TLS 1.3

Team Building 2018 / Brno

Christian Heimes
Senior Software Engineer

christian@python.org / cheimes@redhat.com

@ChristianHeimes

Agenda & Takeaways

Agenda

- history
- high level view
- cryptography 101
- TLS handshake
- TLS 1.3
- books & resources

History

3 > 1

Secure Sockets Layer / Transport Layer Security

- SSL 1.0 – never released
- SSL 2.0 – 1995
- SSL 3.0 – 1996
- TLS 1.0 – 1999
- TLS 1.1 – 2006
- TLS 1.2 – 2008
- TLS 1.3 – ~~2014~~, ~~2015~~, ~~2016~~, ~~2017~~, 2018?

~~SSL~~

TLS

Attacks

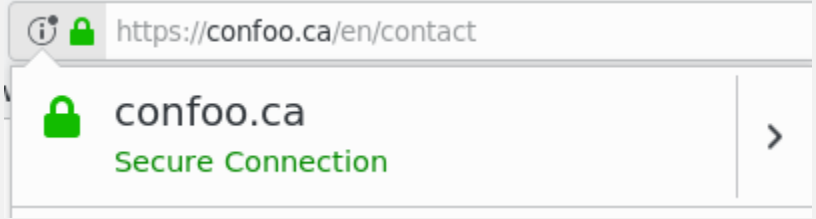
- **Bleichenbacher** (1998), **ROBOT** (2018), CCA2
- **Renegotiation** (2009)
- **BEAST** (2011), TLS 1.0 CBC IV
- **POODLE** (2014), padding oracle
- **CRIME**, **TIME**, **BREACH** (2012-13), compression
- **Heartbleed** (2014), OpenSSL
- **FREAK**, **Logjam** (2015), downgrade
- **SLOTH**, **SWEET32** (2016), weak crypto

RFC 7457 “Summarizing Known Attacks on TLS and DTLS”

Libraries

- OpenSSL
  - LibreSSL (OpenBSD, partly incompatible fork)
  - BoringSSL (Google, API incompatible fork)
- NSS (Mozilla Firefox)
- SChannel (Microsoft)
- Secure Transport (Apple)
- more: GnuTLS, Java JSSE, Go crypto/tls, kTLS

10,000 foot high level view


https://confoo.ca/en/contact
confoo.ca
Secure Connection

“TLS”

=

”encryption”

Wikipedia definition

Transport Layer Security (TLS) – and its predecessor, Secure Sockets Layer (SSL) – are cryptographic protocols that provide ***communications security*** *over a computer network. The TLS protocol aims primarily to provide* ***privacy*** *and* ***data integrity*** *between two communicating computer applications.*

TLS core features

- encrypted transport stream
- application protocol agnostic
- integrity check
- replay attack protection
- strong authentication of server
- strong authentication of client (optional)
- extensible protocol

TLS standard

- IETF standard (Internet engineering task force)
- IANA (Internet assigned number authority)
- TLS (TCP) / DTLS (UDP)
- ASN.1
- PKI with X.509 certificates

Crypto 101

redhat.

Cryptographically secure hash functions

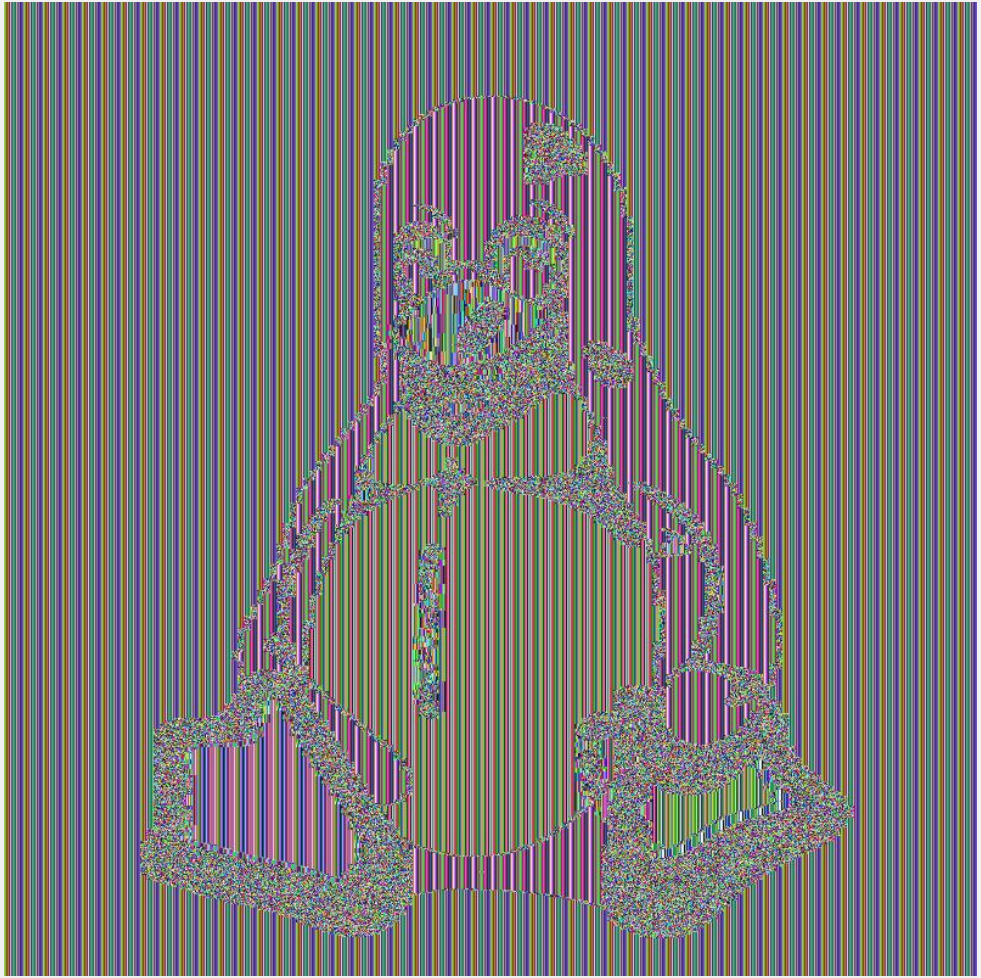
Hash functions for MAC, signatures, and more.

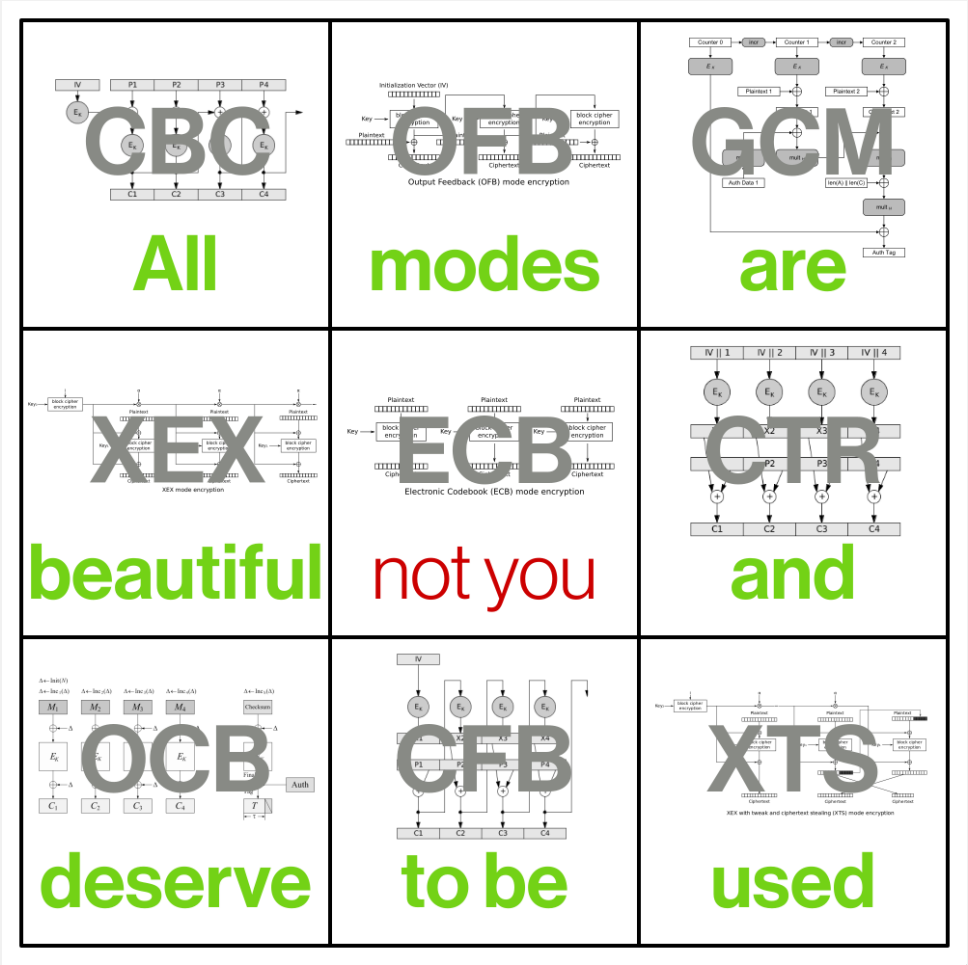
- MD5
- SHA (SHA1)
- SHA2
  - SHA-256
  - SHA-384

Symmetric-key algorithm (bulk encryption)

Same key for encryption and decryption.

- DES
- Triple DES (3DES)
- RC4
- AES (AES-128, AES-256)
- CHACHA20




CBC
All
OFB
modes
GCM
are
XEX
beautiful
ECB
not you
CTR
and
OCB
deserve
CFB
to be
XTS
used

Symmetric-key algorithm (2)

- Padding
- Mode of operation
  - CBC
  - GCM
- Authenticated encryption
  - CBC with MtE
  - AEAD (GCM, Poly1305)

Asymmetric cryptographic algorithms

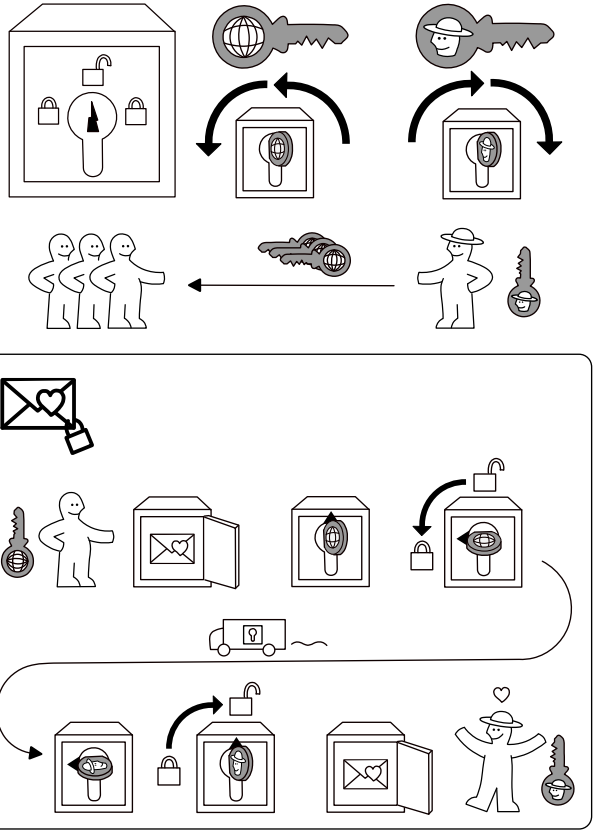
public / private key cryptography

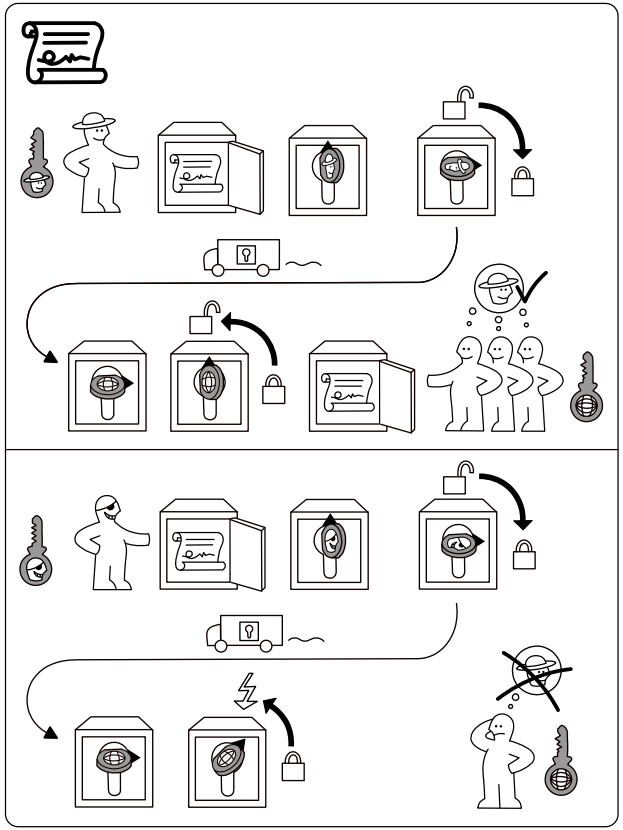
- asymmetric encryption
- signatures
- key agreement

PUBLIC KEY KRÜPTO

idea-instructions.com/public-key/

IDEA





Asymmetric encryption

public key encrypts, private key decrypts

- ElGamal encryption
- RSA encryption (PKCS#1)
  - RSAES-PKCS1-v1.5
  - RSAES-OAEP

Asymmetric signatures

private key signs hash of message message, public key verifies

- RSA signature (PKCS#1)
  - RSASSA-PKCS1-v1.5
  - RSAES-PSS
- DSS (DSA)
- ECDSA (secp256r1, secp384r1, …)
- EdDSA (Edward Curve25519, …)

Key agreement protocol

own private key + peer's public = key

- finite field Diffie-Hellman (DH)
- elliptic curve Diffie-Hellman (ECDH)
- ephemeral DH / ECDH

Misc

- random numbers generator (CPRNG)
- HMAC
- Key Derivation Function (KDF)
- Key Wrapping (KW)
- …

Cryptographic building blocks

- key agreement / exchange
- authentication algorithm
- bulk encryption (symmetric)
- cipher mode
- one-way function for message authentication (MAC)

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

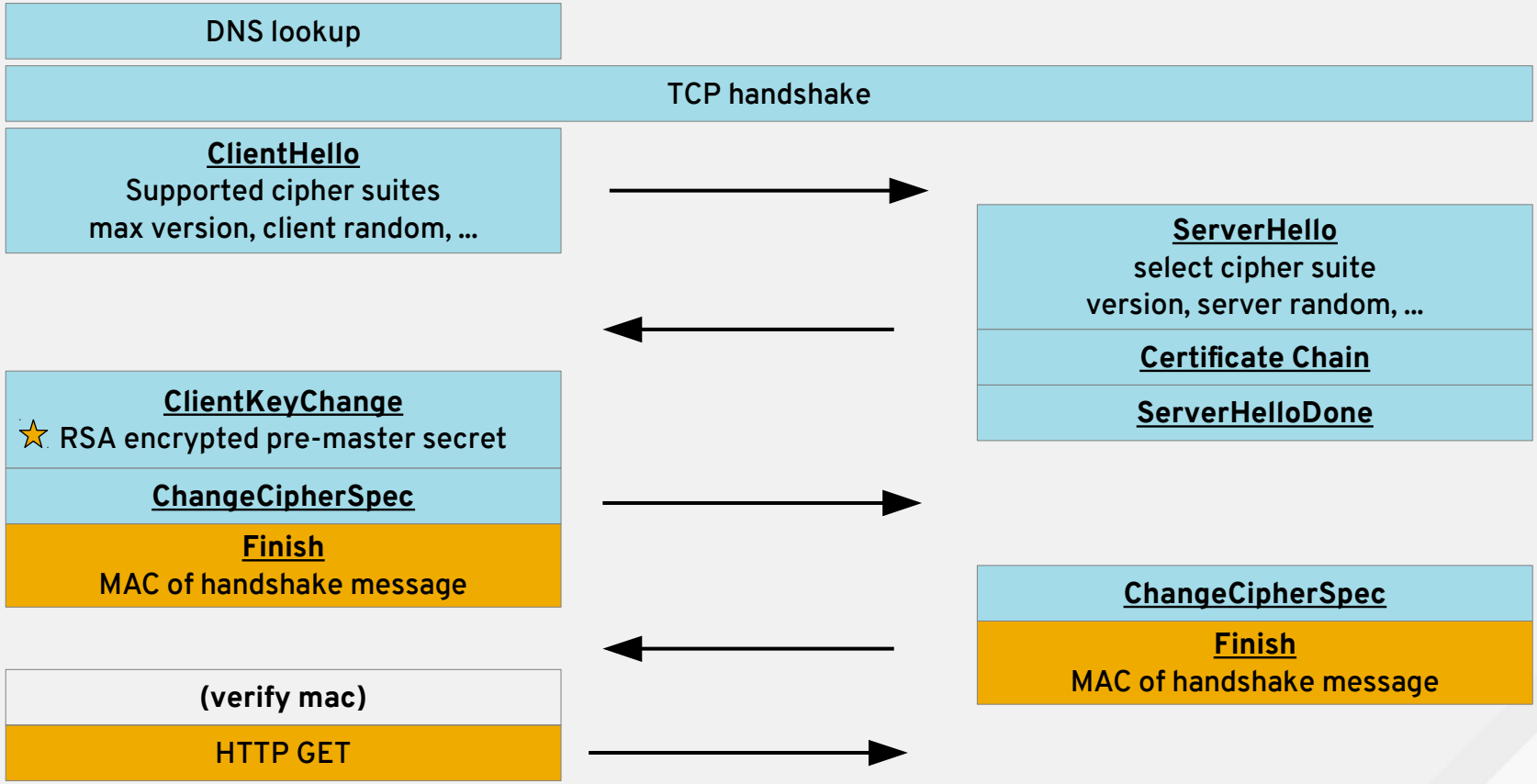
TLS_RSA_WITH_3DES_EDE_CBC_SHA

TLS_DHE_PSK_WITH_CHACHA20_POLY1305_SHA256

SSL_RSA_WITH_NULL_MD5

TLS handshake

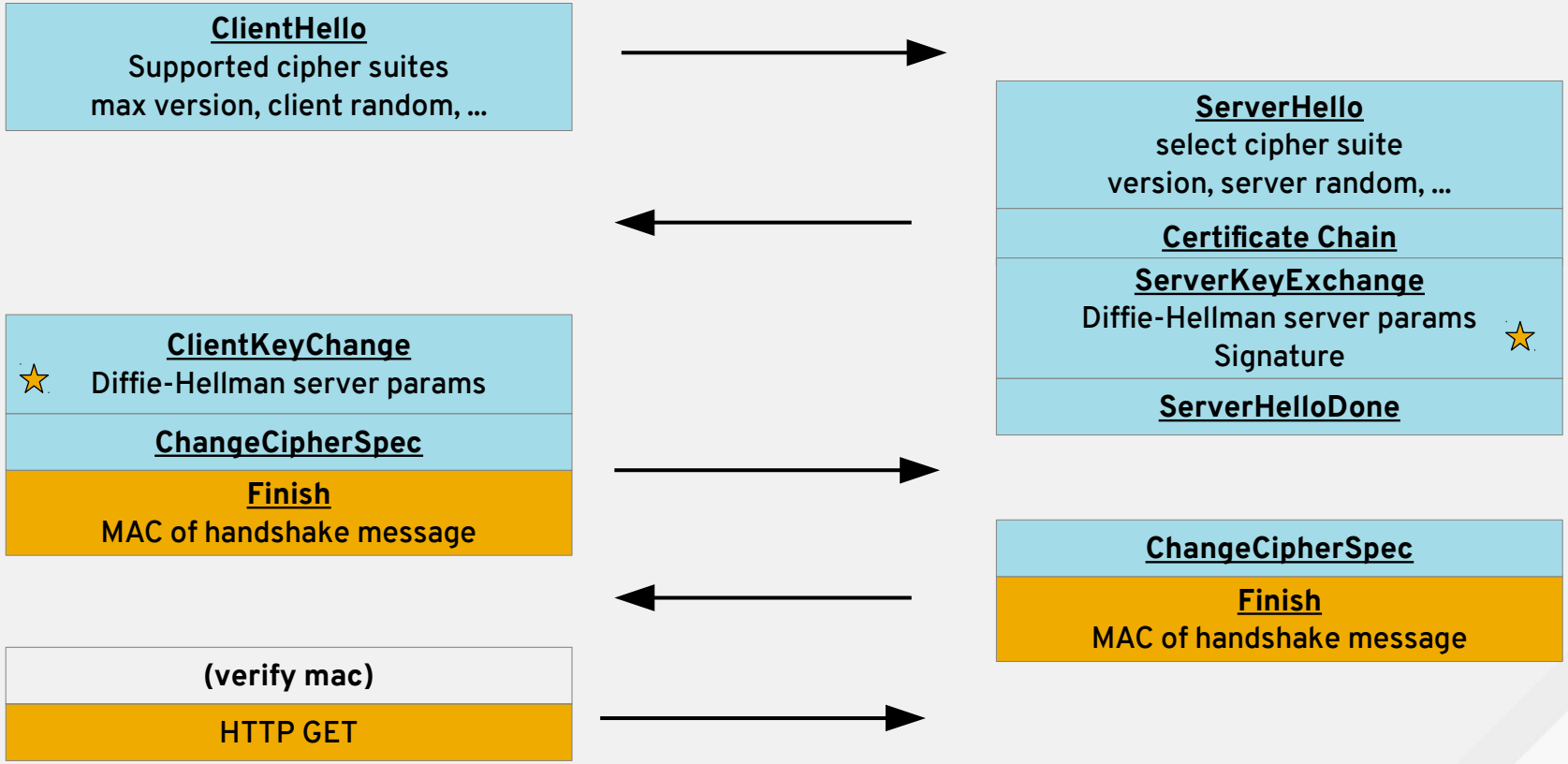
TLS handshake with RSA key exchange



TLS handshake with RSA key exchange

- ✔ negotiate TLS version
- ✔ negotiate cipher suite
- ✔ validate server cert chain
- ✔ replay protection: MAC client/server random
- ✘ **no forward secrecy**

TLS handshake with Diffie-Hellman



Ephemeral Diffie-Hellman

- ✔ negotiate TLS version
- ✔ negotiate cipher suite
- ✔ validate server cert chain
- ✔ replay protection: MAC client/server random
- ✔ perfect forward secrecy
- ✘ **actually no PFS…**

TLS
extensions & alerts

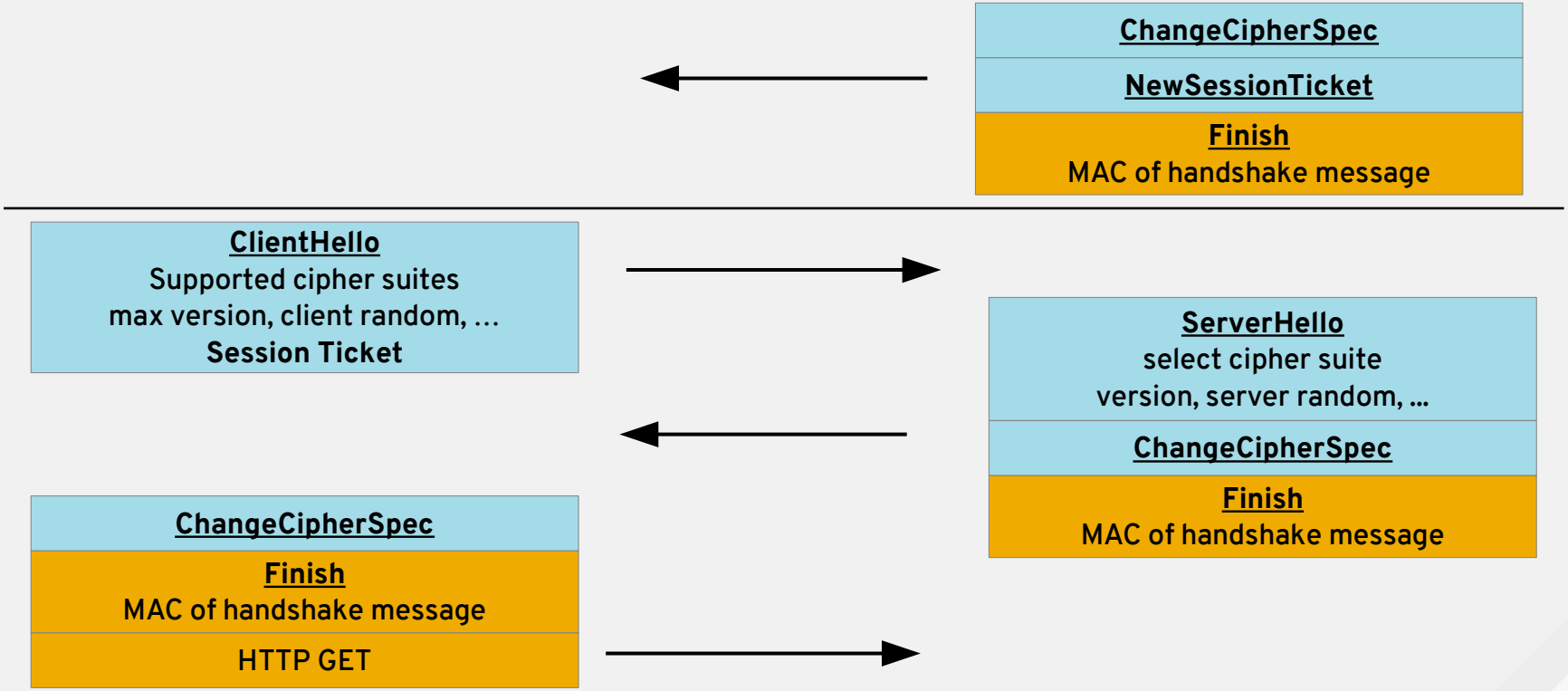
TLS alert

- close_notify
- handshake failure
  - no ciphers available
- unknown_ca
- decryption_failed
- …

TLS extensions

- heart beat
- server name indication (virtual hosting)
- ALPN (HTTP/2)
- session resumption
- signature algorithms
- supported groups (ECDH curves)

TLS handshake with session resumption



Session resumption

- session ticket contains encrypted key
  - ticket encrypted with server's **Session Ticket Key**
  - STK is a shared key
- browsers usually request session resumption
  - server sends ticket in first request
- bad design
  - ticket is not TLS encrypted
  - **current** master key!

TLS 1.3

redhat.

TLS 1.3 is a major change

```
Versions: (draft-ietf-tls-rfc5246-bis)  00 01
          02 03 04 05 06 07 08 09 10 11 12 13
          14 15 16 17 18 19 20 21 22 23 24

Network Working Group                                    E. Rescorla
Internet-Draft                                            RTFM, Inc.
Obsoletes: 5077, 5246 (if approved)                   February 15, 2018
Updates: 4492, 5705, 6066, 6961 (if
         approved)
Intended status: Standards Track
Expires: August 19, 2018


        The Transport Layer Security (TLS) Protocol Version 1.3
                        draft-ietf-tls-tls13-24

Abstract

   This document specifies version 1.3 of the Transport Layer Security
   (TLS) protocol.  TLS allows client/server applications to communicate
   over the Internet in a way that is designed to prevent eavesdropping,
   tampering, and message forgery.
```

Kill all the bad crypto!

- RC4, 3DES, AES-CBC
- MD5, SHA1
- NULL ciphers
- arbitrary DH groups and curves
- static RSA authentication
- renegotiation
- compression
- PKCS#1 v1.5
- MAC then Encrypt

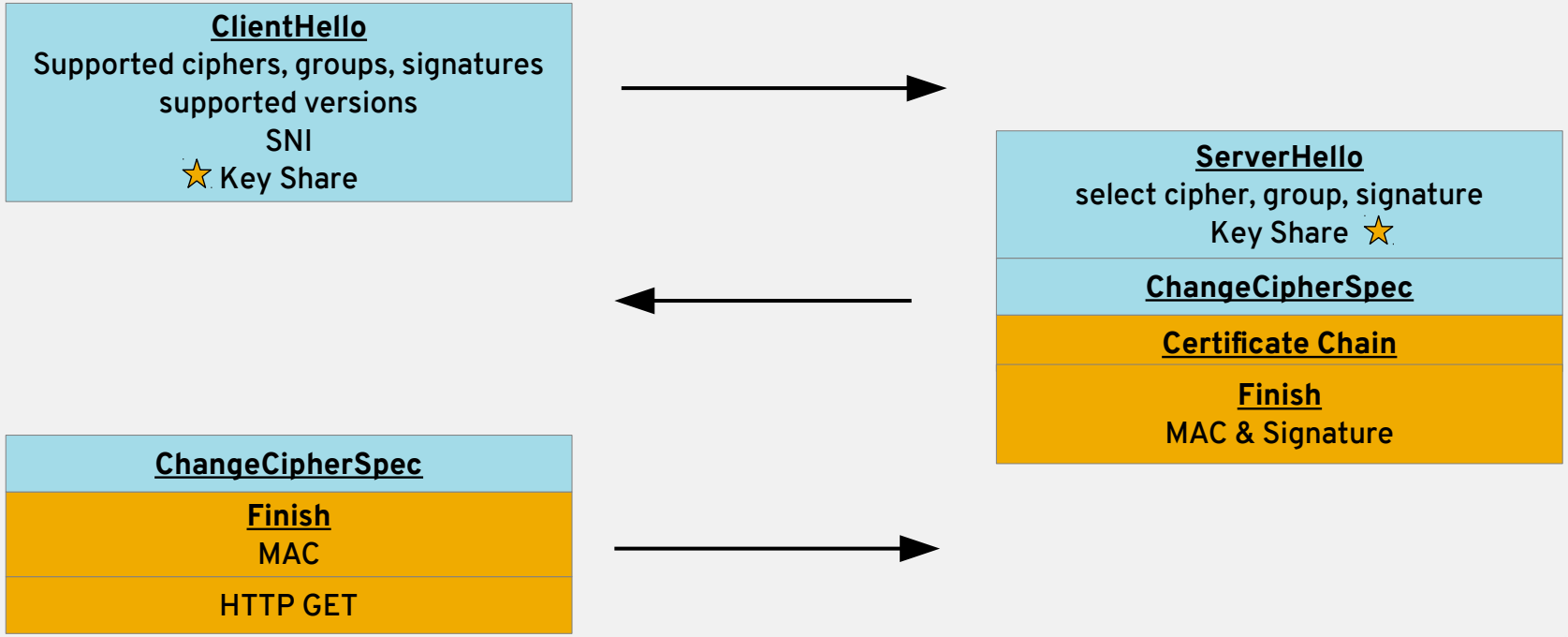
New modern crypto

- Elliptic Curve Crypto
  - Edwards Curve (Ed25519)
  - Curve25519, X25519
- authenticated encryption (AEAD)
  - AES-GCM
  - CHACHA20-Poly1305
- mandatory Diffie-Hellman → PFS
  - no “out-of-band TLS decryption”
- RSAES-PSS signatures

Protocol improvements

- Cipher negotiation protected by *Finish* MAC (LogJam)
- Separation of key agreement, ciphers, authentication
- Cipher suites
  - TLS_AES_128_GCM_SHA256, TLS_CHACHA20_POLY1305_SHA256
- TLS extensions
  - signature_algorithms, cert_signature_algorithms, elliptic_curves
- Session resumption with PSK-ECDHE
  - after *Finish*, TLS encrypted, next master key
- Encrypted Post-Handshake auth (client cert)

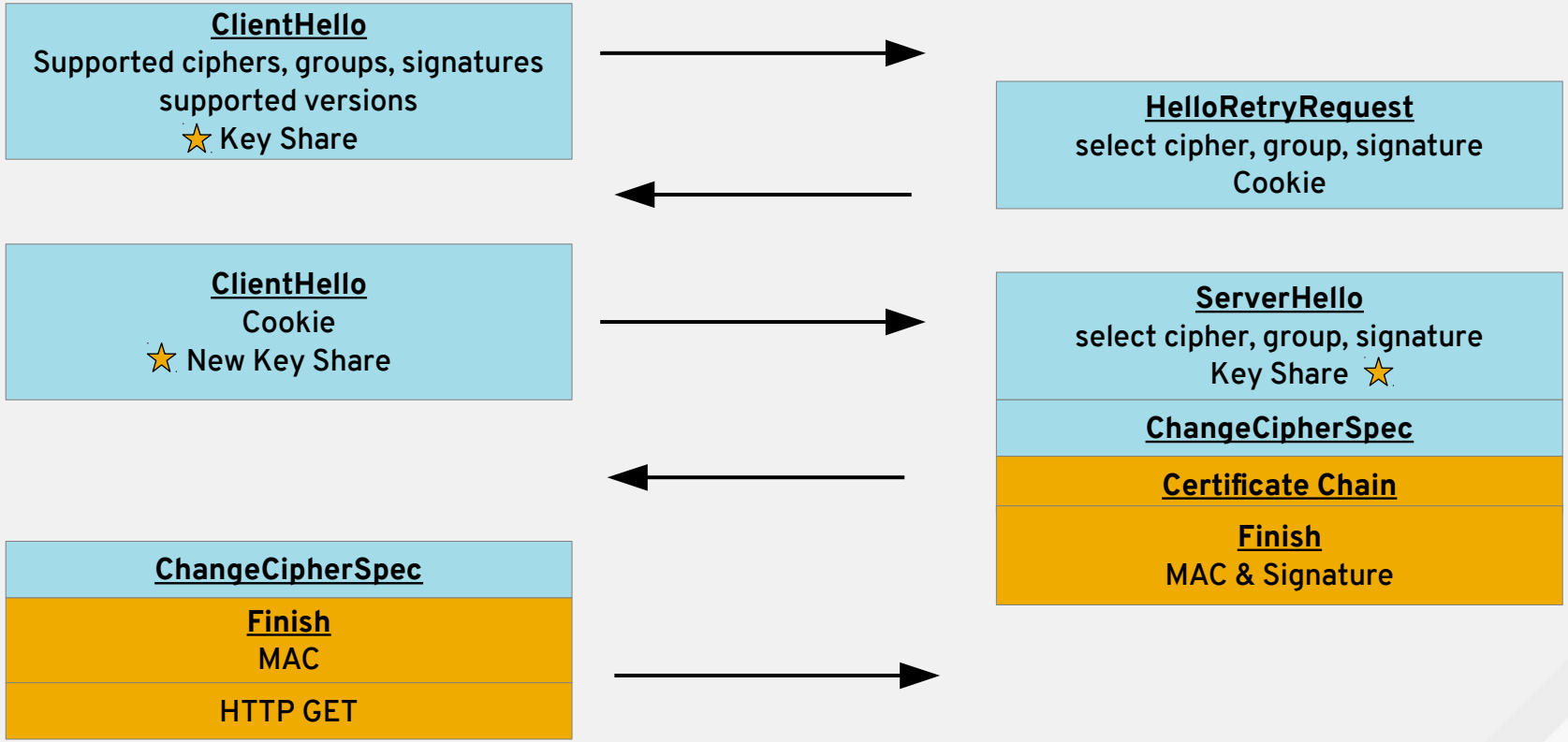
TLS 1.3 with 1-RTT handshake



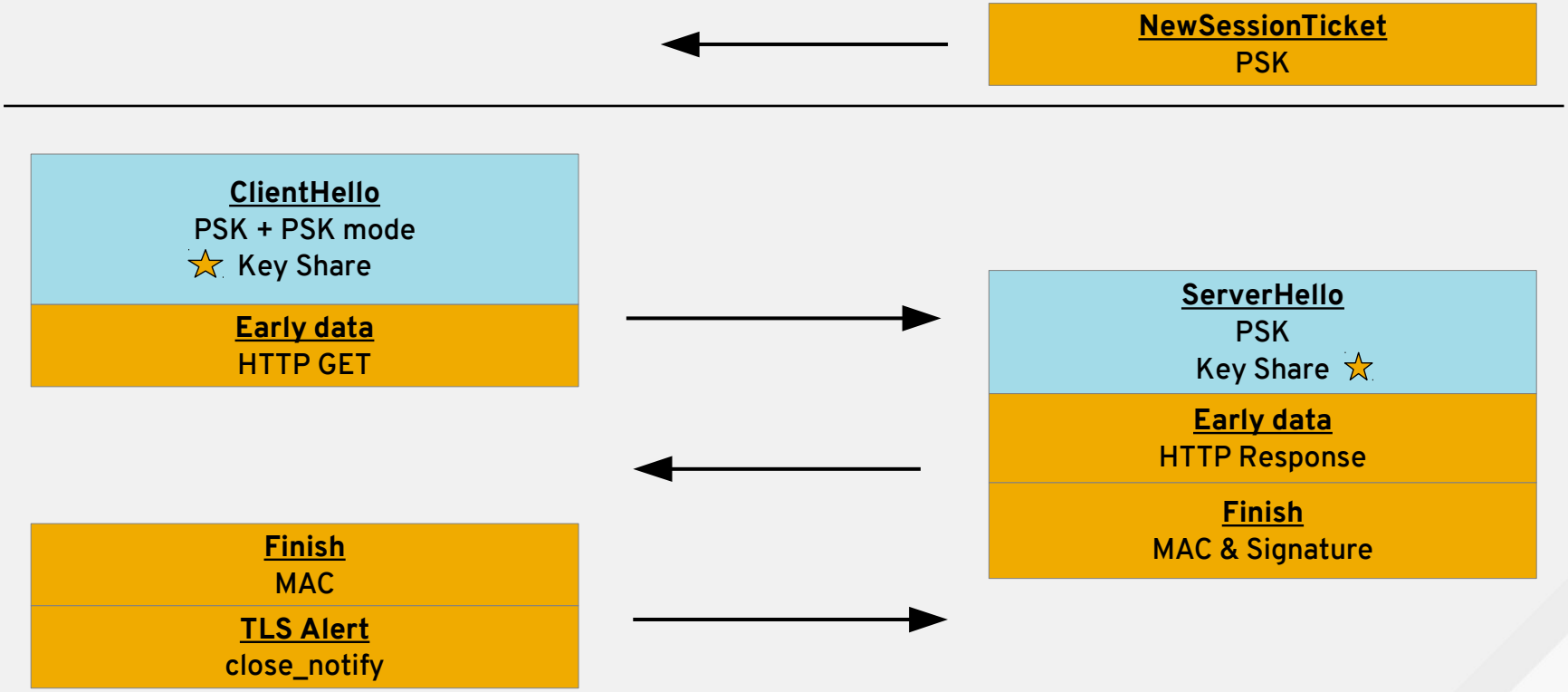
Hacks

- Middlebox compatibility mode
  - Version: TLSv1.2
  - TLS ext: supported_version `0x304`
- Downgrade protection in server random
  - = TLSv1.2 `DOWNGRD\x00`
  - < TLSv.1.2 `DOWNGRD\x01`
- HelloRetryRequest random `SHA256("HelloRetryRequest")`

TLS 1.3 with 2-RTT retry handshake



TLS 1.3 with 0-RTT early data



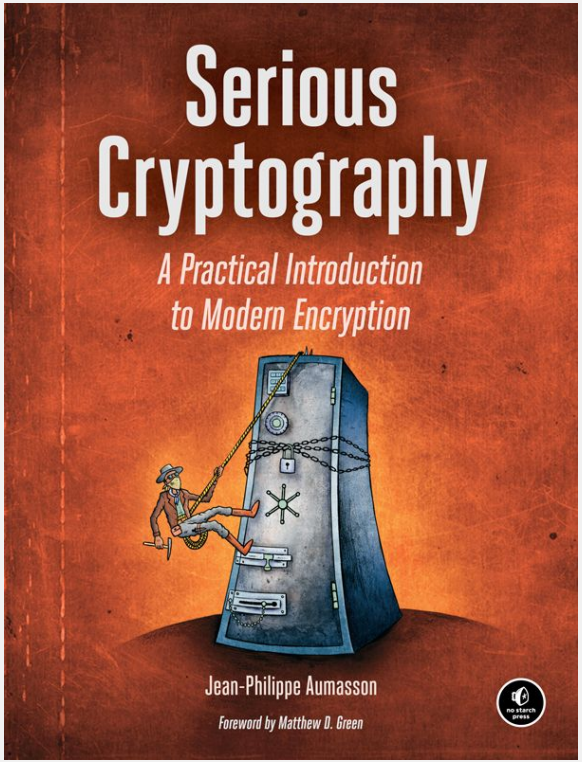
0-RTT caveats

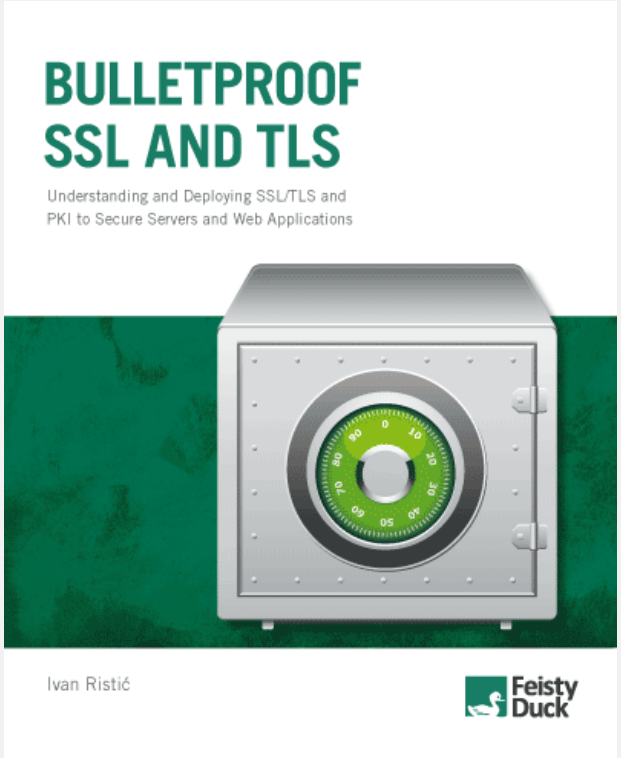
- Replay attack
- No forward secrecy

Applications must define a profile for early data

Resources

Books


Serious Cryptography
A Practical Introduction to Modern Encryption
Jean-Philippe Aumasson
Foreword by Matthew D. Green
no starch press


BULLETPROOF SSL AND TLS
Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications
Ivan Ristić
Feisty Duck

Resources

- https://www.ssllabs.com/ssltest/
- https://istlsfastyet.com/
- Deploying TLS 1.3: the great, the good and the bad (33c3)
  https://www.youtube.com/watch?v=0opakLwtPWk

Let's Encrypt

FreeIPA
Open Source Identity Management Solution

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos