DEPLOYING TLS 1.3: THE GREAT, THE GOOD AND THE BAD

IMPROVING THE ENCRYPTED THE WEB, ONE ROUND-TRIP AT A TIME

FILIPPO VALSORDA, NICK SULLIVAN

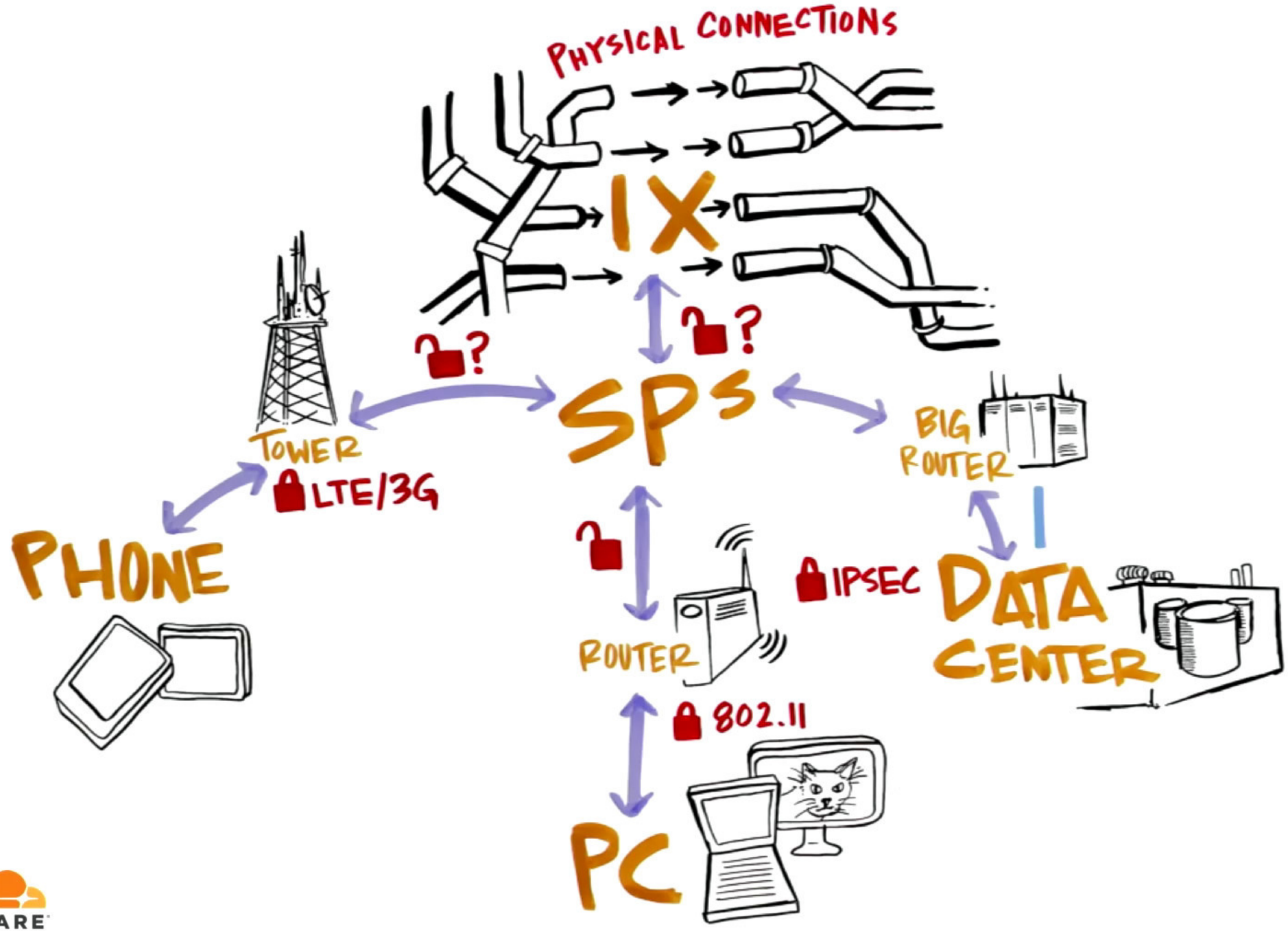
#8348

TLS 1.3

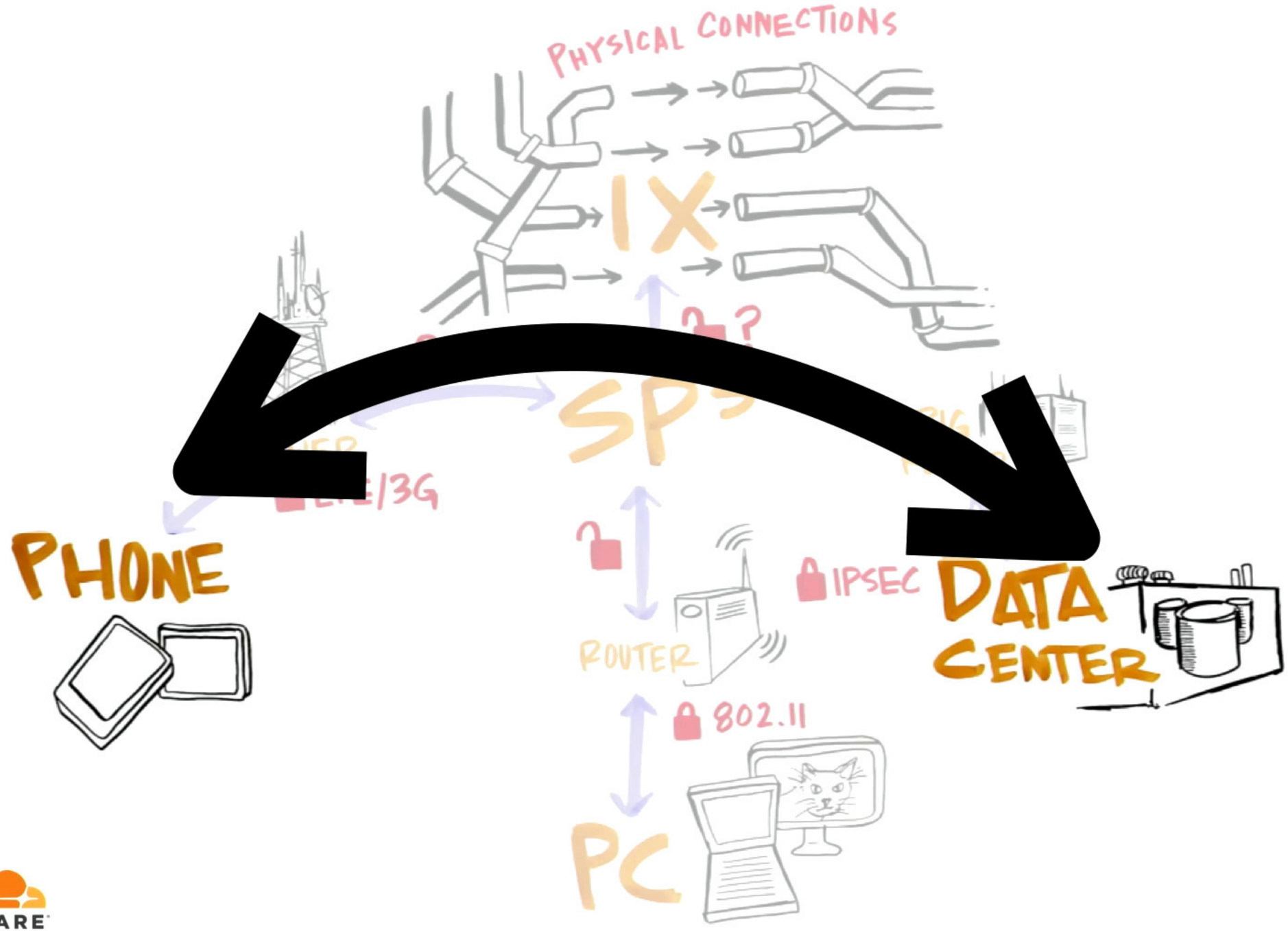
Nick Sullivan
@grittygrease

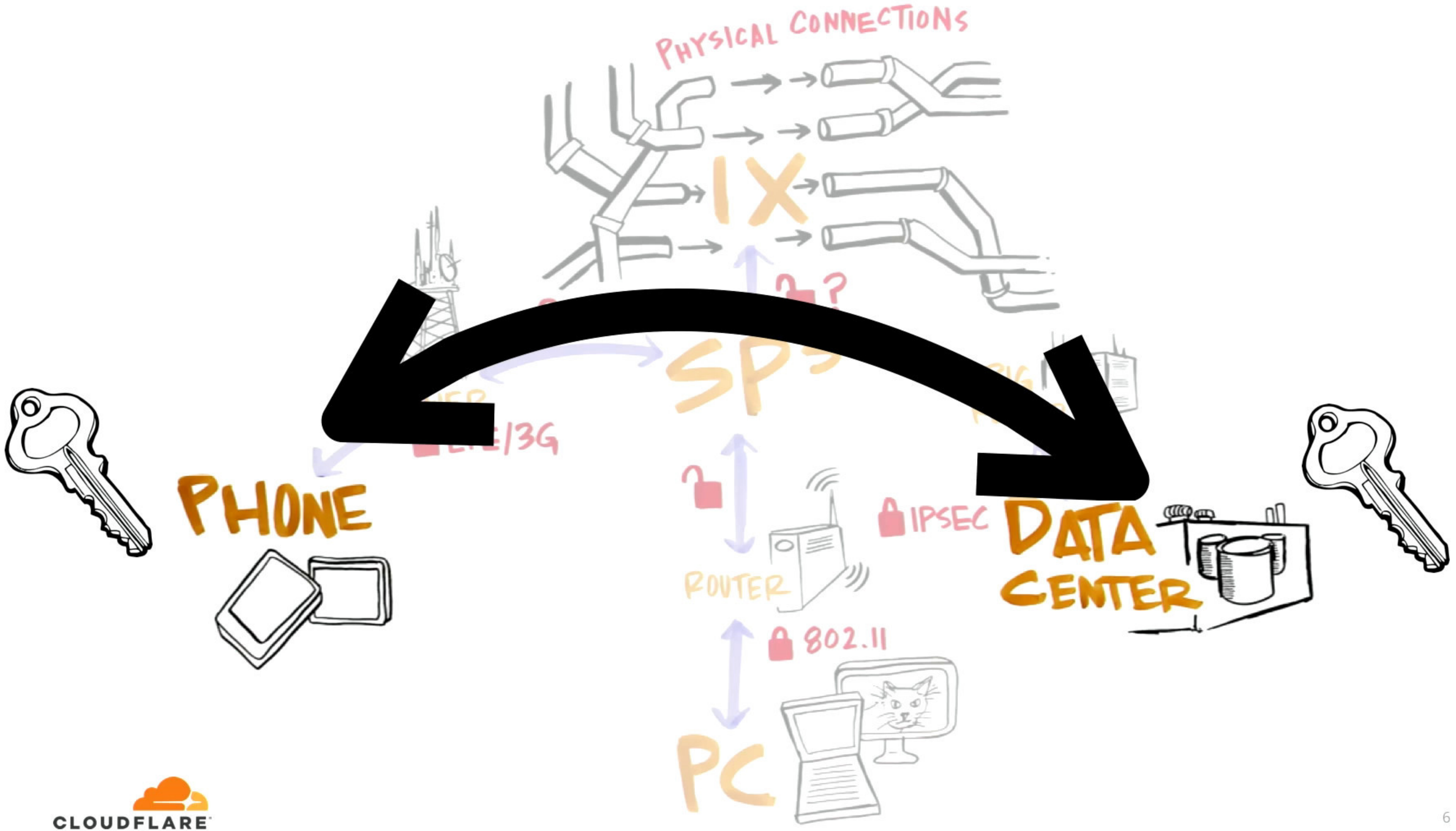
Filippo Valsorda
@FiloSottile

CLOUDFLARE


PHYSICAL CONNECTIONS
IX
?
?
SPS
TOWER
LTE/3G
PHONE
BIG ROUTER
DATA CENTER
IPSEC
ROUTER
802.11
PC

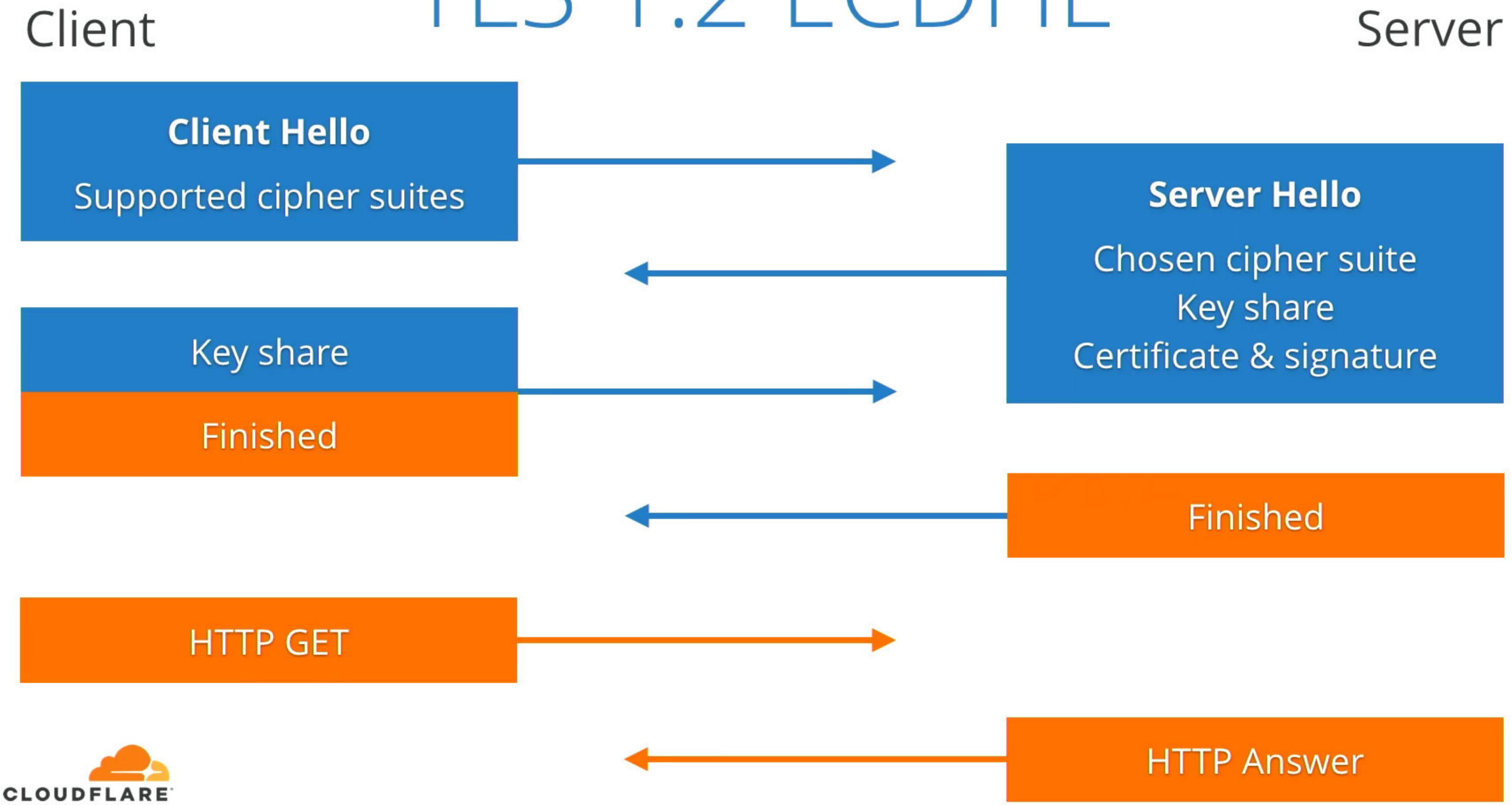
CLOUDFLARE


PHYSICAL CONNECTIONS
IX
?
SP
LTE/3G
PHONE
ROUTER
IPSEC
DATA CENTER
802.11
PC


PHYSICAL CONNECTIONS
IX
?
SP
LTE/3G
PHONE
IPSEC
DATA CENTER
ROUTER
802.11
PC

CLOUDFLARE

TLS 1.2 ECDHE



CLOUDFLARE

TLS 1.2 ECDHE

Time	Source	Destin	Protocol	Lengtl	Info
1051.721834	10.1…	104…	TLSv1.2	251	Client Hello
1051.747907	104.…	10.…	TLSv1.2	1404	Server Hello
1051.747993	104.…	10.…	TLSv1.2	811	CertificateCertificate Status, Server Key Exchange, Server Hello Done
1051.749915	10.1…	104…	TLSv1.2	170	Client Key Exchange, Change Cipher Spec, Hello Request, Hello Request
1051.774692	104.…	10.…	TLSv1.2	302	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
1051.774846	104.…	10.…	TLSv1.2	113	Application Data
1052.104322	10.1…	104…	TLSv1.2	97	Application Data
1052.104447	10.1…	104…	TLSv1.2	100	Application Data
1052.104574	10.1…	104…	TLSv1.2	86	Application Data
1052.104984	10.1…	104…	TLSv1.2	82	Application Data
1052.105458	10.1…	104…	TLSv1.2	540	Application Data
1052.128825	104.…	10.…	TLSv1.2	82	Application Data
1052.132494	104.…	10.…	TLSv1.2	449	Application Data
1052.132561	104.…	10.…	TLSv1.2	82	Application Data

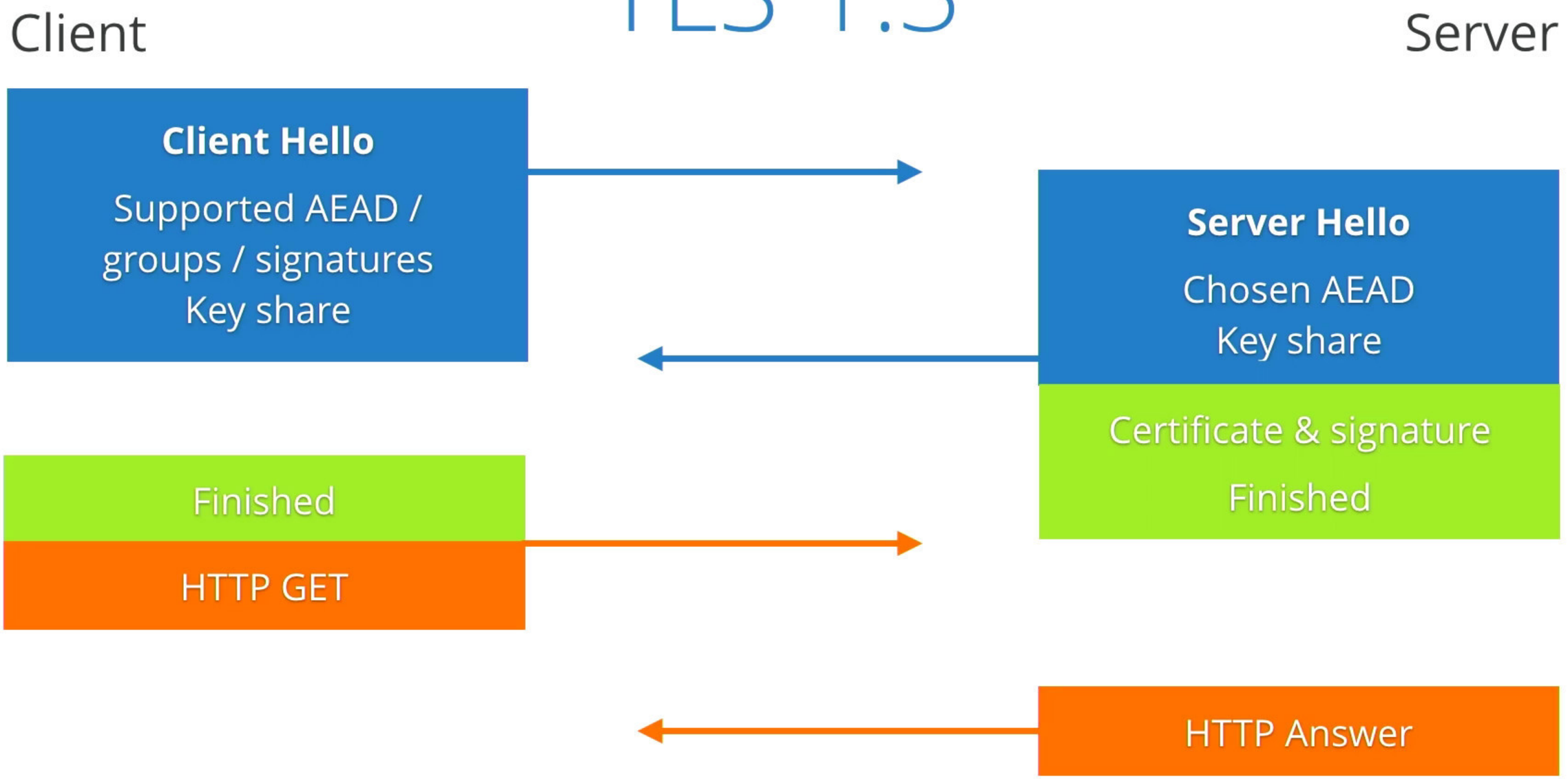
CLOUDFLARE

TLS 1.0, 1.1 and 1.2 are not that different

TLS 1.3 is a BIG jump

CLOUDFLARE

TLS 1.3



CLOUDFLARE

TLS 1.3

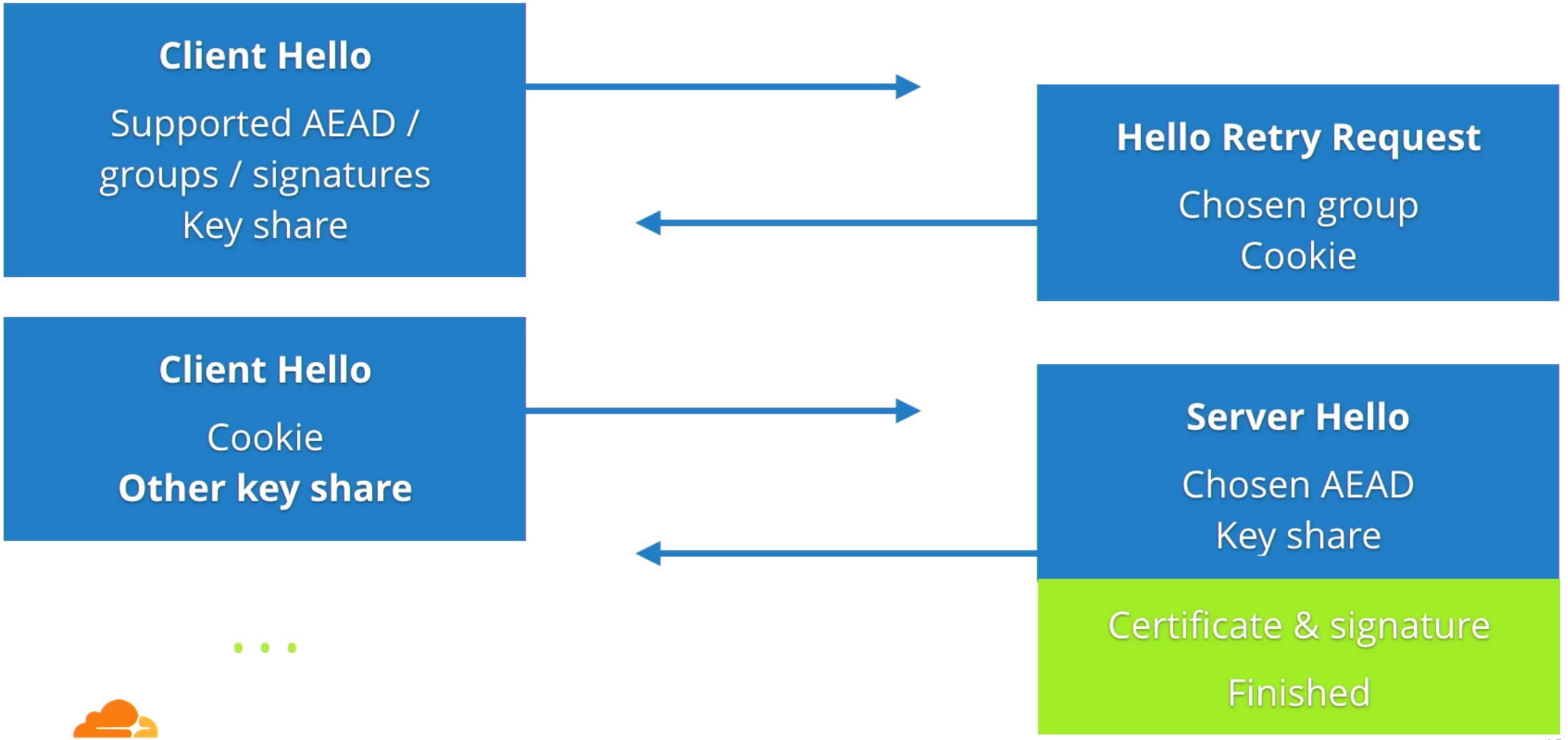
Time	Source	Destin	Protocol	Lengtl	Info
3.819792	10.1…	104…	TLSv1.3	561	Client Hello
3.848558	104.…	10.…	TLSv1.3	1404	Server Hello, Application Data
3.849548	104.…	10.…	TLSv1.3	580	Application DataApplication Data, Application Data
3.849608	104.…	10.…	TLSv1.3	93	Application Data
3.851004	10.1…	104…	TLSv1.3	102	Application Data
3.875635	104.…	10.…	TLSv1.3	221	Application Data

CLOUDFLARE

Hello Retry Request

Client

Server



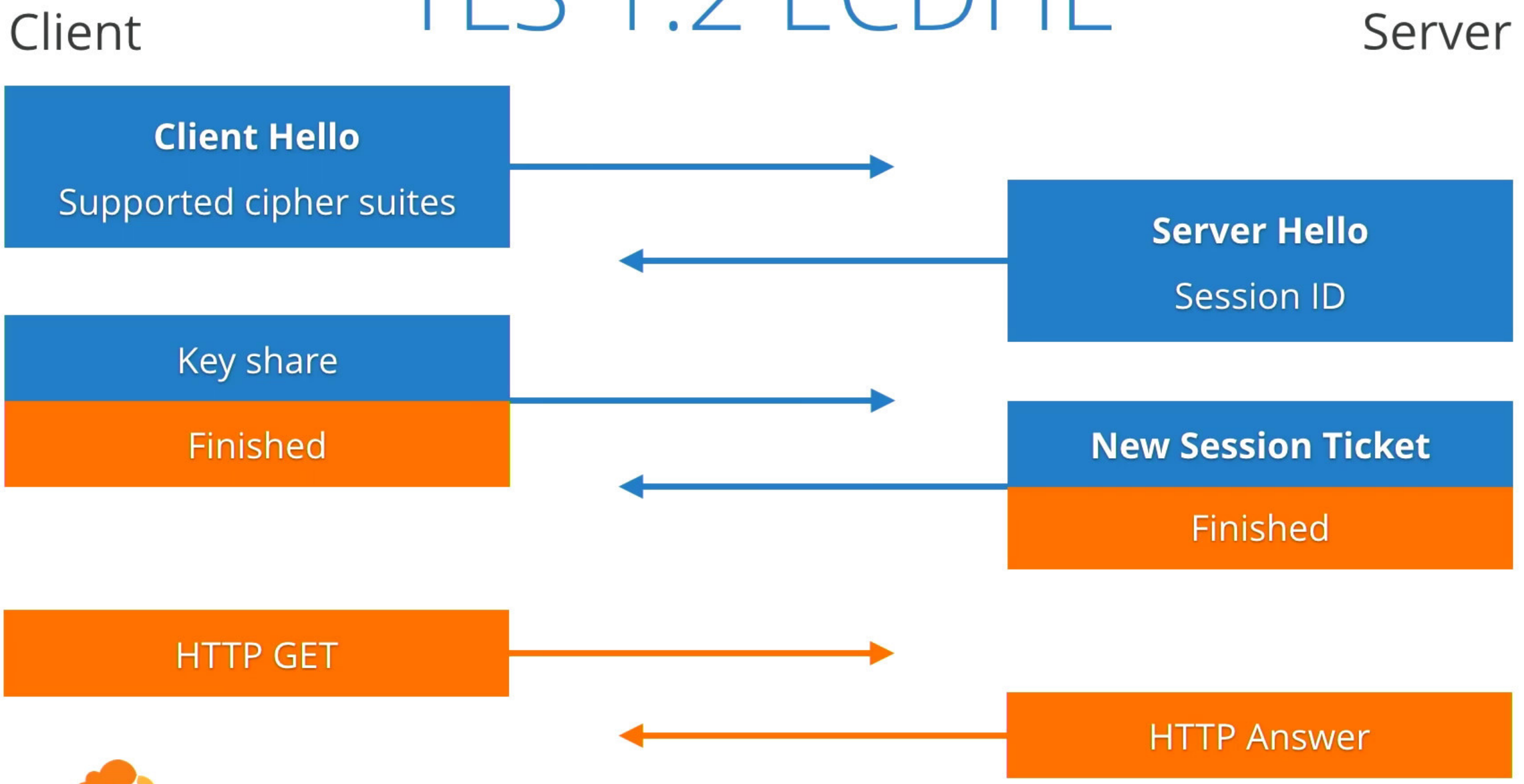
CLOUDFLARE

Resumption

"Hey, I know you!"

CLOUDFLARE

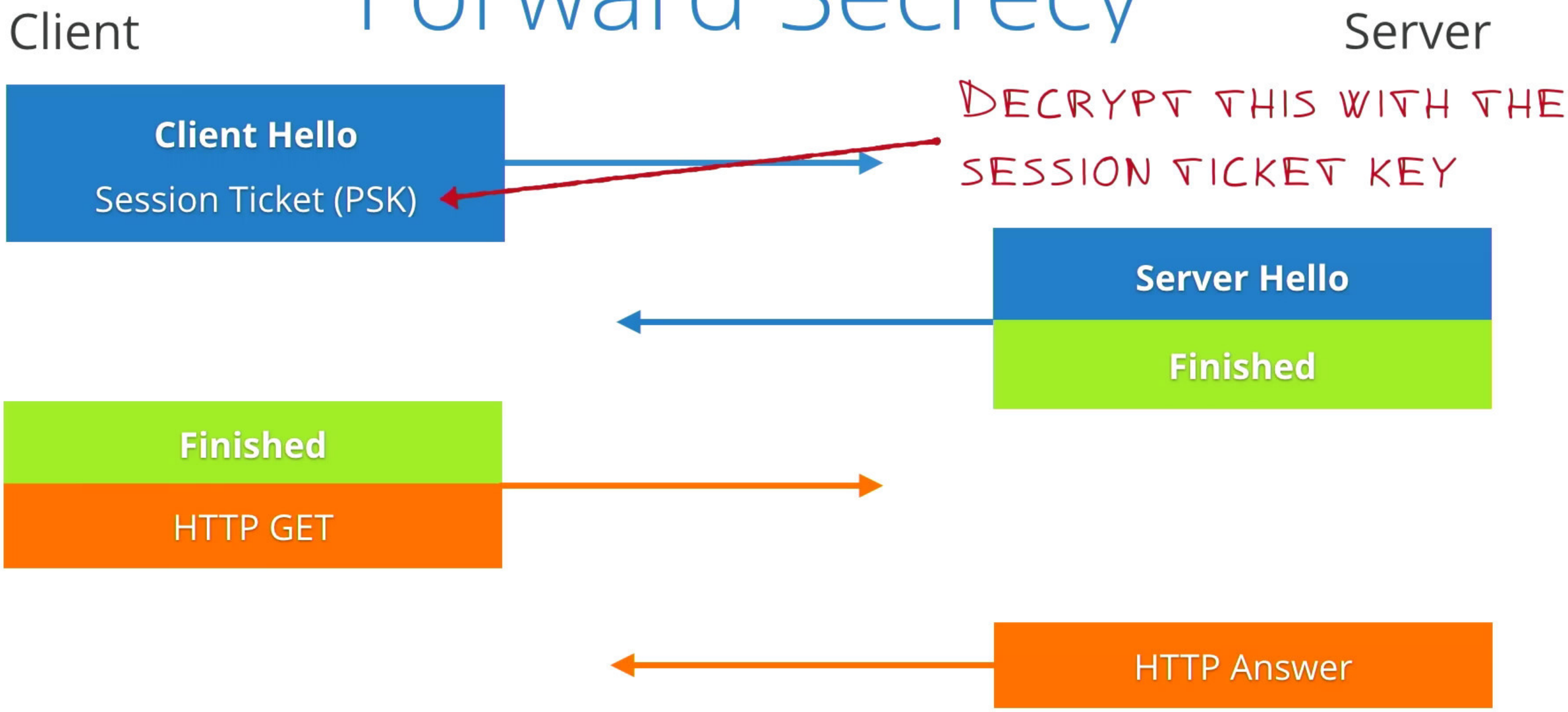
TLS 1.2 ECDHE



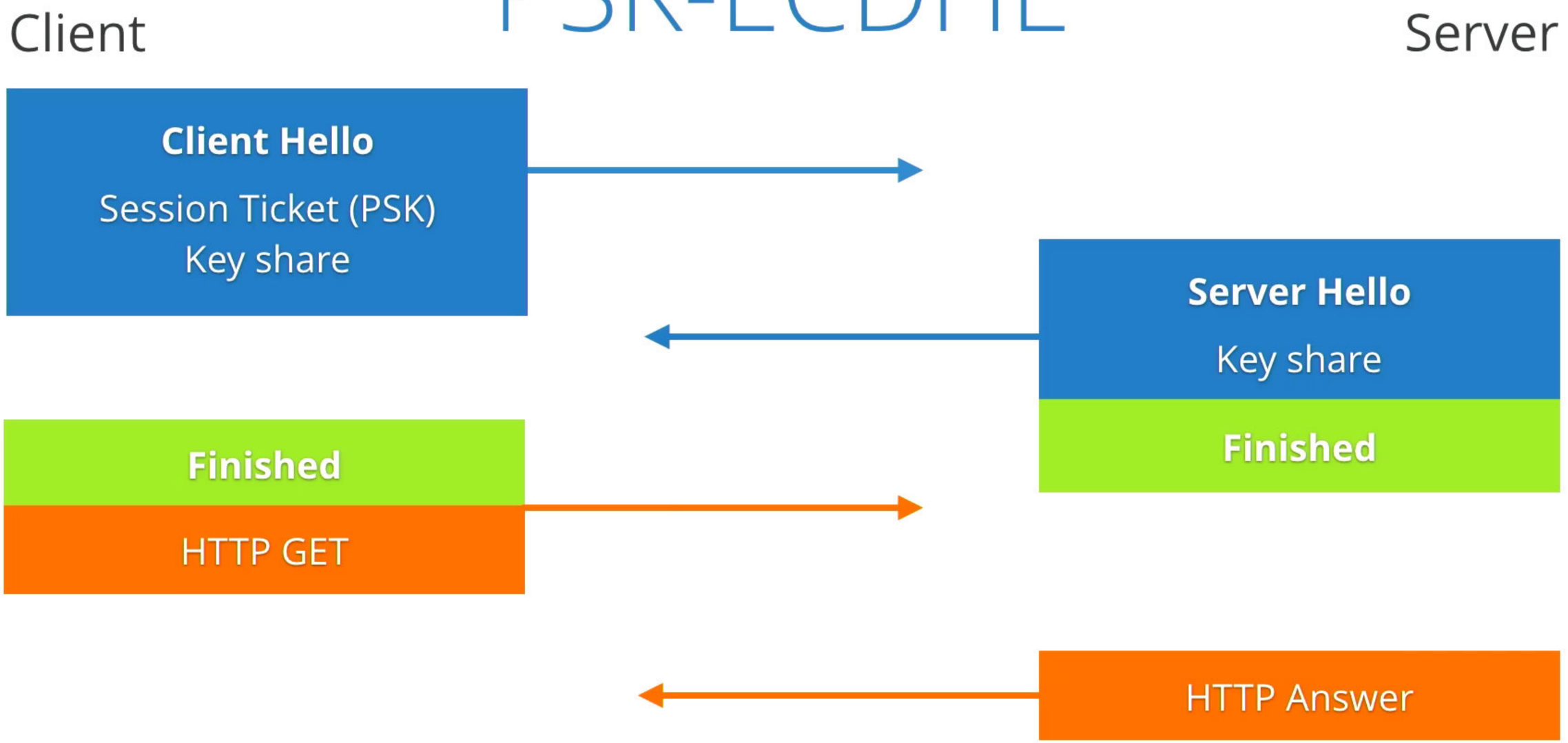
CLOUDFLARE

TLS 1.2 Resumption

Forward Secrecy



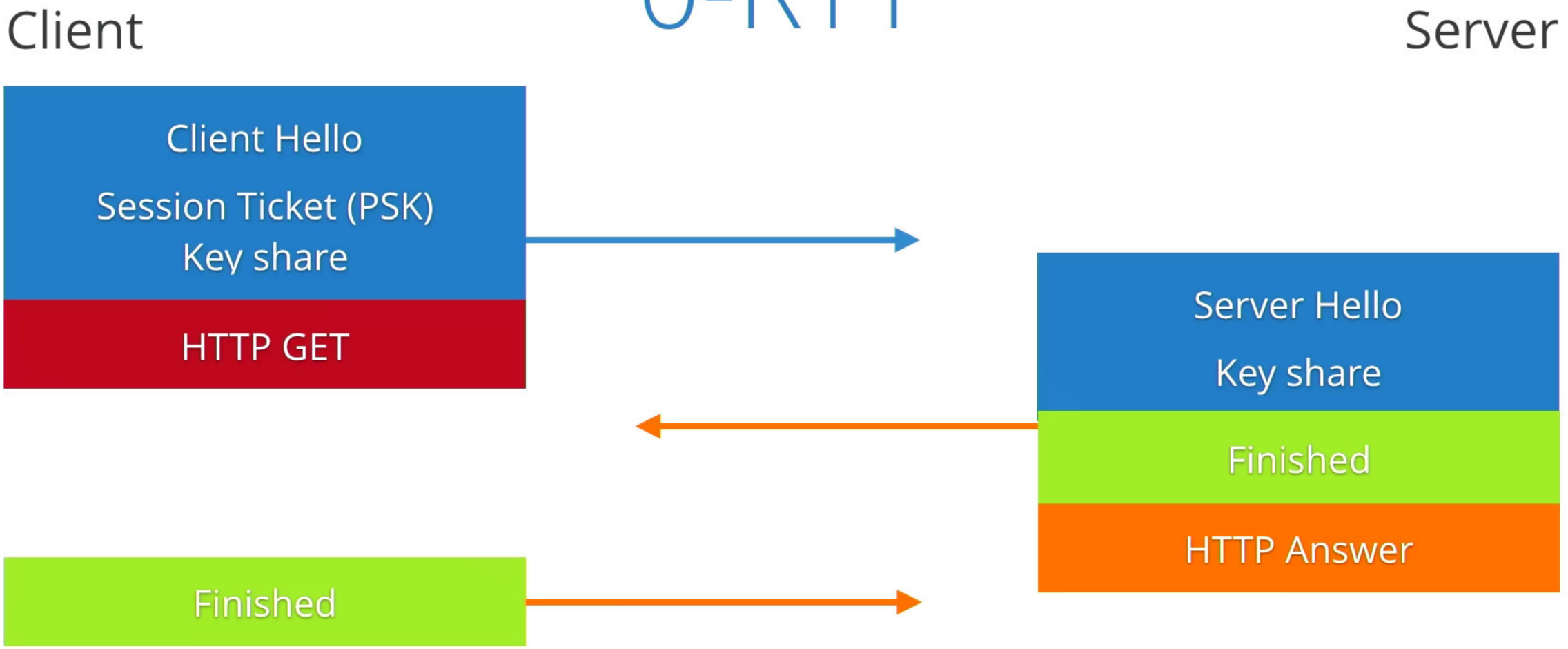
PSK-ECDHE


Client
Server
Client Hello
Session Ticket (PSK)
Key share
Server Hello
Key share
Finished
Finished
HTTP GET
HTTP Answer

0-RTT!

CLOUDFLARE

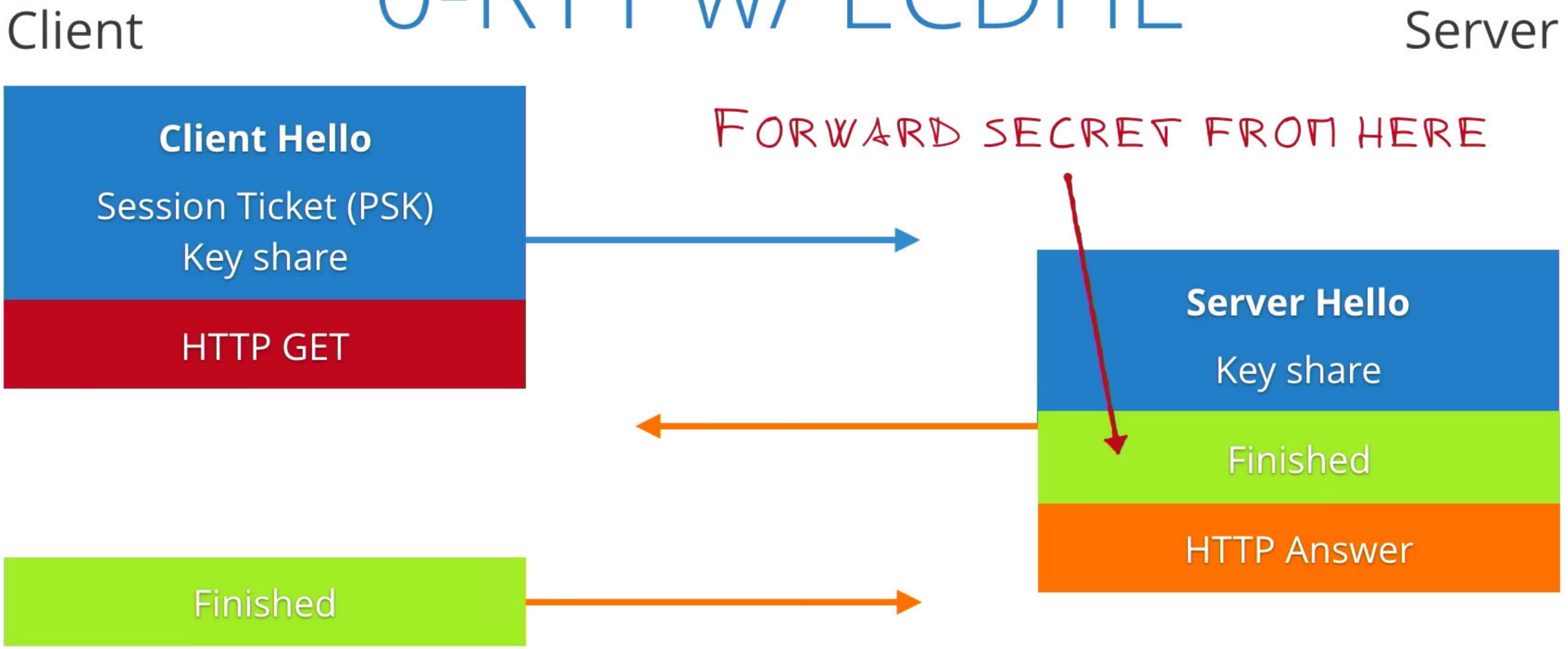
0-RTT


Client
Server
Client Hello
Session Ticket (PSK)
Key share
HTTP GET
Server Hello
Key share
Finished
HTTP Answer
Finished

0-RTT

No PSK-ECDHE

0-RTT w/ ECDHE


Client
Server
Client Hello
Session Ticket (PSK)
Key share
HTTP GET
FORWARD SECRET FROM HERE
Server Hello
Key share
Finished
HTTP Answer
Finished

TLS 1.2 is forward secret:

- Relatively to the certificate: **always** (using ECDHE)
- Relatively to the ticket key: **never**

TLS 1.3 is forward secret:

- Relatively to the certificate: **always**
- Relatively to the ticket key: **except 0-RTT early data** (w/ PSK-ECDHE)

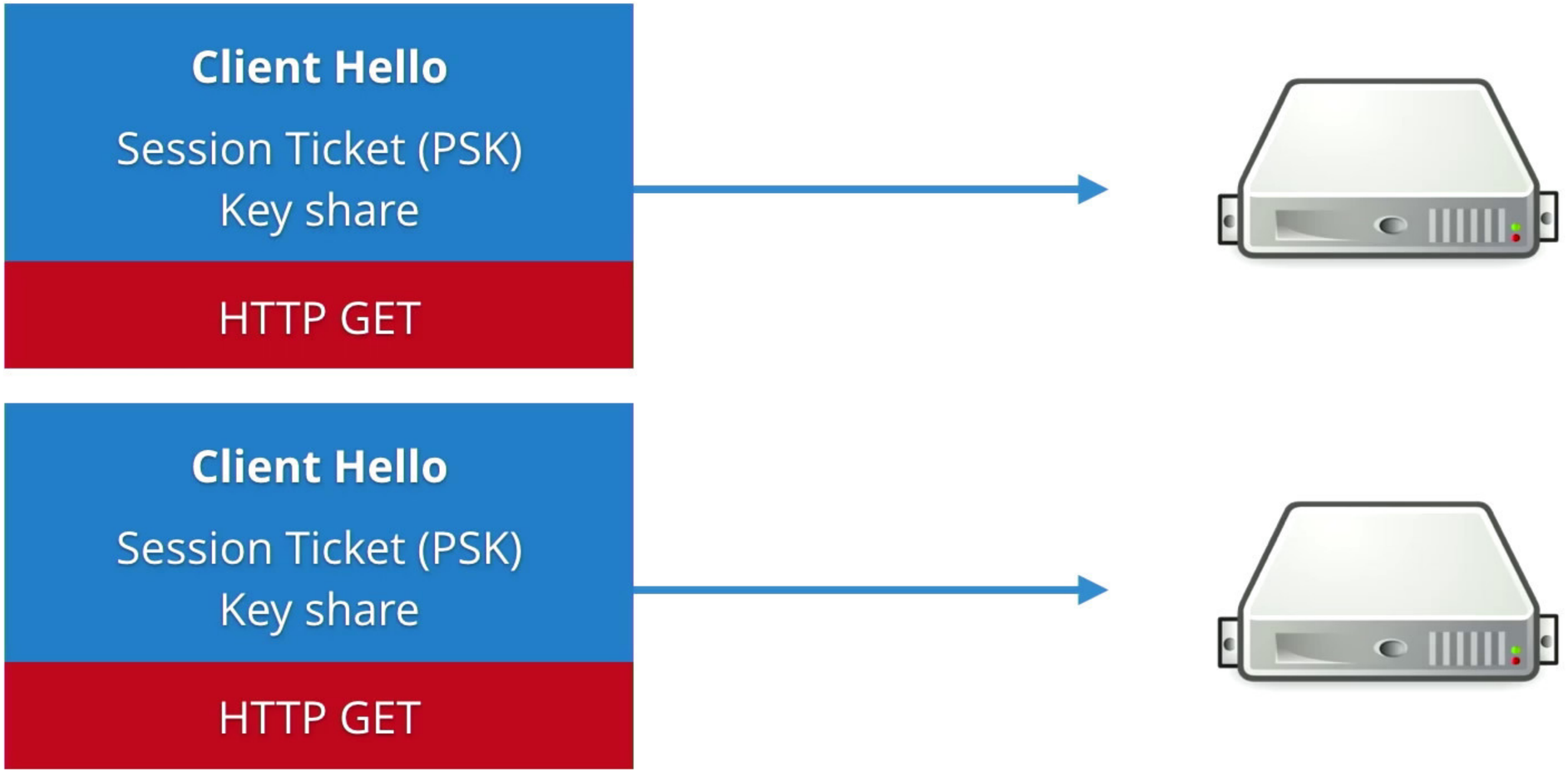
CLOUDFLARE

0-RTT

Replays

CLOUDFLARE

0-RTT replay



CLOUDFLARE

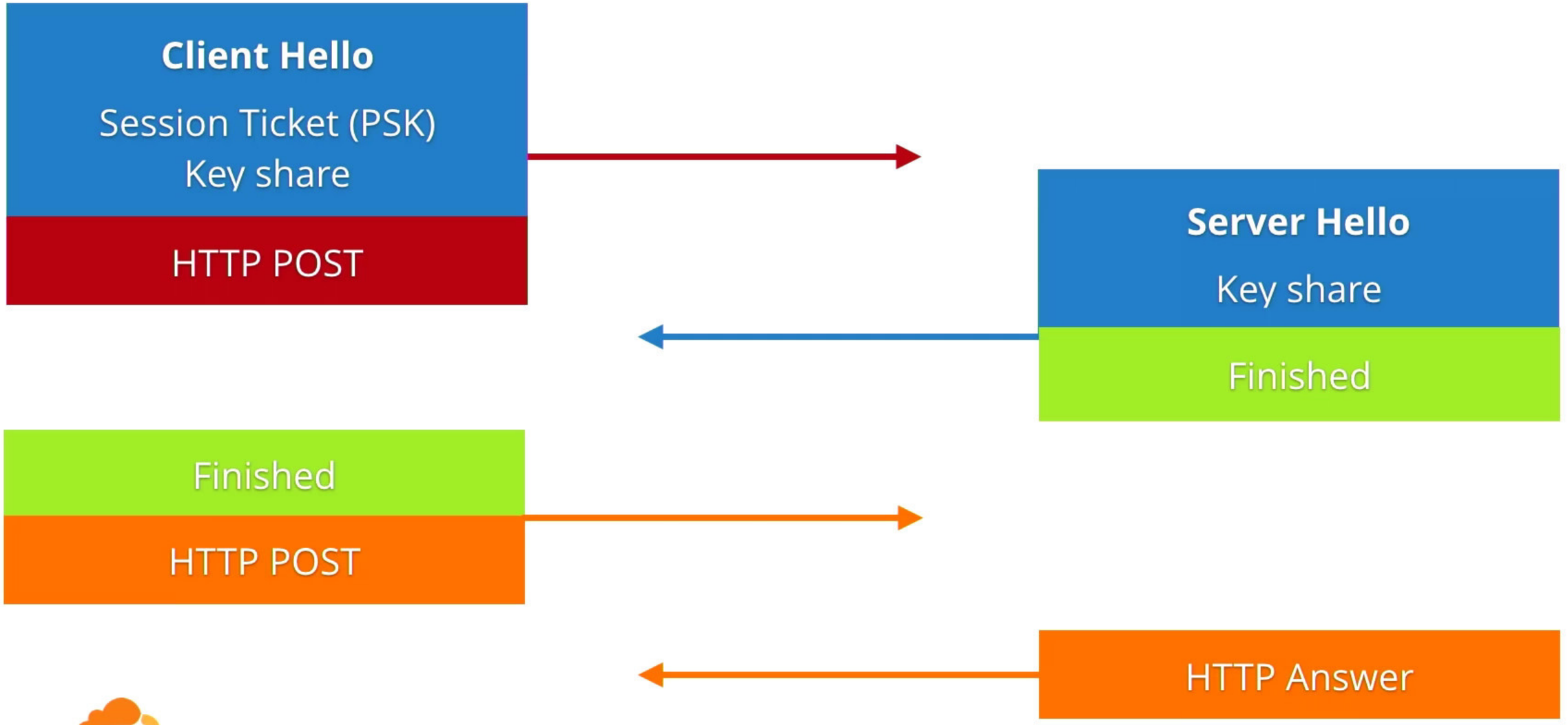
obfuscated_ticket_age

- The client sends the age in milliseconds of the ticket
- The server checks it matches its view, with some leeway
- Obfuscated with a `ticket_age_add` value sent as part of the `New Session Ticket` message

```
struct {
    opaque identity<1..2^16-1>;
    uint32 obfuscated_ticket_age;
} PskIdentity;
```

CLOUDFLARE

0-RTT confirmation



CLOUDFLARE

`max_early_data_size`

- The server must either *accept* or *reject* the early data, entirely, without knowing how much there will be
- If it accepts it and can't process it, it must **buffer** it
- Once the `Finished` comes, all early data is confirmed
- `max_early_data_size` limits the buffer size
- Devised with Drew Springall

CLOUDFLARE

It's the application's responsibility

Protocols MUST NOT use 0-RTT data
without a profile that defines its use.

It's the API's responsibility

- Default to 1-RTT
- Allow the server to reject / wait for the `Finished`
- Let the client to decide what to send in the early data

HTTP and 0-RTT

- Utopia: **GET** is idempotent!
- Reality: nope.

```
GET /send_money.php?to=filippo&amount=1000
```

CLOUDFLARE

HTTP and 0-RTT

- Utopia: **GET** is idempotent!
- Reality: nope.

CLOUDFLARE

HTTP and 0-RTT

Thai Duong, Thiago Valverde, Quan Nguyen
Google Security Team
{thaidn, valverde, quannguyen}@google.com

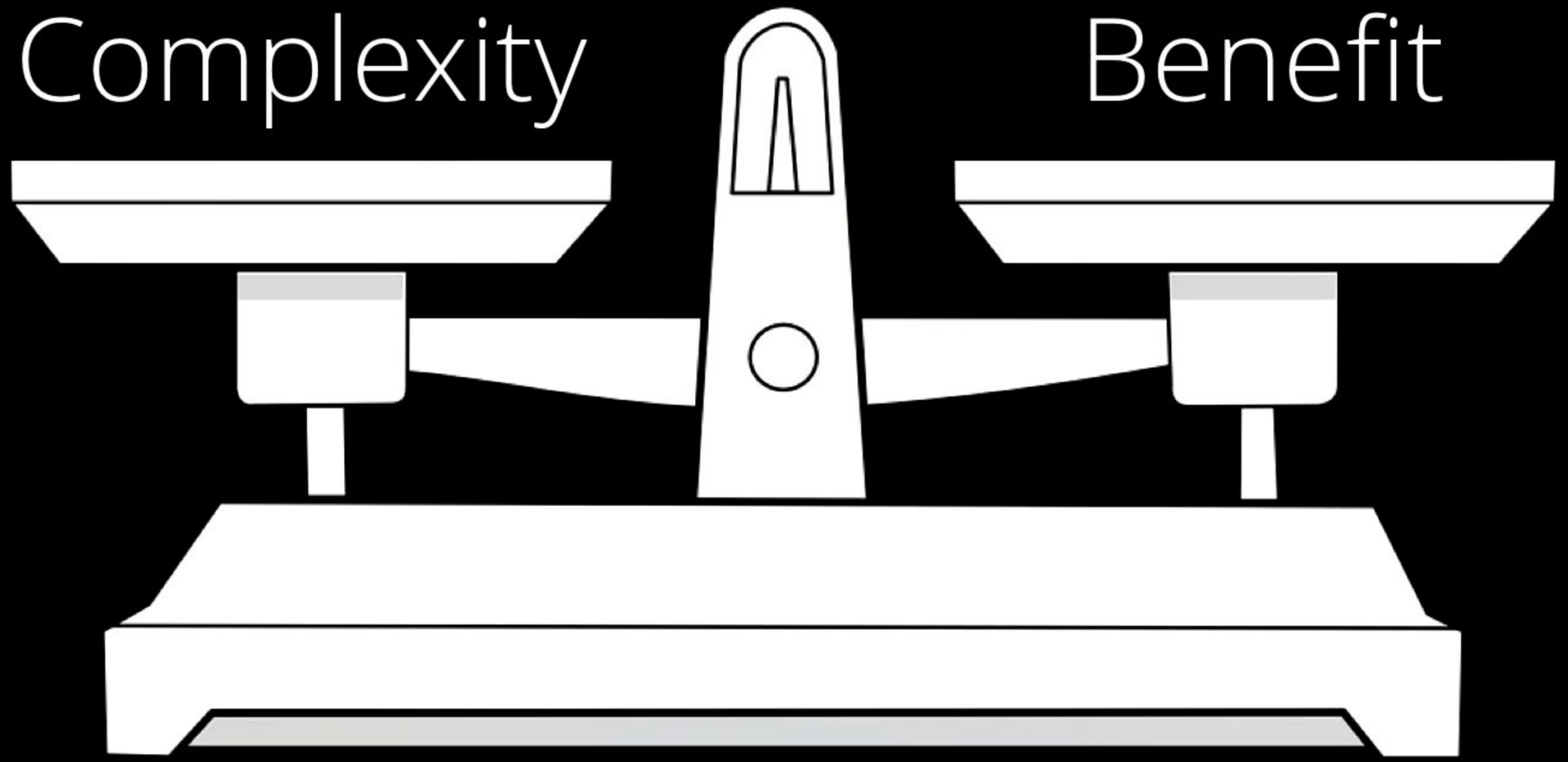
Never, never, never, never give up.

Winston Churchill

CLOUDFLARE

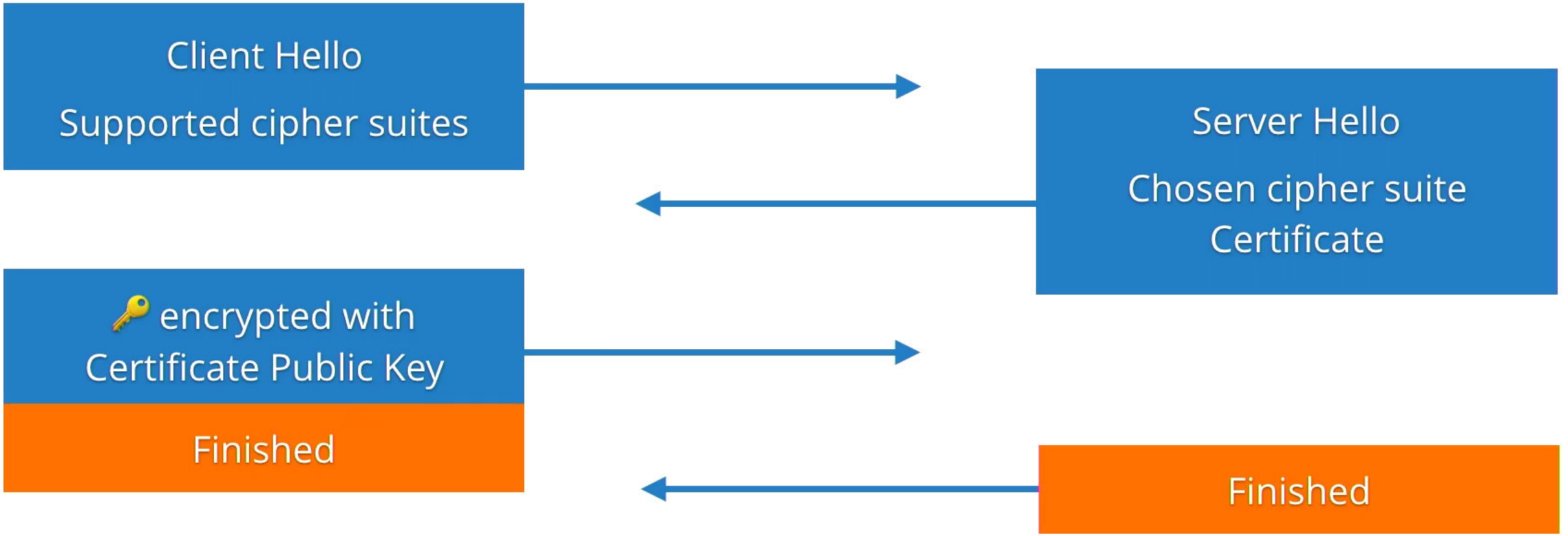
Complexity

Benefit



CLOUDFLARE

TLS 1.2 Static RSA mode



No Forward Secrecy

CLOUDFLARE

To: IETF TLS 1.3 Working Group Members

My name is Andrew Kennedy and I work at BITS, the technology policy division of the Financial Services Roundtable (http://www.fsroundtable.org/bits). My organization represents approximately 100 of the top 150 US-based financial services companies including banks, insurance, consumer finance, and asset management firms.

[...]

Deprecation of the RSA key exchange in TLS 1.3 will cause significant problems for financial institutions, almost all of whom are running TLS internally and have significant, security-critical investments in out-of-band TLS decryption.

[...]

Out-of-band TLS decryption?

Yes, please!

Hi Andrew,

My view concerning your request: no.

Rationale: We're trying to build a more secure internet.

Meta-level comment:

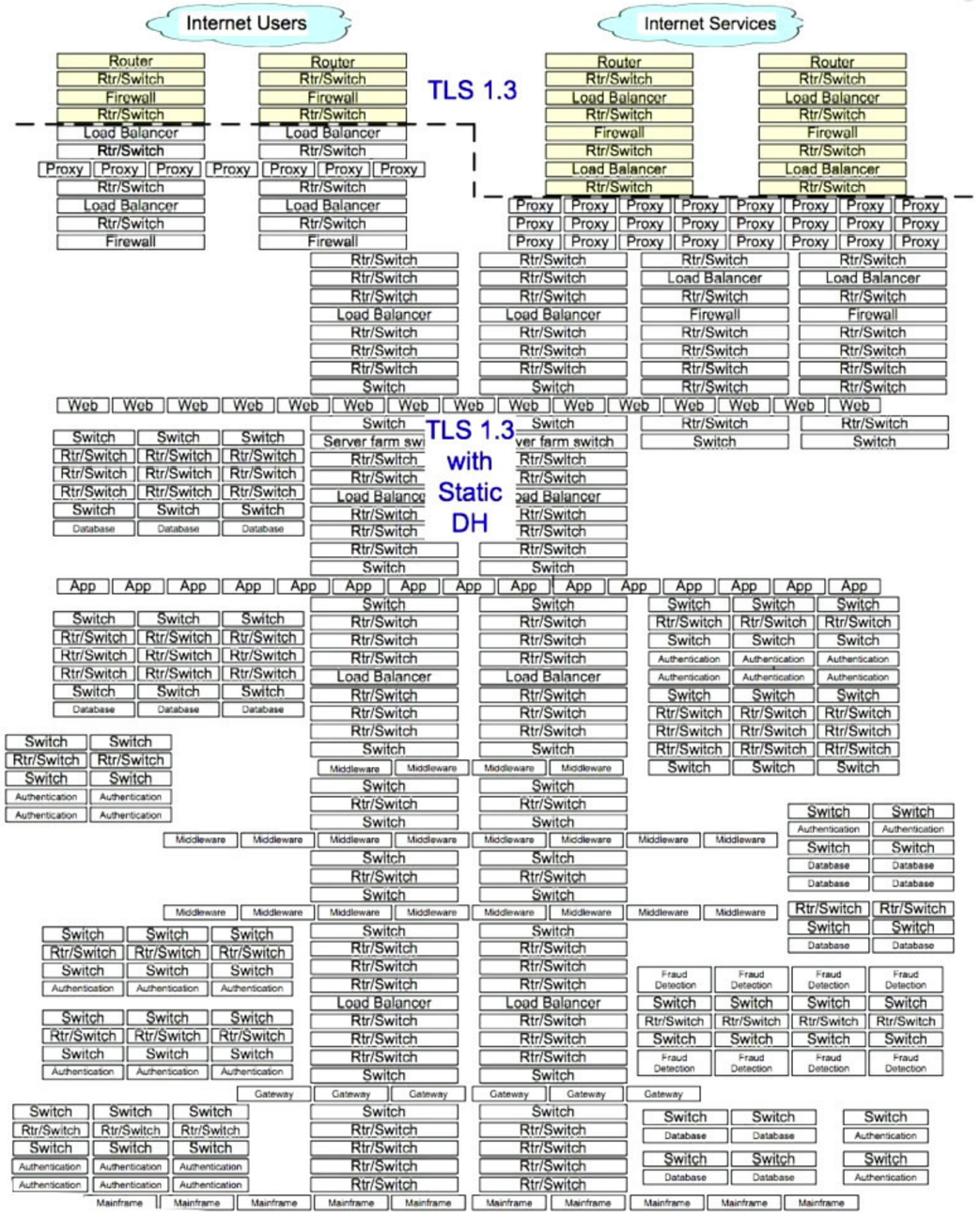
You're a bit late to the party. We're metaphorically speaking at the stage of emptying the ash trays and hunting for the not quite empty beer cans.

More exactly, we are at draft 15 and RSA key transport disappeared from the spec about a dozen drafts ago. I know the banking industry is usually a bit slow off the mark, but this takes the biscuit.

Cheers,

Kenny

Proposed Data Center Visibility Solution



[Docs] [txt|pdf] [Tracker] [Email] [Nits]

Versions: 00

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 1, 2017

M. Green
Cryptography Engineering LLC
October 31, 2016

Data Center use of Static Diffie-Hellman in TLS 1.3
<draft-green-tls-static-dh-in-tls13-00>

MD5 & SHA1

SLOTH 2016

AES-CBC

Vaudenay 2002

Boneh/Brumley 2003

BEAST 2011

Lucky13 2013

POODLE 2014

Lucky Microseconds 2015

RSA-PKCS1-1.5

Bleichenbacher 1998(!!)

Jager 2015

DROWN 2016

Compression

CRIME 2012

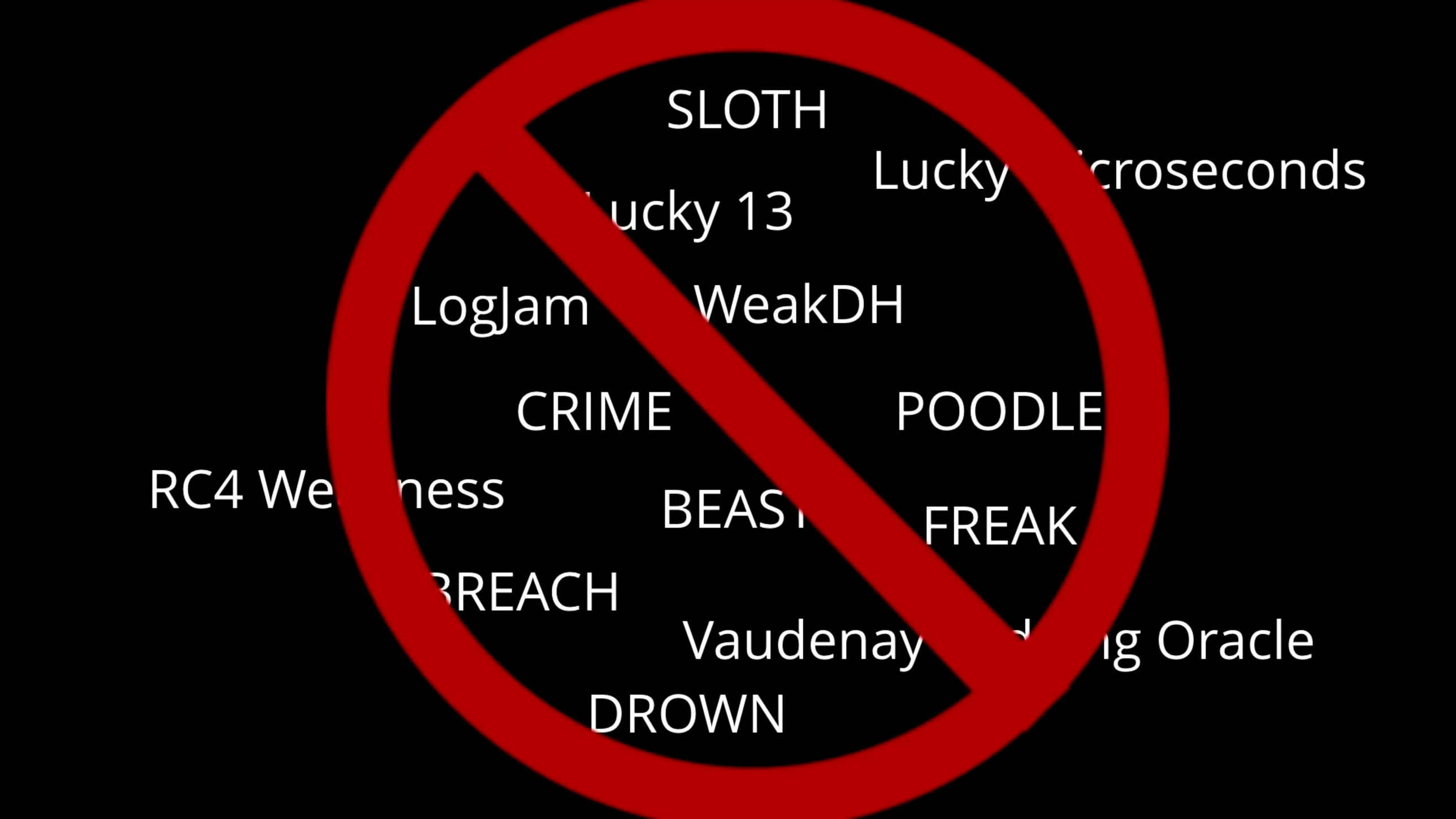
Renegotiation

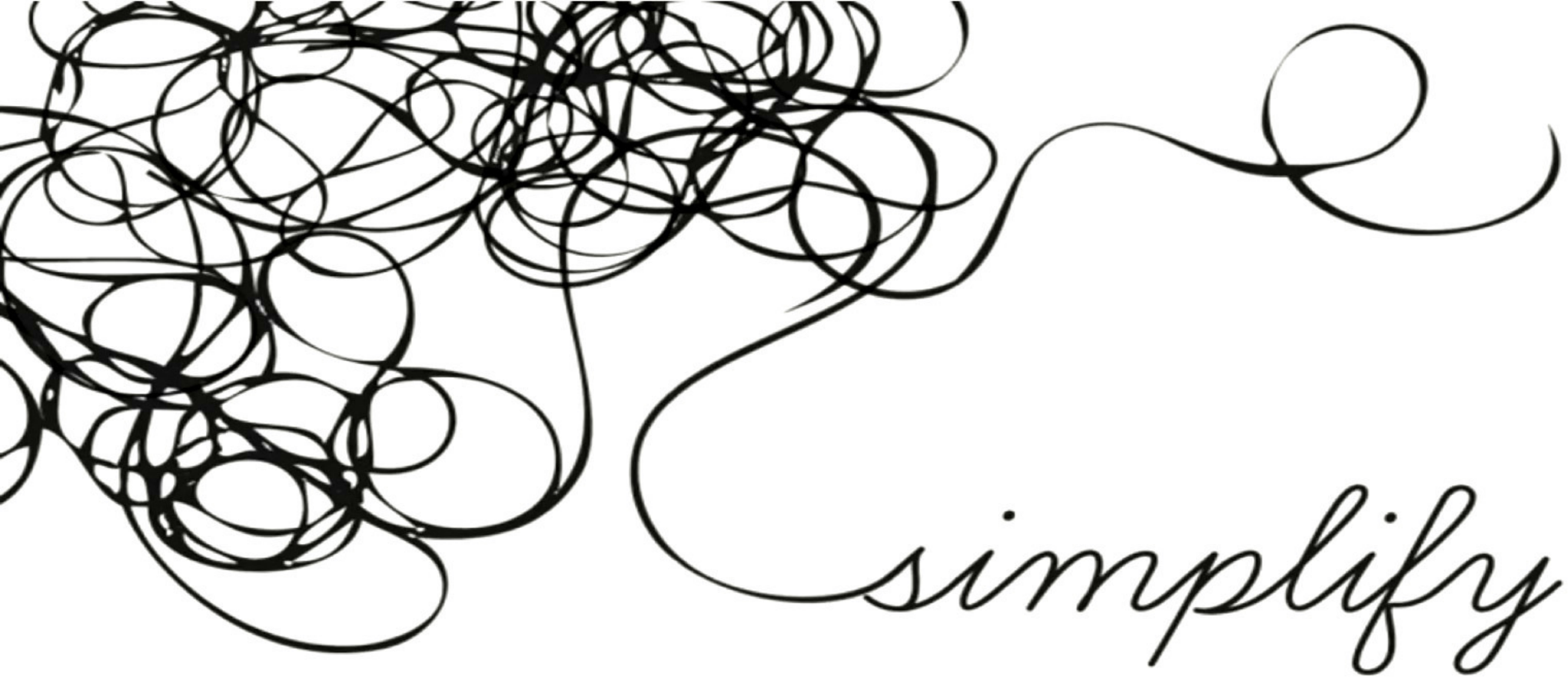
Marsh Ray Attack 2009

Renegotiation DoS 2011

Triple Handshake 2014

Replaced with lightweight key update


SLOTH
Lucky
croseconds
ucky 13
LogJam
WeakDH
CRIME
POODLE
RC4 We ness
BEAST
FREAK
REACH
Vaudenay
g Oracle
DROWN



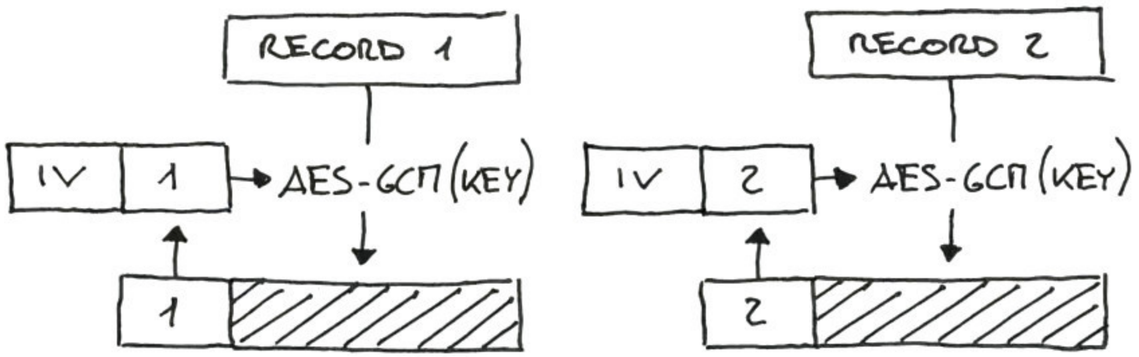
& Fortify

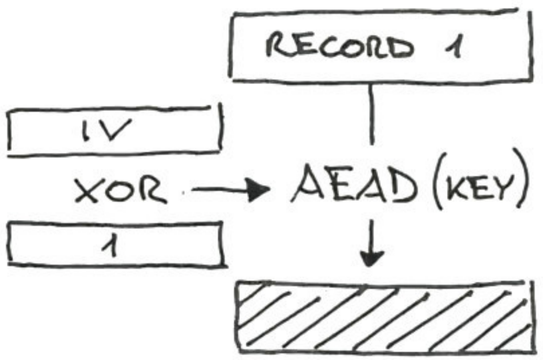
CLOUDFLARE

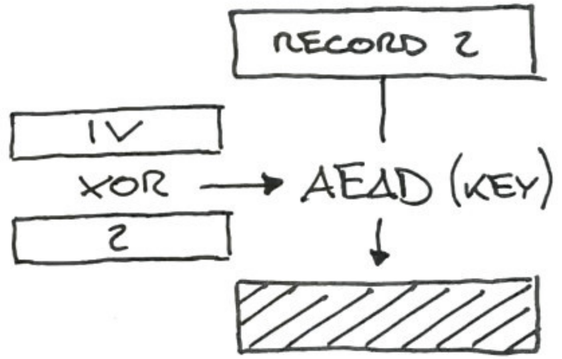
Nonce-Disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS

Hanno Böck[*] Aaron Zauner[#‡] Sean Devlin[§]
Juraj Somorovsky[¶] Philipp Jovanovic[‖]

May 17, 2016







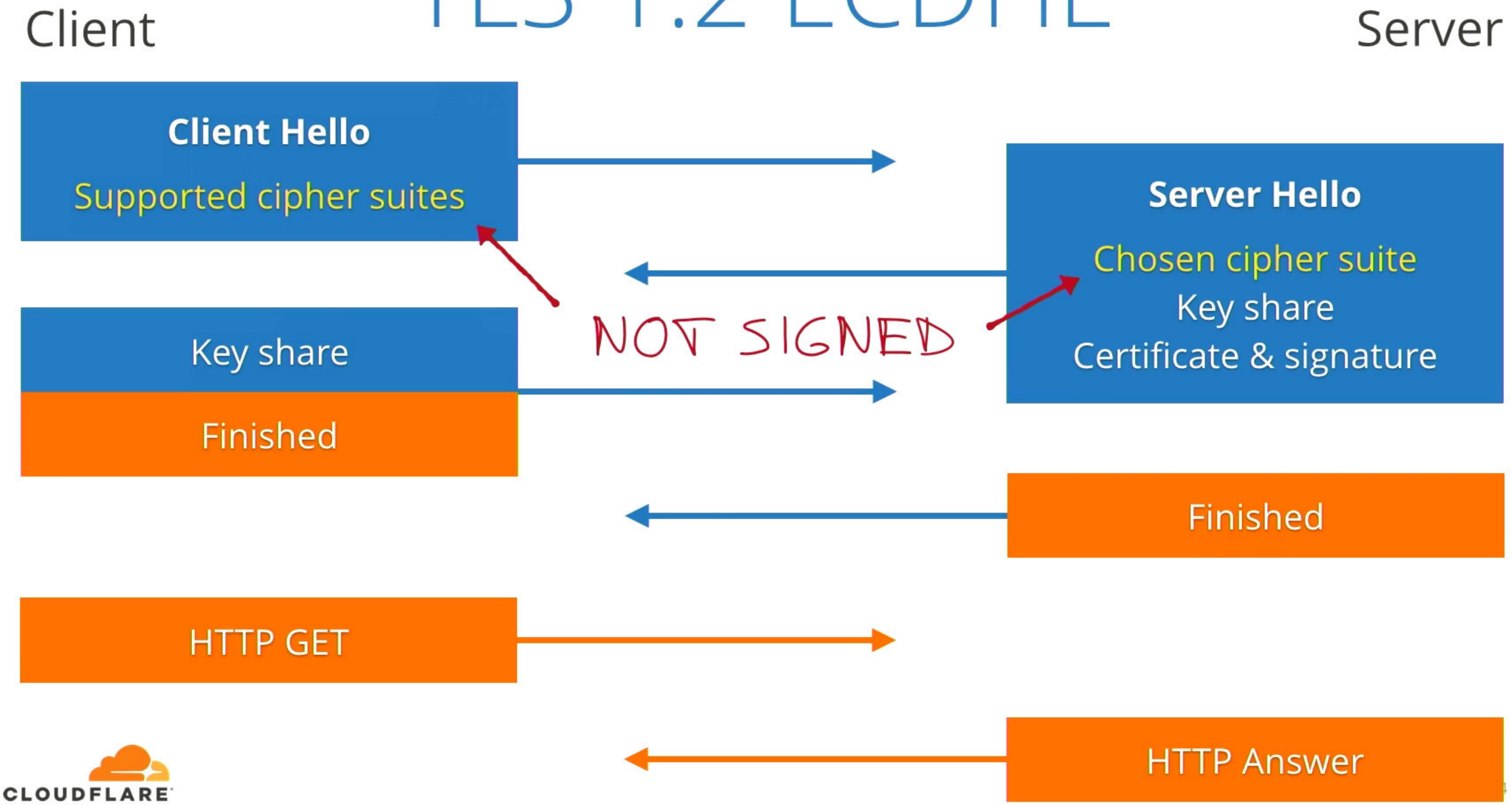
TLS 1.2 Certificate Authentication

- Cipher negotiation protected by Finished Message (MAC)
- MAC algorithm determined by cipher negotiation

- FREAK, LogJam, CurveSwap: choose weak parameters

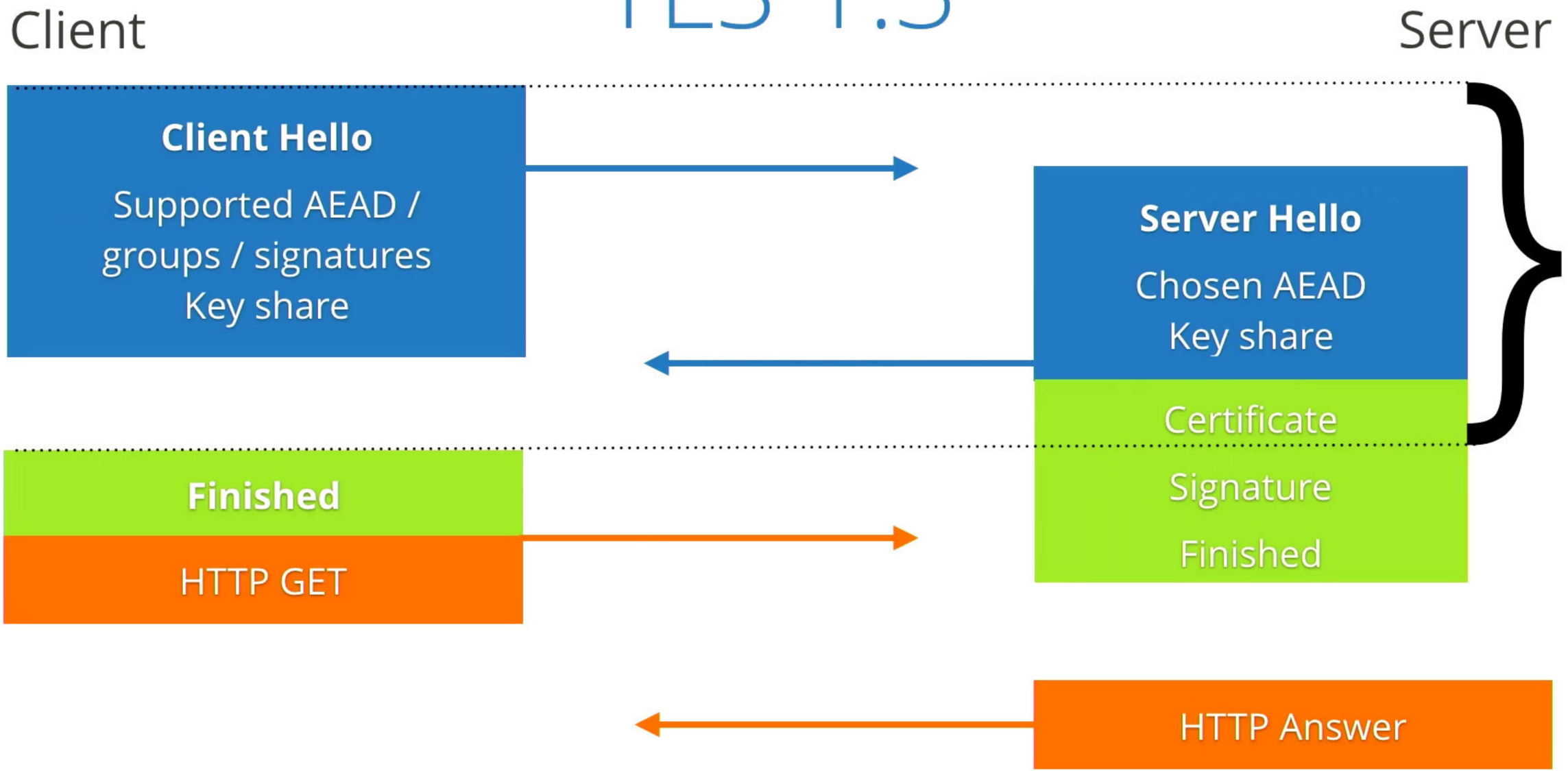
CLOUDFLARE

TLS 1.2 ECDHE



CLOUDFLARE

TLS 1.3



CLOUDFLARE

Fewer, better choices

- Key Exchange, Cipher, Authentication negotiated separately
- No arbitrary DH groups
- No arbitrary curves

▼ Cipher Suites (19 suites)
    Cipher Suite: Unknown (0xdada)
    Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
    Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
    Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)

▼ Extension: signature_algorithms (len=20)
    Type: signature_algorithms (13)
    Length: 20
    Signature Hash Algorithms Length: 18
  ▶ Signature Hash Algorithms (9 algorithms)

▼ Extension: elliptic_curves (len=10)
    Type: elliptic_curves (10)
    Length: 10
    Elliptic Curves Length: 8
  ▼ Elliptic curves (4 curves)
      Elliptic curve: Unknown (0x9a9a)
      Elliptic curve: ecdh_x25519 (0x001d)
      Elliptic curve: secp256r1 (0x0017)
      Elliptic curve: secp384r1 (0x0018)

Safer Resumption

TLS 1.2 tickets

- **Current** session keys encrypted with session ticket key
- Session ticket key compromise a risk for all connections

TLS 1.3 tickets

- **Next** session keys encrypted with session ticket key
- Session ticket key compromise only risk for resumed connections

CLOUDFLARE

Safer Resumption

TLS 1.2 tickets

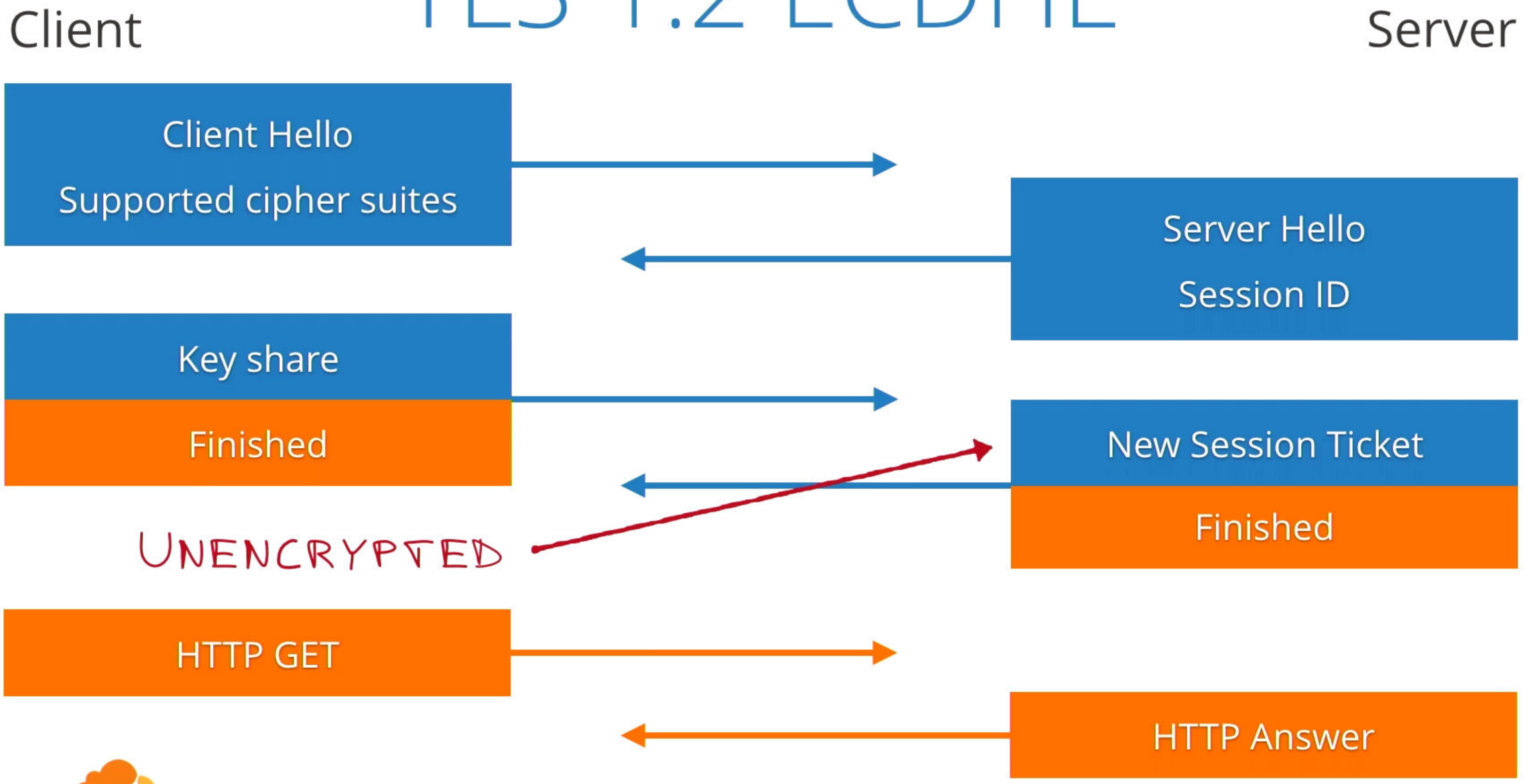
- **Current** session keys encrypted with session ticket key
- Session ticket key compromise a risk for all connections

TLS 1.3 tickets

- **Next** session keys encrypted with session ticket key
- Session ticket key compromise only risk for resumed connections

CLOUDFLARE

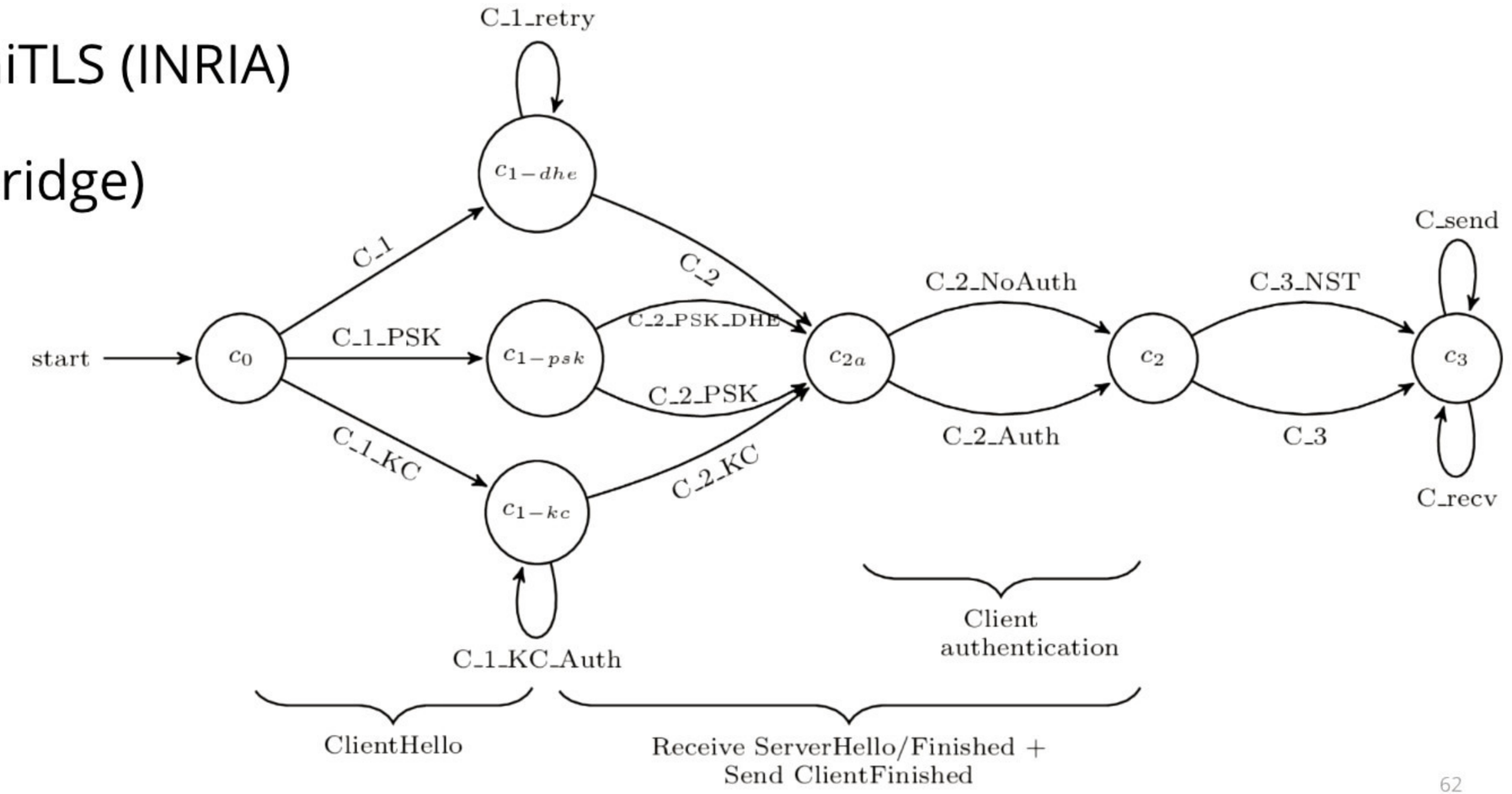
TLS 1.2 ECDHE



CLOUDFLARE

Formal Verification

- Tamarin (Oxford, Royal Holloway)
- ProScript-TLS, miTLS (INRIA)
- nqsb-TLS (Cambridge)



Standards

The IETF way

CLOUDFLARE

Nick Sullivan
@grittygrease

TLS 1.3 Wish List - it's a short list.
ietf.org/proceedings/87 …

5:26 PM - 4 Sep 2013

CLOUDFLARE

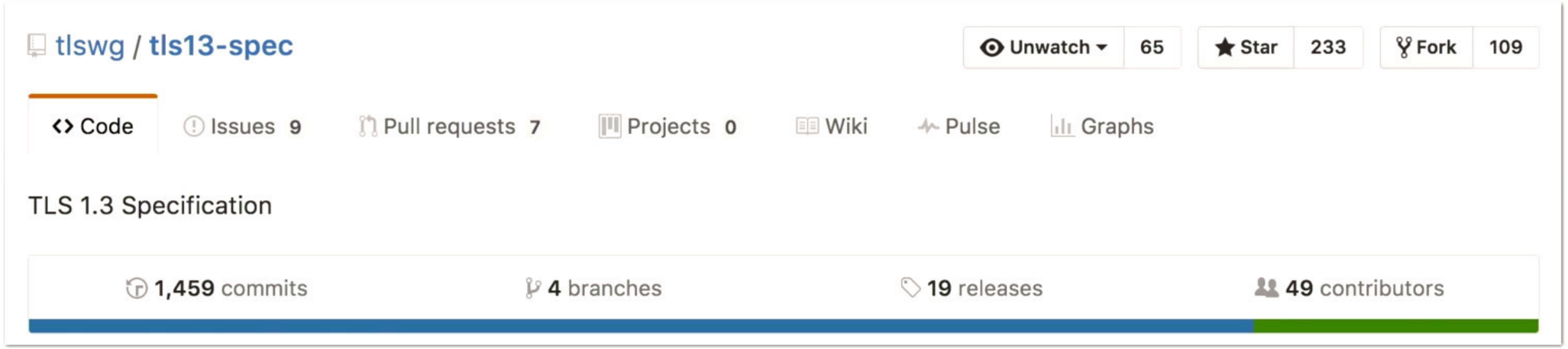
Timeline

- First Draft: April 17, 2014
  - 3Shake, POODLE, FREAK, LogJam, DROWN, Lucky Microseconds, SLOTH, more...
- Draft 18: October 26, 2016
- Final draft: **February, 2017** (we hope)

- TLS 1.2: **79 pages**
- TLS 1.3: **81 pages** (minus references and appendices)

CLOUDFLARE

Github + Mailing List



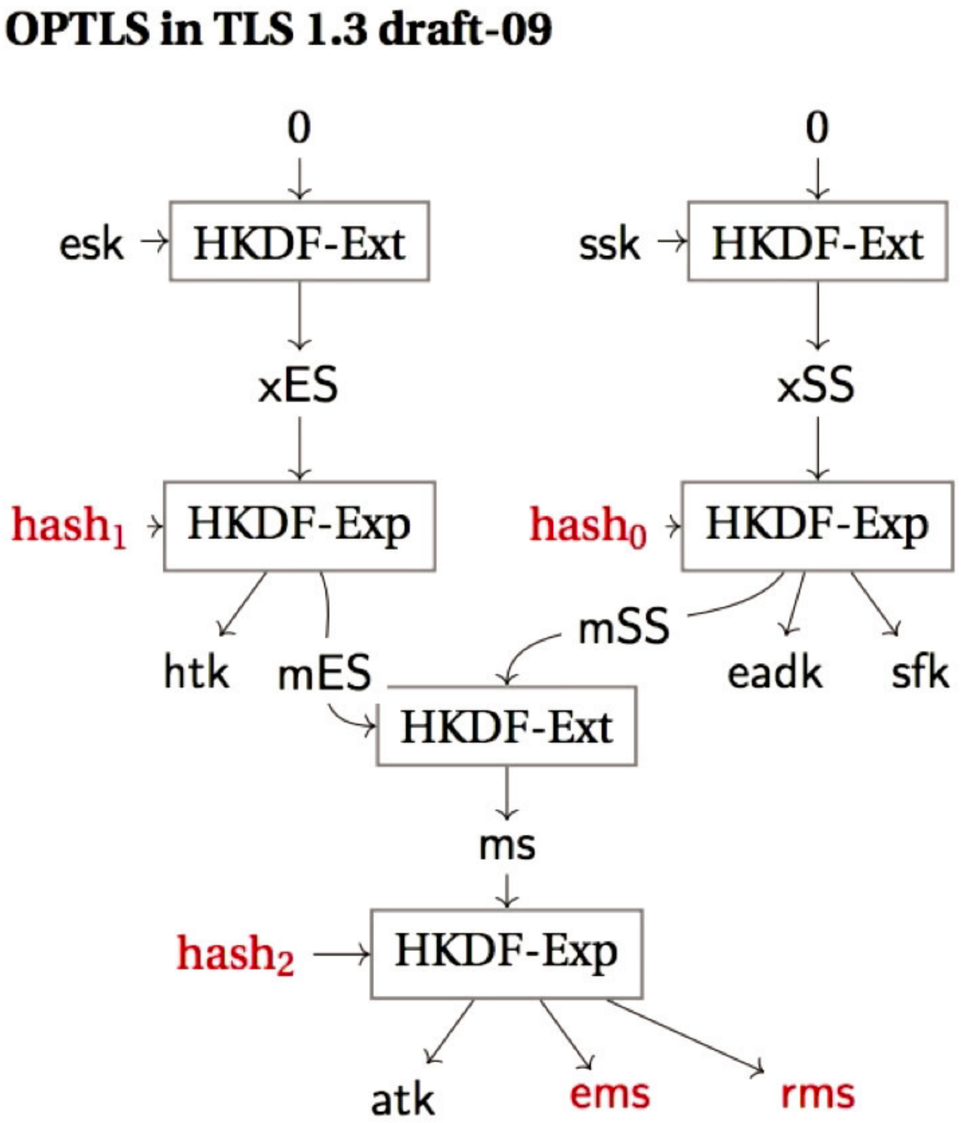
Transport Layer Security working group of the IETF Discussion Archive - Date Index

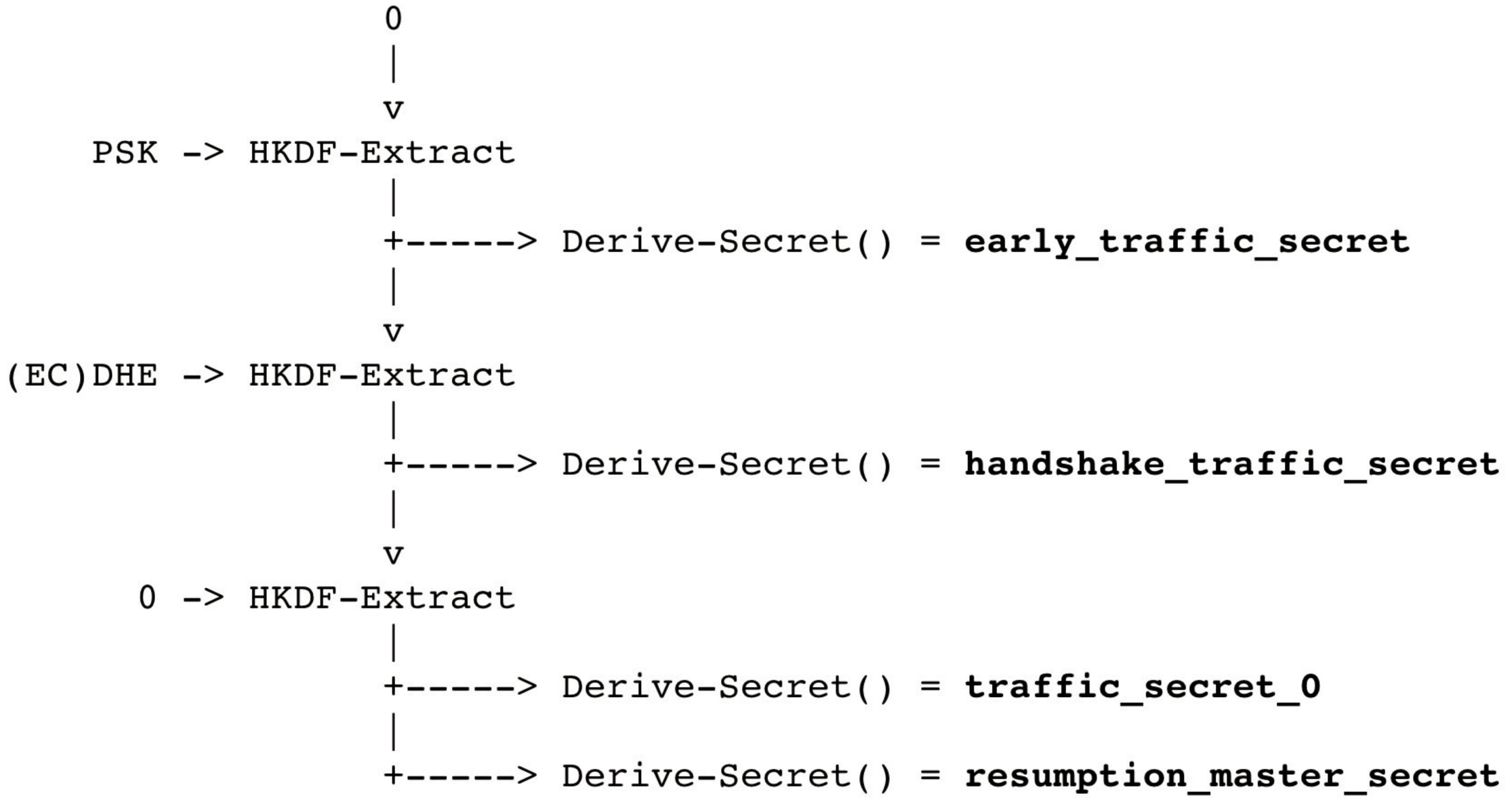
[Prev Page] [Next Page] [Thread Index] [IETF Main Archive Directory]

- **Dec 22 2016**
  - Re: [TLS] Using both External PSK and (EC)DH in TLS 1.3, *Eric Rescorla*
  - Re: [TLS] Using both External PSK and (EC)DH in TLS 1.3, *Russ Housley*
  - Re: [TLS] Using both External PSK and (EC)DH in TLS 1.3, *Joseph Salowey*
  - Re: [TLS] Using both External PSK and (EC)DH in TLS 1.3, *David Benjamin*
  - [TLS] Using both External PSK and (EC)DH in TLS 1.3, *Russ Housley*

Key Schedule

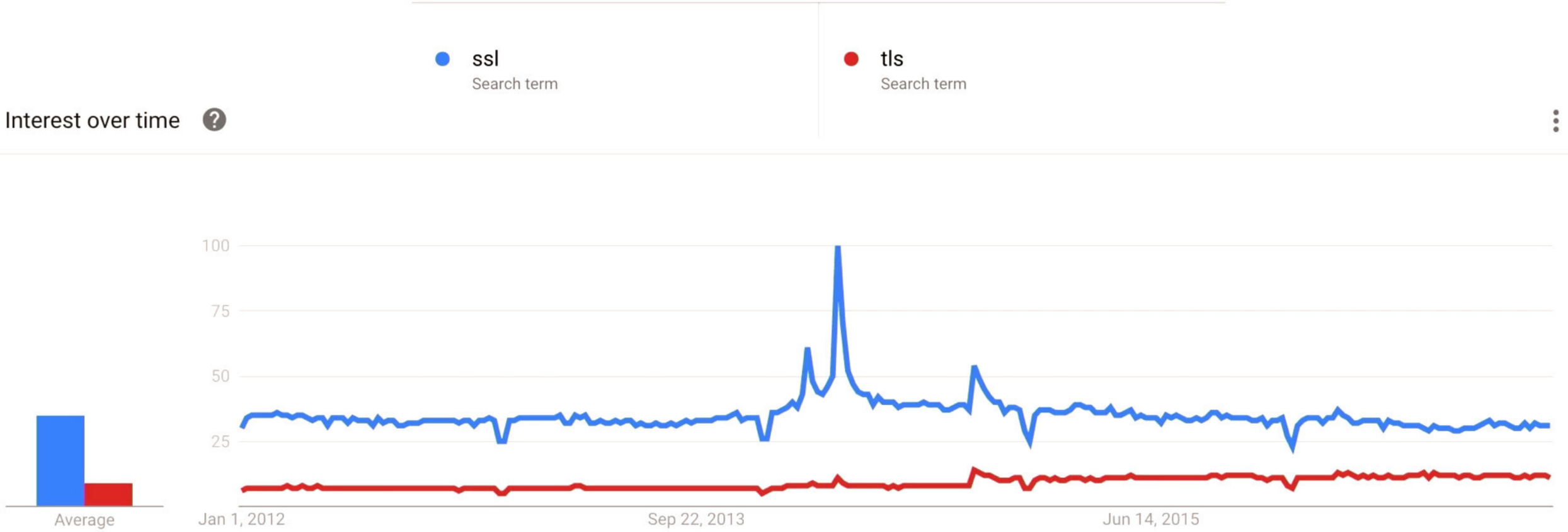
- Inspired by QUIC crypto
- Semi-static DH key shared out of band
- Tree-based key schedule




0
PSK -> HKDF-Extract
Derive-Secret() = early_traffic_secret
(EC)DHE -> HKDF-Extract
Derive-Secret() = handshake_traffic_secret
0 -> HKDF-Extract
Derive-Secret() = traffic_secret_0
Derive-Secret() = resumption_master_secret

What's in a name?

Is it TLS 1.3, TLS 2, TLS 2.0, TLS 4, TLS 7, TLS 2017?



Whitney Merrill @wbm312 · Aug 31

What should the next version of TLS (after TLS 1.2) be called? ietf.org/mail-archive/w …

39% TLS 1.3

28% TLS 2.0

13% TLS/2

20% TLS 3

1,105 votes • Final results

50 22 14 •••

Nick Sullivan @grittygrease · Nov 29

What should the next version of TLS be called? (it's currently called TLS 1.3)

34% TLS 1.3

27% TLS 2

15% TLS 4

24% TLS 7

545 votes • Final results

39 12 4 •••

Ryan Hurst @rmhrisk · Aug 31

What should the new version of TLS be called? #ssl #bikeshedding #standardsbycomitee

21% TLS/42

27% TLS/Bacon

52% Mmmm Bacon

322 votes • Final results

9 16 8 •••

CLOUDFLARE

Version Intolerance

- Wire versions
  - SSL 3.0: 3.0
  - TLS 1.0: 3.1
  - TLS 1.1: 3.2
  - TLS 1.2: 3.3
  - TLS 1.3: 3.4 ???
- Servers are intolerant of 3.4
  - >2% of servers fail connection
  - Solution: “3.3” in ClientHello, real versions in extension
  - GREASE by David Benjamin

CLOUDFLARE

Version Intolerance

```
▼ Extension: supported_versions (len=11)
    Type: supported_versions (43)
    Length: 11
    Supported Versions length: 10
    Supported Versions: Unknown (0x8a8a)
    Supported Versions: TLS 1.3 (draft 18) (0x7f12)
    Supported Versions: TLS 1.2 (0x0303)
    Supported Versions: TLS 1.1 (0x0302)
    Supported Versions: TLS 1.0 (0x0301)
```

CLOUDFLARE

Implementation

Getting our hands dirty

CLOUDFLARE

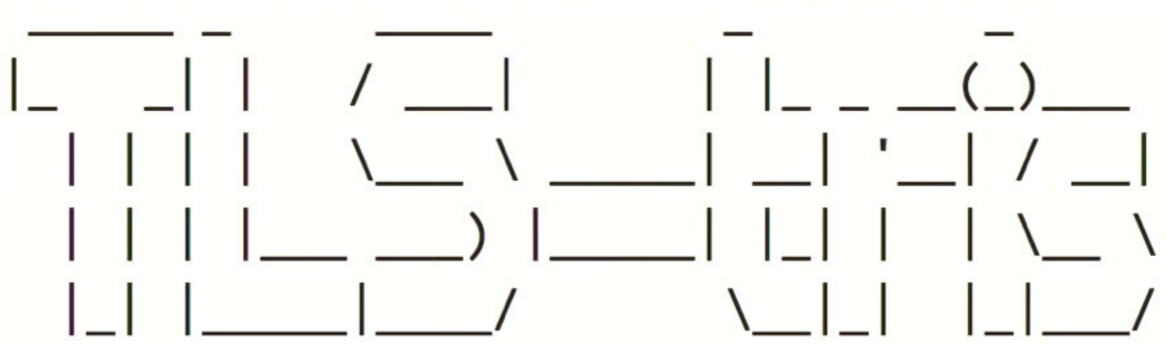
IETF 95 Hackathon - April 2016

- NSS (C): Martin Thomson and Eric Rescorla
- Mint (Go): Richard Barnes and Nick Sullivan

Result:

Firefox was able to load https://tls13.cloudflare.com!

CLOUDFLARE



- Based on Go crypto/tls
- Server only
- Audited

crypto/tls, now with 100% more 1.3.

DO NOT USE THIS FOR THE SAKE OF EVERYTHING THAT'S GOOD AND JUST.

CLOUDFLARE

[dev.tls] crypto/tls: implement TLS 1.3 cipher suites

[dev.tls] crypto/tls: implement TLS 1.3 messages

[dev.tls] crypto/tls: implement TLS 1.3 record layer

https://go-review.googlesource.com/q/branch:+dev.tls

CLOUDFLARE

Deploying is hard

- First deployed Tris: draft 13
- Supported multiple drafts at a time (“hybrids”)
- Browsers sometimes… diverged

Build Jobs

✓	# 206.1		Go: 1.7	MODE=interop CLIENT=boring	5 min 26 sec
✓	# 206.2		Go: 1.7	MODE=interop CLIENT=bogo	4 min 9 sec
✓	# 206.3		Go: 1.7	MODE=interop CLIENT=tstclnt	6 min 49 sec
✓	# 206.4		Go: 1.7	MODE=interop CLIENT=picotls ZRTT=1	3 min 28 sec
✓	# 206.5		Go: 1.7	MODE=interop CLIENT=mint	2 min 57 sec
✓	# 206.6		Go: 1.7	MODE=gotest	3 min 3 sec

Allowed Failures ?

✗	# 206.7		Go: 1.7	MODE=interop CLIENT=tstclnt ZRTT=1	6 min 21 sec
✓	# 206.8		Go: 1.7	MODE=interop CLIENT=boring REVISION=	5 min 17 sec
✗	# 206.9		Go: 1.7	MODE=interop CLIENT=tstclnt REVISION=	6 min 56 sec

You may already be using it

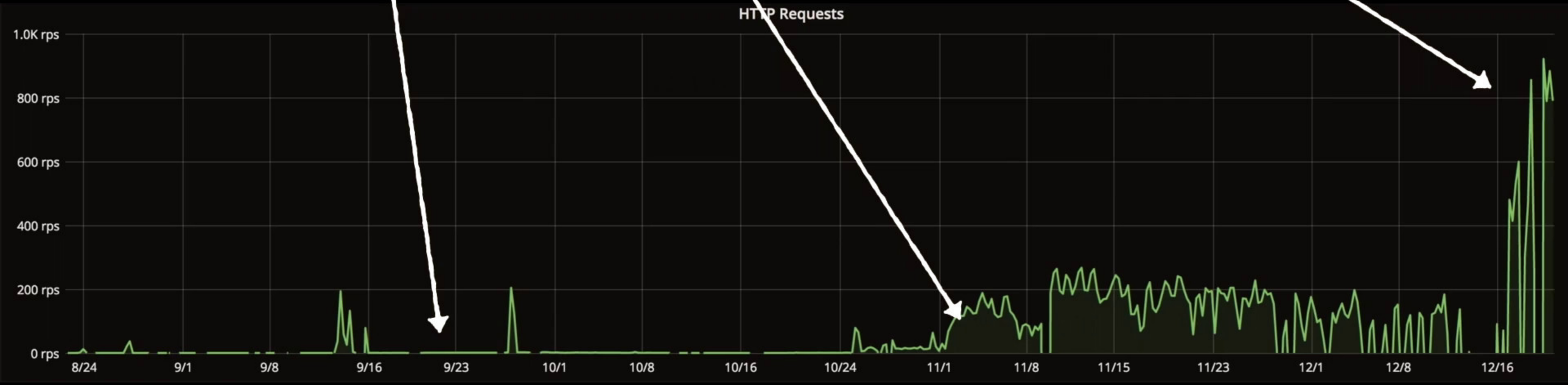
- Firefox Nightly
- Chrome Beta (50%) / Canary

CLOUDFLARE

Cloudflare Launch

Firefox Nightly

Chrome Field Test



CLOUDFLARE

https://tlswg.github.io/tls13-spec/

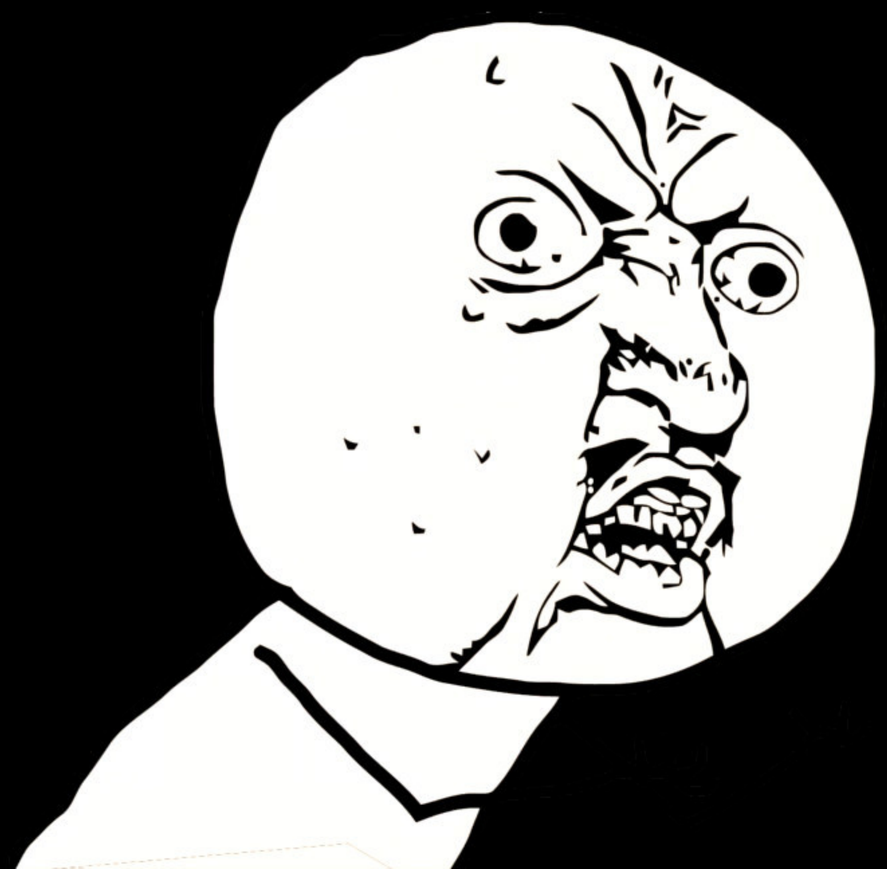
https://github.com/cloudflare/tls-tris

https://blog.cloudflare.com/tag/tls-1-3/

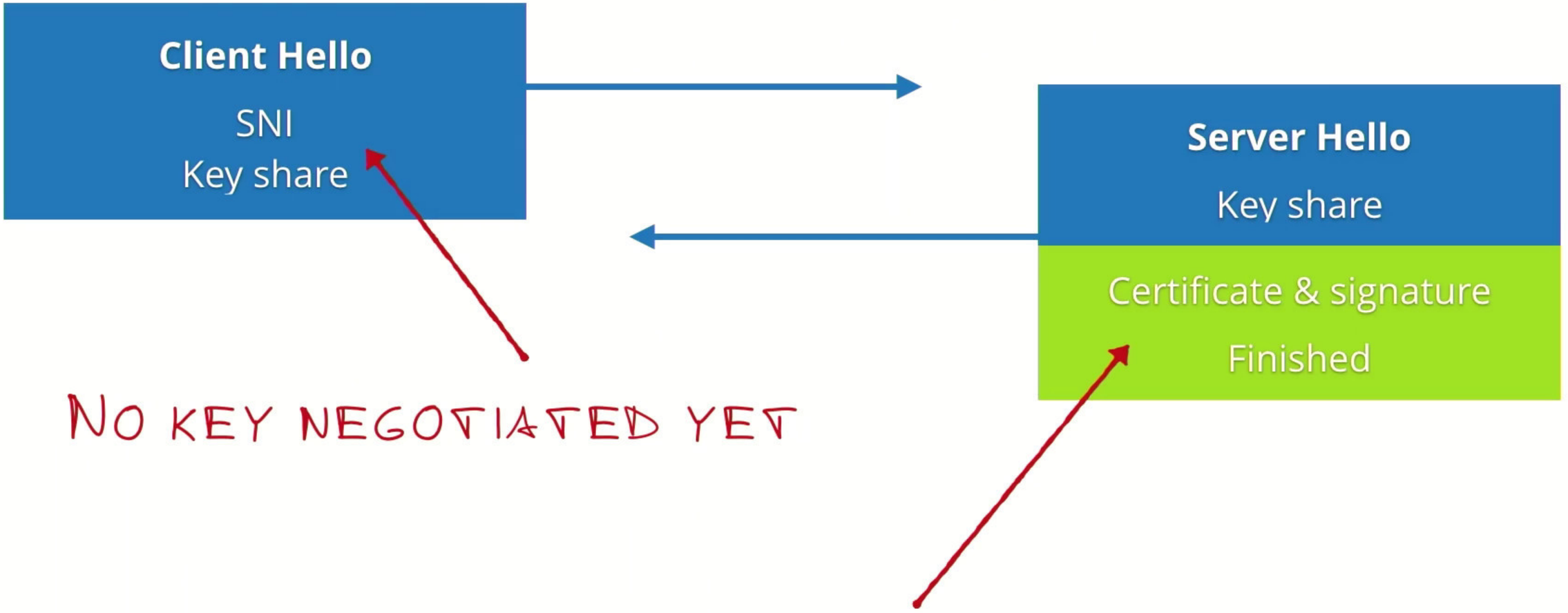
Nick Sullivan
@grittygrease

Filippo Valsorda
@FiloSottile

Y U NO ENCRYPT SNI!?



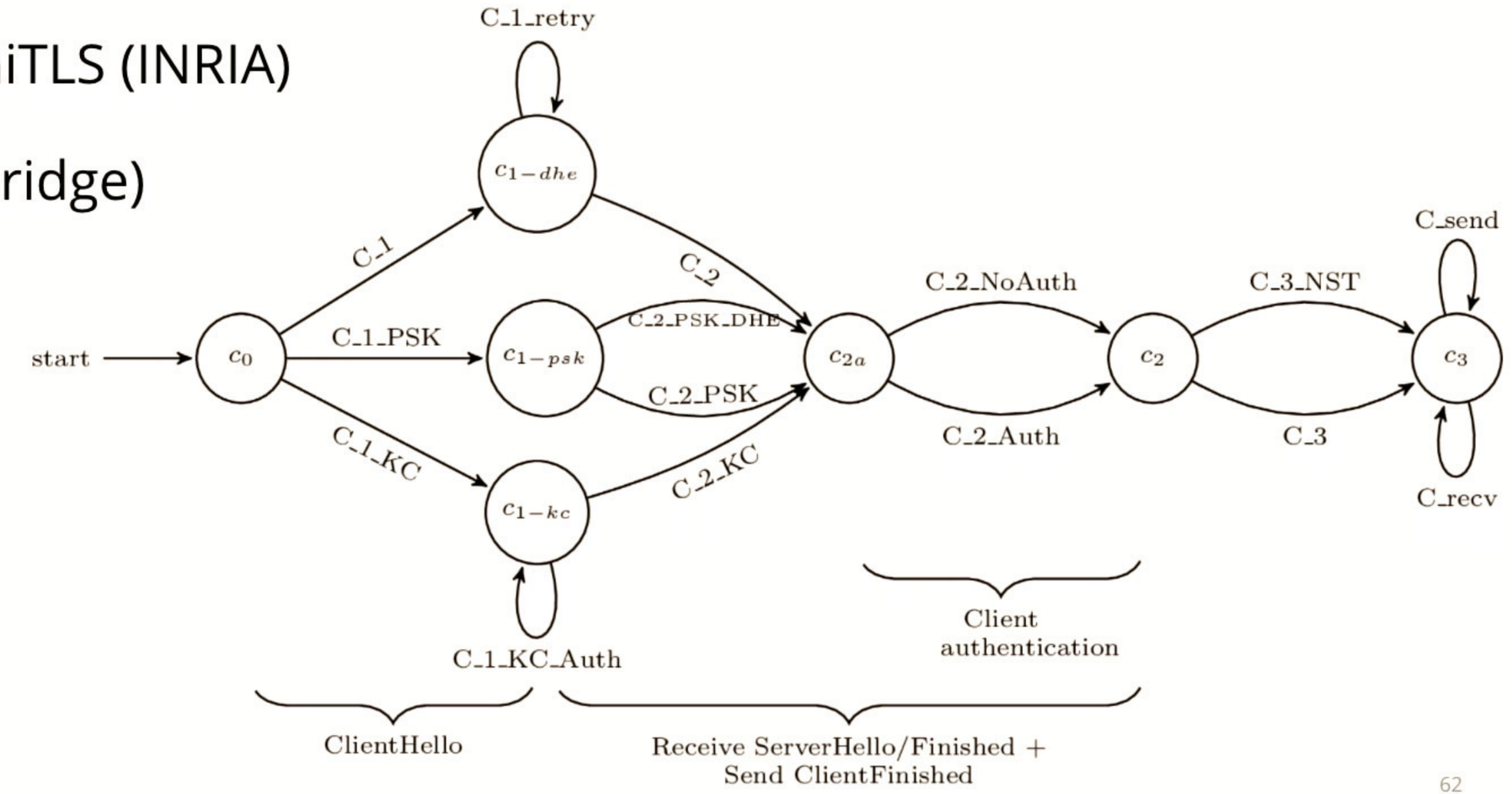
TLS 1.3 can't encrypt SNI



CLOUDFLARE

Formal Verification

- Tamarin (Oxford, Royal Holloway)
- ProScript-TLS, miTLS (INRIA)
- nqsb-TLS (Cambridge)



https://tlswg.github.io/tls13-spec/

https://github.com/cloudflare/tls-tris

https://blog.cloudflare.com/tag/tls-1-3/

Nick Sullivan
@grittygrease

Filippo Valsorda
@FiloSottile

CREATIVE COMMONS

ATTRIBUTION 4.0 INTERNATIONAL

http://creativecommons.org/licenses/by/4.0/

RECORDED BY

TOGETHER WITH

ags

ags – Wissenschaftliche Arbeitsgemeinschaft
für Studio- und Senderfragen
an der TU Braunschweig e.V.

FEM

Forschungsgemeinschaft
elektronische Medien e.V.

iSystems

VISUALS & SOUND

VEITH YÄGER
ARNE BENSE

FOR MORE TALKS AND
CONFERENCE RECORDINGS
VISIT

MEDIA.CCC.DE