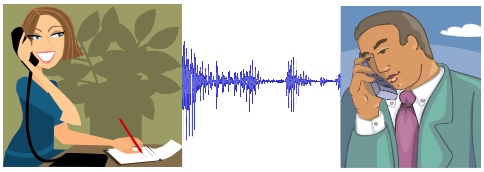
The VoIP-stream is divided into **intervals** of adjustable length – based on time, packet amount or another criterion. Both parties – caller and callee – collect received packets in buffers. Packets in each interval are electronically signed together with relevant meta-data, building the **interval signature**.

Signed intervals alone do not ensure cohesion. An attacker could exchange parts of the conversation or cut them out. Therefore we make use of **hash chains**: Every interval contains, embedded in its metadata, a hash of the last interval including its signature. In this way signatures and hashes are interleaved ensuring that there is a continuous stream of signatures building an **unbreakable chain**. The chaining of intervals is further extended to factor into the bidirectional nature of the call. **Both channels are interweaved** and the chaining applies to both channels. An interval of packets from the channel A→B contains a hash of the last signed interval from the channel B→A and so on.



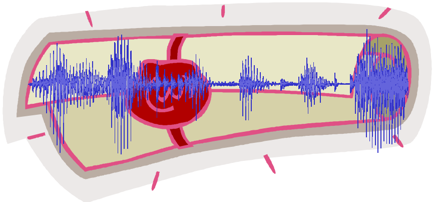
Use-case scenario: Non-repudiation of verbal contracts

Two business parties **negotiate terms and conditions of a contract** using VoIP telephony.



At a certain stage, they decide to **conclude the contract verbally**, e.g. to save time and resources.

Using an **authentication token**, e.g. smart card with PIN, biometric features, etc. they initiate the signing process.



Both parties are **signalled** that the call is signed and recorded for later use.

Termination of the signing is done explicitly or implicitly by hanging up

They store the signed portion of the call as a **documentation of the contract**

Later one of the parties **disputes** certain terms of the agreement or the existence of the contract

The other party presents the signed VoIP data as **proof** in the ensuing **court case**

An expert witness testifies that
- the digital **signature is valid**
- the **method** of signing & chaining **is secure** and cohesion is maintained

Another expert witness specialised in forensic voice evaluation attests
- the **identity of speakers**
- phonetic quality and **understandability**
- that the conversation was natural and **not forged**

The court confirms the validity of the contract. Thus we achieved

Legally binding verbal contract over VoIP and without witnesses

This scenario rests on certain pre-conditions:

➢ Consent to recording

Consent to recording is required to comply with **data protection and privacy** regulations. It is **implicitly** given by usage of the authentication tokens for signing. Nevertheless, **caller and callee should be notified**.

➢ Quality of the channel

Quality of Service (of the VoIP channel) needs due consideration. A sufficient **QoS must be maintained** during the signed portion of the call to ensure understandability Otherwise attacks and attempts to forgery cannot be detected, **limiting the probative value** of the signed call.

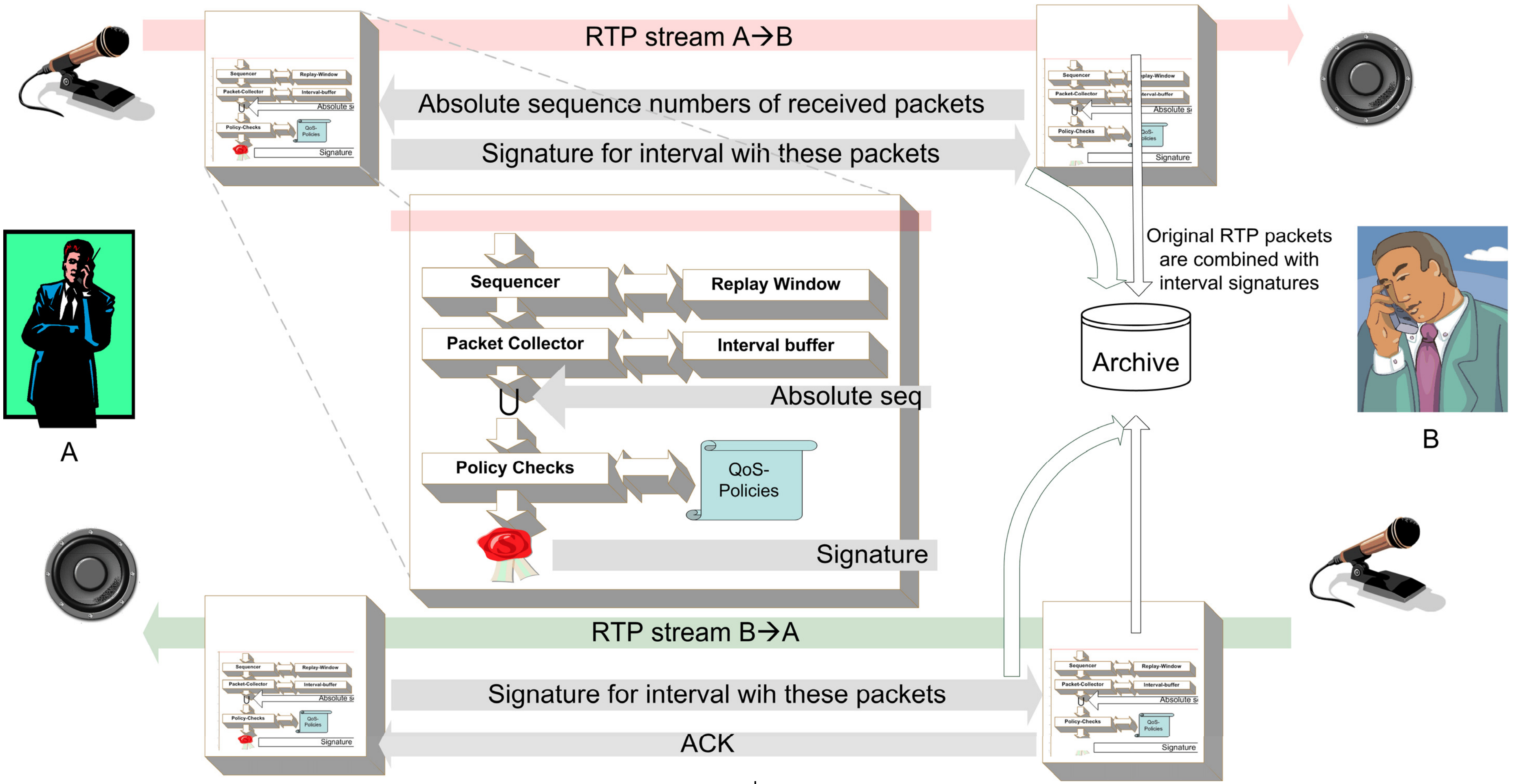
➢ Understanding and awareness

The communicating parties need to be aware of the characteristics of verbal negotiations. For instance they should frequently make further inquiries to assure themselves about the meaning of what was spoken and to **avoid ambiguities** (what is understood is what is signed)

➢ Forensic voice evaluation

The **biometric features inherent in voice** are complementary to the probative force of digital signatures. VoIP is able to preserve these features.

Caller A signs a conversation: architecture and procedure



RTP stream A→B

Absolute sequence numbers of received packets

Signature for interval wih these packets

Sequencer

Replay Window

Packet Collector

Interval buffer

Absolute seq

Policy Checks

QoS-Policies

Signature

Original RTP packets are combined with interval signatures

Archive

A

B

RTP stream B→A

Signature for interval wih these packets

ACK

The original RTP-stream is forwarded to the recipient in real time without noticeable delay.

The **sequencer** extends the truncated 16Bit-sequence number of RTP-packets to 64 Bit, absolute sequence numbers starting with zero. A **replay window** as defined in annex A of the SRTP-RFC is used to detect duplicate packets.

The **packet-collector** collects all sent or received packets and sorts them by their absolute sequence number. It buffers all packets belonging to the current interval. This has much less and only static memory requirements than storing the complete call for later signature.

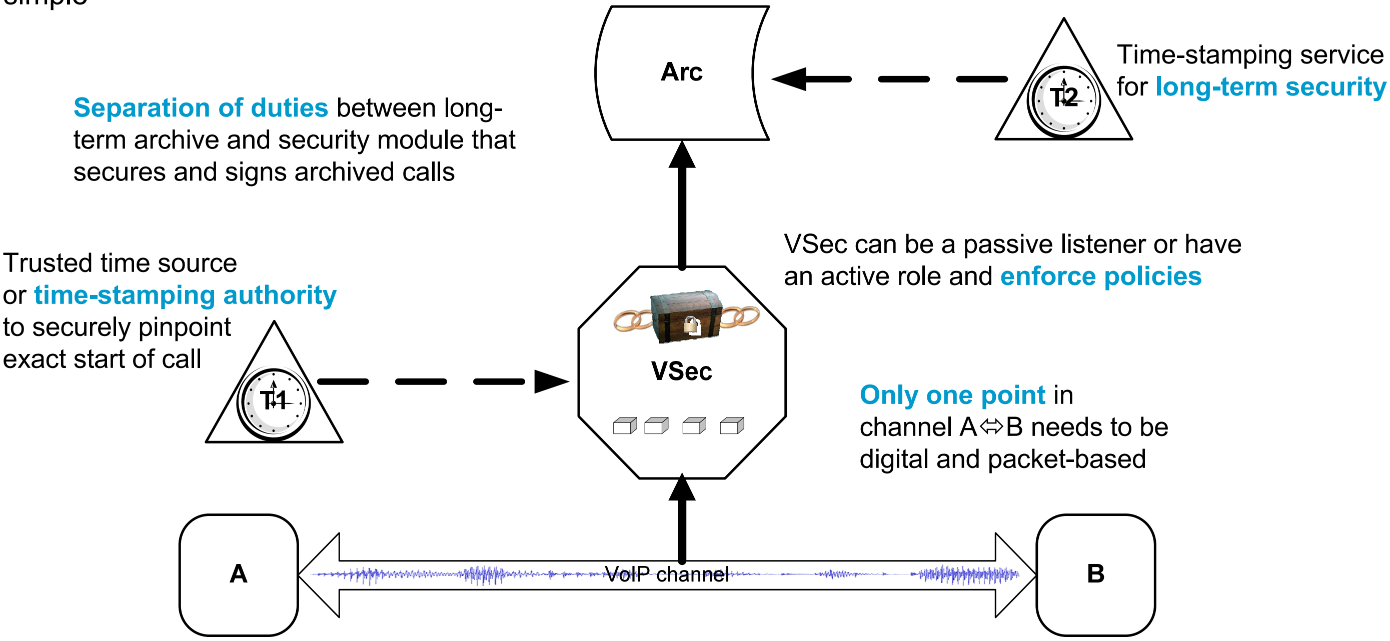
For the channel A to B: A gets from B the **list with packet numbers that B actually received**. A as the sender of these packets was able to collect them all. A then discards the packets that B did not receive. After that **QoS-policies** can be applied: If B lost to many packets, the call becomes ambiguous and A may terminate the call or take other measures.

A builds the **interval signature** package with metadata and the hashes of the contained RTP-packets and sends this to B. The full RTP-packets need not be send again over the wire thus resulting in an efficient implementation.

Application: A self-signed voice archive

Application of continuous protection for archiving voice

Archived digital documents are always susceptible to undetected forgery and need special protection of their integrity. Ideally an integrity protection for creating a secure archive for internet telephony should be applied already during the ongoing conversation. This is easy using our basic idea. The architecture is simple



Time-stamping service for **long-term security**

Separation of duties between long-term archive and security module that secures and signs archived calls

VSec can be a passive listener or have an active role and **enforce policies**

Trusted time source or **time-stamping authority** to securely pinpoint exact start of call

Only one point in channel A⇔B needs to be digital and packet-based

Main design principle is minimal requirements at the communication clients. The securing component **VSec** can be placed at the site of either of the parties A or B or anywhere in between, as long as at least some part of the communication is based on SIP/RTP. **ARC** is the component to which the secured VoIP-communication is submitted. It handles long-term storage of archived conversations. VSec listens to the communication and secures it. It can either have a passive role or a dual role, also enforcing policies on the quality of the conversation. It then plays the role of a **reference monitor**. T1 und T2 are additional **time-stamping authorities** to raise resilience against attacks and ensure long-term security.

VSec is implemented in the demonstrator as an **outbound proxy** substituting A's original outbound proxy. The proxy modifies RTP ports and IP addresses contained in the SIP packets/SDP bodies redirecting them to itself and in turn forwards them to the original recipients.

Archive data format

Intervals are signed by VSec using PKCS#7 with the private key of VSec.

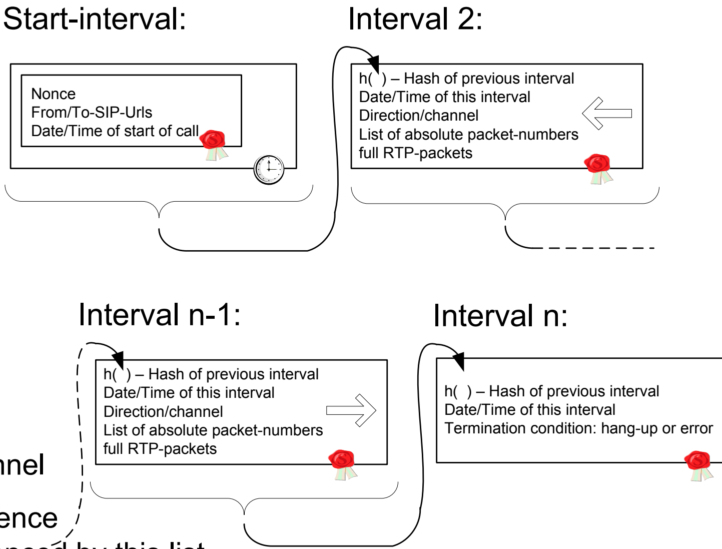
The **first interval** contains SIP-URLs of caller and callee, date and time of start of call, mapping of RTP-payload numbers to codecs

A **PKCS#7 container** contains whole certificate chain (can be omitted in all following intervals) and is additionally time-stamped by T1

Regular intervals stem from either channel A to B or the other direction and stores flag for direction/channel

They contain the time of interval, list of absolute sequence numbers of packets, the complete RTP packets referenced by this list, including their payload type and the truncated timestamps and sequence numbers.

The **final interval** contains the reason for termination: e.g. regular hangup by A or B, QoS-under-run, or tamper-detection



Start-interval:

Nonce
From/To-SIP-Urls
Date/Time of start of call

Interval 2:

h( ) – Hash of previous interval
Date/Time of this interval
Direction/channel
List of absolute packet-numbers
full RTP-packets

Interval n-1:

h( ) – Hash of previous interval
Date/Time of this interval
Direction/channel
List of absolute packet-numbers
full RTP-packets

Interval n:

h( ) – Hash of previous interval
Date/Time of this interval
Termination condition: hang-up or error

Channel quality: Treatment of packet loss

Any service method based on VoIP must handle different levels of channel quality. The signing method treats **packet loss** by **only signing what was actually received**. If packet loss is too large or generally quality of service (QoS) is too low, something must be done (or else an attacker could even use low QoS to hide meaning of speech)!

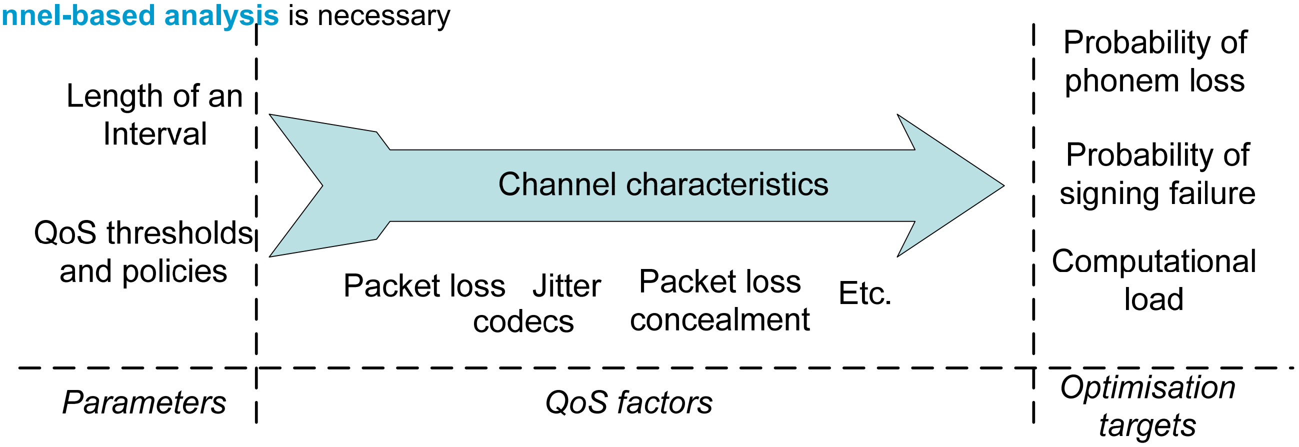
QoS policy implementation

During voice signing the system constantly **monitors the QoS** of the voice connection, in particular packet loss and jitter. If a certain **threshold is under-run** then either the connection quality is poor and the participants cannot understand each other with a sufficient quality, or there is an ongoing attempt to attack the communication. It is a **matter of policy** how to deal with this QoS under-run:

- **ignore** it completely and continue to archive
- **notify** the user while continuing the archiving
- **abort the signing**, but not the conversation
- **terminate the call**

Termination of the call was the **favoured policy** for maximum security and because the QoS threshold is seldom reached without a breakdown of the connection, insufficient understandability, hang-ups or software timeouts. Forced termination of calls was implemented by injecting a BYE command and terminating SIP and RTP forwarding.

The length of a interval and the QoS threshold are the main **free parameters** in the concept, to adjust and tune understandability, scalability, computational resources for signing and time to attack. In general, a **channel-based analysis** is necessary



Length of an Interval

QoS thresholds and policies

Channel characteristics

Packet loss

Jitter

codecs

Packet loss concealment

Etc.

Probability of phonem loss

Probability of signing failure

Computational load

Parameters

QoS factors

Optimisation targets

Worst case analysis (by thumb)

What kind of **phonetic items should not be lost**?
Take a fast-spoken **negation “No!”.** The duration of a “no” is approximately 150ms.
Take **100ms as a threshold** for the minimum time span in which understandability must be ensured.

The G.711 codec produces 64kbit/s, RTP packets come with 160 bytes of payload. Therefore a choice of **1 second for interval length** and at most **5 lost packets per interval** as QoS threshold, together with the policy that **no consecutive packets must be lost** is safe

Signing is then **computationally feasible on a 700MHz PC** without special crypto hardware.
The QoS of VoIP connections seems mostly to allow for signing with these settings.

Towards a more detailed analysis

Human-centric measures of understandability should be taken into account, similar to ITU‘s **mean opinion score** (MOS). They need to be **evaluated from a semantic** (ambiguities, attacks on meaning) **and legal viewpoint** (non-repudiation and probative force).

Bursts of lost packets pose a serious problem (as to any VoIP application).

QoS policies and target characteristics need formal treatment for optimisation.

References

VoIP Technology

J. Kavanagh: Voice over IP special report: From dial to click. http://www.computerweekly.com/Articles/2006/02/14/214129/VoiceoverIPspecialreportFromdialtoclick.htm, visited 1.3.2006

R. Barbieri; D. Bruschi; E. Rosti: Voice over IPSec: Analysis and Solutions. Proceedings of the 8th Annual Computer Security Applications Conference, IEEE, 2002

P. Markopoulou, F. A. Tobagi, and M. J. Karam: Assessment of VoIP quality over internet backbones. In: Proceedings of INFOCOM 2002, pp. 150–159, IEEE 2002.

C. Hett, N. Kuntze, A. U. Schmidt: A secure archive for Voice-over-IP conversations. In: Proceedings of the third VoIP Security Workshop, Berlin, 1.-2. June 2006.

J. Rosenberg, C. Huitema, R. Mahy: Traversal Using Relay NAT (TURN). http://www.jdrosen.net/midcom_turn.html

Voice Forensic and usability issues

H. Hollien: Forensic Voice Identification. Academic Press, London, 2001.

C. Goodwin: Conversational organization: Interaction between speakers and hearers. Academic Press, New York, 1981.

C. Hoene, H. Karl, and A. Wolisz: A perceptual quality model for adaptive VoIP Applications. In Proceedings of SPECTS'04, San Jose, CA, July 2004.

P. Landrock, T. Pedersen: WYSIWYS? What you see is what you sign? Information Security Technical Report, 3 (1998) 55–61

J. Gonzalez-Rodriguez, et al. :Robust estimation, interpretation and assessment of likelihood ratios in forensic speaker recognition. In Computer Speech & Language, 20 (2006) 331-355

M. Gamer, H.-G. Rill, G. Vossel, H. W. Gödert: Psychophysiological and vocal measures in the detection of guilty knowledge. In International Journal of Psychophysiology, 60 (2006) 76-87

A. Alexander, F. Botti, D. Dessimoz, A. Drygajlo: The effect of mismatched recording conditions on human and automatic speaker recognition in forensic applications. In Forensic Science International, 146 Supplement 1 (2004) S95-S99

L. Cerrato, M. Falcone, A. Paoloni: Subjective age estimation of telephonic voices. In Speech Communication, 31 (2000) 107-112

L.-J. Boë: Forensic voice identification in France. In Speech Communication, 31 (2000) 205-224

Standards

CEN Workshop Agreement 14170

Trusted Computing Group. TCG Specification Architecture Overview

M. Spencer. et al.: IAX: Inter-Asterisk eXchange Version 2. http://www.ietf.org/internet-drafts/draft-guy-iax-01.txt

H. Schulzrinne, et al.: RTP: A Transport Protocol for Real-Time Applications. RFC 3550, July 2003. http://www.ietf.org/rfc/rfc3550.txt

M. Baugher, et al.: The Secure Real-time Transport Protocol (SRTP). RFC 3711, March 2004. http://www.ietf.org/rfc/rfc3711.txt

M. Handley and V. Jacobson. SDP: Session Description Protocol. RFC 2327 (Proposed Standard), Apr. 1998. Updated by RFC 3266.

J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320.

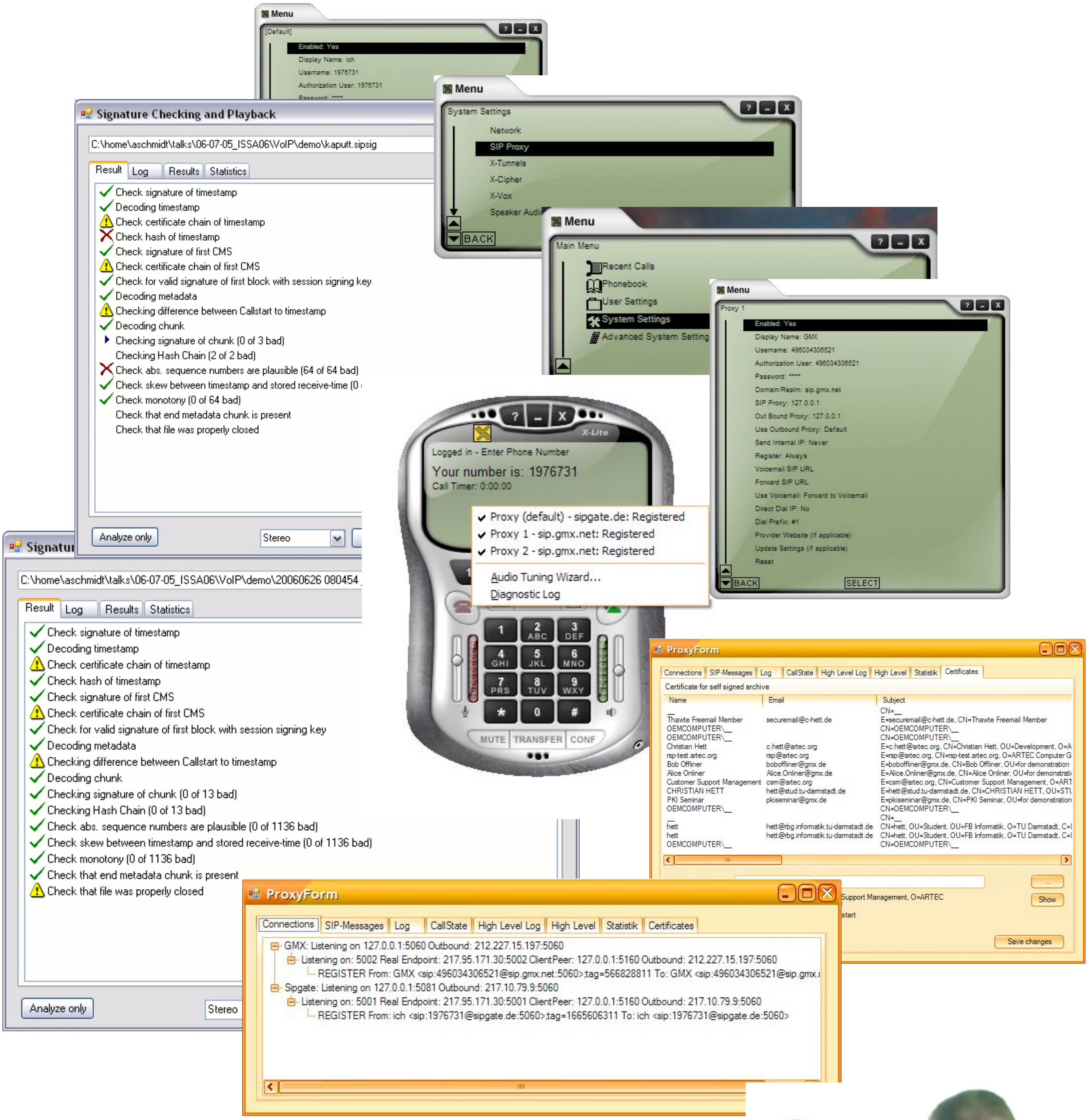
Related Work

D. Lekkas and D. Gritzalis. Cumulative notarization for long-term preservation of digital signatures. Computers & Security, 23(5):413–424, 2004. 2

A. U. Schmidt and Z. Loebl. Legal security for transformations of signed documents: Fundamental concepts. In D. Chadwick and G. Zhao, editors, EuroPKI 2005, volume 3545 of Lecture Notes in Computer Science, pages 255–270, 2005.

A. Perrig, R. Canetti, J. D. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In IEEE Symposium on Security and Privacy, pages 56–73, 2000.

Contact



Signature Checking and Playback

C:\home\aschmidt\talks\06-07-05_ISSA06\VoIP\demo\kaputt.sipsig

- ✓ Check signature of timestamp
- ✓ Decoding timestamp
- ⚠ Check certificate chain of timestamp
- ✗ Check hash of timestamp
- ✓ Check signature of first CMS
- ⚠ Check certificate chain of first CMS
- ✓ Check for valid signature of first block with session signing key
- ✓ Decoding metadata
- ⚠ Checking difference between Callstart to timestamp
- ✓ Decoding chunk
- ▸ Checking signature of chunk (0 of 3 bad)
- Checking Hash Chain (2 of 2 bad)
- ✗ Check abs. sequence numbers are plausible (64 of 64 bad)
- ✓ Check skew between timestamp and stored receive-time (0
- ✓ Check monotony (0 of 64 bad)
- Check that end metadata chunk is present
- Check that file was properly closed

Signatur

C:\home\aschmidt\talks\06-07-05_ISSA06\VoIP\demo\20060626 080454

- ✓ Check signature of timestamp
- ✓ Decoding timestamp
- ⚠ Check certificate chain of timestamp
- ✓ Check hash of timestamp
- ✓ Check signature of first CMS
- ⚠ Check certificate chain of first CMS
- ✓ Check for valid signature of first block with session signing key
- ✓ Decoding metadata
- ⚠ Checking difference between Callstart to timestamp
- ✓ Decoding chunk
- ✓ Checking signature of chunk (0 of 13 bad)
- ✓ Checking Hash Chain (0 of 13 bad)
- ✓ Check abs. sequence numbers are plausible (0 of 1136 bad)
- ✓ Check skew between timestamp and stored receive-time (0 of 1136 bad)
- ✓ Check monotony (0 of 1136 bad)
- ✓ Check that end metadata chunk is present
- ⚠ Check that file was properly closed

[Default]

Enabled: Yes
Display Name: ich
Username: 1976731
Authorization User: 1976731
Password: ****

System Settings

Network
SIP Proxy
X-Tunnels
X-Cipher
X-Vox
Speaker Audi

Logged in - Enter Phone Number
Your number is: 1976731
Call Timer: 0:00:00

- ✓ Proxy (default) - sipgate.de: Registered
- ✓ Proxy 1 - sip.gmx.net: Registered
- ✓ Proxy 2 - sip.gmx.net: Registered

Proxy 1

Enabled: Yes
Display Name: GMX
Username: 496034306521
Authorization User: 496034306521
Password: ****
Domain/Realm: sip.gmx.net
SIP Proxy: 127.0.0.1
Out Bound Proxy: 127.0.0.1
Use Outbound Proxy: Default
Send Internal IP: Never
Register: Always
Voicemail SIP URL
Forward SIP URL
Use Voicemail: Forward to Voicemail
Direct Dial IP: No
Dial Prefix: #1
Provider Website (if applicable)
Update Settings (if applicable)
Reset

Certificate for self signed archive

Name	Email	Subject
__		CN=__
Thawte Freemail Member	securemail@c-hett.de	E=securemail@c-hett.de, CN=Thawte Freemail Member
OEMCOMPUTER__		CN=OEMCOMPUTER__
OEMCOMPUTER__		CN=OEMCOMPUTER__
Christian Hett	c.hett@artec.org	E=c.hett@artec.org, CN=Christian Hett, OU=Development, O=A
rsp-test.artec.org	rsp@artec.org	E=rsp@artec.org, CN=rsp-test.artec.org, O=ARTEC Computer G
Bob Offliner	boboffliner@gmx.de	E=boboffliner@gmx.de, CN=Bob Offliner, OU=for demonstration
Alice Onliner	Alice.Onliner@gmx.de	E=Alice.Onliner@gmx.de, CN=Alice Onliner, OU=for demonstrati
Customer Support Management	csm@artec.org	E=csm@artec.org, CN=Customer Support Management, O=ART
CHRISTIAN HETT	hett@stud.tu-darmstadt.de	E=hett@stud.tu-darmstadt.de, CN=CHRISTIAN HETT, OU=STU
PKI Seminar	pkiseminar@gmx.de	E=pkiseminar@gmx.de, CN=PKI Seminar, OU=for demonstration
OEMCOMPUTER__		CN=OEMCOMPUTER__
__		CN=__
hett	hett@rbg.informatik.tu-darmstadt.de	CN=hett, OU=Student, OU=FB Informatik, O=TU Darmstadt, C=I
hett	hett@rbg.informatik.tu-darmstadt.de	CN=hett, OU=Student, OU=FB Informatik, O=TU Darmstadt, C=I
OEMCOMPUTER__		CN=OEMCOMPUTER__

Support Management, O=ARTEC

start

```
⊟ GMX: Listening on 127.0.0.1:5060 Outbound: 212.227.15.197:5060
   ⊟ Listening on: 5002 Real Endpoint: 217.95.171.30:5002 ClientPeer: 127.0.0.1:5160 Outbound: 212.227.15.197:5060
      REGISTER From: GMX <sip:496034306521@sip.gmx.net:5060>;tag=566828811 To: GMX <sip:496034306521@sip.gmx.r
⊟ Sipgate: Listening on 127.0.0.1:5081 Outbound: 217.10.79.9:5060
   ⊟ Listening on: 5001 Real Endpoint: 217.95.171.30:5001 ClientPeer: 127.0.0.1:5160 Outbound: 217.10.79.9:5060
      REGISTER From: ich <sip:1976731@sipgate.de:5060>;tag=1665606311 To: ich <sip:1976731@sipgate.de:5060>
```

Fraunhofer Institute for Secure Information Technology SIT

Dr. Andreas U. Schmidt

Dipl.-Inform. Nicolai Kuntze

Rheinstraße 75
D-64295 Darmstadt

Telefon: +49-6151-869-60227

Telefax: +49-6151-869-224

mail: andreas.u.schmidt@sit.fraunhofer.de

www: http://www.sit.fraunhofer.de