

TLS 1.3

Lessons Learned from Implementing
and Deploying the Latest Protocol



Nick Sullivan
@grittygrease

November 11, 2016

PLAY

SP

0:00:00

- MENU -



PAST

PRESENT

FUTURE

Transport Layer Security

- Point-to-point secure communication protocol
- Client-server model, with server authentication, optional client authentication





OSIModel

Layer 6

HTTP

Application

Presentation

TLS

Session

TCP

Transport

IP

Network

Internet

Data link

Physical

HTTP

TLS

TCP

IP

Ethernet

Physical

Application

Presentation

Session

Transport

Network

Data link

Physical

Layer 6

TLS

HTTP

SMTP

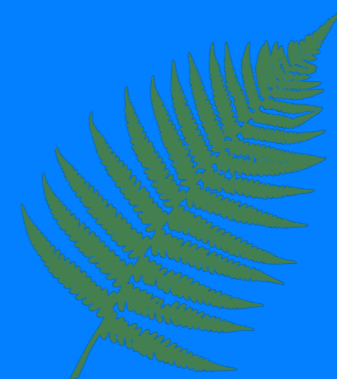
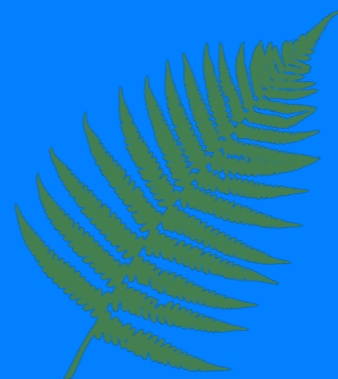
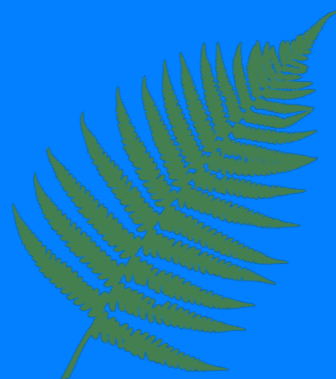
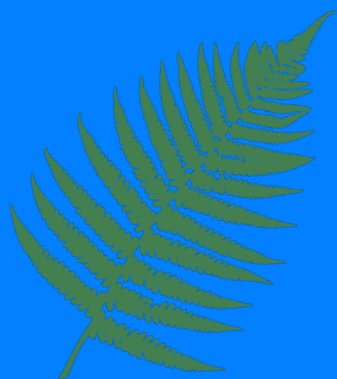
gRPC



HTTP

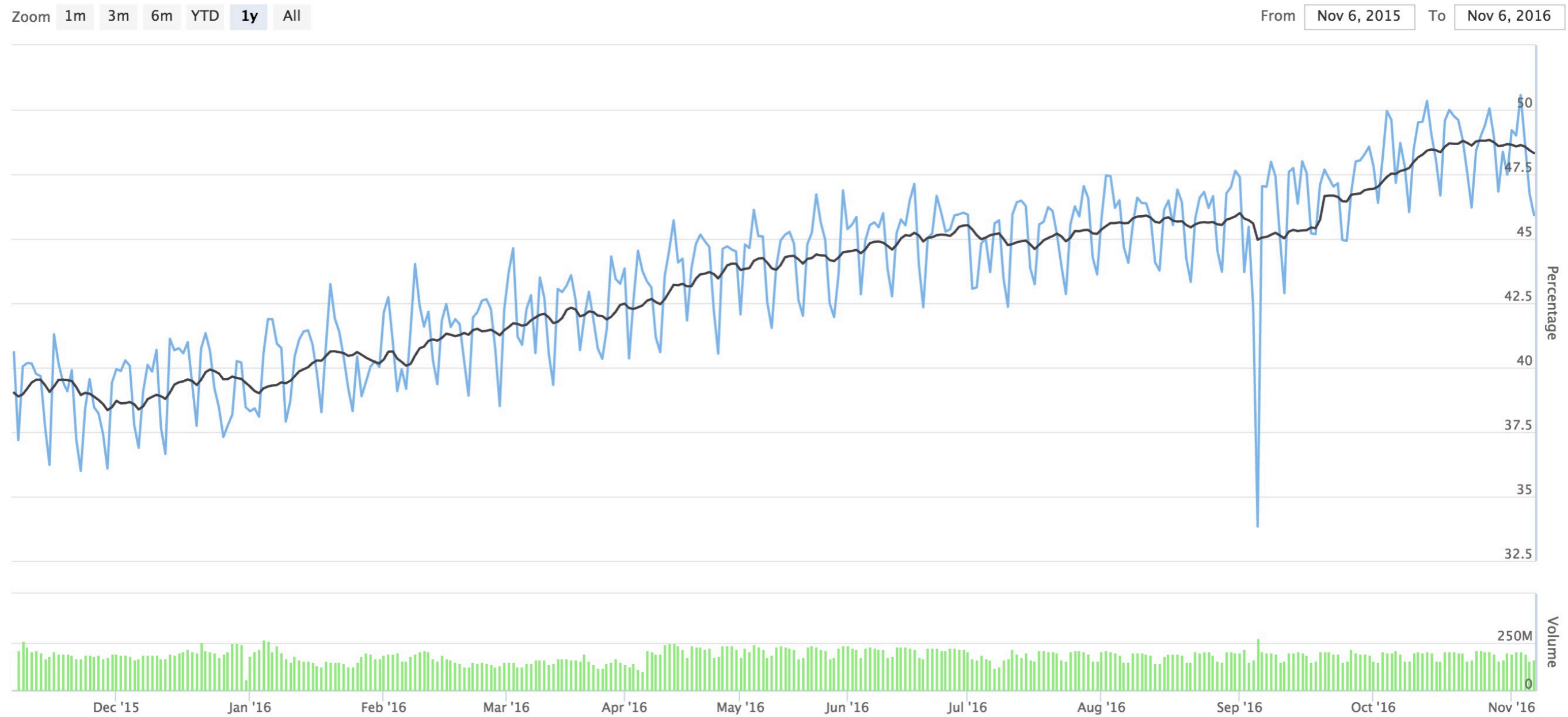
SMTP

gRPC



50% of page loads are HTTPS

Time series for HTTP_PAGELOAD_IS_SSL, bin(s) 1 (in %)





The Evolution of T L S

- SSLv1 (1993?) 🍌
- SSLv2 (1994) 🌊
- SSLv3 (1995) 🐩
- TLS 1.0 (1999) 🍌

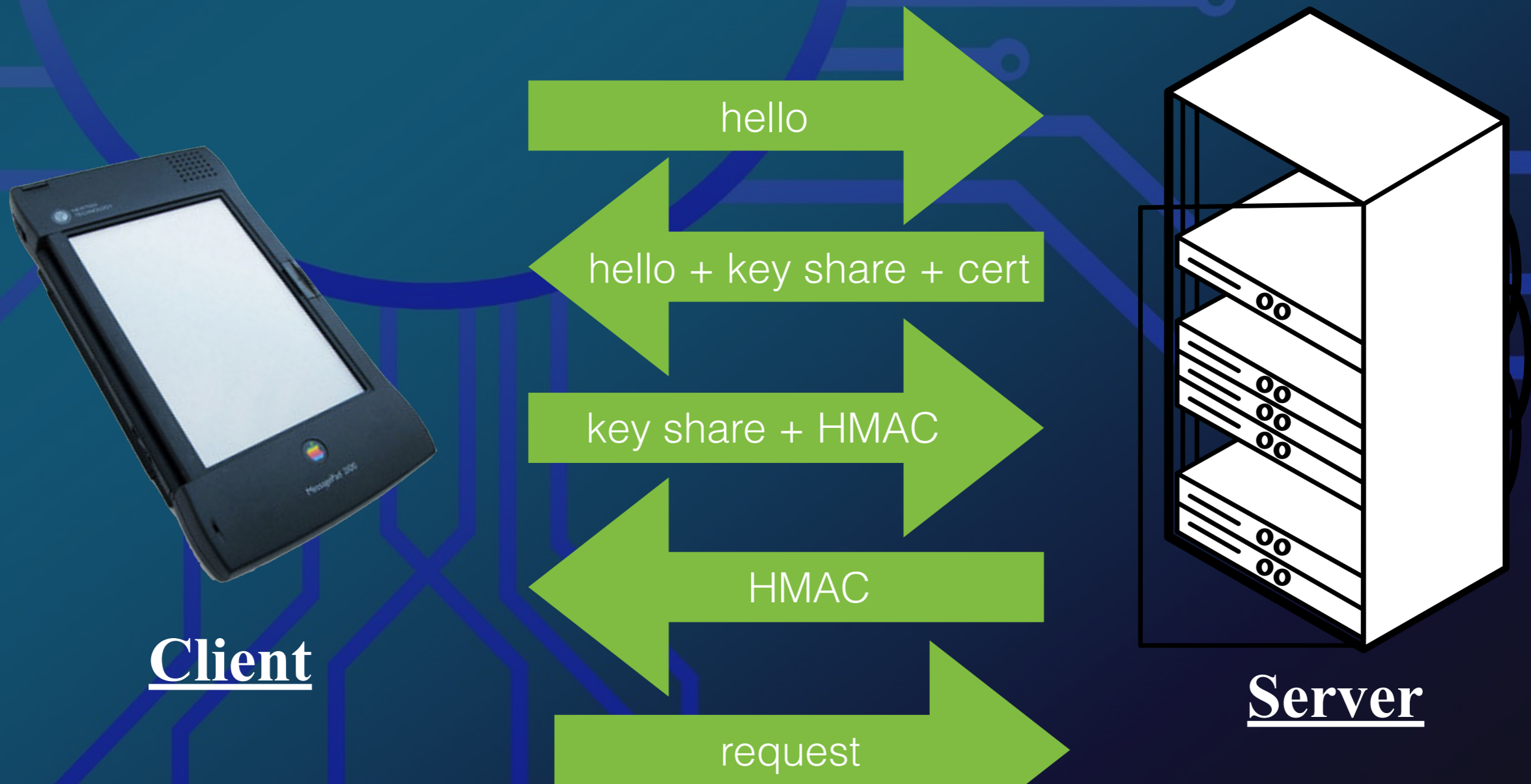
- TLS 1.1 (2006)
 - Lucky 13
 - RC4 Biases
 - SWEET32
- TLS 1.2 (2008)
 - Safe with the right configuration

E s s e n t i a l C o m p o n e n t s

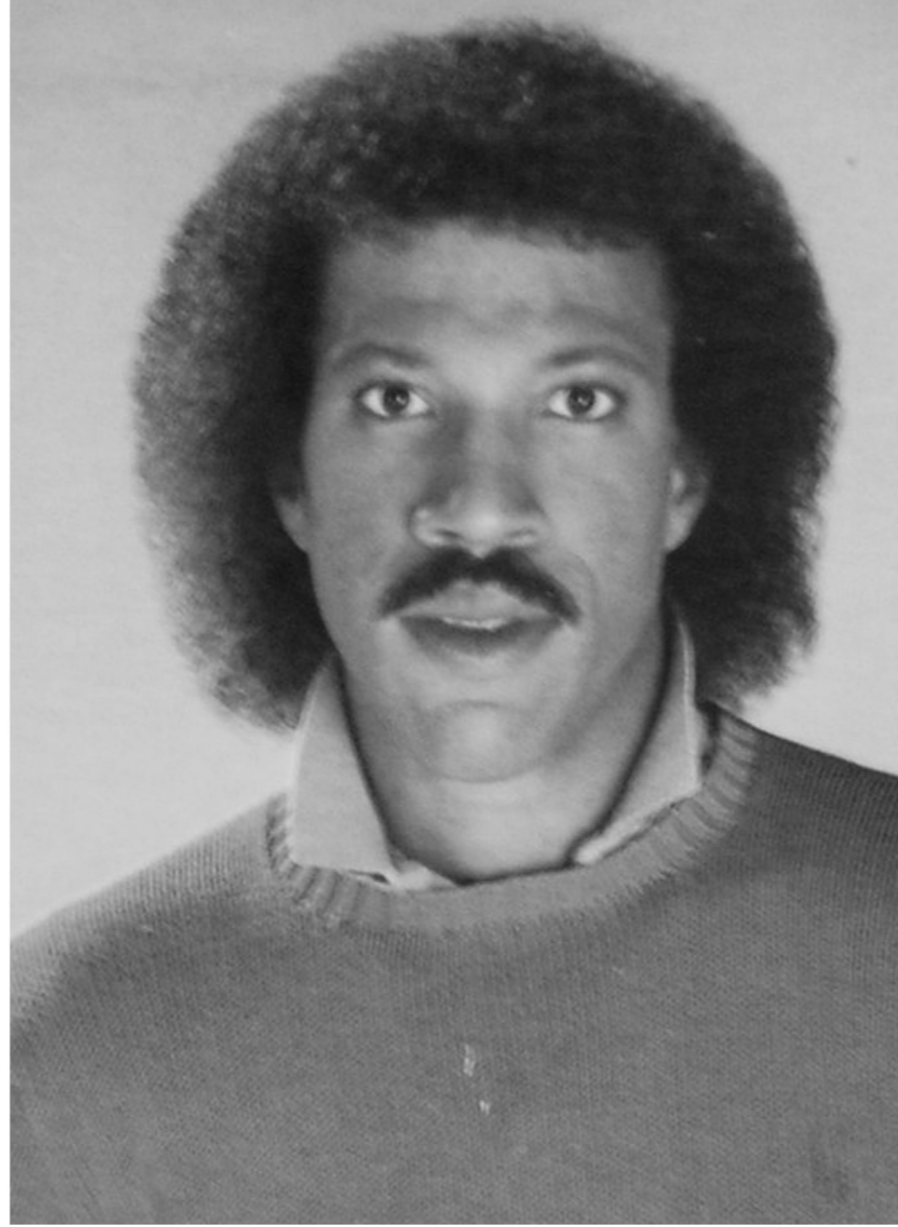
- Key Exchange
- Authentication
- Encipherment



The TLS 1.2 HANDSHAKE



Hello?



Is it me you're looking for?

I can see it in
your eyes

I can see it in
your smile

You're all I've
ever wanted

(and) my arms
are open wide

'Cause you know
just what to say

And you know
just what to do

And I want to
tell you so much

I love you

'Cause I wonder
where you are

And I wonder
what you do

Are you somewhere
feeling lonely?

Or is someone
loving you?

Tell me how to
win your hear

For I haven't got
a clue

But let me start
by saying

I love you

K-A-C

Key Exchange

Authentication

Cipher

ECDHE-RSA-AES256-GCM-SHA384

K - A - C

KAC1

KAC2

KAC3

>>>

KAC3

KAC2

KAC4

<<<

KAC3

Key Exchange

Static RSA - oldest form, take the pre-master secret and encrypt with the public key of the cert

DH - Diffie-Hellman with arbitrary group for pre-master secret

ECDHE - Diffie-Hellman with elliptic curves for pre-master secret



Key Exchange

Static RSA - No Forward Secrecy.
The NSA will retroactively decrypt
your conversations.

DH - People choose bad parameters
and there's no way to know.

ECDHE - You're cool, but drop the
old curves.





Authentication

Who you are is who you are.

Authentication in 1.2

- Certificate with public key (RSA or ECDSA)
- With RSA PKCS#1 1.5 is known to be fragile but no known direct attacks. PSS would be better.
- ECDSA: just don't reuse random nonce (Android PRNG, etc.)
- Use a strong hash function, MD5 collisions exist resulting in SLOTH



Authentication in 1.2

- What do you sign?
- Nonces and public key: No authentication of the cipher or curve choices, leading to FREAK, LogJam, CurveSwap
- Extended Master Secret: derive the key from the entire transcript to sure you can't just choose params so that two connections have the same keys (Triple Handshake)





Encryption

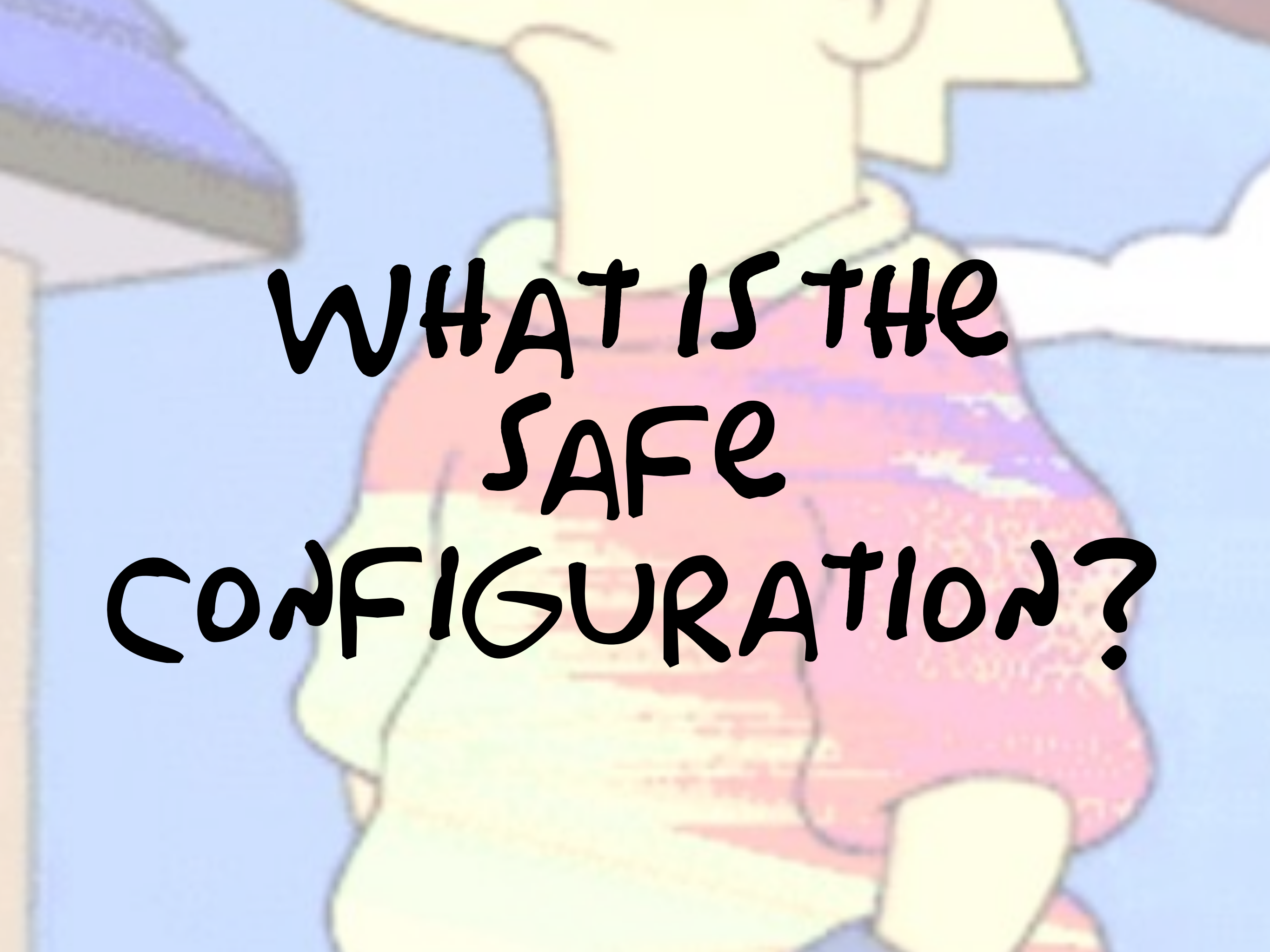
- CBC-mode ciphers with sign-then-encrypt: BEAST, padding problems galore (Lucky 13), birthday collisions (SWEET32)
- Only stream cipher is RC4: predictable
- TLS 1.2 introduced AEAD: AES-GCM, ChaCha20/Poly1305

Session Resumption

A grand hall with a blue carpet and a screen displaying 'WELCOME'. The hall is flanked by white columns topped with statues. The screen is the central focus, showing the word 'WELCOME' in yellow letters on a blue background.

Encrypt the session keys with a session ticket key (STK)

This makes the STK a long-term secret that kills forward secrecy

The background is a colorful, abstract illustration. It features a central yellow figure that resembles a person or a stylized character, with a pink and purple shape below it. The overall style is reminiscent of a children's book or a whimsical cartoon. The text is overlaid on this background.

WHAT IS THE
SAFE
CONFIGURATION?

- AEAD cipher (RC4 and CBC vulns)
- EMS (FREAK/LogJam, Triple Handshake, etc.)
- ECDHE (new point per connection)
- Restricted resumption

- MENU -



PAST
PRESENT
FUTURE

Fixing T L S

- TLS 1.3 Draft 00 on April 17, 2014
- Currently: Draft 18
- It's 118 pages vs. 104 for TLS 1.2



G O A L S

- Remove broken cryptography
- Clear, simple to implement specification
- Formal verification
- Backwards compatibility
- Make the handshake faster (more on that)



K, A, C

K1 A1 C1

K2 A2 C2 >>>

K3 C3

K3, K2
A2
C2, C3

<<< K3, A2, C2

Key Exchange

ECDHE (no weak curves)

x25519, x448 for djb hipsters

ffdhe (safe groups)

Authentication

RSA-PSS

ECDSA

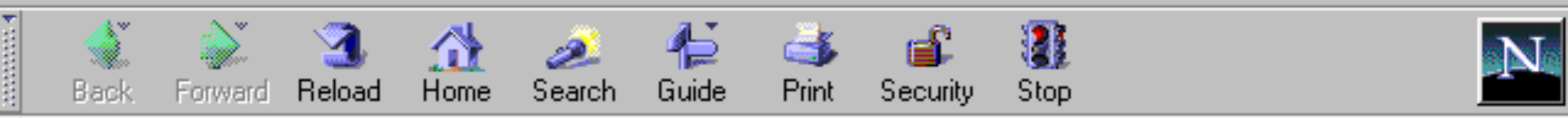
Entire transcript is signed

Cipher

AEADs only

AES-GCM, ChaCha20-Poly1305

No weak KDFs (SLOTH)



Bookmarks Location: about:

Instant Message Internet Lookup New&Cool

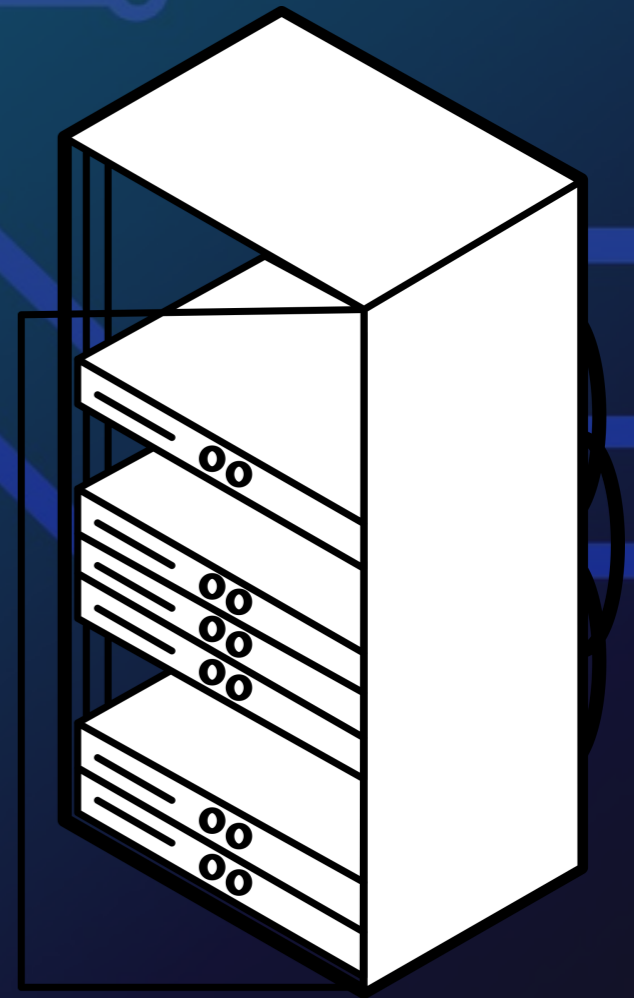
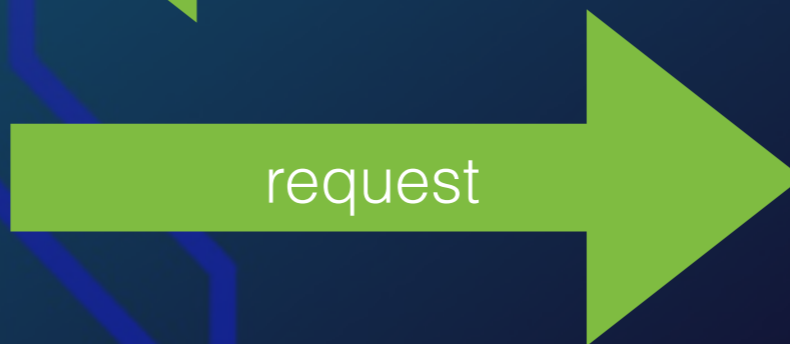
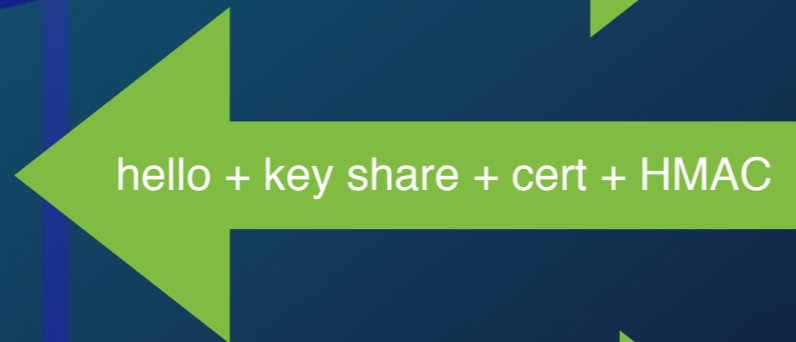
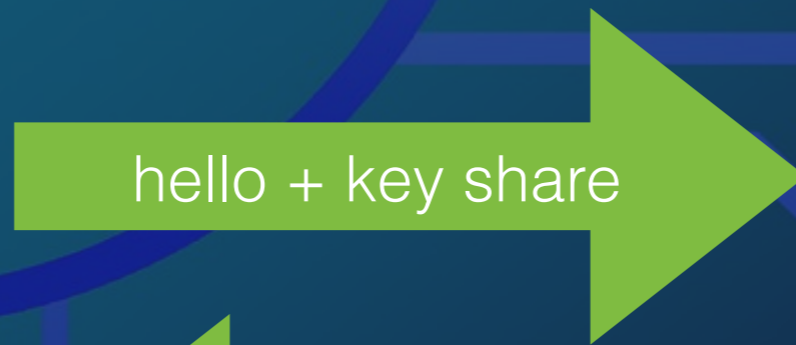
■ Secure Connection

The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.3), a strong key exchange (X25519), and a strong cipher (AES_128_GCM).

The TLS 1.3 HANDSHAKE



Client



Server

The TLS 1.3 HANDSHAKE



Session Resumption

Encrypt the resumption master secret with a session ticket key (STK)

New sessions use new key exchange

BUILDING AND DEPLOYING TLS 1.3

Cloudflare's stack

OpenSSL

|

nginx

|

origin

Go Go Go

- Let's build a TLS 1.3 stack in Go: `tls-tris`
- Hand off the TCP socket from nginx to a Go-based reverse proxy using `tris`.
- Inspect first two bytes, if 3.4, send to Go. Go can accept or reject based on customer settings.

Cloudflare's stack

OpenSSL

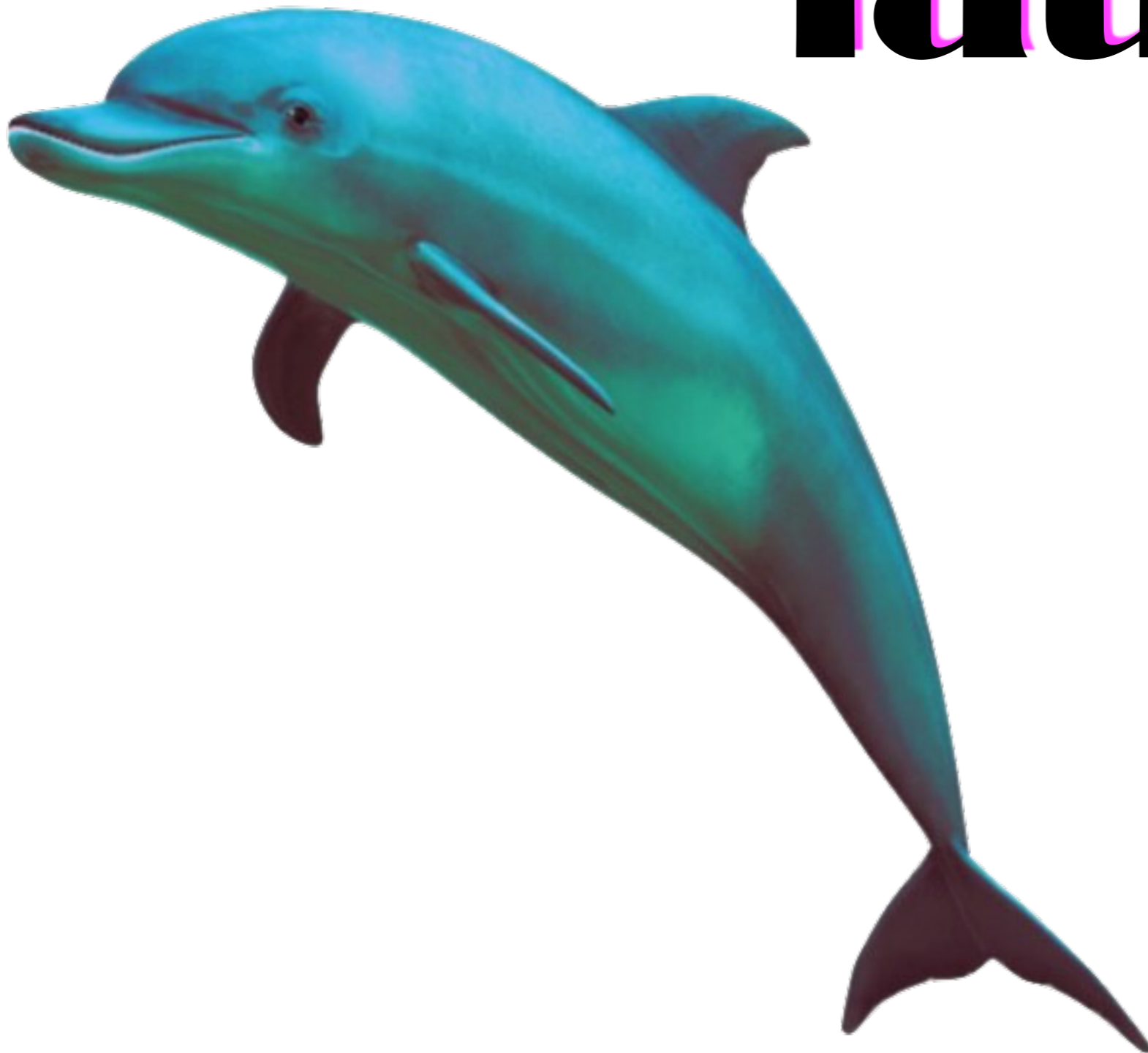
| |

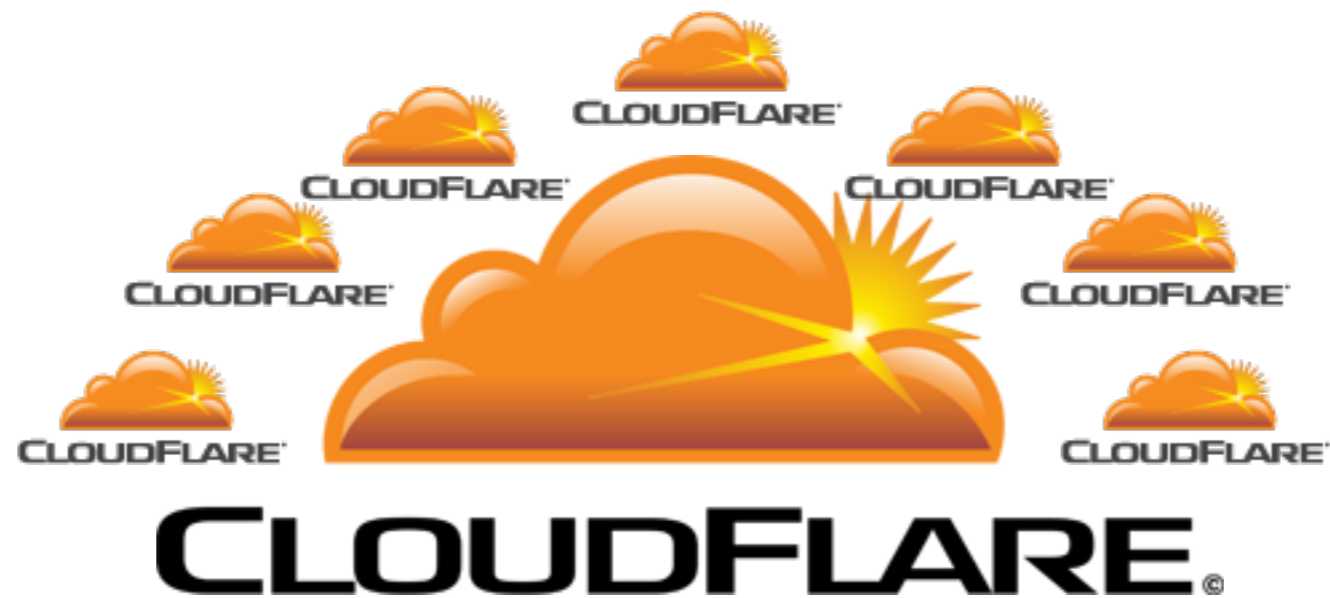
tris nginx

| |

origin

The big launch





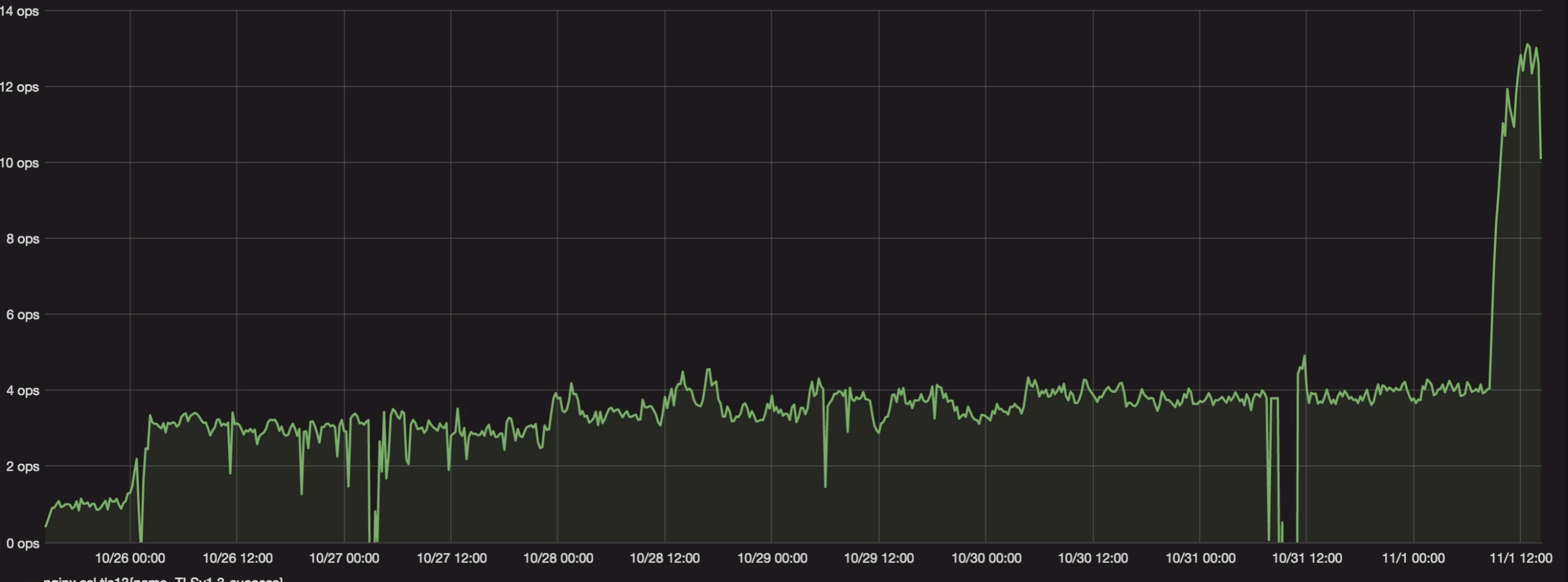
Encryption Week

Enabled for >3 million sites

September 20th

NGINX

Worldwide Nginx Upgrades



Launch

- Draft 14 support
- Firefox Nightly and Chrome Canary, but disabled by default
- We only saw around 1 connection per second globally



Compatibility error



Your dreams and hopes are incompatible with reality.

Force things

Accept reality

Kill myself

Version Intolerance

- Version number 3.4 breaks $>2\%$ of servers
- Chrome could either
 - Break these sites
 - Implement insecure fallback
 - Lobby the IETF to change the negotiation



Version Intolerance

- Version number in Draft 16 is now 3.4
- TLS 1.3 negotiated via an extension
- Our implementation was broken for a week
- SSL Labs is still broken



Amazing!

- MENU -

PAST

PRESENT

FUTURE



The future of tls-tris

Attempting to upstream to Go
standard library

NCC Group audit



- Chrome Canary enabled field test
- Firefox Nightly enabled by default
- Firefox 52 (March 2017) on by default
- OpenSSL 1.1.1 in 6 months

- Draft 18 submitted for last call
- Final submission IESG: January 2017

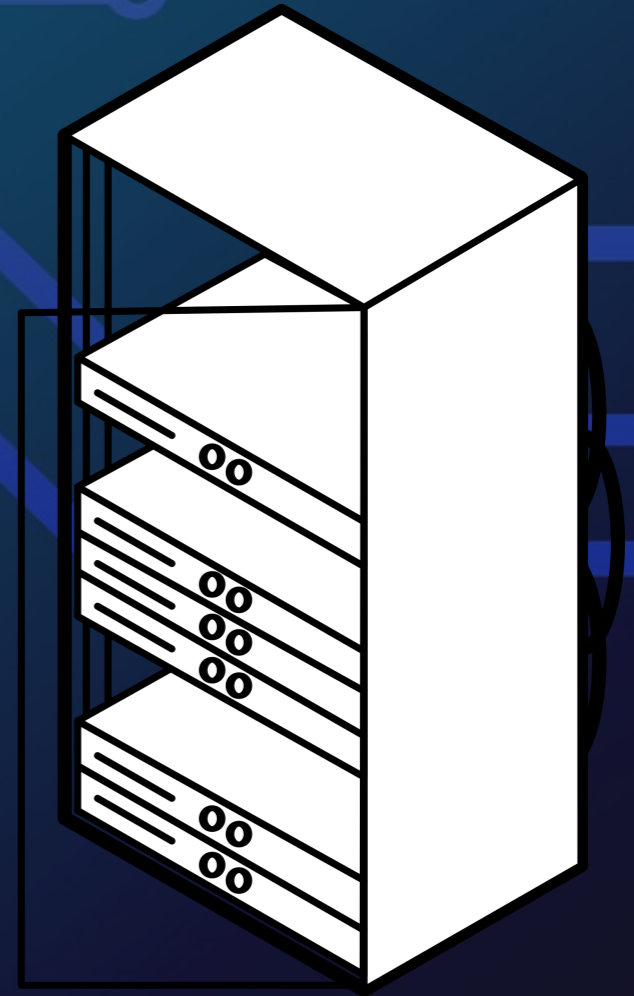
The TLS 1.3 0-RTT HANDSHAKE



Client

hello + key share + request

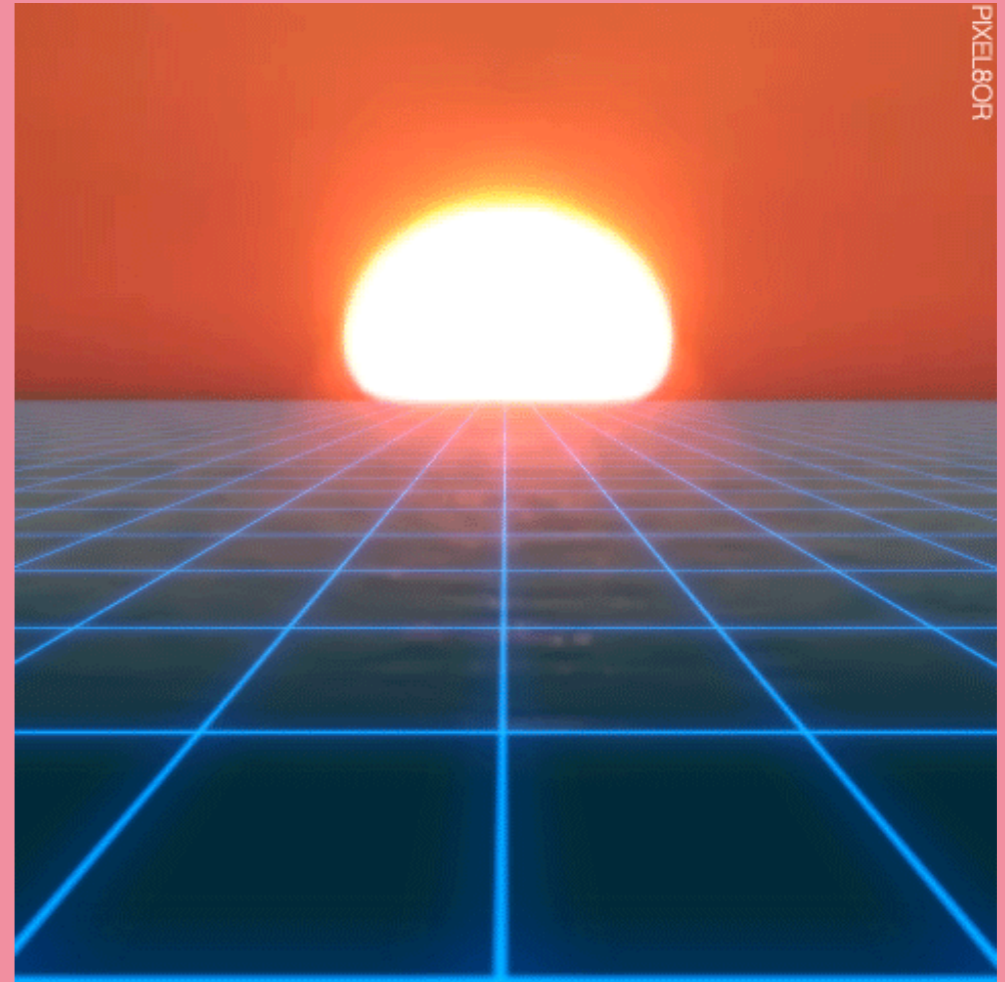
hello + key share + cert + HMAC +
response



Server

0-RTT Is Replayable

- Requests should be idempotent
- Idempotent requests can leak data
- Small time window



0-RTT Attack



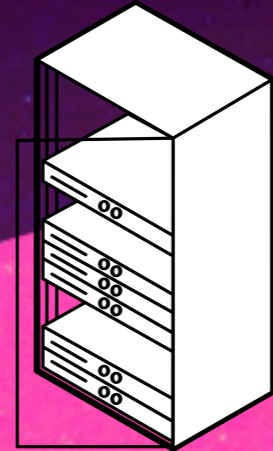
Client

hello + key share + POST request

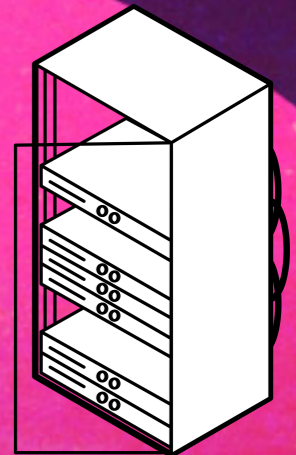


Attacker

hello + key share + POST request



Server



DB

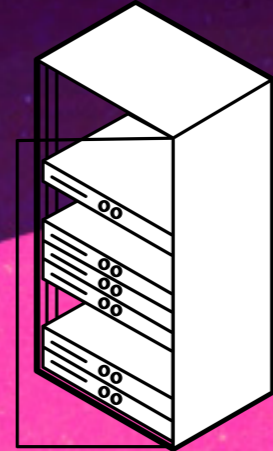


0-RTT Attack



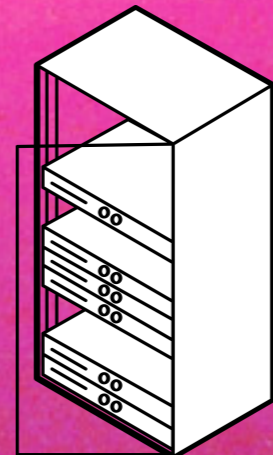
Client

hello + key share + GET request



Attacker

hello + key share + GET request



hello + key share + cert + HMAC +
response

Server





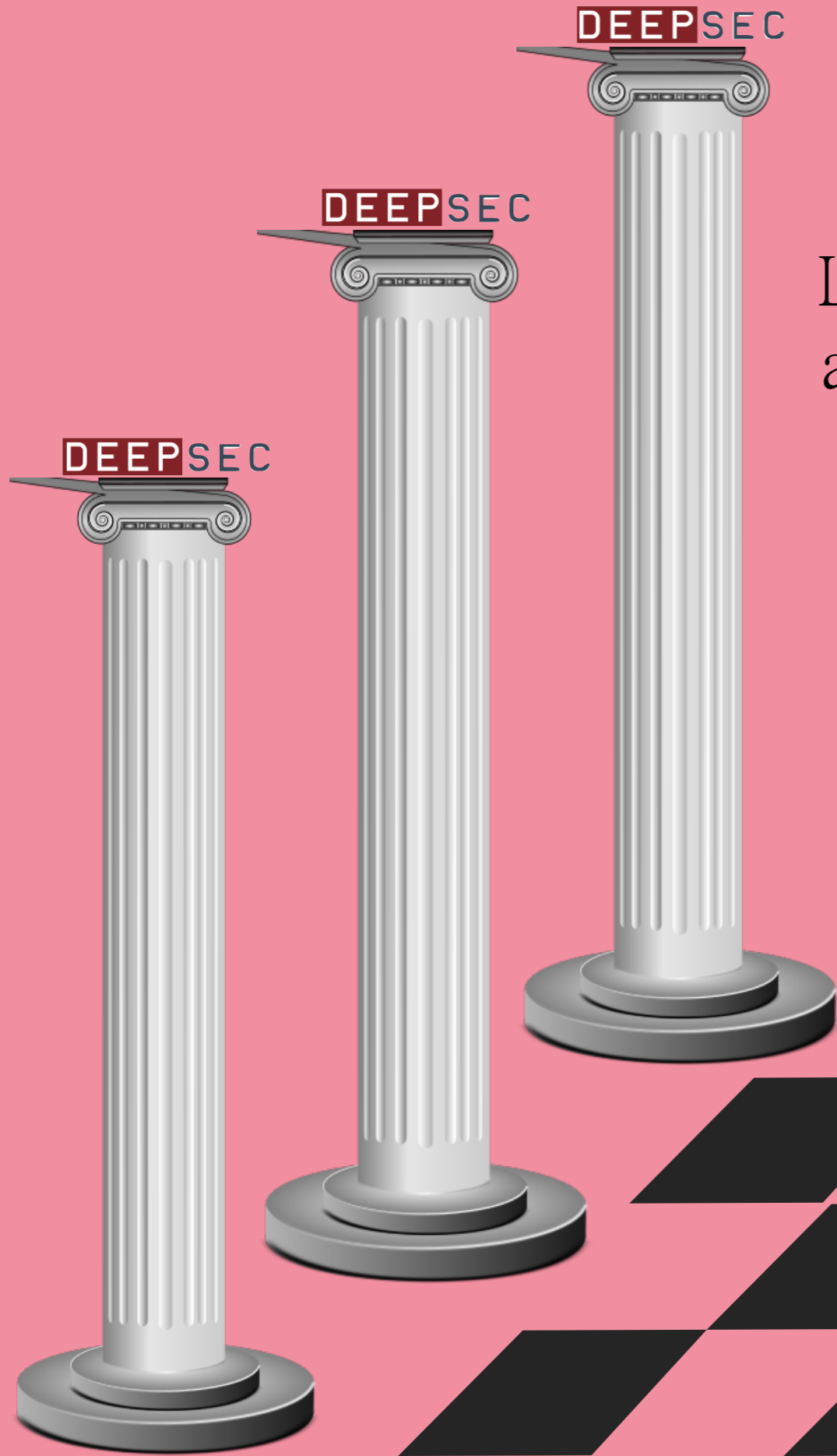
“It’s a superb
thing.”

-Tim Cook on encryption

STOP

SP

0:40:00



TLS 1.3

Lessons Learned from Implementing
and Deploying the Latest Protocol



Nick Sullivan
@grittygrease

November 11, 2016