Voice over IP Security
Overview on Threats and Solutions

Lorenzo Miniero
lorenzo.miniero@unina.it

Scuola Politecnica e delle Scienze di Base
Università degli Studi di Napoli “Federico II”
Corso di Network Security

December 1st 2017

Outline

1. Some context: VoIP and Standards
   - SIP, SDP and RTP
2. Security Threats
   - Securing Signalling and Negotiation
   - Securing Media (and Media Transfer)
3. Secure Real-time Transport Protocol (SRTP)
   - Key Exchange
   - Secure Description (SDES)
   - Datagram Transport Layer Security (DTLS)
4. WebRTC and Security
   - Identity Providers
   - Private Media Requirements in Privacy Enhanced RTP Conferencing (PERC)

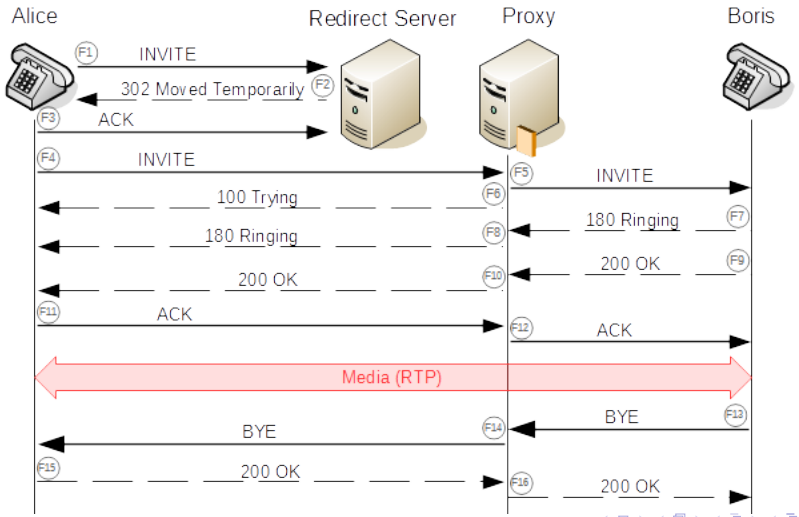
VoIP and Standards

- VoIP quite widespread as of now
  - Several solutions, standards and not
- Internet Enginering Task Force (IETF)
  - http://www.ietf.org
  - Standardized mostly everything on the Internet
    - HTTP, FTP, SMTP, POP3, IMAP, SNMP, etc.
    - Standardized suite of protocols for VoIP as well
    - Session Initiation Protocol (SIP)
    - Session Description Protocol (SDP)
    - Real-Time Transport Protocol (RTP)

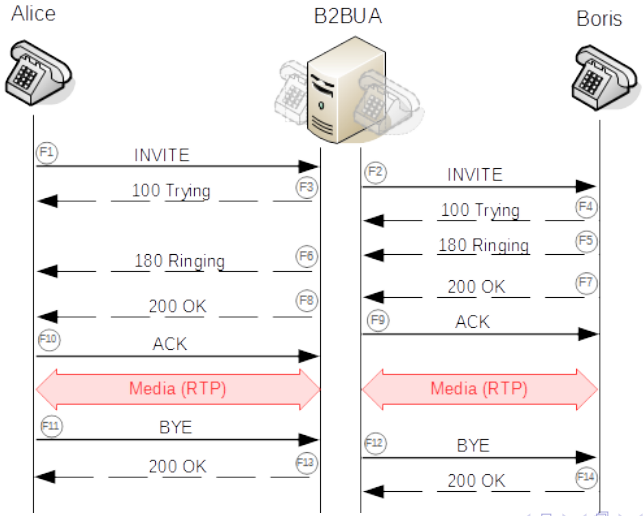
Standard Protocols

- Session Initiation Protocol (SIP)
  - http://tools.ietf.org/html/rfc3261
    - Handles signalling (register, call, answer, hangup, ...)
- Session Description Protocol (SDP)
  - http://tools.ietf.org/html/rfc3264
    - Handles negotiation (media to involve, supported encodings and features, IP/ports, etc.)
- Real-Time Transport Protocol (RTP)
  - http://tools.ietf.org/html/rfc3550
    - Handles transport of media frames between peers

A sample SIP call (with SDP and RTP)



A sample SIP call (with SDP and RTP)



What's wrong with this?

- Several security threats [1]
  - Interception and modification
  - Abuse of Service (fraud)
  - Interruption of Service (Denial of Service attacks)
  - Social attacks (SPAM over Internet Telephony)
- Hard to take care of them all
  - Several protocols/components/topologies involved
  - Completely different attacks

Where can we start?

Securing the protocols themselves!

[1] VoIP Security: technology and challenges (S.Niccolini, NEC)

What's wrong with this?

- Several security threats [1]
  - Interception and modification
  - Abuse of Service (fraud)
  - Interruption of Service (Denial of Service attacks)
  - Social attacks (SPAM over Internet Telephony)
- Hard to take care of them all
  - Several protocols/components/topologies involved
  - Completely different attacks

Where can we start?

Securing the protocols themselves!

[1] VoIP Security: technology and challenges (S.Niccolini, NEC)

SIP Security Issues

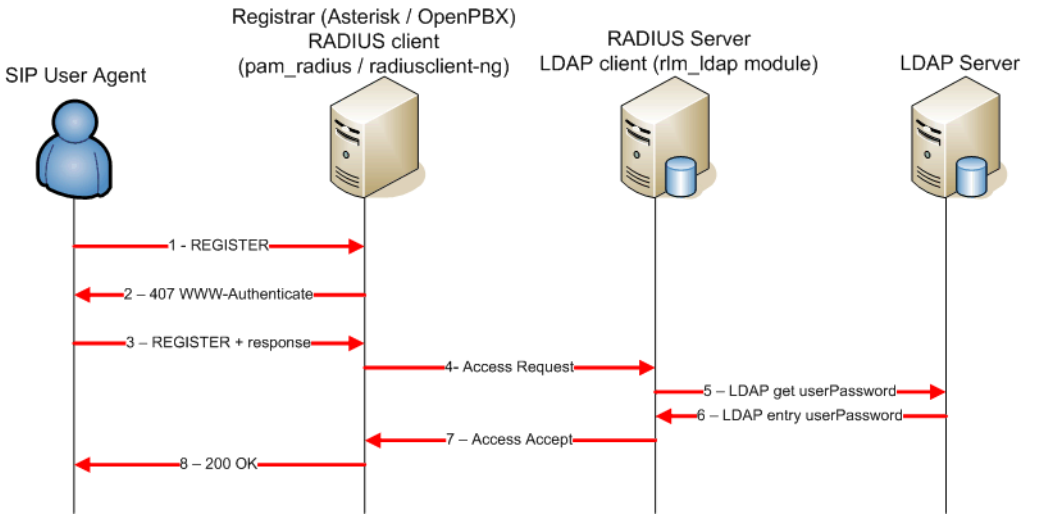
- SIP is usually transported clear-text over UDP
  - Simple, quick and effective, but...
    - ... it (and its SDP too) can be modified and/or intercepted!
- It can be used without authentication
  - Simple for PBX [2]/IVR [3] scenarios, but again...
    - ... it can be easily exploited for fraud/abuse/DoS attacks!
- It can involve several components
  - A good thing, per se
    - It allows for a separation of responsibilities/concerns...
    - ... as long as you can trust them all!

[2] Private Branch eXchange
[3] Interactive Voice Response

Securing SIP/SDP

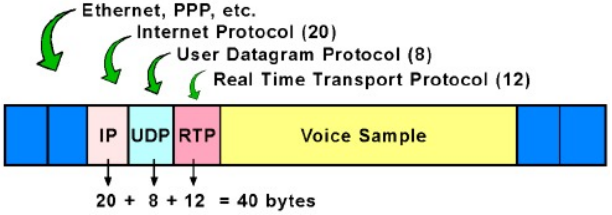
- SIP can be transported over TLS as well
  - Pretty much as HTTPS works
  - Prevents interception/modification...
  - ... but is harder on proxies too
    - UDP != TCP, in terms of SIP usage and scalability
    - Several crypto sessions/contexts to be handled
- SIP supports authentication too
  - UA-Proxy using challenge (Digest/MD5)
    - It obviously works better if SIP channel is secured too
    - Registrar can implement backend the way it wants
  - Proxy-Proxy using TLS authentication (DNS)

SIP UA Authentication Example



What about Media?

- Securing SIP and SDP is only one step
  - What about the media transport?
- RTP, just as SIP, by default is sent in the clear
  - Securing SIP/SDP can make it harder to detect...
    - Negotiation parameters are encrypted
  - ... but wiretapping/eavesdropping is still possible!
    - Several tools available to make this really easy



Securing RTP

- Not as easy as securing SIP
  - Might use RTP/TLS, but...
    - RTP is almost always transported over UDP
    - TCP not suitable for its real-time requirements
  - Might use RTP/IPsec, but...
    - Assumes IPsec is available (e.g., in a VPN [4])
    - A lot of overhead involved

Secure Real-time Transport Protocol (SRTP)

http://tools.ietf.org/html/rfc3711

- Extends RTP to make it “secure”
- Authentication, integrity, protection against replay

[4] Virtual Private Network

Securing RTP

- Not as easy as securing SIP
  - Might use RTP/TLS, but...
    - RTP is almost always transported over UDP
    - TCP not suitable for its real-time requirements
  - Might use RTP/IPsec, but...
    - Assumes IPsec is available (e.g., in a VPN [4])
    - A lot of overhead involved

Secure Real-time Transport Protocol (SRTP)

http://tools.ietf.org/html/rfc3711

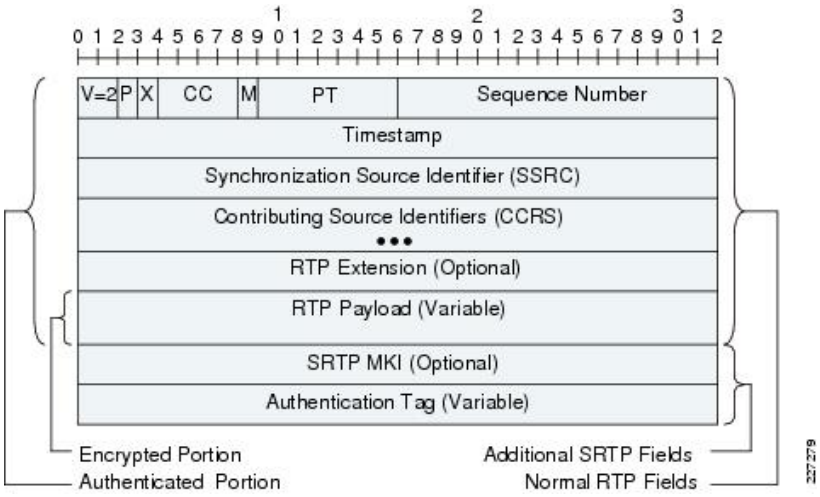
- Extends RTP to make it “secure”
- Authentication, integrity, protection against replay

[4] Virtual Private Network

Secure Real-time Transport Protocol (SRTP)

- Extension to standard RTP/RTCP
  - Encrypts payload, but **not** header
    - Advanced Encoding Scheme (AES)
    - Counter mode/f8, 128/192/256 bits
  - Authenticates the **whole** packet
    - HMAC-SHA1 32/80 (160)
    - Authentication + Message integrity
- Can have NULL cipher
  - Basically like RTP, but with hashing

Secure Real-time Transport Protocol (SRTP)



AES Encoding Scheme (AES)

Developed by J. Daemen and V. Rijmen

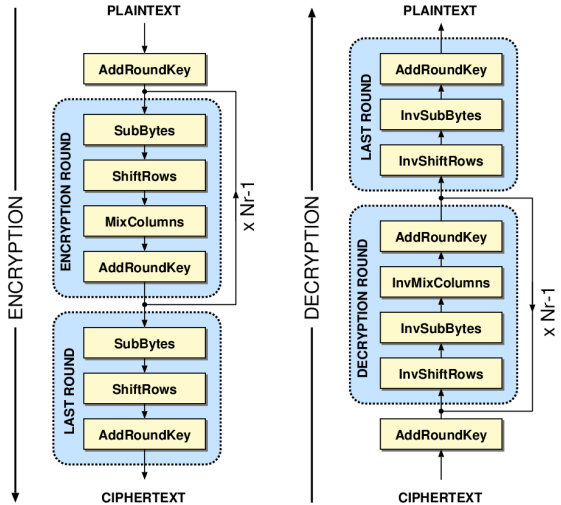
- Proposed in AES competition as Rijndael
  - Competition had specific requirements
    - Block length of 128 bits
    - Key lengths of 128, 192 and 256 bits
    - Easy to implement in hardware and software
    - Intended to replace Data Encryption Standard (DES)
- Standardized by National Institute of Standards and Technology (NIST) as FIPS 197 in 2001
  - Almost unbreakable
    - All attacks are just theoretical
  - Eventually replaced Data Encryption Scheme (DES)
    - U.S. Government Standard

AES Description

Block cipher that works iteratively

- Operates several rounds on “states” (4x4 bytes)
  - 10 rounds needed when key is 128 bits (192/12, 256/14)
- Four steps for each round (except last)
  - `SubBytes`
    - Each byte of the 4x4 state is replaced (lookup table)
  - `ShiftRows`
    - Each row of the state matrix is left rotated (different positions)
  - `MixColumns`
    - The data in each column is mixed up (combination)
  - `AddRound`
    - Each column of the state is XORed with key schedule

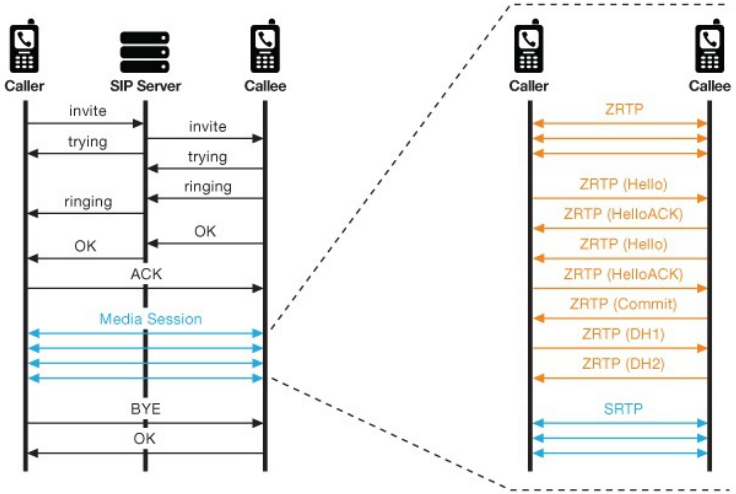
Encryption/Decryption Scheme



Key Exchange (1)

- SRTP specifies how to encrypt packets...
  - ... but not how to exchange keys
- Several alternatives
  - MIKEY (Multimedia Internet KEYing)
    - http://tools.ietf.org/html/rfc3830
    - Ticket-Based system
    - Too complex for VoIP? (but used by 3GPP)
  - ZRTP (Zimmermann RTP)
    - http://tools.ietf.org/html/rfc6189 (Informational!)
    - Focus on end-to-end (no need for PKI)
    - Diffie-Hellman on Media path (no need for encrypted signalling)

ZRTP (Zimmermann Secure RTP)



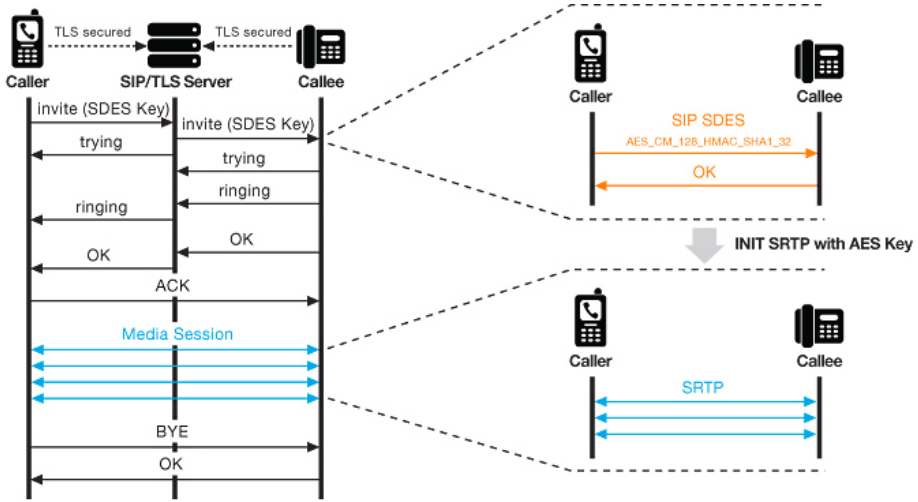
Key Exchange (2)

- SDES-SRTP (Secure Description)
  - http://tools.ietf.org/html/rfc4568
  - Simple, widespread
  - Exchanges keys in SDP (requires secure signalling)
- DTLS-SRTP (Datagram Transport Layer Security)
  - http://tools.ietf.org/html/rfc5763
  - Exploits DTLS, which is “like TLS” but for UDP
  - SDP transports certificate fingerprints (keys exchanged in DTLS)
- Hot topic in the IETF
  - RTPSEC BOF
    - http://www.ietf.org/proceedings/68/rtpsec.html
    - DTLS-SRTP was the “future”, is now the “present”

Secure Description (SDES)

- Simple mechanism to negotiate parameters
  - Negotiation in SDP as additional a-line
    `a=crypto:<tag> <crypto-suite> inline:<key||salt> [session-parms]`
- Master key and salt provided inline
  - Concatenated and base64 encoded
- Supported crypto-suites (AES/SHA1 variations)
  - `AES_CM_128_HMAC_SHA1_80`
  - `AES_CM_128_HMAC_SHA1_32`
  - `F8_128_HMAC_SHA1_32`

Secure Description (SDES)



Example of SDP Negotiation (SDES)

- Just plain RTP (no SDES)...

```
m=audio 13916 RTP/AVP 0 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
```

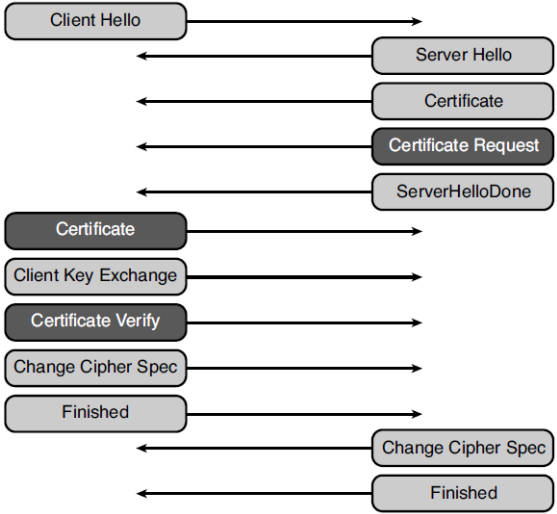
- ... and with SRTP (SDES)

```
m=audio 16284 RTP/SAVP 0 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:NErLjk8AYFyeTmtP39k80lygmPP+ZWQv8bUn8Uv+
```

Datagram Transport Layer Security (DTLS)

- Recent standard, favourite in the IETF
  - But not very deployed at the moment...
- SRTP keys exchanged over DTLS
  - Ad-hoc extensions for SRTP keys
  - Media still SRTP! (DTLS **not** used as a transport)
- SDP does not contain keys
  - Cryptographic handshake over voice channel
    - Remember ZRTP?
  - Handshake authenticated via certificate fingerprint
    - Fingerprint is what is exchanged via SDP
    - Some additional parameters related to DTLS and roles

Datagram Transport Layer Security (DTLS)



Example of SDP Negotiation (DTLS-SRTP)

- Offerer (is going to **expect** ClientHello)...

```
a=setup:actpass
a=fingerprint:  SHA-1
4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
m=audio 6056 RTP/AVP 0
a=sendrecv
a=tcap:1 UDP/TLS/RTP/SAVP RTP/AVP
a=pcfg:1 t=1
```

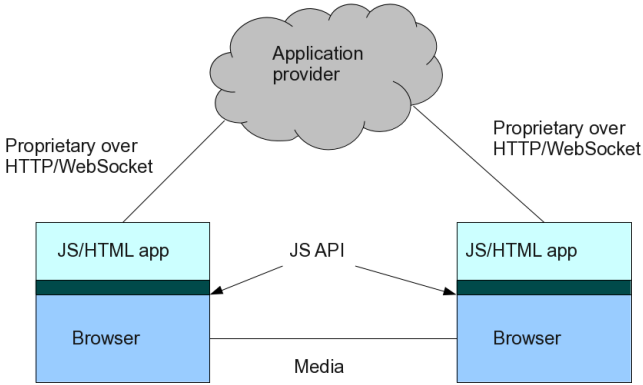
- Answerer (is going to **send** ClientHello)

```
a=setup:active
a=fingerprint:  SHA-1
07:0B:0E:E8:F7:22:59:72:6A:1C:68:05:05:CF:2E:6F:59:43:48:99
m=audio 12000 UDP/TLS/RTP/SAVP 0
a=acfg:1 t=1
```

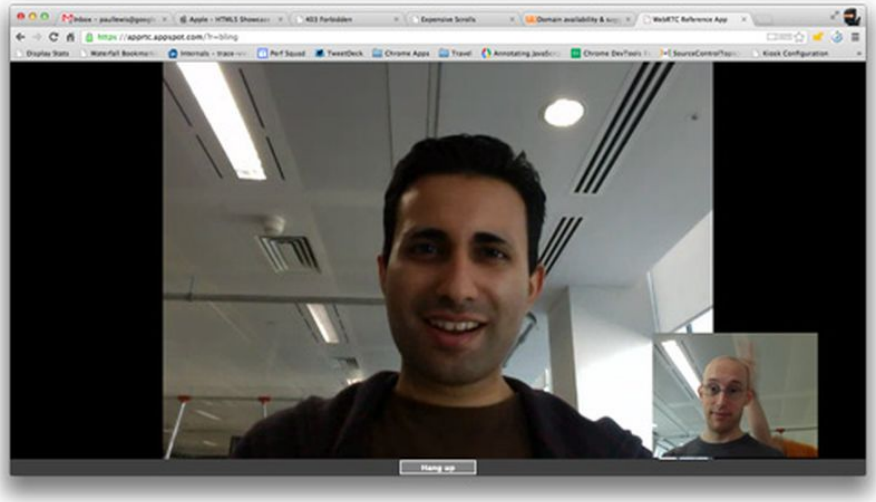
WebRTC/RTCWEB and Secure VoIP

- Standard effort to build real-time communications integrated in browsers
  - Re-uses pre-existing standards
    - SDP, SRTP, ICE, ... (but not SIP)
  - Strong emphasis on security and privacy
    - http://tools.ietf.org/html/draft-ietf-rtcweb-security
    - http://tools.ietf.org/html/draft-ietf-rtcweb-security-arch
  - Media security is **mandatory**
    - http://tools.ietf.org/html/draft-ietf-rtcweb-rtp-usage
    - **MUST** implement DTLS-SRTP
    - SDES-SRTP **MUST NOT** be supported

The WebRTC “trapezoid” (hey, it looks like SIP!)



Let's try a secure live call with WebRTC!



Let's try a secure live call with WebRTC!

Demo 1:

- WebRTC call
  - Users register a username
  - Users can call each other
- Open this link!
  - https://srv128.conf.meetecho.com/demo-ns

Ok, let's try again... different link!

Demo 2:

- WebRTC call (peer-to-peer, this time)
  - Users register a username
  - Users can call each other
- Open this link!
  - https://srv128.conf.meetecho.com:9101

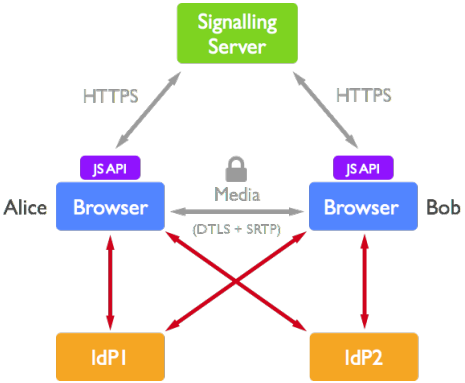
One last time..!

Demo 3:

- WebRTC call (peer-to-peer, and no monitor!)
  - Users register a username
  - Users can call each other
- Open this link!
  - https://srv128.conf.meetecho.com:9201

What's missing? Identity Providers!

- Two main issues
  - Can I trust that website?
  - Can I trust a user/verify the fingerprint?



Private Media Requirements in Privacy Enhanced RTP Conferencing (PERC)

- Media security works “great” for peer-to-peer
  - But what if we want to do a media conference?
- Several approaches to conferencing
  - Full-mesh (everybody connects to everybody)
  - Multi-point Control Unit (MCU) → **server!**
  - Selective Forwarding Unit (SFU) → **server!**

Private Media Requirements in Privacy Enhanced RTP Conferencing (PERC)

https://datatracker.ietf.org/wg/perc/charter/

- Ensure end-to-end confidentiality/authentication
- Trusted elements on the media path

Private Media Requirements in Privacy Enhanced RTP Conferencing (PERC)

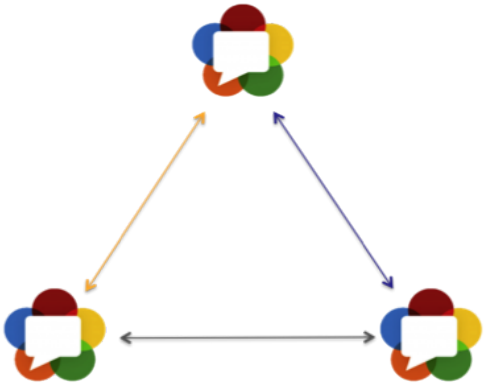
- Media security works “great” for peer-to-peer
  - But what if we want to do a media conference?
- Several approaches to conferencing
  - Full-mesh (everybody connects to everybody)
  - Multi-point Control Unit (MCU) → **server!**
  - Selective Forwarding Unit (SFU) → **server!**

Private Media Requirements in Privacy Enhanced RTP Conferencing (PERC)

https://datatracker.ietf.org/wg/perc/charter/

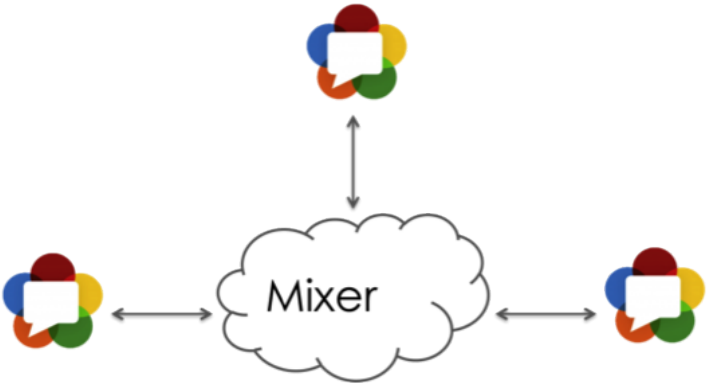
- Ensure end-to-end confidentiality/authentication
- Trusted elements on the media path

Full-mesh



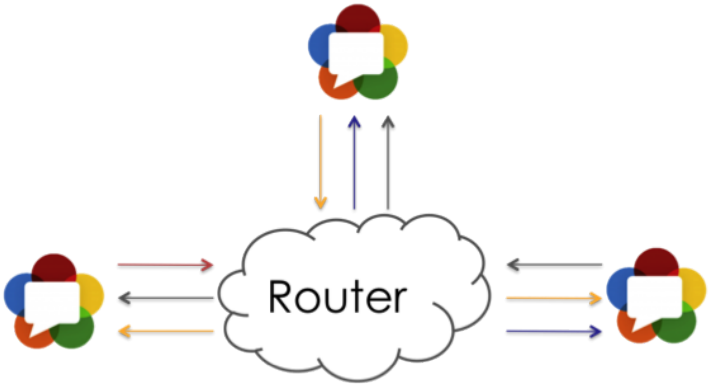
https://webrtchacks.com/webrtc-beyond-one-one/

Multi-point Control Unit (MCU)



https://webrtchacks.com/webrtc-beyond-one-one/

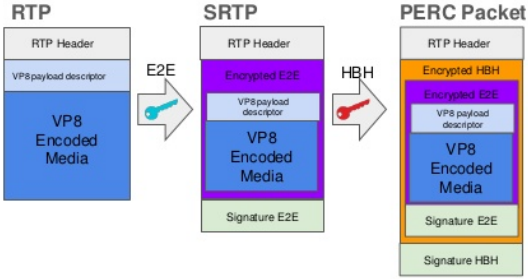
Selective Forwarding Unit (SFU)



https://webrtchacks.com/webrtc-beyond-one-one/

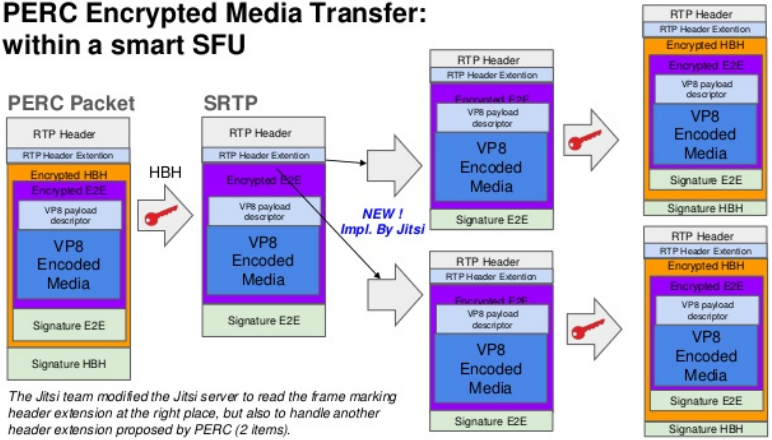
Double Encryption with PERC

Double-Encrypted Media Transfer: PERC (ex: WebRTC)



https://www.slideshare.net/alexpiwi5/perc-webrtc-e2e-media-encryption-with-sfu

Double Encryption with PERC



The Jitsi team modified the Jitsi server to read the frame marking header extension at the right place, but also to handle another header extension proposed by PERC (2 items).

https://www.slideshare.net/alexpiwi5/perc-webrtc-e2e-media-encryption-with-sfu

Questions? Comments?