Catching up with…

Internet Protocol Evolution

Mark Nottingham

QUIC

A long time ago in a working group far, far away…

- **HTTP/1** used multiple TCP connections for parallelism
  - This caused congestion control / fairness problems…
  - … and was still fundamentally limited.
- **HTTP/2** introduced multiplexing
  - Now, a single connection per origin was possible.
  - Successfully deployed.
  - BUT…

TCP Head of Line Blocking

Enter QUIC

- Google project (again) to evolve Internet protocols
- Started ~2013; now 30%+ of Google's egress traffic
- New transport protocol for HTTP over UDP - "gQUIC"
- Always encrypted
- Now an IETF Working Group - "iQUIC"

gQUIC Layering

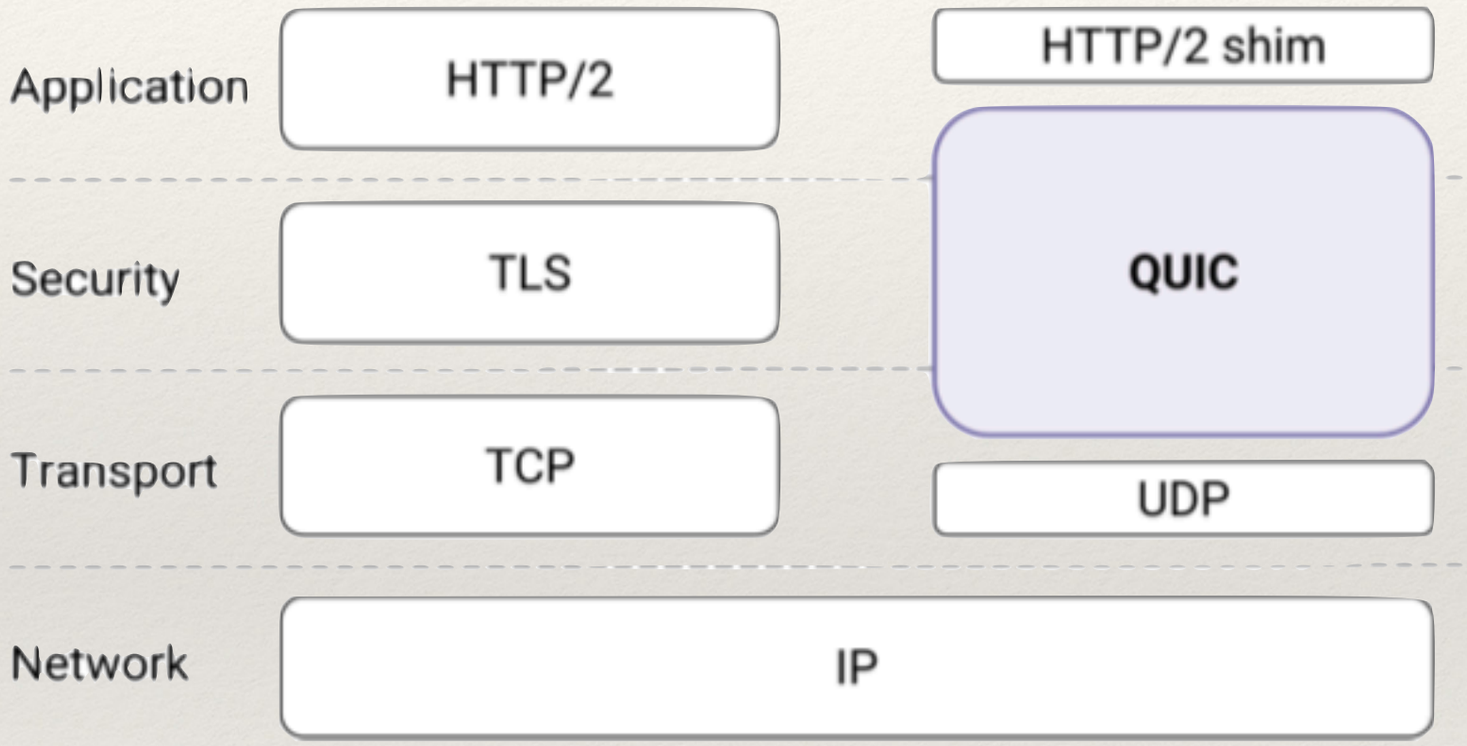


Figure 1: QUIC in the traditional HTTPS stack.

gQUIC Results

		% latency reduction by percentile						
		Lower latency				Higher latency		
	Mean	1%	5%	10%	50%	90%	95%	99%
Search								
Desktop	8.0	0.4	1.3	1.4	1.5	5.8	10.3	16.7
Mobile	3.6	-0.6	-0.3	0.3	0.5	4.5	8.8	14.3
Video								
Desktop	8.0	1.2	3.1	3.3	4.6	8.4	9.0	10.6
Mobile	5.3	0.0	0.6	0.5	1.2	4.4	5.8	7.5

Table 1: Percent reduction in global Search and Video Latency for users in $QUIC_g$, at the mean and at specific percentiles. A 16.7% reduction at the 99th percentile indicates that the 99th percentile latency for $QUIC_g$ is 16.7% lower than the 99th percentile latency for TCP_g.

gQUIC Results

	% rebuffer rate reduction by percentile				
	Fewer rebuffers			More rebuffers	
Mean	< 93%	93%	94 %	95%	99%
Desktop 18.0	*	100.0	70.4	60.0	18.5
Mobile 15.3	*	*	100.0	52.7	8.7

Table 2: Percent reduction in global Video Rebuffer Rate for users in $QUIC_g$ at the mean and at specific percentiles. An 18.5% reduction at the 99th percentile indicates that the 99th percentile rebuffer rate for $QUIC_g$ is 18.5% lower than the 99th percentile rate for TCP_g. An $*$ indicates that neither $QUIC_g$ nor TCP_g have rebuffers at that percentile.

gQUIC Results

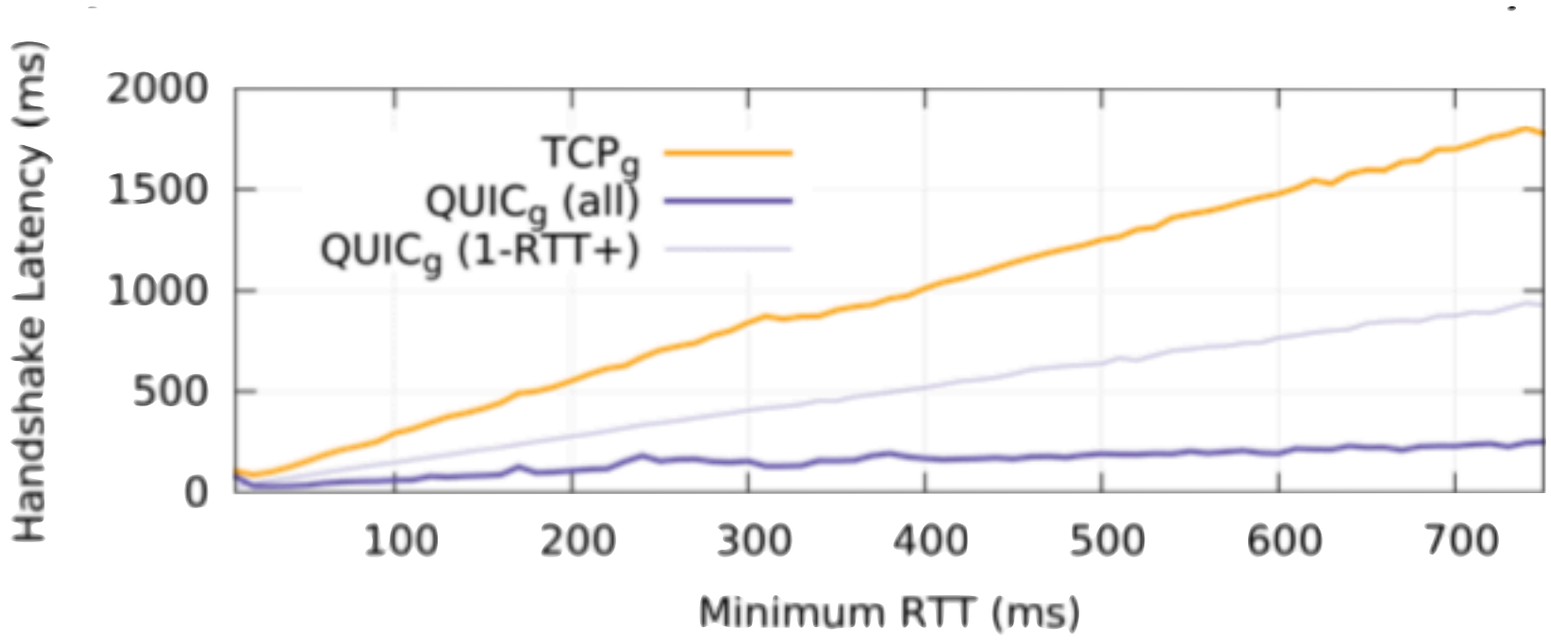


Figure 7: Comparison of handshake latency for QUIC$_g$ and TCP$_g$ versus the minimum RTT of the connection. Solid lines indicate the mean handshake latency for all connections, including 0-RTT connections. The dashed line shows the handshake latency for only those QUIC$_g$ connections that did not achieve a 0-RTT handshake. Data shown is for Desktop connections, mobile connections look similar.

Enter iQUIC

- Start with gQUIC
- Substantial rewrite of documents
- Use TLS 1.3 for handshake to derive session keys
- Initial focus on HTTP use case, other application protocols to follow

iQUIC Progress

- Currently on draft -07
- Holding third interop at Singapore IETF
  - More than ten partial experimental implementations
- Interop currently focusing on handshake and basic data transfer (HTTP/0.9 over QUIC)

While the door is open…

- One RT / Zero RT handshake (transport + crypto) ✅
- Mobility ❓
- Multipath ❓
- Forward Error Correction 🚫
- Middlebox accomodations ⁉️
- …

"The QUIC working group will provide a standards-track specification for a UDP-based, stream-multiplexing, encrypted transport protocol."

– *QUIC Charter*

Basic Questions

- What is a Stream? (issue #175)
  - Unidirectional? Bidirectional?
  - Reliable? Partially Reliable?
- What is an ACK frame? (issue #644)
- What should / can be encrypted? (various)

attractive noosens

FAIL

iQUIC's Current Focus

- V1 of QUIC will only worry about HTTP
- Subsequent versions will add things like multipath, etc.
- This implies that the V1 wire signature is invariant
- Straw-man V1 milestone: December 2018

https://quicwg.github.io

Interim Meeting in Melbourne: January 2018

But wait, there's more…

Ossification?

Ossification

- The Internet is big. Very big.
- If someone CAN do something, they will. Cf.
  - “Transparent” proxies
  - “Helpful” TCP optimisations
  - “Legal” pervasive monitoring
- We *can't* know about all of the ways people (ab)use protocols
- We can’t update the whole internet on a flag day

Ossification

- It's assumed that the Internet doesn't change. Cf.
  - TLS version numbers / extensions
  - HTTP methods
  - TCP options
- Extension points become "rusted" when they aren't used.

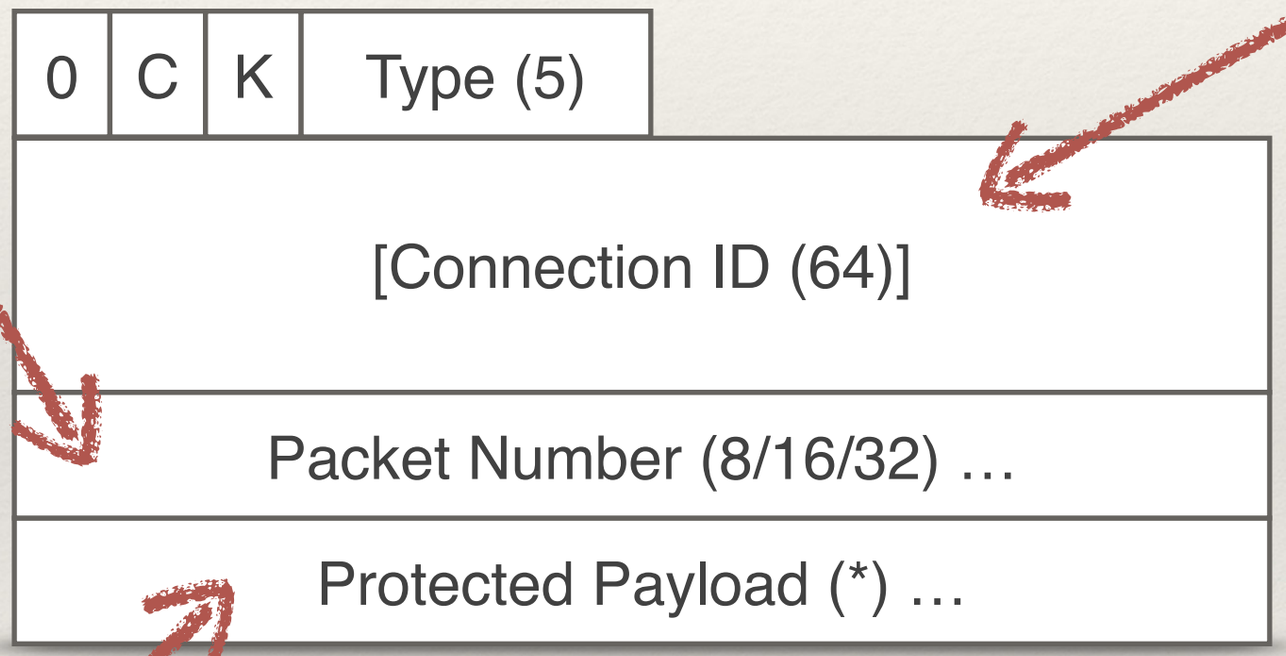
Designing Protocols Defensively

- **Encryption** - enforces two-party nature of protocols
- **Grease** - keeps intentional extension points available
- **Versioning** - regularly update protocols

Encryption in QUIC

monotonically increasing

optional



EVERYTHYING else is encrypted (and optionally padded)

Grease in QUIC

- *Greasing* assures that protocol extension points continue to be useable. E.g.,
  - Randomise port number usage (#495)
  - Add entropy to packet types (#311)
  - Protocol versioning (quic-transport, Section 4):
    “Versions that follow the pattern 0x?a?a?a?a are reserved for use in forcing version negotiation to be exercised.“

Versioning in QUIC

- Major protocol version defines message types, semantics, crypto layer
- Negotiated extensions can modify anything
- New versions can change anything
  - Document “invariants” explicitly
- New versions are expected to be fairly common

What does this mean for Networks?

"Because the communication subsystem is frequently specified before applications that use the subsystem are known, the designer may be tempted to "help" the users by taking on more function than necessary. Awareness of end-to-end arguments can help to reduce such temptations."

– End-to-End Arguments in System Design

TLS
Internet-Draft
Intended status: Informational
Expires: May 3, 2018

F. Andreasen
N. Cam-Winget
E. Wang
Cisco Systems

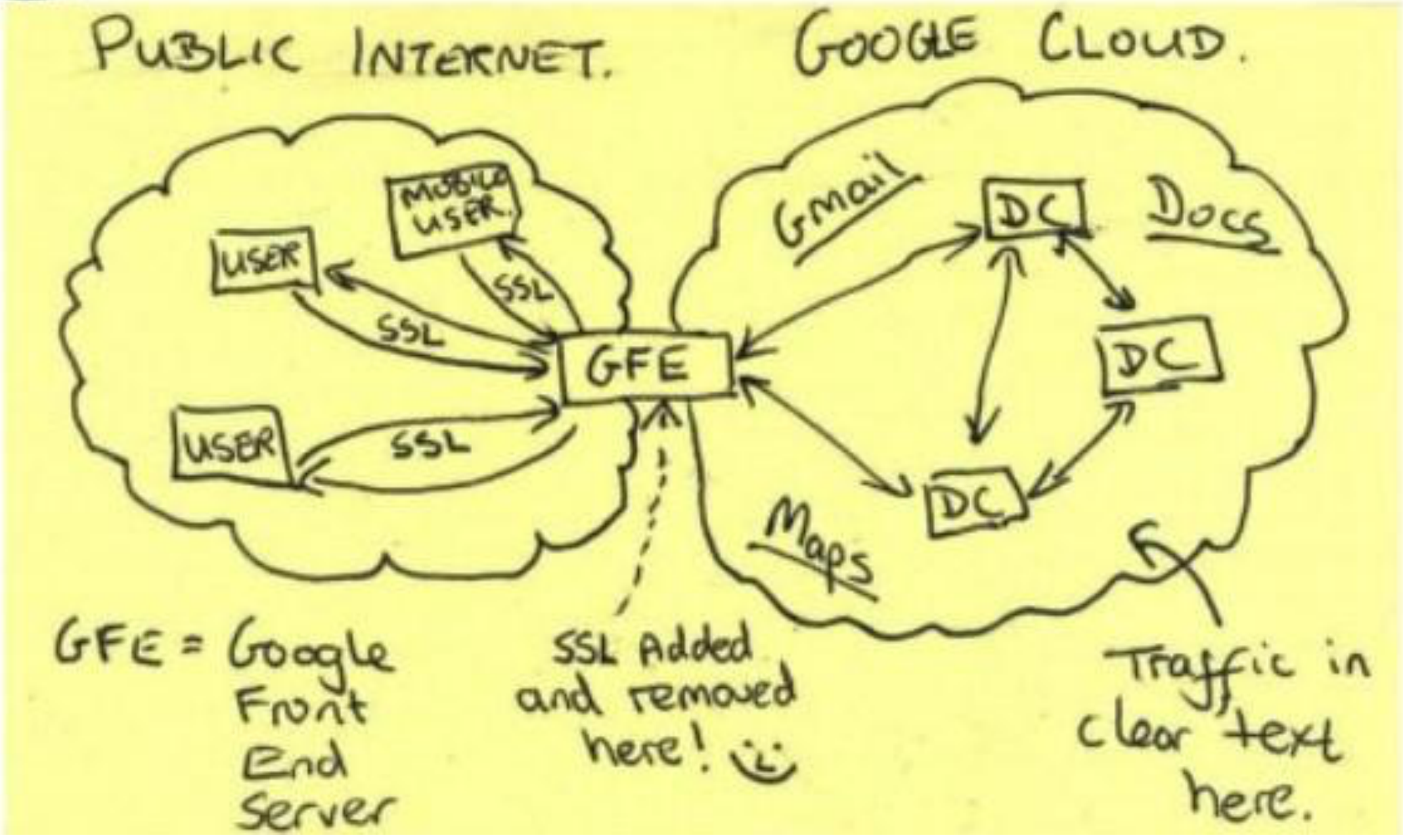
October 30, 2017

TLS 1.3 Impact on Network-Based Security
draft-camwinget-tls-use-cases-00

Abstract

Network-based security solutions are used by enterprises, public sector, and cloud service providers today in order to both complement and augment host-based security solutions. TLS 1.3 introduces several changes to TLS 1.2 with a goal to improve the overall security and privacy provided by TLS. However some of these changes have a negative impact on network-based security solutions. While this may be viewed as a feature, there are several real-life use case scenarios that are not easily solved without such network-based security solutions. In this document, we identify the TLS 1.3 changes that may impact network-based security solutions and provide a set of use case scenarios that are not easily solved without such solutions.

Current Efforts - Google



ZEYNEP TUFEKCI

TWITTER AND TEAR GAS

THE POWER AND FRAGILITY OF NETWORKED PROTEST

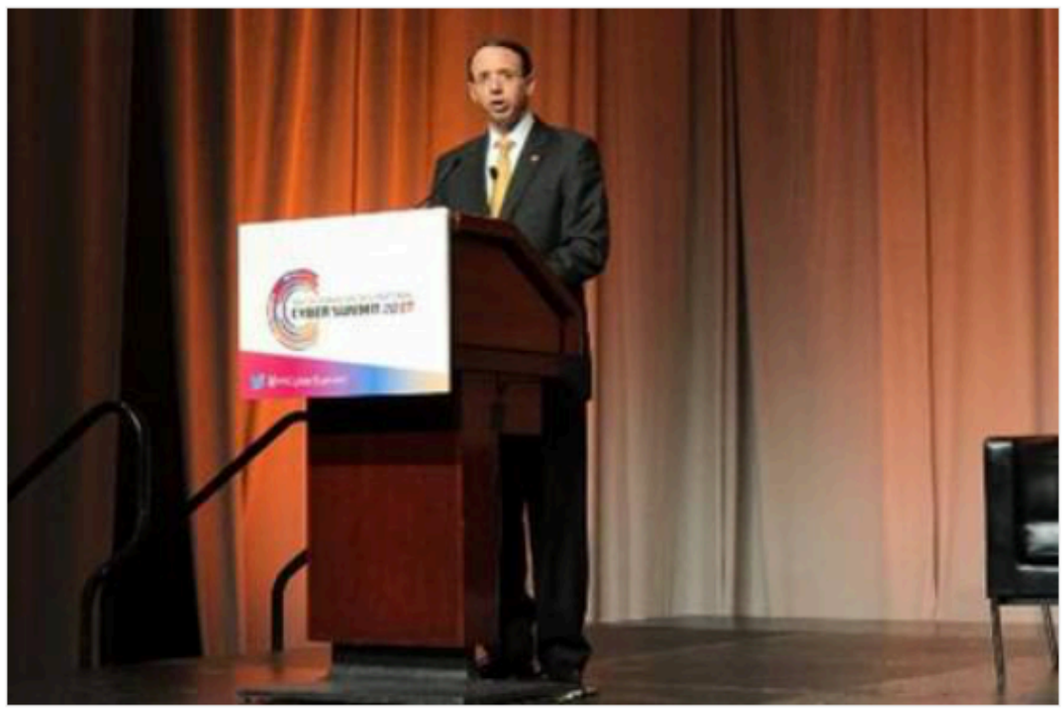
The Register®
Biting the hand that feeds IT

Security

Fine, OK, no backdoors, says Deputy AG. Just keep PLAINTEXT copies of everyone's messages

Sure, that won't go wrong at all

By Iain Thomson in San Francisco 30 Oct 2017 at 20:52 27 SHARE ▼



On stage today ... Rod Rosenstein has yet another bright idea

The US Deputy Attorney General has told business leaders that Uncle Sam won't demand mandatory backdoors in encryption – so long as companies can cough up an unencrypted copy of every message, call, photo or other form of communications they handle.

INDY/TECH

THERESA MAY TO CREATE NEW INTERNET THAT WOULD BE CONTROLLED AND REGULATED BY GOVERNMENT

One More Thing.

"This working group will standardize encodings for DNS queries and responses that are suitable for use in HTTPS. This will enable the domain name system to function over certain paths where existing DNS methods (UDP, TLS, and DTLS) experience problems."

–DNS Over HTTP (DOH!) Working Group Charter

“May you live in interesting times.”

–Sir Austen Chamberlain (probably)