### Measuring TLS key exchange with post-quantum KEM

The Second PQC Standardization Conference

AUGUST 22, 2019



# OIREAL WORLD CRYPTOGRAPHYO2DESIGNING THE EXPERIMENTO3OUR EXPERIMENTO4PRELIMINARY RESULTSO5QUESTIONS

#### INTRODUCTION: TEAM







Cloudflare @\_henrycase

### Adam Langley

Google @agl\_\_



**Nick Sullivan** 

Cloudflare @grittygrease



### **Dave Levin** University of Maryland

@DistributedDave



**Alan Mislove** Northeastern University @amislove



Luke Valenta Cloudflare @lukevalenta





## 

-

-----

BORD ROAD

NICHERAL )

ARCHURCHES.

CHARLEN AND



MOTIVATION

## How best to contribute to the PQC standardization process?

## Pick an ecosystem and explore the upgrade path.

Б

#### MOTIVATION



#### Meaningful deployment and impact

The cryptographic system should be widely deployed and relied on by millions of people for important communications.

#### Non-trivial complexity

Deployments with many different dependencies and complex moving parts are trickier to upgrade. Pick a project that helps uncover potential stumbling blocks.

### Methodology for migration established

There should be well-established requirements and norms for upgrading the ecosystem based on previous upgrades.

### **HTTPS in browsers**





### Meaningful deployment and impact

The cryptographic system should be widely deployed and relied on by millions of people for important communications.

#### Non-trivial complexity

Deployments with many different dependencies and complex moving parts are trickier to upgrade. Pick a project that helps uncover potential stumbling blocks.



### Methodology for migration established

There should be well-established requirements and norms for upgrading the ecosystem based on previous upgrades.

MOTIVATION

### Meaningful deployment and impact

#### Online life is encrypted

Web traffic is **75%** HTTPS

This includes almost all banking, eCommerce, social media, search, webmail, video streaming, news, and other online activity in a browser or mobile app.



### Non-trivial complexity



#### **Browser complexity**

- Modern "auto-update" browsers
- Older browsers (IE6/7)
- Embedded app browsing
- HTTPS API clients
- SSL libraries
- BoringSSL, NSS, CommonCrypto, sChannel...

### Other "hidden" participants

- Forward Proxies (with SNI-filtering)
- Anti-virus proxies
- TCP proxies of other types
- National Firewalls



Veb server developers: Market share of active sites



#### Server complexity

- Site abandonment
- Slow updates that require downtime
- Library dependencies: OpenSSL etc.
- TLS terminator boxes/cloud services



### Methodology for migration established

- HTTPS is HTTP with TLS/SSL
- TLS 1.2 standardized in 2008 as RFC 5246
- Not widely deployed in browsers until 2013-14
- No feedback loop from real world to standardization process

### Ruleofthumb

DEPLOYABLE PROTOCOLS

### Breakage rate should be comfortably below 1%

```
446
6, <u>5878</u>, <u>6176</u>, <u>7465</u>, <u>7507</u>, <u>7568</u>,
7, 7685, 7905, 7919, 8447
 Group
ments: 5246
 , <u>4346</u>, <u>4366</u>
```

lards Track

the Transport Layer Security (TLS) Protocol Version 1.2

#### Memo

t specifies an Internet standards track prot munity, and requests discussion and suggesti Please refer to the current edition of th Standards" (STD 1) for the standardiza protocol. Distribution of this memo

Specify Deploy Measure Adjust

### 2008

Publish the specification

**20I2** 

Use test population

PROF

#### **20I2-I3**

Detect various breakages

#### 2012-2014

Fix bad servers and middleboxes until breakage is low enough Insecure fallback



### **Rule of thumb**

**DEPLOYABLE PROTOCOLS** 

Breakage rate should be comfortably below 1%



### TLS1.3

## Specify Deploy Measure Adjust

### 2014-2018

**Develop draft specifications** 

### 2016-2018

Use test population

### 2016-2018

Detect various breakages

### 2016-2018

Change the specification to adjust for the reality of the world



### **NIST PQC Process**

Additional standards for incorporating PQC into applications TLS, SSH, etc.

Deploy widely

## Specify Deploy Measure Adjust

### How deployable?

Detect various breakages

### This experiment!

Change the specification to adjust for the reality of the world





## Designing the experiment



CPUUsage

### Protocol Shape

## **Key Size**



DESIGNING THE EXPERIMENT

### TLS 1.3



### **Protocol Shape**

### 1-RTT

The client initiates with its key share, the server responds with its key share, signature and potentially initial data. Post-quantum KEMs fit nicely into this model.

### No client challenge

There is no current requirement for a proof-of-work to be done on the client side. Expensive key exchange is a DoS risk.

#### DESIGNING THE EXPERIMENT

### **CPUUsage**

### **Higher latency**

If an algorithm takes clock time on the order of network time (10ms+), it may noticeably delay the connection.

#### Power consumption

Mobile and lightweight devices optimize for battery life. Expensive computations hurt this.

#### Asymmetry risk

If an attacker can cause a lot of work using only a small amount of work, it is an amplification attack vector.







#### The Original Packet



### **Key Size**

### Hard-coded sizes in legacy systems Network congestion causes latency Unknowns

### Previous work

### Post-quantum confidentiality for TLS

2018 experiment by Adam Langley to measure impact of key size on latency. Implemented "dummy" extension to simulate larger key sizes.

Control group: no extension sent Supersingular isogenies (SI): 400 bytes Structured lattices (SL): 1100 bytes Unstructured lattice standing (ULS): 3300 bytes





### Results

Additional latency over control group (ms) Configuration

	Structured Lattices	Supersingular Isogenies
Mobile, Median	9.6	3.5
Mobile, 95th	159	18.4
Desktop, Median	5.5	2.6
Desktop, 95th	137	19.2







## Our experiment



OUR EXPERIMENT

## Implement and deploy two realistic key agreement ciphers

### CECPQ2

![](_page_21_Picture_3.jpeg)

CECPQ2b

![](_page_21_Picture_5.jpeg)

## Implement and deploy two realistic key agreement ciphers

### CECPQ2

### **NTRU-HRSS**

- Closest to "ntruhrss701" from Round 1
  - NIST level 1
- 1138 byte public key/ciphertext
- C, x86-64 ASM, aarch64 ASM
- Fast KeyGen, Encaps, Decaps
  - 4000, 76000, 22000/s on Skylake (**<1ms**)
  - 2057, 33287, 13605/s on ARM Cortex-A75

### CECPQ2b

### SIKE/p434

- Round 2 submission
  - NIST level 1
- 330 byte public key, 346 byte ciphertext
- C, x86-64 ASM, aarch64 ASM
- **Slow** KeyGen, Encaps, Decaps
  - 420, 265, 245/s on Skylake (~**5ms**)
  - 269, 165, 155/s on ARM Cortex-A75

OUR EXPERIMENT

### **Some tweaks**

### No SHA-3

HKDF-SHA-2 instead of SHA-3 for KEM

### **Code available**

### Hybrid mode with x25519

Concatenate and combine with HKDF

### HTTPS://BORINGSSL.GOOGLESOURCE.COM/BORINGSSL/

## You could be using postquantum cryptography right now

### Where is it deployed

### Cloudflare edge

Over 20 million Internet properties Both suites supported Located **"close"** to users

Note: SIKE not cleared for Google servers due to DoS risk

### **Chrome Canary**

A diverse group of beta testers Computers, some ARM devices 1/3 control

Typically worse-than-average networking and PQ limited to x86-64 and Aarch64, i.e. much better-than-average CPUs.

![](_page_25_Figure_8.jpeg)

![](_page_26_Figure_0.jpeg)

![](_page_27_Picture_0.jpeg)

![](_page_27_Picture_1.jpeg)

![](_page_27_Picture_2.jpeg)

### handshake duration - 30m buckets - median

![](_page_28_Figure_1.jpeg)

### handshake duration - 30m buckets - 95th percentile

![](_page_29_Figure_1.jpeg)

![](_page_30_Figure_0.jpeg)

![](_page_30_Figure_2.jpeg)

### handshake duration - logscale x-axis - Android

#### SERVER-SIDE RESULTS

![](_page_31_Figure_1.jpeg)

Additional latency over control group w/ 95% confidence intervals (ms)

ECPQ2	CECPQ2b	
N/A	[51, 63]	
N/A	[47, 62]	
5, 537]	N/A	
N/A	N/A	
.9, 3.1]	[16, 20]	
9, 100]	[48, 74]	

\* only 64-bit devices

#### **CLIENT-SIDE RESULTS**

Chrome is measuring TLS handshake latency, not including TCP connection costs.

The metrics system calculates 95% confidence intervals for changes and a result is considered "significant" if no-change isn't inside that interval.

W

Win

W

And

Configuration	Additional latency with 95% confidence		
	CECPQ2	CECPQ2b	
Windows, 25th	N/A	[53%, 102%]	
/indows, Median	N/A	[20%, 76%]	
Windows, 99th	N/A	N/A	
Android, 25th	N/A	[30%, 96%]	
Android, Median	N/A	N/A	
Android, 99th	[19%, 278%]	N/A	

SIKE has very expensive floor with respect to latency

Likely not to improve due to Moore's law and growth of IoT

Key size plays a factor at 95th percentile, but more data needed

![](_page_33_Figure_4.jpeg)

,
10
34

PRELIMINARY CONCLUSIONS

### More measurements to be done

Investigation of configuration of clients with particularly bad NTRU performance

![](_page_34_Picture_3.jpeg)

![](_page_34_Picture_4.jpeg)

![](_page_35_Picture_0.jpeg)

## Questions?

![](_page_35_Picture_2.jpeg)

### Measuring TLS key exchange with post-quantum KEM

The Second PQC Standardization Conference

AUGUST 22, 2019

![](_page_36_Picture_3.jpeg)