TLS Security Training

Juraj Somorovsky (@jurajsomorovsky)

HACKMANIT

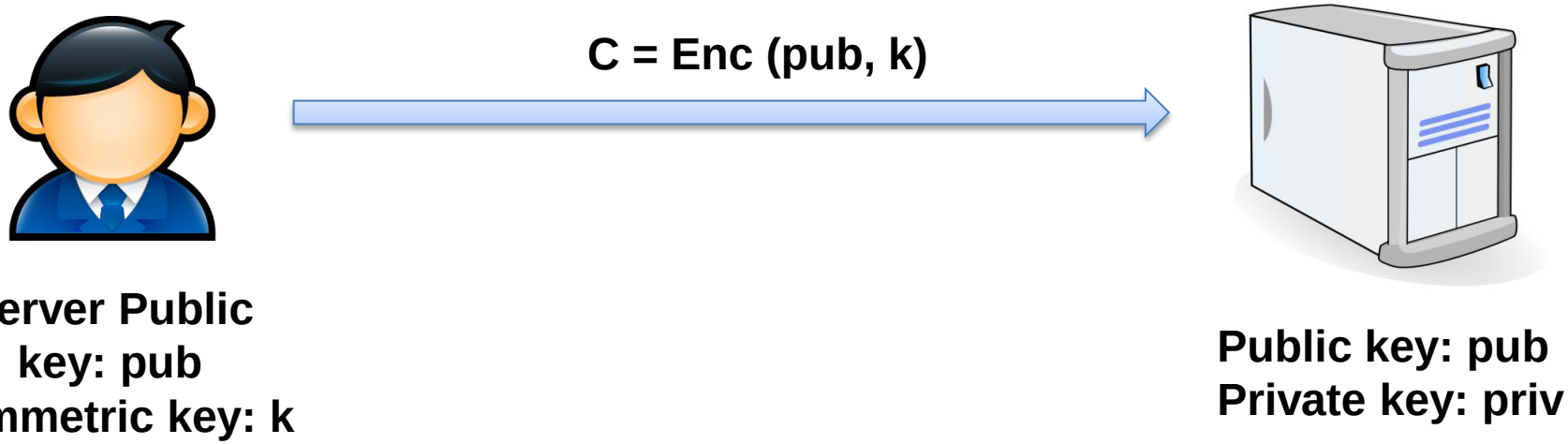
RUHR UNIVERSITÄT BOCHUM RUB

Overview of the Training

1. Crypto Basics
2. Transport Layer Security
3. Certificates
4. Attacks on PKI
5. Attacks on TLS
6. TLS Evaluation Tools
7. TLS Implementations
8. Outlook

How to Distribute Symmetric Keys?

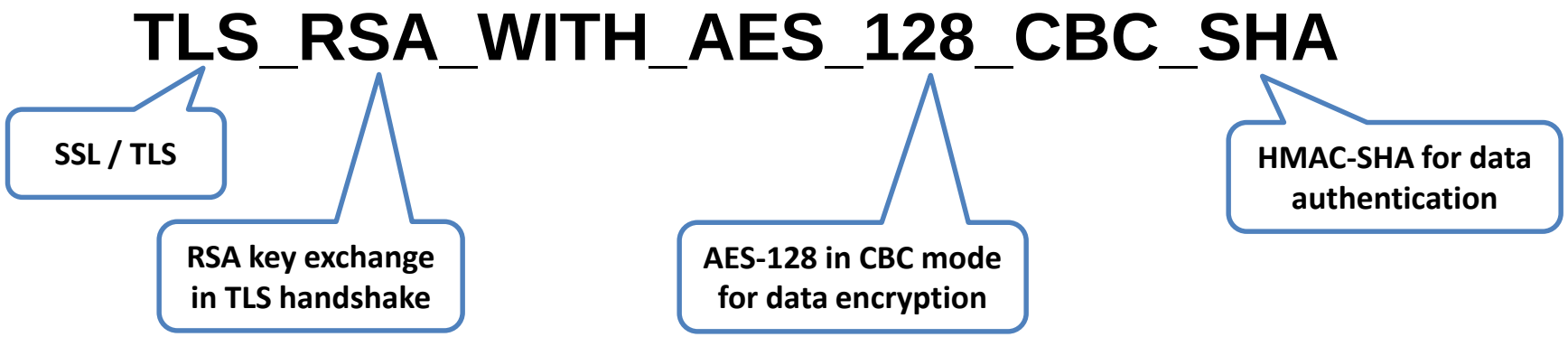
- Public-key (asymmetric-key) crypto



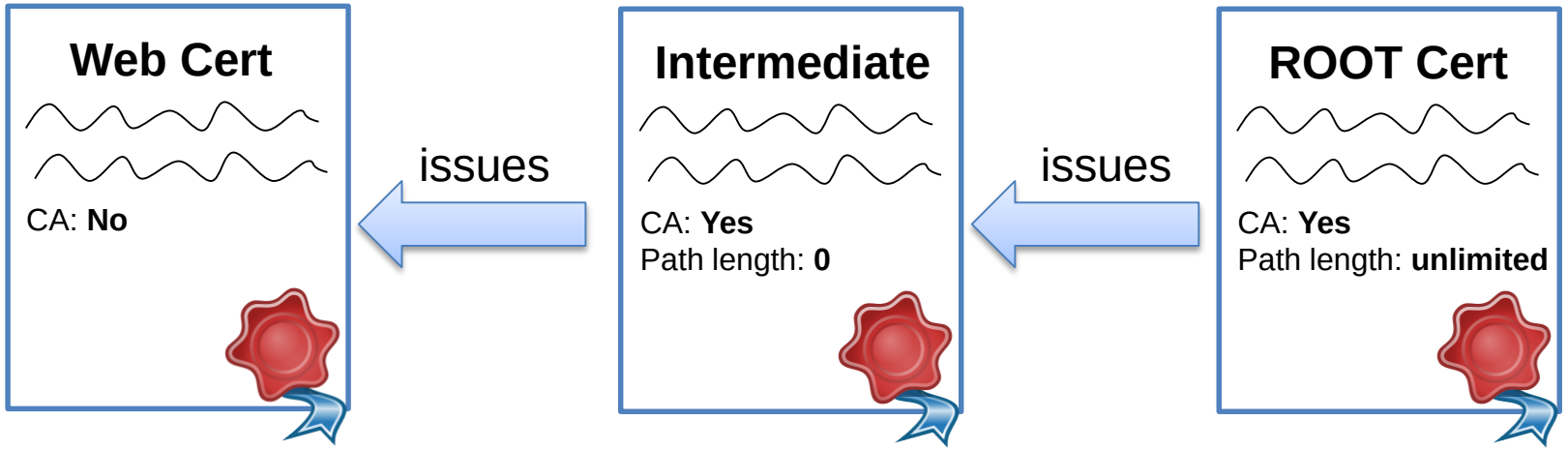
- Algorithms: RSA

TLS Cipher Suite

- Collection of crypto algorithms used in a TLS session



Certificate Chains



- Root CAs issue intermediate certs
- Intermediate CAs issue certificates for subscribers

TLS History

Secure Sockets Layer (SSL), SSLv2

SSLv3

1995

Wagner, Schneier: Analysis of SSLv3

Bleichenbacher's attack

Trasnsport Layer Security

2000

Padding oracle attack

2005

TLS 1.1

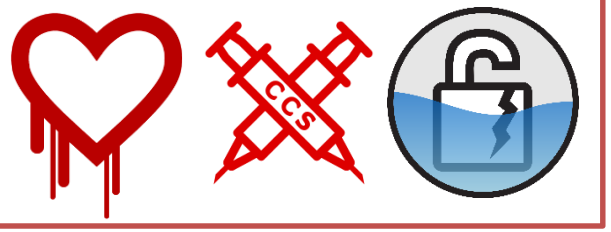
TLS 1.2

2010

BEAST, CRIME, BREACH, Lucky 13

TLS 1.3

2015

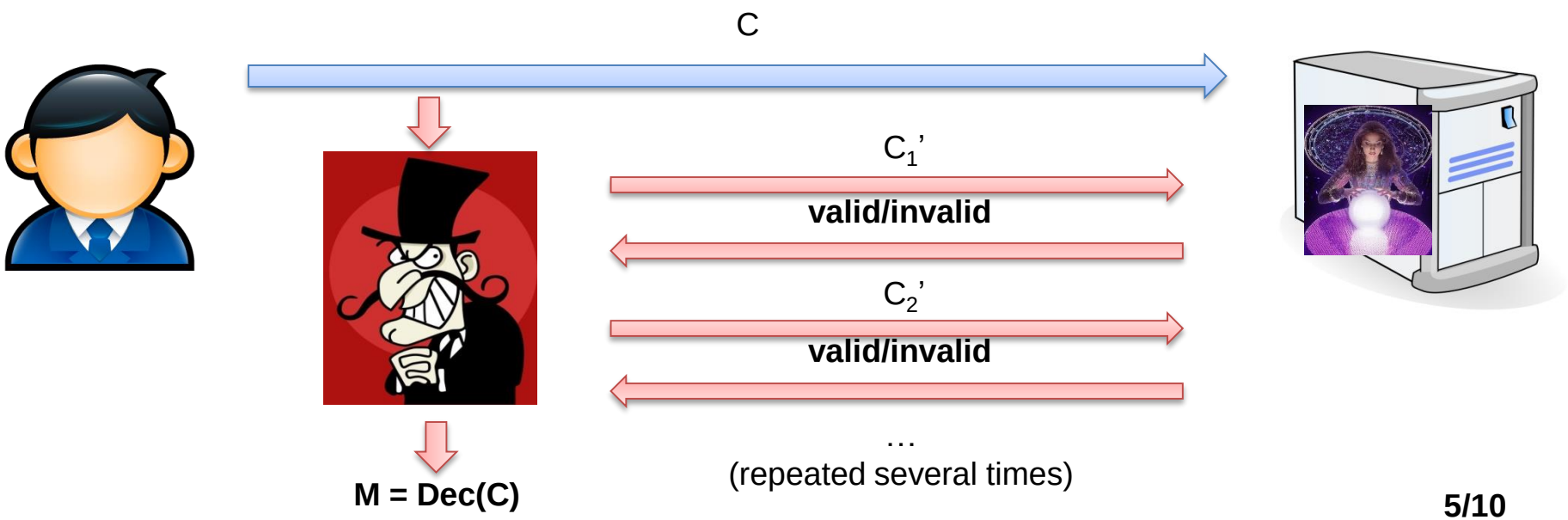


TLS Server as an Oracle

- Attacker can query the server
- The server decrypts and responds with **valid/invalid**
- Possible side channels:
  - Direct messages
  - Timing

Million message attack

Padding oracles



CRIME: Compression

- TLS offers compression
- Deflate compression (used in ZIP, GZIP…)
- Compression:

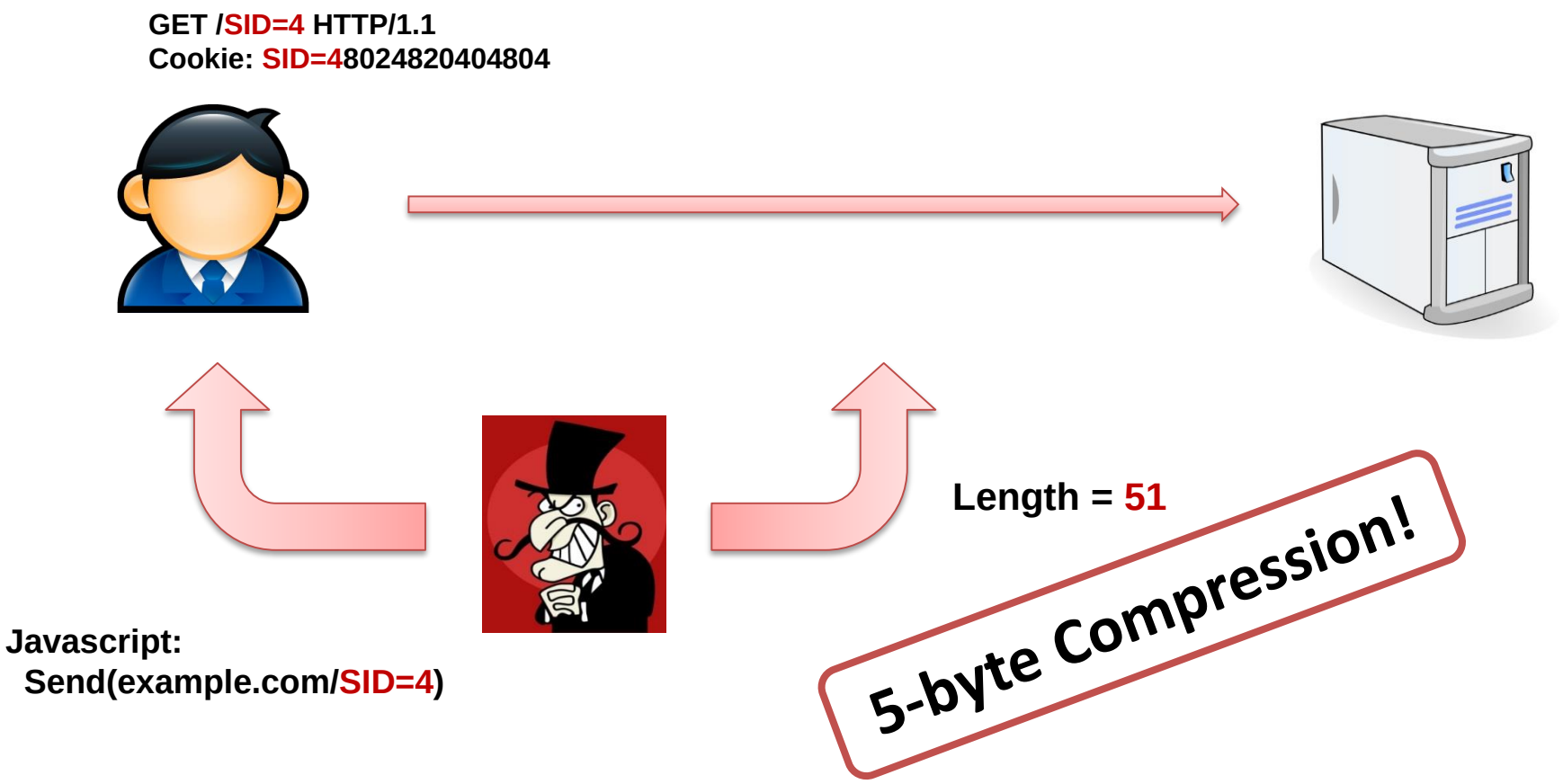
compression is **comp**lex  **comp**ression is **(-15,4)**lex

compression is easy  compression is easy

- More redundancy -> more compression
- Less redundancy -> less (no) compression

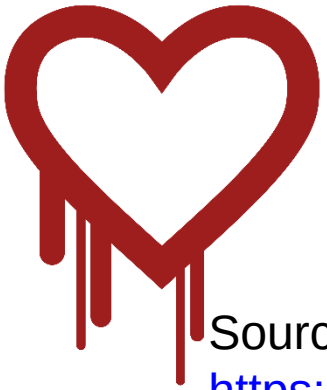
CRIME: Scenario

- Assumption: Streaming-based cipher

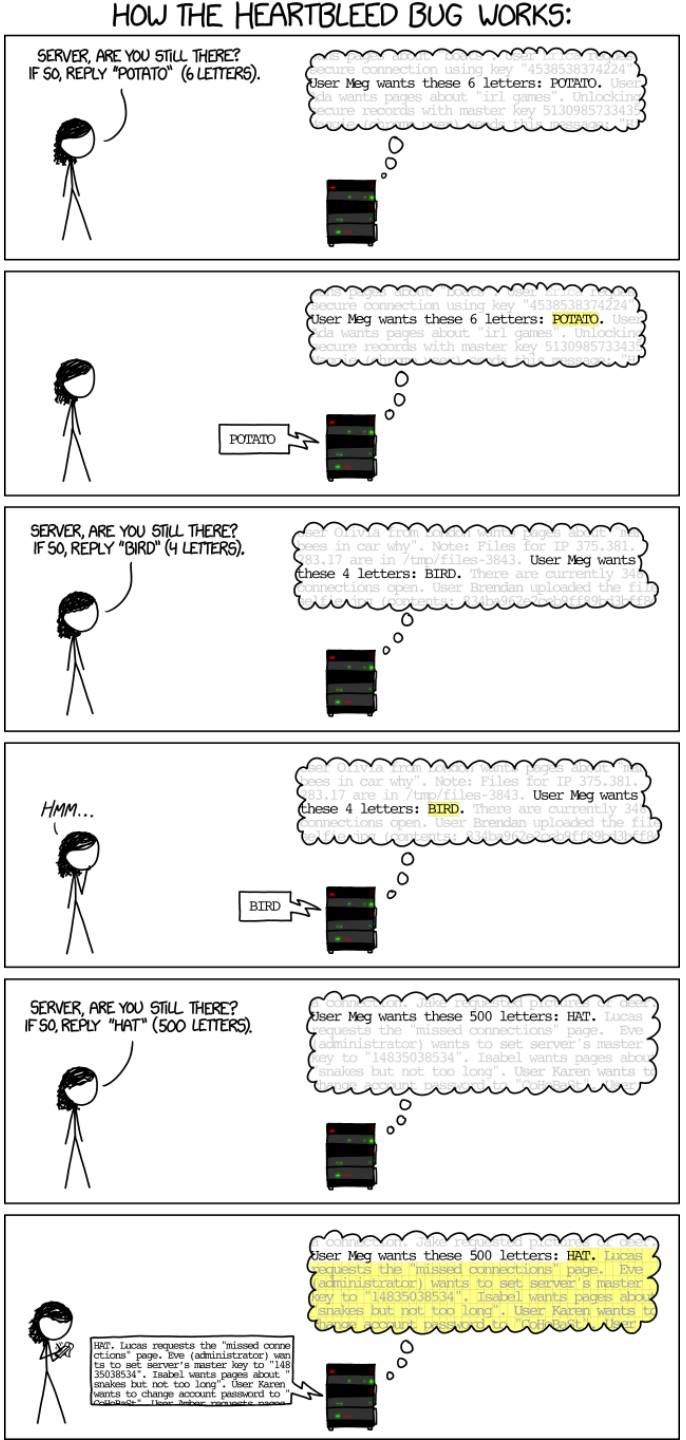


Heartbleed

- Discovered in April 2014
- Riku, Antti and Matti (Codenomicon) and Mehta (Google Security)
- Buffer Overread
- Improper validation of a Heartbeat request



Source: https://xkcd.com/1354/



mbed TLS

- Formerly known as PolarSSL
- https://tls.mbed.org/
- Especially for hardware devices

- Running:
  - Compile the library
  - Dummy test server located in:

  mbedtls-2.0.0/programs/ssl/**ssl_server2**
  - Run with the following parameters:

  ssl_server2 server_port=54001 key_file=[rsakey.pem] crt_file=[rsacert.pem]

Complex Cipher Suite Configuration

- Only SHA1 cipher suites:
  - openssl ciphers ‘SHA’
- No SHA1 cipher suites:
  - openssl ciphers ‘!SHA’
- ECDH and DH cipher suites, no RSA:
  - openssl ciphers ‘ECDH:DH:!RSA’
- “Whitebox” configuration example:
  - openssl ciphers ‘ECDHE-ECDSA-AES256-GCM-SHA384,ECDHE-ECDSA-AES256-SHA384,ECDHE-ECDSA-AES128-GCM-SHA256‘

Does not mean your server will offer them all. You have to provide appropriate keys!

Apache httpd Installation

- Better do not compile yourself … but you can try it
- Ubuntu 14.04:

```
$ sudo apt-get install apache2
$ apache2 -version
   Server version: Apache/2.4.7 (Ubuntu)
   Server built:   Oct 14 2015 14:20:21
```

- Install mod_ssl

```
$ sudo a2enmod ssl
```

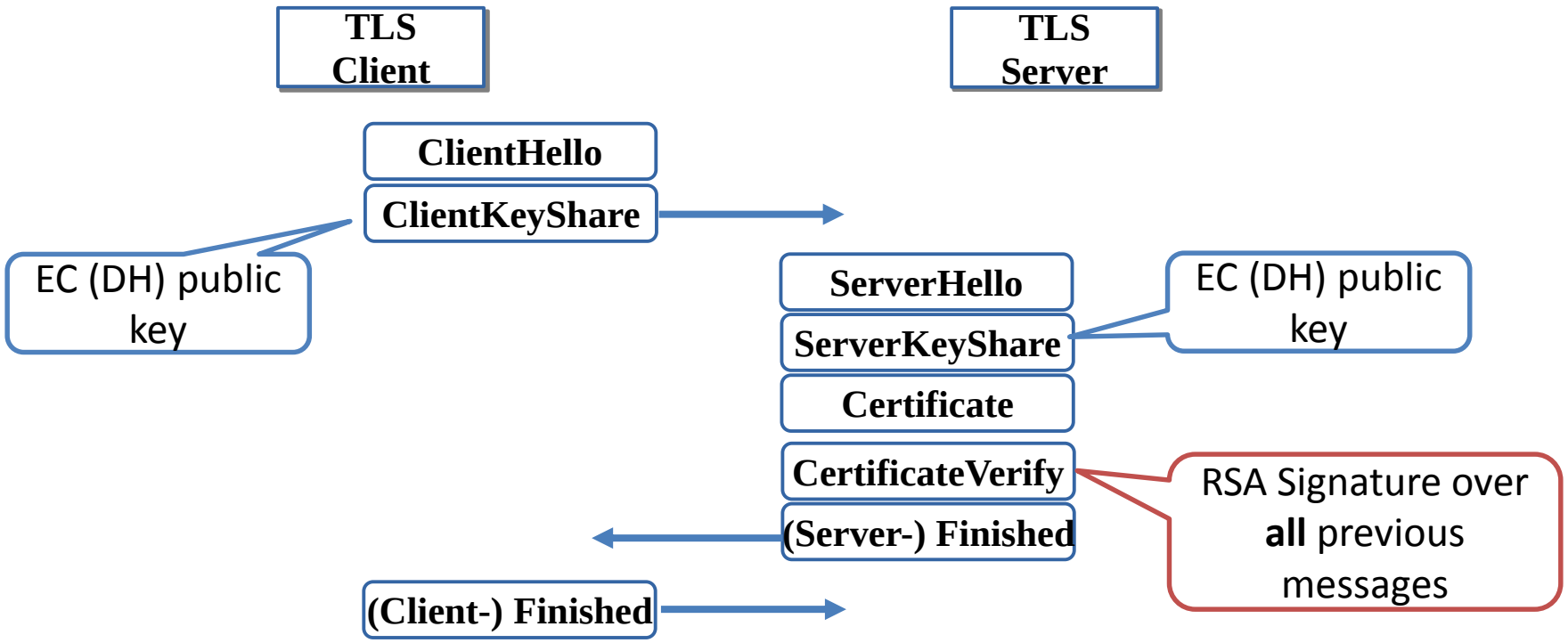
Apache httpd: Client Authentication

- Given we want to protect “/secure” directory
- Access given only to specific certificates from the company Hackmanit

```
SSLVerifyClient none
SSLVerifyDepth  10

<Location "/secure">
     SSLVerifyClient require
     SSLVerifyDepth 10
     SSLRequireSSL
     SSLRequire %{SSL_CLIENT_S_DN_O}  eq „hackmanit"
</Location>
```

TLS 1.3


TLS Client
TLS Server
ClientHello
ClientKeyShare
EC (DH) public key
ServerHello
ServerKeyShare
EC (DH) public key
Certificate
CertificateVerify
RSA Signature over all previous messages
(Server-) Finished
(Client-) Finished