An Introduction to Voice over IP Security

July 2006

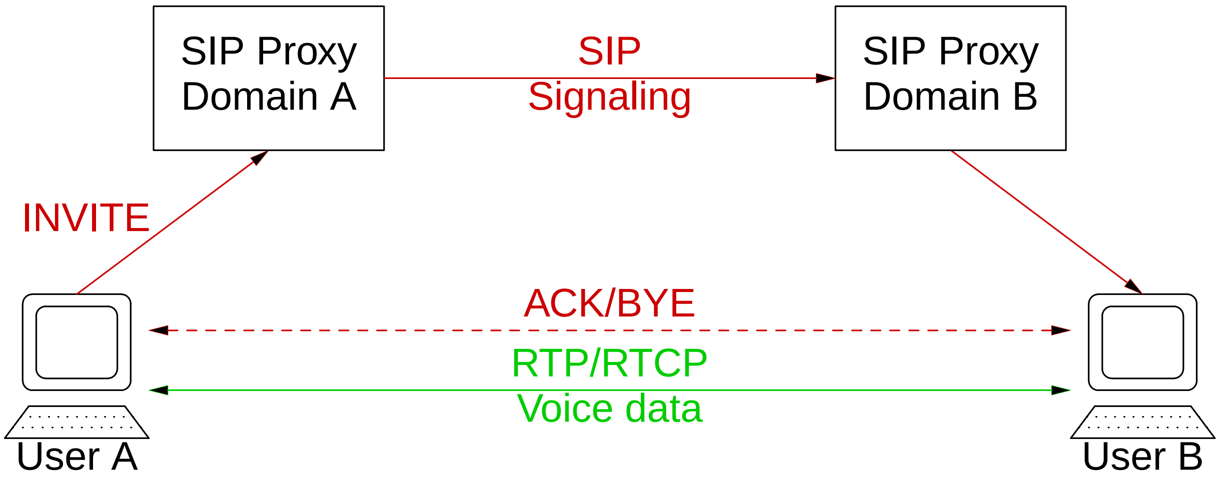
Holger.Zuleger@hznet.de

> c

Preface

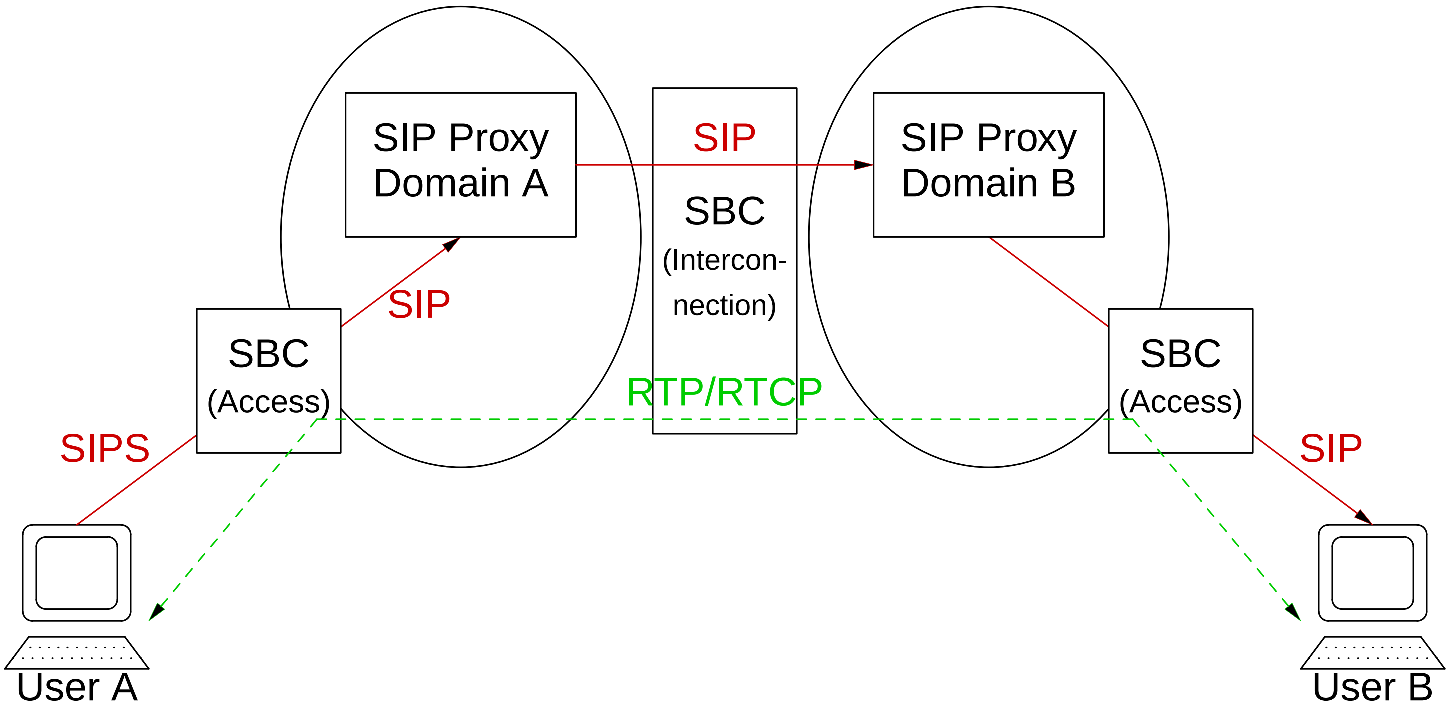
- What is meant by security?
  - Not address or topology hiding
  - Not (D)DoS prevention
  - Not user authorization
  - Just encryption, integrity and confidentially
- What is meant by encryption?
  - Encryption of RTP data (no eavesdropping)
  - Encryption of Signaling traffic (Not necessarily privacy)
- All statements in this presentation based on reading the following material ... and shameless copying something
  - SIP (RFC3261), SDP (RFC2327), RTP (RFC3550), SRTP (RFC3711)
  - ZRTP (draft-zimmermann-avt-zrtp-01)
  - SIP Security (Andreas Steffen / 3rd DeNIC-ENUM Tag)
- No practical experiences, nothing tested!

Classical VoIP Trapezoid



- Bearer traffic flows end to end
  Good for end to end encryption
- Hop by hop signaling path via proxy
  Assurance of signaling security is difficult to manage
  - Secure SIP allows encryption up to the first hop
  - S/MIME for end to end encryption (only SDP)

VoIP in Carrier Networks



- SBC (B2BUA) used for topology hiding
  B2BUA terminates session and rewrites most of the SIP headers
- RTP has to flow through SBC (media relay)
  SBC must be able to read SIP header **and** SDP body

VoIP Security Variants

IPsec (Network Layer)

- Large overhead / QoS problem / Timing problem
- End to end security scheme difficult to set up
  NAT / Proxy / B2BUA (Session Border Controller)
- PKI required
- VoIP via IPsec VPNs may be feasible

SSL/TLS (Transport Layer)

- Only for signaling (SIPS) / Not for RTP
- Presumes the use of TCP instead of UDP as transport protocol
  Bad performance
- PKI required
- End to end security scheme difficult to set up

SRTP (Presentation/Application Layer)

- See the following slides

SRTP – Secure RTP/RTCP (RFC3711)

- Profile of RTP/RTCP (RFC 3550 / RFC 3551)
  Overhead: 4 - 14 Byte (4 Byte opt. MKI, 4-10 Byte authentication Tag)

- Defined for unicast and multicast RTP

- Out of band key management
  - Only one Masterkey required
  - All SRTP keys derived from master key
    - Session encryption key
    - Session authentication key
    - Session salting key
  - Independent session keys for SRTP and SRTCP

- Default encryption algorithm: AES-CM 128 Bit

- Default message authentication algorithm: HMAC-SHA1

SRTP – Secure RTP/RTCP (2)

- Encryption and authentication (integrity) of RTP/RTCP packets
  - — RTP payload encryption
  - — Authentication is recommended but optional (allows header compression)

<table>
<tr><td>RTP</td><td>Version, Flags</td><td>Payload Type</td><td colspan="2">sequence #</td></tr>
<tr><td></td><td colspan="4">timestamp</td></tr>
<tr><td></td><td colspan="4">sync. source identifier</td></tr>
<tr><td></td><td colspan="4">cont. source identifier</td></tr>
<tr><td></td><td colspan="4">Header extension (opt)</td></tr>
<tr><td></td><td colspan="4">RTP Payload</td></tr>
<tr><td></td><td colspan="2"></td><td>padding</td><td>pad count</td></tr>
<tr><td>SRTP</td><td colspan="4">SRTP master key identifier (4 Byte opt)</td></tr>
<tr><td></td><td colspan="4">authentication tag (4-10 Byte recommended)</td></tr>
</table>

- Also some minor header changes for SRTCP

SRTP – Keymanagement

- Derivation of session keys out of one master key
- Key exchange via external mechanism
  - — Session Description Protocol (RFC2327)
    Key Mangement Extensions for SDP (encrypted signaling required)
  - — MIKEY (RFC3830)
    Multimedia Internet KEYing (no encrypted signaling required)
  - — SDP Security Descriptions for Media Streams
    (draft-ietf-mmusic-sdescriptions-12.txt)
  - — Encrypted key Transport for SRTP (draft-mcgrew-srtp-ekt-00.txt)
  - — HIP-SRTP (draft-tschofenig-hiprg-hip-srtp-01.txt)
    Using SRTP transport format with HIP
  - — ZRTP (draft-zimmermann-avt-zrtp-01.txt)
    Extension to RTP for Diffie-Hellman Key Agreement for SRTP
- Currently available mechanism?
  SDP, ZRTP, MIKEY(?)

SIP & SDP

- Session Initiation Protocol (RFC3261) for call signaling
  - — Header format is similar to HTTP
  - — UDP Port 5060 used (recommended)
    TCP is also allowed (required for SIPS)
  - — Responsible for connection setup and release
    INVITE, OK, ACK, BYE, CANCEL
  - — Registration service for mobile user agents
    REGISTER
  - — Uses DNS for routing (SRV-Record RFC3263; NAPTR).
- Session Description Protocol (RFC 2327) for parameter exchange
  - — Body of SIP-Messages
  - — Looks (a little bit) like sendmail mail queue format
  - — Contact address (ip address, port #) `c=IN IP4 1.25.43.66`
  - — Codec `m=audio 7078 RTP/AVP 8 0 2 102 100 97 101`
  - — (Master)Key for SRTP `k=clear:geheim`

SIP/SDP Example Packet

```
INVITE sip:642022@example.net SIP/2.0
Via: SIP/2.0/udp 1.25.43.66:5060;branch=z9hG4bK5E04B432A7CE4D494016D27E86B2D
From: <sip:hoz@sip.example.de>;tag=1167B5B5D227AA6656B12714F8441
To: <sip:642022@example.net>
Call-ID: 135F08716ED07E5D0C0B7B855BC21@1.25.43.66
CSeq: 9 INVITE
Contact: <sip:hoz@1.25.43.66;uniq=964E34A1883165EE1829BFAE36988>
Max-Forwards: 70
User-Agent: AVM FRITZ!Box Fon WLAN 7170 29.04.02 (Jan 25 2006)
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, UPDATE, PRACK, INFO, SUBSCRIBE, NOTIFY, REFER, MESSAGE
Content-Type: application/sdp
Accept: application/sdp, multipart/mixed
Content-Length:   381

v=0
o=user 10512055 10512055 IN IP4 1.25.43.66
s=call
c=IN IP4 1.25.43.66
t=1144829986 1144833586
k=base64:acx4fimF1pQdu6y2QTzttXjr5Z3eOVmmVu4YRZQoKqc=
m=audio 7078 RTP/AVP 8 0 2 102 100 97
a=sendrecv
a=rtpmap:2 G726-32/8000
a=rtpmap:102 G726-32/8000
a=rtpmap:100 G726-40/8000
a=rtpmap:97 iLBC/8000
a=fmtp:97 mode=30
a=rtcp:7079
```

SIPS – SIP-Secure over TLS

- SIPS is like HTTPS
  Is set on top of TCP only
- Signaling over sips URI: `sips:user@example.de;transport=tcp`
  Demands for TLS along the (signaling)path
- Recommended for mobile user agents
  Allows for „first mile“ encryption (but now deprecated)
- Server authentication via Certificate
- Client authentication (mostly) via username/digest
  What about incoming invites? How to authenticate the UA?
- Client authentication via Certificate possible
  Difficult because of changing contact address / network attachment points
- Only Hop by Hop Security
  Enables legal (and not so legal) interception

S/MIME − secure SDP

- Data format based on S/MIME mail
- Encryption of the SDP portion of the SIP messsage
  See example on next slide
- End-to-End or Hop by Hop allowed
  Tunneled (and S/MIME encrypted) SDP also allowed
- Supports UDP or TCP
  TCP is recommended because of UDP fragmentation
- Prior (public)key exchange necessary
  (We have to encrypt the SDP with the public key of the communication partner)

SIP – S/MIME Example (IPv6) Packet

```
INVITE sip:642022@example.net SIP/2.0
Via: SIP/2.0/udp [2001:db8::27:2]:5060;branch=z9hG4bK5E04B432A7CE4D494016D27E86B2D
From: <sip:hoz@example.de>;tag=1167B5B5D227AA6656B12714F8441
To: <sip:642022@example.net>
Call-ID: 135F08716ED07E5D0C0B7B855BC21@example.de
CSeq: 9 INVITE
Contact: <sip:hoz@[2001:db8::27:2];uniq=964E34A1883165EE1829BFAE36988>
Max-Forwards: 70
Content-Type: multipart/signed;boundary=e4ef8847482d240d0
Accept: application/sdp, multipart/mixed
Content-Length:   3381

--e4ef8847482d240d0
Content-Type: application/pkcs7-mime
smime-type=envelopeddata; name=smime.p7m
Content-Disposition: attachment;handling=required;filename=smime.p7m
Content-Transfer-Encoding: binary
*** envelopedData object containing encrypted SDP body ***
*  v=0
*  o=- 0 0 IN IP6 2001:db8::27:2
*  c=IN IP6 2001:db8::27:2
*  k=base64:acx4fimF1pQdu6y2QTzttXjr5Z3eOVmmVu4YRZQoKqc=
*   ...
**********************************************************
--e4ef8847482d240d0
Content-Type: application/pkcs7-signature;name=smime.p7s
Content-Disposition: attachment;handling=required;filename=smime.p7s
Content-Transfer-Encoding: binary
  ... signedData object containing signature ...
--e4ef8847482d240d0
```

ZRTP – Zimmermann secure RTP

- Use of SRTP for media encryption (end to end)
- No signaling protocol required, but ZRTP **could** use SDP/MIKEY keys
- Instead, ZRTP uses RTP extensions/options for key exchange
- Diffie Hellman Key exchange with man-in-the-middle detection
  Defined by draft-zimmermann-avt-zrtp-01

— Reference implementation works under MAC OSX, Linux and Windows XP

— "Bump on the cord" implementation

— Works with (nearly) all SIP soft clients

— Encrypt and decrypt voice packets on the fly

ZRTP (2)

- ZRTP use short authentication strings (SAS) for MITM detection
  - Both partners read a short string to each other
  - Compare with written down string on screen
  - If both strings match, the session key will be stored in a cache
    Like ssh keys
- This kind of authentication is needed only once
- SAS algorithm is difficult to use if callee is not a person

Secure VoIP (SRTP) in the field

- Hardware
  - SNOM Phones (e.g. 360; also Softphone 360)
  - Cisco 28xx Router
  - Siemens Hi-Path?
- Software
  - Patches for Kphone / Asterisk (Linux)
  - CounterPath (eyeBeam, formerly x-ten)
  - VOCAL (www.vovida.org)
  - Application independent SRTP (ZRTP) (Linux, MAC, Windows)
  - minisip (Linux, Windows, PocketPC)
- Librarys
  - libsrtp (srtp.sourceforge.net/srtp.html)
  - S/MIME enabled SIP Stack (www.sipfoundry.org/reSIProcate)
- Secure VoIP Providers
  - Sipgate
  - dus.net

Summary

- Secure RTP
  - + End-to-End encryption feasible
  - + Hard- and Software available
  - + Already deployed by some SIP carrier
  - − Could be difficult if media relays in-between the path
  - − Most of the media gateways don’t support SRTP
- SIPS
  - + Allows „first mile“ encryption (only for signaling)
  - + Useful for SRTP master key exchange
    SIP-Proxy should also work as (secure) media gateway
  - − No end-to-end encryption/privacy
- S/MIME
  - + Secure End-to-End key exchange
  - − Not feasible with B2BUA because of SDP inspection
  - − Requires PKI

Questions ?

http://www.hznet.de/security/voipsec.pdf

Questions ?

http://www.hznet.de/security/voipsec.pdf

Thank you for your attention

CONTENTS