Investigating RTC Security in Consumer IoT

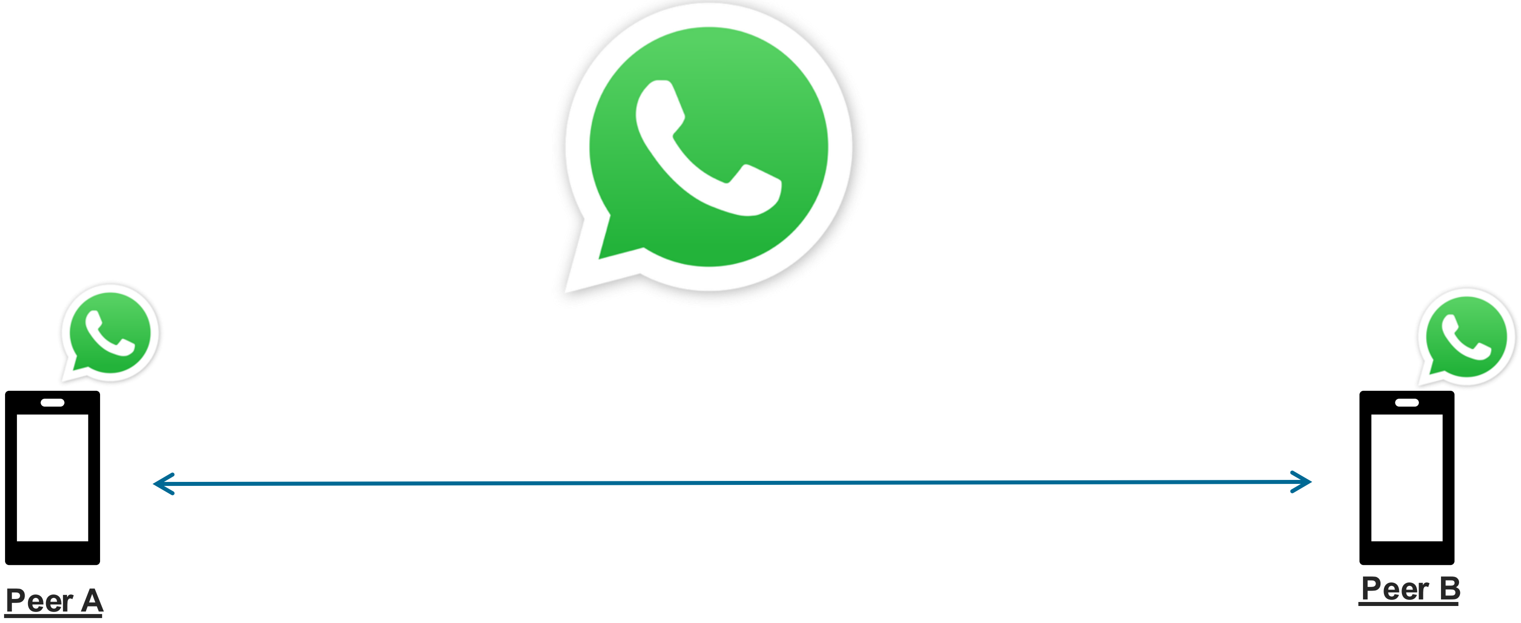
Victor Goeman

DistriNet

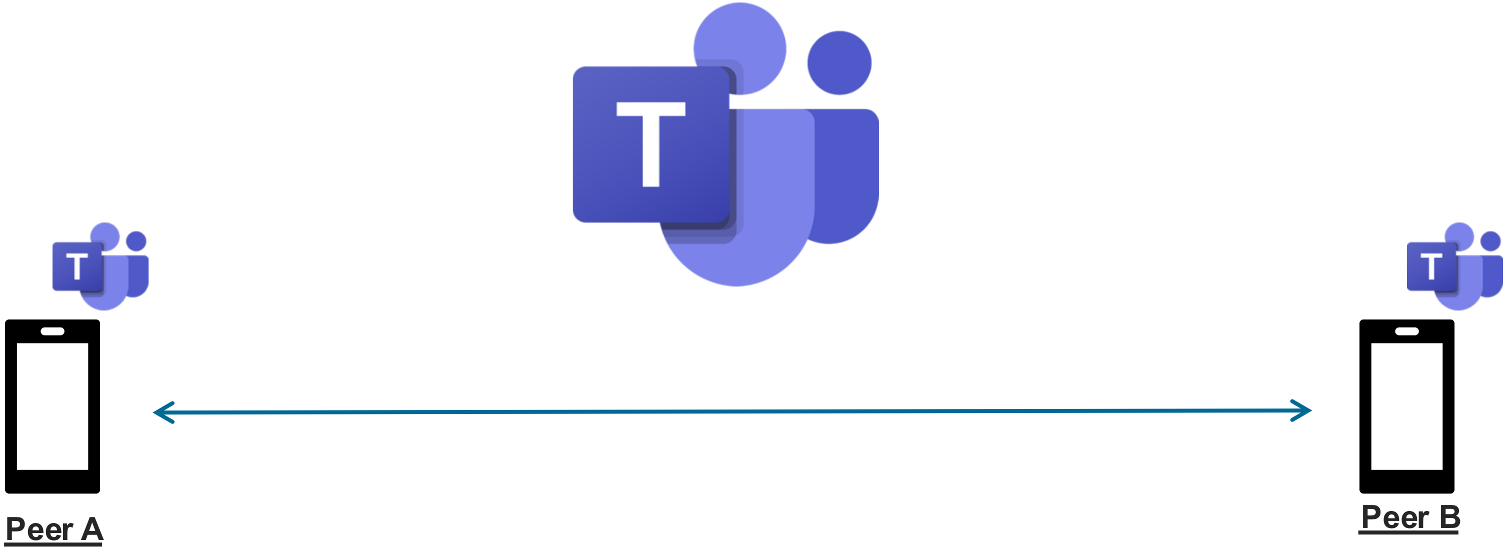
Real-Time Communication

- What is real-time communication?
- Where is real-time communication used?

Real-Time Communication



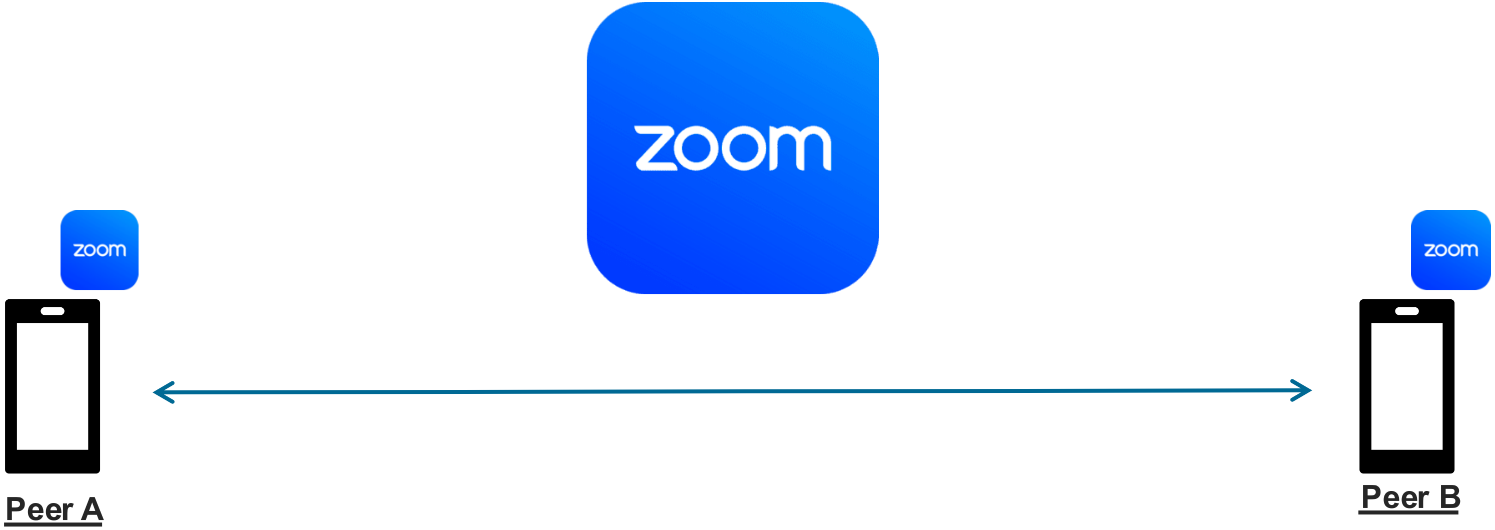
Real-Time Communication



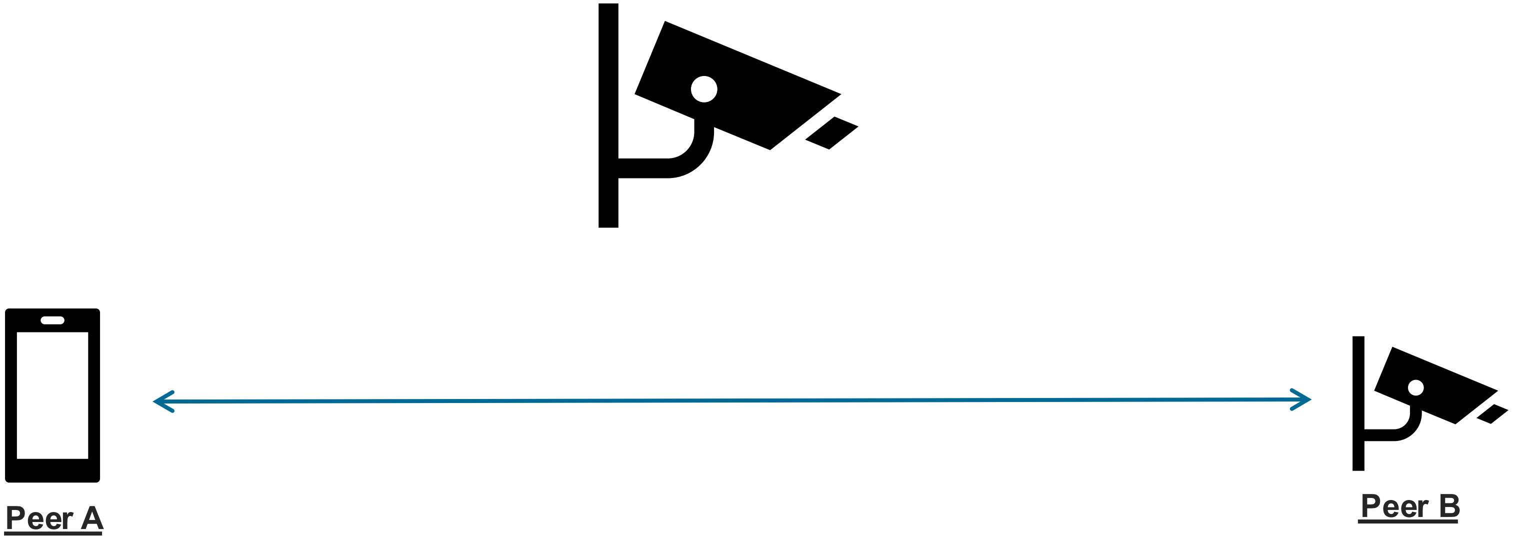
Peer A

Peer B

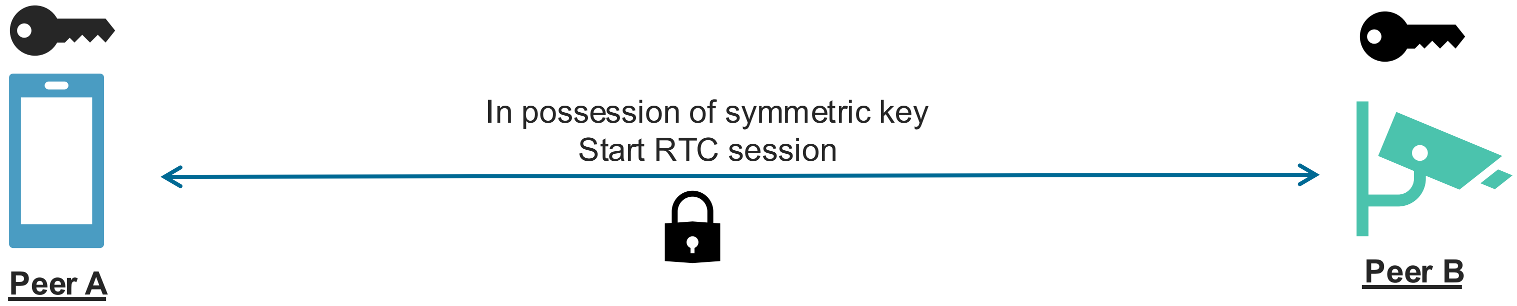
Real-Time Communication



Real-Time Communication



Before an RTC Connection



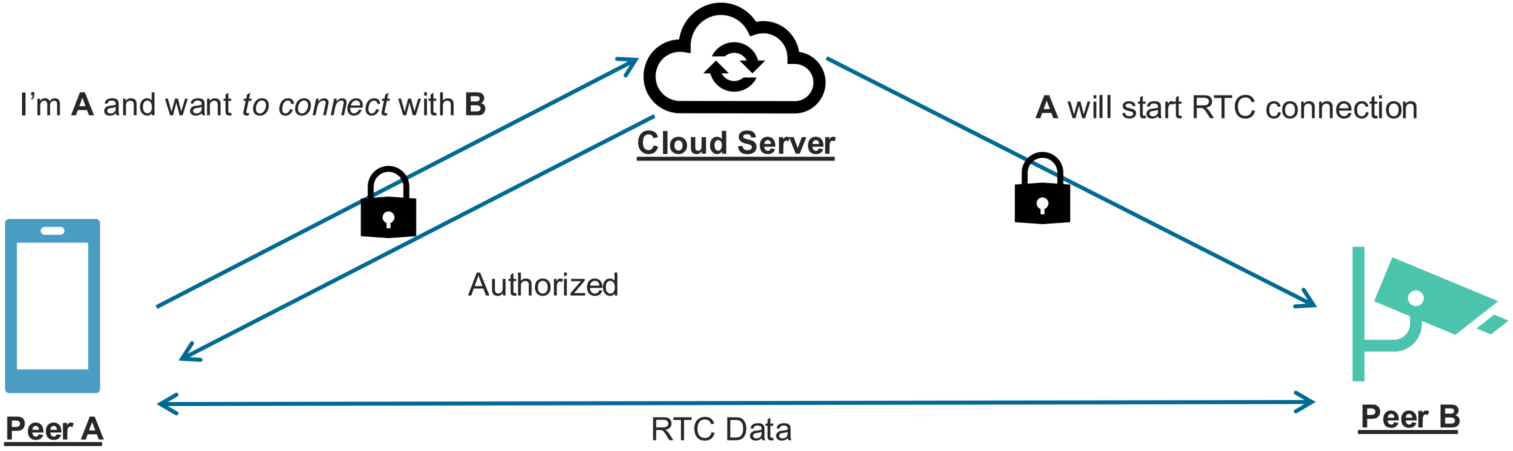
In possession of symmetric key
Start RTC session

Peer A

Peer B

Before an RTC Connection

Peer 1	Peer 2	Authorized
Peer A	Peer B	Yes
Peer A	Peer C	No



I'm **A** and want *to connect* with **B**

Cloud Server

A will start RTC connection

Authorized

Peer A

RTC Data

Peer B

Problem Statement

- What is the architecture of an RTC session?
- How is the security of RTC implementations?
  - What about IoT?

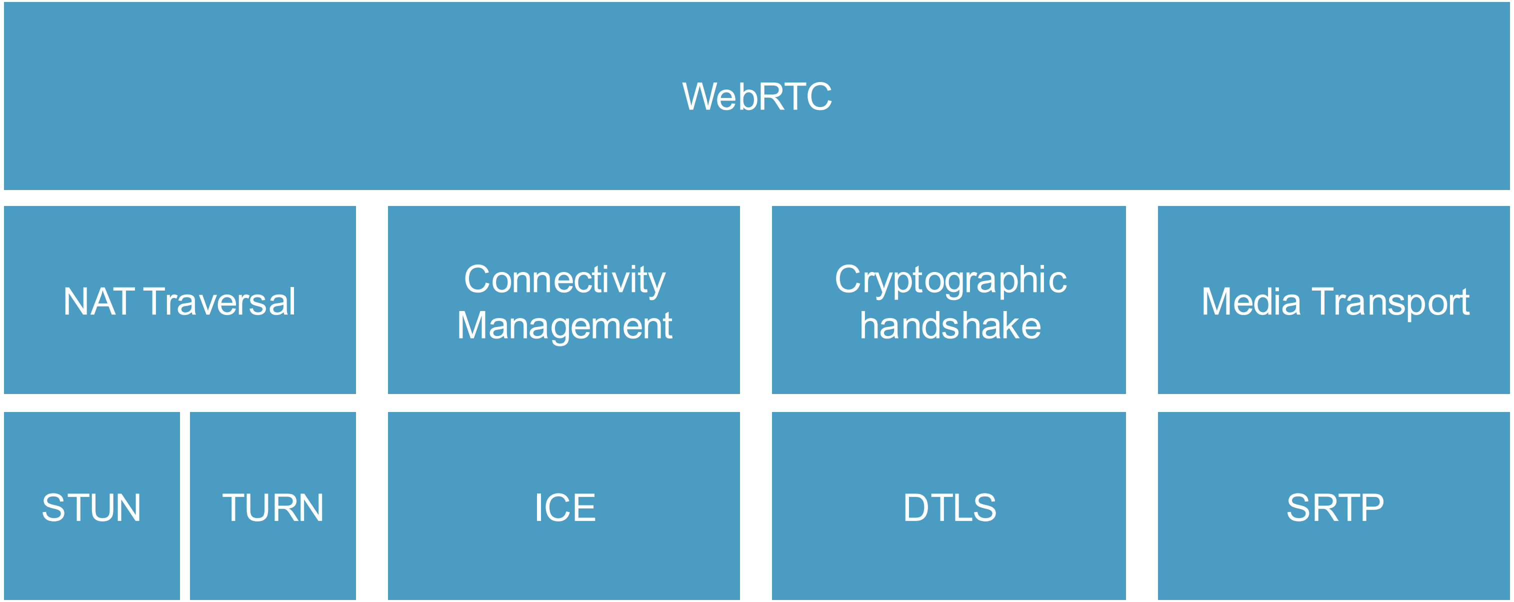
WebRTC

WebRTC

- Collection of standardized APIs and protocols
- Enables real-time peer-to-peer communication
- Provides efficiency and security out-of-the-box

- Built for web

WebRTC



WebRTC

NAT Traversal

Connectivity Management

Cryptographic handshake

Media Transport

STUN

TURN

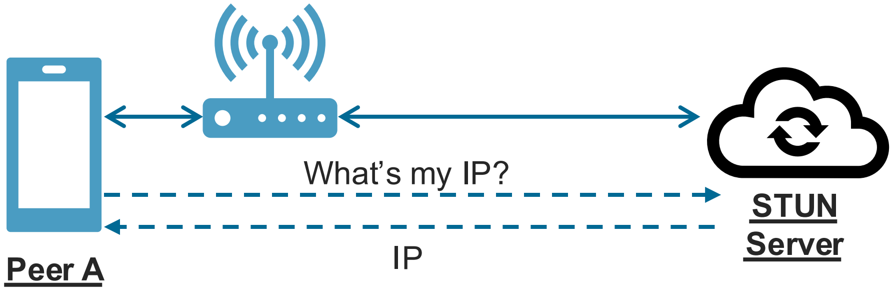
ICE

DTLS

SRTP

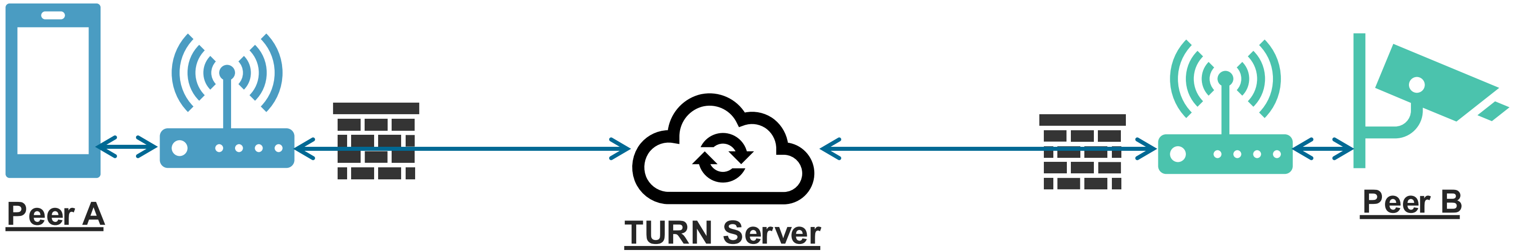
STUN

- Peer behind **NATs** discovers their public IP address and port



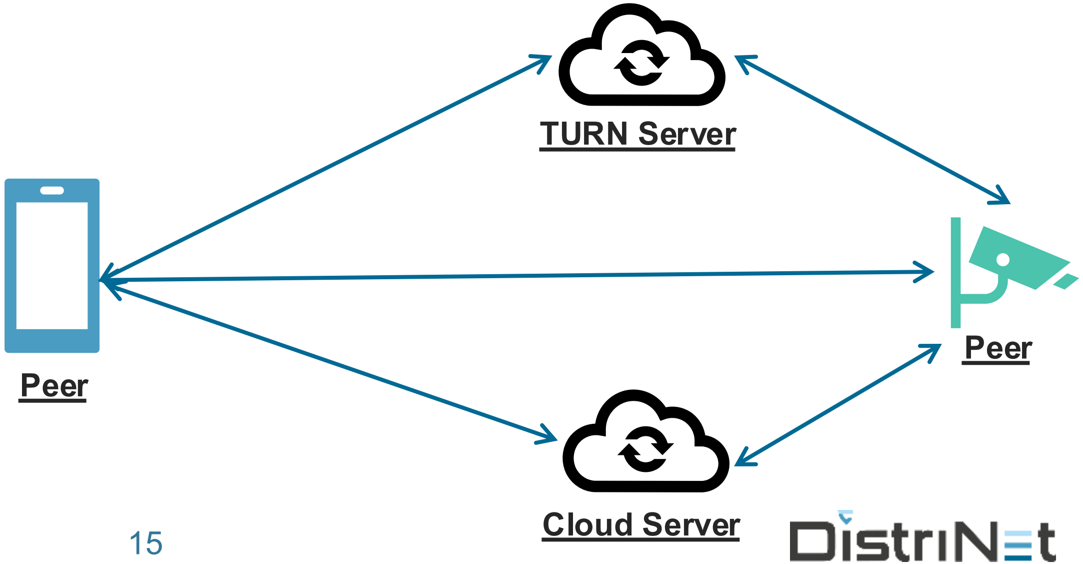
TURN

- Enables peers behind NATs or firewalls to participate in peer-to-peer communication



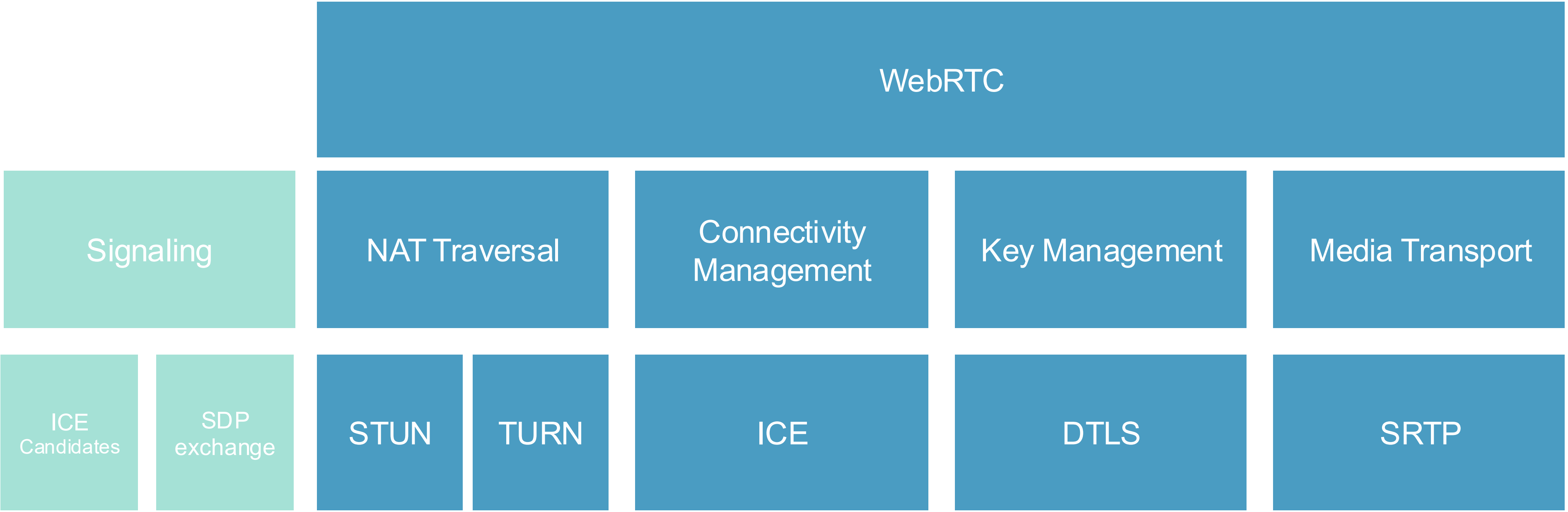
ICE

- Finding the best path between peers
  - Gathering paths
  - Testing paths



Signaling

Signaling



WebRTC

Signaling

NAT Traversal

Connectivity Management

Key Management

Media Transport

ICE Candidates

SDP exchange

STUN

TURN

ICE

DTLS

SRTP

Signaling

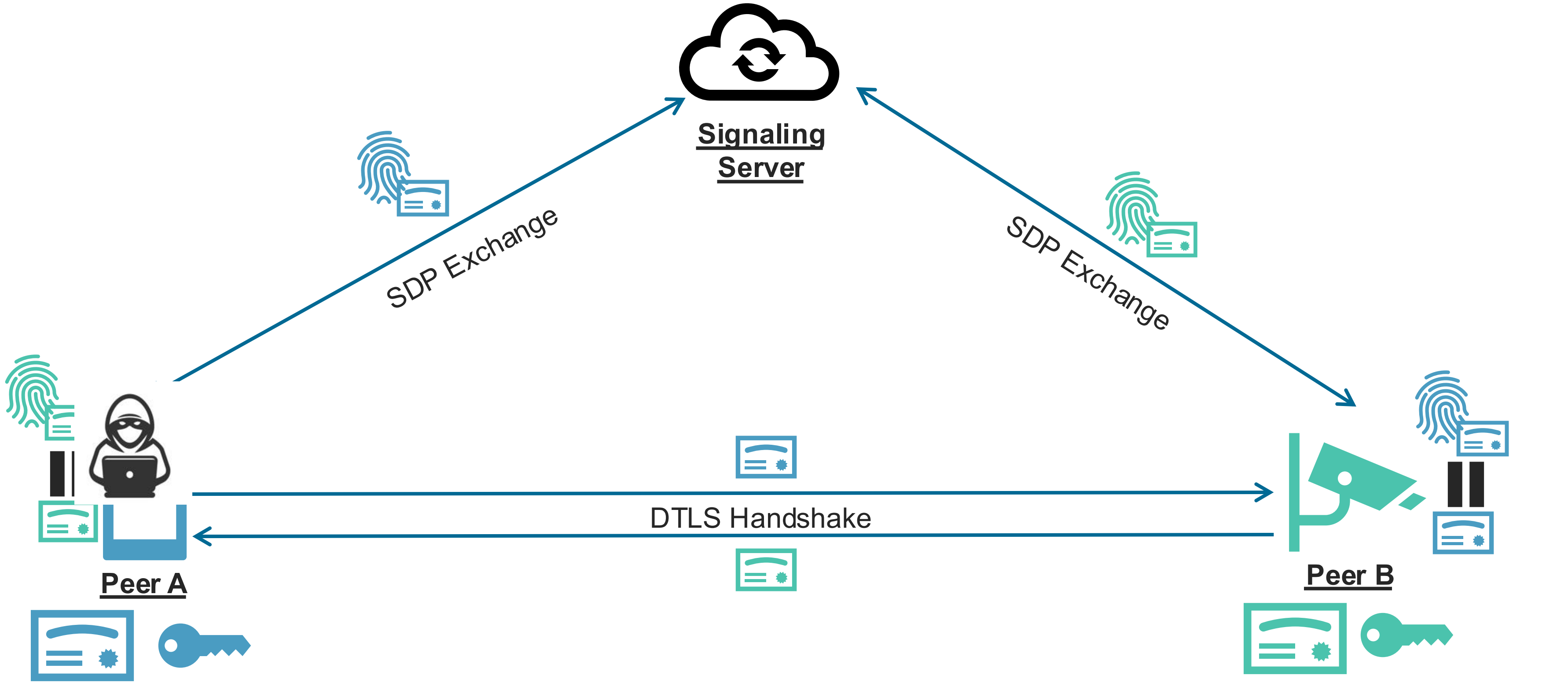
- Exchanging messages between peers to establish and manage connections

- WebRTC
  - Signaling is an application-layer responsibility
  - Defines what should happen, not how

Signaling

- SDP exchange
  - **DTLS** fingerprints
  - Session metadata
  - Optional:
    - Authentication data
    - Application-specific data

Signaling in WebRTC



Security in WebRTC

Security in WebRTC

Identity and Trust Establishment

- Secure cloud connection
- Verify identity
- Receive peer address

Signaling

- Exchange certificate fingerprint

Connectivity Establishment

- Establish transport path

Secure Channel Setup

- Verify certificates
- Establish encryption keys

Security in WebRTC

Identity and Trust Establishment

- Secure cloud connection
- Verify identity
- Receive peer address

Signaling

- **Exchange certificate fingerprint**

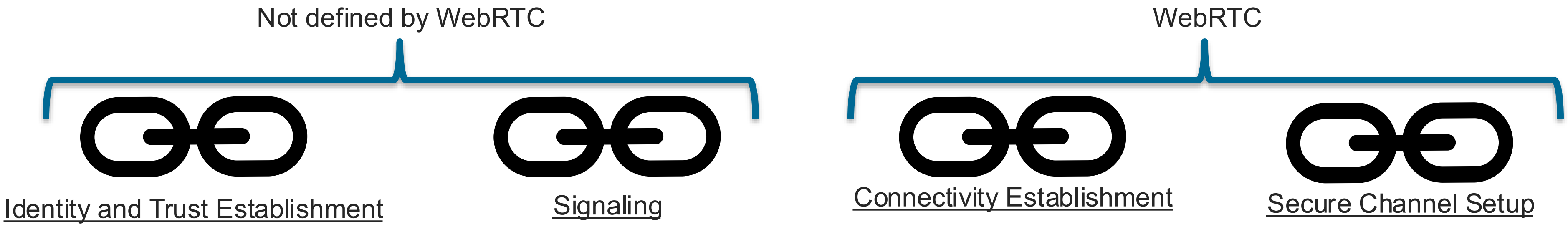
Connectivity Establishment

- Establish transport path

Secure Channel Setup

- **Verify certificates**
- Establish encryption keys

Security in WebRTC



Not defined by WebRTC

WebRTC

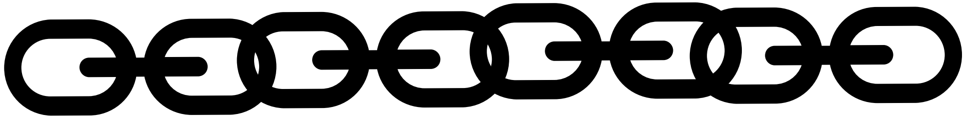
Identity and Trust Establishment

Signaling

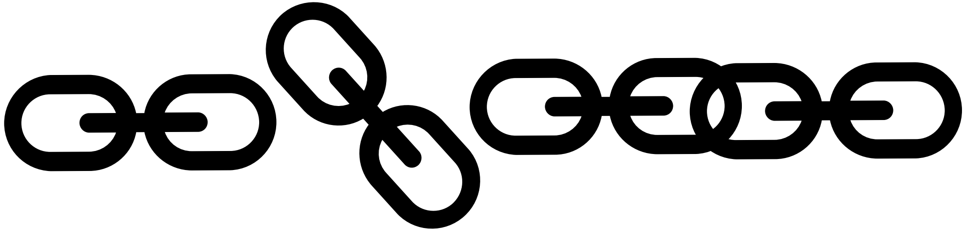
Connectivity Establishment

Secure Channel Setup

Security in WebRTC



Security in WebRTC



Tool

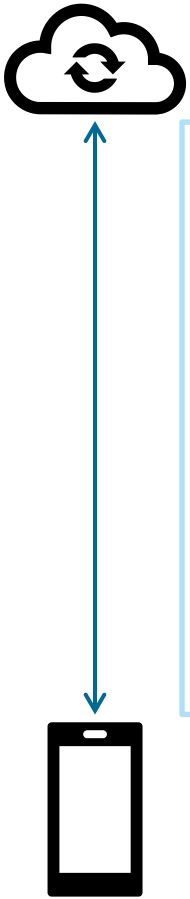
RTCInspect

- Open-source tool
- Automates security analysis of RTC traffic

- Input: pcaps
- Output: network-observable security issues in RTC

Signaling Issues

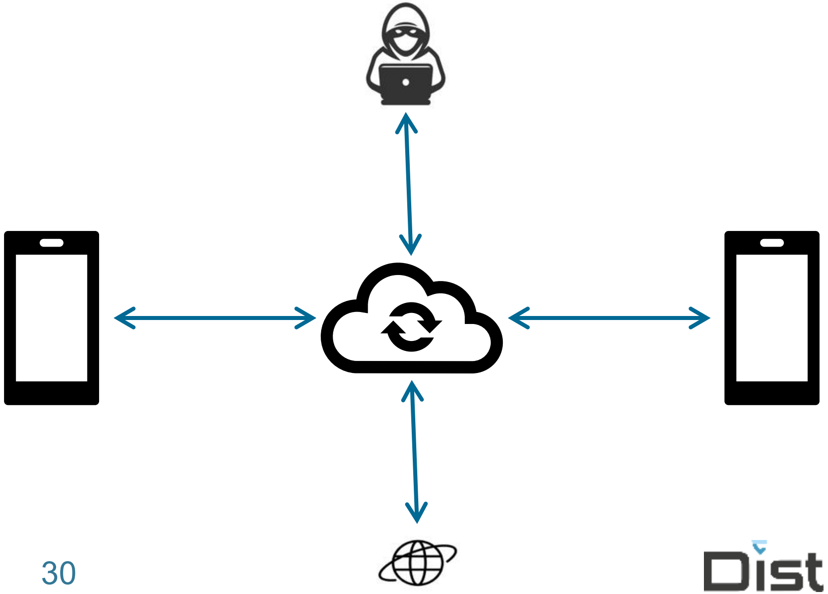
- No integrity protection
  - MITM
- No encryption



```
{"sdp":"
no=- 1857778606972110432 IN IP4 127.0.0.1
c=IN IP4 0.0.0.0\
a=rtcp:9 IN IP4 0.0.0.0
a=ice-ufrag:GNXr
a=ice-pwd:cb8f6tV+CE86+nWaXGU1u2gZ
a=ice-options:trickle
a=fingerprint:sha-256
22:E9:C3:45:A5:5E:39:73:41:82:78:98:6E:D5:8A:E8:37:C3:D
9:F8:64:09:88:6E:3F:F8:16:4F:52:C6:55:4C
a=setup:active
a=turn-password:securerelaypassword
…..
"}
{"ice_candidates":
1 1 udp 2122317823 192.168.100.133 46587 typ host
generation 0
na=candidate:2 1 udp 2122317567 192.168.32.2 46587 typ host
generation 0\r\na=candidate:3 1 udp 1686109695 87.67.66.220
46587 typ srflx generation 0
na=candidate:4 1 udp 8387839 3.123.109.172 50229 typ relay
generation 0\r\nm=video 1
}
```

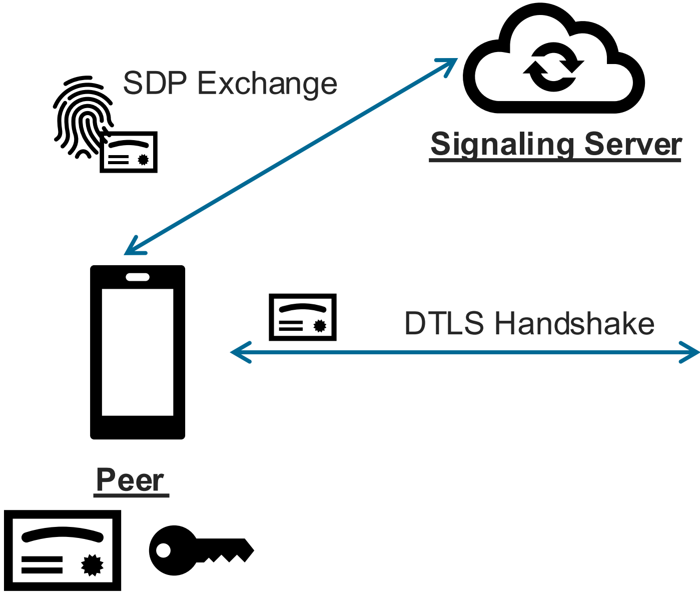
Connectivity Establishment Issues

- Leaked credentials
- Leaked ICE paths
- Outdated libraries



Secure Channel Setup Issues

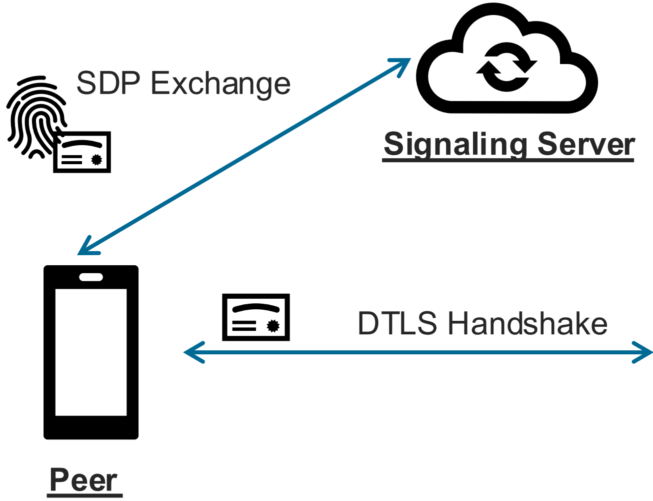
- Insecure cryptographic parameters
- Reused certificate



Secure Channel Setup Issues

- Insecure cryptographic parameters
- Reused certificate

Ephemeral	Hardcoded
Per session	Per Vendor
Not impactful	Impactful

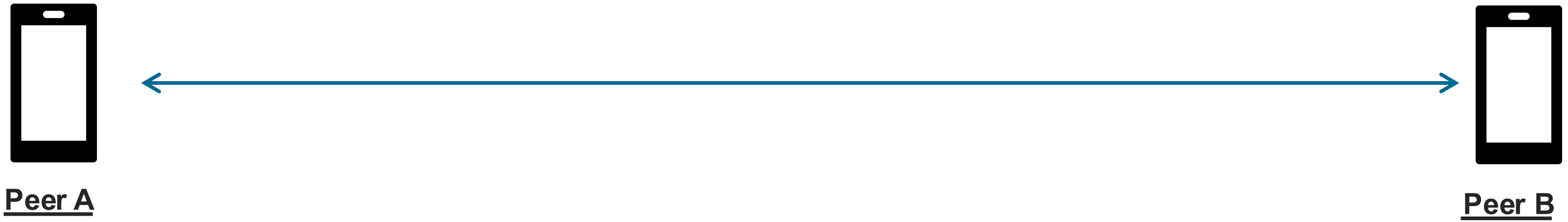


RTCInspect

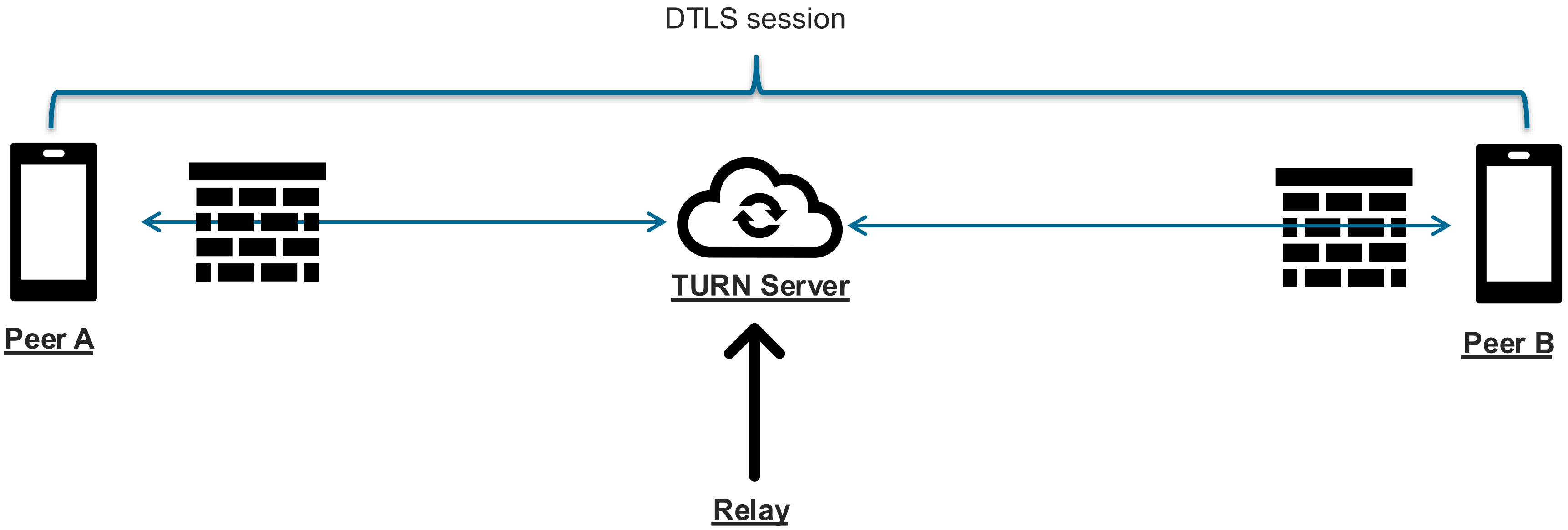
- Case studies – 11 IoT RTC devices tested
  - 6 proprietary RTC implementations
  - 6 lack integrity protected signaling
  - 6 lack encrypted signaling
  - 7 lack E2E encrypted media
  - 2 reuse certificates across devices
  - 2 lack strong encryption ciphers

Web

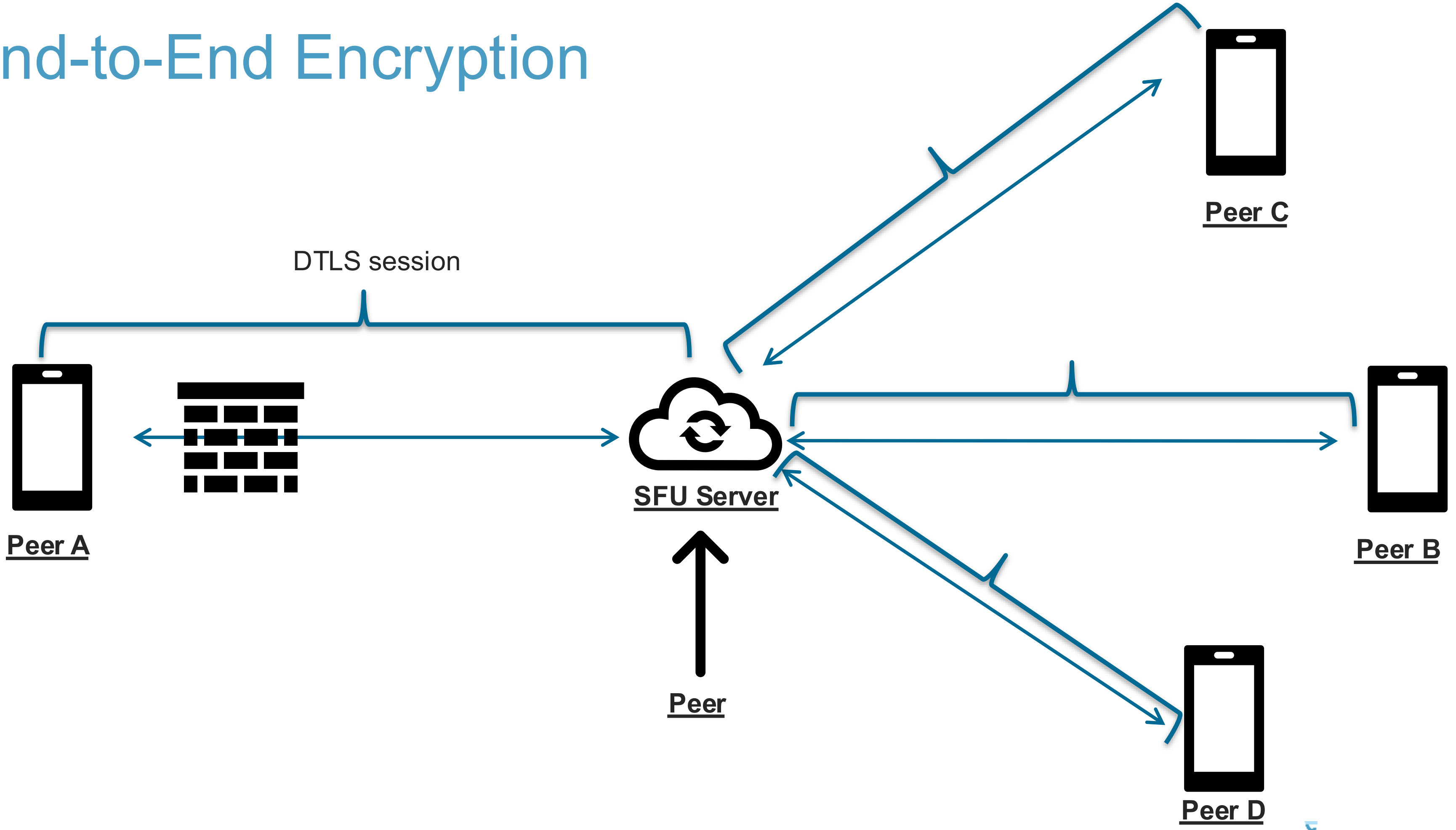
End-to-End Encryption



End-to-End Encryption



End-to-End Encryption

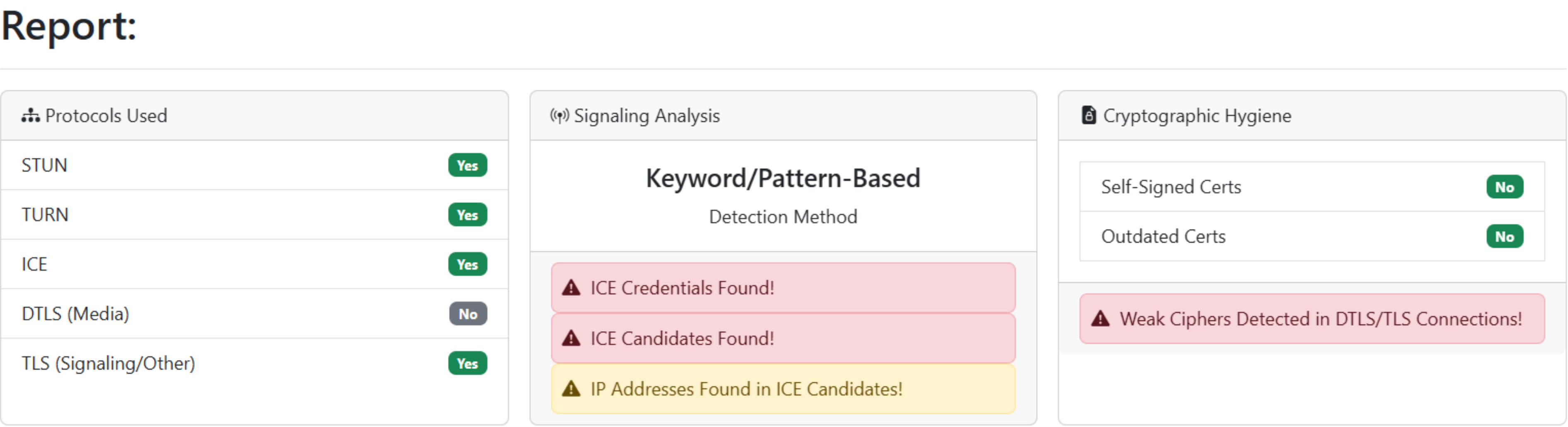


IoT

http://127.0.0.1:8080/report/Foscam/4G-test

Use Cases

Foscam



Report:

Protocols Used

STUN	Yes
TURN	Yes
ICE	Yes
DTLS (Media)	No
TLS (Signaling/Other)	Yes

Signaling Analysis

Keyword/Pattern-Based

Detection Method

⚠ ICE Credentials Found!

⚠ ICE Candidates Found!

⚠ IP Addresses Found in ICE Candidates!

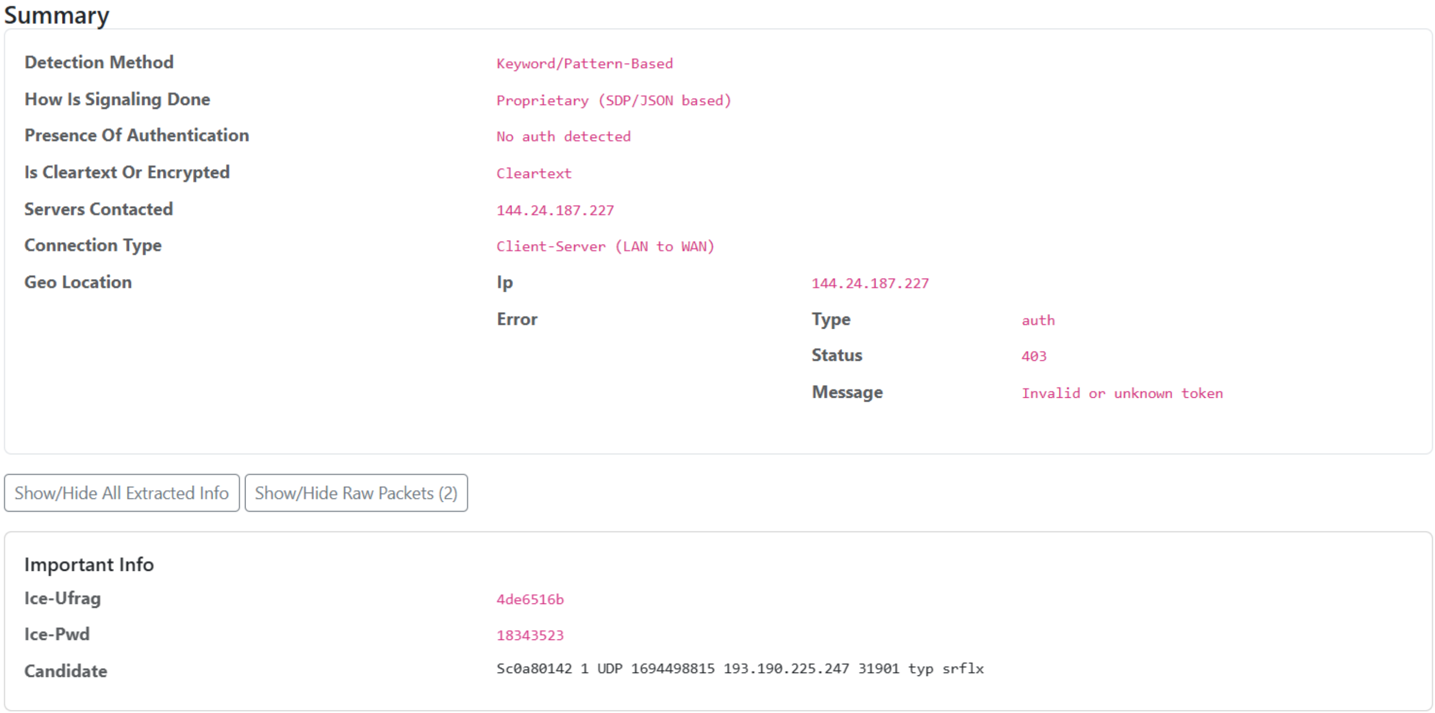
Cryptographic Hygiene

Self-Signed Certs	No
Outdated Certs	No

⚠ Weak Ciphers Detected in DTLS/TLS Connections!

Use Cases

Foscam



Summary

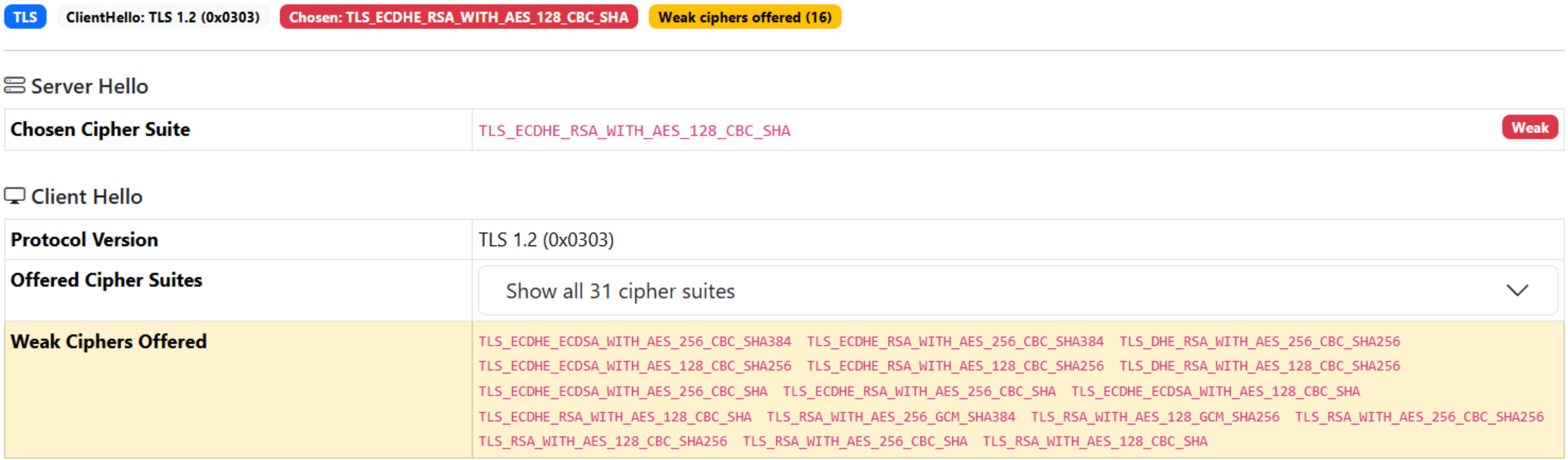
Detection Method	Keyword/Pattern-Based		
How Is Signaling Done	Proprietary (SDP/JSON based)		
Presence Of Authentication	No auth detected		
Is Cleartext Or Encrypted	Cleartext		
Servers Contacted	144.24.187.227		
Connection Type	Client-Server (LAN to WAN)		
Geo Location	Ip	144.24.187.227	
	Error	Type	auth
		Status	403
		Message	Invalid or unknown token

Important Info

Ice-Ufrag	4de6516b
Ice-Pwd	18343523
Candidate	Sc0a80142 1 UDP 1694498815 193.190.225.247 31901 typ srflx

Use Cases

Foscam



TLS   ClientHello: TLS 1.2 (0x0303)   Chosen: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA   Weak ciphers offered (16)

Server Hello

Chosen Cipher Suite	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA Weak

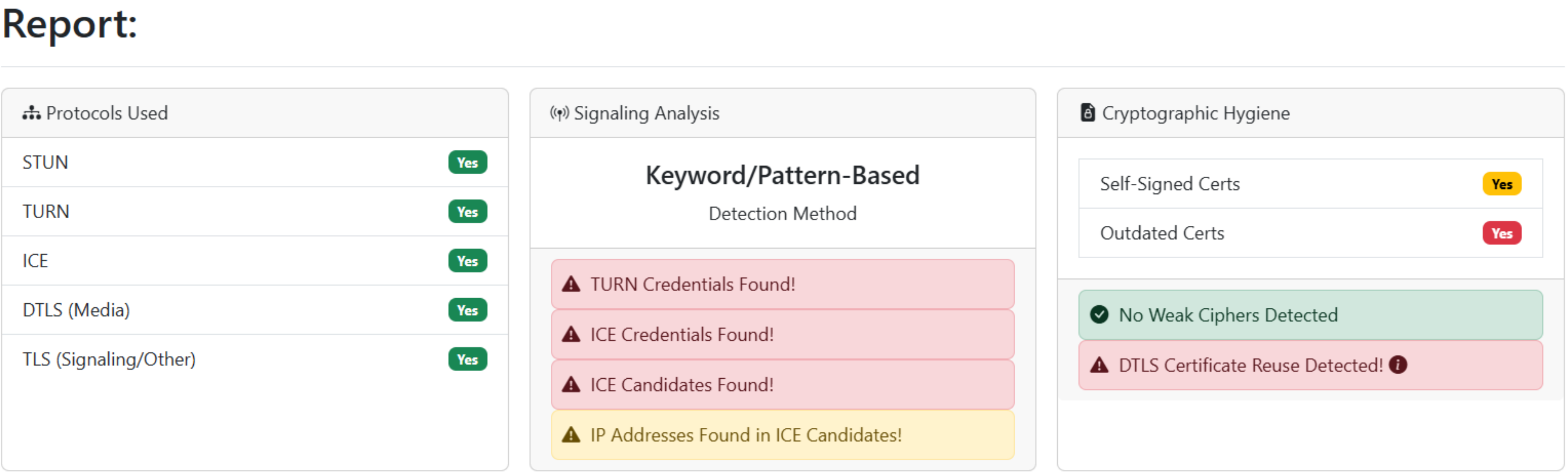
Client Hello

Protocol Version	TLS 1.2 (0x0303)
Offered Cipher Suites	Show all 31 cipher suites
Weak Ciphers Offered	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_GCM_SHA384 TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA256 TLS_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA

http://127.0.0.1:8080/report/Eufy/Homebase2/Homebase_WEB-External

Use Cases

Eufy



Report:

Protocols Used

STUN	Yes
TURN	Yes
ICE	Yes
DTLS (Media)	Yes
TLS (Signaling/Other)	Yes

Signaling Analysis

Keyword/Pattern-Based

Detection Method

⚠ TURN Credentials Found!

⚠ ICE Credentials Found!

⚠ ICE Candidates Found!

⚠ IP Addresses Found in ICE Candidates!

Cryptographic Hygiene

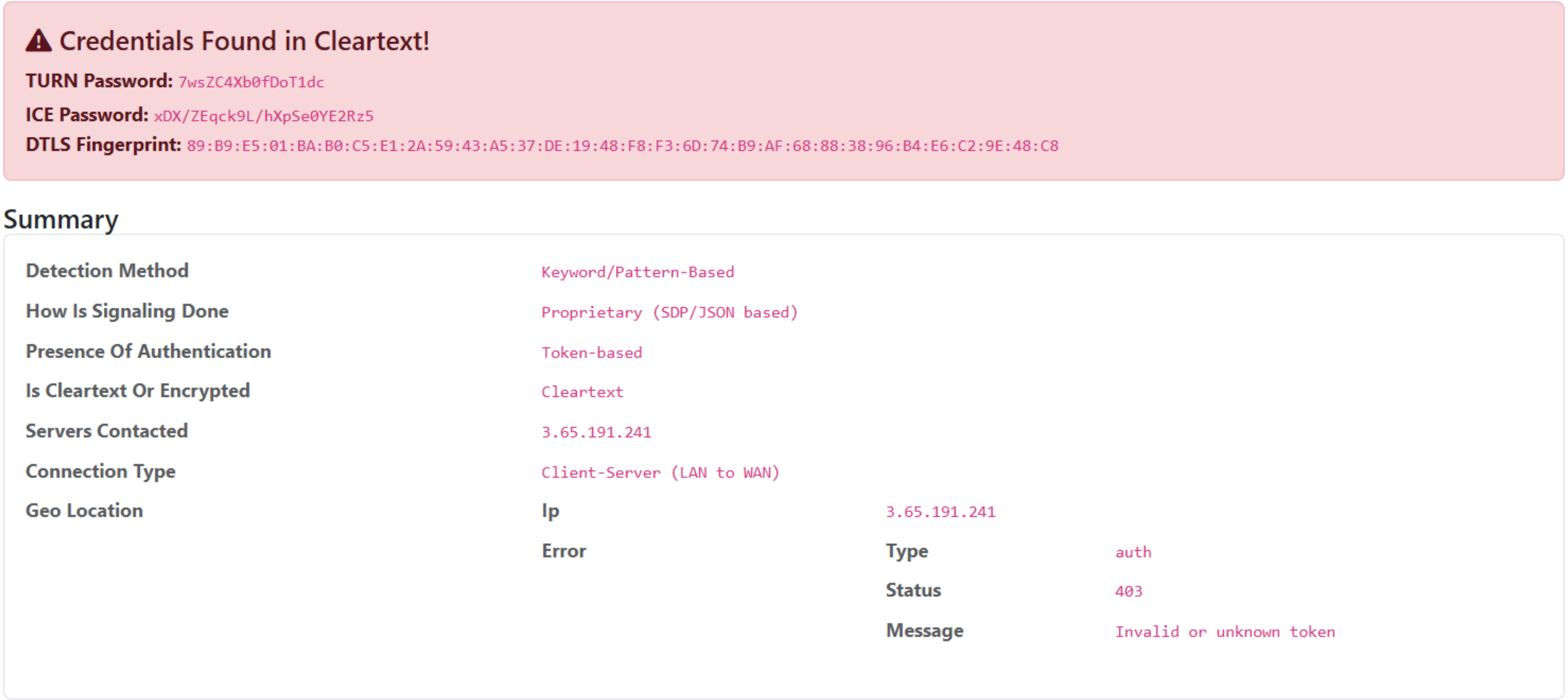
Self-Signed Certs	Yes
Outdated Certs	Yes

✓ No Weak Ciphers Detected

⚠ DTLS Certificate Reuse Detected!

Use Cases

Eufy



⚠ Credentials Found in Cleartext!

TURN Password: 7wsZC4Xb0fDoT1dc

ICE Password: xDX/ZEqck9L/hXpSe0YE2Rz5

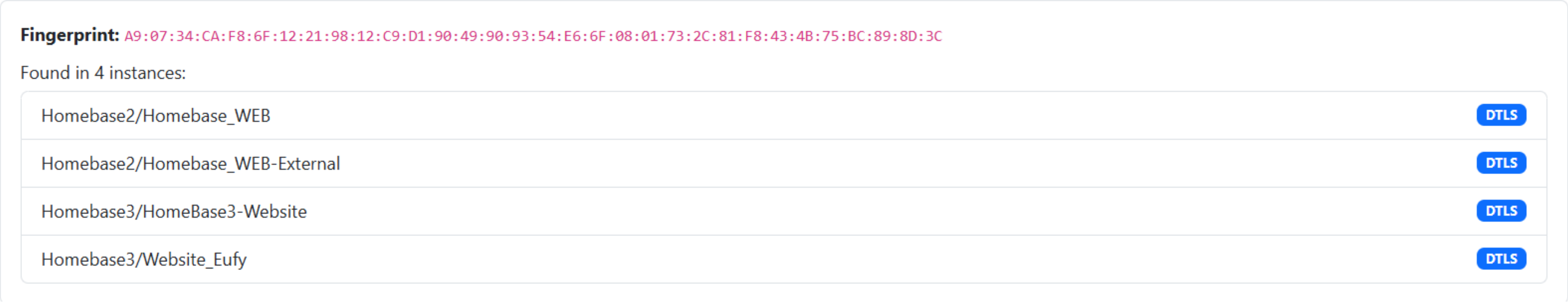
DTLS Fingerprint: 89:B9:E5:01:BA:B0:C5:E1:2A:59:43:A5:37:DE:19:48:F8:F3:6D:74:B9:AF:68:88:38:96:B4:E6:C2:9E:48:C8

Summary

Detection Method	Keyword/Pattern-Based		
How Is Signaling Done	Proprietary (SDP/JSON based)		
Presence Of Authentication	Token-based		
Is Cleartext Or Encrypted	Cleartext		
Servers Contacted	3.65.191.241		
Connection Type	Client-Server (LAN to WAN)		
Geo Location	**Ip**	3.65.191.241	
	Error	**Type**	auth
		Status	403
		Message	Invalid or unknown token

Use Cases

Eufy



Fingerprint: `A9:07:34:CA:F8:6F:12:21:98:12:C9:D1:90:49:90:93:54:E6:6F:08:01:73:2C:81:F8:43:4B:75:BC:89:8D:3C`

Found in 4 instances:

Homebase2/Homebase_WEB	DTLS
Homebase2/Homebase_WEB-External	DTLS
Homebase3/HomeBase3-Website	DTLS
Homebase3/Website_Eufy	DTLS

Conclusion

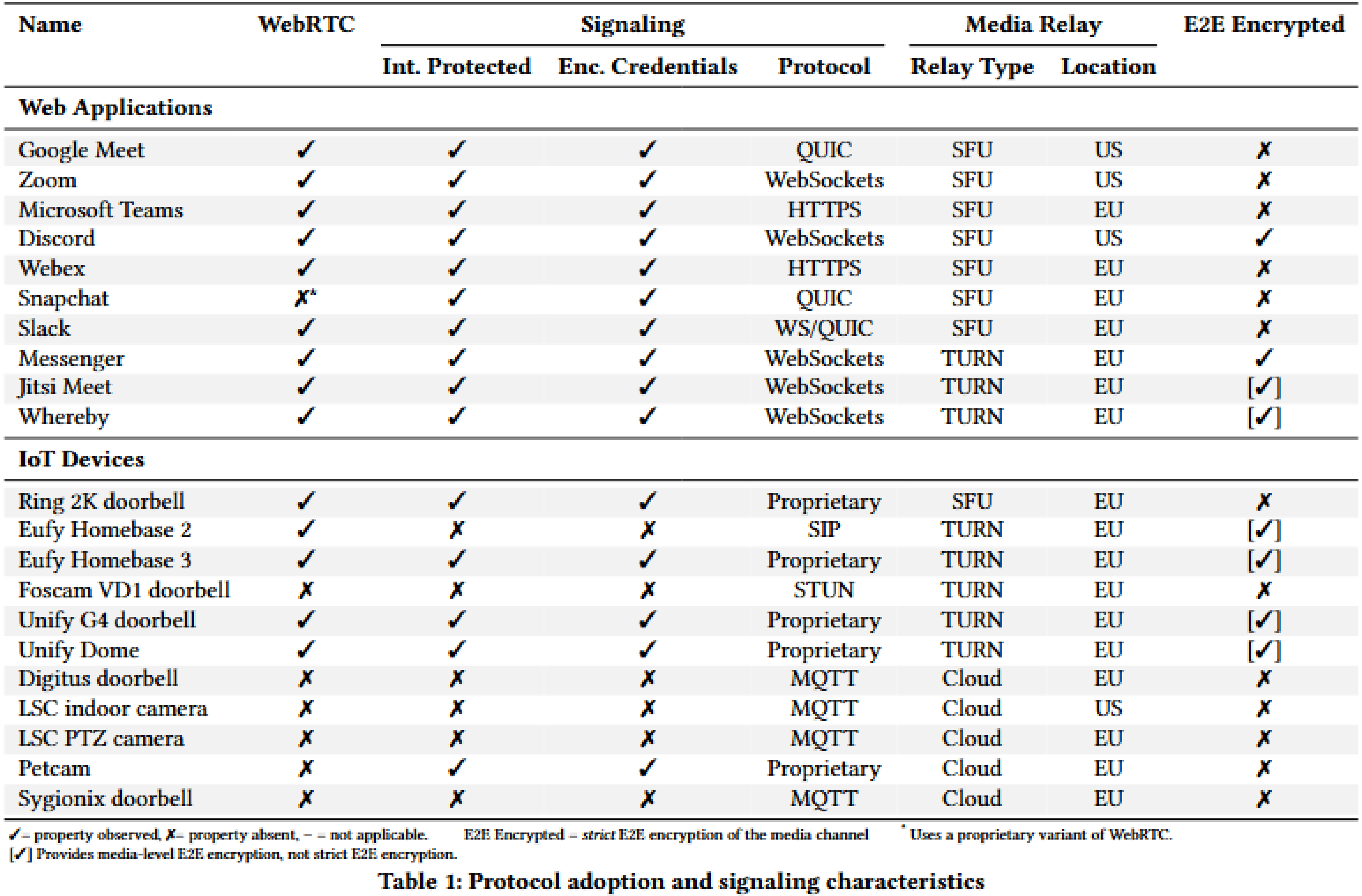
- RTC in Web
  - Standardized
  - Secure
  - E2E issues

- RTC in IoT
  - Proprietary
  - Fragmented
  - Weak implementations

DistriNet

Thank you!

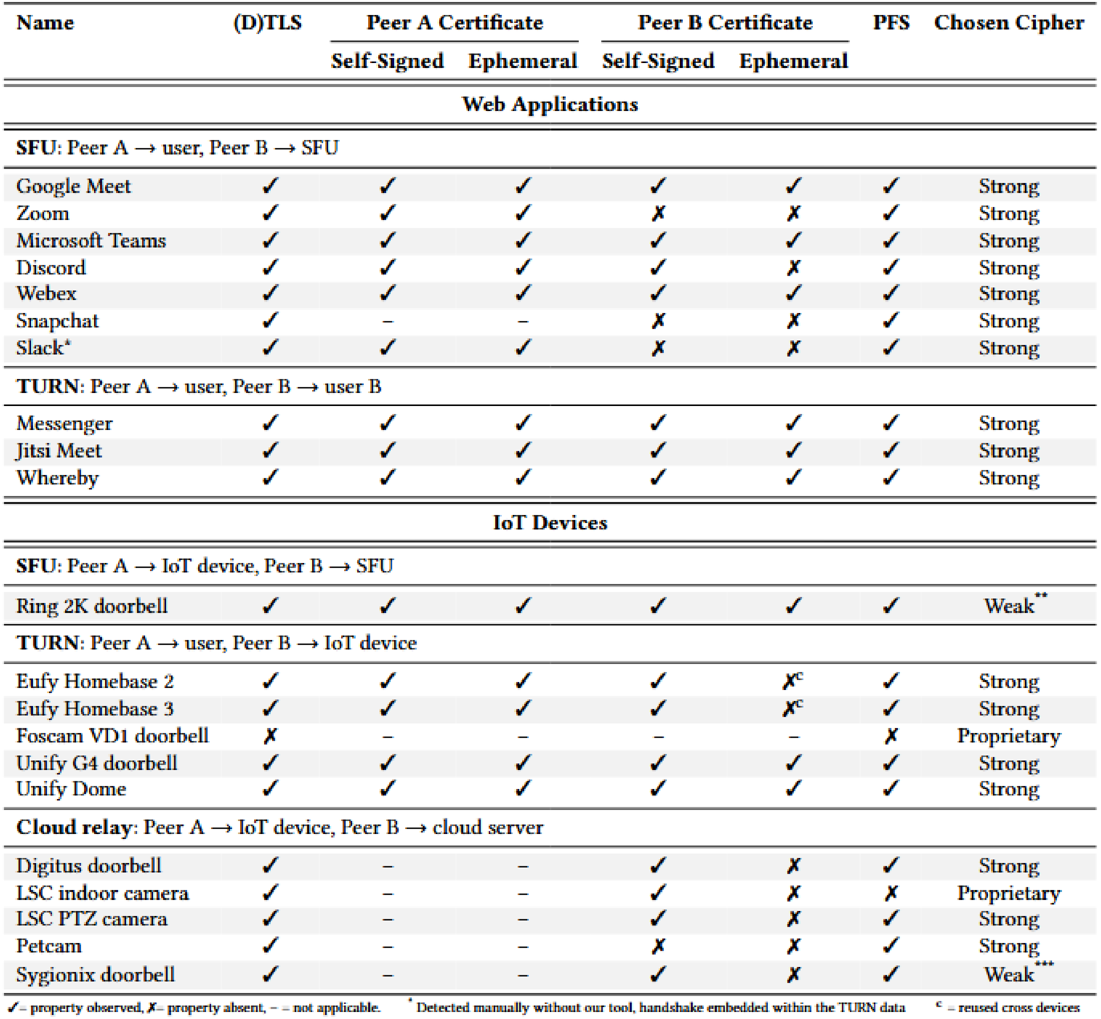
https://distrinet.cs.kuleuven.be/



Name	WebRTC	Signaling			Media Relay		E2E Encrypted
		Int. Protected	Enc. Credentials	Protocol	Relay Type	Location	
Web Applications							
Google Meet	✓	✓	✓	QUIC	SFU	US	✗
Zoom	✓	✓	✓	WebSockets	SFU	US	✗
Microsoft Teams	✓	✓	✓	HTTPS	SFU	EU	✗
Discord	✓	✓	✓	WebSockets	SFU	US	✓
Webex	✓	✓	✓	HTTPS	SFU	EU	✗
Snapchat	✗*	✓	✓	QUIC	SFU	EU	✗
Slack	✓	✓	✓	WS/QUIC	SFU	EU	✗
Messenger	✓	✓	✓	WebSockets	TURN	EU	✓
Jitsi Meet	✓	✓	✓	WebSockets	TURN	EU	[✓]
Whereby	✓	✓	✓	WebSockets	TURN	EU	[✓]
IoT Devices							
Ring 2K doorbell	✓	✓	✓	Proprietary	SFU	EU	✗
Eufy Homebase 2	✓	✗	✗	SIP	TURN	EU	[✓]
Eufy Homebase 3	✓	✓	✓	Proprietary	TURN	EU	[✓]
Foscam VD1 doorbell	✗	✗	✗	STUN	TURN	EU	✗
Unify G4 doorbell	✓	✓	✓	Proprietary	TURN	EU	[✓]
Unify Dome	✓	✓	✓	Proprietary	TURN	EU	[✓]
Digitus doorbell	✗	✗	✗	MQTT	Cloud	EU	✗
LSC indoor camera	✗	✗	✗	MQTT	Cloud	US	✗
LSC PTZ camera	✗	✗	✗	MQTT	Cloud	EU	✗
Petcam	✗	✓	✓	Proprietary	Cloud	EU	✗
Sygionix doorbell	✗	✗	✗	MQTT	Cloud	EU	✗

✓ – property observed, ✗ – property absent, – – not applicable. E2E Encrypted – *strict* E2E encryption of the media channel * Uses a proprietary variant of WebRTC.
[✓] Provides media-level E2E encryption, not strict E2E encryption.

Table 1: Protocol adoption and signaling characteristics



Name	(D)TLS	Peer A Certificate		Peer B Certificate		PFS	Chosen Cipher
		Self-Signed	Ephemeral	Self-Signed	Ephemeral		
Web Applications							
SFU: Peer A → user, Peer B → SFU							
Google Meet	✓	✓	✓	✓	✓	✓	Strong
Zoom	✓	✓	✓	✗	✗	✓	Strong
Microsoft Teams	✓	✓	✓	✓	✓	✓	Strong
Discord	✓	✓	✓	✓	✗	✓	Strong
Webex	✓	✓	✓	✓	✓	✓	Strong
Snapchat	✓	–	–	✗	✗	✓	Strong
Slack*	✓	✓	✓	✗	✗	✓	Strong
TURN: Peer A → user, Peer B → user B							
Messenger	✓	✓	✓	✓	✓	✓	Strong
Jitsi Meet	✓	✓	✓	✓	✓	✓	Strong
Whereby	✓	✓	✓	✓	✓	✓	Strong
IoT Devices							
SFU: Peer A → IoT device, Peer B → SFU							
Ring 2K doorbell	✓	✓	✓	✓	✓	✓	Weak**
TURN: Peer A → user, Peer B → IoT device							
Eufy Homebase 2	✓	✓	✓	✓	✗^{c}	✓	Strong
Eufy Homebase 3	✓	✓	✓	✓	✗^{c}	✓	Strong
Foscam VD1 doorbell	✗	–	–	–	–	✗	Proprietary
Unify G4 doorbell	✓	✓	✓	✓	✓	✓	Strong
Unify Dome	✓	✓	✓	✓	✓	✓	Strong
Cloud relay: Peer A → IoT device, Peer B → cloud server							
Digitus doorbell	✓	–	–	✓	✗	✓	Strong
LSC indoor camera	✓	–	–	✓	✗	✗	Proprietary
LSC PTZ camera	✓	–	–	✓	✗	✓	Strong
Petcam	✓	–	–	✗	✗	✓	Strong
Sygionix doorbell	✓	–	–	✓	✗	✓	Weak***

✓ = property observed, ✗ = property absent, – = not applicable. * Detected manually without our tool, handshake embedded within the TURN data ^{c} = reused cross devices

Table 2: Cryptographic hygiene and certificate management in the media channel