Fraunhofer
FKIE

friTap: Decrypting TLS Traffic on the Fly

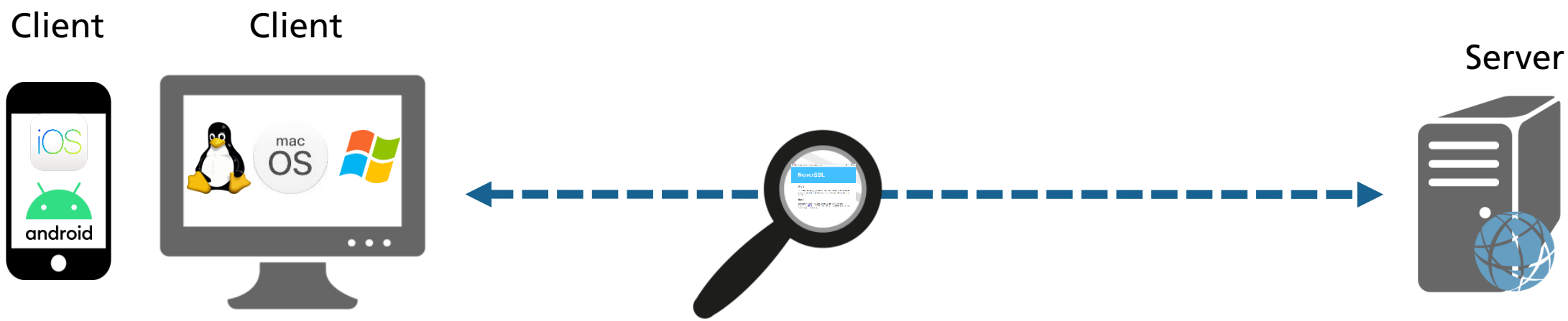
Daniel Baier **daniel.baier@fkie.fraunhofer.de**
Francois Egner francois.egner@fkie.fraunhofer.de
Max J. Ufer max.jens.ufer@fkie.fraunhofer.de

Fraunhofer FKIE

Introduction

Fraunhofer FKIE

Introduction



Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telephonie Wireless Tools

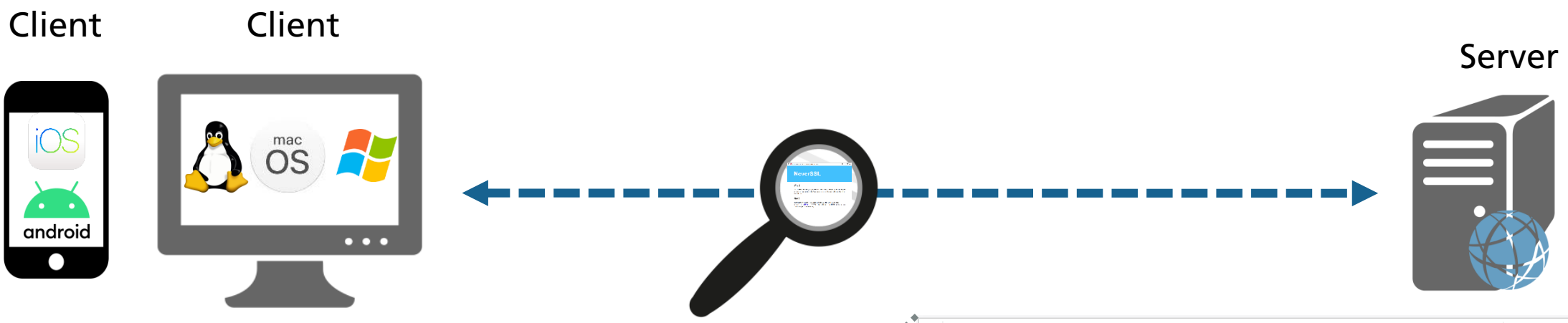
Aktueller Filter: tcp.stream eq 2

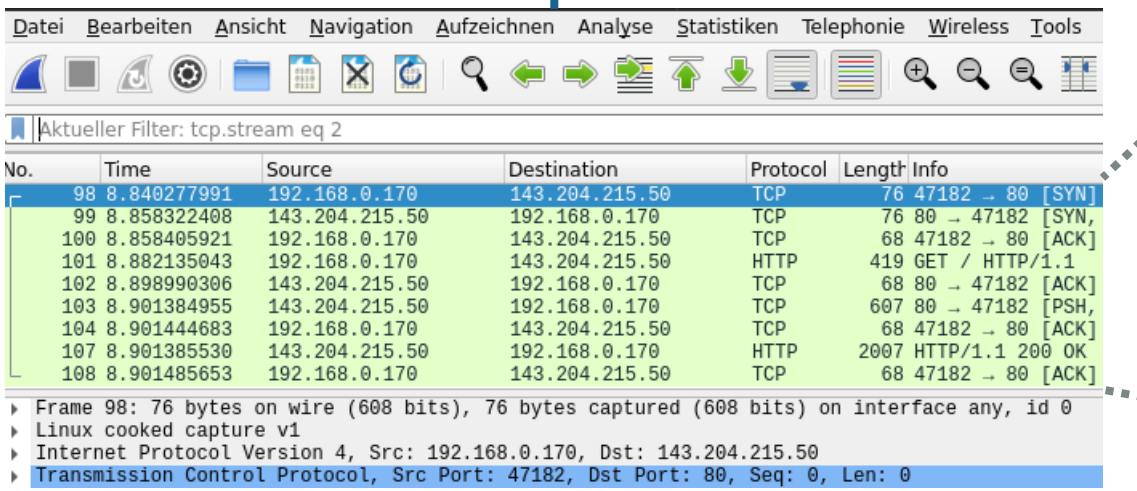
No.	Time	Source	Destination	Protocol	Length	Info
98	8.840277991	192.168.0.170	143.204.215.50	TCP	76	47182 → 80 [SYN]
99	8.858322408	143.204.215.50	192.168.0.170	TCP	76	80 → 47182 [SYN,
100	8.858405921	192.168.0.170	143.204.215.50	TCP	68	47182 → 80 [ACK]
101	8.882135043	192.168.0.170	143.204.215.50	HTTP	419	GET / HTTP/1.1
102	8.898990306	143.204.215.50	192.168.0.170	TCP	68	80 → 47182 [ACK]
103	8.901384955	143.204.215.50	192.168.0.170	TCP	607	80 → 47182 [PSH,
104	8.901444683	192.168.0.170	143.204.215.50	TCP	68	47182 → 80 [ACK]
107	8.901385530	143.204.215.50	192.168.0.170	HTTP	2007	HTTP/1.1 200 OK
108	8.901485653	192.168.0.170	143.204.215.50	TCP	68	47182 → 80 [ACK]

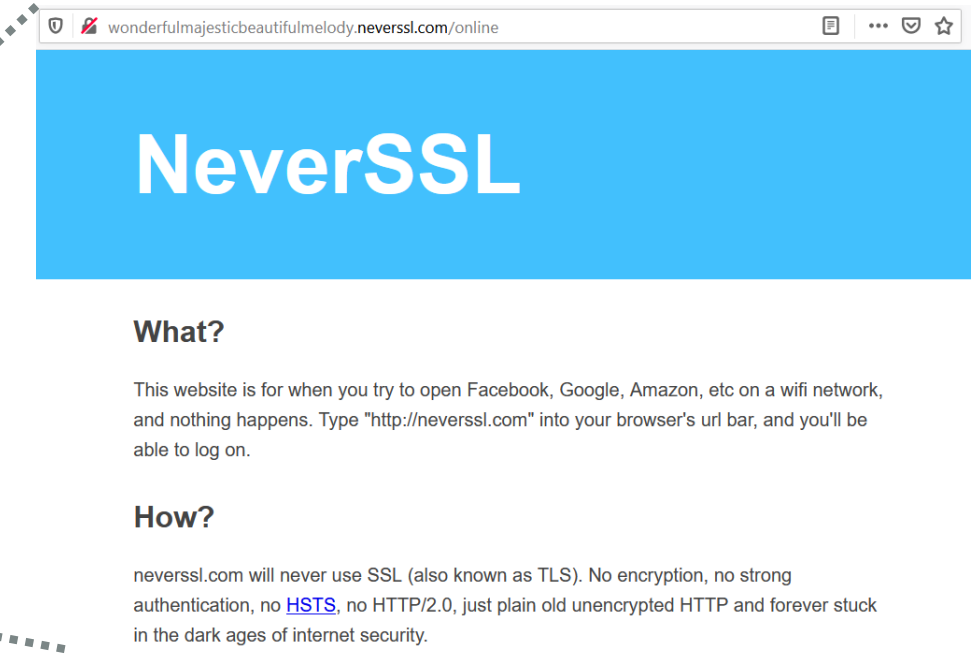
```
Frame 98: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.168.0.170, Dst: 143.204.215.50
Transmission Control Protocol, Src Port: 47182, Dst Port: 80, Seq: 0, Len: 0
```

Fraunhofer FKIE

Introduction

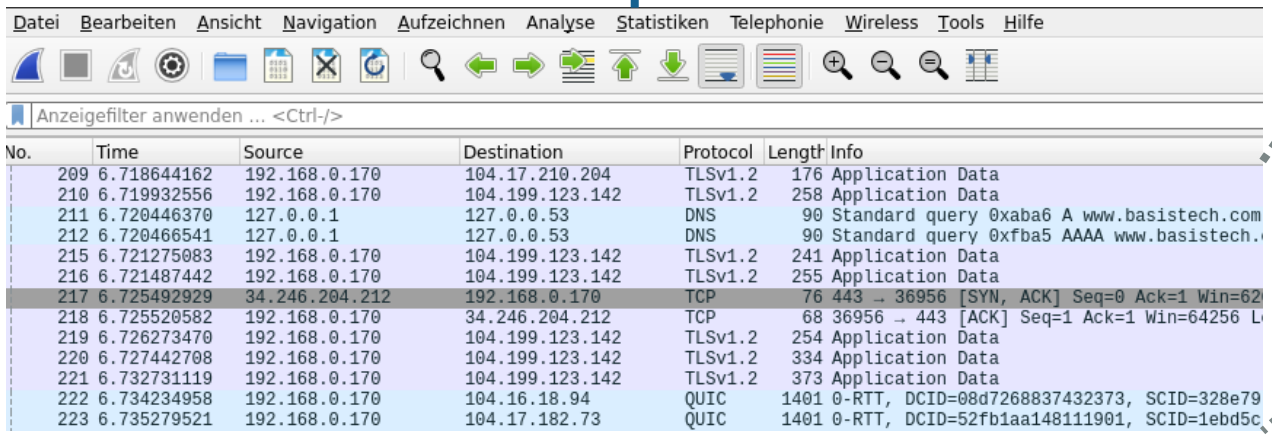






Fraunhofer FKIE

Challenge: SSL/TLS



Fraunhofer FKIE

Challenge: SSL/TLS

Client

Client

Server

Nowadays most network traffic is encrypted. Mostly utilizing SSL/TLS ☹

Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telephonie Wireless Tools Hilfe

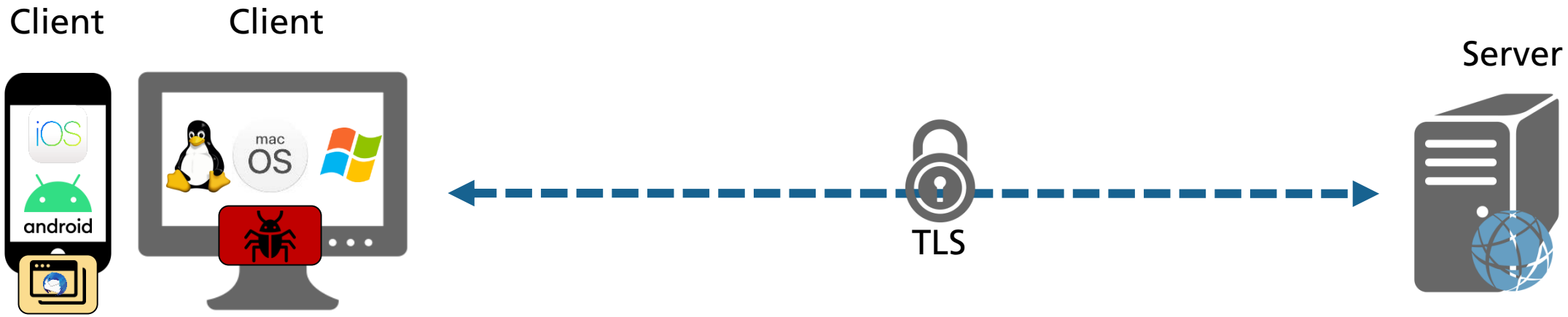
Anzeigefilter anwenden … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
209	6.718644162	192.168.0.170	104.17.210.204	TLSv1.2	176	Application Data
210	6.719932556	192.168.0.170	104.199.123.142	TLSv1.2	258	Application Data
211	6.720446370	127.0.0.1	127.0.0.53	DNS	90	Standard query 0xaba6 A www.basistech.com
212	6.720466541	127.0.0.1	127.0.0.53	DNS	90	Standard query 0xfba5 AAAA www.basistech.
215	6.721275083	192.168.0.170	104.199.123.142	TLSv1.2	241	Application Data
216	6.721487442	192.168.0.170	104.199.123.142	TLSv1.2	255	Application Data
217	6.725492929	34.246.204.212	192.168.0.170	TCP	76	443 → 36956 [SYN, ACK] Seq=0 Ack=1 Win=62
218	6.725520582	192.168.0.170	34.246.204.212	TCP	68	36956 → 443 [ACK] Seq=1 Ack=1 Win=64256 L
219	6.726273470	192.168.0.170	104.199.123.142	TLSv1.2	254	Application Data
220	6.727442708	192.168.0.170	104.199.123.142	TLSv1.2	334	Application Data
221	6.732731119	192.168.0.170	104.199.123.142	TLSv1.2	373	Application Data
222	6.734234958	192.168.0.170	104.16.18.94	QUIC	1401	0-RTT, DCID=08d7268837432373, SCID=328e79
223	6.735279521	192.168.0.170	104.17.182.73	QUIC	1401	0-RTT, DCID=52fb1aa148111901, SCID=1ebd5c

This is perfect for our privacy **but…**

Fraunhofer FKIE

Challenge: Traffic analysis

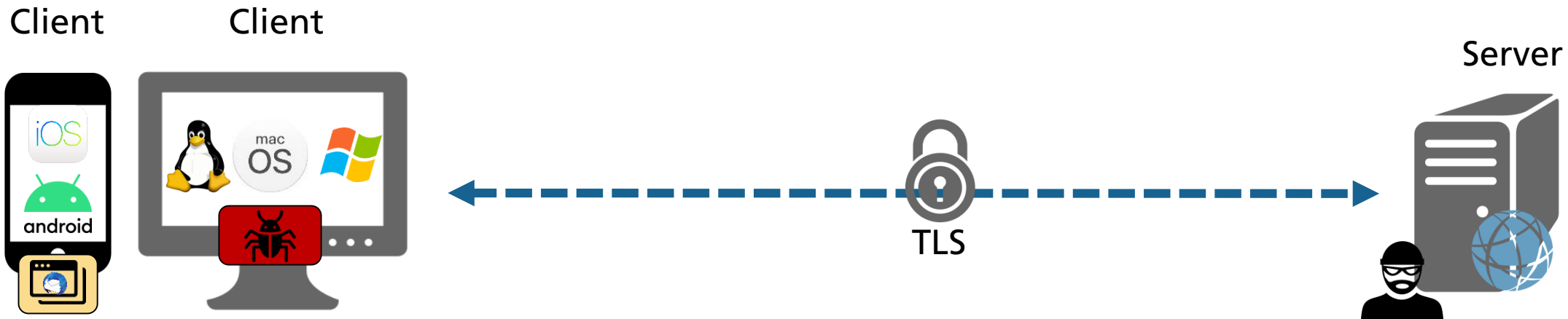


TLS encryption prevents forensic insights:

☒ Monitoring

Fraunhofer FKIE

Challenge: Traffic analysis

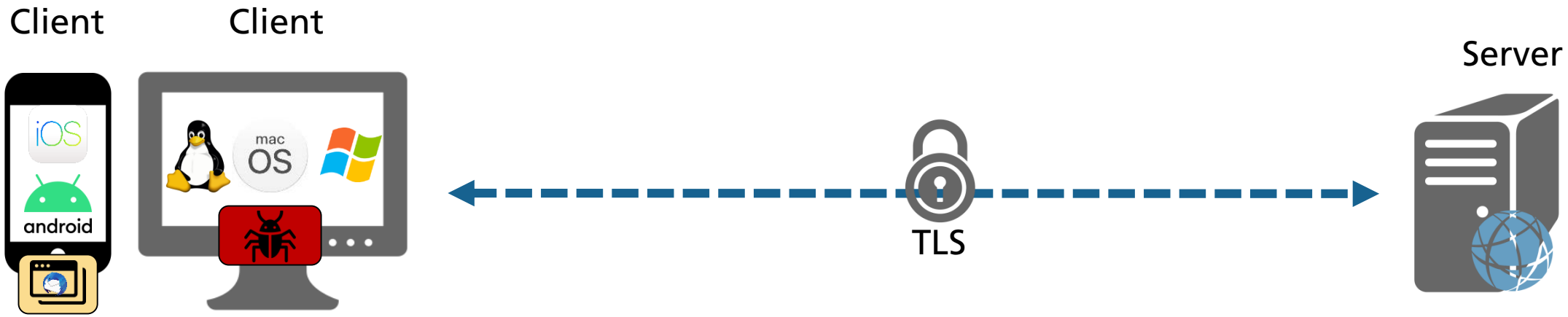


TLS encryption prevents forensic insights:

- ☒ Monitoring
- ☒ Malware analysis

Fraunhofer FKIE

Challenge: Traffic analysis

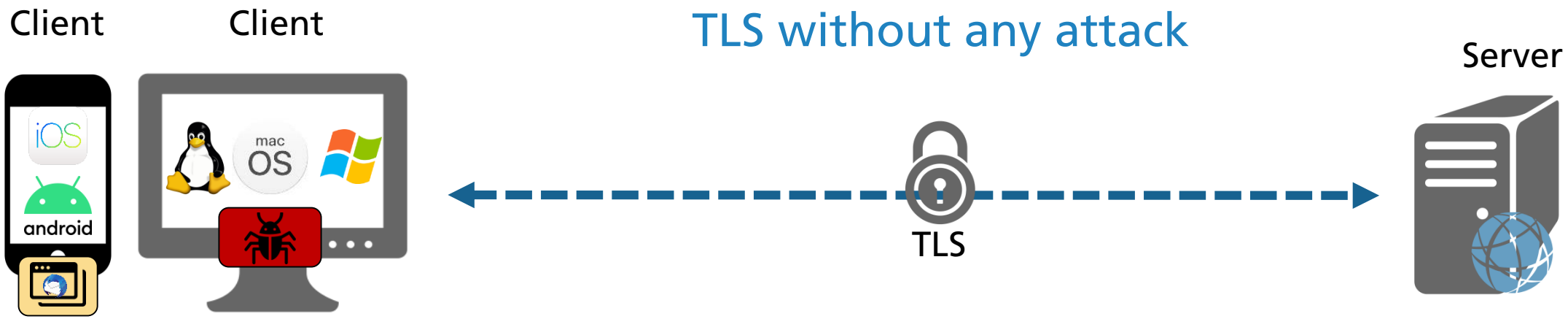


TLS encryption prevents forensic insights:

- ☒ Monitoring
- ☒ Malware analysis
- ☒ Proprietary protocols

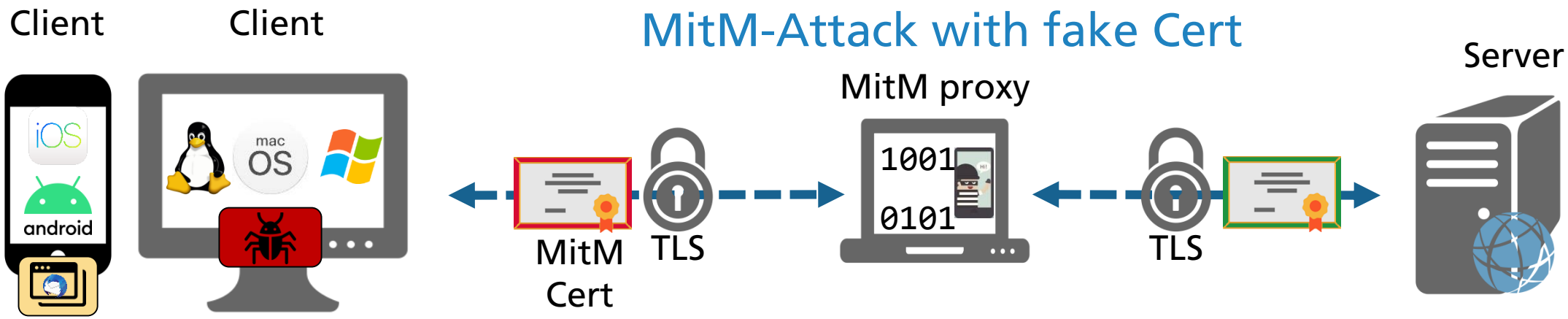
Fraunhofer FKIE

Current state: MitM proxy



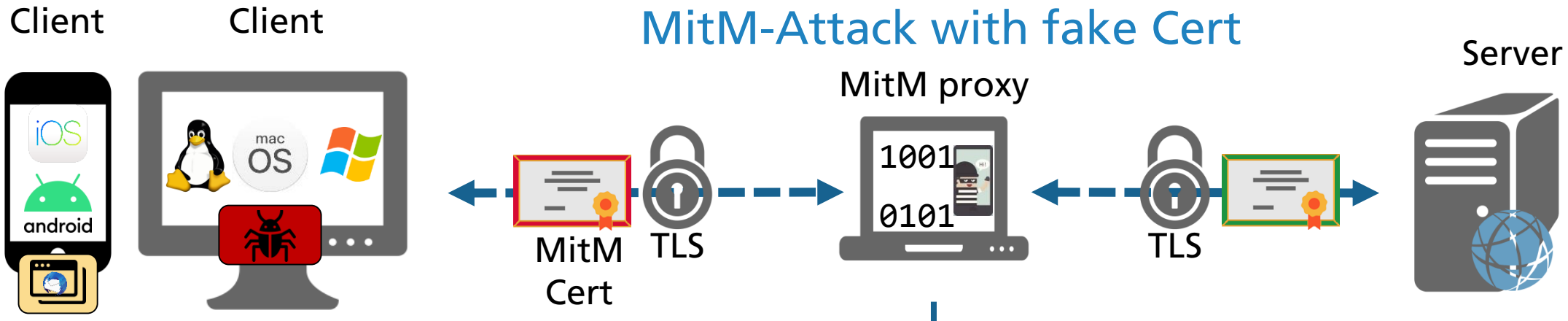
Fraunhofer FKIE

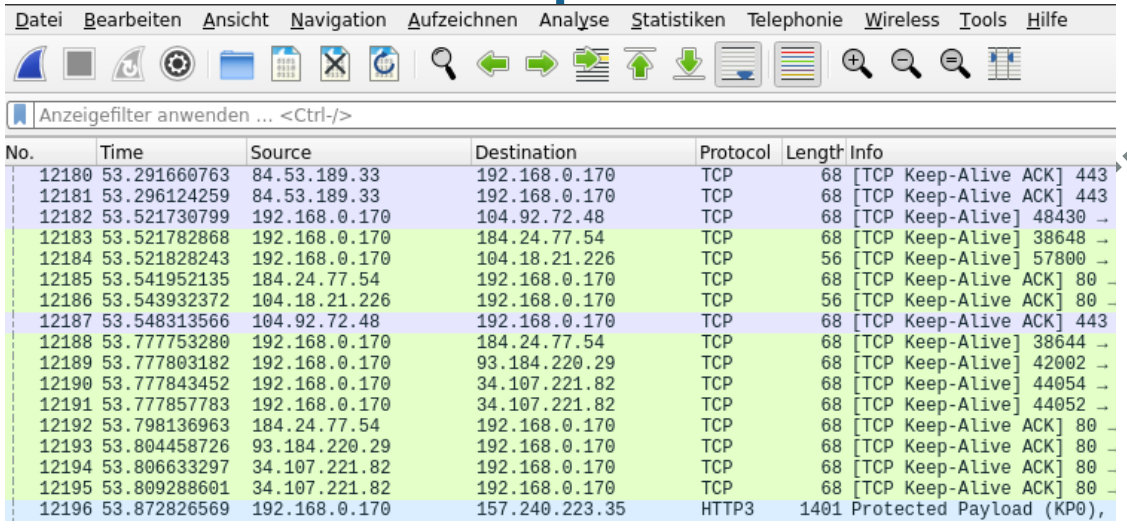
Current state: MitM proxy

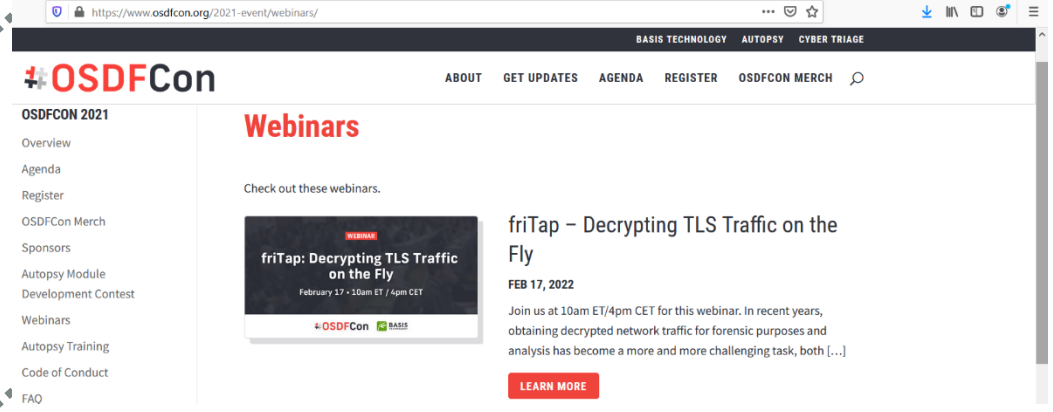


Fraunhofer FKIE

Current state: MitM proxy

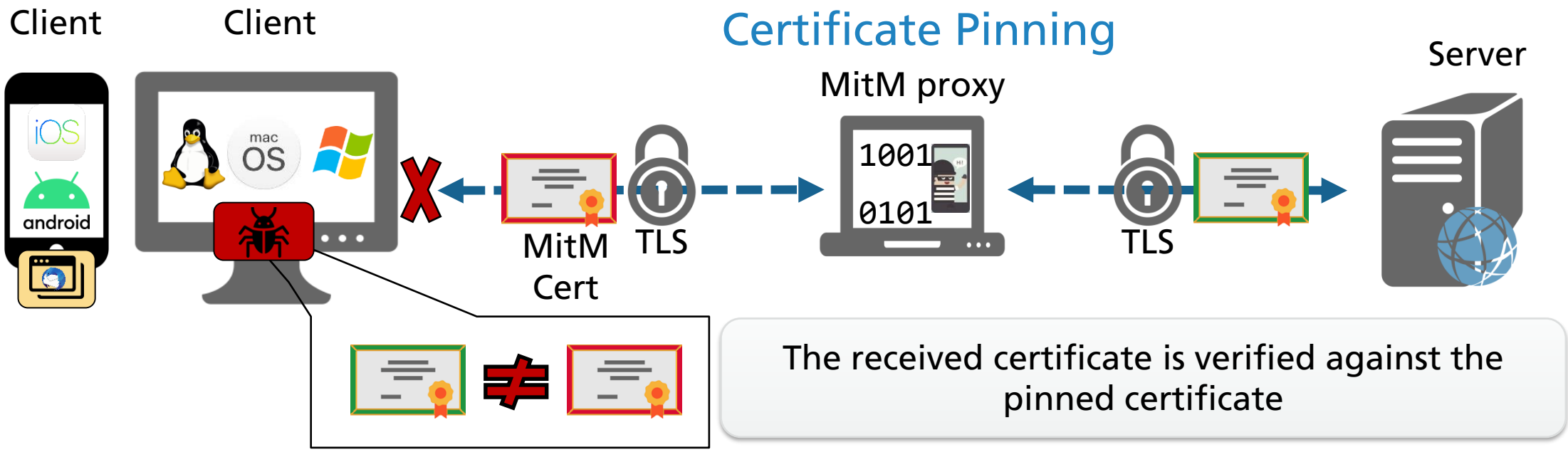






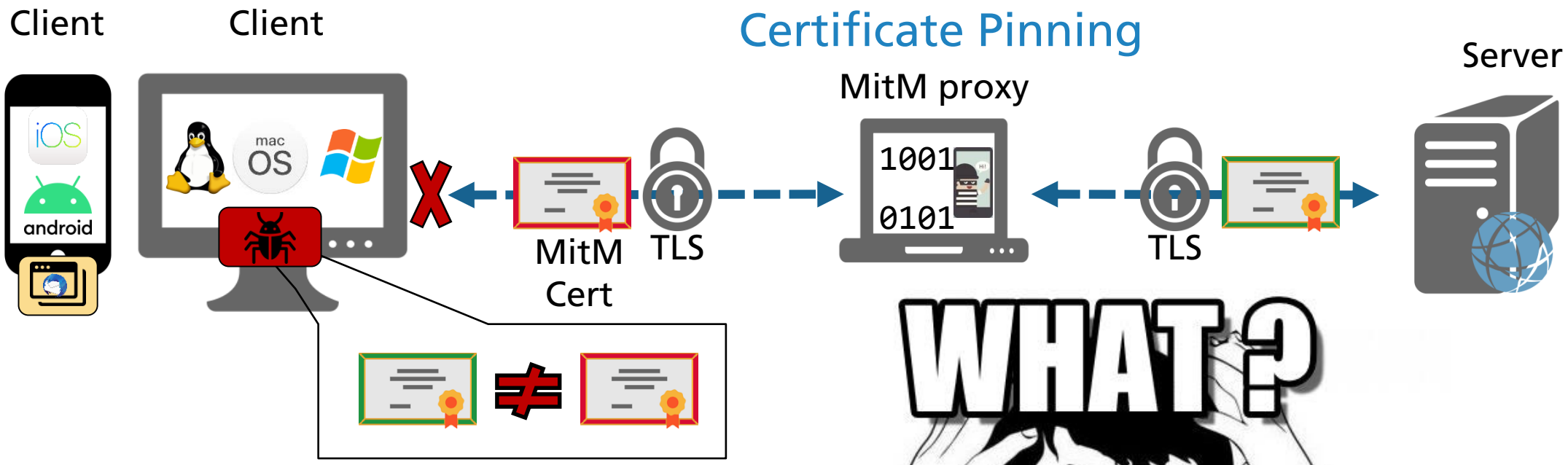
Fraunhofer FKIE

Current state: MitM proxy

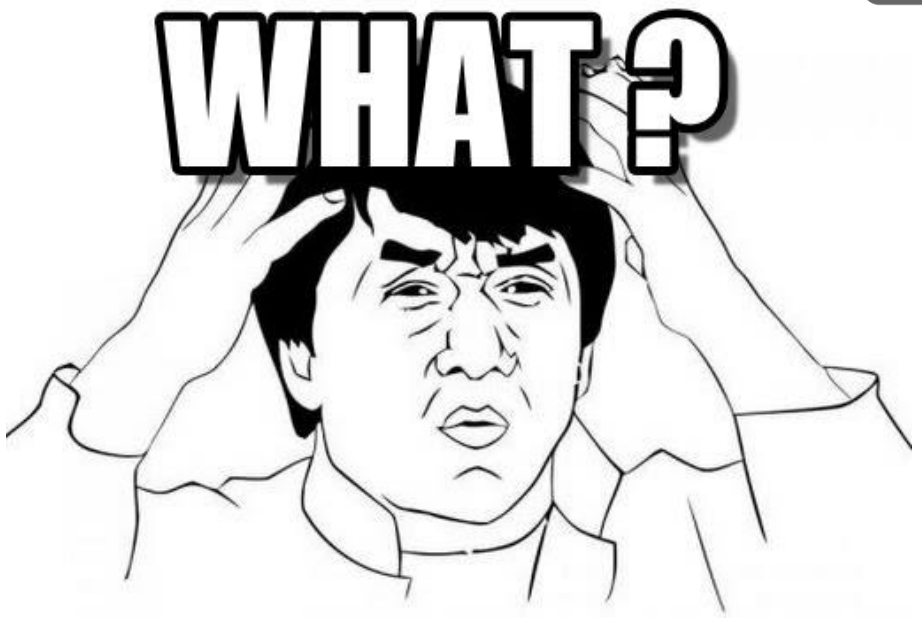


Fraunhofer FKIE

Current state: MitM proxy

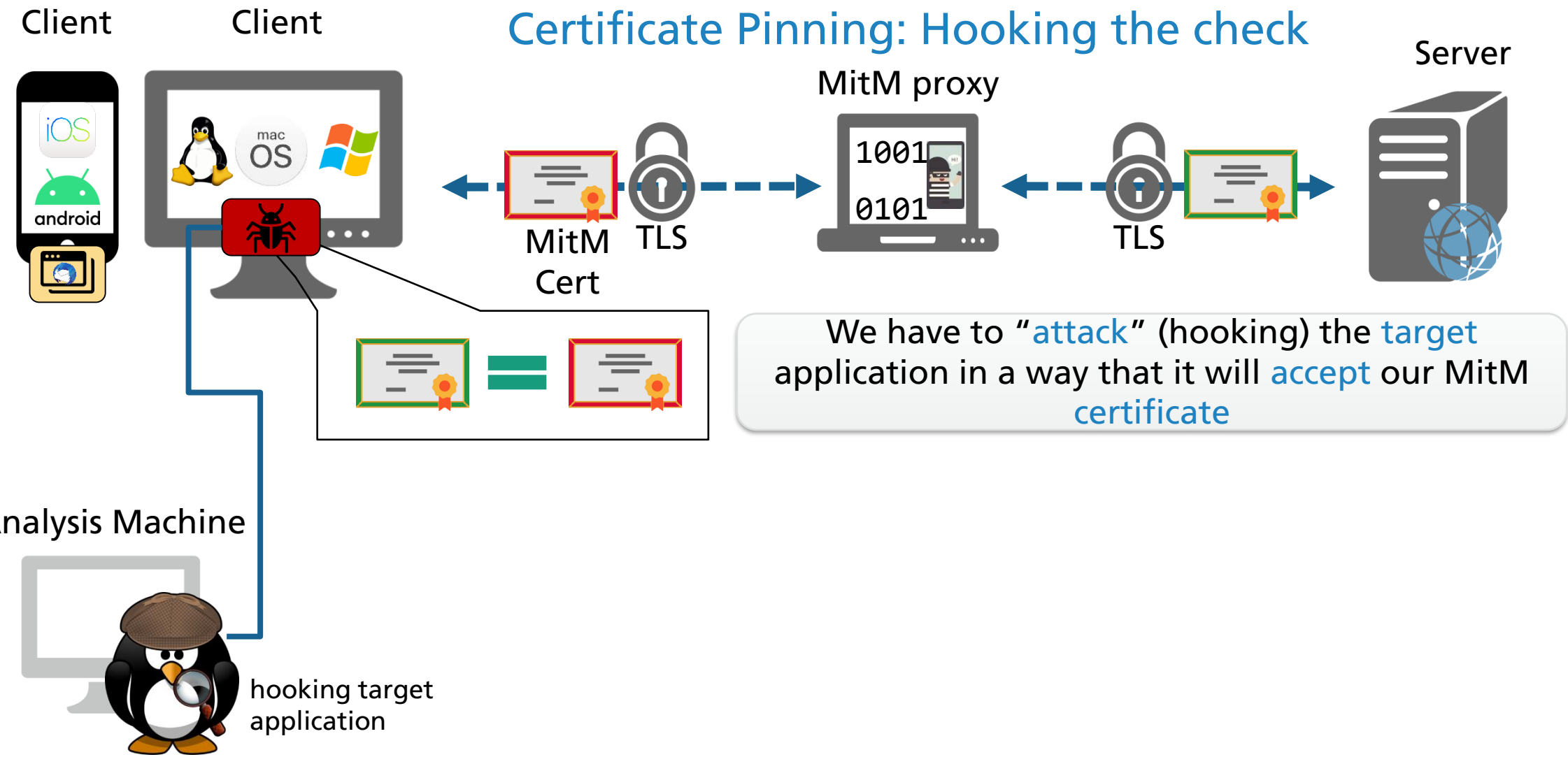


Without disabling this check in the application we are not able to redirect the traffic into our MitM proxy



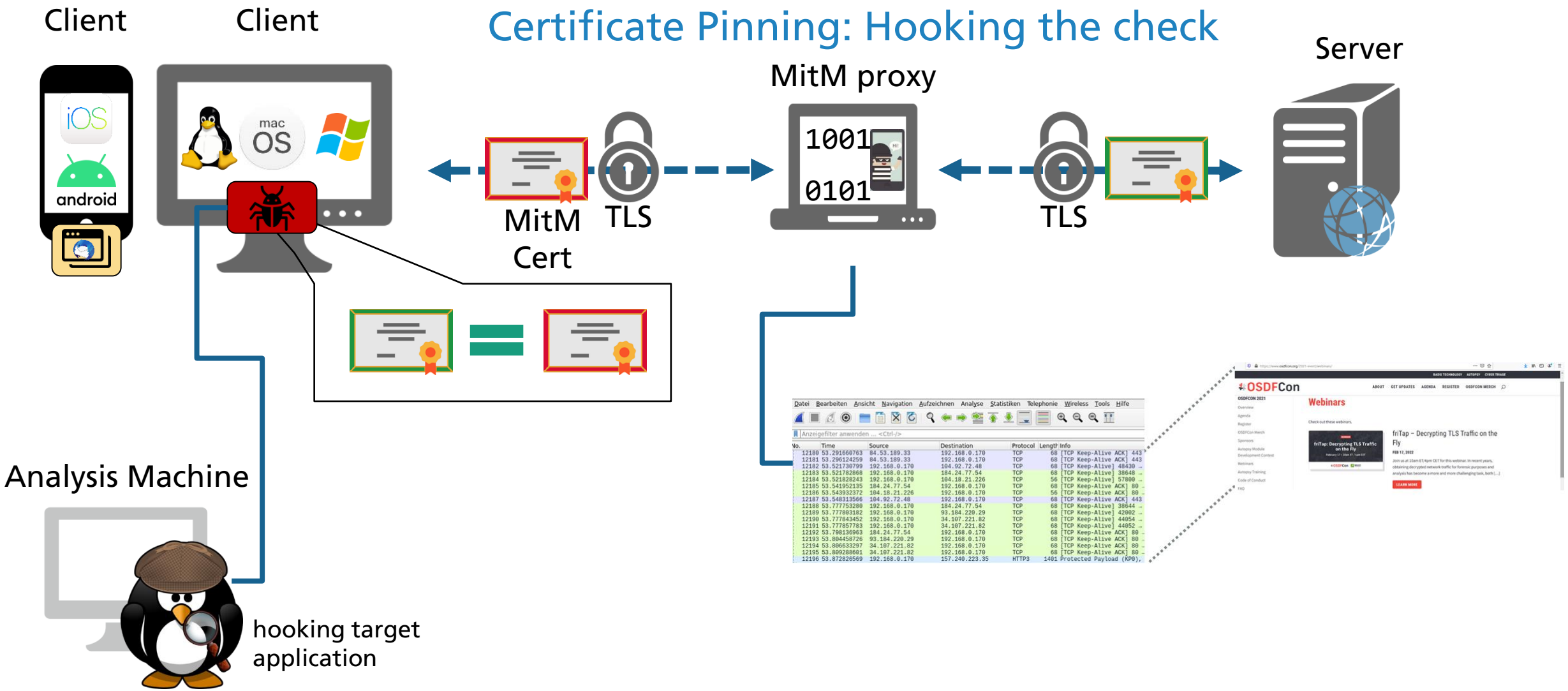
Fraunhofer FKIE

Current state: MitM proxy



Fraunhofer FKIE

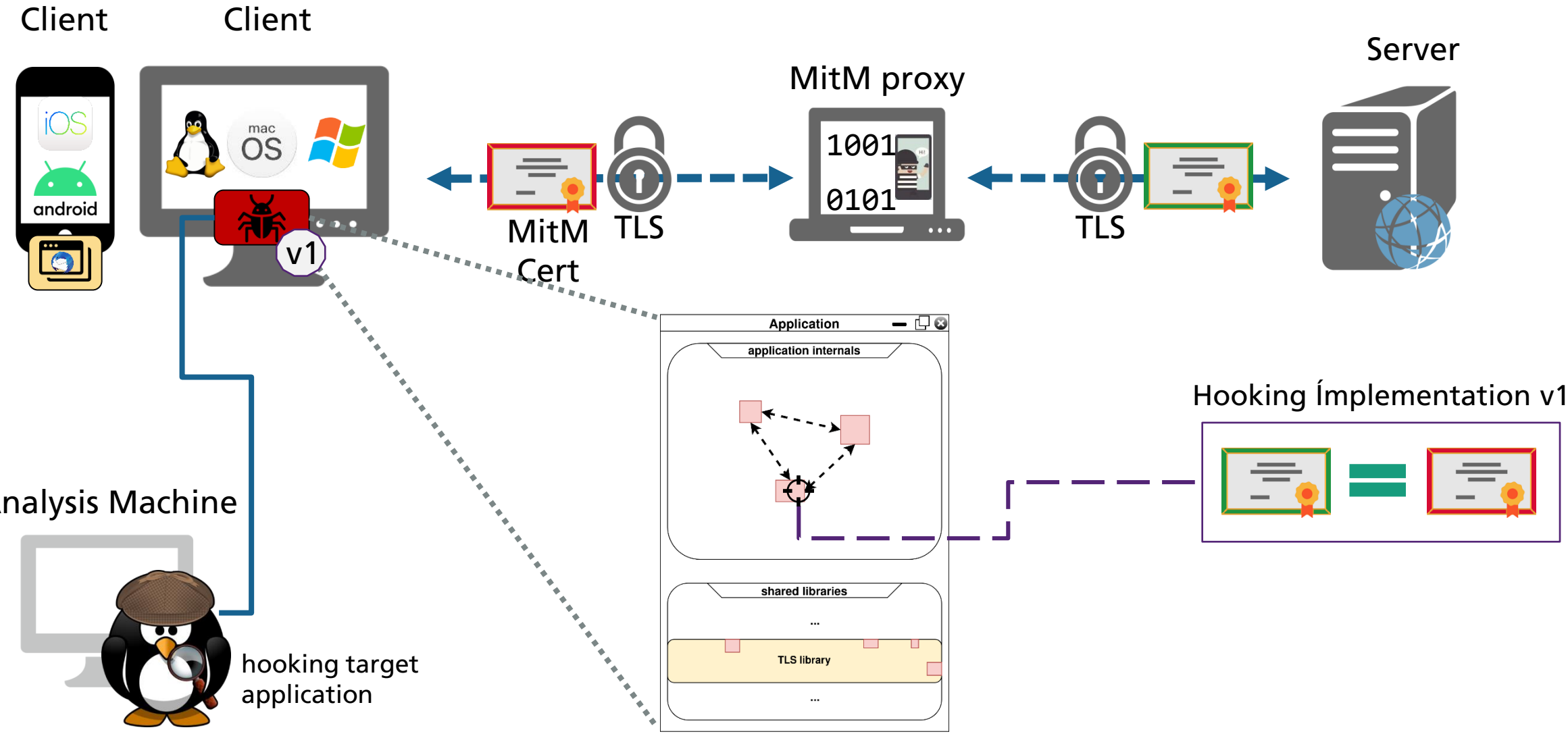
Current state: MitM proxy



Fraunhofer FKIE

Problem: Certificate Pinning implementations

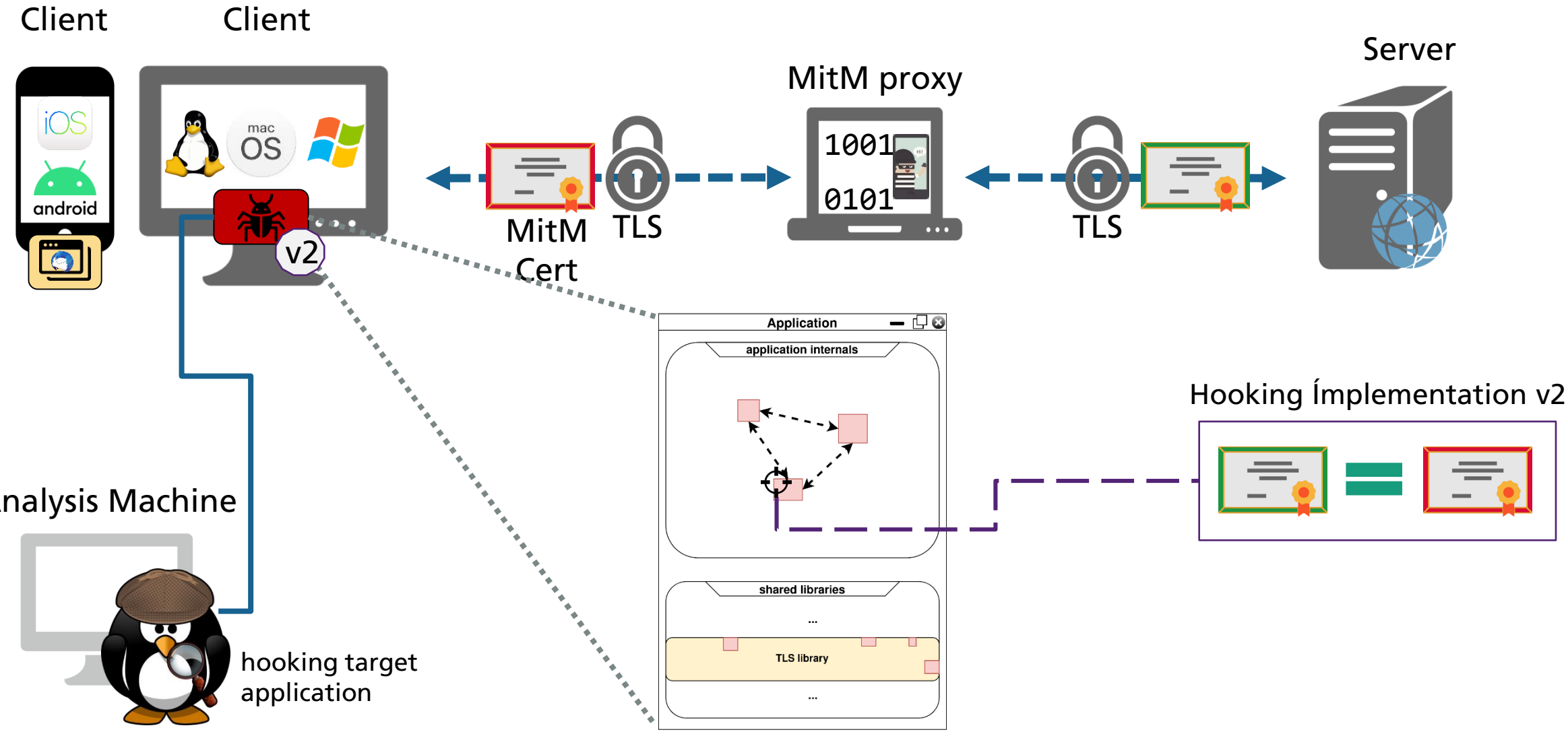
There are many changes between different versions



Fraunhofer FKIE

Problem: Certificate Pinning implementations

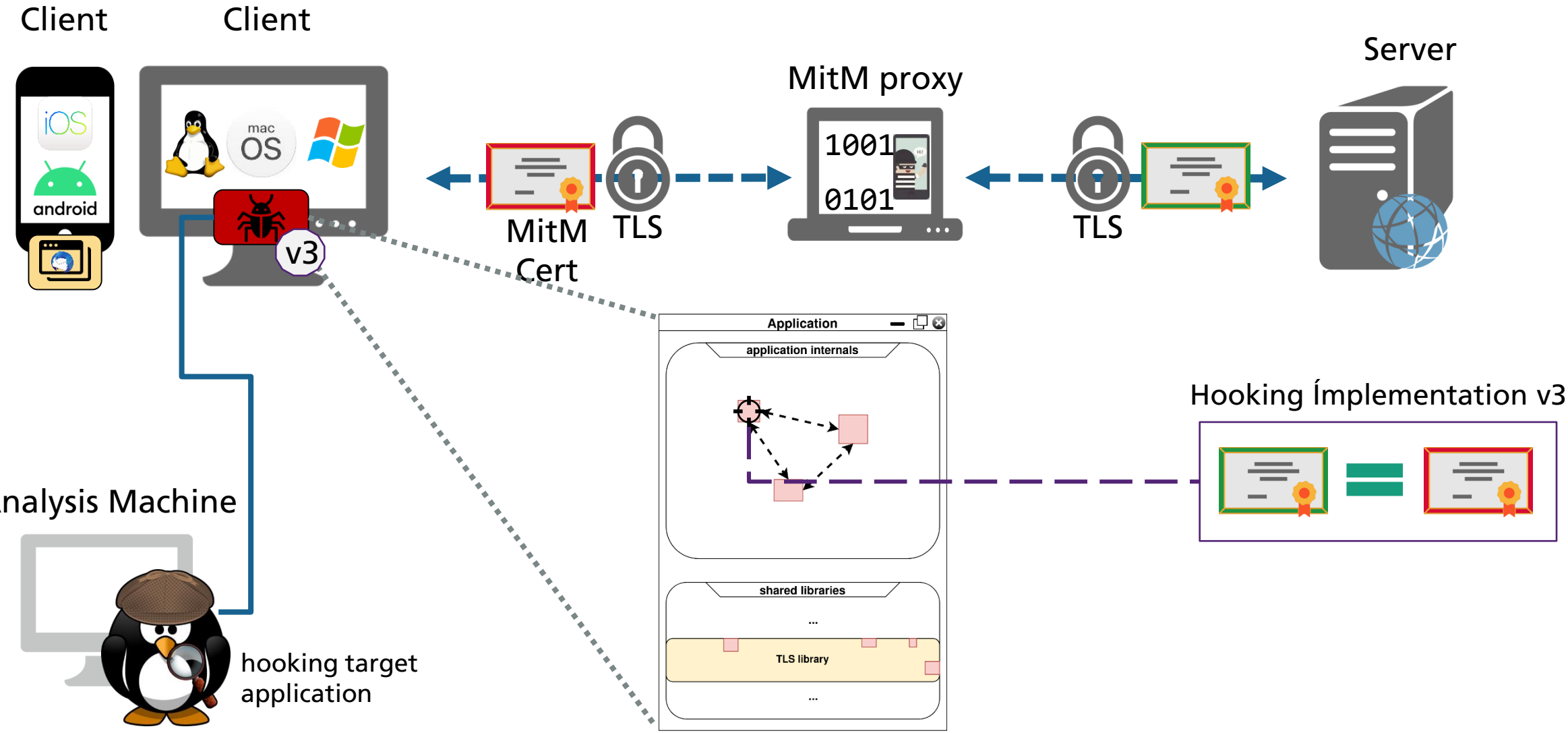
There are many changes between different versions



Fraunhofer FKIE

Problem: Certificate Pinning implementations

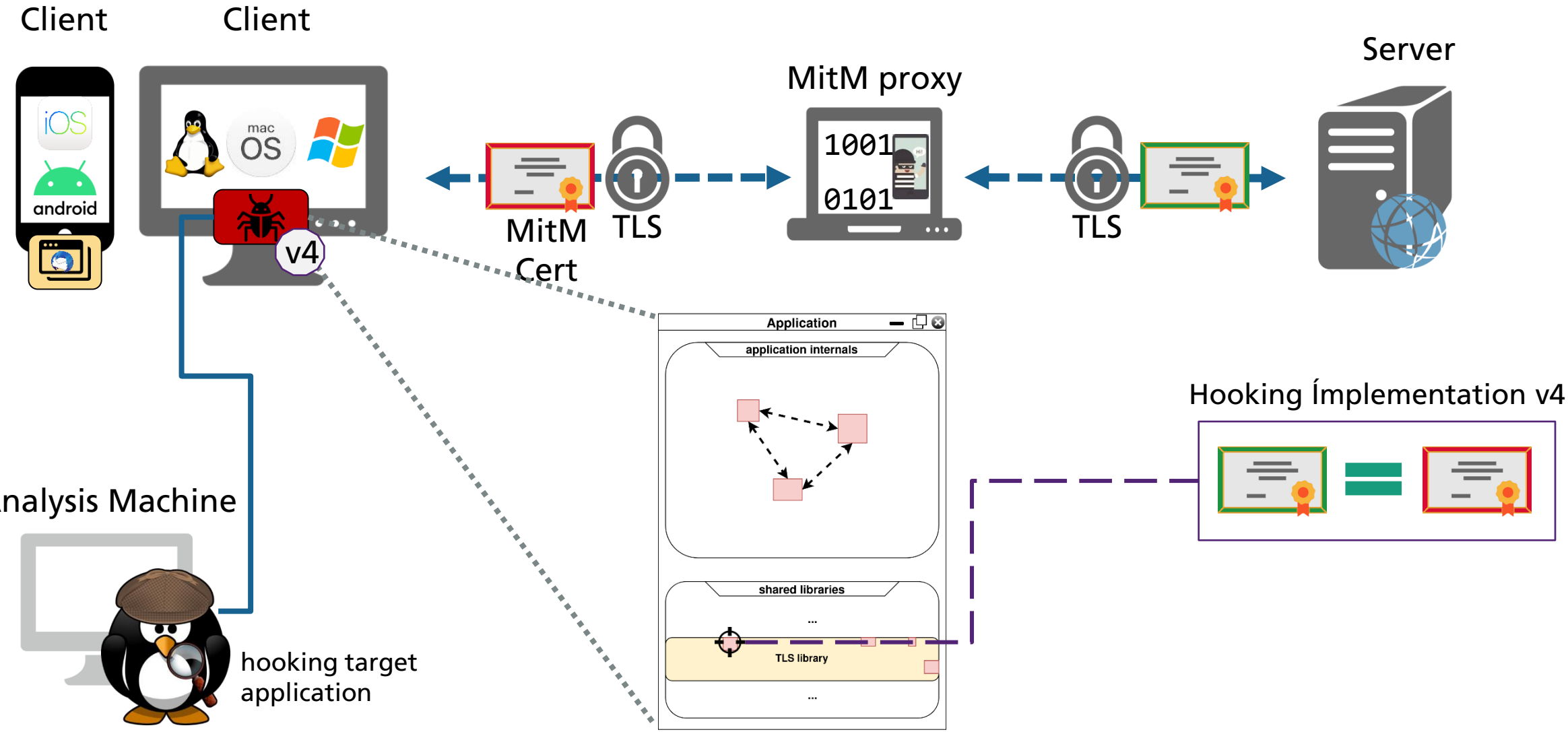
There are many changes between different versions



Fraunhofer FKIE

Problem: Certificate Pinning implementations

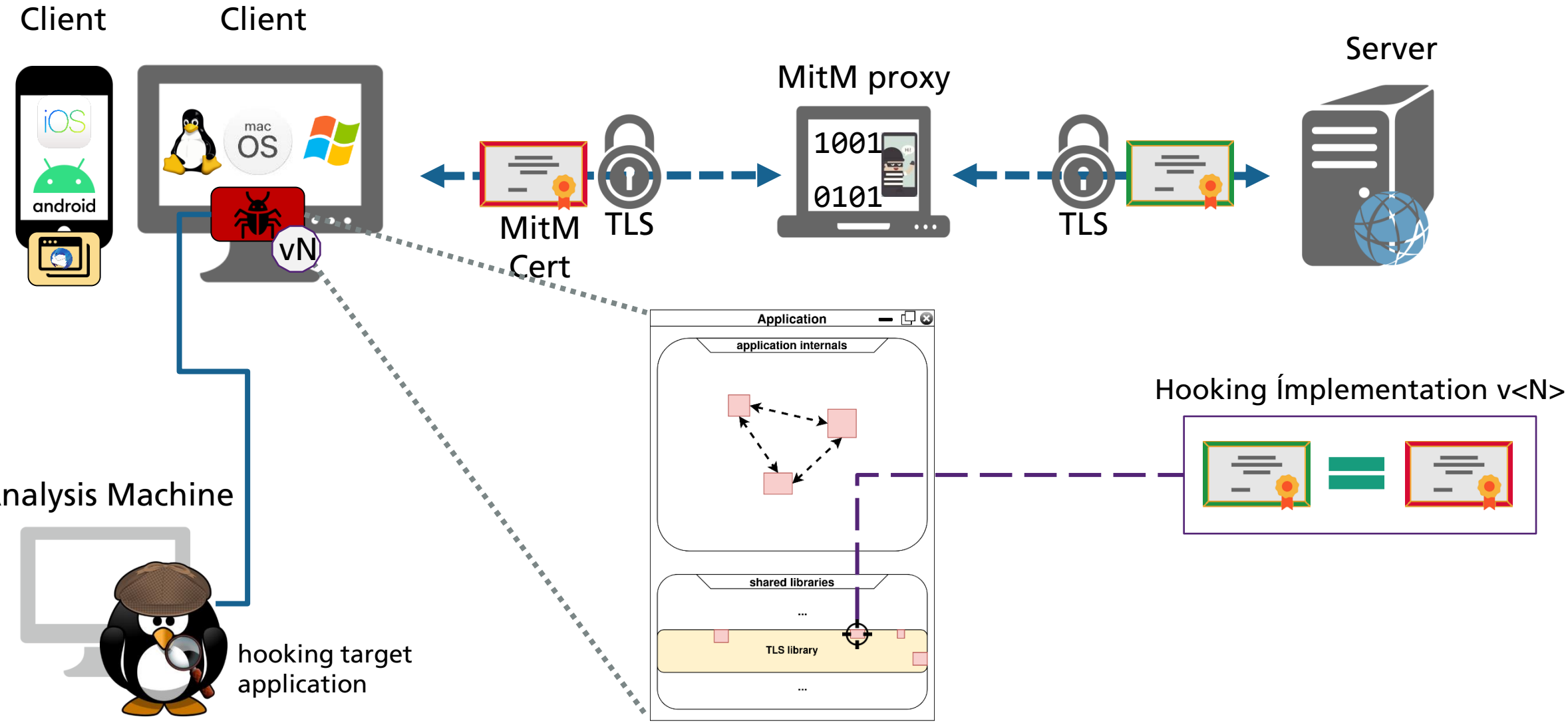
There are many changes between different versions



Fraunhofer FKIE

Problem: Certificate Pinning implementations

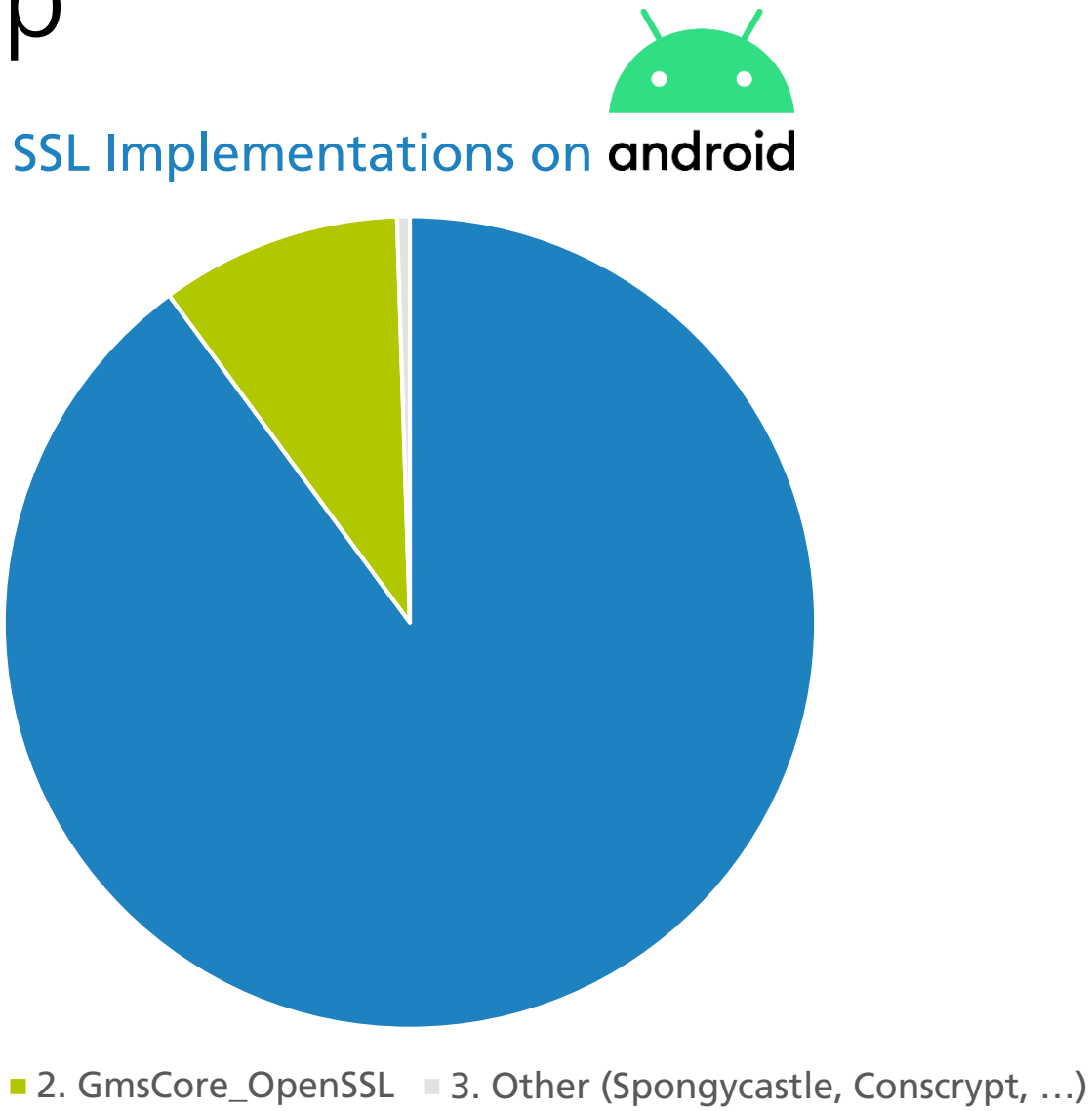
There are many changes between different versions



When we have to „attack“ the App why not directly extract the payload of the TLS stream?

Fraunhofer FKIE

Solution: friTap



Fraunhofer FKIE

Solution: friTap

https://github.com/fkie-cad/friTap

- ☑ TLS keys
- ☑ Decypted TLS payload as PCAP

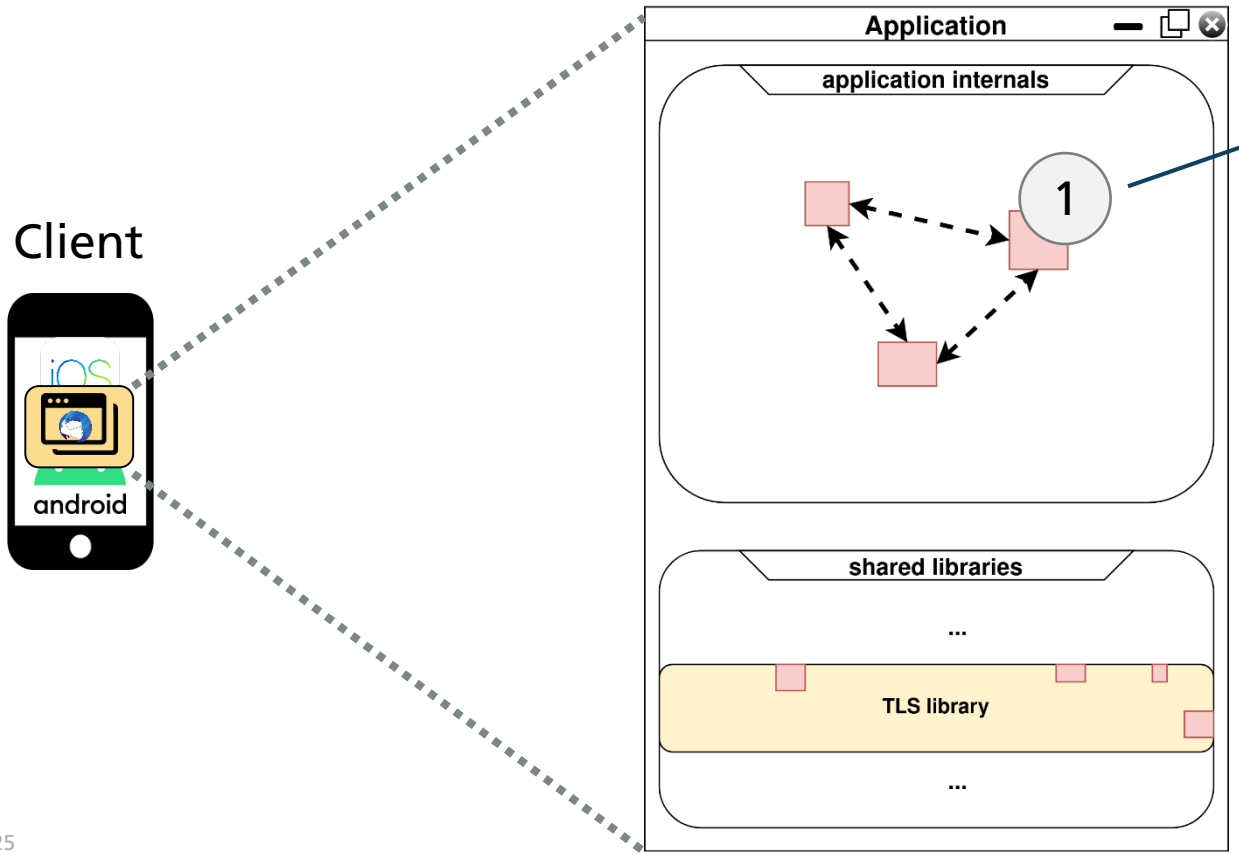
```
FKIE ~/DEF/research/friTap ./friTap.py –m –k TLS_keys.log <target_app>
Usage: friTap.py [-m] [-d] [-k <path>] [-l] [-p <path>] [-s] [-env <env.json>] [-v] [--enable_spawn_gating] <executable/app name/pid>

Decrypts and logs an executables or mobile applications SSL/TLS traffic.

Arguments:
  -m, --mobile          Attach to a process on android or iOS
  -d, --debug           Set the debug output of friTap
  -k <path>, --keylog <path>
                        Log the keys used for tls traffic
  -l, --live            Creates a named pipe /tmp/sharkfin which can be read by Wireshark during the capturing process
  -p  <path>, --pcap <path>
…
```

Fraunhofer FKIE

friTap: Approach

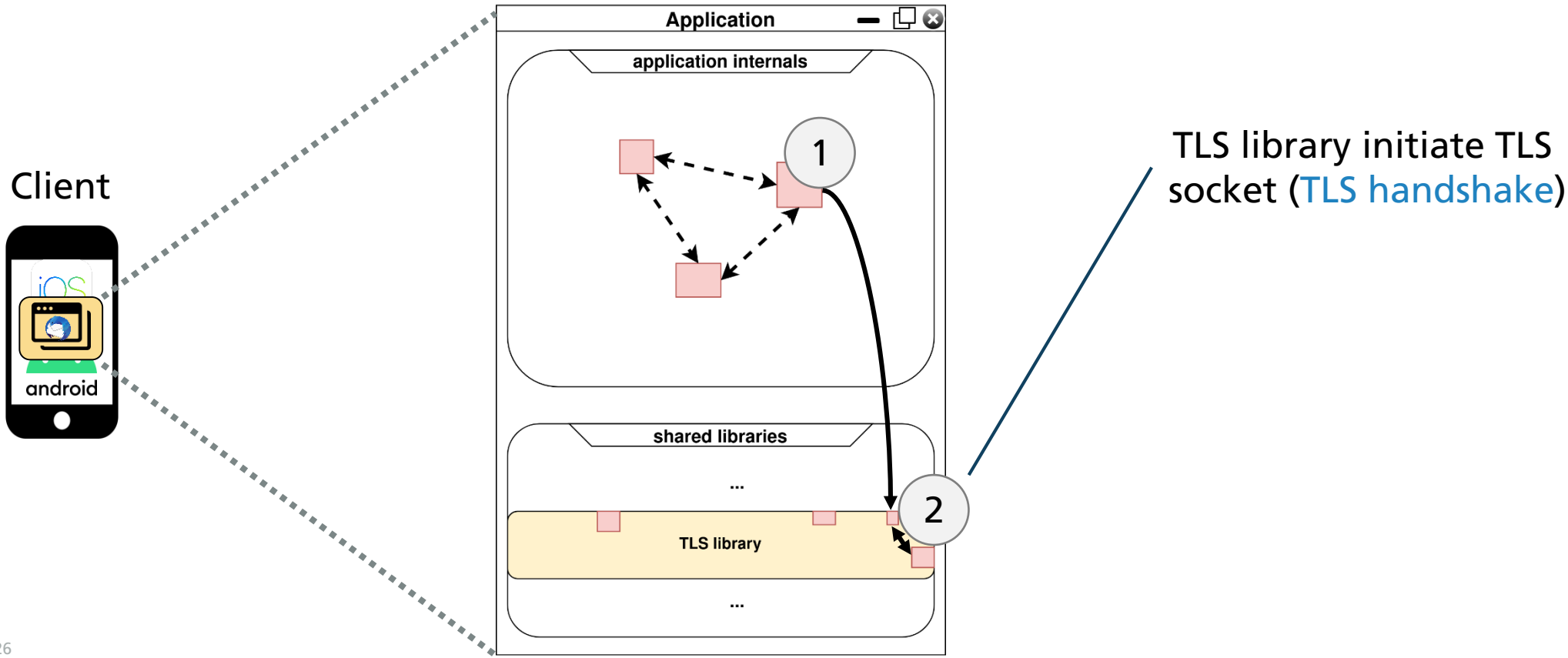


Application decides to create TLS connection.

Server

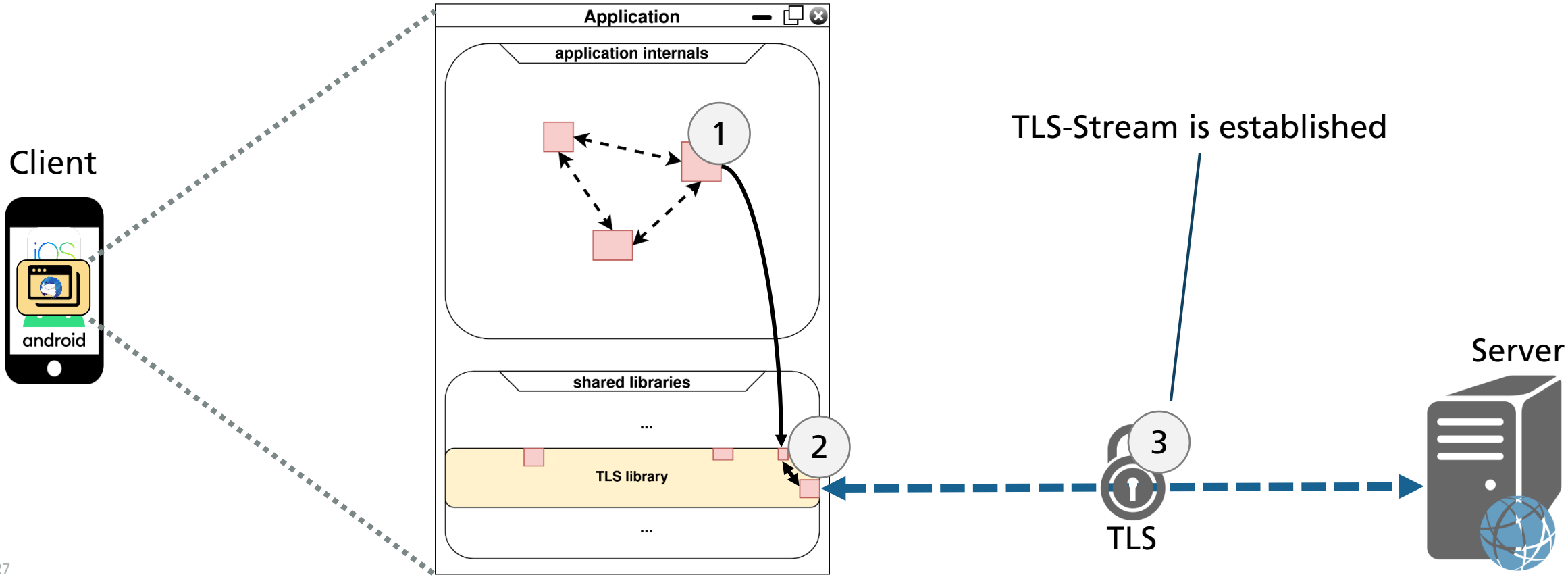
Fraunhofer FKIE

friTap: Approach



Fraunhofer FKIE

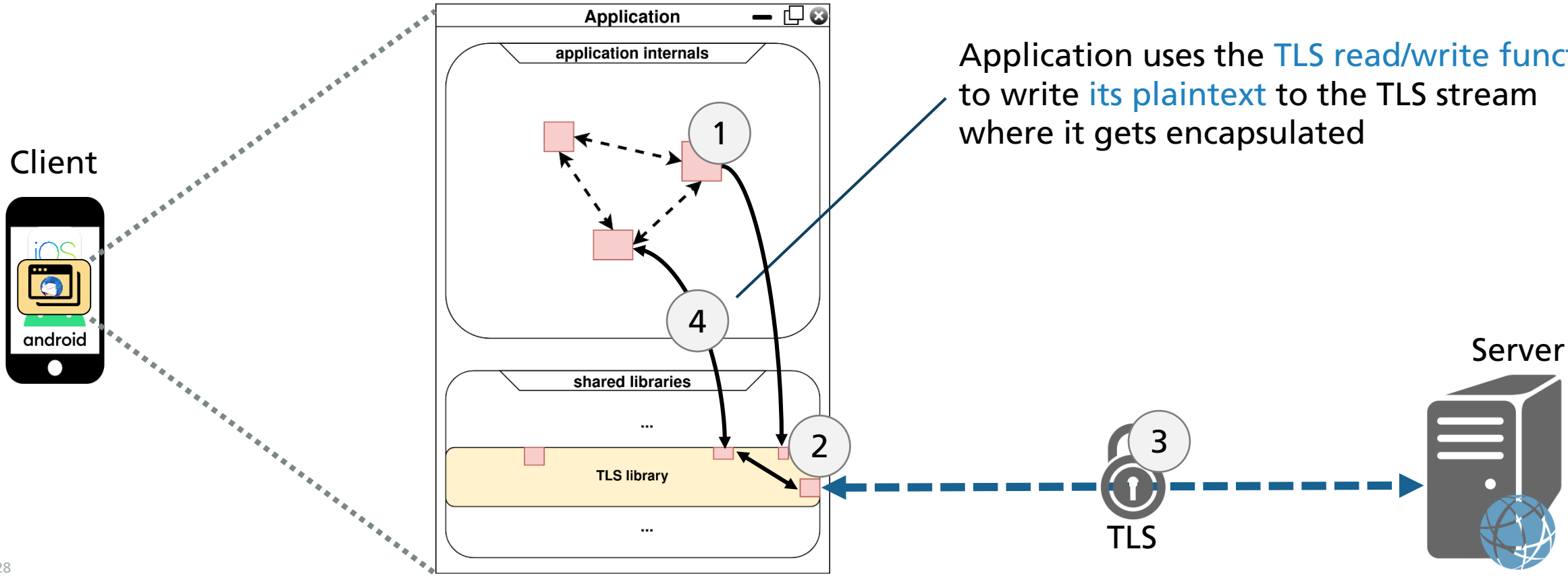
friTap: Approach



Fraunhofer FKIE

friTap: Approach

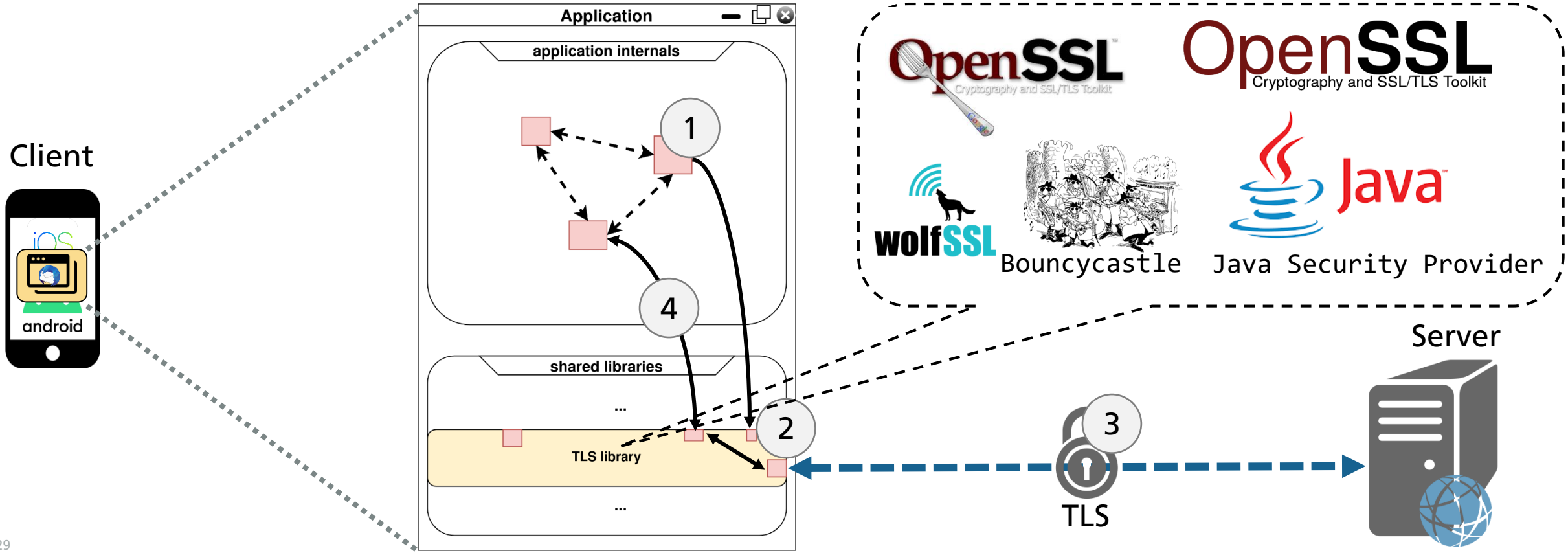
Application uses the TLS read/write functions to read and write the plaintext from and to the TLS stream



Fraunhofer FKIE

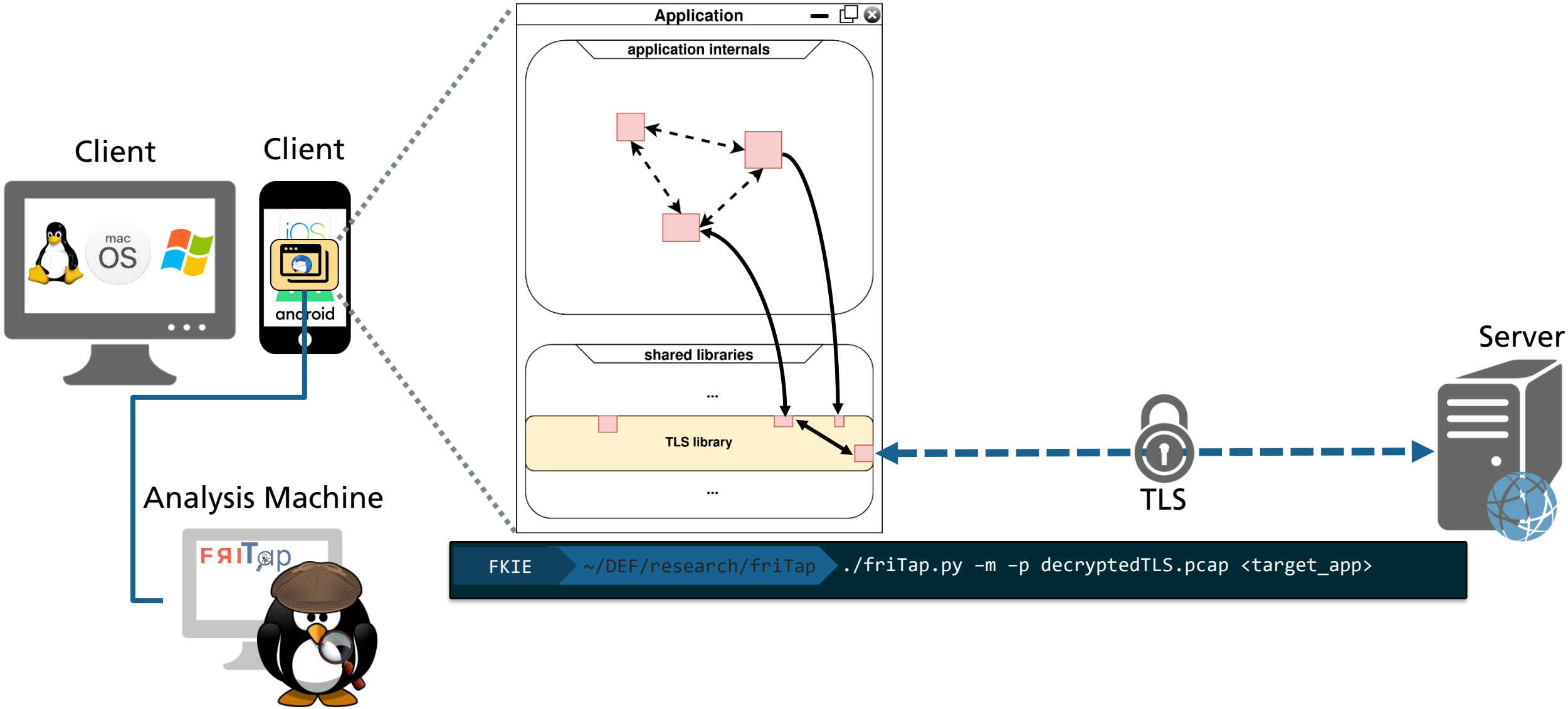
friTap: Approach

Application uses the TLS read/write functions to read and write the plaintext from and to the TLS stream and friTap is hooking them



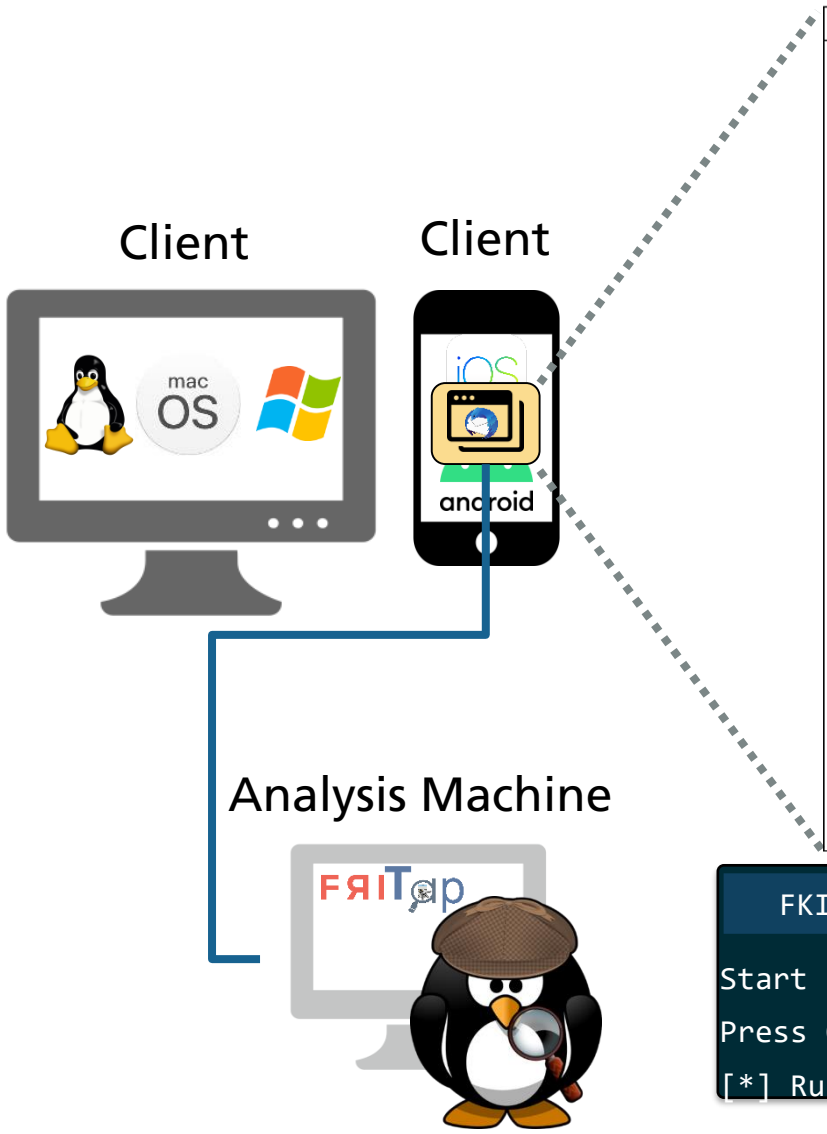
Fraunhofer FKIE

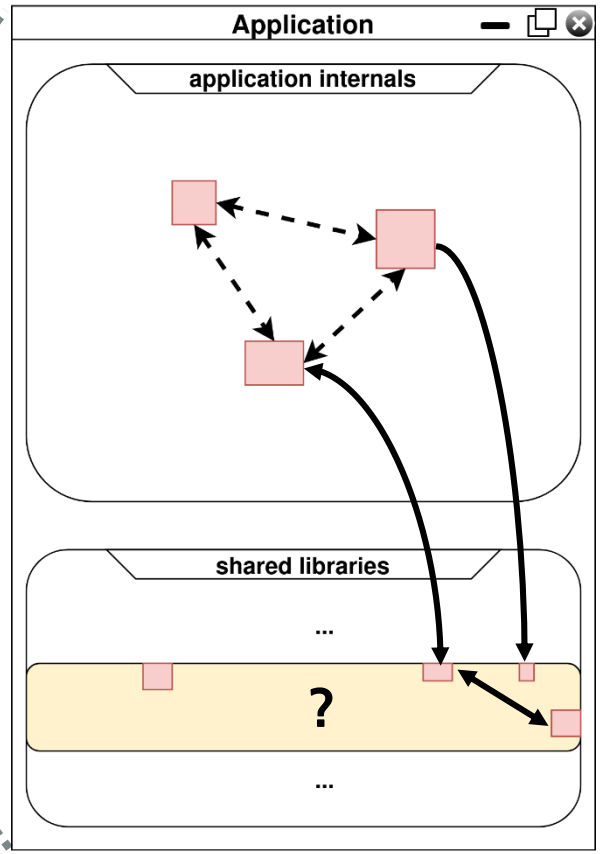
friTap: Decrypted TLS payload as PCAP



Fraunhofer FKIE

friTap: Decrypted TLS payload as PCAP



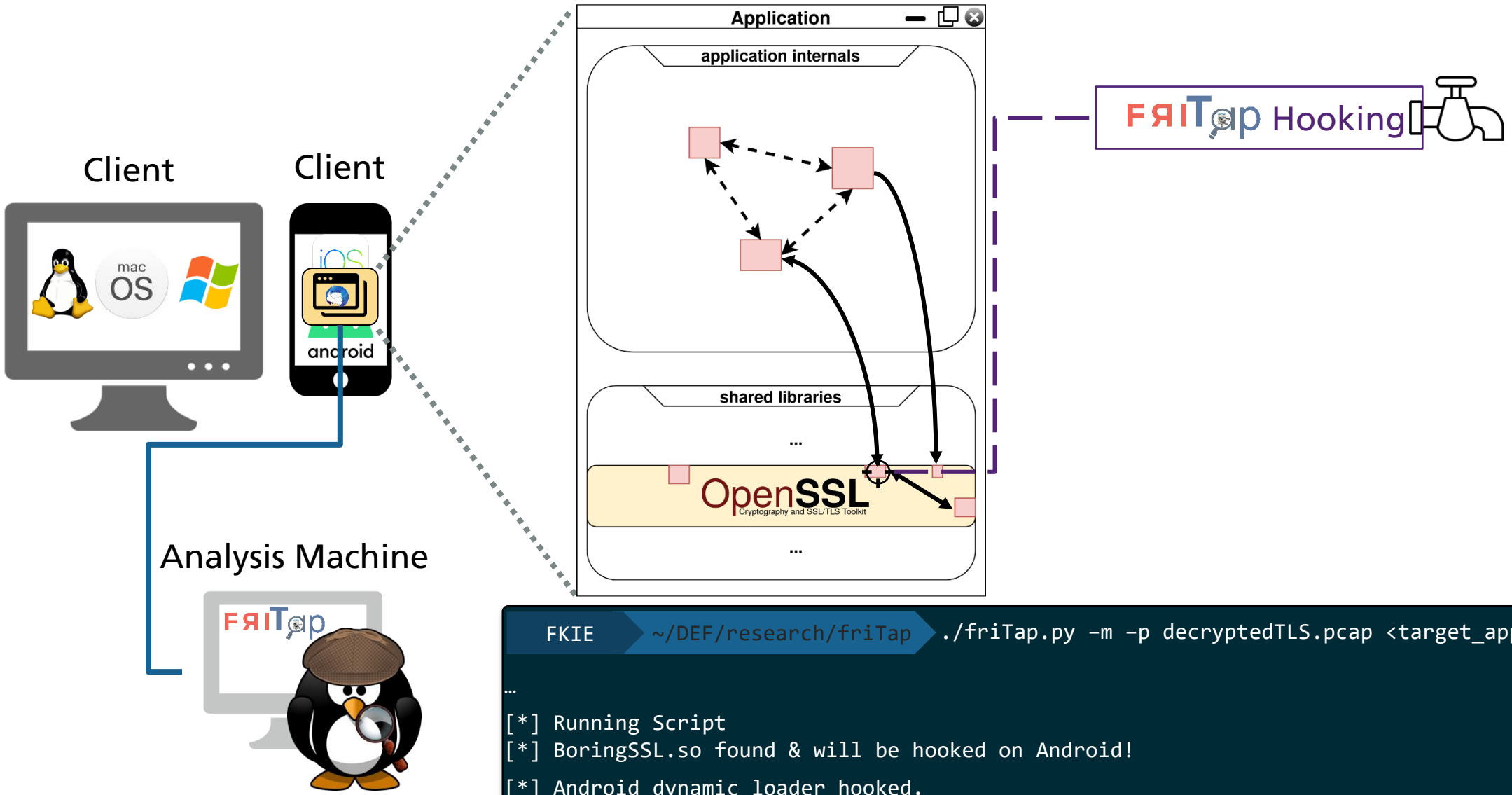


FRITap Hooking

```
FKIE ~/DEF/research/friTap ./friTap.py -m -p decryptedTLS.pcap <target_app>
Start logging
Press Ctrl+C to stop logging.
[*] Running Script
```

Fraunhofer FKIE

friTap: Decrypted TLS payload as PCAP

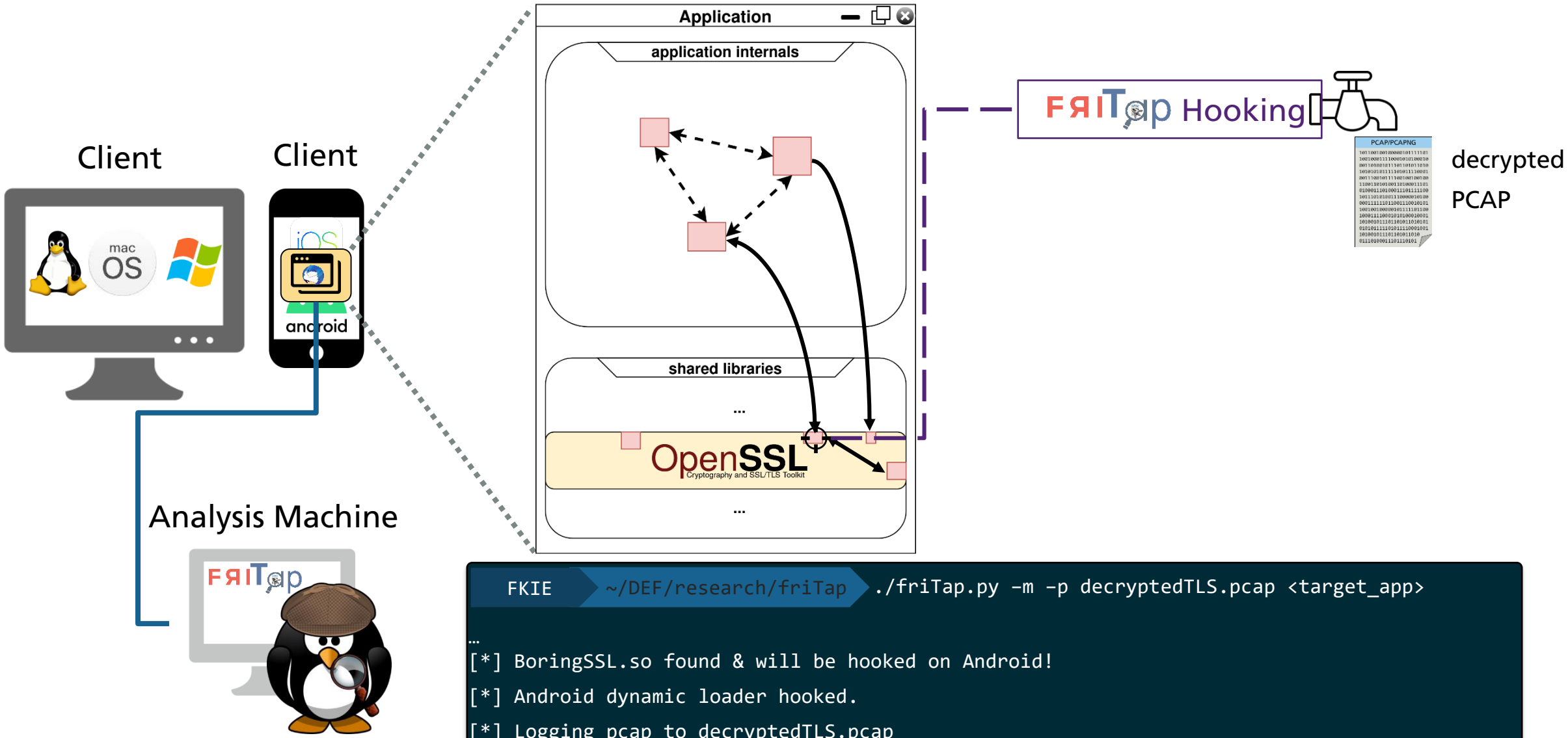



```
FKIE ~/DEF/research/friTap ./friTap.py -m -p decryptedTLS.pcap <target_app>

…
[*] Running Script
[*] BoringSSL.so found & will be hooked on Android!
[*] Android dynamic loader hooked.
```

Fraunhofer FKIE

friTap: Decrypted TLS payload as PCAP

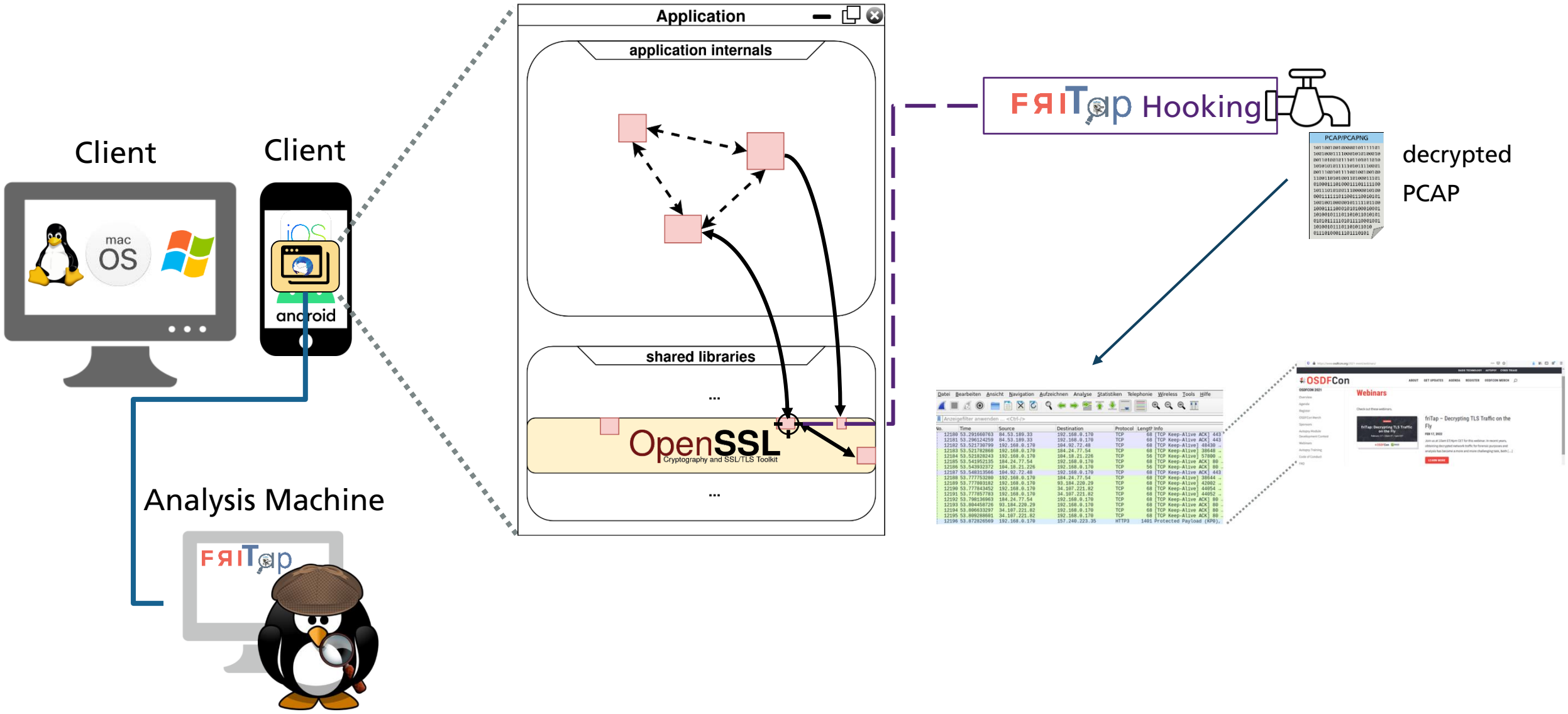



```
FKIE ~/DEF/research/friTap ./friTap.py -m -p decryptedTLS.pcap <target_app>

…
[*] BoringSSL.so found & will be hooked on Android!
[*] Android dynamic loader hooked.
[*] Logging pcap to decryptedTLS.pcap
```

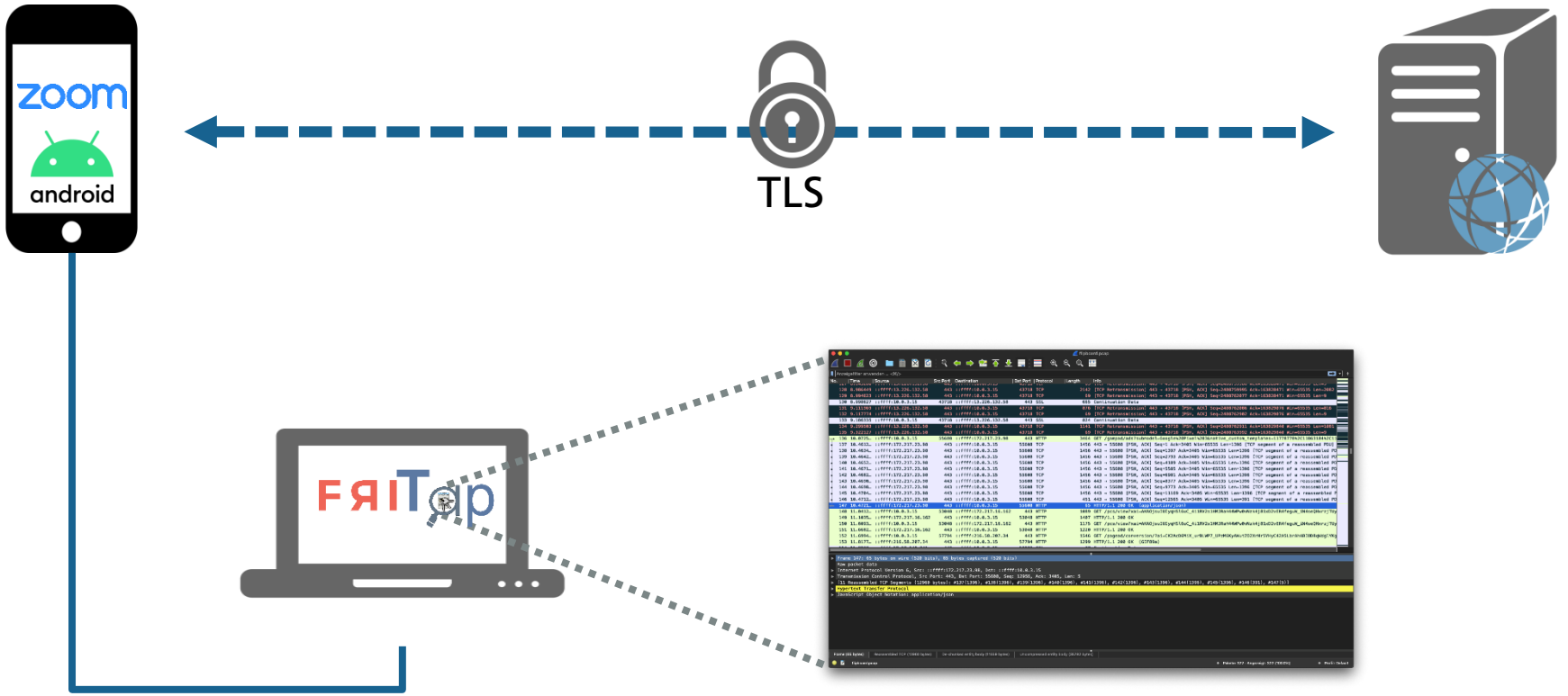
Fraunhofer FKIE

friTap: Decrypted TLS payload as PCAP



Fraunhofer FKIE

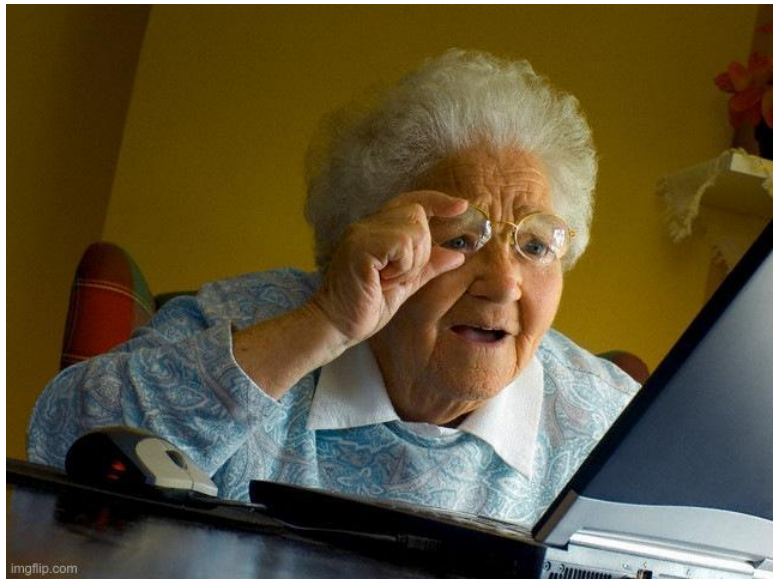
Demo: *zoom



*We aren't showing any vulnerabilities; These techniques are only possible when we have full control over the device

Fraunhofer FKIE

friTap: What else

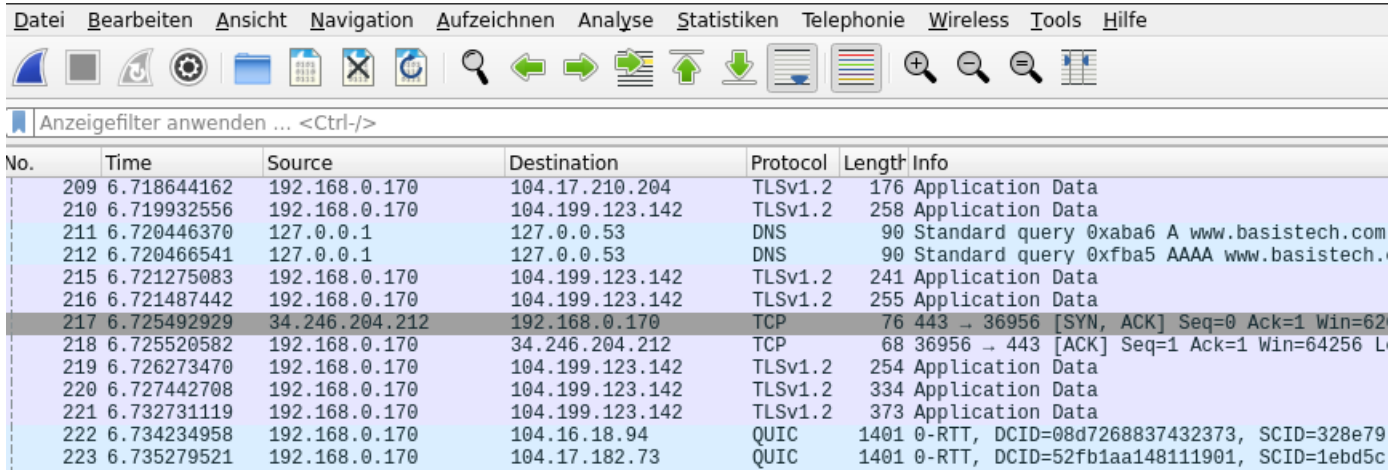


Hooking functions of TLS libraries to extract TLS keys

THE SSLKEYLOG-file can also be logged. SSL libraries offers ways to write the used keys into a file which can later on be used by Wireshark to decrypt the captured traffic.

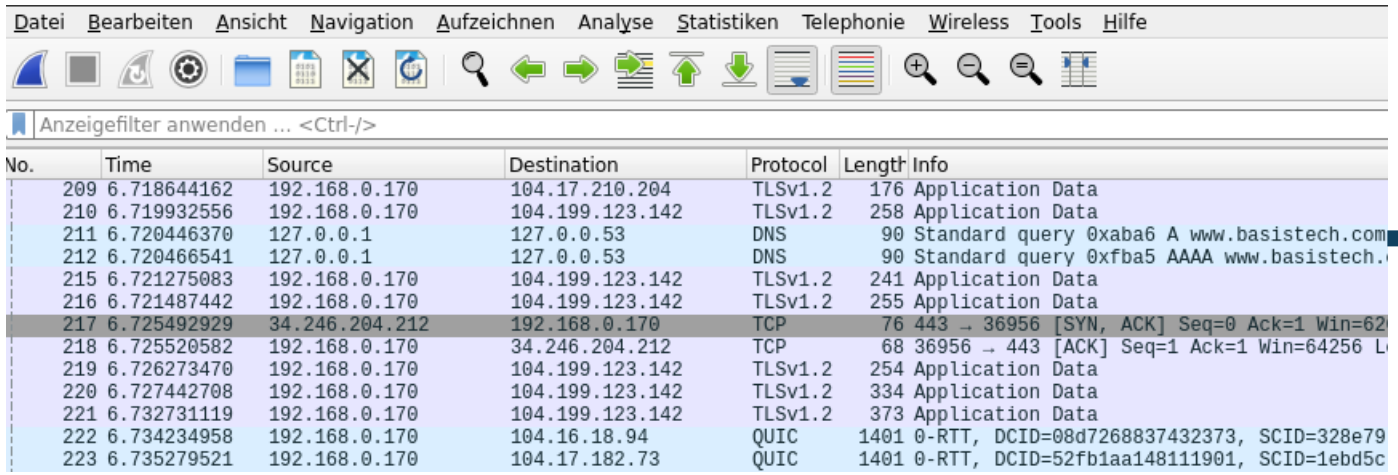
Fraunhofer FKIE

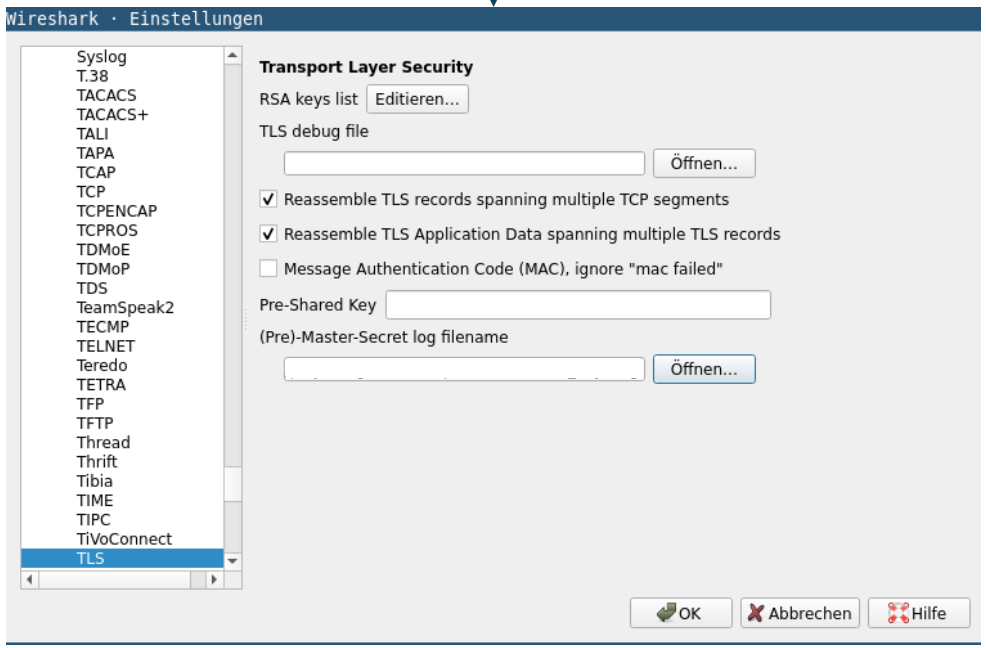
friTap: What else



Fraunhofer FKIE

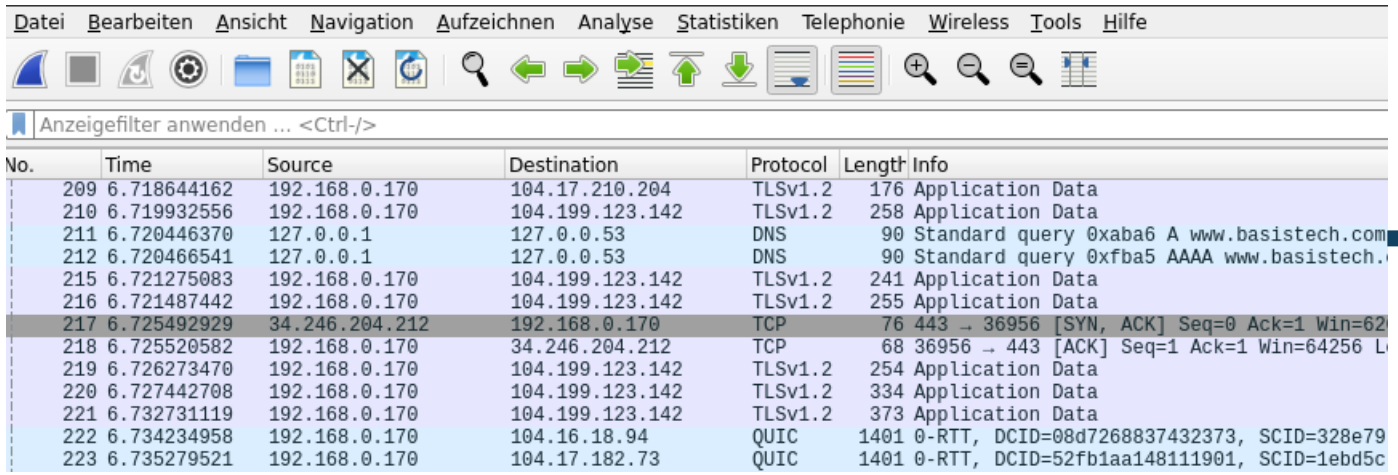
friTap: What else

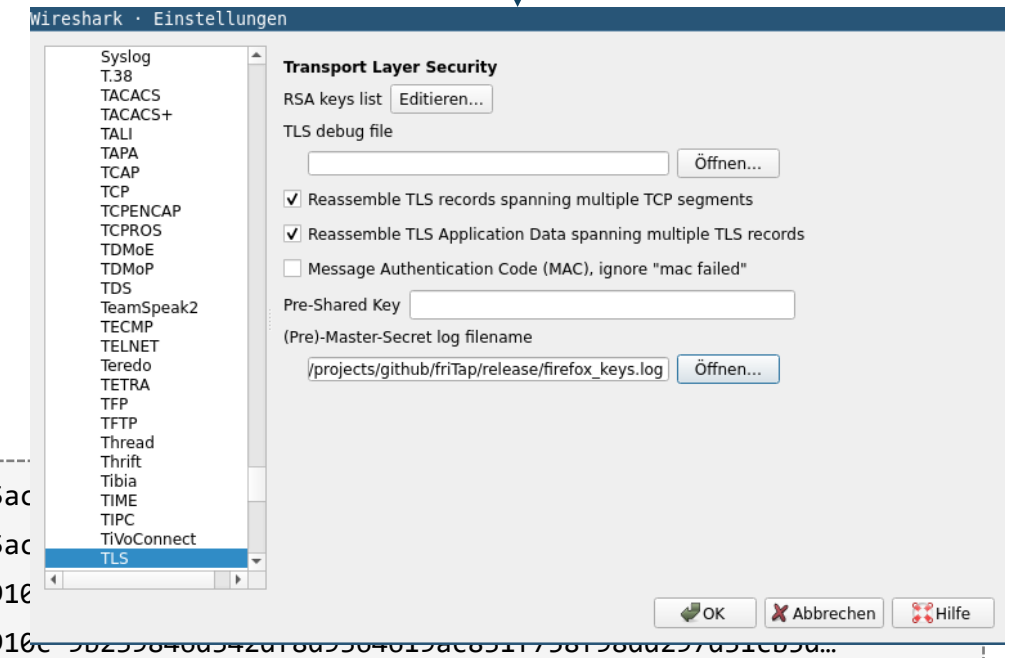




Fraunhofer FKIE

friTap: What else



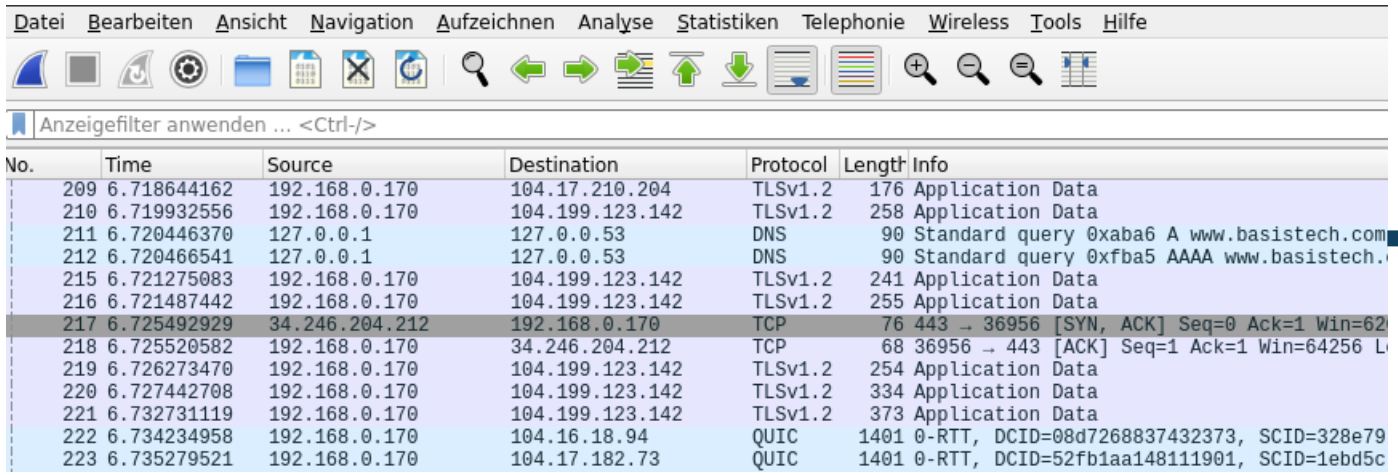


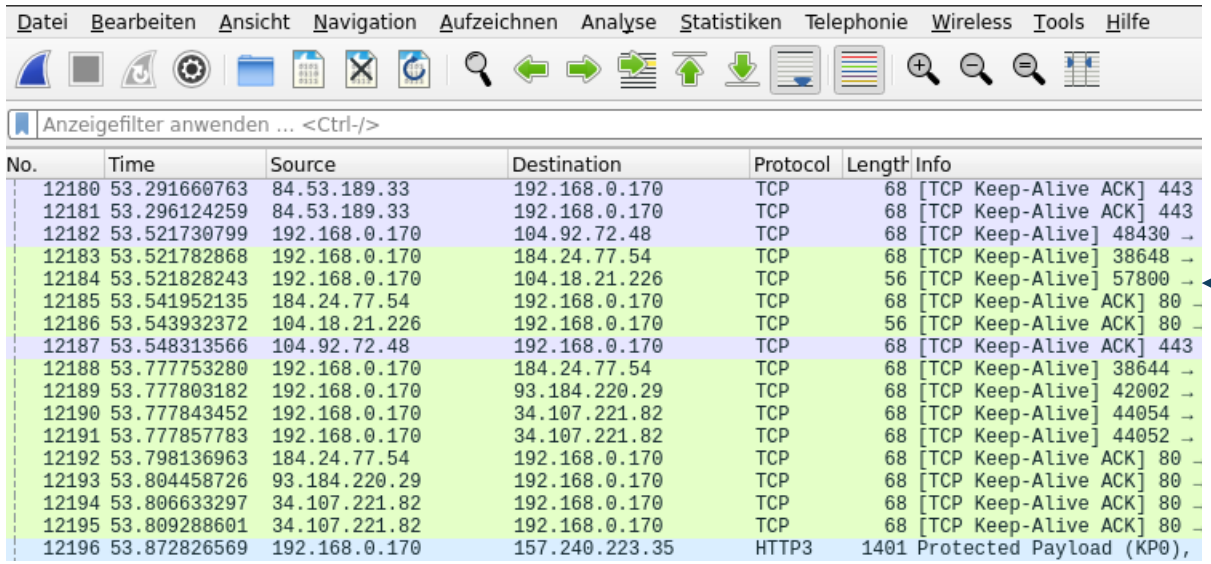
SSLKEYLOG file extracted by friTap

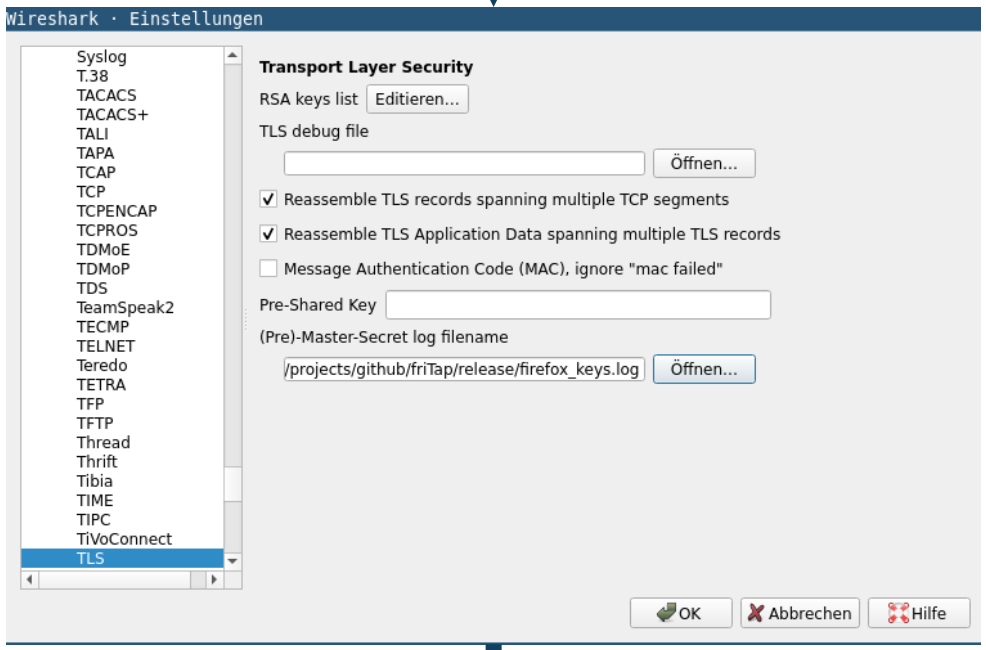
```
CLIENT_HANDSHAKE_TRAFFIC_SECRET 679e8c731884f1c36200e338816b52f86be6a3699ed83805b3445ac
SERVER_HANDSHAKE_TRAFFIC_SECRET 679e8c731884f1c36200e338816b52f86be6a3699ed83805b3445ac
CLIENT_TRAFFIC_SECRET_0 679e8c731884f1c36200e338816b52f86be6a3699ed83805b3445acd0707910
SERVER_TRAFFIC_SECRET_0 679e8c731884f1c36200e338816b52f86be6a3699ed83805b3445acd0707910
EXPORTER_SECRET 679e8c731884f1c36200e338816b52f86be6a3699ed83805b3445acd0707910c 658e36ef804d07017d798d67cfa6157c168604892e66a1a0d640501…
CLIENT_RANDOM 55c984ac6aacdb1af0873a9eb891a88babf2fe1f35b50eb7e93a1515f9db0b36 6a7ecc21ea11f8cf4a7954def08b965bb53903d37e9b95a9ee6dbd271…
…
```

Fraunhofer FKIE

friTap: What else

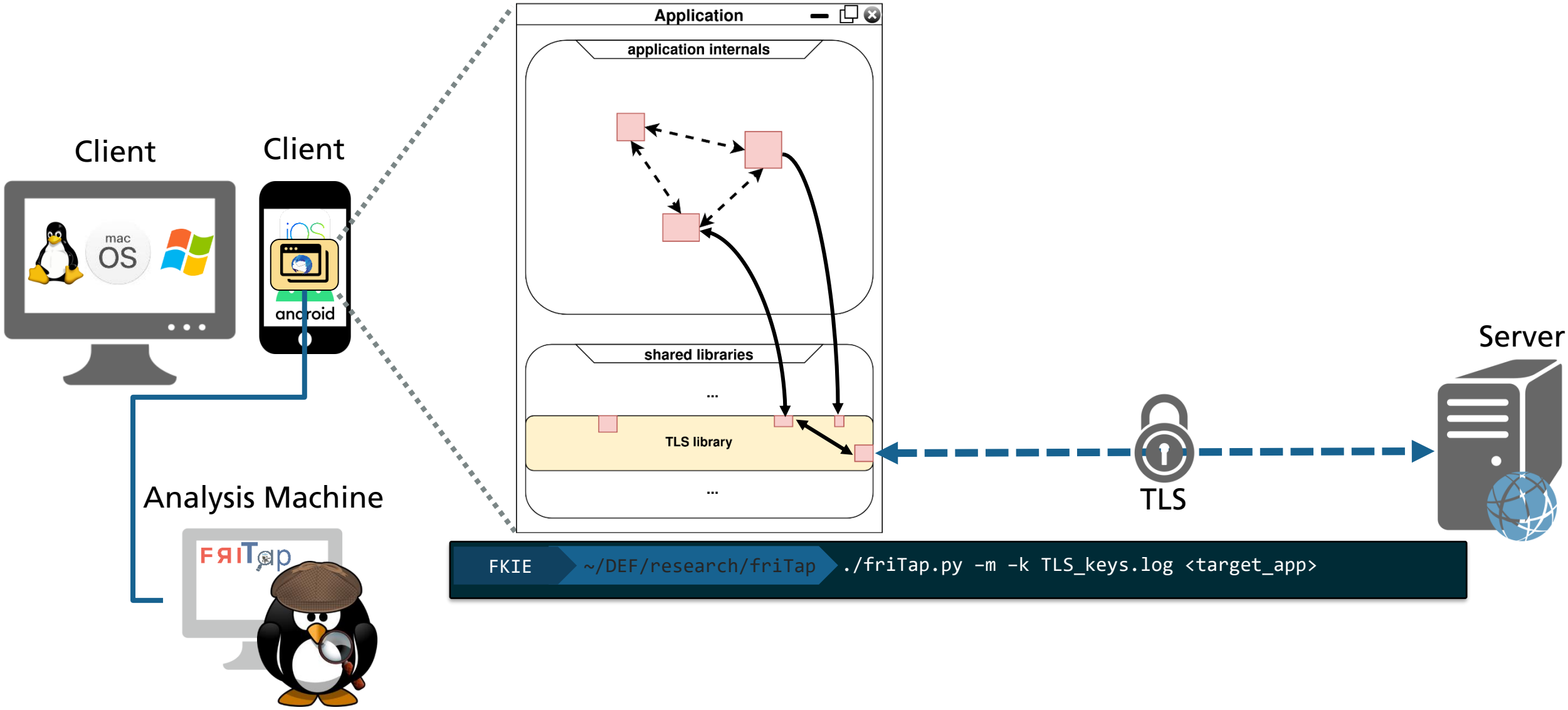






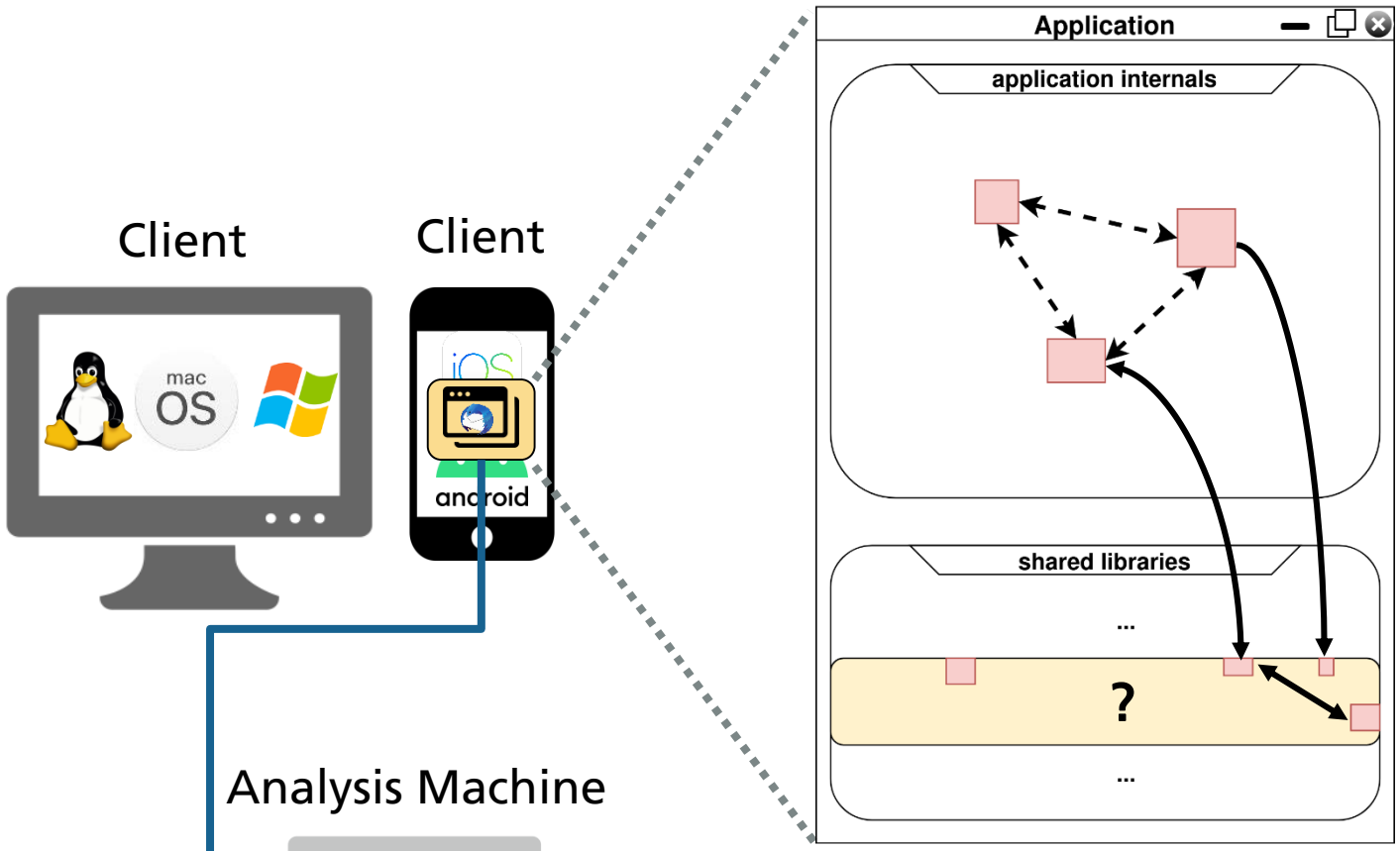
Fraunhofer FKIE

friTap: TLS key extraction



Fraunhofer FKIE

friTap: TLS key extraction

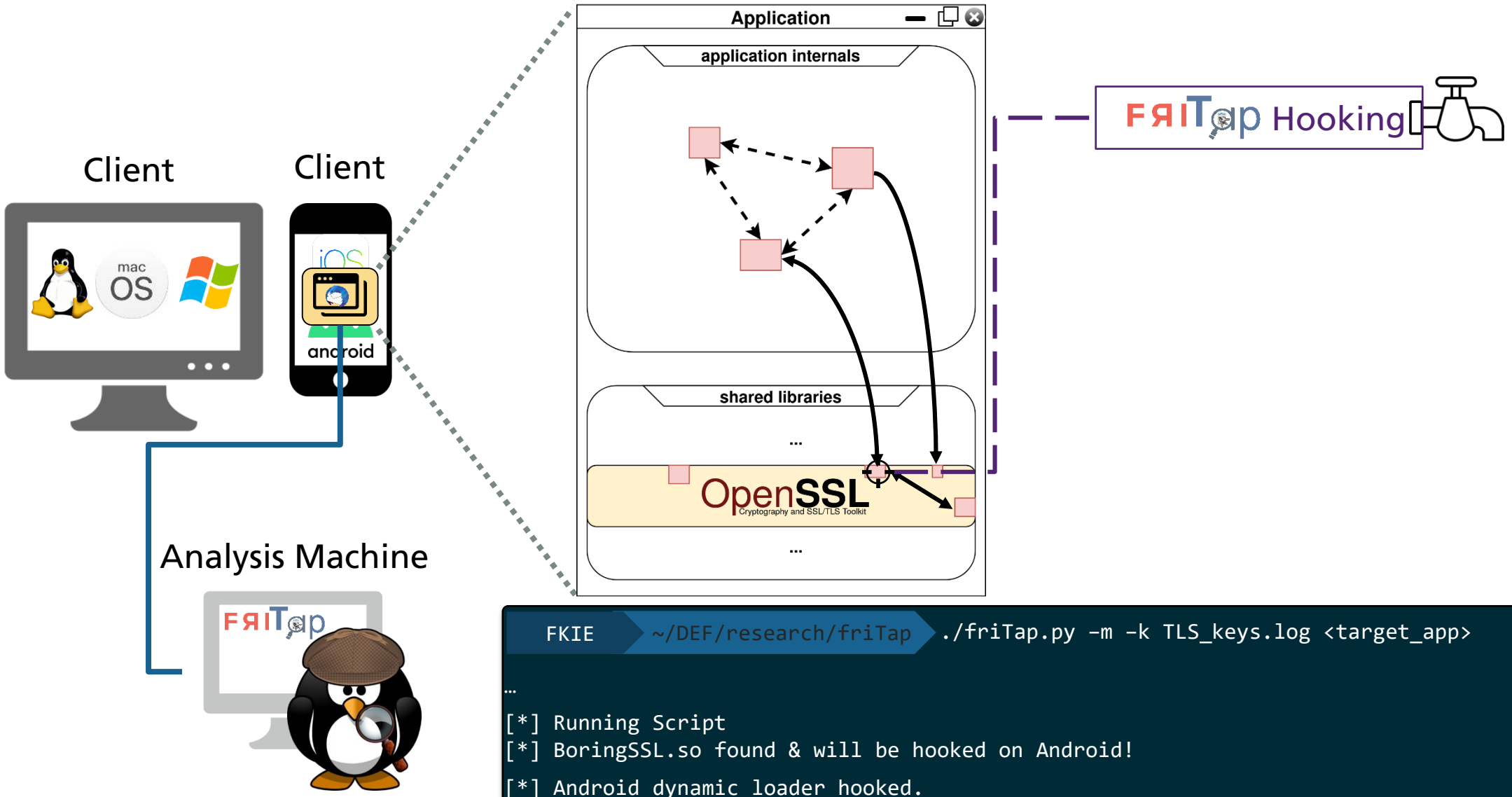


FRITAP Hooking

```
FKIE ~/DEF/research/friTap ./friTap.py -m -k TLS_keys.log <target_app>
Start logging
Press Ctrl+C to stop logging.
[*] Running Script
```

Fraunhofer FKIE

friTap: TLS key extraction

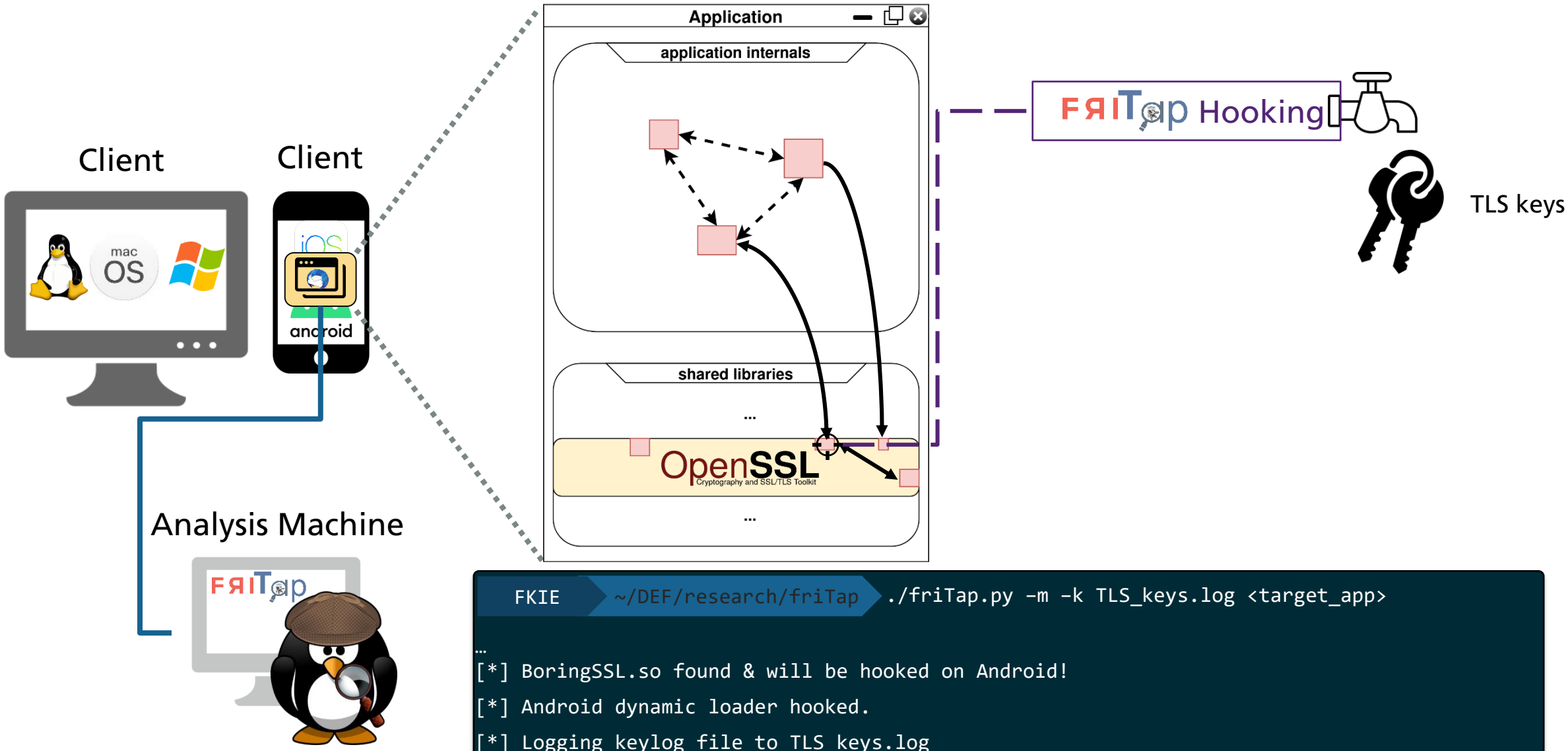



```
FKIE ~/DEF/research/friTap ./friTap.py -m -k TLS_keys.log <target_app>

…
[*] Running Script
[*] BoringSSL.so found & will be hooked on Android!
[*] Android dynamic loader hooked.
```

Fraunhofer FKIE

friTap: TLS key extraction

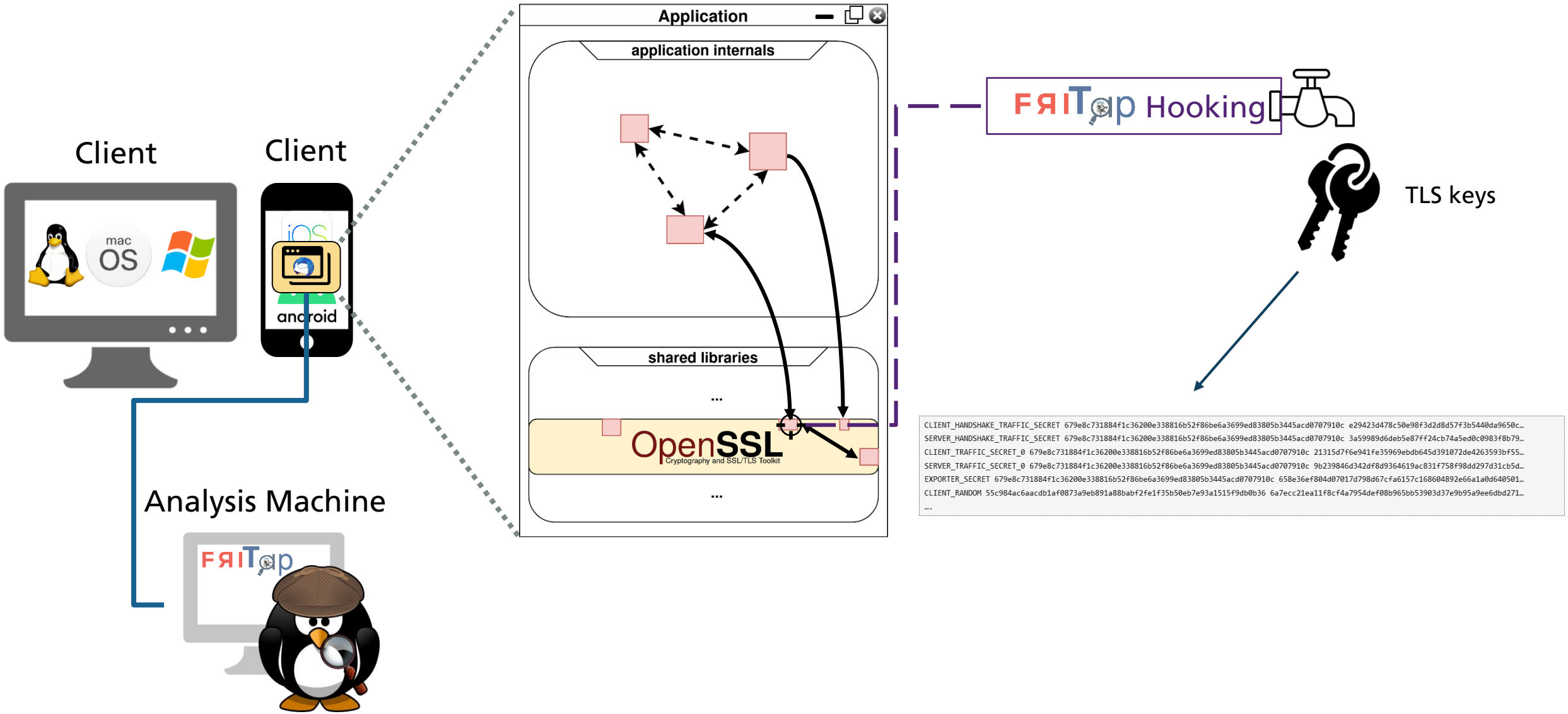



```
FKIE ~/DEF/research/friTap ./friTap.py -m -k TLS_keys.log <target_app>

…
[*] BoringSSL.so found & will be hooked on Android!
[*] Android dynamic loader hooked.
[*] Logging keylog file to TLS_keys.log
```

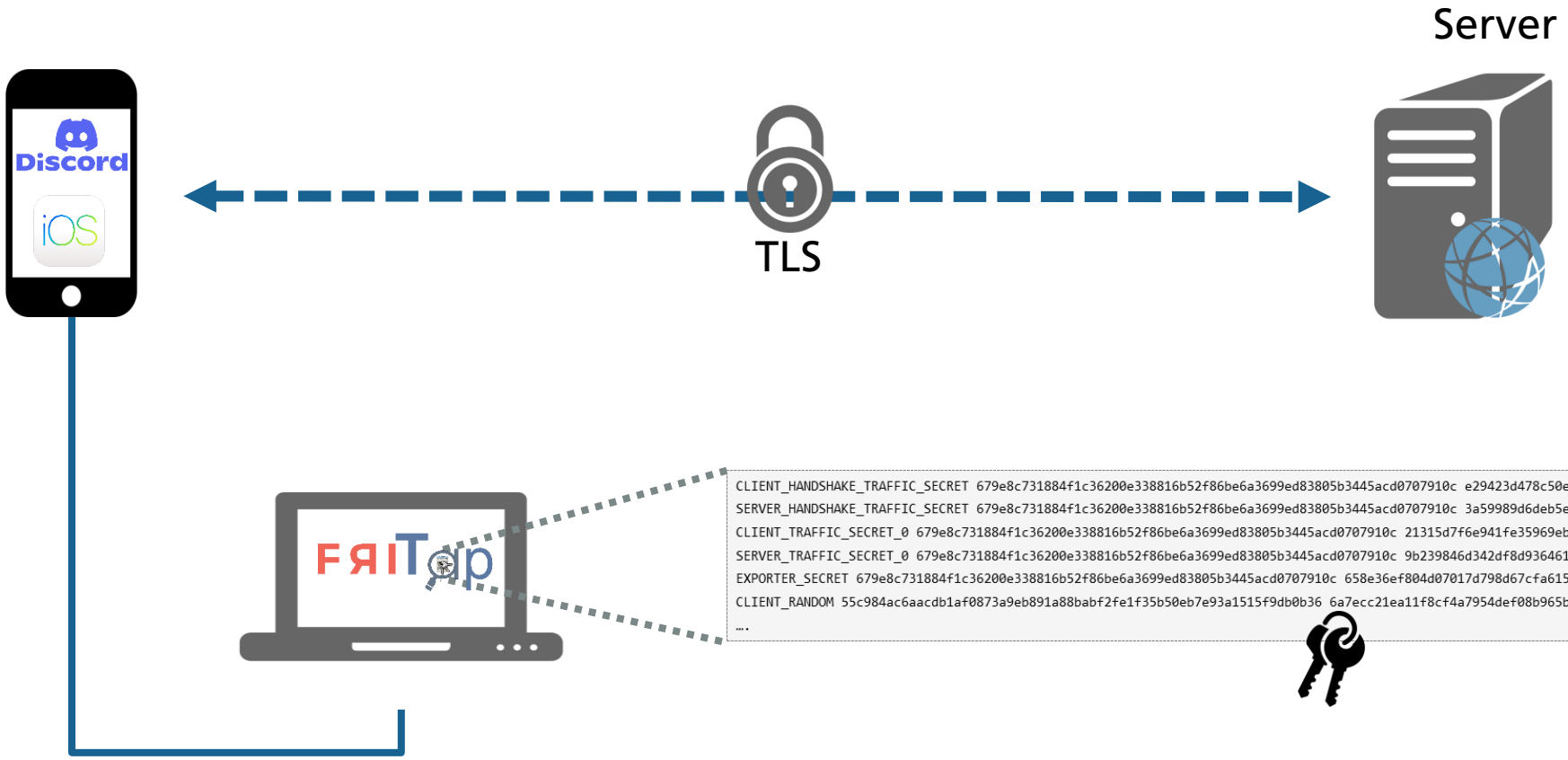
Fraunhofer FKIE

friTap: TLS key extraction



Fraunhofer FKIE

Demo: *Discord

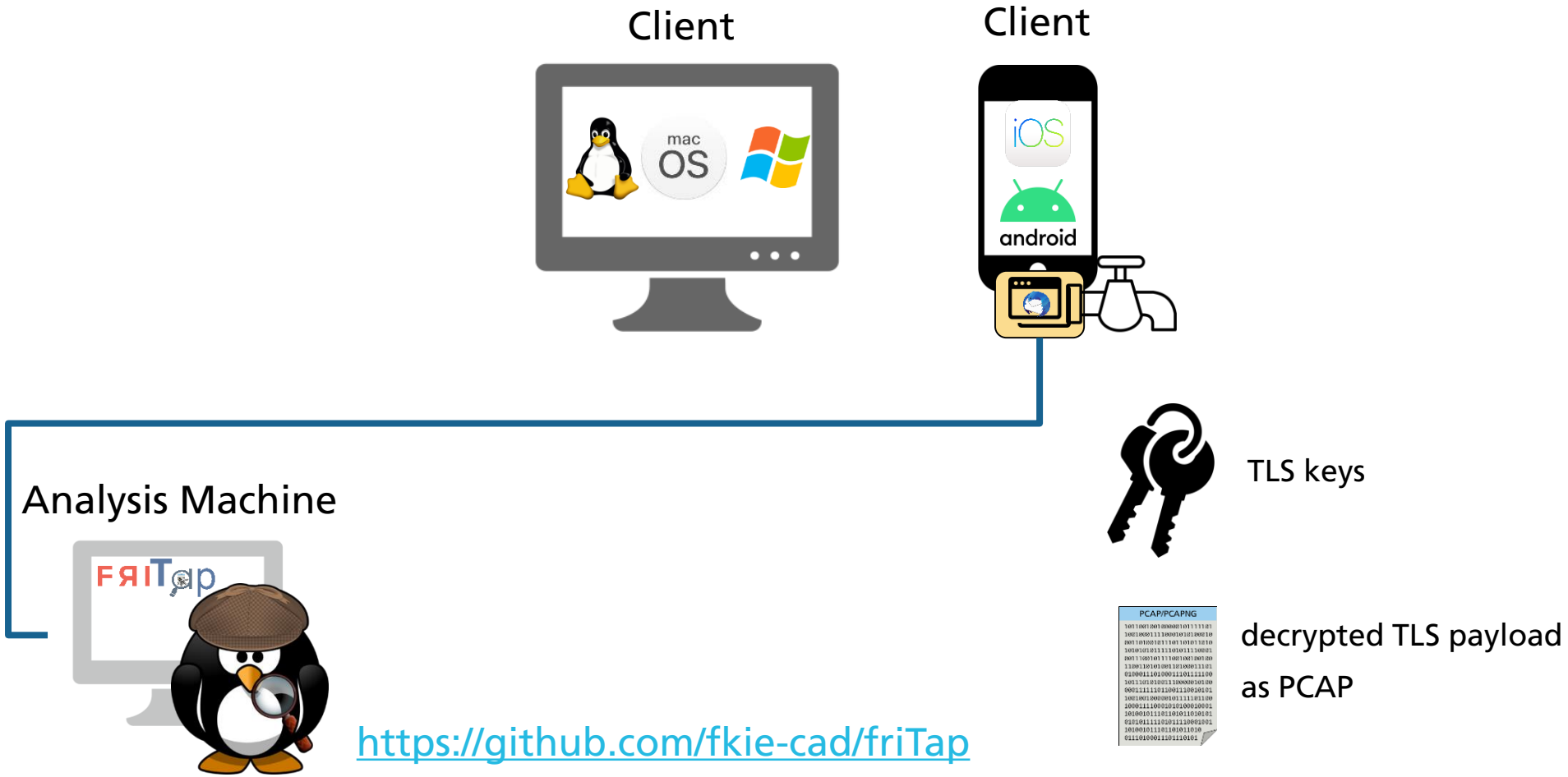


*We aren’t showing any vulnerabilities; These techniques are only possible when we have full control over the device

Fraunhofer FKIE

Conclusion

friTap provides us with TLS keys and the decrypted TLS payload

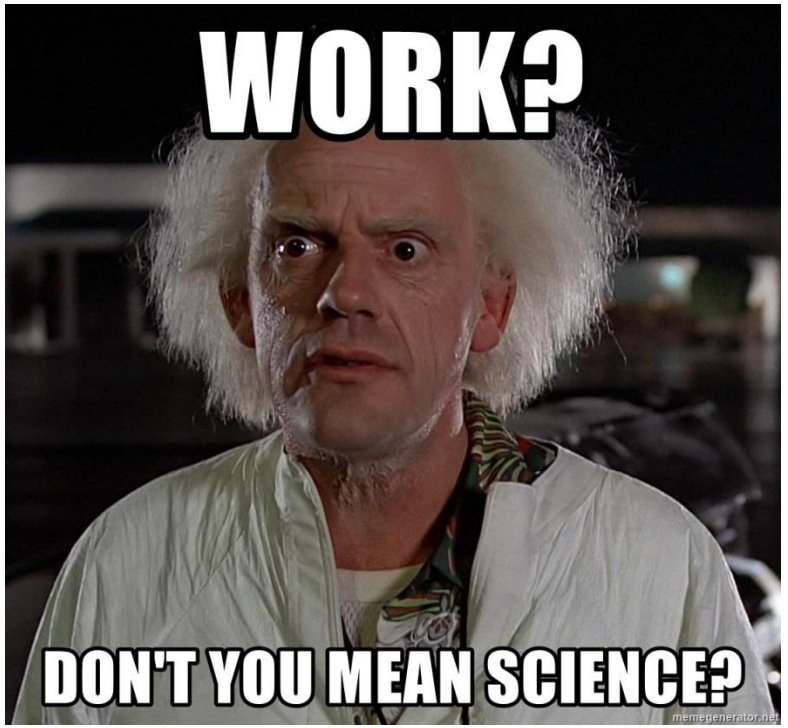


https://github.com/fkie-cad/friTap

Future Work

There is still a lot of work to do….

- Support for other SSL libraries
- Add feature to prototype TLS Read/Write/SSLKEY function
- Capturing all traffic of an application directly with friTap
- Add the capability to alter the decrypted payload
  - integration with mitmproxy
  - integration with Burp Suite
- Working with static linked libraries
- …



Fraunhofer FKIE

Q&A

Thanks for your time!

https://github.com/fkie-cad/friTap