SANDBOXAQ™

FIDO2 in the Quantum Realm

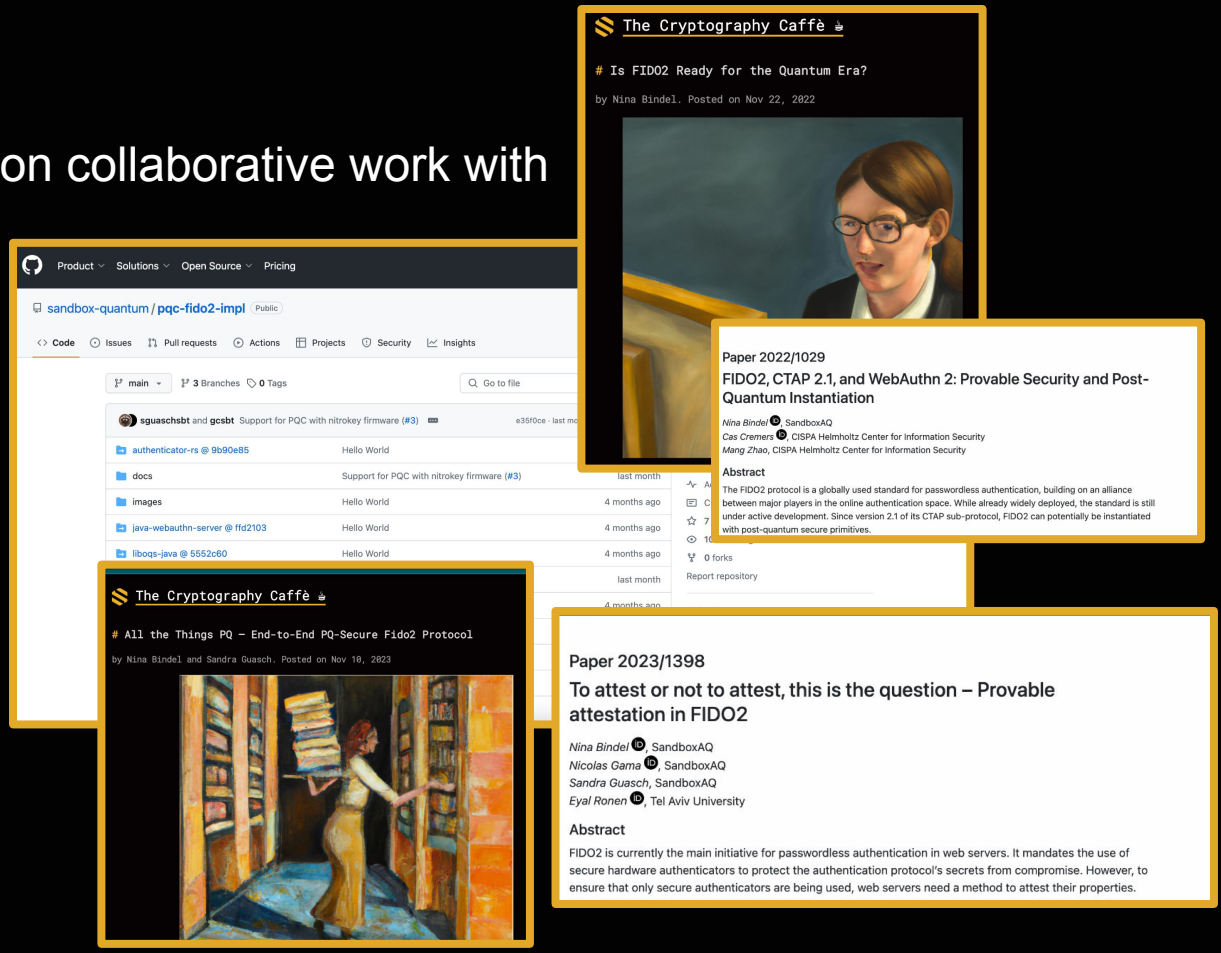
James Howe
Staff Research Scientist

Acknowledgment

This presentation is based on collaborative work with

- Nina Bindel
- Gabriel Campagna
- Cas Cremers
- Nicolas Gama
- Sandra Guasch
- Tarun Yadav
- Duc Nguyen
- Eyal Ronen
- Spencer Wilson
- Mang Zhao

All icons are from flaticon premium.



AGENDA

01 **Introduction**
PQC challenges to authentication systems

02 **Use case: FIDO2**
Introduction to the FIDO2 protocol

03 **PQ-readiness of FIDO2**
Is FIDO2 ready for PQC?

04 **Practical limitations and alternatives**
- Storage
- Runtime
- Potential adoption timeline

01

Introduction

PQC challenges to authentication systems

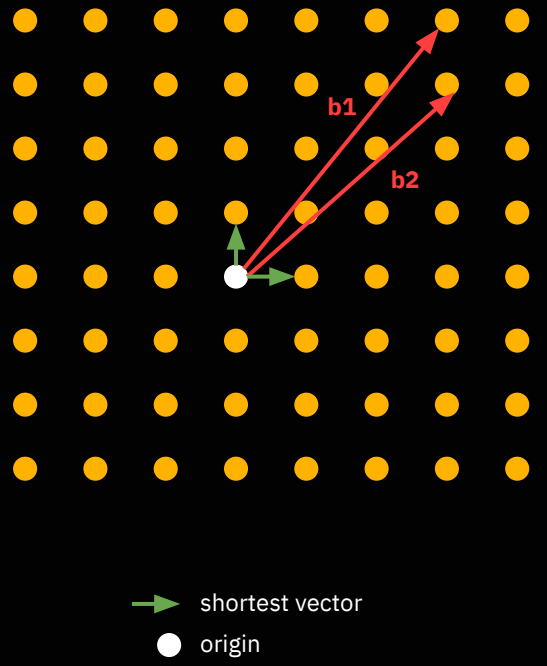
What are the PQC standards we have?

FIPS 203 (**ML-KEM**) (aka Kyber) is the *only* KEM and **FIPS 204** (**ML-DSA**) (aka Dilithium) is the *primary* signature.

Both are *lattice-based*, a problem akin to:

- Given ***A*** and ***b***, where ***b*** = ***A*********s*** + ***e*** *mod q*, find ***s***.
- Equivalent to finding short vector in a lattice.

They also significantly overlap codebases.

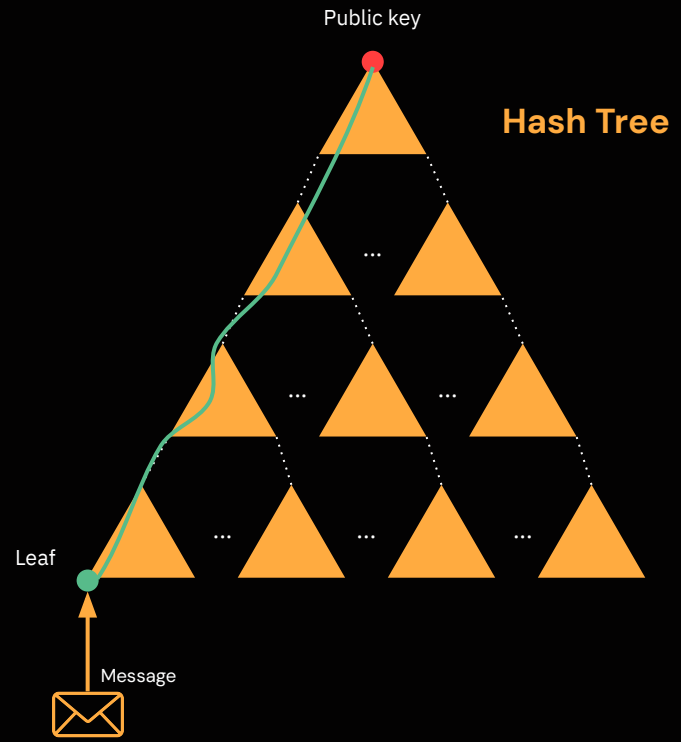


What are the PQC standards we have?

We have two other PQ signatures:

- **FIPS 205** (**SLH-DSA**) (aka SPHINCS+), a hash-based scheme, provides diversity.
- Signature scheme based on hardness of cryptographic hash functions.

- **FIPS ???** (**FN-DSA**) (aka Falcon), an upcoming lattice-based signature scheme.



What are the PQC standards we may have

We have 3 KEMs remaining in Round 4:

- BIKE, HQC, & Classic McEliece
- All based on hardness problems in coding theory.
- NIST will standardise BIKE or HQC.

NIST PQC on-ramp for more signatures:

- 6 code-based, 1 isogeny, 5 ‘misc.’
- 7 more lattice-based, 4 ‘symmetric’-based
- 7 based on MPC, 10 multivariate-based

(Some) challenges of PQC to existing systems

Longer keys, signatures, ciphertexts, certificates…

Migration to new algorithms requires cryptographic agility

How do we transition? Hybrid vs pure PQC?

Interconnected systems, dependencies

Remote / long-lived systems

(Also some) challenges of PQ authentication

Reliance on hardware

Low capacity devices (hardware tokens, smartcards…)

End-user distribution

We are first focusing on migrating encryption systems due to SNDL attacks

01½

PQC performance in the real world

PQC Performances

PQ vs Classical compare for key agreement in TLS

Algorithm	PQ?	Keyshare size (in bytes)		Operations per second	
		Client	Server	Client	Server
X25519	👎	32	32	19,000	19,000
Kyber-512	👍	800	768	45,000	70,000
Kyber-768	👍	1,184	1,088	29,000	45,000
Kyber-1024	👍	1,568	1,568	20,000	30,000

Results from 'The state of the post-quantum Internet' by Bas Westerbaan, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

PQ vs Classical compare for signatures in TLS

Algorithm	PQ?	Sizes (in bytes)		Relative CPU Runtime	
		Public-Key	Signature	Signing	Verifying
Ed25519	👎	32	64	1	1
RSA-2048	👎	256	256	70	0.3
Dilithium-2	👍	1,312	2,420	4.8	0.5
Falcon-512	👍	897	666	8	0.5
SPHINCS+s	👍	32	7,856	8,000	2.8
SPHINCS+f	👍	32	17,088	550	7

Results from 'The state of the post-quantum Internet' by Bas Westerbaan, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

How does PQ vs Classical compare for signatures?

Algorithm	PQ?	Sizes (in bytes)		Relative CPU Runtime	
		Public-Key	Signature	Signing	Verifying
Ed25519	👎	32	64	1	1
RSA-2048	👎	256	256	70	0.3
Dilithium-2	👍	1,312	2,420	4.8	0.5
Falcon-512	👍	897	666	8	0.5
SPHINCS+s	👍	32	7,856	8,000	2.8
SPHINCS+f	👍	32	17,088	550	7

Results from 'The state of the post-quantum Internet' by Bas Westerbaan, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

How does PQ vs Classical compare for signatures?

Algorithm	PQ?	Sizes (in bytes)		Relative CPU Runtime	
		Public-Key	Signature	Signing	Verifying
Ed25519	👎	32	64	1	1
RSA-2048	👎	256	256	70	0.3
Dilithium-2	👍	1,312	2,420	4.8	0.5
Falcon-512	👍	897	666	8	0.5
SPHINCS+s	👍	32	7,856	8,000	2.8
SPHINCS+f	👍	32	17,088	550	7

Results from 'The state of the post-quantum Internet' by Bas Wes, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

How does PQ vs Classical compare for signatures?

Algorithm	PQ?	Sizes (in bytes)		Relative CPU Runtime	
		Public-Key	Signature	Signing	Verifying
Ed25519	👎	32	64	1	1
RSA-2048	👎	256	256	70	0.3
Dilithium-2	👍	1,312	2,420	4.8	0.5
Falcon-512	👍	897	666	8	0.5
SPHINCS+s	👍	32	7,856	8,000	2.8
SPHINCS+f	👍	32	17,088	550	7

Results from 'The state of the post-quantum Internet' by Bas Wes, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

How does PQ vs Classical compare for signatures?

Algorithm	PQ?	Sizes (in bytes)		Relative CPU Runtime	
		Public-Key	Signature	Signing	Verifying
Ed25519	👎	32	64	1	1
RSA-2048	👎	256	256	70	0.3
Dilithium-2	👍	1,312	2,420	4.8	0.5
Falcon-512	👍	897	666	8	0.5
SPHINCS+s	👍	32	7,856	8,000	2.8
SPHINCS+f	👍	32	17,088	550	7

Results from 'The state of the post-quantum Internet' by Bas Wes, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

PQ vs Classical compare for signatures in TLS

Algorithm	PQ?	Sizes (in bytes)		Relative CPU Runtime	
		Public-Key	Signature	Signing	Verifying
Ed25519	👎	32	64	1	1
RSA-2048	👎	256	256	70	0.3
Dilithium-2	👍	1,312	2,420	4.8	0.5
Falcon-512	👍	897	666	8	0.5
SPHINCS+s	👍	32	**7,856**	**8,000**	2.8
SPHINCS+f	👍	32	**17,088**	**550**	7

Results from 'The state of the post-quantum Internet' by Bas Westerbaan, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

PQ vs Classical compare for signatures in TLS

Algorithm	PQ?	Sizes (in bytes)		Relative CPU Runtime	
		Public-Key	Signature	Signing	Verifying
Ed25519	👎	32	64	1	1
RSA-2048	👎	256	256	70	0.3
Dilithium-2	👍	**1,312**	**2,420**	4.8	0.5
Falcon-512	👍	897	666	8	0.5
SPHINCS+s	👍	32	7,856	8,000	2.8
SPHINCS+f	👍	32	17,088	550	7

Results from 'The state of the post-quantum Internet' by Bas Westerbaan, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

PQ vs Classical compare for signatures in TLS

Algorithm	PQ?	Sizes (in bytes)		Relative CPU Runtime	
		Public-Key	Signature	Signing	Verifying
Ed25519	👎	32	64	1	1
RSA-2048	👎	256	256	**70**	0.3
Dilithium-2	👍	1,312	2,420	4.8	0.5
Falcon-512	👍	**897**	**666**	**8**	0.5
SPHINCS+s	👍	32	7,856	8,000	2.8
SPHINCS+f	👍	32	17,088	550	**7**

Results from 'The state of the post-quantum Internet' by Bas Westerbaan, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

PQ vs Classical compare for signatures in TLS

Algorithm	PQ?	Sizes (in bytes)		Relative CPU Runtime	
		Public-Key	Signature	Signing	Verifying
Ed25519	👎	32	64	1	1
RSA-2048	👎	256	256	70	0.3
Dilithium-2	👍	1,312	2,420	4.8	0.5
Falcon-512	👍	897	666	8	0.5
SPHINCS+s	👍	32	7,856	8,000	2.8
SPHINCS+f	👍	32	17,088	550	7

Results from 'The state of the post-quantum Internet' by Bas Westerbaan, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

Takeaways:

- 👍 PQ KEM performance is acceptable
- PQ signature performance is not really adequate

PQC Performances

Takeaways:

👍 PQ KEM performance is acceptable
- PQ signature performance is not really adequate

“In the short term, we expect early adoption of post-quantum authentication across the Internet around 2026, but few will turn it on by default.”

Quotes from ‘The state of the post-quantum Internet’ by Bas Westerbaan, available at https://blog.cloudflare.com/pq-2024.

PQC Performances

Takeaways:

👍 PQ KEM performance is acceptable
- PQ signature performance is not really adequate

“In the short term, we expect early adoption of post-quantum authentication across the Internet around 2026, but few will turn it on by default.”

“Unless we can get performance much closer to today’s authentication, we expect the vast majority to keep post-quantum authentication disabled, unless motivated by regulation.”

Quotes from ‘The state of the post-quantum Internet’ by Bas Westerbaan, available at https://blog.cloudflare.com/pq-2024.

02

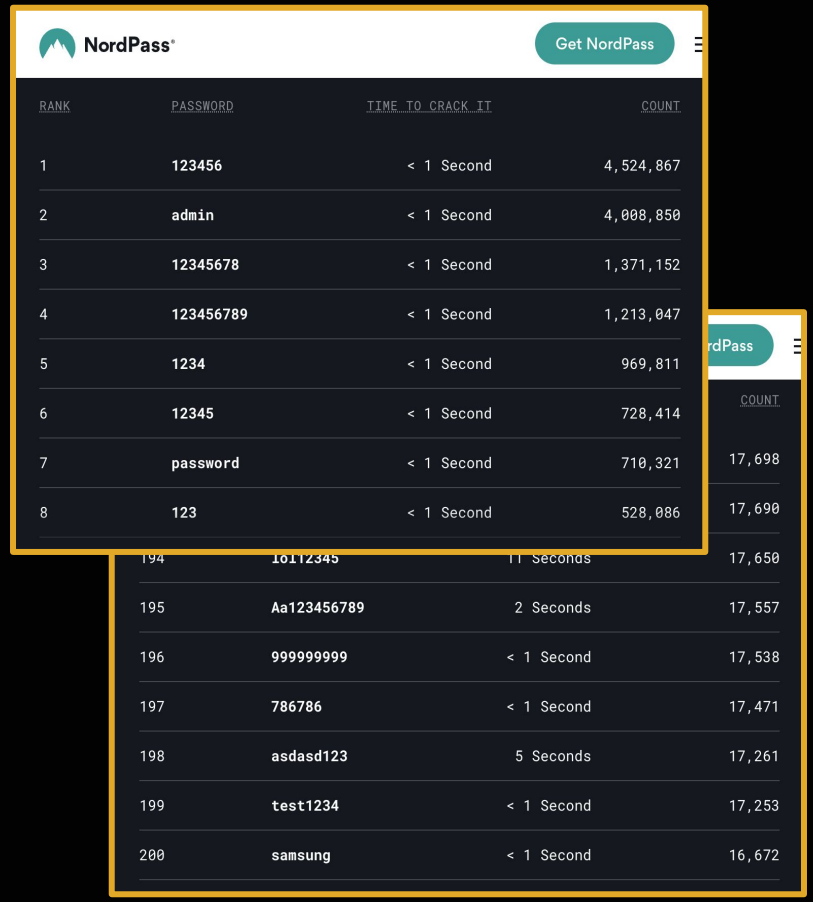
Use case: FIDO2

Introduction to the FIDO2 protocol

Passwords

Nearly every digital service, from email to banking, requires a password for access.

But often they are the first and only line of defense.



Passwords

Nearly every digital service, from email to banking, requires a password for access.

But often they are the first and only line of defense.

>80% of confirmed breaches relate to stolen, weak, or reused passwords[1].

[1]https://us.norton.com/blog/privacy/password-statistics

Password managers

Nearly 2/3 of internet users keep track of their passwords by memory or with handwritten notes[1].

Almost 1/4 people rely on a document on their computer to manage all of their passwords[1].

Less than 40% of organizations require the use of a password manager[1].

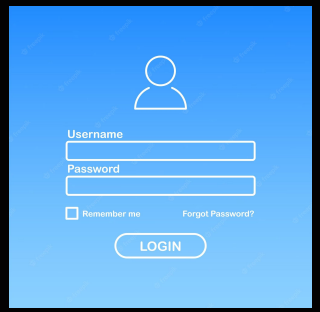
[1]https://us.norton.com/blog/privacy/password-statistics

Problem statement

Classic authentication solutions for web are not working.

Passwords

- Hard to remember / not complex enough
- Vulnerable to phishing attacks
- Synchronisation across devices can be challenging (pwd managers)



Multi-factor authentication / OTPs

- Low usability
- Still vulnerable to phishing
- OTP channels → extra attacks (e.g. malicious SMS)

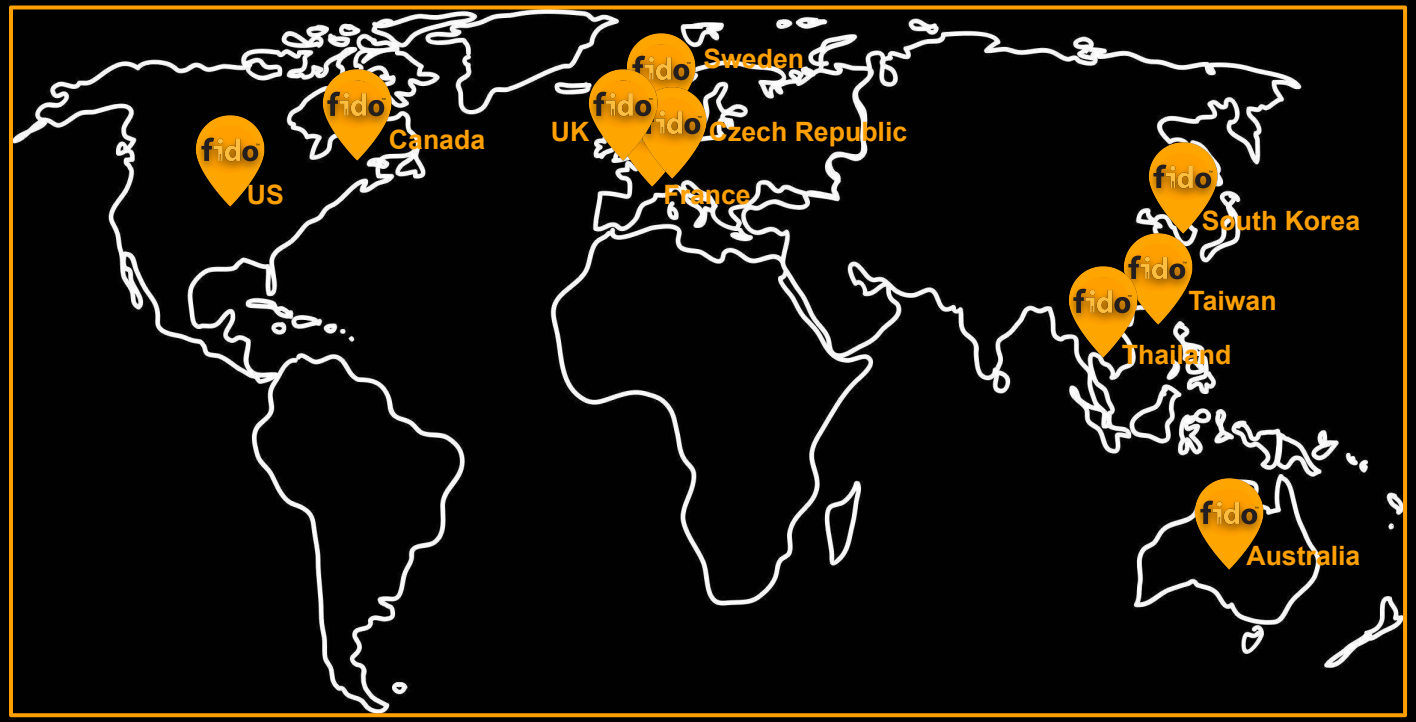
Comprised by more than **40 key companies**, including Amazon, Apple, Google, Intel, Microsoft, RSA, VISA, and Yubico

Defined de facto standard for passwordless authentication: **FIDO2 protocol**

Who is participating?

FIDO Alliance Board members

Government deployments and recommendations (2021)



US:

- General Services Administration
- CISA security advisory
- NIST guidelines
- NIST + NCCOE - best practices
- OMB - Federal PKI updates
- DEA - secure access to drug prescriptions

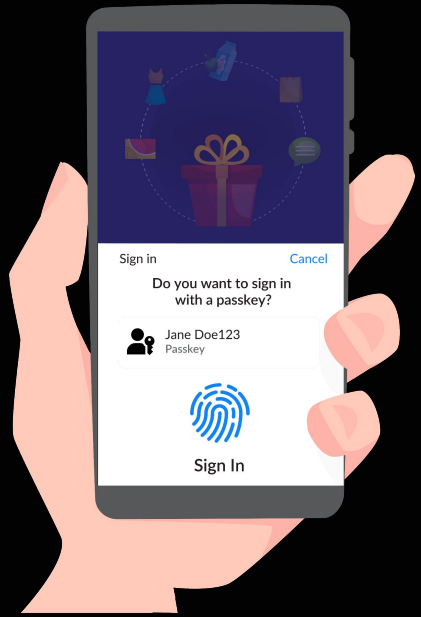
UK:

- DCMS - Digital identity policy and strategy
- Government Digital Service
- NHS login

What is FIDO?

Advantages

- No need to remember passwords
- Easy to use
- Resistant to phishing attacks
- Widely adopted: FIDO Alliance / W3C standards
  - Supported by all major browsers and platforms
  - Wide range of industry partners
- Constant improvements (e.g., Passkeys)



Google Adds Passkey Support to Chrome for Windows, macOS and Android

Dec 12, 2022

Companies are increasingly ditching passwords for passkeys

YubiKeys, passkeys and the future of modern authentication

Christopher Harrell
March 31, 2022 • 10 minute read

What is Apple Passkey, and how will it help you go passwordless?

Ivan Mehta @indianidle / 5:00 PM GMT+2 • September 12, 2022

Momentum for FIDO in Japan Grows as Major Companies Commit to Passwordless Sign-ins with Passkeys

A (very) brief history of FIDO authentication

2014

U2F

2nd factor authentication

A (very) brief history of FIDO authentication

2014

U2F

2nd factor authentication

2019

FIDO2 = CTAP (FIDO) + WebAuthn (W3C)

Security tokens are generate credentials which are registered and used to authenticate

A (very) brief history of FIDO authentication

2014

U2F

2nd factor authentication

2019

FIDO2 = CTAP (FIDO) + WebAuthn (W3C)

Security tokens are generate credentials which are registered and used to authenticate

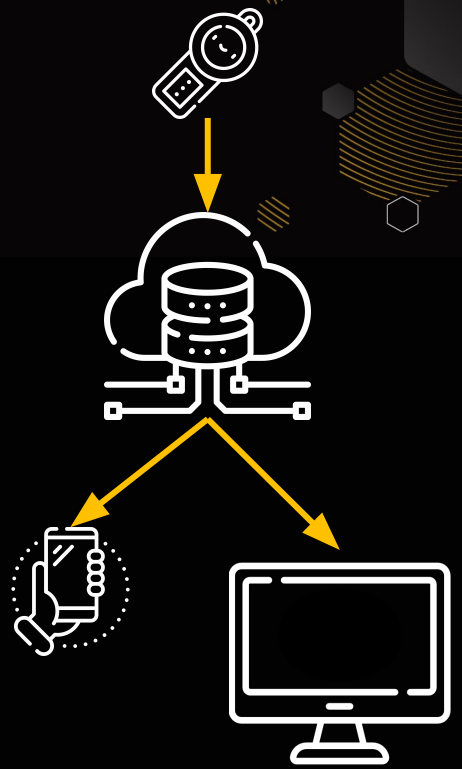
2022

Passkeys

Passkeys = FIDO2 with the option of synchronization of credentials such that synced devices can be used to authenticate

Passkeys

- Credential synchronisation among different devices
- Credentials are encrypted E2E
- Device-bound credentials can still be enforced for critical applications
- Attestation becomes crucial to understand how a credential is managed



A (very) brief history of FIDO authentication

2014

U2F

2nd factor authentication

2019

FIDO2 = CTAP (FIDO) + WebAuthn (W3C)

Security tokens are generate credentials which are registered and used to authenticate

2022

Passkeys

Passkeys = FIDO2 with the option of synchronization of credentials such that synced devices can be used to authenticate

2024

White Paper: Addressing FIDO Alliance's 'Technologies in Post Quantum World'

Acknowledging the quantum threat and need to select suitable PQC algorithms and to prepare for smooth transition

02½ The FIDO2 Protocol Flow

Basic FIDO2 operation flow

Basic FIDO2 operation flow

User

Authenticator

Client

Relying Party

User verification, e.g. provision of PIN

- Generate credential key pair

Registration of authenticators credential public key at RP

- Store credential public key

Basic FIDO2 operation flow

User

Authenticator

Client

Relying Party

User verification, e.g. provision of PIN

- Generate credential key pair

3 Registration of authenticators credential public key at RP

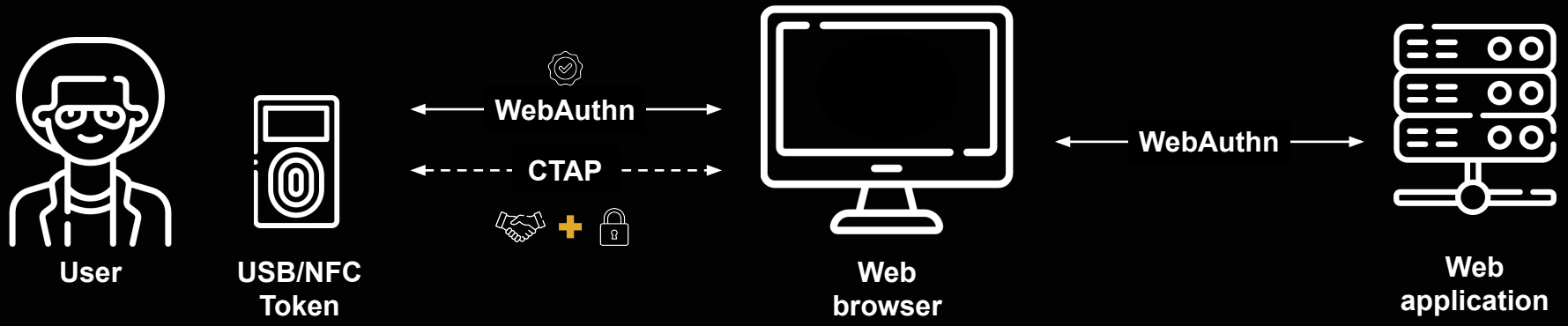
- Store credential public key

- Generate signature using credential secret key

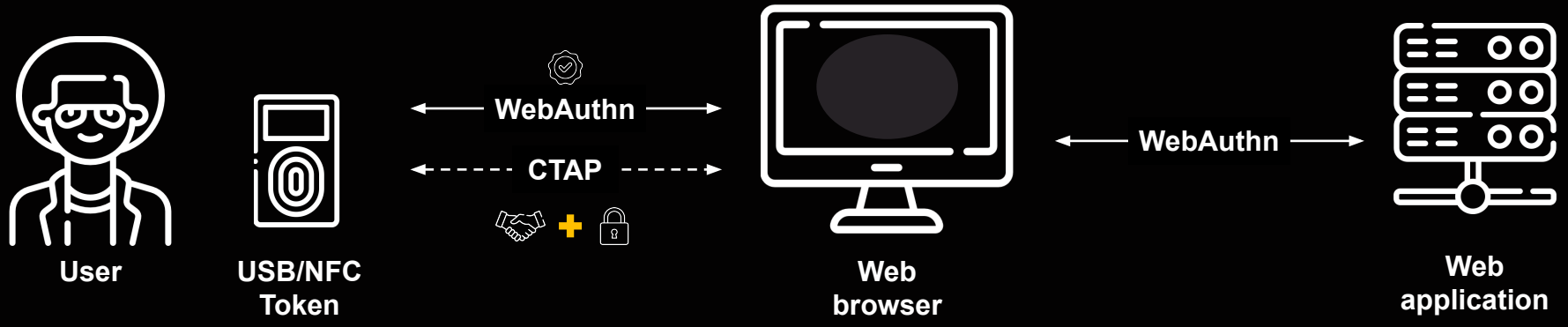
4 Authentication using registered credential public key

- Verify signature using public credential key

FIDO2 protocol



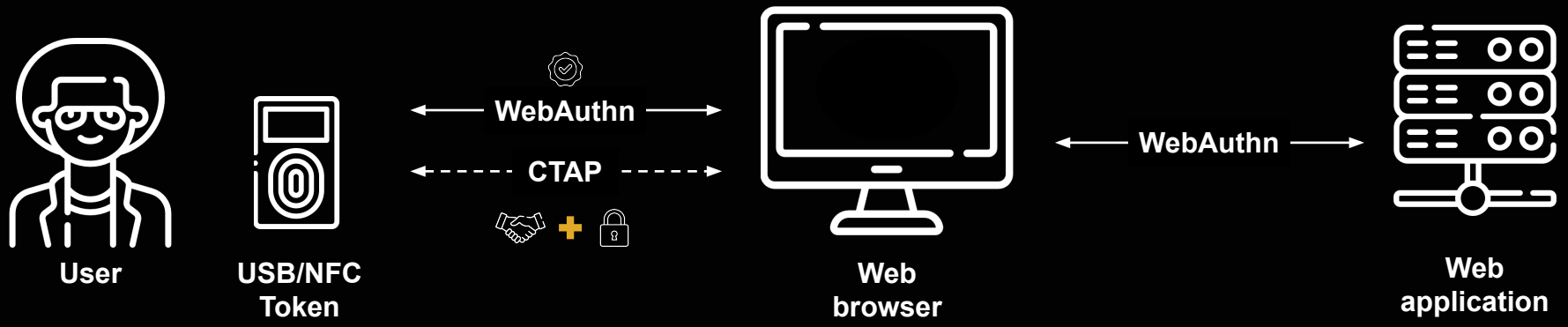
FIDO2 = WebAuthn + CTAP



WebAuthn
Sub-protocol to let the user authenticate into the web service with the hardware token

CTAP (Client To Authenticator Protocol)
Sub-protocol to ensure only a browser trusted by the user can communicate directly with the token.

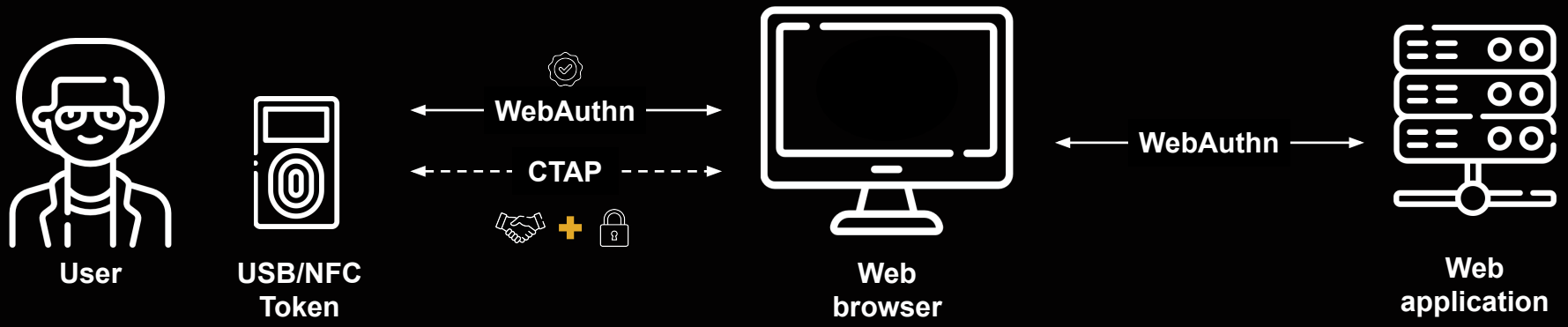
FIDO2 = WebAuthn + CTAP



WebAuthn
Sub-protocol to let the user authenticate into the web service with the hardware token

CTAP (Client To Authenticator Protocol)
Sub-protocol to ensure only a browser trusted by the user can communicate directly with the token.

Registration



- key exchange + symm. encryption
- gesture
- *(sk,vk)* generate **assertion keys**
- *att* generate attestation signature

chall, info ←

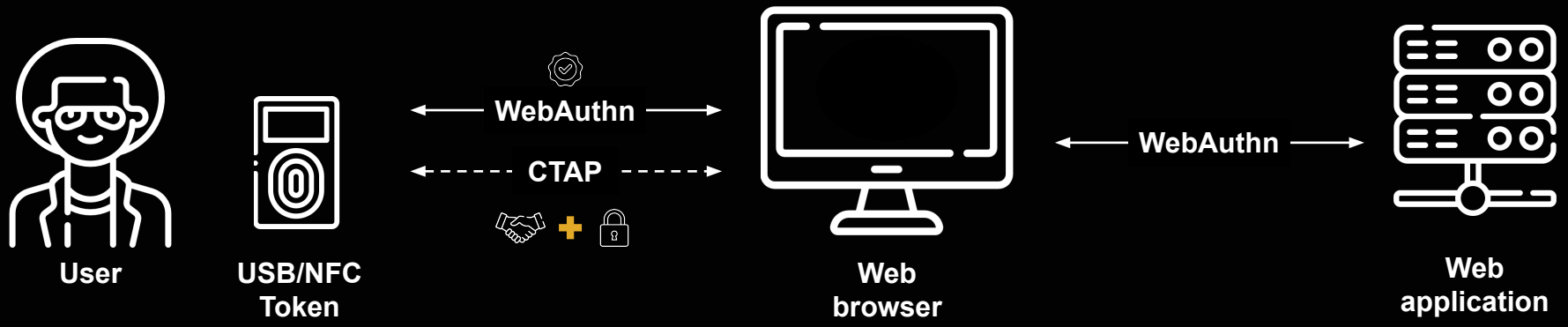
chall, info ←

- *chall* randomly chosen
- *info* session info

vk, att, more info →

- verify *info, att*
- save *vk*

Registration



- key exchange + symm. encryption
- gesture
- *(sk,vk)* generate **assertion keys**
- *att* generate attestation signature

chall, info ←

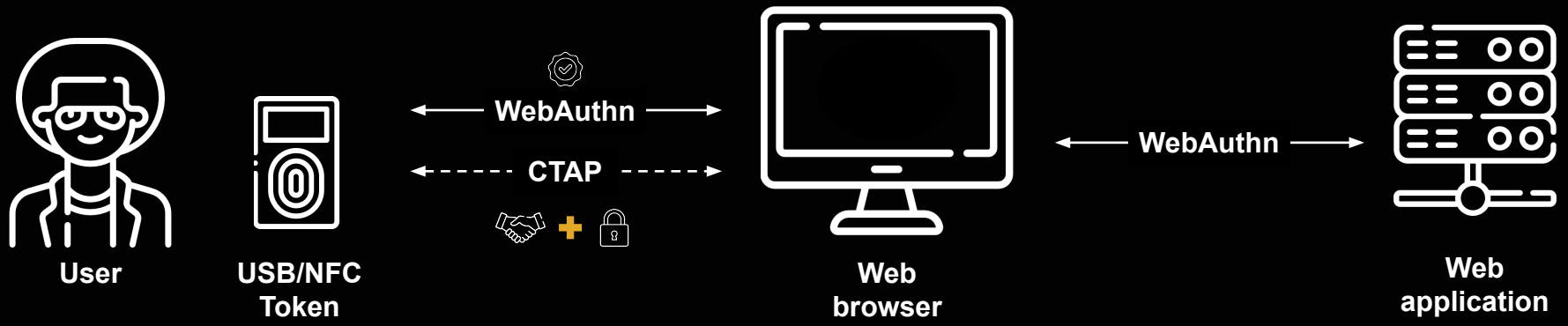
chall, info ←

- *chall* randomly chosen
- *info* session info

vk, att, more info →

- verify *info, att*
- save *vk*

Registration



- key exchange + symm. encryption
- gesture
- *(sk,vk)* generate **assertion keys**
- *att* generate attestation signature

chall, info ←

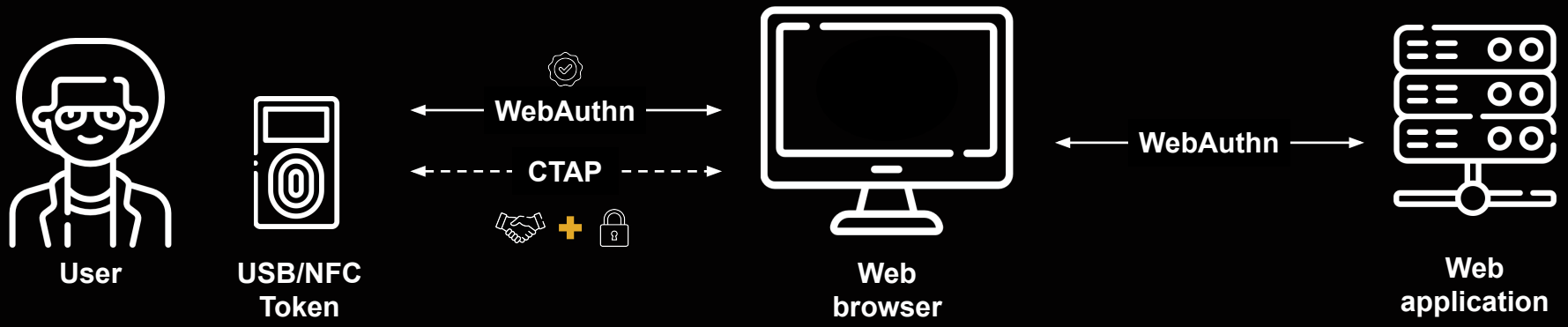
chall, info ←

- *chall* randomly chosen
- *info* session info

vk, att, more info →

- verify *info, att*
- save *vk*

Registration



- key exchange + symm. encryption
- gesture
- *(sk,vk)* generate **assertion keys**
- *att* generate attestation signature

chall, info ←

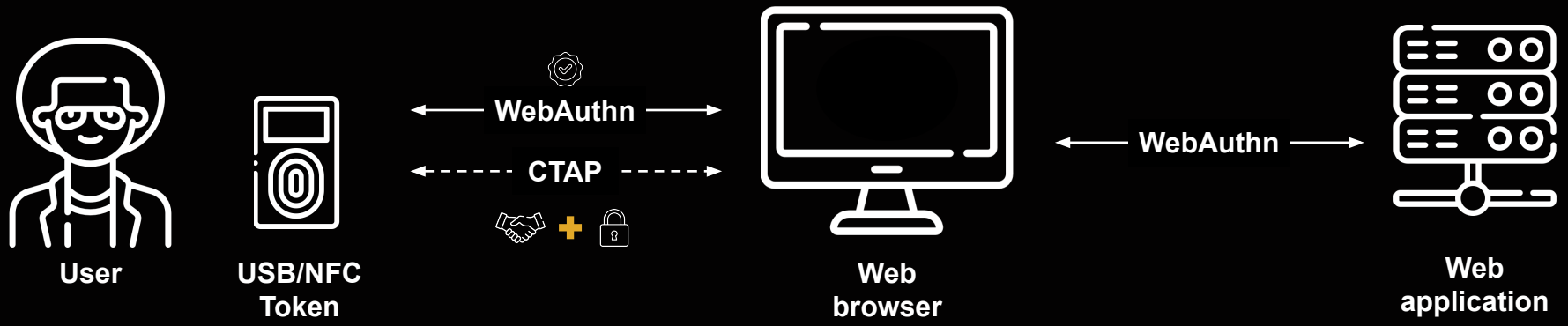
chall, info ←

- *chall* randomly chosen
- *info* session info

vk, att, more info →

- verify *info, att*
- save *vk*

Authentication



- key exchange + symm. encryption
- gesture
- ~~*(sk,vk)* generate **assertion keys**~~
- *sig* generate assertion signature

chall, info ←

chall, info ←

- *chall* randomly chosen
- *info* session info

vk, sig, more info →

- verify *info, sig*
- ~~save *vk*~~

Post-Quantum FIDO2

	WebAuthn	CTAP
PQ readiness	**Yes** if signature scheme is PQ secure	**Yes** if DH-based CTAP subroutine is instantiated with a (PQ) KEM
PQ instantiation	• Use signature negotiation in WebAuthn to include PQ/hybrid signature algorithms. • Use PQ signature.	• *Protocol* negotiation in CTAP 2.1 includes PQ/hybrid KEM. • Use PQ KEM. • Increase output length hash.

Post-Quantum FIDO2

	WebAuthn	CTAP
PQ readiness	**Yes** if signature scheme is PQ secure	**Yes** if DH-based CTAP subroutine is instantiated with a (PQ) KEM
PQ instantiation	• Use signature negotiation in WebAuthn to include PQ/hybrid signature algorithms. • Use PQ signature.	• *Protocol* negotiation in CTAP 2.1 includes PQ/hybrid KEM. • Use PQ KEM. • Increase output length hash.
Backwards Compatibility	• Cryptographic negotiations between User and Web Service similar to TLS. • Ensures backwards compatibility between those supporting PQC and not.	

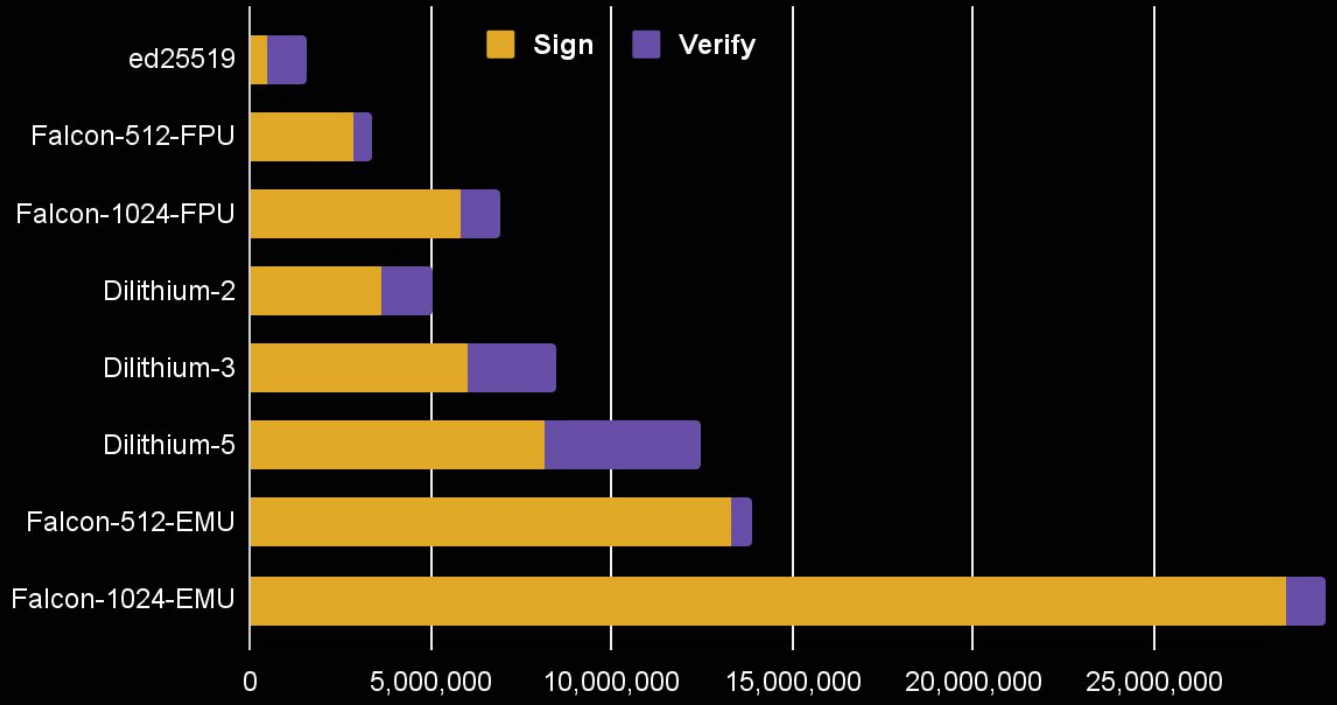
Post-Quantum FIDO2

	WebAuthn	CTAP
PQ readiness	**Yes** if **signature** scheme is PQ secure	**Yes** if DH-based CTAP subroutine is instantiated with a (PQ) KEM
PQ instantiation	• Use **signature** negotiation in WebAuthn to include PQ/hybrid **signature** algorithms. • Use PQ **signature**.	• *Protocol* negotiation in CTAP 2.1 includes PQ/hybrid KEM. • Use PQ KEM. • Increase output length hash.
Backwards Compatibility	• Cryptographic negotiations between User and Web Service similar to TLS. • Ensures backwards compatibility between those supporting PQC and not.	

02 3/4 Signature on Embedded Devices

Comparing Signature on ARM Cortex M7

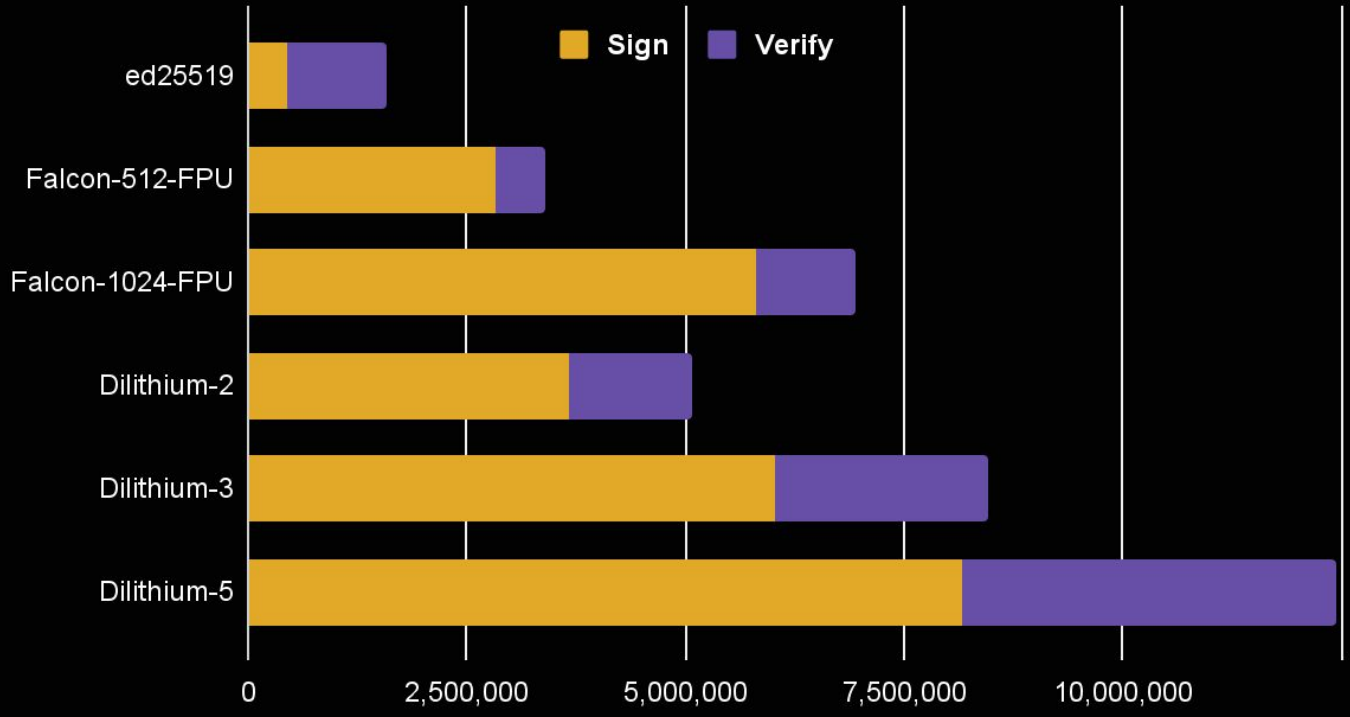
Results given in Clock Cycles



From 'Benchmarking and Analysing the NIST PQC Lattice-Based Signature Schemes Standards on the ARM Cortex M7' by **James Howe** and Bas Westerbaan, AFRICACRYPT 2023, https://eprint.iacr.org/2022/405.

Comparing Signature on ARM Cortex M7

Results given in Clock Cycles



From 'Benchmarking and Analysing the NIST PQC Lattice-Based Signature Schemes Standards on the ARM Cortex M7' by **James Howe** and Bas Westerbaan, AFRICACRYPT 2023, https://eprint.iacr.org/2022/405.

03 E2E PQ FIDO2 OSS

Implementation details

New open-source library!

Post-quantum secure, in particular using Dilithium and Kyber

End-to-end flow is PQ secure

Open source on
https://github.com/sandbox-quantum/pqc-fido2-impl

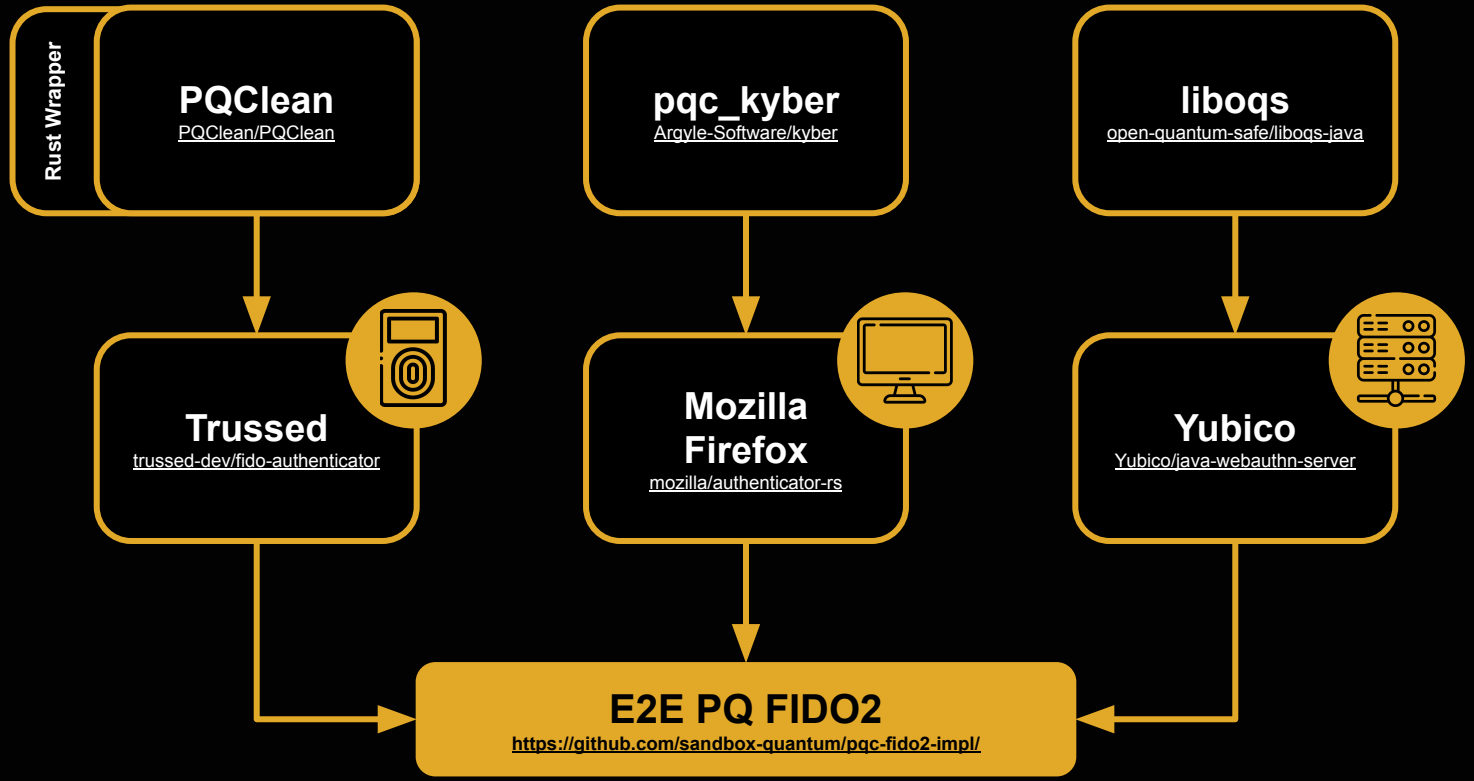
“Libraries are where it all begins” – Rita Dove

E2E PQ FIDO2

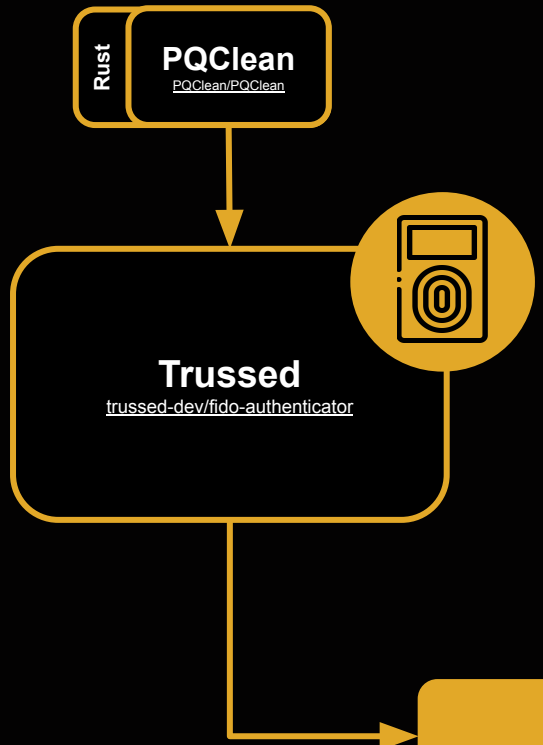
https://github.com/sandbox-quantum/pqc-fido2-impl/

“Libraries are where it all begins” – Rita Dove

“Libraries are where it all begins” – Rita Dove



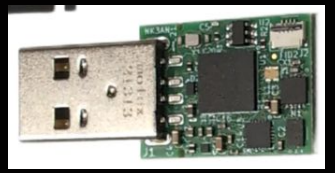
“Libraries are where it all begins” – Rita Dove



Tested on:

- LPCXpresso55S69 development board

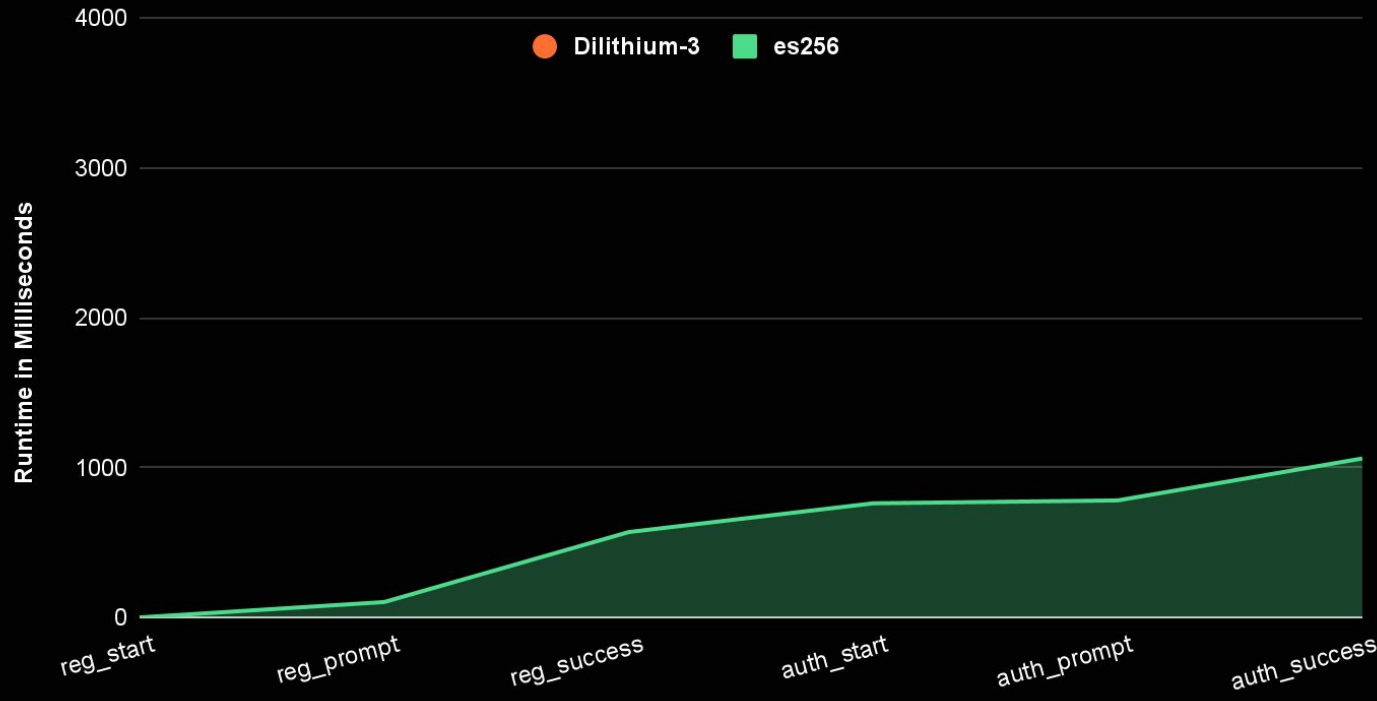
- NitroKey Hacker token with NXP LPC55S69JEV98



Both devices use ARM Cortex-M33 or similar

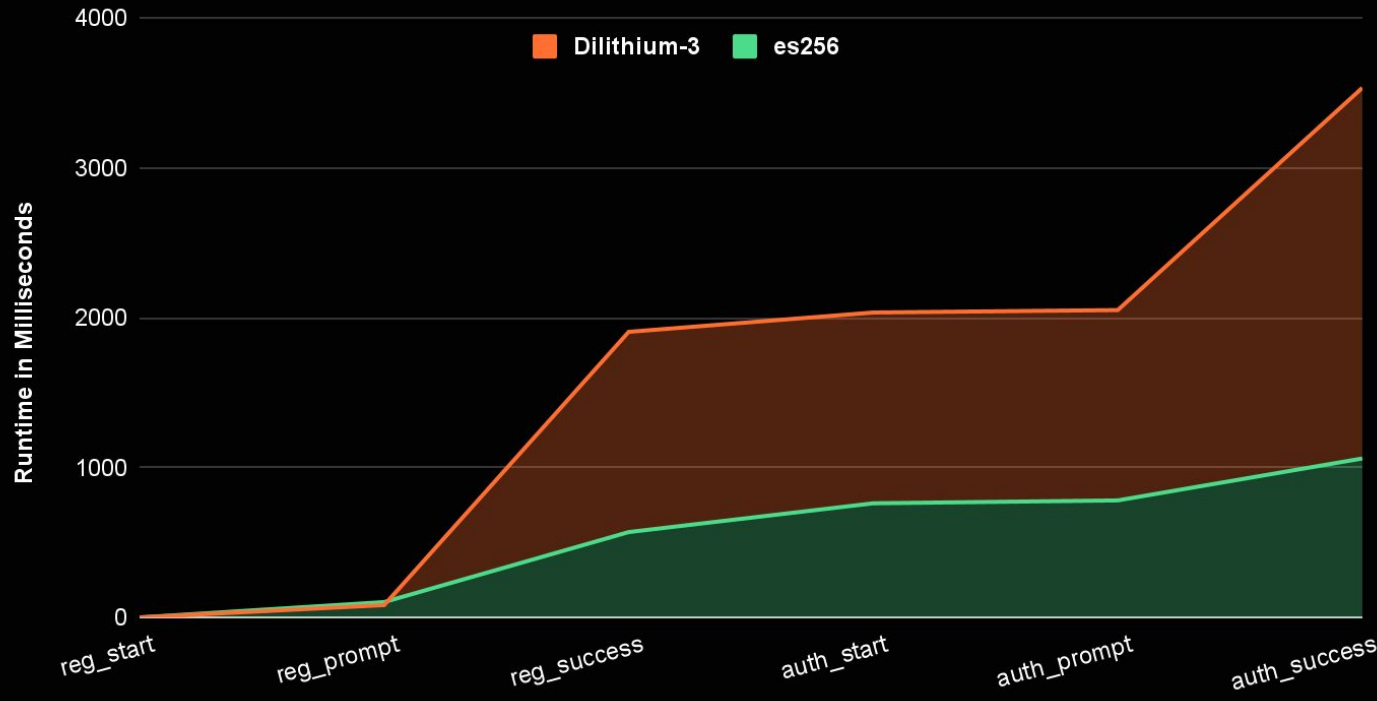
Performance of FIDO2

Comparing Elliptic Curve and Dilithium on ARM Cortex M33

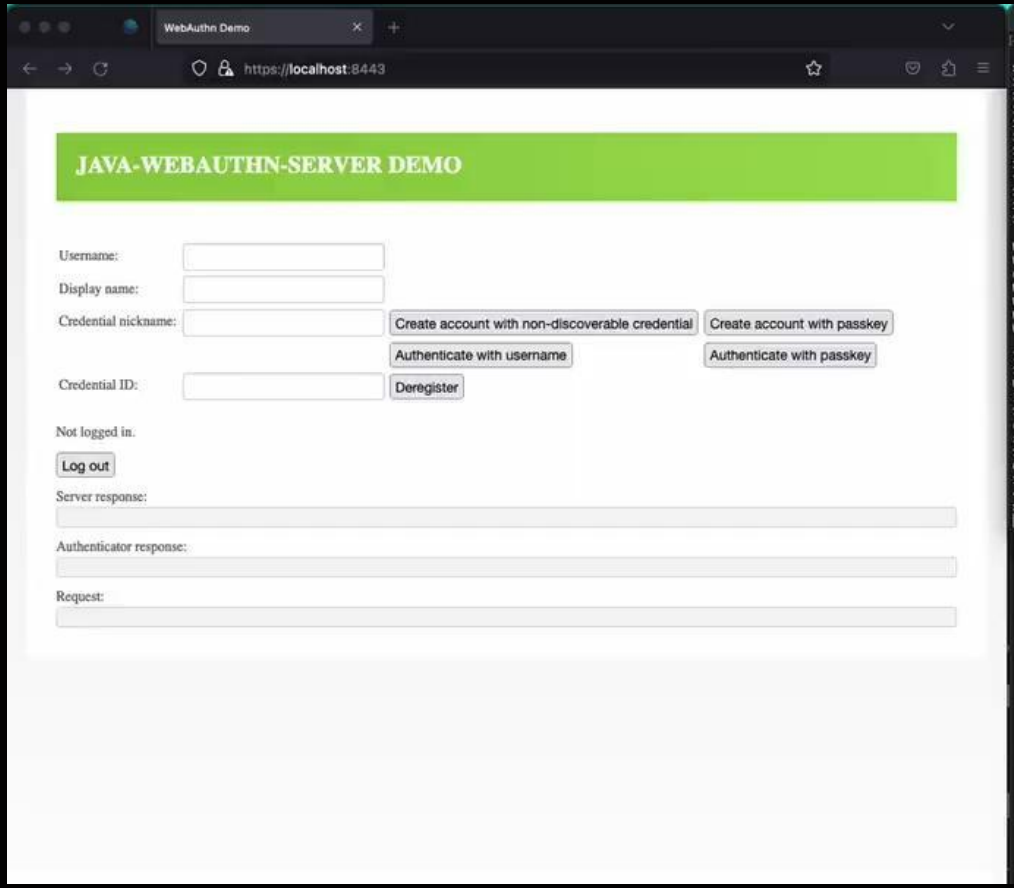


Performance of FIDO2

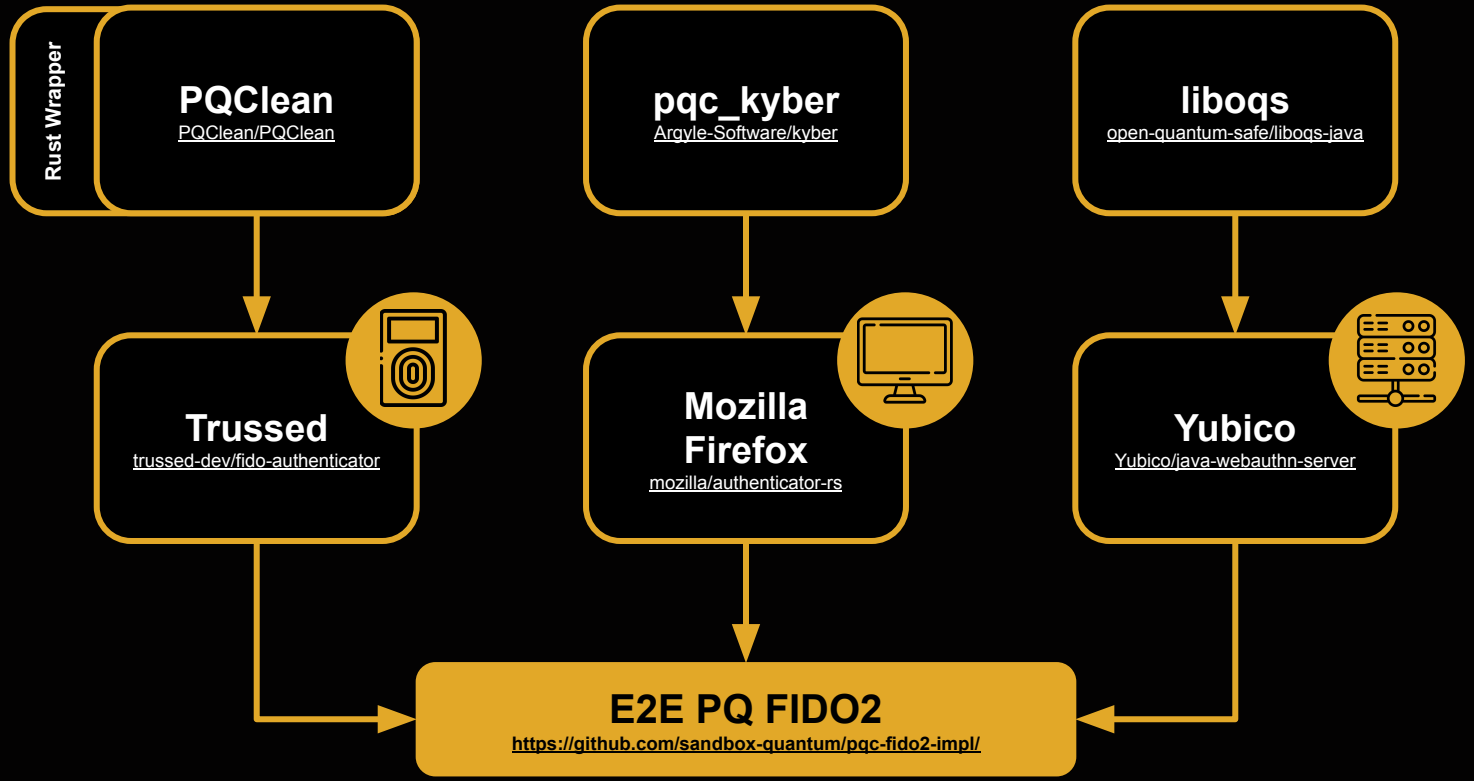
Comparing Elliptic Curve and Dilithium on ARM Cortex M33



PQ Extension of Yubico's Java-Webauthn-server



“Libraries are where it all begins” – Rita Dove



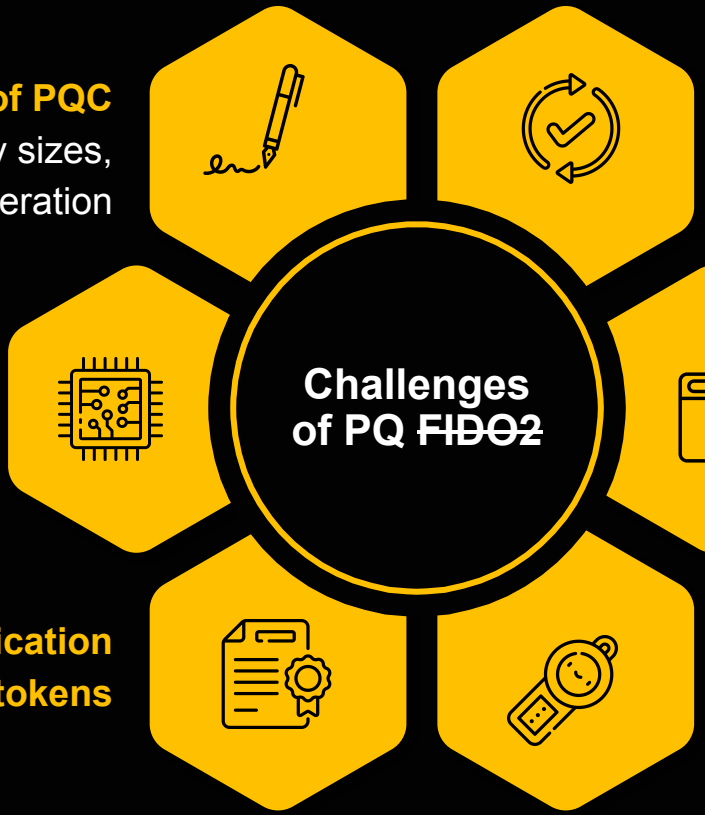
AGENDA

01 **FIDO2**
Introduction to the FIDO2 protocol

02 **PQ-readiness of FIDO2**
Analysis of WebAuthn and CTAP

03 **E2E PQ FIDO2 OSS**
Implementation details

04 **Challenges and future work**
Additional modes to be considered in the PQ migration



Efficiency of PQC
large key sizes, slower signature generation

Update ~~FIDO2~~ specs
choice algorithms, smooth transition, backwards compatibility with legacy hardware, etc

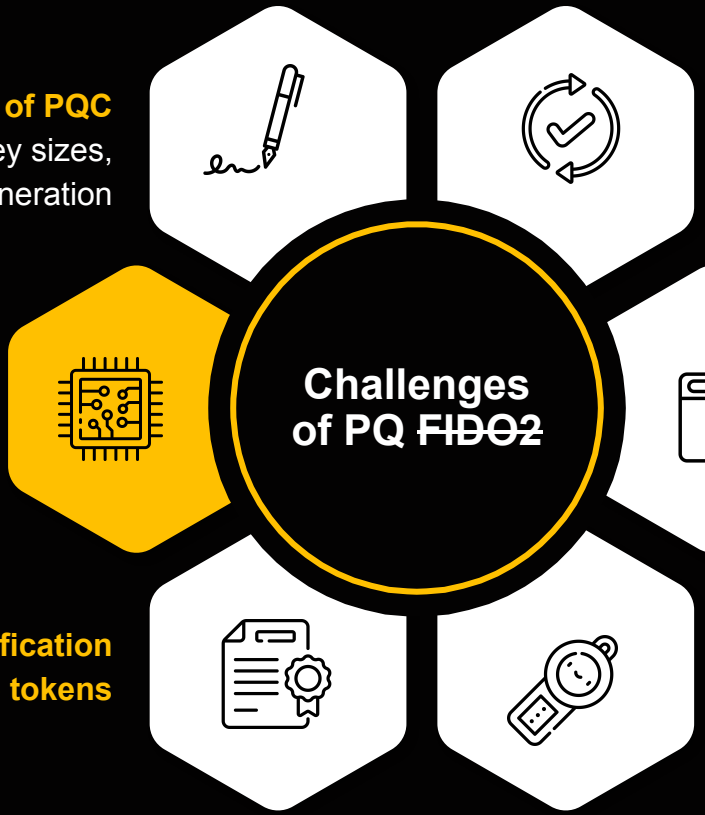
Change of hardware

Update CAs, browsers and web applications

Re-certification of hardware tokens

Distribute new hardware

SANDBOXAQ™



Efficiency of PQC
large key sizes, slower signature generation

Update ~~FIDO2~~ specs
choice algorithms, smooth transition, backwards compatibility with legacy hardware, etc

Change of hardware

Update CAs, browsers and web applications

Re-certification of hardware tokens

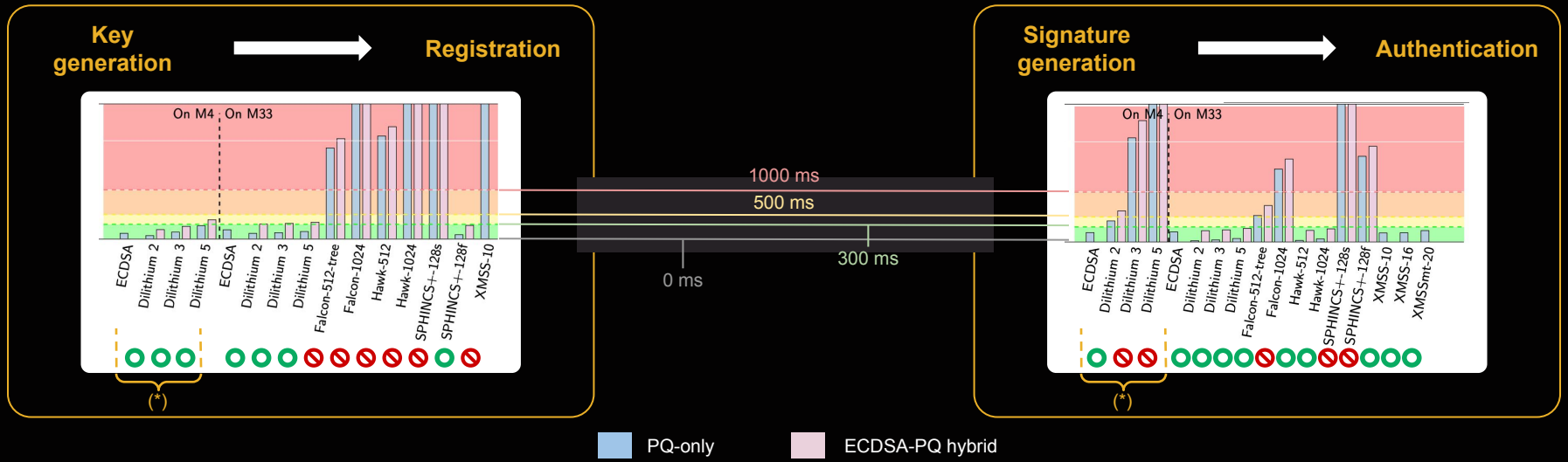
Distribute new hardware

SANDBOXAQ™

Runtime: Preliminary Experimental Results

Signature or Key Generation

- Runtime of PQ/Hybrid algorithms determines waiting time of user.
- In addition, operations of WebAuthn and CTAP, in particular PQ/Hybrid handshake, and communication time between token, client, and server.
- If waiting time is too long, user might behave such that phishing attacks become easier (e.g. pressing a button more than once).

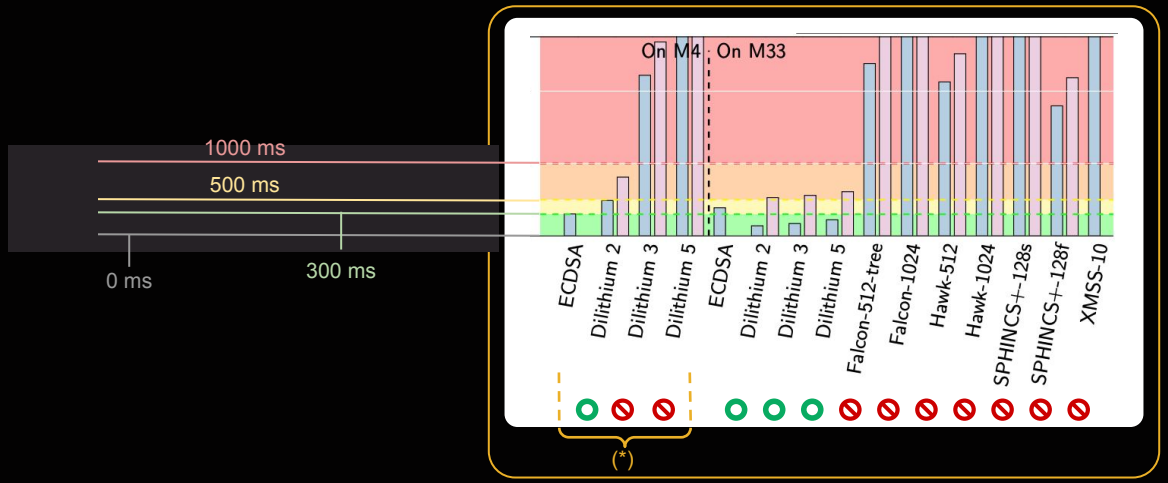


(*) Results from Ghinea, Kaczmarczyck, Pullman, Cretin, Kölbl, Misoczki, Picod, Invernizzi, and Bursztein. Hybrid Post-Quantum Signatures in Hardware Security Keys. IACR ePrint 2022/1225.

Runtime: Preliminary Experimental Results

Signature or Key Generation + Attestation

- If attestation modes are used, (credential) **key and** (attestation) **signature generation have to be performed during registration**.
- Good that Dilithium might fit, it would be better if there would be an alternative.
- Until then clever combination might give advantage.

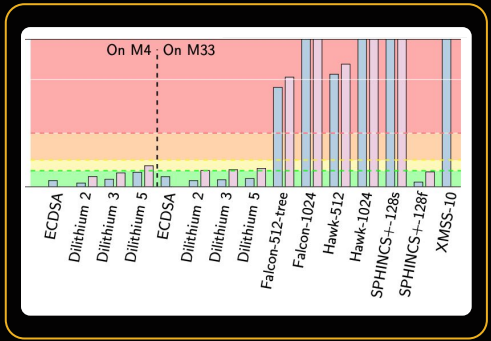


(*) Results from Ghinea, Kaczmarczyck, Pullman, Cretin, Kölbl, Misoczki, Picod, Invernizzi, and Bursztein. Hybrid Post-Quantum Signatures in Hardware Security Keys. IACR ePrint 2022/1225.

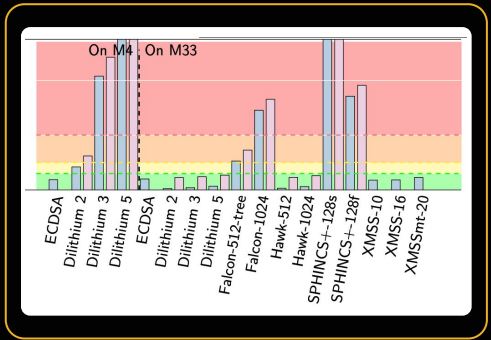
Different Signature Schemes for Attestation and Authentication

Example

Key generation



Signature generation



Attestation mode Basic

- Batch attestation
- No attestation key generation on hardware token

Attestation scheme: Hawk

Authentication scheme: Dilithium

- **Registration:**
  Dilithium key generation + Hawk signing **reasonably fast**
- **Authentication:**
  Dilithium signature generation **fast**

Advantage compared to Dilithium-only:

- Attestation signature only 546 B instead of 2420 B
- Particularly beneficial as during registration other information needs to be sent (e.g., public credential key)

Summary

- First steps in migrating FIDO2 protocol to use PQC taken
- Steps ahead to guide the decision for future specs:
  - benchmarking different PQ algorithms (including hybrid)
  - while considering different modes (attestation, key storage, credential synchronization, extensions)
- Get involved!

Summary

- First steps in migrating FIDO2 protocol to use PQC taken
- Steps ahead to guide the decision for future specs:
  - benchmarking different PQ algorithms (including hybrid)
  - while considering different modes (attestation, key storage, credential synchronization, extensions)
- Get involved!
- This demos steps for other use cases.

We are hiring

Check out sandboxaq.com/careers

Resources

Research papers

- FIDO2, CTAP 2.1, and WebAuthn 2: Provable Security and Post-Quantum Instantiation. Bindel, Cremers, Zhao. [ePrint]
- Attest or not to attest, this is the question – Provable attestation in FIDO2. Bindel et al. [ePrint]

Open source implementation

- E2E PQ FIDO2 OSS using Kyber and Dilithium

Blog posts

- Is FIDO2 Ready for the Quantum Era?
- End-to-End PQ-Secure FIDO2 Protocol

Thank you