

Daniel Baier*, Martin Lambertz

Secrets in Motion: Dynamic TLS Key Recovery



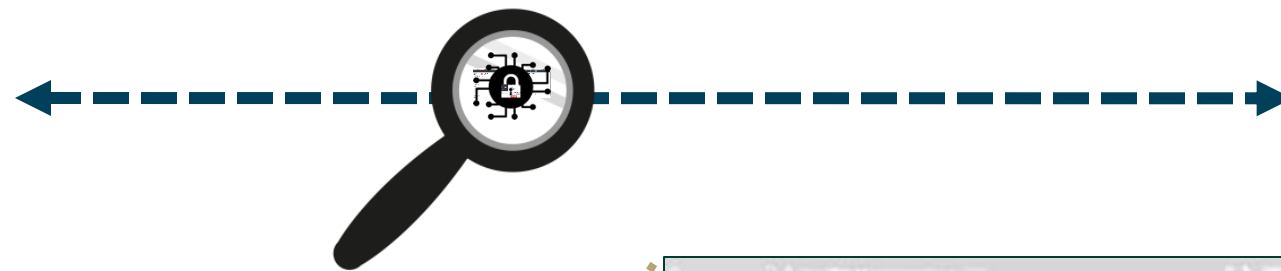
Intro

Traffic analysis today

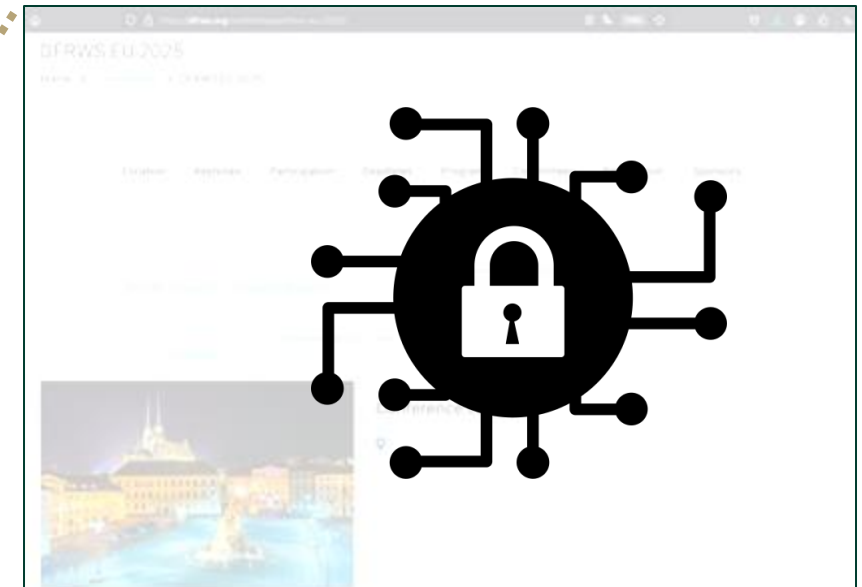
Client



Server

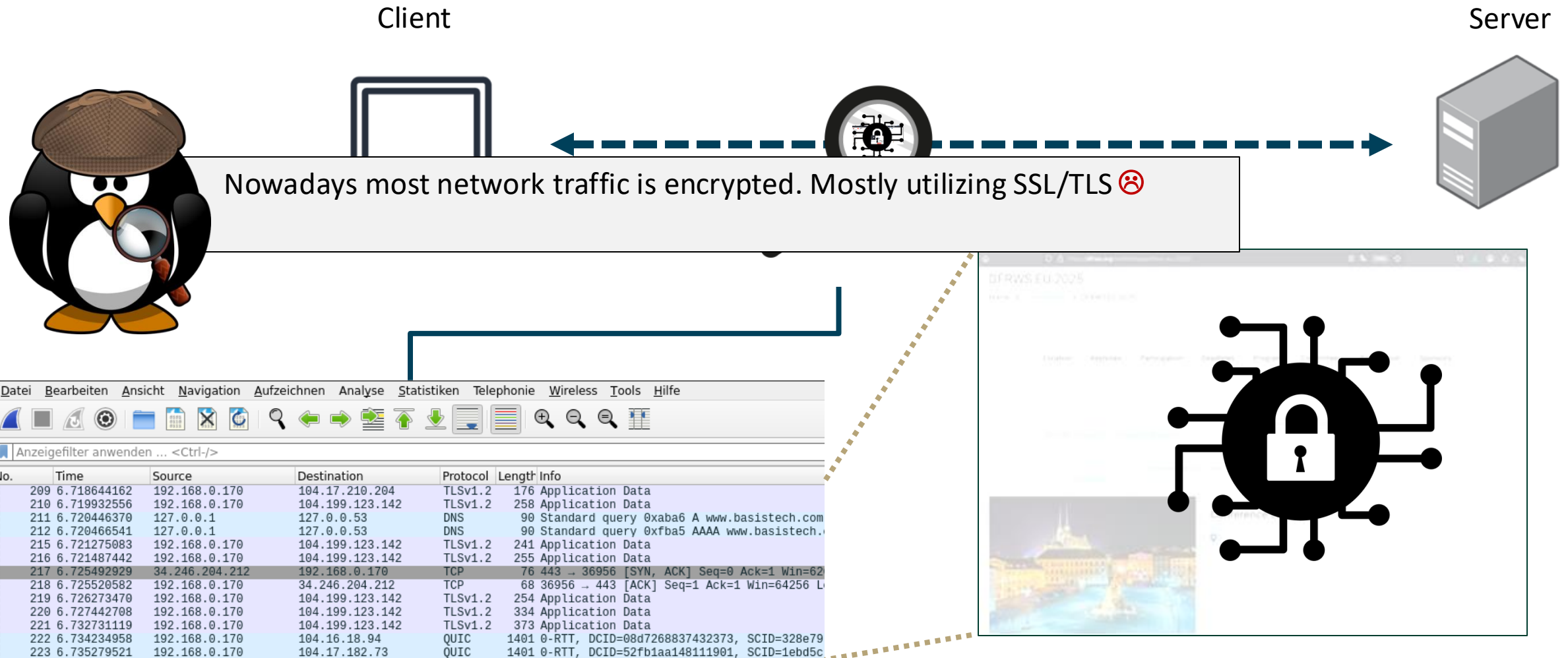


Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telefonie Wireless Tools Hilfe						
Anzeigefilter anwenden ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
209	6.718644162	192.168.0.170	104.17.210.204	TLSv1.2	176	Application Data
210	6.719932556	192.168.0.170	104.199.123.142	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.com
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



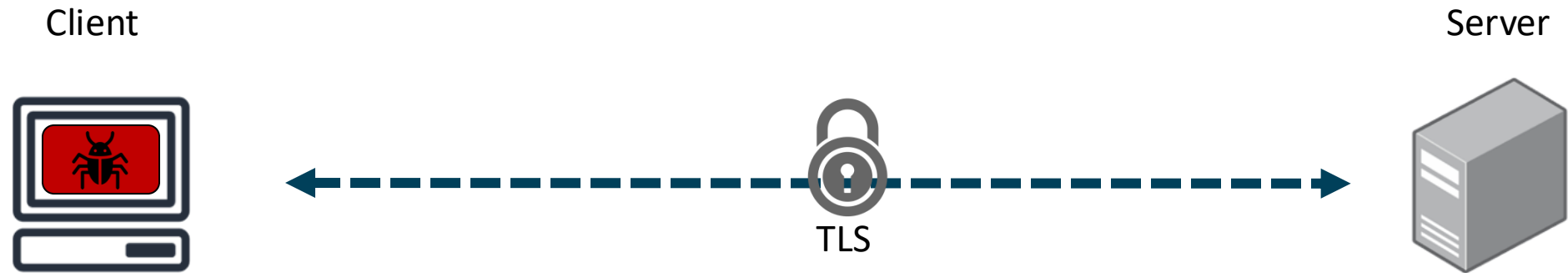
Intro

Traffic analysis today



Intro

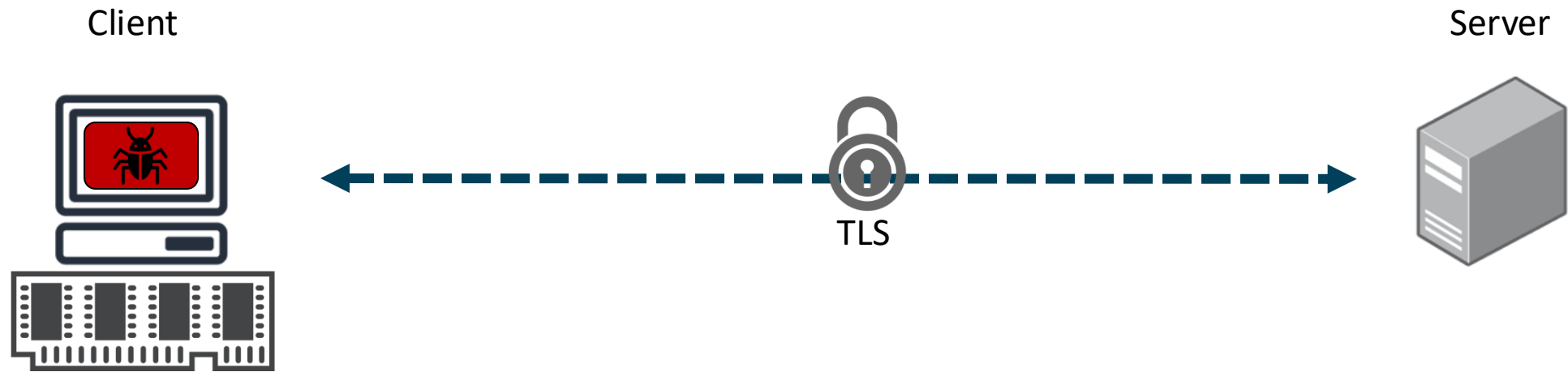
TLS encryption prevents forensic insights



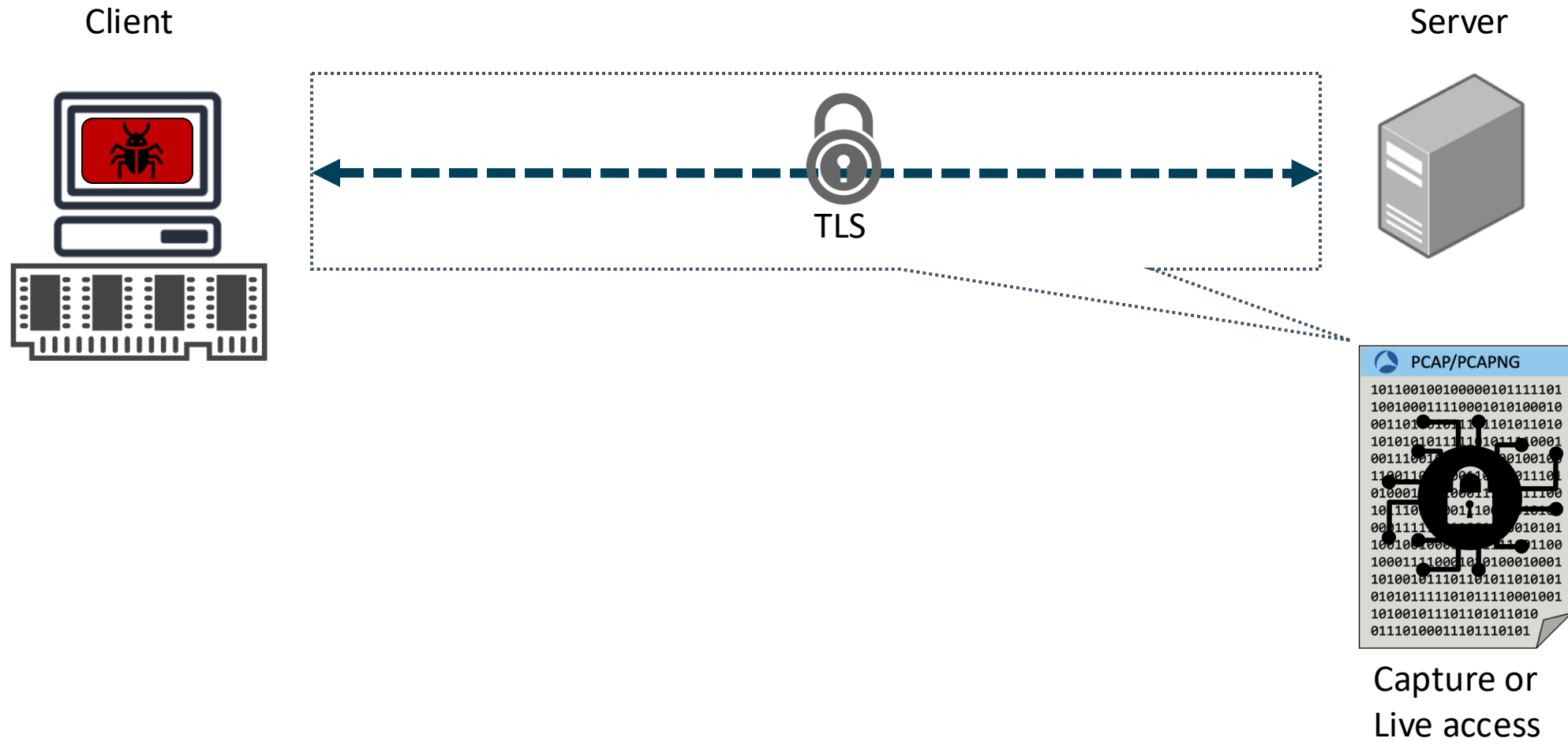
TLS encryption prevents forensic insights:

- ☒ Monitoring
- ☒ Malware analysis
- ☒ API analysis

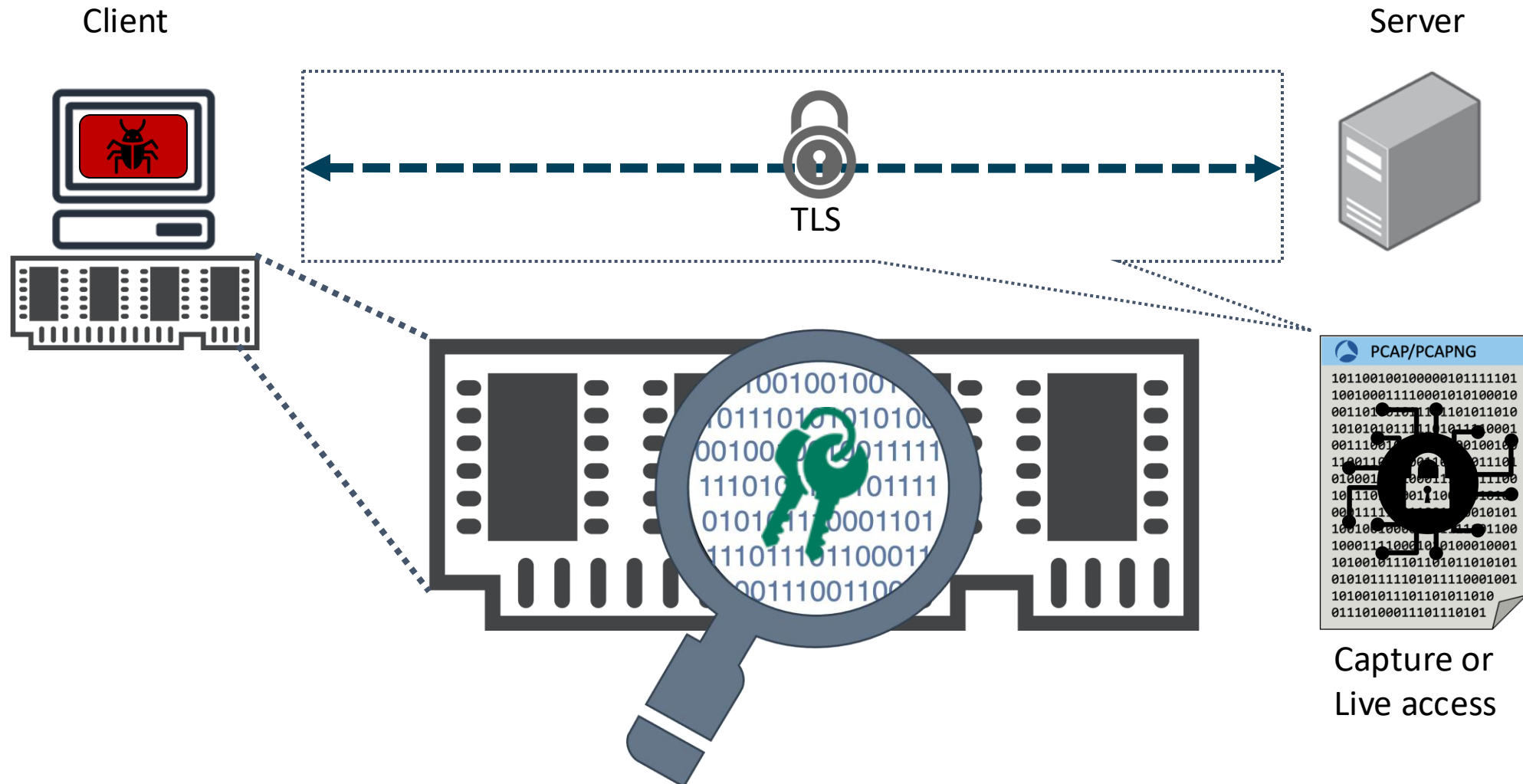
TLS Key Identification



TLS Key Identification



TLS Key Identification





Last year we [systemized the current state of knowledge](#) in the field of [TLS Key Material Identification and Extraction](#) in Memory



READ ME

Intro

Limited TLS library support & minimal TLS 1.3 adoption



Last year we systemized the current state of knowledge in the field of TLS Key Material Identification and Extraction in Memory



- Limited TLS library support ☹️
- TLS 1.3 Key Identification and Extraction mostly unresolved...
- For more details have a look at our paper or the original ones
 - <https://www.sciencedirect.com/science/article/pii/S2666281724000854>



Forensic Science International: Digital Investigation 49 (2024) 301766

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsdi

DFRWS USA 2024 - Selected Papers from the 24th Annual Digital Forensics Research Conference USA

TLS key material identification and extraction in memory: Current state and future challenges

Daniel Baier^{a,*}, Alexander Basse^b, Jan-Niclas Hilgert^a, Martin Lambertz^a

^a Fraunhofer FKIE, Germany
^b Institute of Computer Science, University of Bonn, Germany

ARTICLE INFO

Keywords:
TLS
Transport layer security
Memory forensics
Network forensics
Live forensics
Malware analysis

ABSTRACT

Memory forensics is a crucial part of digital forensics as it can be used to extract valuable information such as running processes, network connections, and encryption keys from memory. The last is especially important when considering the widely used Transport Layer Security (TLS) protocol used to secure internet communication, thus hampering network traffic analysis. Particularly in the context of cybercrime investigations (such as malware analysis), it is therefore paramount for investigators to decrypt TLS traffic. This can provide vital insights into the methods and strategies employed by attackers. For this purpose, it is first and foremost necessary to identify and extract the corresponding TLS key material in memory.

In this paper, we systematize and evaluate the current state of techniques, tools, and methodologies for identifying and extracting TLS key material in memory. We consider solutions from academia but also identify innovative and promising approaches used "in the wild" that are not considered by the academic literature. Furthermore, we identify the open research challenges and opportunities for future research in this domain. Our work provides a profound foundation for future research in this crucial area.

1. Introduction

However, this approach poses problems, such as certificate invalidation and detectability (Jarmoc and Unit, 2012). In the realm of malware

Intro

Limited TLS library support & minimal TLS 1.3 adoption

We are unable to extract the TLS key materials



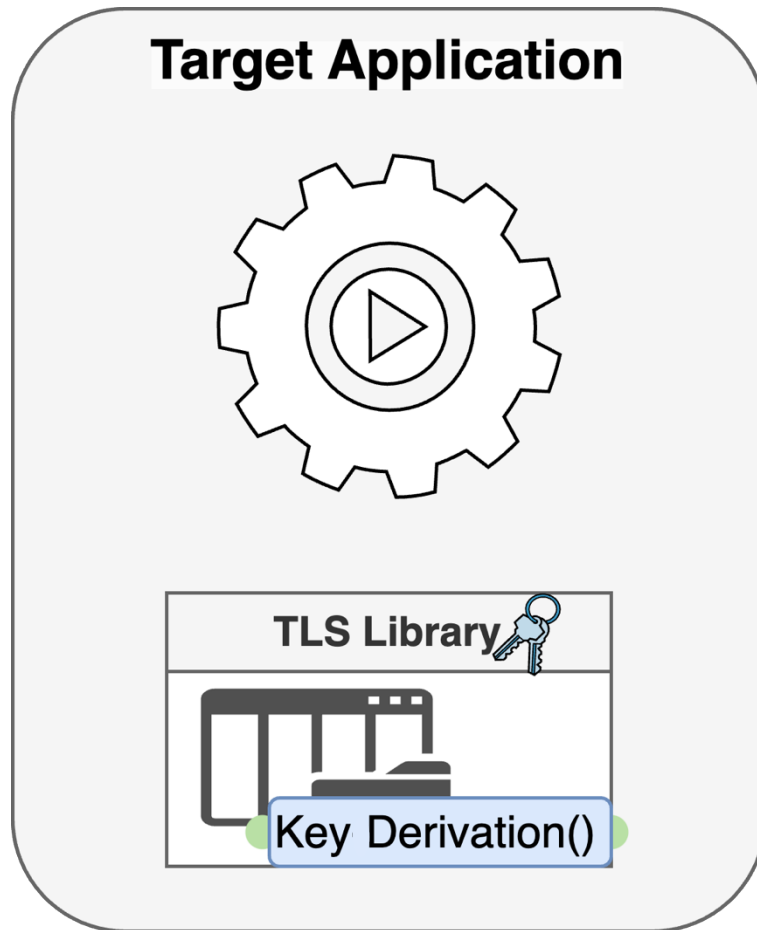
...



What's the Best Way to Extract TLS Keys from Live Memory?

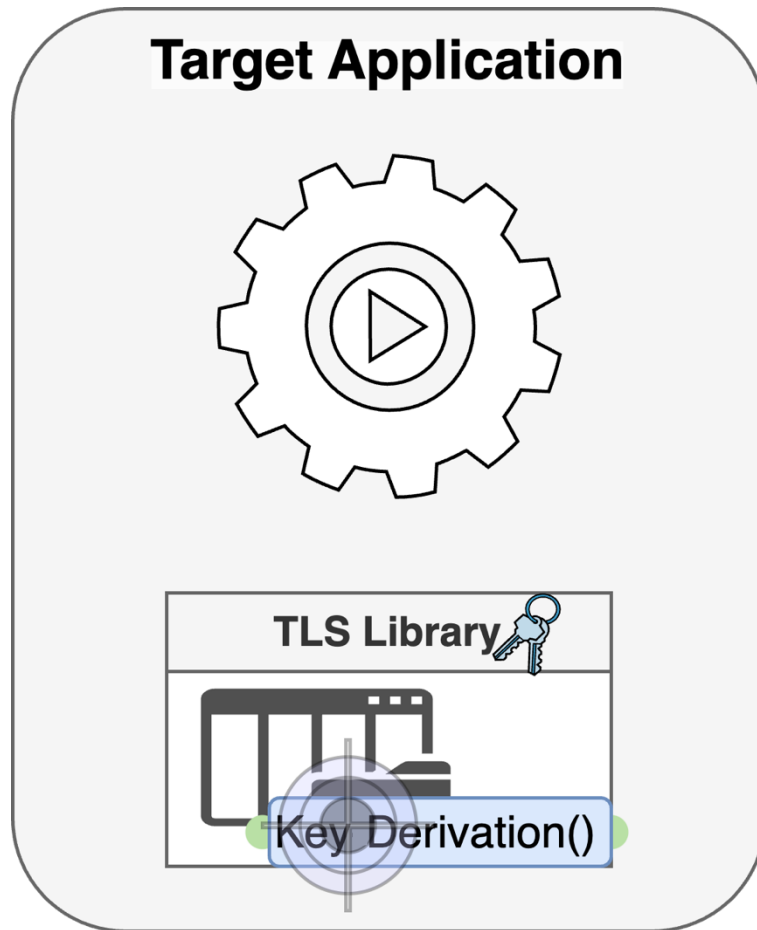
Intercepting the Key Derivation Function

Idea



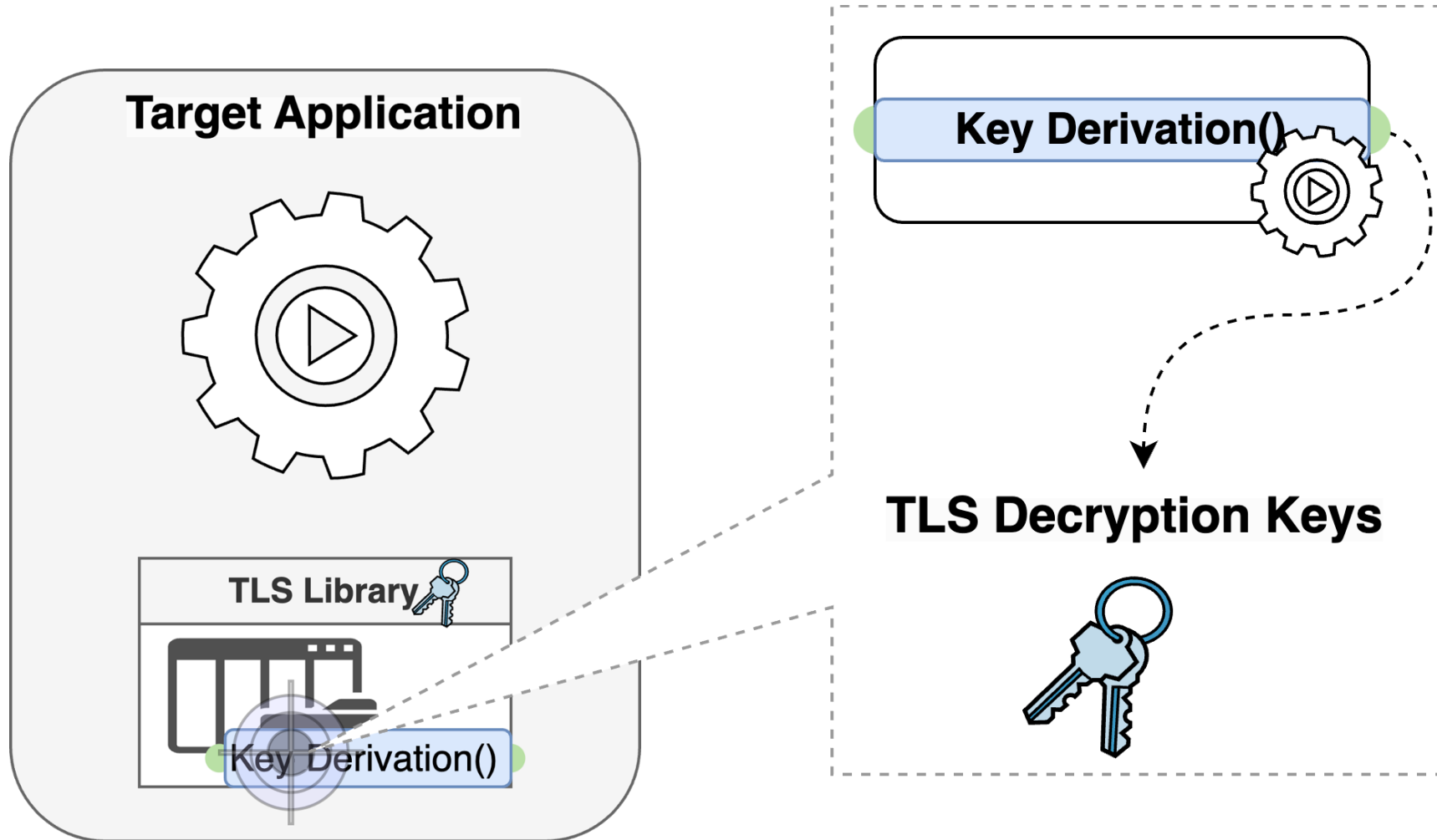
Intercepting the Key Derivation Function

Idea



Intercepting the Key Derivation Function

Idea

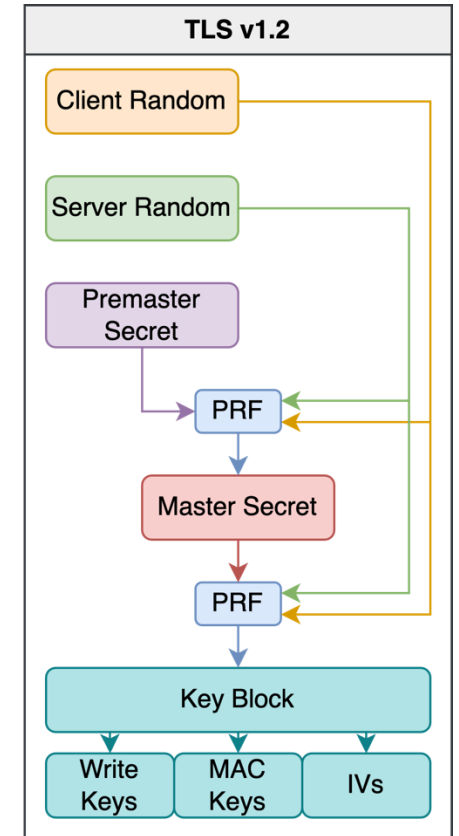




Understanding the TLS Key Schedule

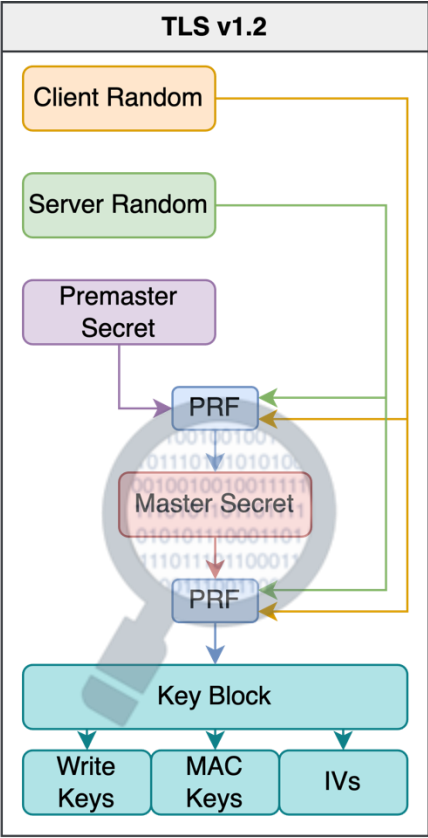
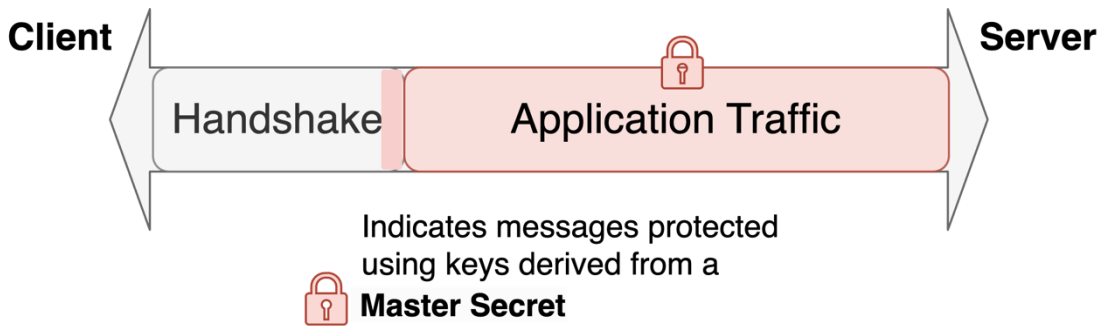
Secrets in Motion: Dynamic TLS Key Recovery Fundamentals

Decryption Keys $\leq$ TLS 1.2



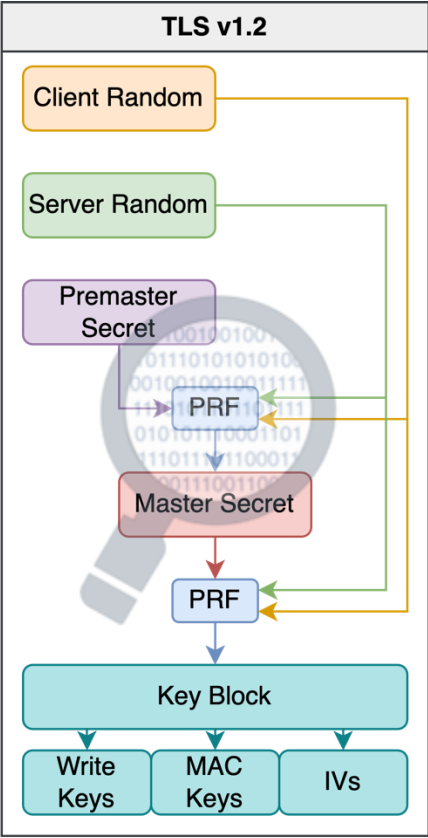
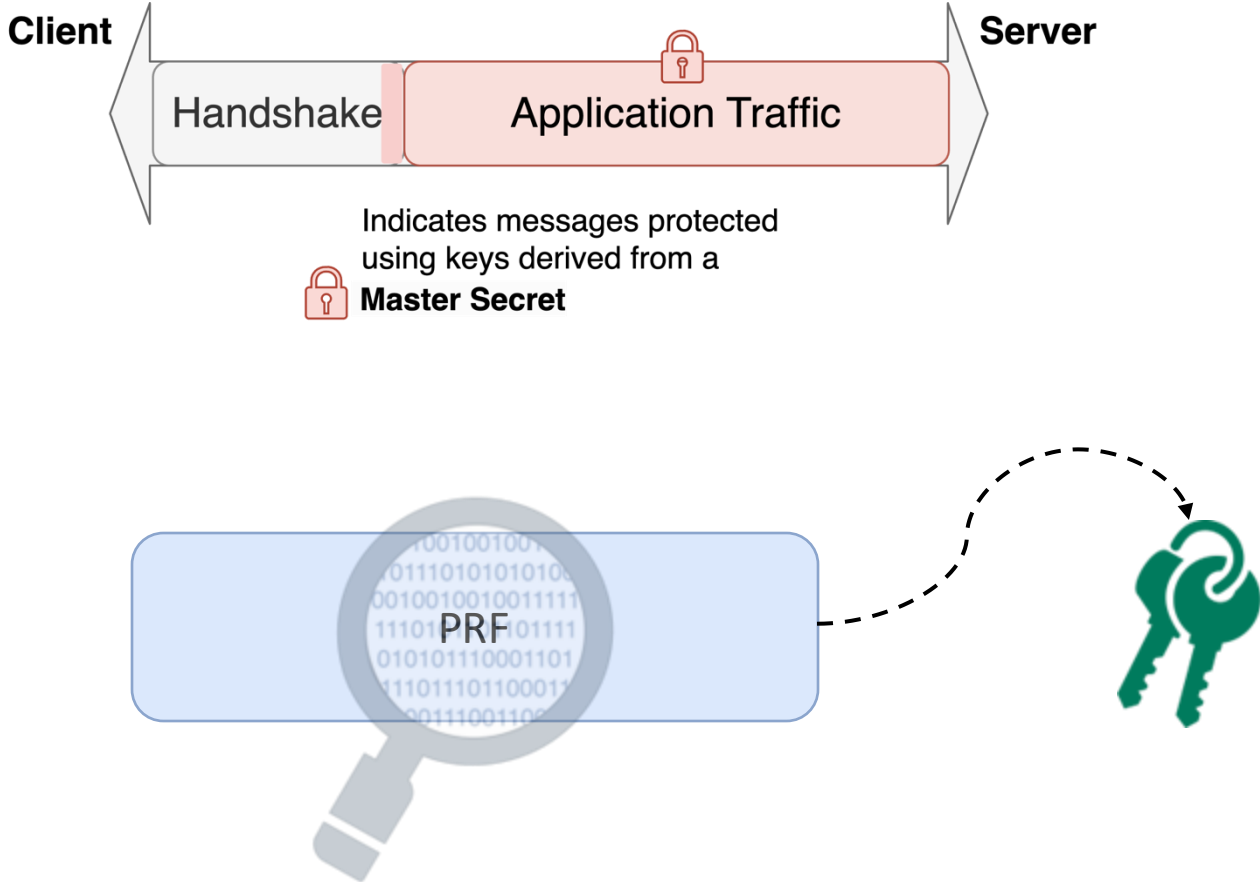
Secrets in Motion: Dynamic TLS Key Recovery Fundamentals

Decryption Keys ≤ TLS 1.2



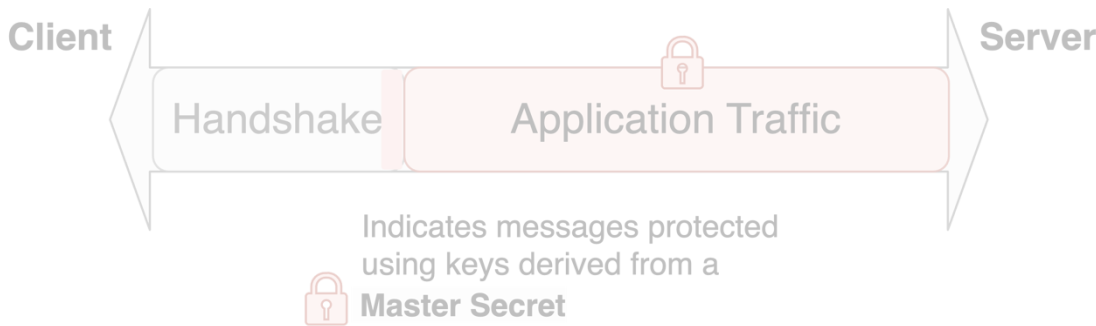
Secrets in Motion: Dynamic TLS Key Recovery Fundamentals

Decryption Keys ≤ TLS 1.2

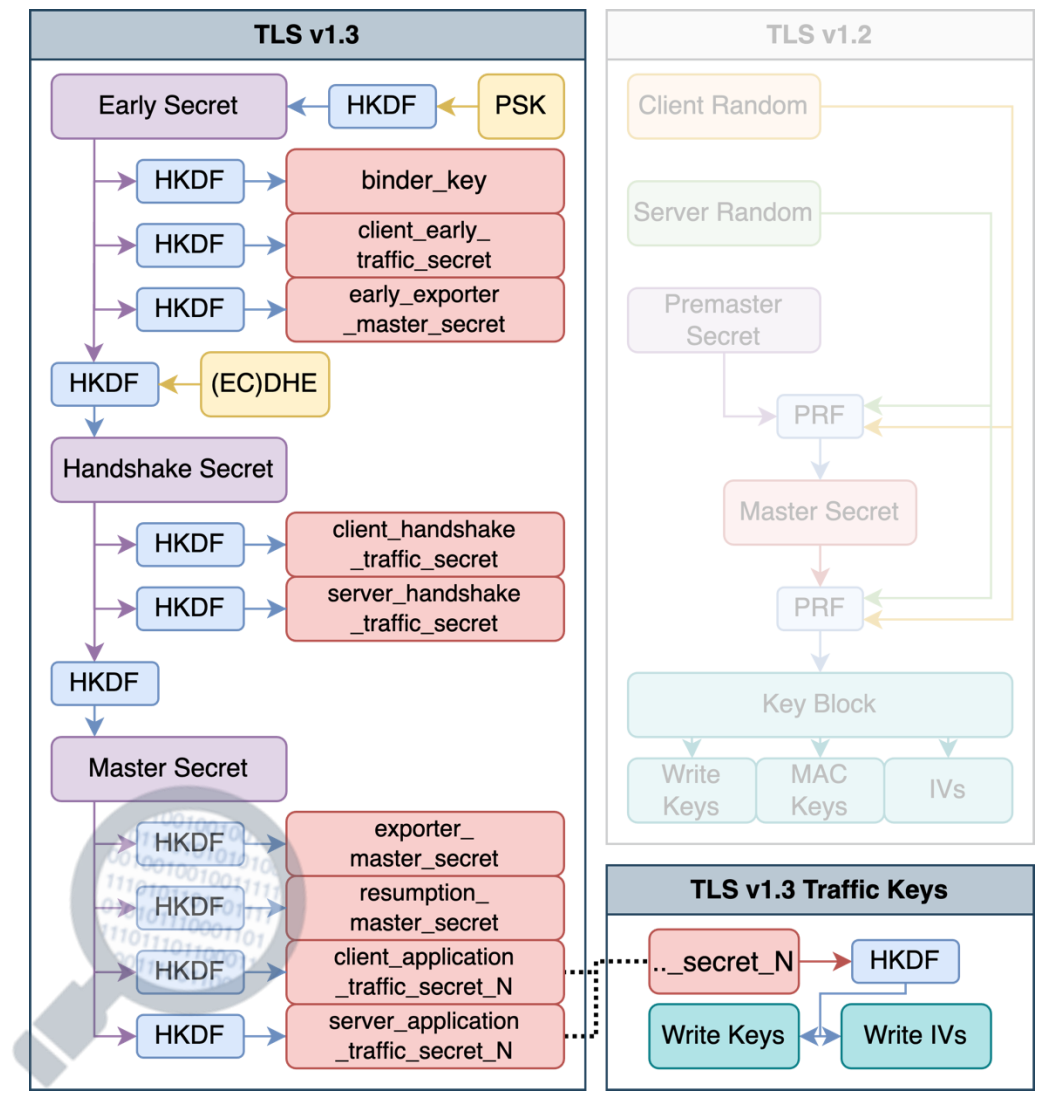
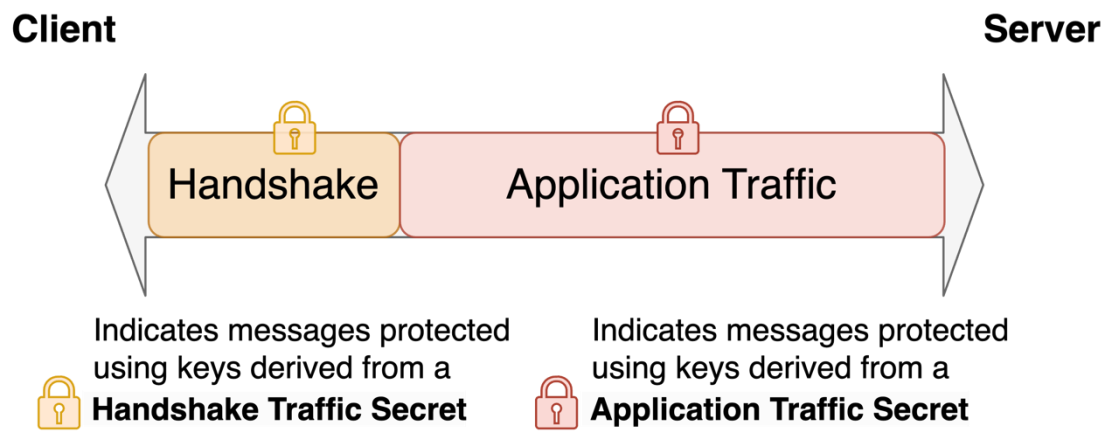


Secrets in Motion: Dynamic TLS Key Recovery Fundamentals

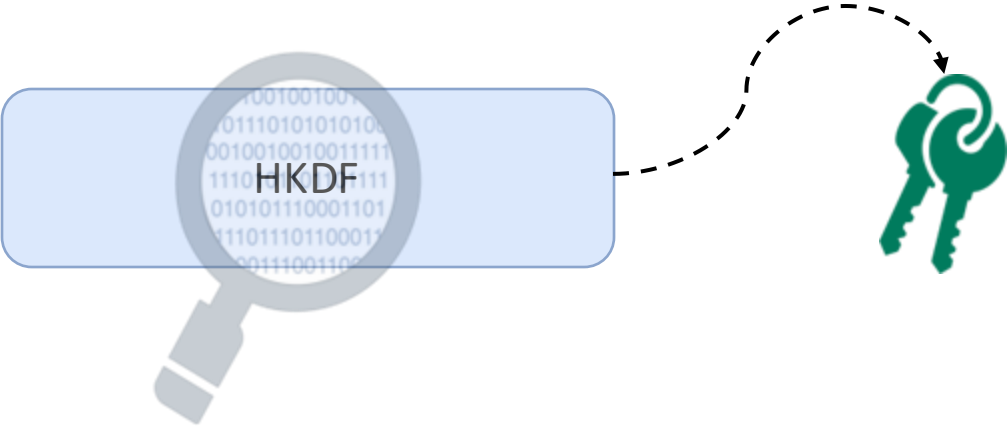
Decryption Keys ≤ TLS 1.2



Decryption Keys TLS 1.3



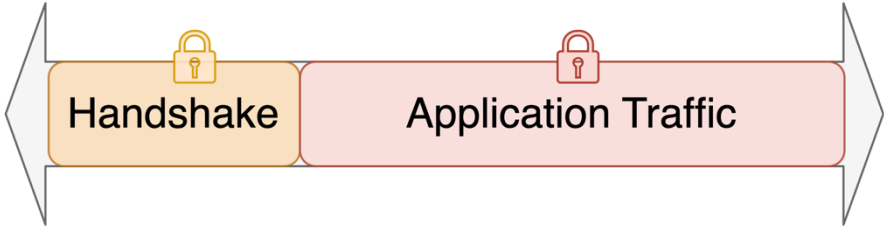
Secrets in Motion: Dynamic TLS Key Recovery Fundamentals



Decryption Keys TLS 1.3

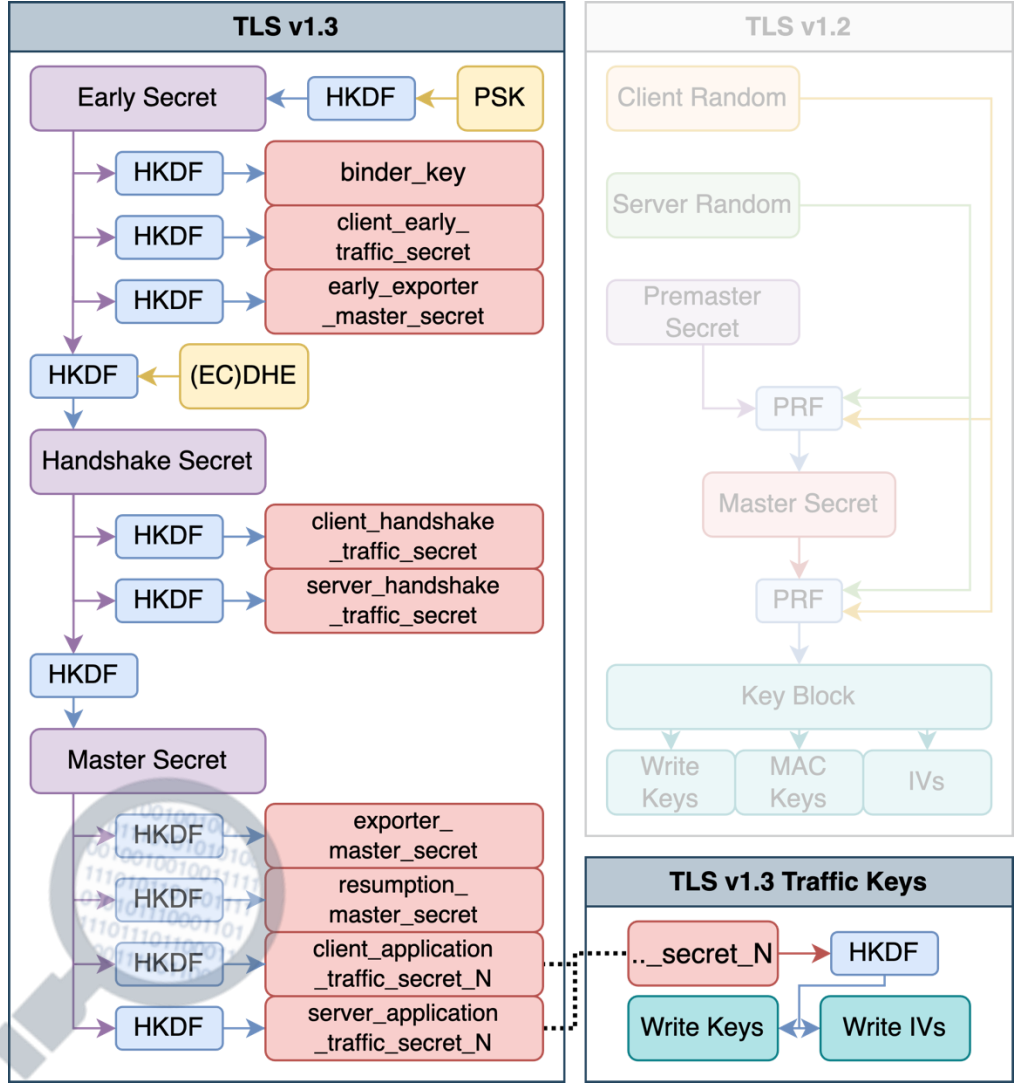
Client

Server

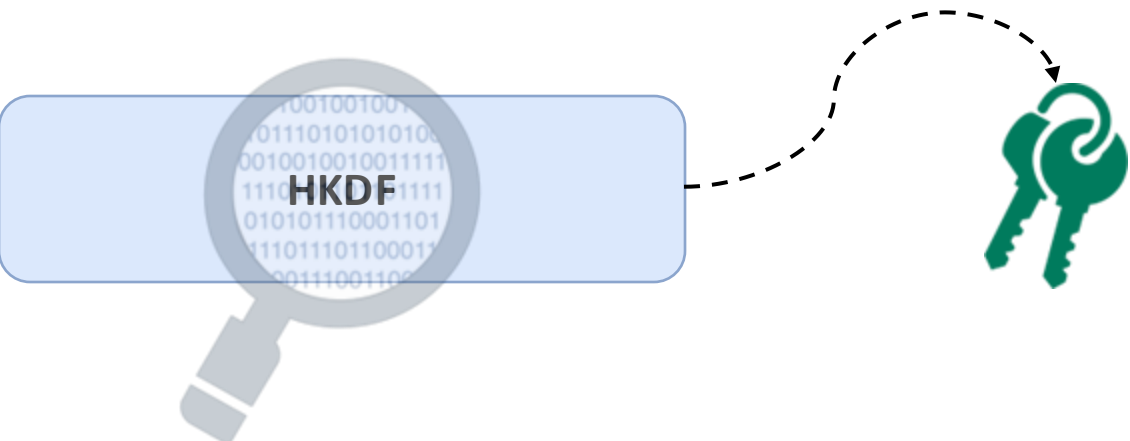
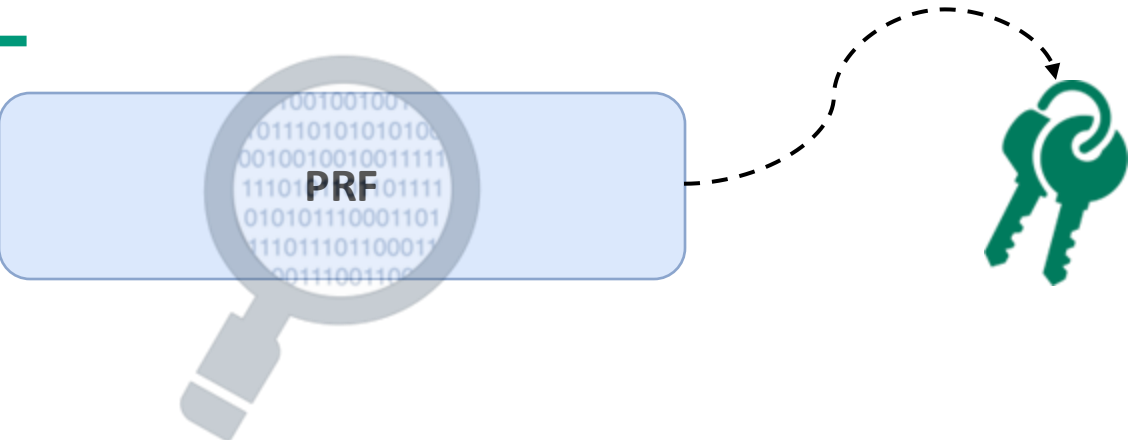


Indicates messages protected using keys derived from a **Handshake Traffic Secret**

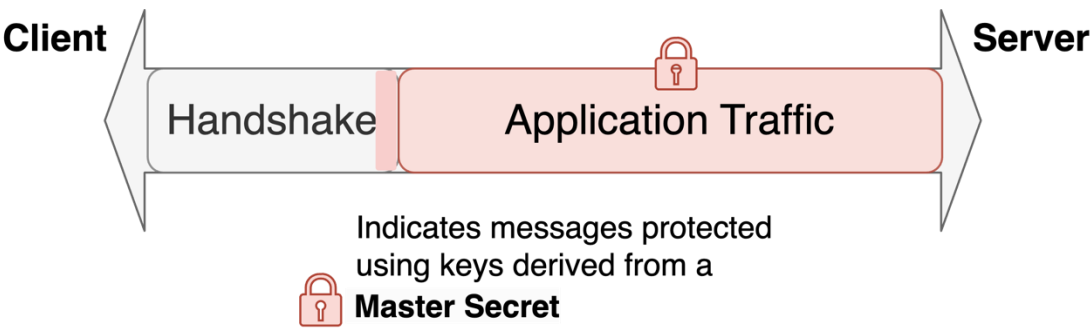
Indicates messages protected using keys derived from a **Application Traffic Secret**



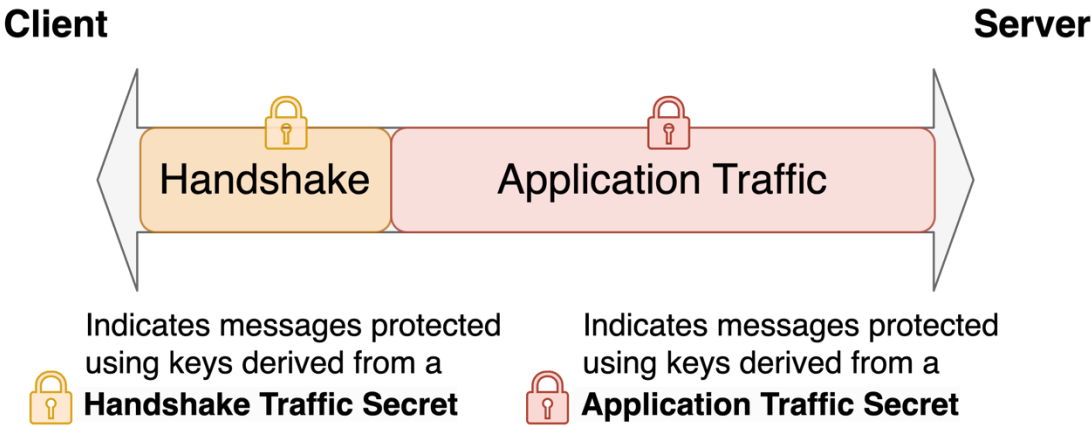
Secrets in Motion: Dynamic TLS Key Recovery Fundamentals

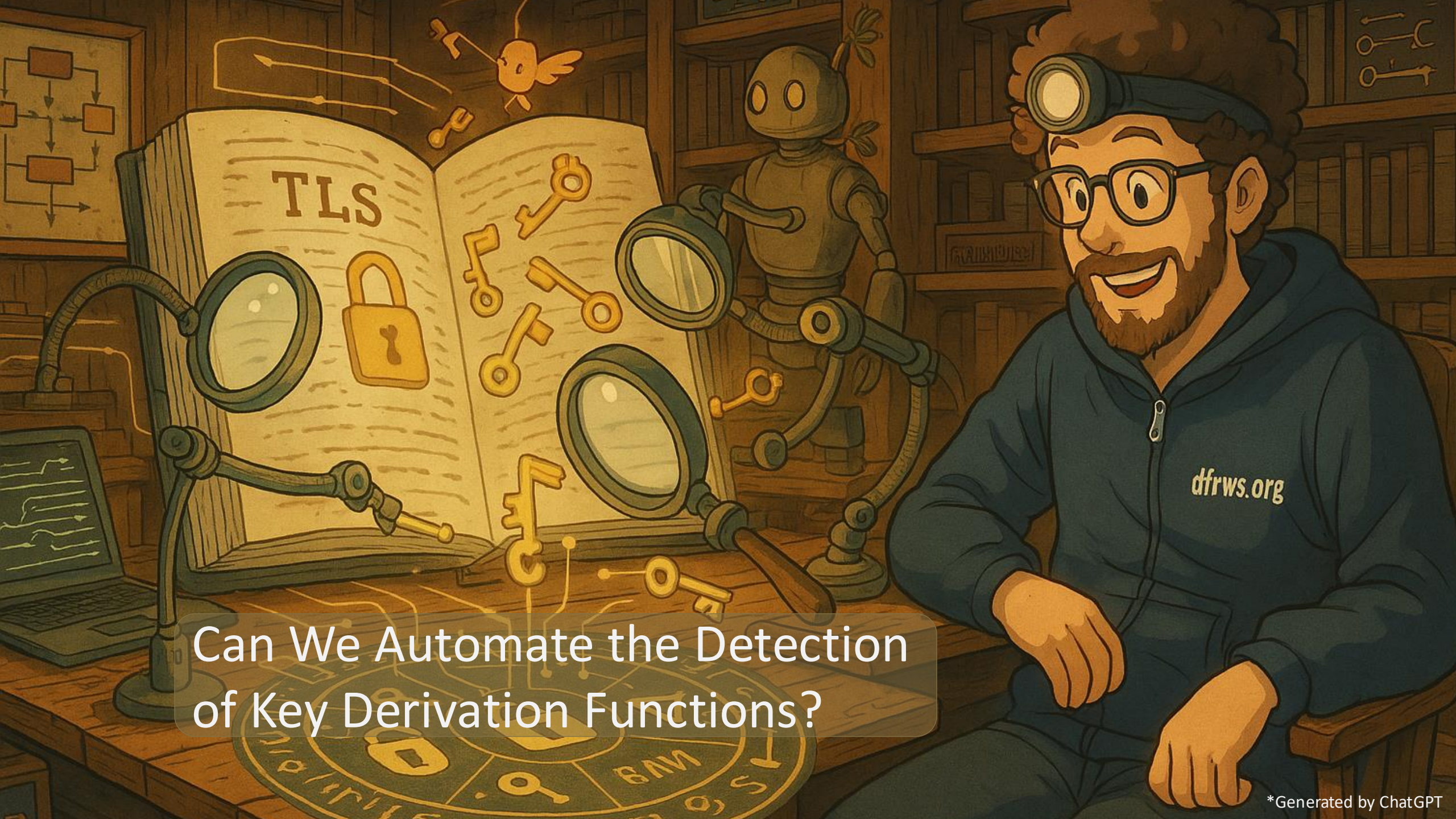


Decryption Keys ≤ TLS 1.2



Decryption Keys TLS 1.3



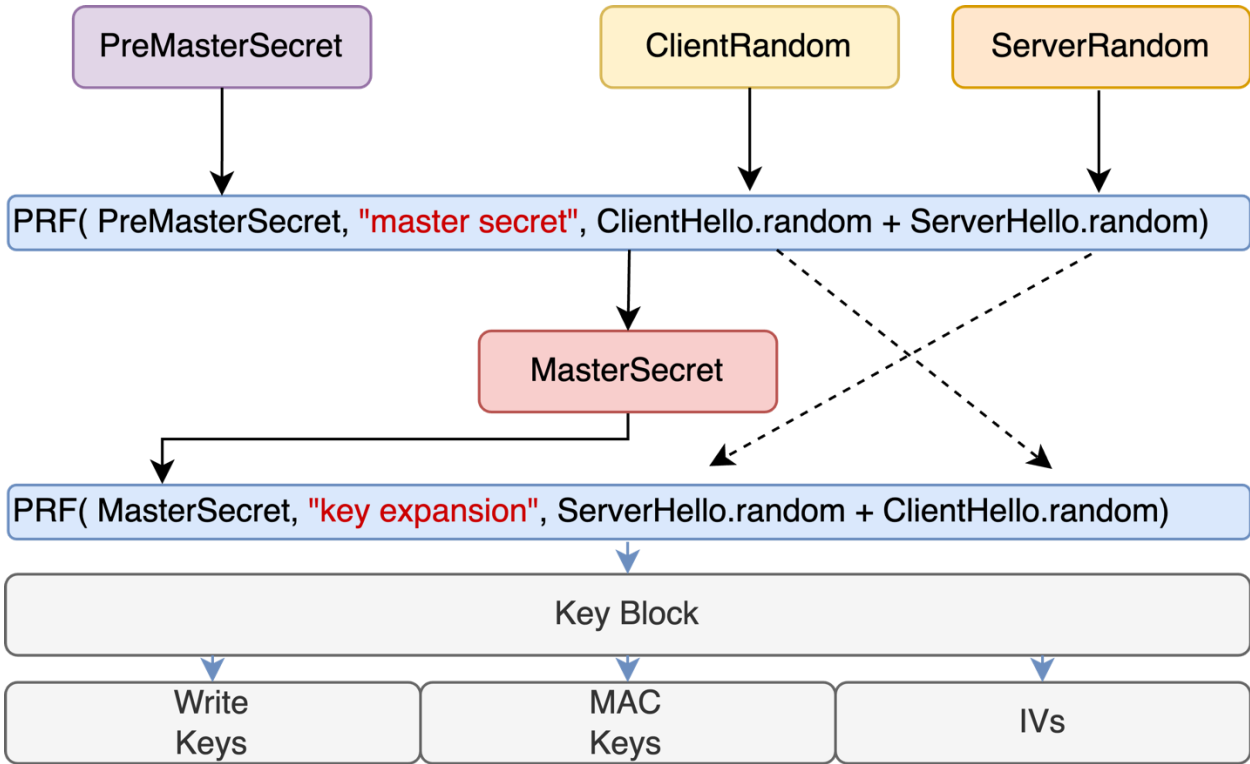
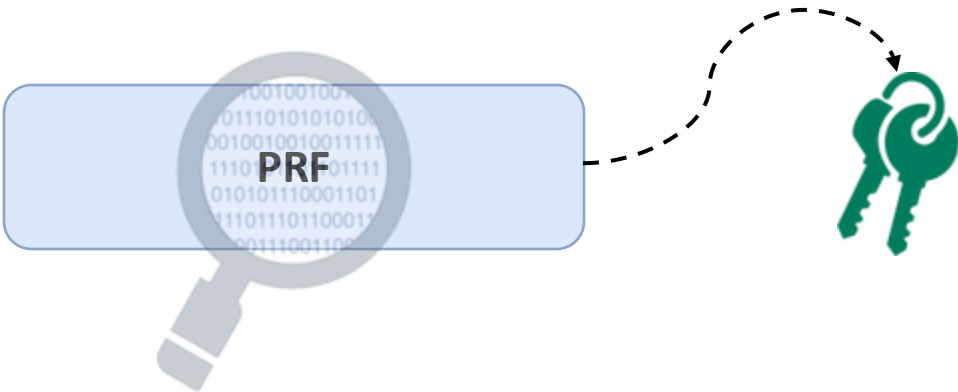


Can We Automate the Detection
of Key Derivation Functions?

Streamlined Identification of TLS Key Derivation Functions

TLS 1.2

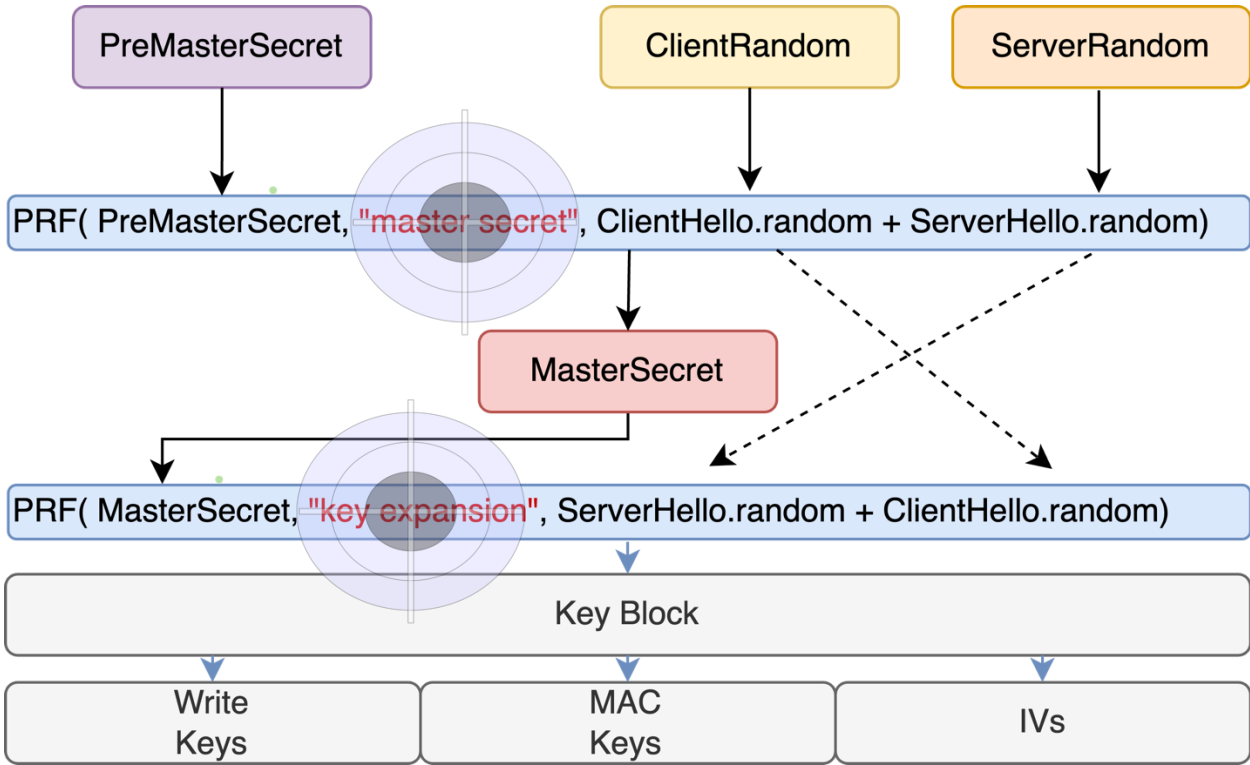
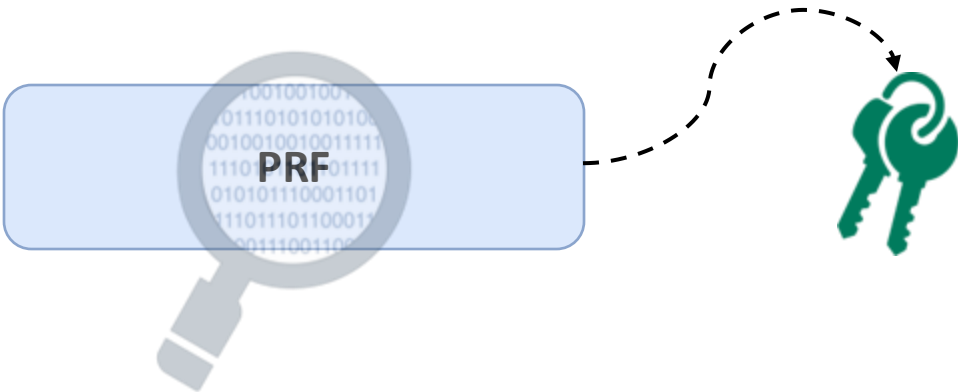
Decryption Keys TLS 1.2



Streamlined Identification of TLS Key Derivation Functions

TLS 1.2

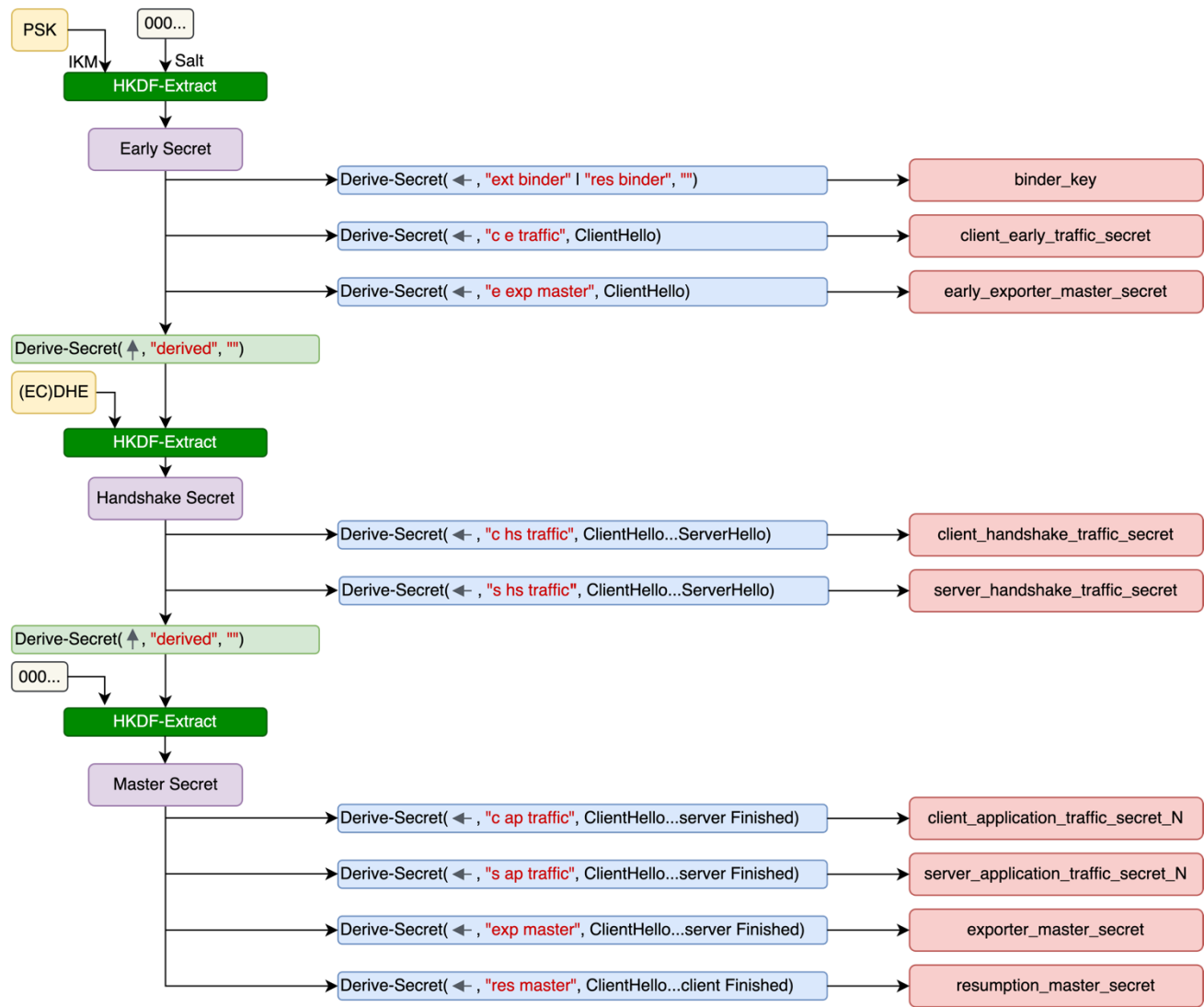
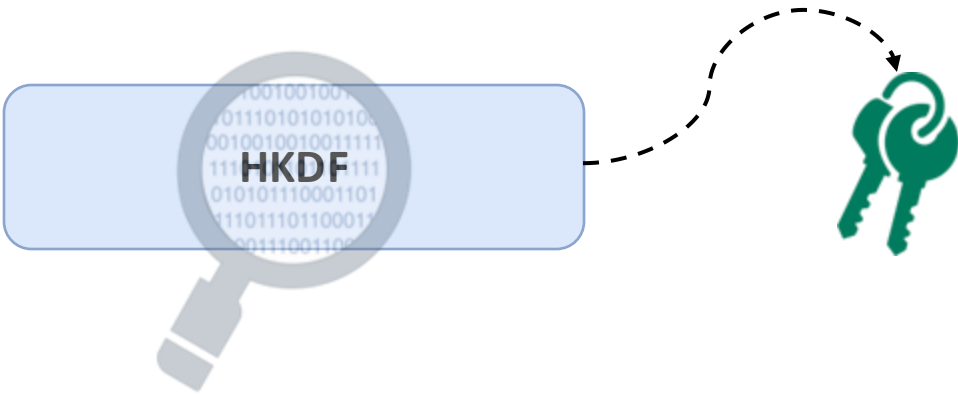
Decryption Keys TLS 1.2



Streamlined Identification of TLS Key Derivation Functions

TLS 1.3

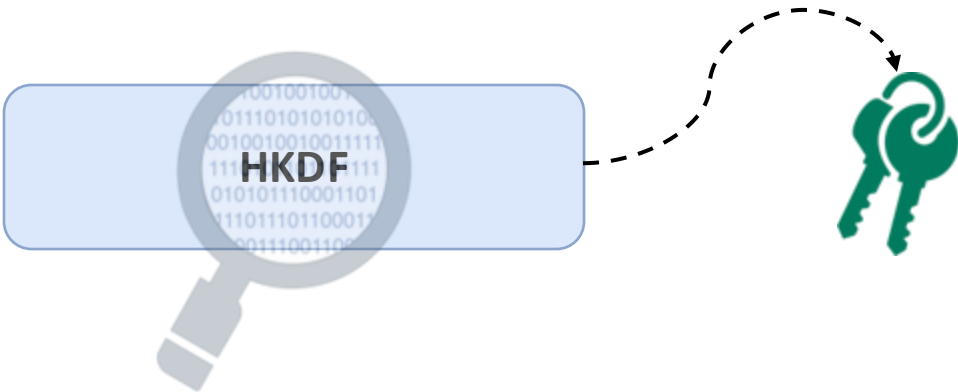
Decryption Keys TLS 1.3



Streamlined Identification of TLS Key Derivation Functions

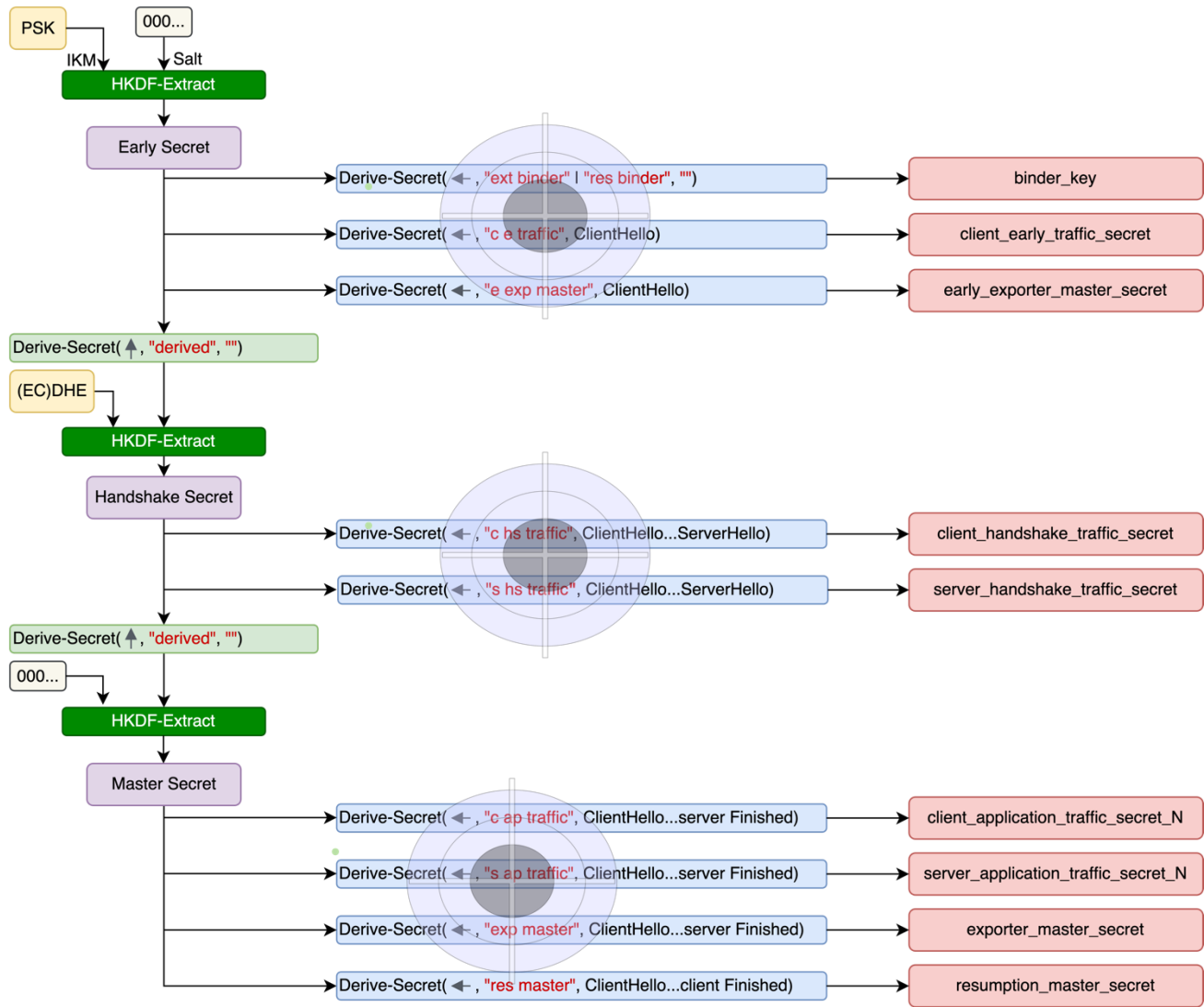
TLS 1.3

Decryption Keys TLS 1.3



```
struct {
    uint16 length = Length;
    opaque label<7..255> = "tls13 " + Label;
    opaque context<0..255> = Context;
} HkdfLabel;

Derive-Secret(Secret, Label, Messages) =
    HKDF-Expand-Label(Secret, Label,
        Transcript-Hash(Messages), Hash.length)
```

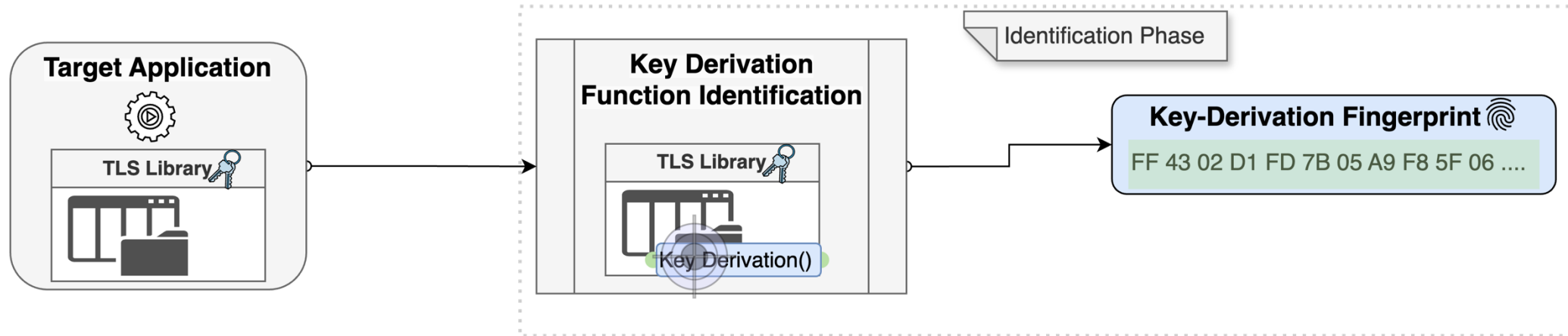




Introducing: TLSKeyHunter

Streamlined Identification of TLS Key Derivation Functions

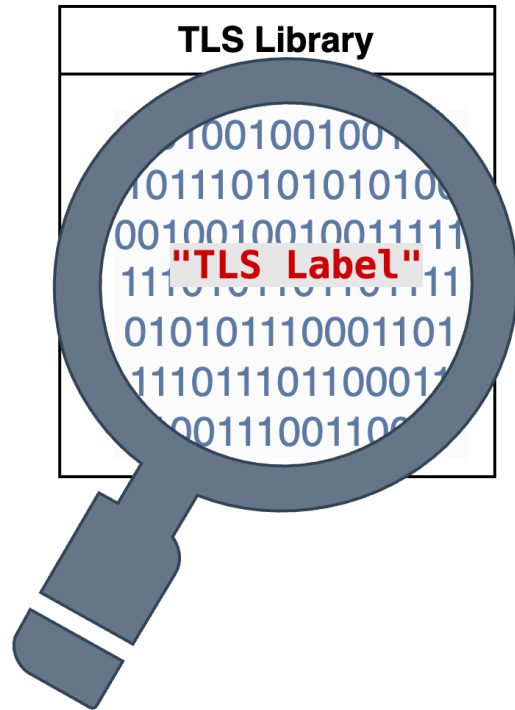
Introducing: TLSKeyHunter



Streamlined Identification of TLS Key Derivation Functions

Introducing: TLSKeyHunter

1 Label Localization



Streamlined Identification of TLS Key Derivation Functions

Introducing: TLSKeyHunter

① Label Localization



② Function Identification

```
lea rcx, referenced_TLS_Label ; "TLS Label"  
mov rdx, [rsp+1C8h+var_188]  
...  
call Key Derivation-Fct
```


Streamlined Identification of TLS Key Derivation Functions

Introducing: TLSKeyHunter

① Label Localization

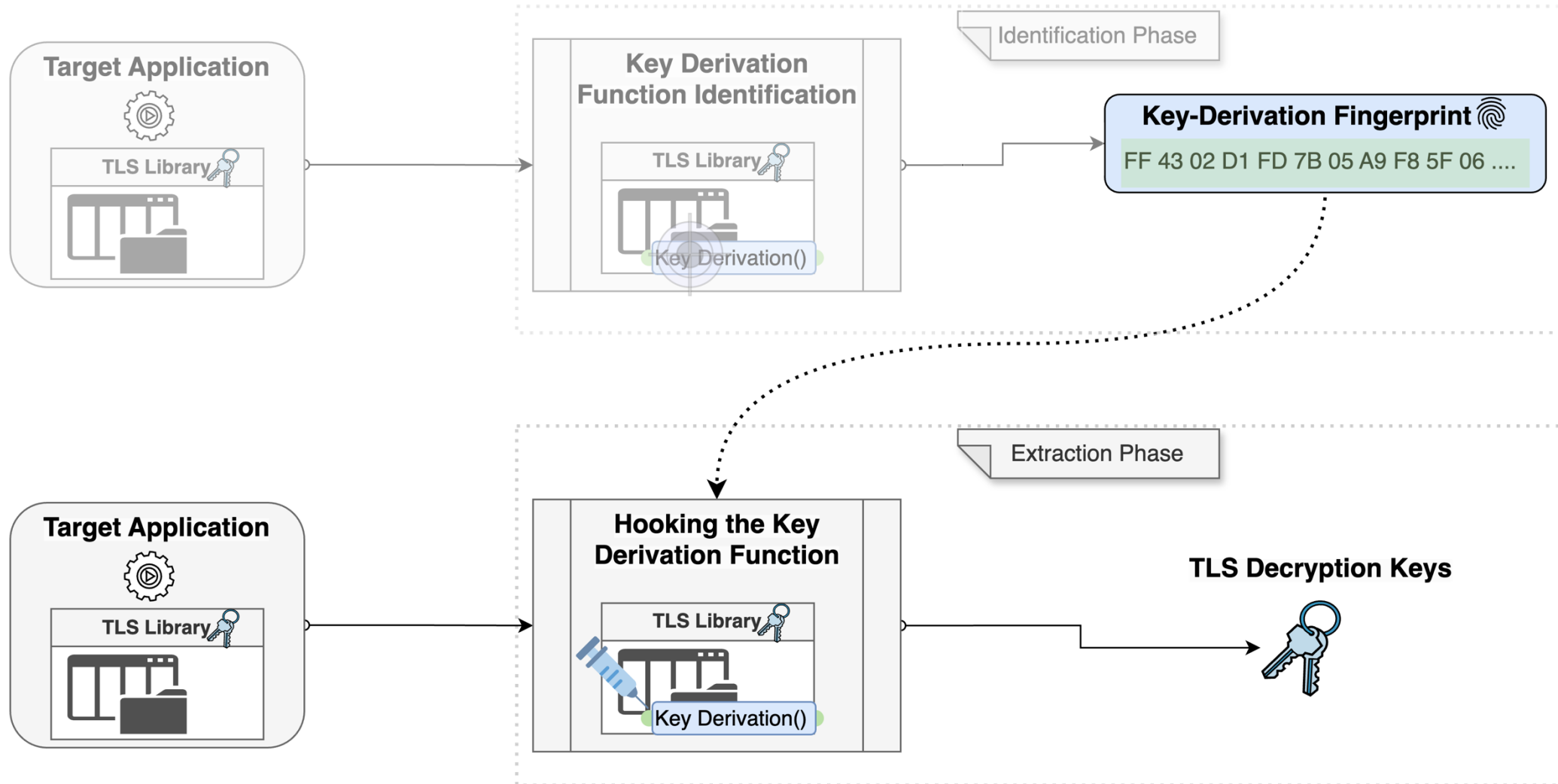


② Function Identification

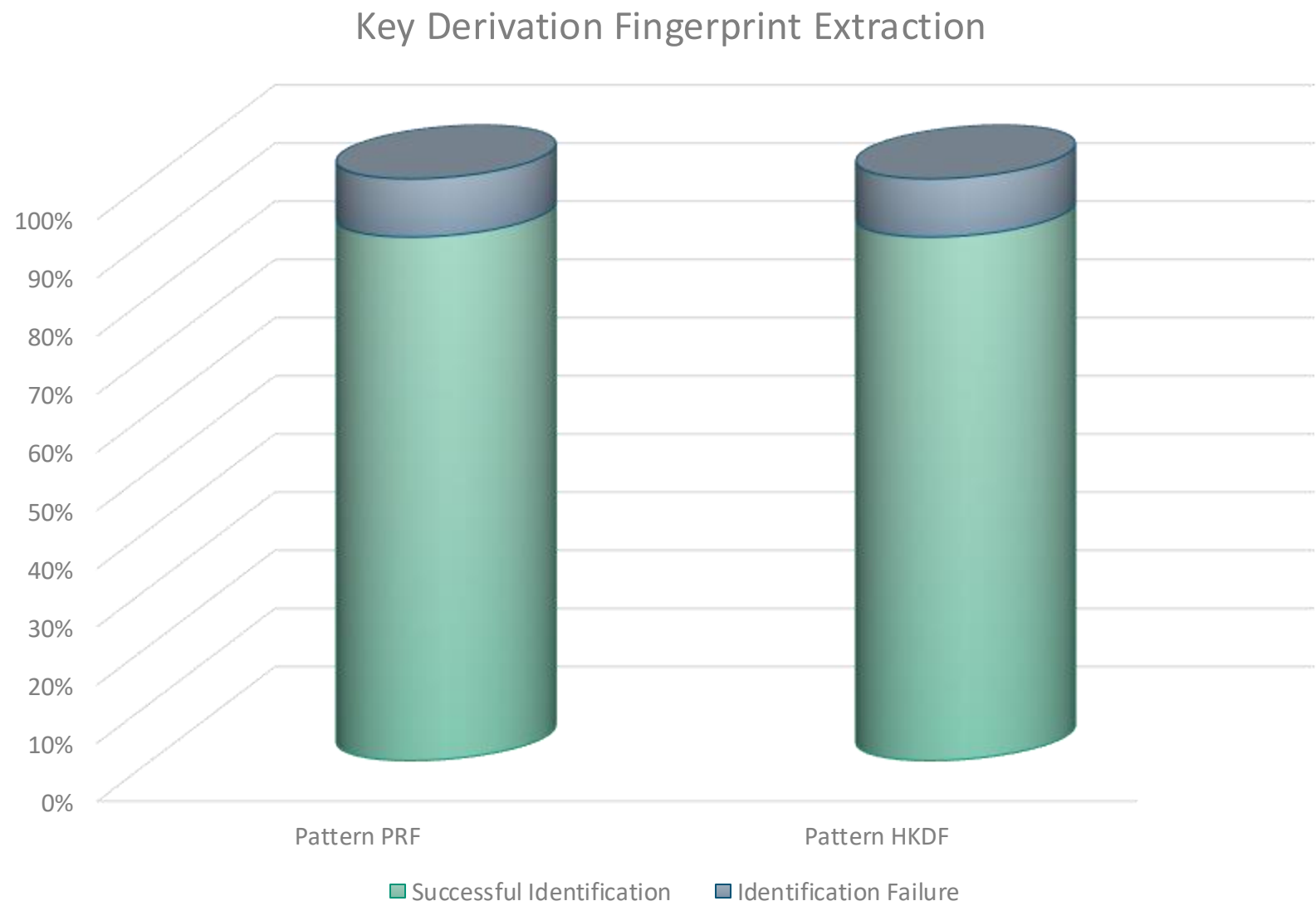
```
lea rcx, referenced_TLS_Label ; "TLS Label"  
mov rdx, [rsp+1C8h+var_188]  
...  
call Key Derivation-Fct
```

Key Derivation-Fct(←, "TLS Label", Session_Hash/Seed)

Live Memory Forensic Key Extraction



Preliminary Results (Ongoing Evaluation): Key Derivation Identification



Preliminary Results (Ongoing Evaluation): Key Derivation Identification

Key Derivation Fingerprint Extraction

TLS Libraries	PRF Pattern	HKDF Pattern
Botan SSL	✓	✓
BoringSSL	✓	✓
Bouncy Castle	✓	✓
GnuTLS	✓	✓
Golang crypto/tls	✓	✓
Mbed TLS	✓	✓
RusTLS	X	X
NSS	✓	✓
OpenSSL	✓	✓
Schannel SSP	✓	✓
....	...	...

Preliminary Results (Ongoing Evaluation): Key Derivation Identification

Key Derivation Fingerprint Extraction

TLS Libraries	PRF Pattern	HKDF Pattern
Botan SSL	✓	✓
BoringSSL	✓	✓
Bouncy Castle	✓	✓
GnuTLS	✓	✓
Golang crypto/tls	✓	✓
Mbed TLS	✓	✓
RusTLS	X	X
NSS	✓	✓
OpenSSL	✓	✓
Schannel SSP	✓	✓
...	...	...

Six libraries remain to be evaluated...

Preliminary Results (Ongoing Evaluation): Key Derivation Identification

Key Derivation Fingerprint Extraction

TLS Libraries		PRF Pattern	
Botan SSL		✓	✓
BoringSSL		✓	✓
Bouncy Castle		✓	✓
GnuTLS		✓	✓
Golang crypto/tls		✓	✓
Mbed TLS		✓	✓
RusTLS		X	X
NSS		✓	✓
OpenSSL		✓	✓
Schannel SSP		✓	✓
...		...	...

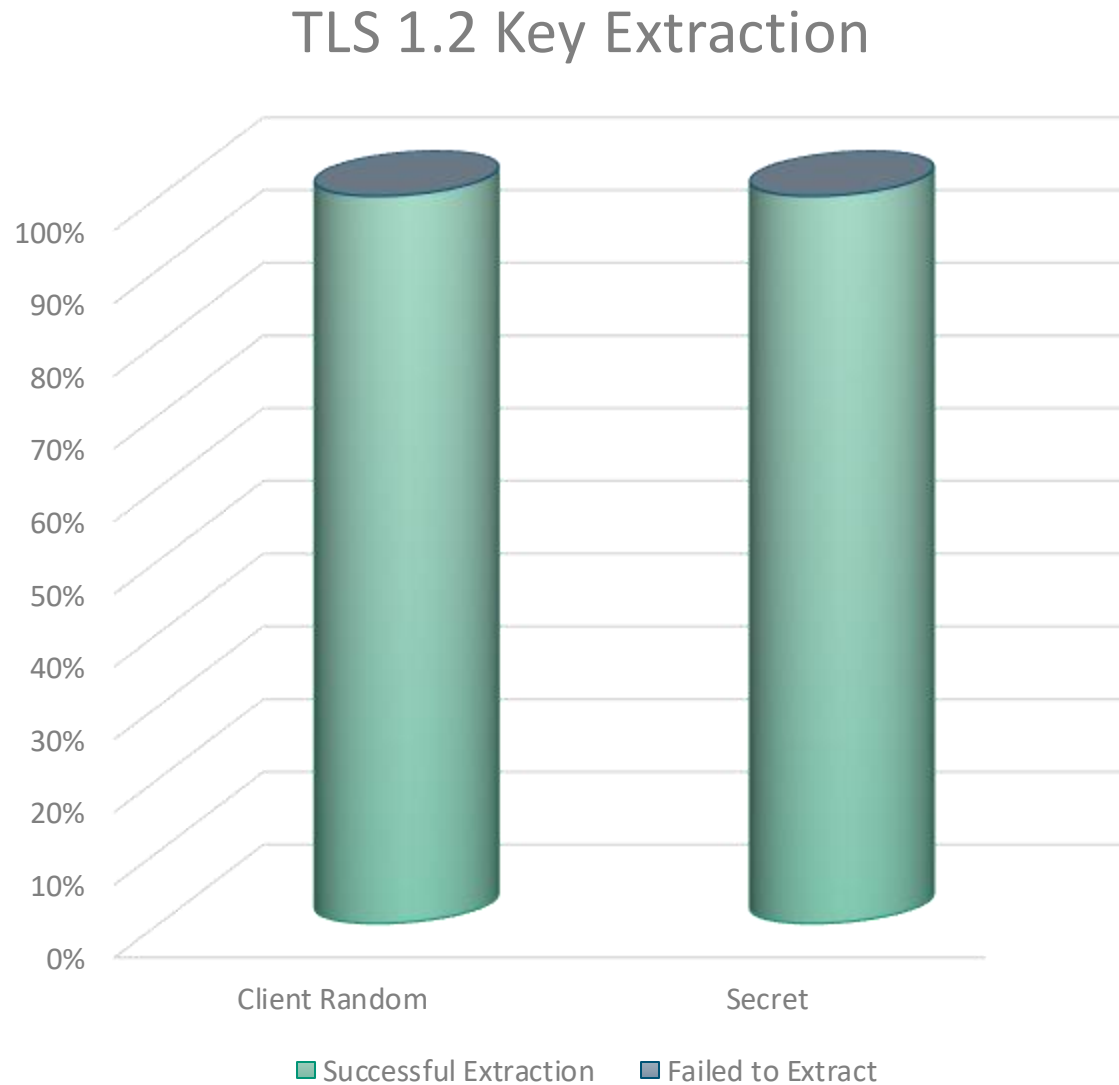


GHIDRA

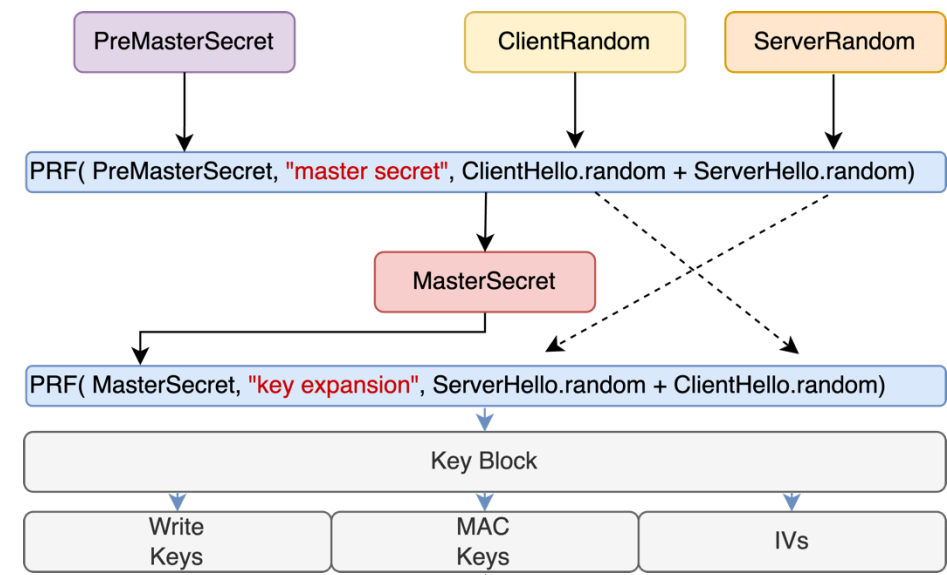
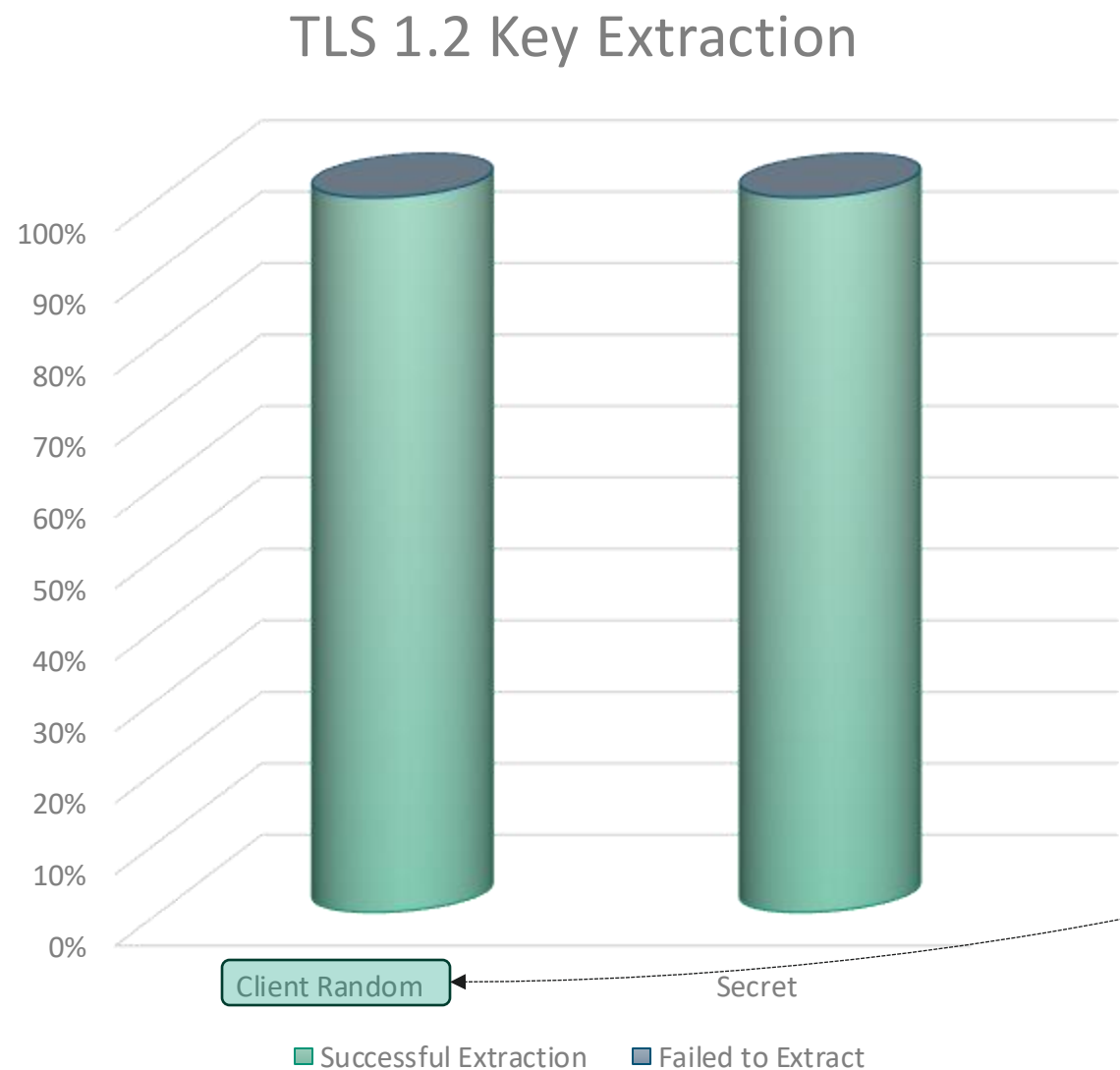
Rust support introduced some issues, but manual verification confirmed our approach.

Six libraries remain to be evaluated...

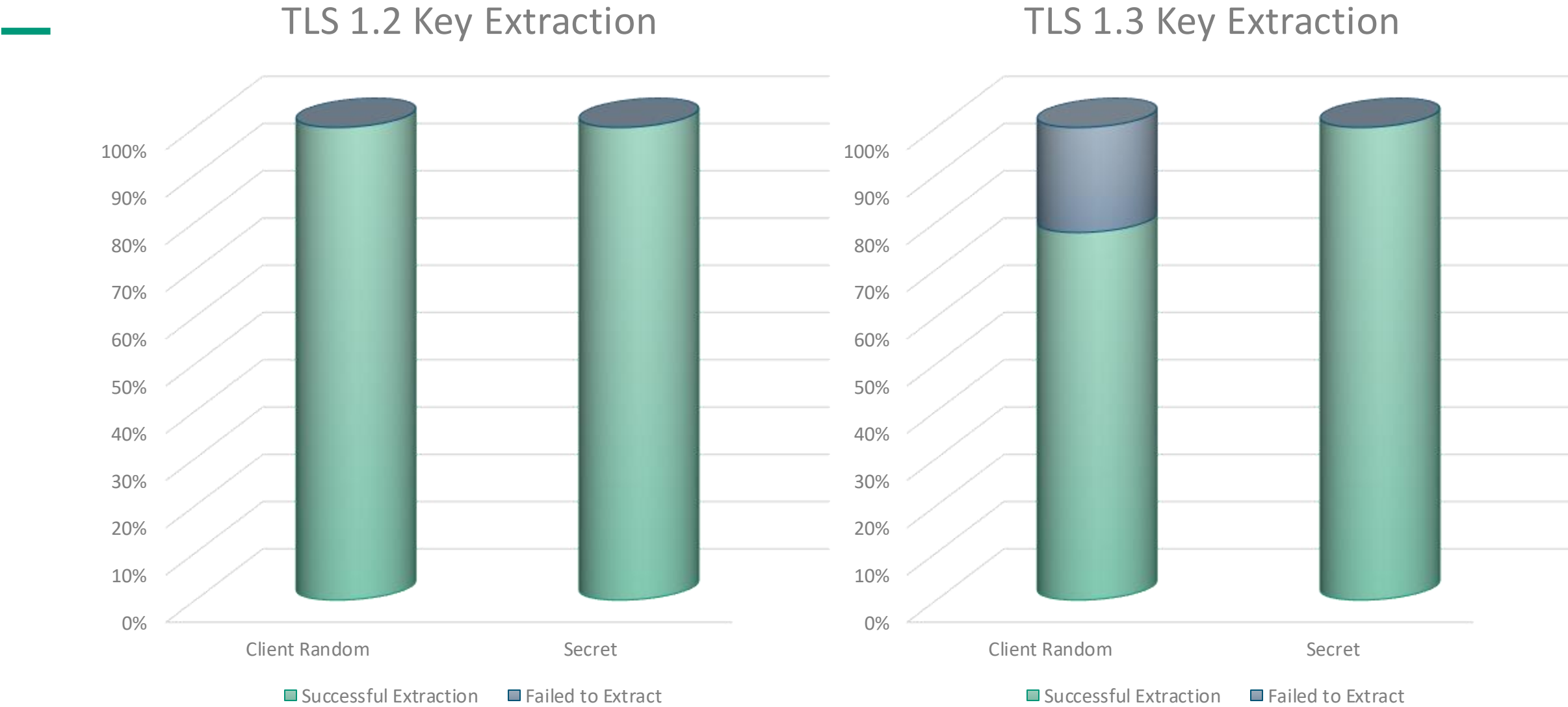
Preliminary Results (Ongoing Evaluation): TLS Key Extraction



Preliminary Results (Ongoing Evaluation): TLS Key Extraction



Preliminary Results (Ongoing Evaluation): TLS Key Extraction





Conclusion



We successfully showed **that...**

- ✓ TLS Labels offer a good anchor to identify the TLS key derivation functions
- ✓ Intercepting (hooking) them always results in the keys needed

Outlook

- Finishing our research Paper
 - Finish writing
 - Test approach against common applications
- Publishing our Framework
- Integration into **FЯITop**



Thank you for your attention!

Daniel Baier
Cyber Analysis & Defense
daniel.baier@fkie.fraunhofer.de

Fraunhofer Institute for Communication,
Information Processing and Ergonomics FKIE
Zanderstrasse 5 | 53177 Bonn

www.fkie.fraunhofer.de

