Post-Quantum

Cryptography Conference

Birth of the post-quantum Internet

Bas Westerbaan
Research Engineer at Cloudflare

Dawn of the Post-Quantum Internet

Dr Bas Westerbaan, Cloudflare Research

PKI Consortium Post-Quantum Cryptography Conference, AMS, Nov 7th, 2023

About Cloudflare

We run a global network spanning 300 cities in over 100 countries.

Started of as a CDN and DDoS mitigation company, we now offer many more services, including

- 1.1.1.1, public DNS resolver
- Workers, serverless compute
- SASE, to protect corporate networks

We serve nearly 20% of all websites and process 46 million HTTP requests per second.

Building a better Internet

Cloudflare cares deeply about a private, secure and fast Internet, helping design, and adopt, among others:

- Free SSL (2014), TLS 1.3 and QUIC
- DNS-over-HTTPS
- Private Relay / OHTTP
- Encrypted ClientHello

And, the topic today:

- Migrating to post-quantum cryptography.

This talk

Overview of the current state of migration of the Internet / WebPKI, and its unique challenges.

Changing the Internet / WebPKI is hard

- Very diverse. Many different users / stakeholders with varying (performance) constraints and update cycles.

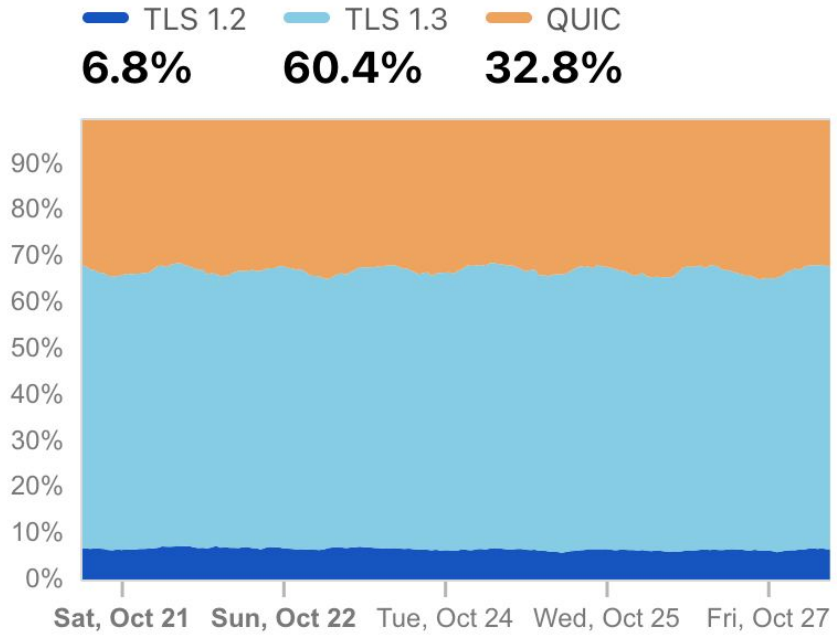
  We can't assume everyone is on fiber, or uses modern CPU, can store state, or can update at all.

- Protocol ossification. Despite being designed to be upgradeable, any flexibility that isn't used in practice, is probably broken, because of faulty implementations.

TLS 1.3 migration

Early versions of TLS 1.3 were completely undeployable because of protocol ossification.

After six more years of testing and adding workarounds, the final version of TLS 1.3 is a success, used by over 90% of our visitors.



Cloudflare Radar

There will be *two* post-quantum migrations.

1. Key agreement 🤝

   Communication can be recorded today and decrypted in the future. We need to upgrade as soon as possible.

2. Signatures 

   Less urgent: need to be replaced before the arrival of cryptographically-relevant quantum computers.

Key agreement 🤝

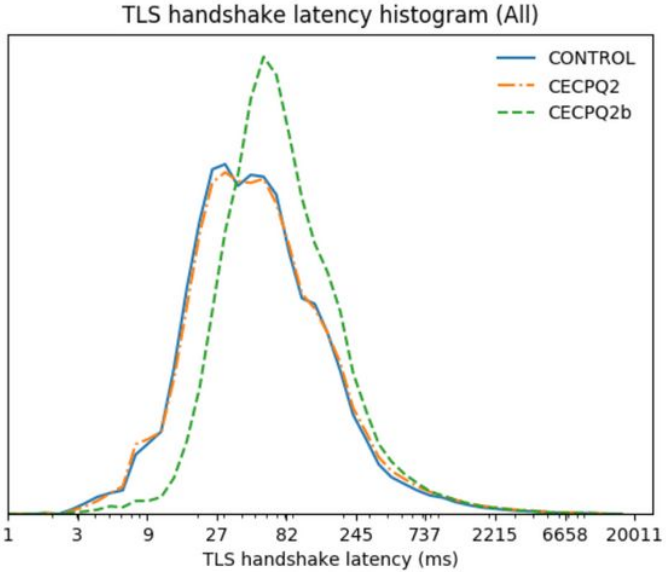
Urgent, and the *easier* one.

Feasibility study with Chrome

In 2019 we performed large-scale test of PQ kex with Chrome. Takeaways:

- Performance of lattice-based KEMs is acceptable.
- Significant amount of broken clients because of protocol ossification (*split ClientHello*.)

Google has been working with vendors to fix issues.



X25519. CECPQ2 is X25519+NTRU-HRSS (lattice) and CECPQ2b is X25519+SIKE (isogenies, broken)

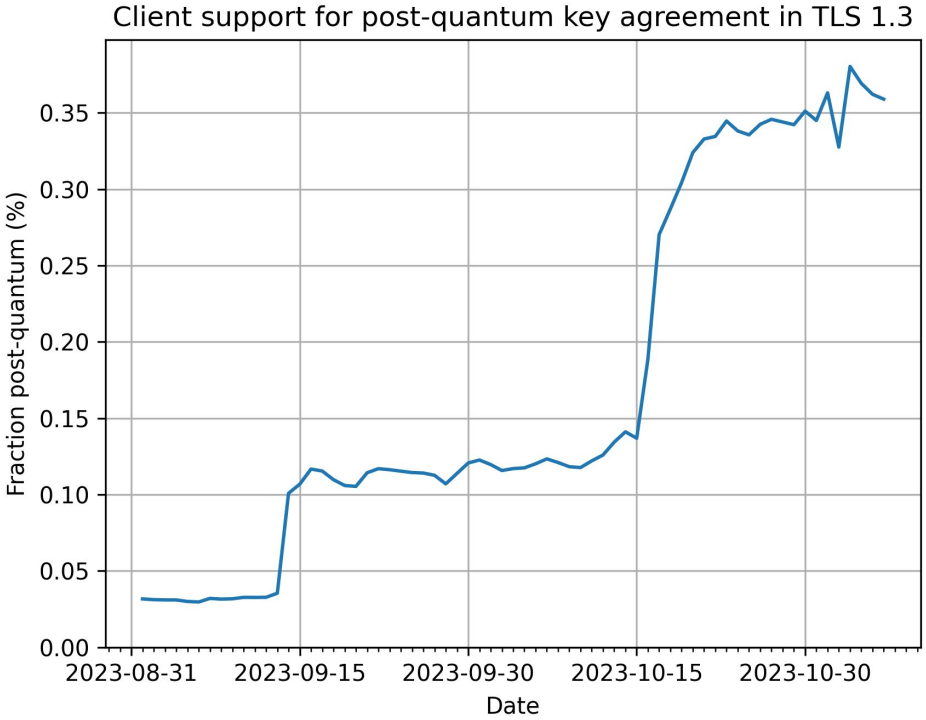
Early deployments

2022 coordinating at IETF, we enabled post-quantum key agreement (~20% Internet.)

In 2023 Google enabled server-side as well.

Browsers:

- Chrome. Enabled for 1% of all traffic.
- Firefox. Expected in 2024.



Promising early results

As of writing, no hard failures for further roll-out identified by Chrome 🤞. Our own testing has shown that there are about a hundred customers with incompatible origin servers. We're reaching out to help fix them.

It is likely that we will see double-digit percentage post-quantum key-agreement in 2024.

Key agreement 🤝

Urgent and the easier of the two to deploy. We're on track for ~30% client-side deployment in 2024.

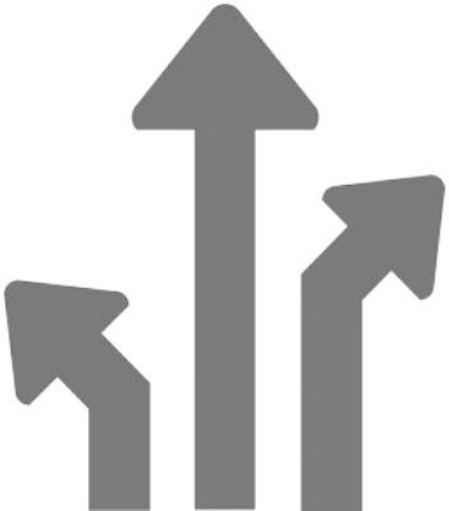
That took 5 years.

Signatures 

Less urgent, but much more challenging.

#1, many more parties involved:

Cryptography library developers, browsers, certificate authorities, HSM manufacturers, CT logs, and every server admin that cobbled together a PKI script.



#2, there is no all-round great PQ signature

		Size (bytes)		CPU time (lower is better)	
	PQ	Public key	Signature	Signing	Verification
Ed25519	❌	32	64	1 (baseline)	1 (baseline)
RSA-2048	❌	256	256	70	0.3
Dilithium2	✅	1,312	2,420	4.8	0.5
Falcon512	✅	897	666	8*	0.5
SPHINCS⁺128s	✅	32	7,856	8,000	2.8
SPHINCS⁺128f	✅	32	17,088	550	7

Online signing — Falcon's Achilles' heel

- For fast signing, Falcon requires a floating-point unit (FPU).
- We do not have enough experience running cryptography securely (constant-time) on the FPU.
- On commodity hardware, Falcon should not be used when signature creation can be timed, eg. TLS handshake.
- Not a problem for signature verification.

#3, there are many signatures on the Web

- Root on intermediate
- Intermediate on leaf
- Leaf on handshake
- Two SCTs for Certificate Transparency
- An OCSP staple

Typically 6 signatures
and 2 public keys
when visiting a website.

(And we're not even counting DNSSEC.)

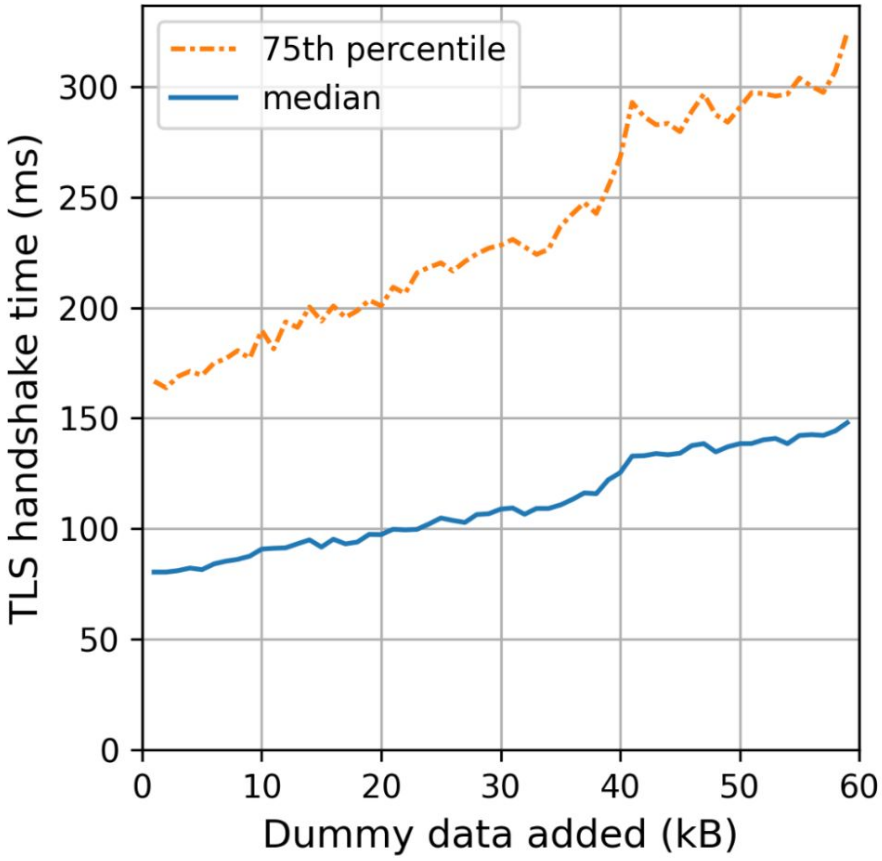
Using only Dilithium2

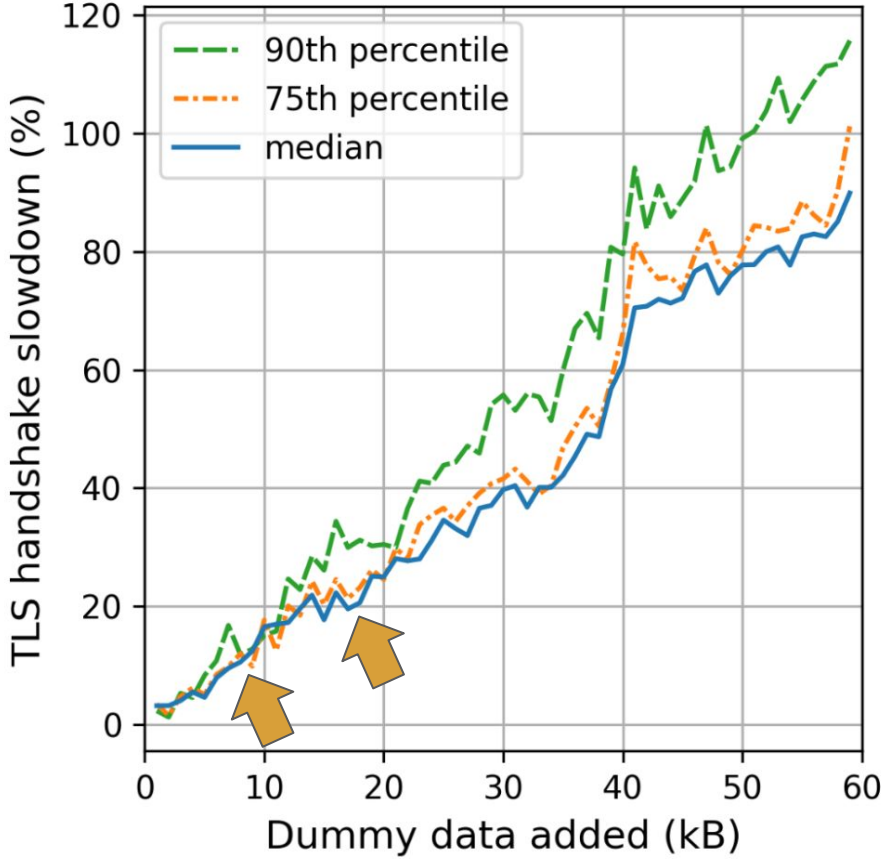
+17,144 bytes

Using Dilithium2 for the TLS handshake and Falcon for the rest

+7,959 bytes

Is that too much? We had a look...





blog.cloudflare.com/sizing-up-post-quantum-signatures, 2021

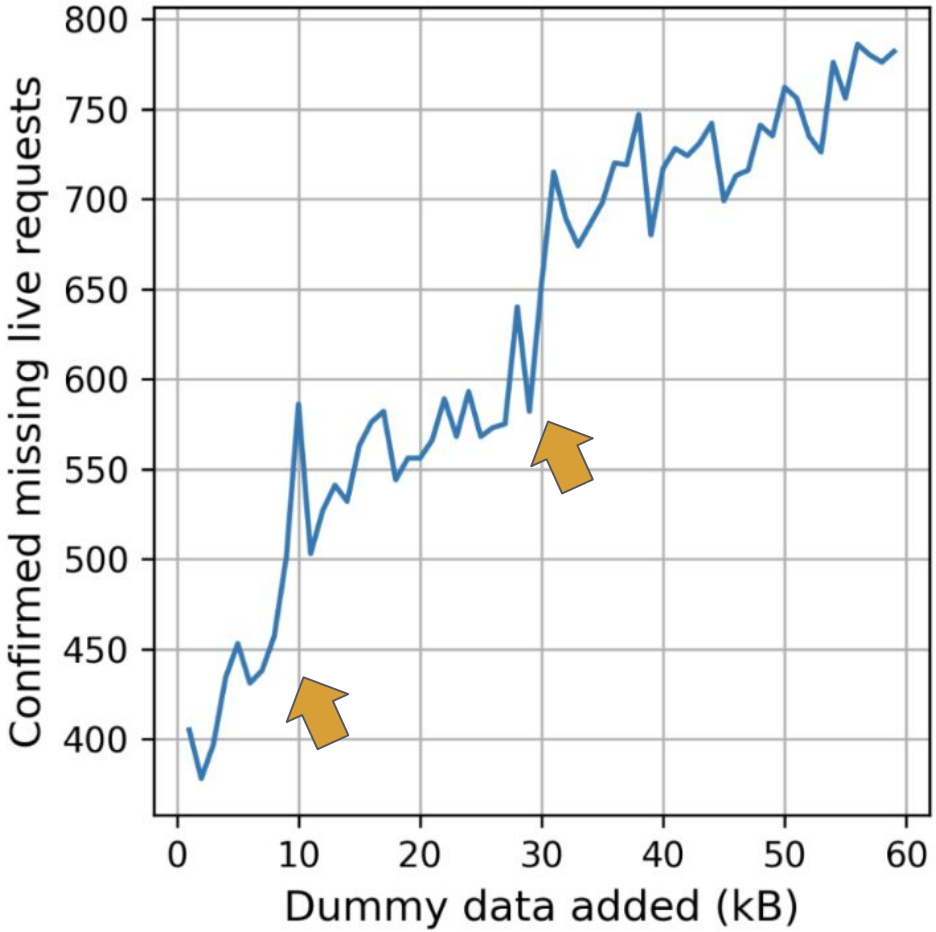
And, of course…

Protocol ossification

Bump in missing requests suggests some clients or middleboxes do not like certificate chains longer than 10kB and 30kB.

This is problematic for composite certificates.

Instead configure servers for multiple separate certificates and let TLS negotiate the one to send.



Not great, not terrible

It probably won’t break the Web, but the performance impact will delay adoption.

NIST signature on-ramp

NIST took notice and has called for new signature schemes to be submitted.

I will cover these in a break-out presentation tomorrow.

The short of it: there are some very promising submissions, but their security is as of yet unclear.

Thus, we cannot assume that a new post-quantum signature will solve our issues.

In the meantime

There are small and larger changes possible to the protocols to reduce the number of signatures.

- Leave out intermediate certificates.
- Use key agreement for authentication.
- Overhaul WebPKI, eg. Merkle Tree Certificates.

I will discuss these in more depth in the break-out tomorrow.

Signatures 

Less urgent, but path is unclear. Real risk we will start migrating too late.

That’s not all: the Internet isn’t just TLS

There is much more cryptography out there with their own unique challenges.

- DNSSEC with its harder size constraints
- Research into post-quantum privacy enhancing techniques, eg. anonymous credentials, is in the early stages.

Thank you, questions?

References

- Follow along at the IETF
- Check out our blog, eg.:
  - 2019 TLS experiment with Google
  - Sizing-up Post-Quantum Signatures
  - Deploying Kyber worldwide
- Reach out: ask-research@cloudflare.com

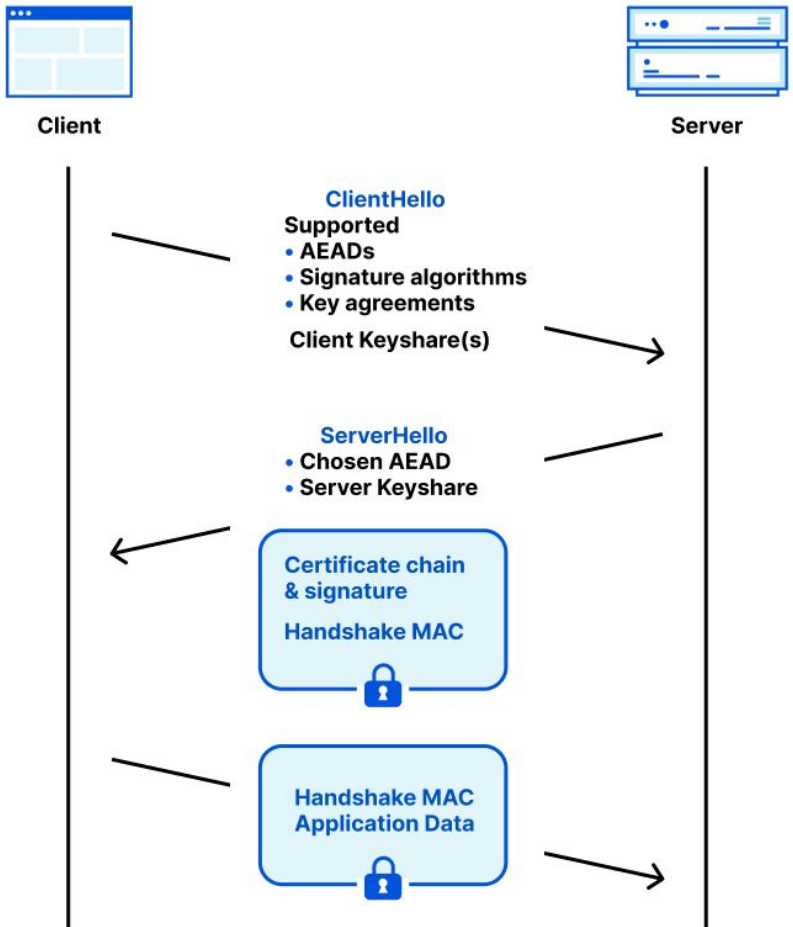
Backup slides

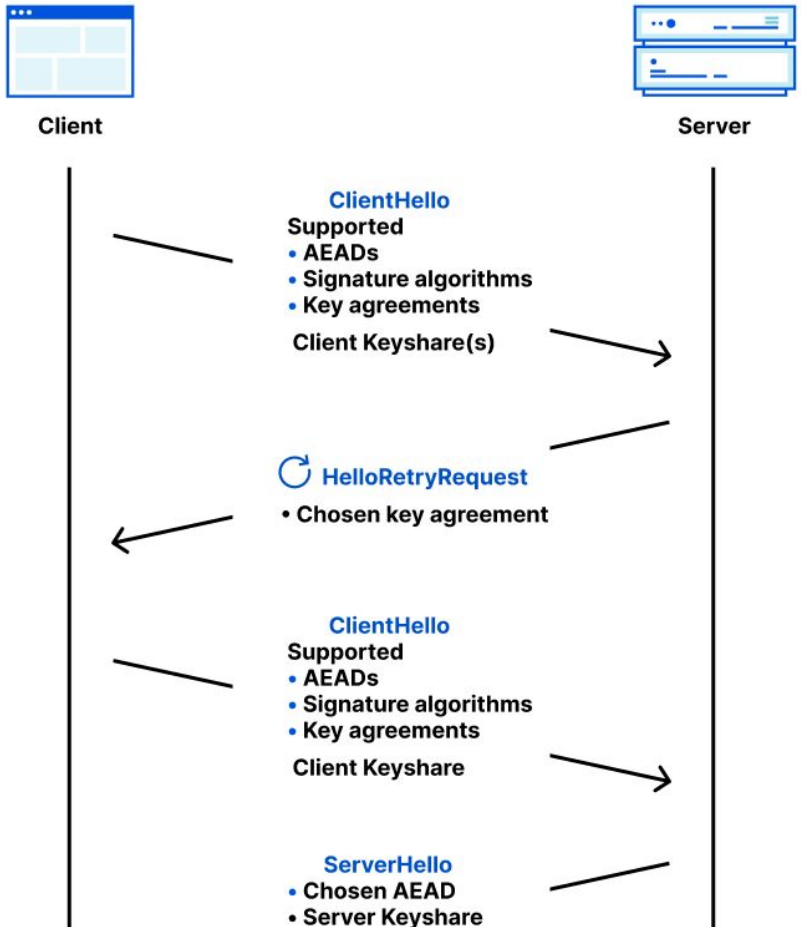
```c
static inline int64_t
fpr_rint(fpr x)
{
	/*
	 * We do not want to use llrint() since it might be not
	 * constant-time.
	 *
	 * Suppose that x >= 0. If x >= 2^52, then it is already an
	 * integer. Otherwise, if x < 2^52, then computing x+2^52 will
	 * yield a value that will be rounded to the nearest integer
	 * with exactly the right rules (round-to-nearest-even).
	 *
	 * In order to have constant-time processing, we must do the
	 * computation for both x >= 0 and x < 0 cases, and use a
	 * cast to an integer to access the sign and select the proper
	 * value. Such casts also allow us to find out if |x| < 2^52.
	 */
	int64_t sx, tx, rp, rn, m;
	uint32_t ub;

	sx = (int64_t)(x.v - 1.0);
	tx = (int64_t)x.v;
	rp = (int64_t)(x.v + 4503599627370496.0) - 4503599627370496;
	rn = (int64_t)(x.v - 4503599627370496.0) + 4503599627370496;

	/*
	 * If tx >= 2^52 or tx < -2^52, then result is tx.
	 * Otherwise, if sx >= 0, then result is rp.
	 * Otherwise, result is rn. We use the fact that when x is
	 * close to 0 (|x| <= 0.25) then both rp and rn are correct;
	 * and if x is not close to 0, then trunc(x-1.0) yields the
	 * appropriate sign.
	 */

	/*
	 * Clamp rp to zero if tx < 0.
	 * Clamp rn to zero if tx >= 0.
	 */
	m = sx >> 63;
	rn &= m;
	rp &= ~m;

	/*
	 * Get the 12 upper bits of tx; if they are not all zeros or
	 * all ones, then tx >= 2^52 or tx < -2^52, and we clamp both
	 * rp and rn to zero. Otherwise, we clamp tx to zero.
	 */
	ub = (uint32_t)((uint64_t)tx >> 52);
	m = -(int64_t)((((ub + 1) & 0xFFF) - 2) >> 31);
	rp &= m;
	rn &= m;
	tx &= ~m;

	/*
	 * Only one of tx, rn or rp (at most) can be non-zero at this
	 * point.
	 */
	return tx | rn | rp;
}
```

This function from Falcon as submitted to round 3 is not constant-time on ARMv7 as claimed.

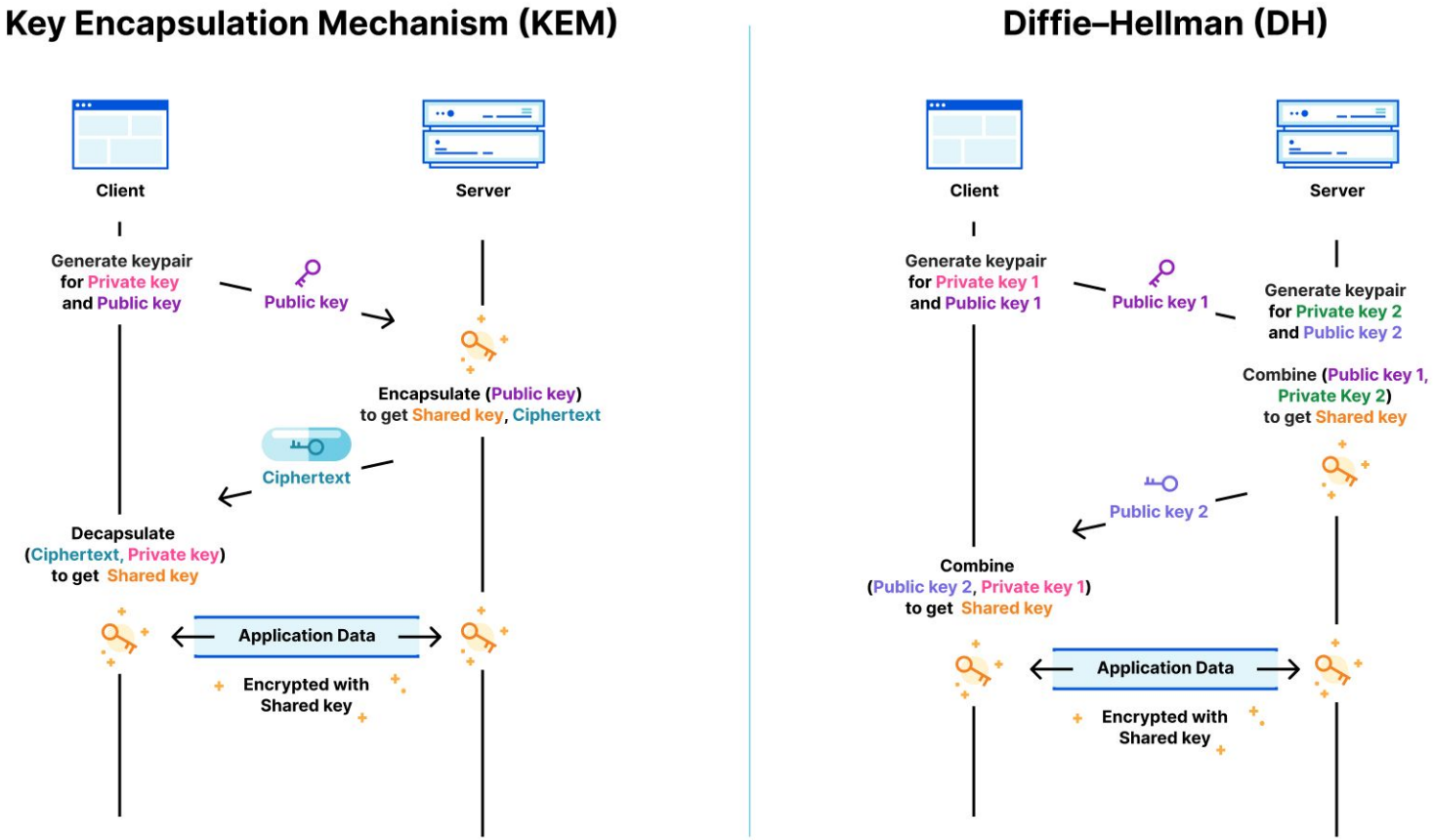
Can you spot the error?

TLS 1.3 handshake





KEM versus Diffie–Hellman



Post-Quantum

Cryptography Conference

Birth of the post-quantum Internet

Bas Westerbaan
Research Engineer at Cloudflare