Wiretapping End-to-End Encrypted VoIP Calls
Real-World Attacks on ZRTP

Dominik Schürmann, Fabian Kabus, Gregor Hildermeier, Lars Wolf, 2017-07-18
End-to-End Security for Voice Calls

**No** End-to-End Security

- PSTN (Public Switched Telephone Network)
- SIP + (S)RTP (Session Initiation Protocol + Secure Real-Time Transport Protocol)
End-to-End Security for Voice Calls

**No End-to-End Security**

- PSTN (Public Switched Telephone Network)
- SIP + (S)RTP (Session Initiation Protocol + Secure Real-Time Transport Protocol)

**End-to-End Encryption**

- SIP + DTLS-SRTP (SIP + Datagram Transport Layer Security-SRTP)
End-to-End Security for Voice Calls

No End-to-End Security
- PSTN (Public Switched Telephone Network)
- SIP + (S)RTP (Session Initiation Protocol + Secure Real-Time Transport Protocol)

End-to-End Encryption
- SIP + DTLS-SRTP (SIP + Datagram Transport Layer Security-SRTP)

End-to-End Encryption & Authentication
- SIP + SRTP + ZRTP
End-to-End Security for Voice Calls

No End-to-End Security
- PSTN (Public Switched Telephone Network)
- SIP + (S)RTP (Session Initiation Protocol + Secure Real-Time Transport Protocol)

End-to-End Encryption
- SIP + DTLS-SRTP (SIP + Datagram Transport Layer Security-SRTP)

End-to-End Encryption & Authentication
- SIP + SRTP + ZRTP
Man-in-the-Middle (Evil Operator)

SIP with Encryption-only:

Alice

INVITE B@example.com
From: A@example.com

200 OK
From: A@example.com

Bob

INVITE B@example.com
From: A@example.com
modified:
INVITE mitm@localhost
From: A@example.com

200 OK
From: mitm@localhost
header added:
mitm: A@example.com

Valid Session!

MitM Client

INVITE mitm@localhost
From: A@example.com
modified:
INVITE B@example.com
From: A@example.com

200 OK
From: A@example.com

SIP Server

INVITE B@example.com
From: mitm@localhost
mitm: A@example.com
modified:
INVITE B@example.com
From: A@example.com

200 OK
From: A@example.com

Valid Session!

1 4
5 8
9

10
connect & record

2 7 3 6
9 10

/one.lf/three.lf/two.lf/zero.lf/one.lf/seven.lf/-zero.lf/seven.lf/-one.lf/eight.lf

Dominik Schürmann

Wiretapping End-to-End Encrypted VoIP Calls
Encryption & Authentication with ZRTP:

Alice

INVITE B@example.com
From: A@example.com

200 OK
From: A@example.com

Bob

ZRTP SAS: bz4f

Valid Session!

MitM Client

Bob

ZRTP SAS: utd9

Valid Session!

Institute of Operating Systems and Computer Networks

2017-07-18 | Dominik Schürmann | Wiretapping End-to-End Encrypted VoIP Calls | Page 4 of 13
ZRTP Attacks

ZRTP

- Complex Protocol
- Authenticates Diffie-Hellman key exchange
- Authentication by comparison of Short Authentication Strings (SAS)
- Hash Commitment constraints online-attacker to one try per call

Evaluation of Real-World Implementations

- Excluded closed-network implementations
- Excluded attacks with speech synthesis
- Assume correctly compared SAS
## Evaluation

### Apps

<table>
<thead>
<tr>
<th>Application</th>
<th>OS</th>
<th>Version</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acrobits Softphone</td>
<td>iOS</td>
<td>5.8.1</td>
<td>-</td>
</tr>
<tr>
<td>CSipSimple</td>
<td>Android</td>
<td>1.02.03</td>
<td>ZRTP4PJ</td>
</tr>
<tr>
<td>Jitsi</td>
<td>Win, Lin, MacOS</td>
<td>2.9.0</td>
<td>ZRTP4J</td>
</tr>
<tr>
<td>Linphone</td>
<td>Android</td>
<td>3.1.1</td>
<td>bzrtp</td>
</tr>
<tr>
<td>Signal</td>
<td>Android</td>
<td>3.15.2</td>
<td>-</td>
</tr>
<tr>
<td>Signal</td>
<td>iOS</td>
<td>2.6.4</td>
<td>-</td>
</tr>
</tbody>
</table>

### Tests

- **Paper**: 7 protocol tests, 4 non-protocol tests
- **Presentation**: Most interesting results
ZRTP in a Nutshell (Highly Simplified)

F1-F4 Hello

\[ pvi = g^{svi} \mod p \]

F5 Commit \((hvi)\)

\[ hvi = hash(pvi) \]

F6 DHPart1 \((pvr)\)

\[ pvr = g^{svr} \mod p \]

F7 DHPart2 \((pvi)\)

\[ DHResult = pvr^{svi} \mod p \]

\[ DHResult = pvi^{svr} \mod p \]

\[ SAS = KDF(DHResult \parallel IDs \parallel HashOfMessages) \]

Confirm F8-F10 – Verbal Comparison of SAS
Check for Invalid Commit

$\text{F1-F4 Hello}$

$\text{pvi} = g^{svi} \mod p$

$\text{hvi} = \text{hash}(pvi)$

$\text{F5 Commit (hvi)}$

$\text{pvr} = g^{svr} \mod p$

$\text{F6 DHPart1 (pvr)}$

$\text{DHResult} = pvr^{svi} \mod p$

$\text{F7 DHPart2 (pvi)}$

$\text{DHResult} = pvi^{svr} \mod p$

$\text{hvi} \overset{?}{=} \text{hash}(pvi)$

$\text{SAS} = \text{KDF(DHResult} \parallel \text{IDs} \parallel \text{HashOfMessages})$

$\text{Confirm F8-F10 – Verbal Comparison of SAS}$
Invalid Commit: Linphone

Figure: Linphone CVE-2016-6271: Probability of hitting a targeted SAS
RFC: Error on Invalid Shared Secret

- ZRTP stores secrets when user confirms SAS
- Cache: ZRTP ID assigned to $rs1 = KDF(DHResult)$ (highly simplified)
- Next call no longer requires Diffie-Hellman and no SAS comparison

RFC

"If either party discovers a cache mismatch, the user agent who makes this discovery must treat this as a possible security event and MUST alert their own user that there is a heightened risk of a MiTM attack [...]"
RFC: Error on Invalid Shared Secret

- Questionable requirement in RFC
- CSipSimple, Linphone do not implement this

Bug in Jitsi (ZRTP4J)

- A new cache entry copies the secrets and flags from the last saved one
- Invalid security warning is raised for new clients

```
1 src/gnu/java/zrtp/zidfile/ZidFile.java

250 250 /**
251 251 * If we reached end of file, then no record with matching ZID
252 252 * \n253 253 * if (unread) {
254 254         rec = new ZidRecord();
255 255         rec.setIdentifier(zid);
256 256         rec.setValid();
257 257         try {
```
Shared Man-in-the-Middle

Attack

1. Call between Eve & Alice, confirm SAS $\Rightarrow rs1_A$ for Eve in Alice’ cache
2. Call between Eve & Bob, confirm SAS $\Rightarrow rs1_B$ for Eve in Bob’s cache
3. Eve conducts MitM attack (evil operator) $\Rightarrow$ No SAS confirmation, Eve has $rs1_A, rs1_B$ in her cache
4. SIP addresses shown: Alice: B@example.com, Bob: A@example.com
Shared Man-in-the-Middle

Attack
1. Call between Eve & Alice, confirm SAS ⇒ rs1_A for Eve in Alice’ cache
2. Call between Eve & Bob, confirm SAS ⇒ rs1_B for Eve in Bob’s cache
3. Eve conducts MitM attack (evil operator) ⇒ No SAS confirmation, Eve has rs1_A, rs1_B in her cache
4. SIP addresses shown: Alice: B@example.com, Bob: A@example.com

Why Does This Work?
- No ID binding to outer protocol
- ZRTP works independent of SIP addresses with random IDs
  ⇒ Cache uses ZRTP ID for lookup
- Alice and Bob’s cache lookup by Eve’s ZRTP ID
Shared Man-in-the-Middle

- Signal: No cache ⇒ Secure
- Acrobits Softphone: RFC-compliant protection
- Other implementations: Insecure
Conclusion

Current Status

- CVE-2016-6271 responsibly disclosed on 2016-07-05, fixed in Linphone 3.2.04
- Upstream fix for Jitsi always reading the last entry from the ID cache
- Signal no longer uses ZRTP (independent decision)

Future

- Most apps fallback to insecure mode
- Discussion about shared MitM attack
- Discussion about security indicators
Conclusion

Current Status

- CVE-2016-6271 responsibly disclosed on 2016-07-05, fixed in Linphone 3.2.04
- Upstream fix for Jitsi always reading the last entry from the ID cache
- Signal no longer uses ZRTP (independent decision)

Future

- Most apps fallback to insecure mode
- Discussion about shared MitM attack
- Discussion about security indicators

Any questions?
Dominik Schürmann <schuermann@ibr.cs.tu-bs.de>
Twitter: @domschuermann
Quiz Time: Security Indicators
Are you end-to-end secure?
Linphone

Alice 00:24

Alice 00:18