Real-time Encrypted Speech Communication Over Low Bandwidth Channels

Markus Kammerstetter
Motivation

- broadly existing speech communication systems usually only provide limited voice security

- GSM security can be considered broken today
  - broken encryption algorithms
  - only client-side authentication to the network

- UMTS is more secure, but suffers from GSM interoperability issues
Consequences

- active attacks can evade encryption
- eavesdropping is possible
- conversations are not secure
- Oh no, now everyone knows my secrets!
Existing Solutions

- end-to-end speech encryption
- usually closed design, user has to trust manufacturer that device is secure
- constrained to a single medium (e.g. GSM)
- need to have separate solution for each medium
- expensive
Our Goals

- versatile and generic embedded system, usable on a broad range of media
- high security
- ultra low bandwidth requirements ($\leq 9600$ baud)
- based solely on established and practically proven principles
- open and affordable design
**Information Security Properties**

- **Confidentiality:** Alice and Bob want to be sure that no one else can read their messages.

- **Integrity:** Besides Alice and Bob no one should be able to change the content of a message without notice.

- **Authenticity:** Alice and Bob need proof that exchanged messages originated from each other.

- **Perfect Forward Secrecy:** A compromise of secret key material must not compromise the security of previous communications.

- **Repudiation:** It should be infeasible that Alice or Bob can prove the content of a conversation to a third party.
Customly designed hardware including

- **Atmel AT91SAM9260** (180 MHz) ARM9 System On Chip (SoC)
- **AMBE-3000** speech compression DSP
- **TLV320AIC23** audio codec
Interconnected Communication Units

Two interconnected encrypted speech communication units
- ported bootloader stages (*AT91Bootstrap*, *U-Boot*)
- ported Linux kernel *2.6.36-rc1*
- implemented ALSA SoC (ASoC) **drivers** for TLV320AIC23 **codec**
- implemented transparent ALSA **speech compression plugin** for AMBE-3000 DSP
- implemented **cryptophone** application based on **libtomcrypt**
- overall implementation: ~16500 lines of C source code
• speech is captured and played back through the **TLV320AIC23** audio codec

• ALSA plugin exchanges data with the **DSP** for de-/compression
Authenticated Key Exchange

- based on **OTR** (Off The Record) protocol
- employs formally proven **SIGMA** protocol (also used in IKE/IPSec)
- makes use of Elliptic Curve Cryptography (ECC)
initially Alice and Bob do *not* have the public keys of each other

out-of-band authentication necessary

we use **Short Authentication Strings** (SAS)

strings are based on fingerprint of **DH secret key**

users *verbally compare* strings
Alice

Please read these words to your communication partner:
preshrunk hurricane village maverick

Check that the response from the other party matches the following words:
talon tambourine snapline Cherokee

Bob

You should now hear the following words from your communication partner:
preshrunk hurricane village maverick

If these words match, please respond with:
talon tambourine snapline Cherokee
- **AES-256** in Counter (CTR) mode for encryption
- **HMAC** for message authentication
- separate key pairs for each direction

![Diagram]

Compressed speech from microphone is encrypted using AES-256 in CTR mode, and HMAC for message authentication. The key stream is generated for each direction, and separate key pairs are used for encryption and decryption.
- we implemented two highly configurable working prototypes
- tested for baud rates down to **4800 baud**
- acceptable **speech quality** at low baud rates
- one-way latency ~200ms, but depends on available baudrate
- echo cancellation and comfort noise generation
tradeoff between required baudrate and latency

required baudrate vs. latency @2250bps speech rate

- measured total latency
- calculated additional latency
- measured baudrate
- calculated baudrate

Markus Kammerstetter
Future Work

- **Hybrid Forward Error Correction** (HARQ) to deal with bit errors
- Support for channels with high **BER** (Bit Error Rate)
- Miniaturization
- Software speech codec (Codec2) instead of AMBE-3000 DSP
Conclusion

- we implemented a fully working system
- due to generic design, it is not restricted to a specific medium
- we achieved all security goals and information security properties
- no tests on cellular networks have been done yet
More detailed information can be found in my master’s thesis.

The end, thank you for your attention