



FIDO2 in the Quantum Realm

James Howe

Staff Research Scientist

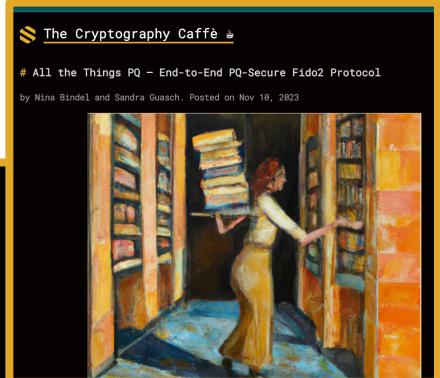
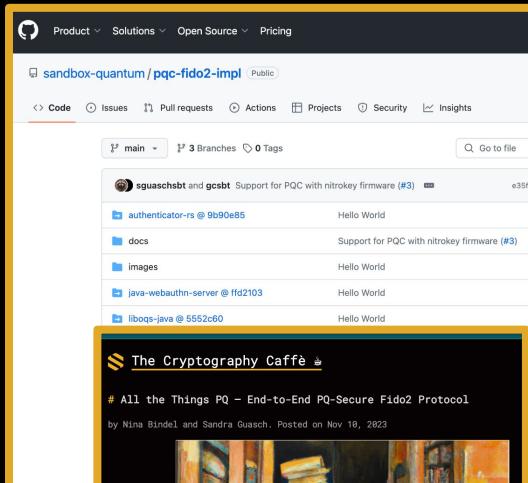


Acknowledgment

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.



The Cryptography Caffè

Is FIDO2 Ready for the Quantum Era?

by Nina Bindel. Posted on Nov 22, 2022

Paper 2022/1029
FIDO2, CTAP 2.1, and WebAuthn 2: Provable Security and Post-Quantum Instantiation

Nina Bindel, SandboxAQ
Cas Cremers, CISPA Helmholtz Center for Information Security
Mang Zhao, CISPA Helmholtz Center for Information Security

Abstract

The FIDO2 protocol is a globally used standard for passwordless authentication, building on an alliance between major players in the online authentication space. While already widely deployed, the standard is still under active development. Since version 2.1 of its CTAP sub-protocol, FIDO2 can potentially be instantiated with post-quantum secure primitives.

The Cryptography Caffè

All the Things PQ – End-to-End PQ-Secure Fido2 Protocol

by Nina Bindel and Sandra Guasch. Posted on Nov 18, 2023

Paper 2023/1398
To attest or not to attest, this is the question – Provable attestation in FIDO2

Nina Bindel, SandboxAQ
Nicolas Gama, SandboxAQ
Sandra Guasch, SandboxAQ
Eyal Ronen, Tel Aviv University

Abstract

FIDO2 is currently the main initiative for passwordless authentication in web servers. It mandates the use of secure hardware authenticators to protect the authentication protocol's secrets from compromise. However, to ensure that only secure authenticators are being used, web servers need a method to attest their properties.

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

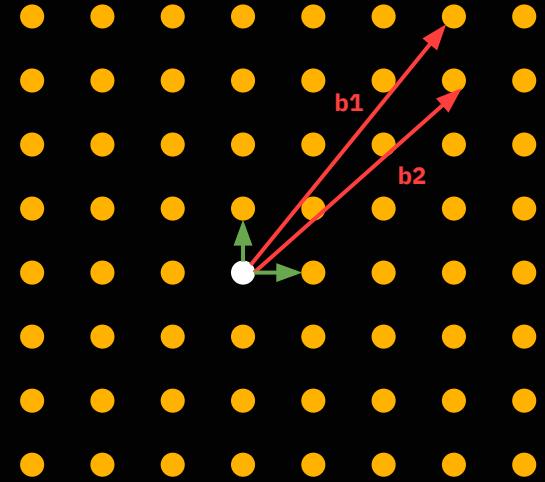
What are the PQC standards we have?

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 \mathbf{A} and \mathbf{b} , where $\mathbf{b} = \mathbf{A}^* \mathbf{s} + \mathbf{e} \text{ mod } q$, find \mathbf{s} .
- Equivalent to finding short vector in a lattice.

They also significantly overlap codebases.

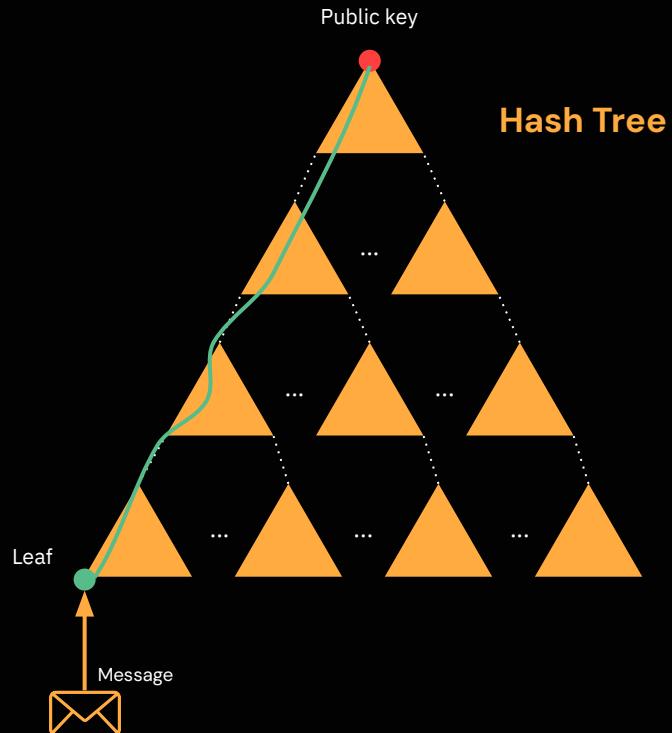


→ shortest vector
● origin

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?

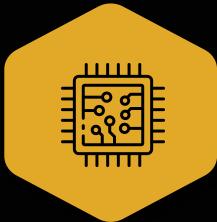


Interconnected systems, dependencies



Remote / long-lived systems

(Also some) challenges of PQ authentication

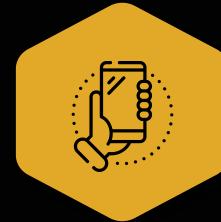


Reliance on
hardware



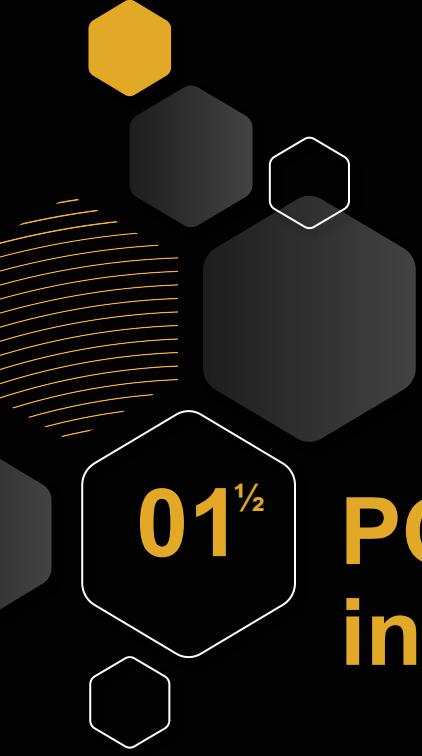
Low capacity devices
(hardware tokens,
smartcards...)

End-user distribution



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





01^½

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

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

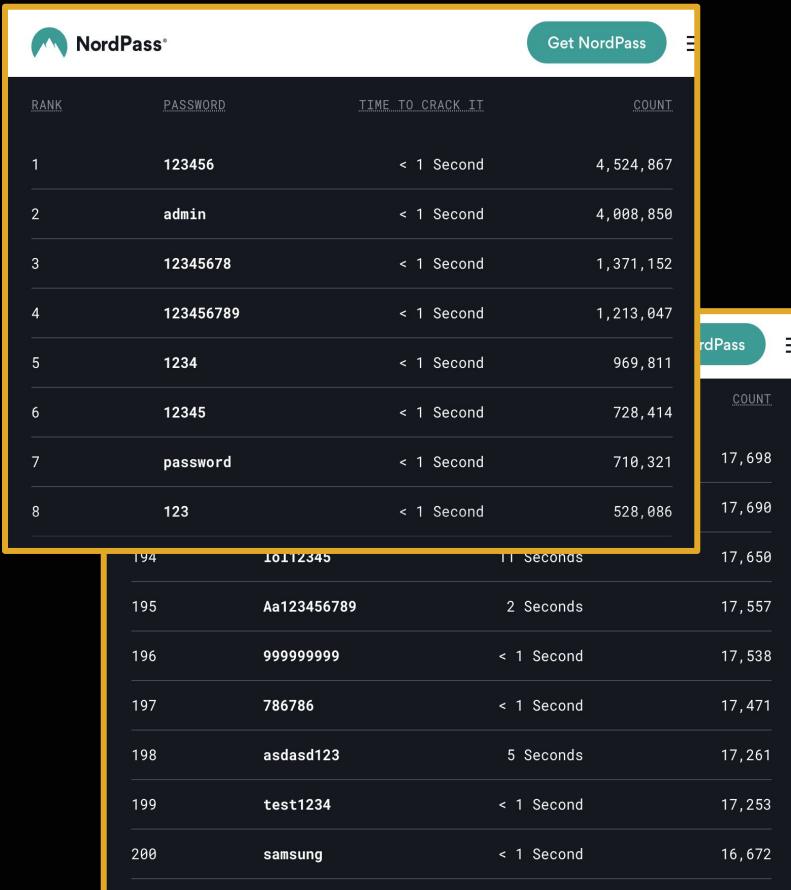
Use case: FIDO2

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.



The screenshot shows a NordPass interface with a search bar and a button labeled "Get NordPass". Below is a table of common passwords:

RANK	PASSWORD	TIME TO CRACK IT	COUNT
1	123456	< 1 Second	4,524,867
2	admin	< 1 Second	4,008,850
3	12345678	< 1 Second	1,371,152
4	123456789	< 1 Second	1,213,847
5	1234	< 1 Second	969,811
6	12345	< 1 Second	728,414
7	password	< 1 Second	718,321
8	123	< 1 Second	528,086
194	10112345	11 Seconds	17,698
195	Aa123456789	2 Seconds	17,557
196	999999999	< 1 Second	17,538
197	786786	< 1 Second	17,471
198	asdasd123	5 Seconds	17,261
199	test1234	< 1 Second	17,253
200	samsung	< 1 Second	16,672

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 passwords¹.

¹<https://us.norton.com/blog/privacy/password-statistics>

The collage consists of three distinct sections, each highlighted with a yellow border:

- Forbes Article Screenshot:** A news article titled "Warning As 26 Billion Records Leak: Dropbox, LinkedIn, Twitter Named" by Davey Winder. The article discusses the "RockYou2024" leak, stating it's the biggest password leak ever, involving nearly 10 billion credentials. It notes that such leaks could give hackers a significant advantage.
- CheckMyEmail.org Screenshot:** A screenshot of the website 'have i been pwned?' showing a search for the email address "admin@google.com". The result indicates that the email has been found in 88 data breaches and 32 pastes, with the word "pwned?" displayed prominently.
- LinkedIn Profile Screenshot:** A screenshot of a LinkedIn profile for Matt Binder, a Senior Contributor at Forbes. The profile includes a bio describing him as a "Veteran cybersecurity and tech analyst, journalist, hacker, author". It shows his connection to the RockYou2024 leak.

Password managers

Nearly 2/3 of internet users keep track of their passwords by memory or with handwritten notes¹.

Almost 1/4 people rely on a document on their computer to manage all of their passwords¹.

Less than 40% of organizations require the use of a password manager¹.



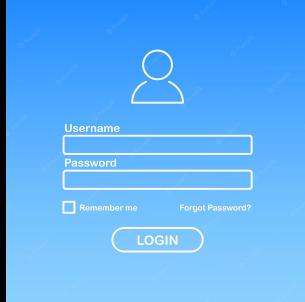
¹<https://us.norton.com/blog/privacy/password-statistics>

Problem statement

Classic authentication solutions for web are not working.

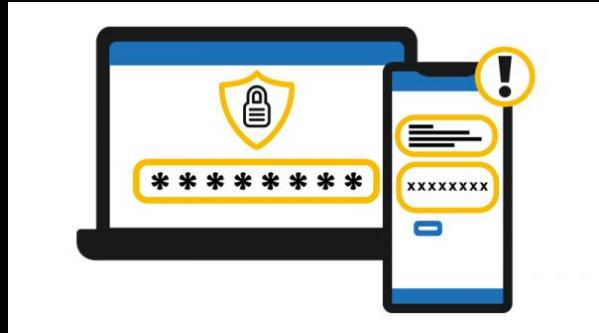
Passwords

- 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)



FIDO Authentication

A Passwordless Vision



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

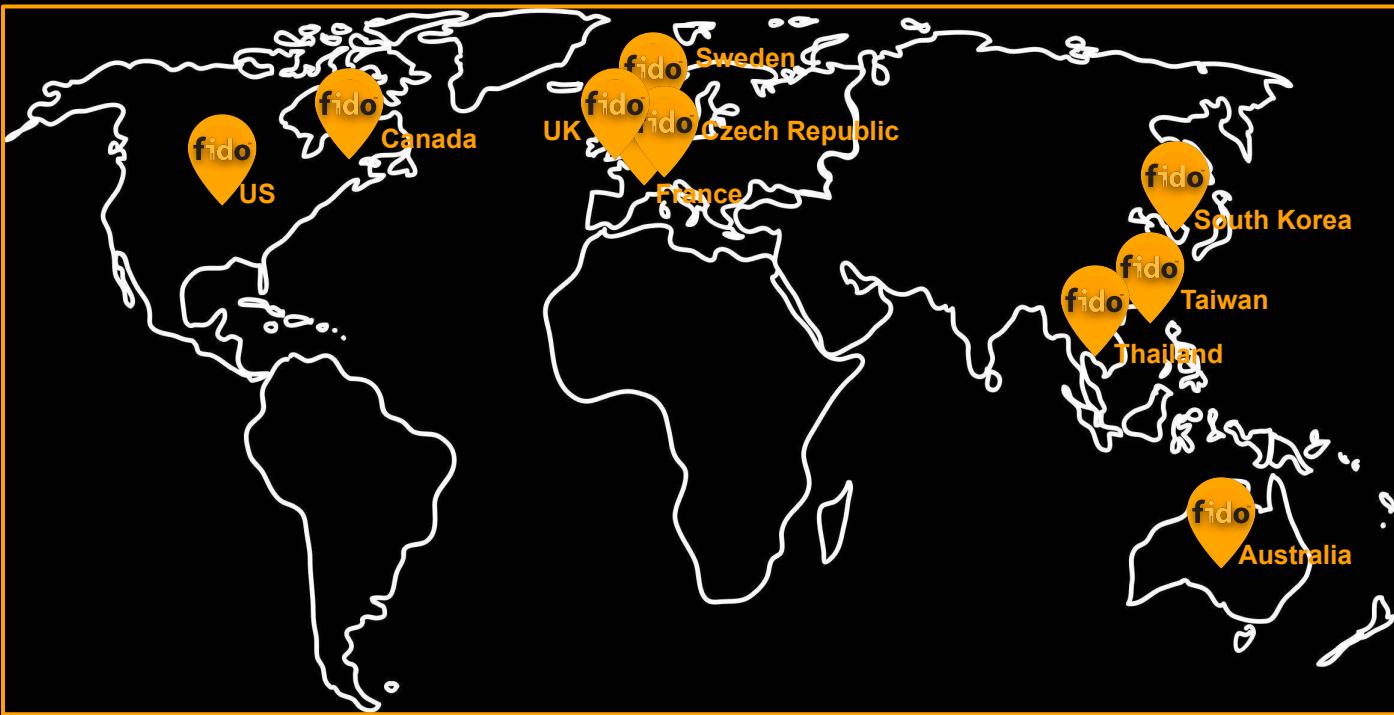
Board Level Members				
1Password	amazon	AMERICAN EXPRESS	apple	axiod
BEYOND IDENTITY	CISCO	CVSHealth	Daon	
DASHLANE	DELL	gis	EDITION	Google
HYPR	IDEAMIL	Infineon	intel	INTUIT
JUMIO	LastPass	Lenovo	LINEヤフー	mastercard
mercari	Meta	Microsoft	nok	dōcoMo
OneSpan	PayPal	PNCBANK	Prove	Qualcomm
RAON	RSA	SAMSUNG	THALES	TIKTOK
TRUSONA	usbank	VISA	WELLS FARGO	yubico

Sponsor Level Members				
1KOSMOS	Cloud	Akamai	AUOTIX	bankID bankID
bitwarden	BINANCE	Capital	CB	CHASE
coinbase	ComodoSecure	CYBERARK	DISCOVER	DocuSign
ebay	entersekt	EccelSocu	fime	FUJITSU
FUTURAE	GD	OPPO	Hedera	HID
HITACHI	HSBC	HUAWEI	IBM Security	IDnow
intercede	ISPR	iProov	JSB	
KDDI	KEEPER	M&T Bank	moz://a	NEC
NETFLIX	NRI SECURE	okta	onfido	persona
Ping	pwc	Rakuten	Red Hat	Siemens
RoboForm	salesforce	SANDBOXAQ	NEOBANK	Siemens
SK telecom	Socure	SoftBank	SPYDER	SONY
ST	smg	swissbit	SZFU	target
MITRE	twilio	Vanguard	VERIUM	Viaccess
Wisenet	WORLD PTE LTD	yahoo!		

Sponsor Level Members				
1KOSMOS	Cloud	Akamai	AUOTIX	bankID bankID
bitwarden	BINANCE	Capital	CB	CHASE
coinbase	ComodoSecure	CYBERARK	DISCOVER	DocuSign
ebay	entersekt	EccelSocu	fime	FUJITSU
FUTURAE	GD	OPPO	Hedera	HID
HITACHI	HSBC	HUAWEI	IBM Security	IDnow
intercede	ISPR	iProov	JSB	
KDDI	KEEPER	M&T Bank	moz://a	NEC
NETFLIX	NRI SECURE	okta	onfido	persona
Ping	pwc	Rakuten	Red Hat	Siemens
RoboForm	salesforce	SANDBOXAQ	NEOBANK	Siemens
SK telecom	Socure	SoftBank	SPYDER	SONY
ST	smg	swissbit	SZFU	target
MITRE	twilio	Vanguard	VERIUM	Viaccess
Wisenet	WORLD PTE LTD	yahoo!		



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

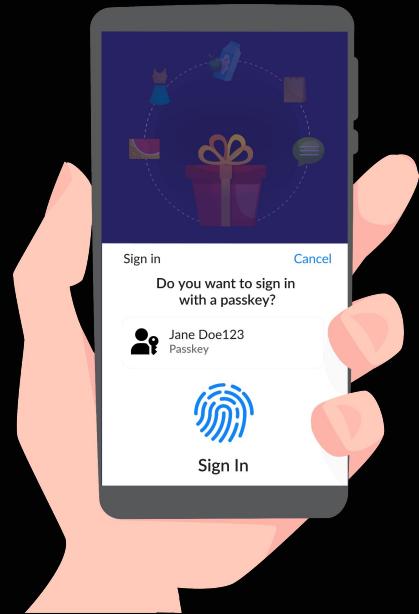
UK:

- 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 · Ravie

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



U2F

2nd factor authentication

A (very) brief history of FIDO authentication



U2F

2nd factor authentication



FIDO2 = CTAP (FIDO) + WebAuthn (W3C)

Security tokens generate credentials which are registered and used to authenticate

A (very) brief history of FIDO authentication



U2F

2nd factor authentication



FIDO2 = CTAP (FIDO) + WebAuthn (W3C)

Security tokens generate credentials which are registered and used to authenticate

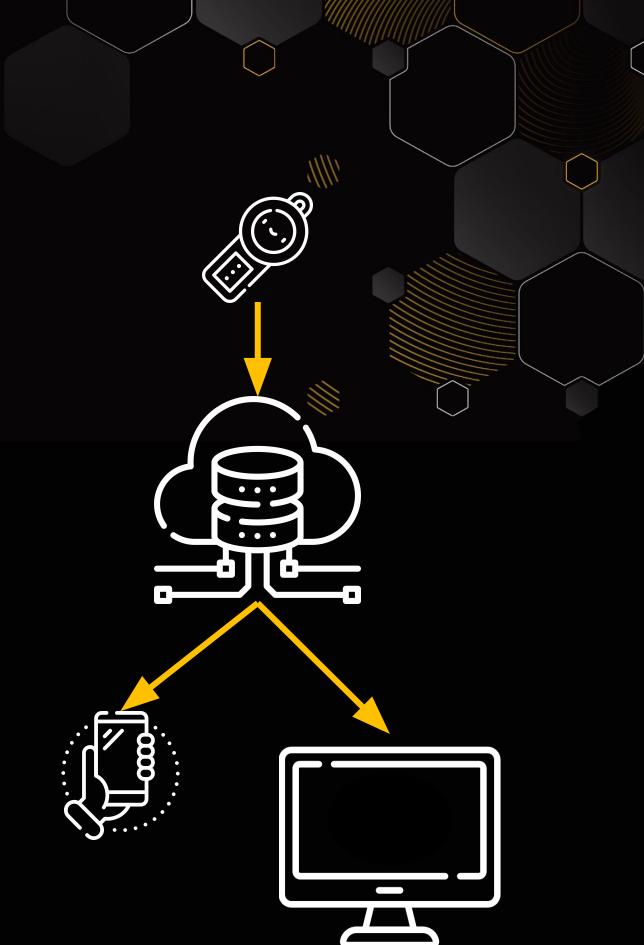


Passkeys

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



U2F

2nd factor authentication



FIDO2 = CTAP (FIDO) + WebAuthn (W3C)

Security tokens generate credentials which are registered and used to authenticate



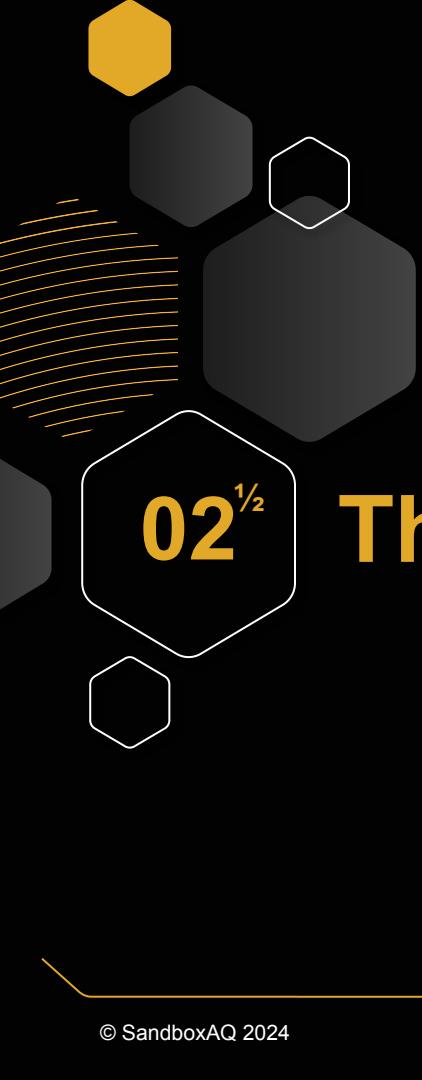
Passkeys

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



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

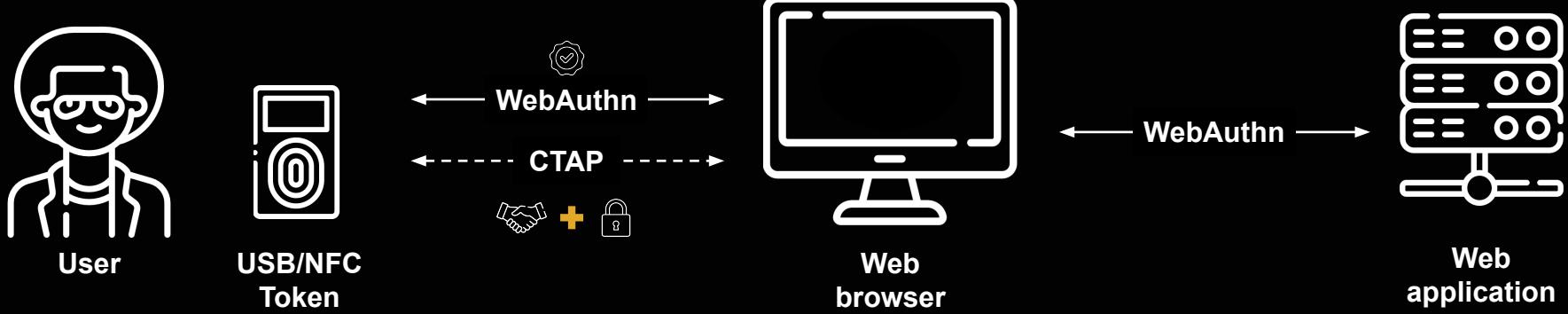
Acknowledging the quantum threat and need to select suitable PQC algorithms and to prepare for smooth transition



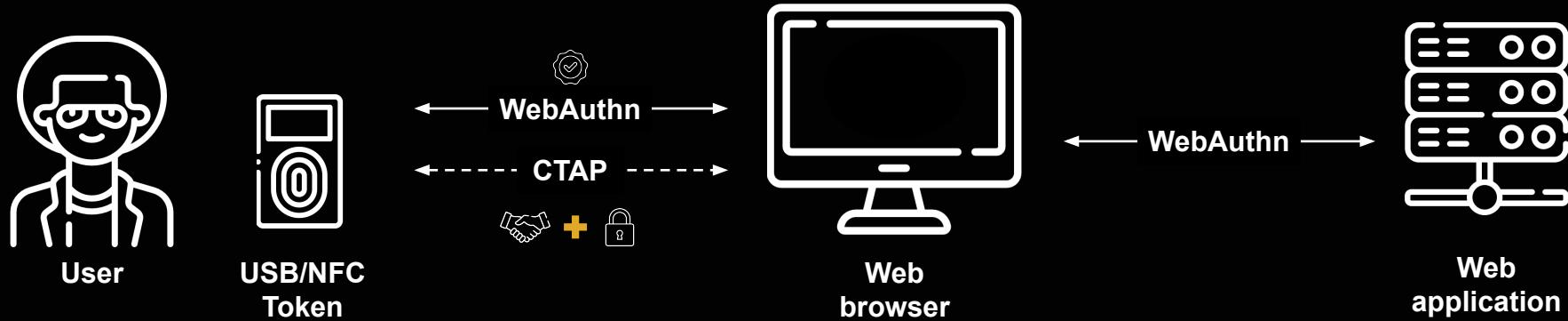
02^{1/2}

The FIDO2 Protocol Flow

FIDO2 protocol



FIDO2 = WebAuthn + CTAP



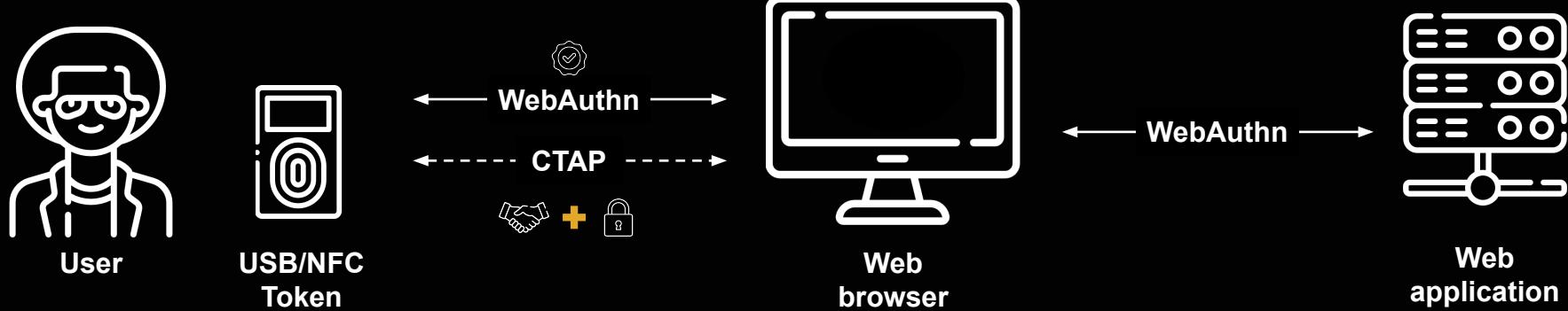
WebAuthn

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

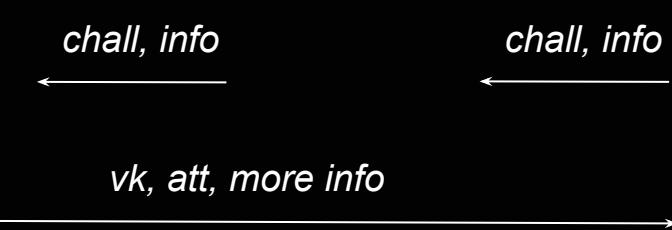
CTAP (Client To Authenticator Protocol)

Sub-protocol to ensure only a browser trusted by the user can communicate directly with the token.

Registration

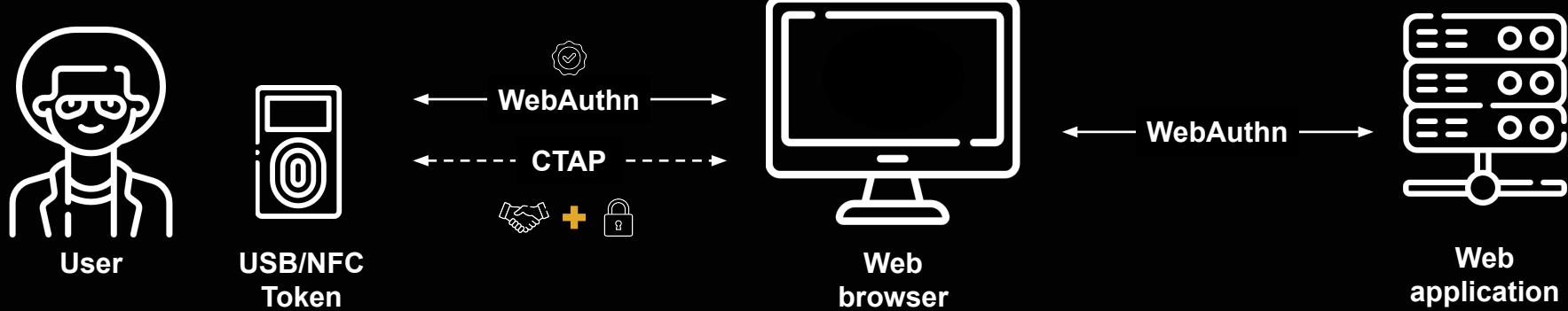


- key exchange + symm. encryption
- gesture
- (sk, vk) generate **assertion keys**
- att generate attestation signature

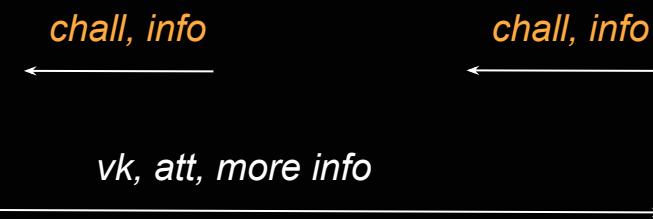


- $chall$ randomly chosen
- $info$ session info
- verify $info, att$
- save vk

Registration

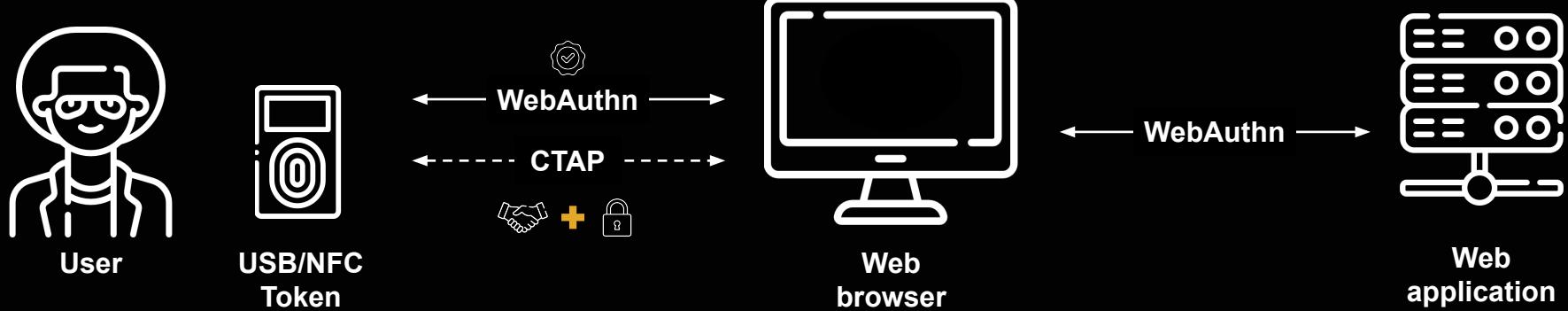


- key exchange + symm. encryption
- gesture
- (sk, vk) generate **assertion keys**
- att generate attestation signature

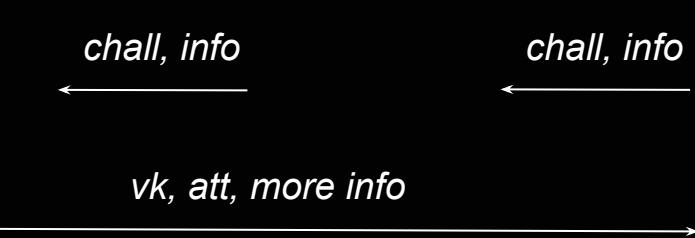


- *chall* randomly chosen
- *info* session info
- verify *info, att*
- save *vk*

Registration

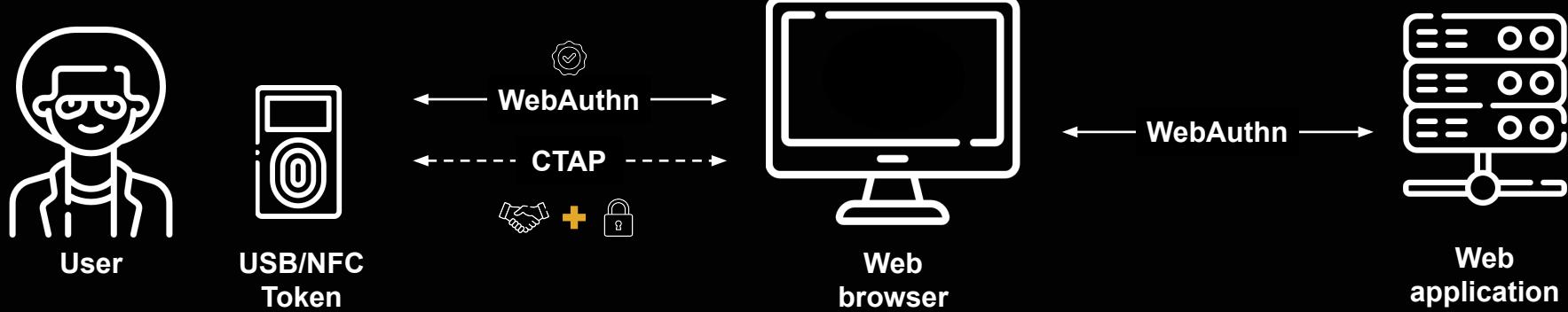


- key exchange + symm. encryption
- gesture
- (sk, vk) generate **assertion keys**
- att generate attestation signature



- $chall$ randomly chosen
- $info$ session info
- verify $info, att$
- save vk

Registration

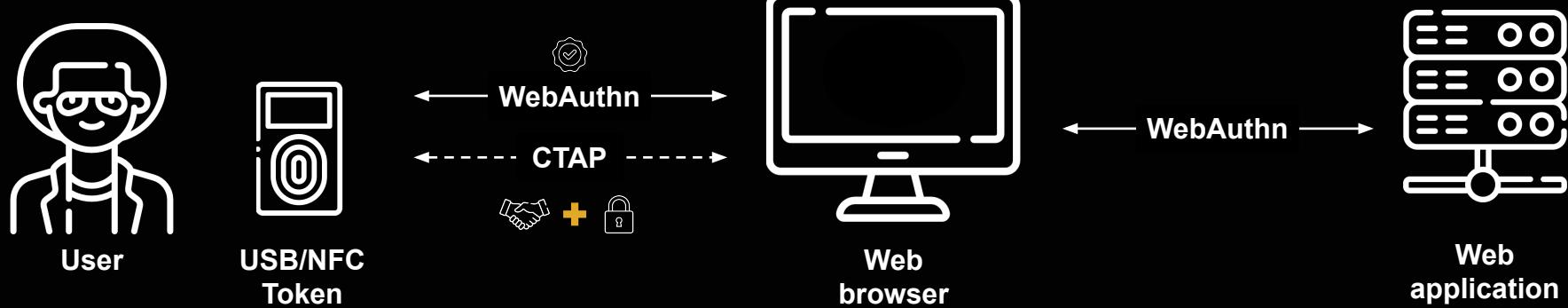


- key exchange + symm. encryption
- gesture
- (sk, vk) generate **assertion keys**
- att generate attestation signature



- *chall* randomly chosen
- *info* session info
- verify *info, att*
- save *vk*

Authentication



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



- *chall* randomly chosen
- *info* session info
- verify *info*, *sig*
- save *vk*

Post-Quantum FIDO2

PQ
readiness

WebAuthn

Yes
if signature scheme is PQ secure

PQ
instantiation

- Use signature negotiation in WebAuthn to include PQ/hybrid signature algorithms.
- Use PQ signature.

CTAP

Yes
if DH-based CTAP subroutine is instantiated with a (PQ) KEM

- *Protocol* negotiation in CTAP 2.1 includes PQ/hybrid KEM.
- Use PQ KEM.
- Increase output length hash.

Post-Quantum FIDO2

PQ
readiness

WebAuthn

Yes
if signature scheme is PQ secure

PQ
instantiation

- Use signature negotiation in WebAuthn to include PQ/hybrid signature algorithms.
- Use PQ signature.

CTAP

Yes
if DH-based CTAP subroutine is instantiated with a (PQ) KEM

Backwards
Compatibility

- Cryptographic negotiations between User and Web Service similar to TLS.
- Ensures backwards compatibility between those supporting PQC and not.

Post-Quantum FIDO2

PQ
readiness

WebAuthn

Yes
if **signature** scheme is PQ secure

PQ
instantiation

- Use **signature** negotiation in WebAuthn to include PQ/hybrid **signature** algorithms.
- Use PQ **signature**.

Backwards
Compatibility

CTAP

Yes
if DH-based CTAP subroutine is instantiated with a (PQ) KEM

- *Protocol* negotiation in CTAP 2.1 includes PQ/hybrid KEM.
- Use PQ KEM.
- Increase output length hash.

- Cryptographic negotiations between User and Web Service similar to TLS.
- Ensures backwards compatibility between those supporting PQC and not.

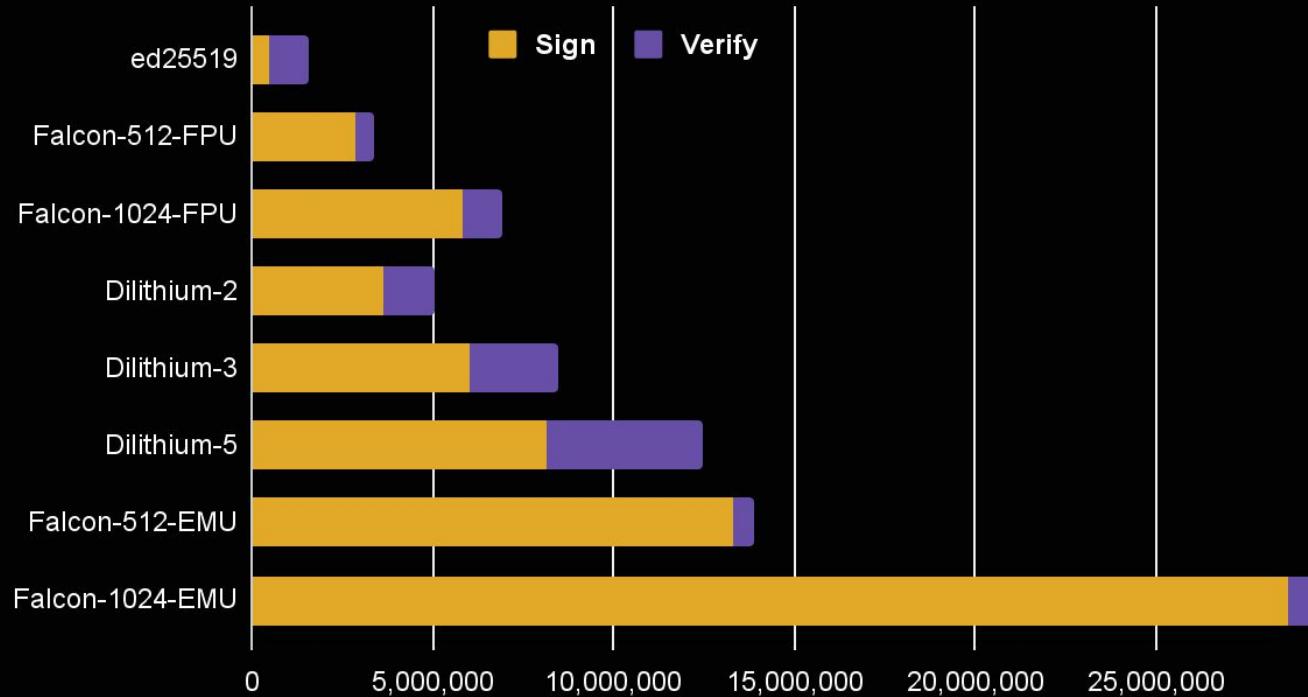


02^¾

Signature on Embedded Devices

Comparing Signature on ARM Cortex M7

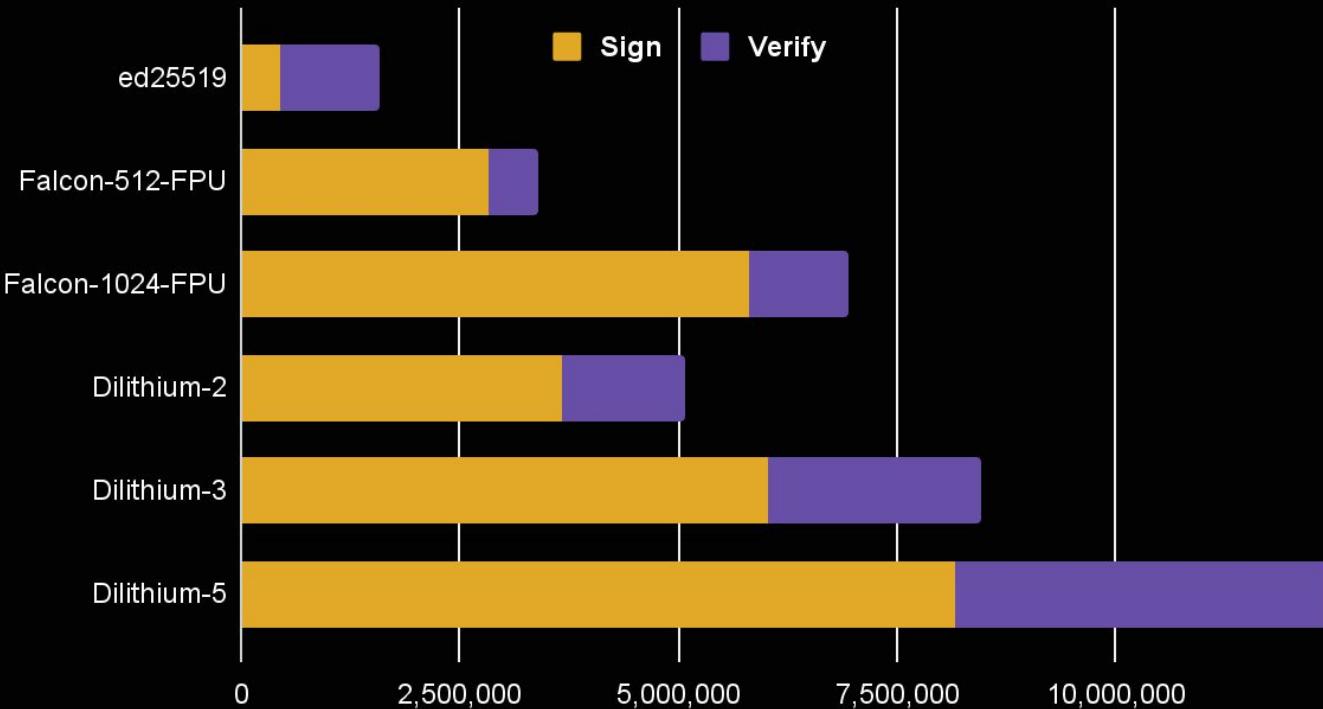
Results given in Clock Cycles



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>.

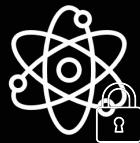


03

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

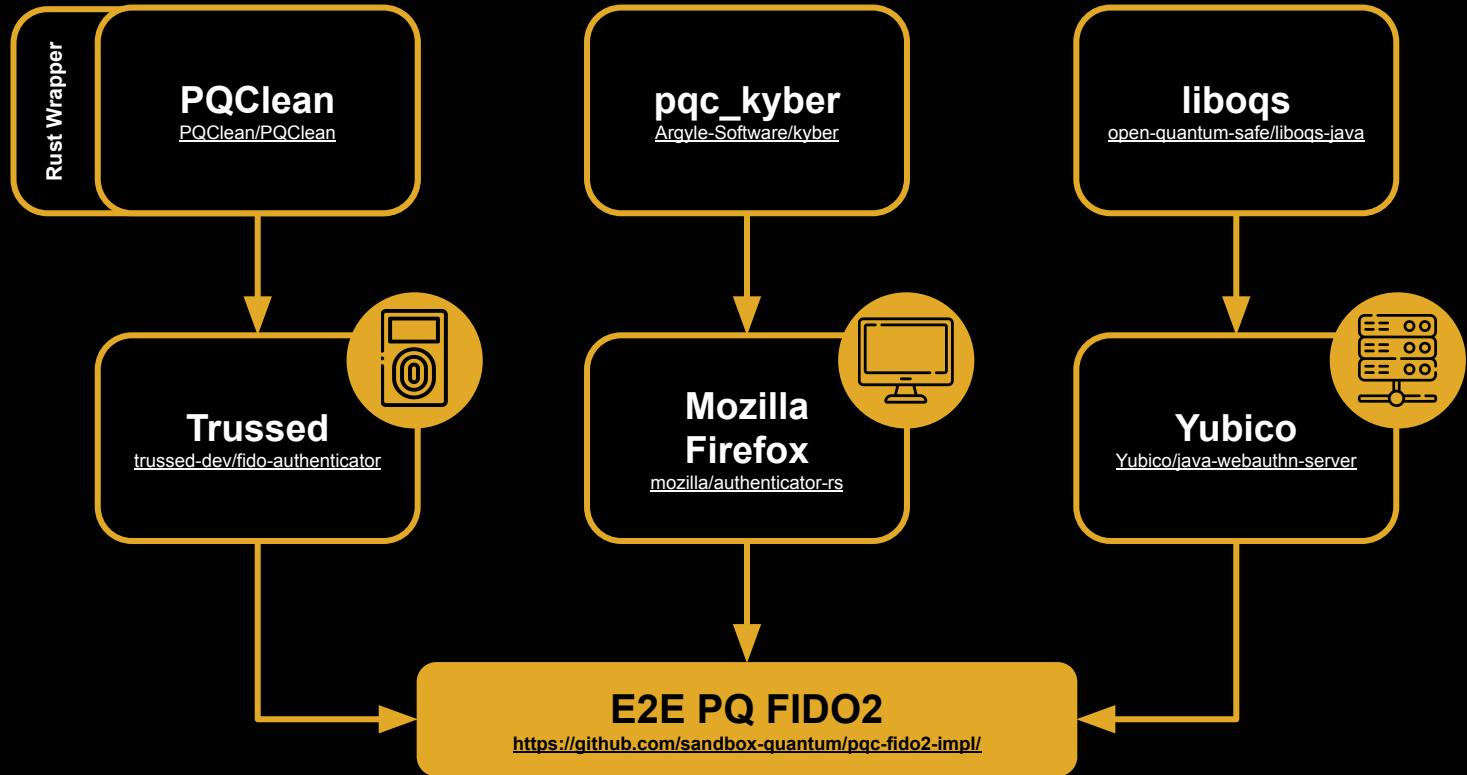
E2E PQ FIDO2

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

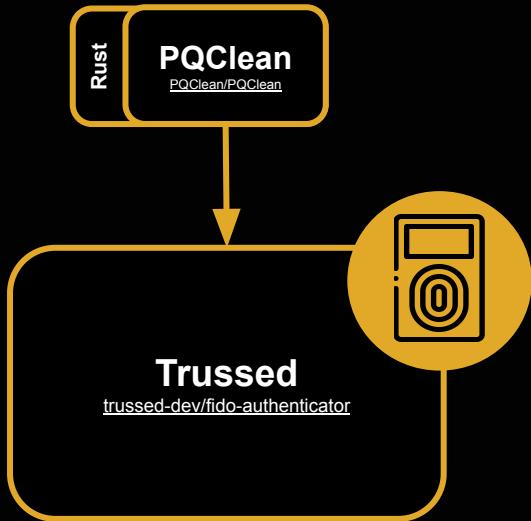
“Libraries are where it all begins” – Rita Dove



“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

E2E PQ FIDO2

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

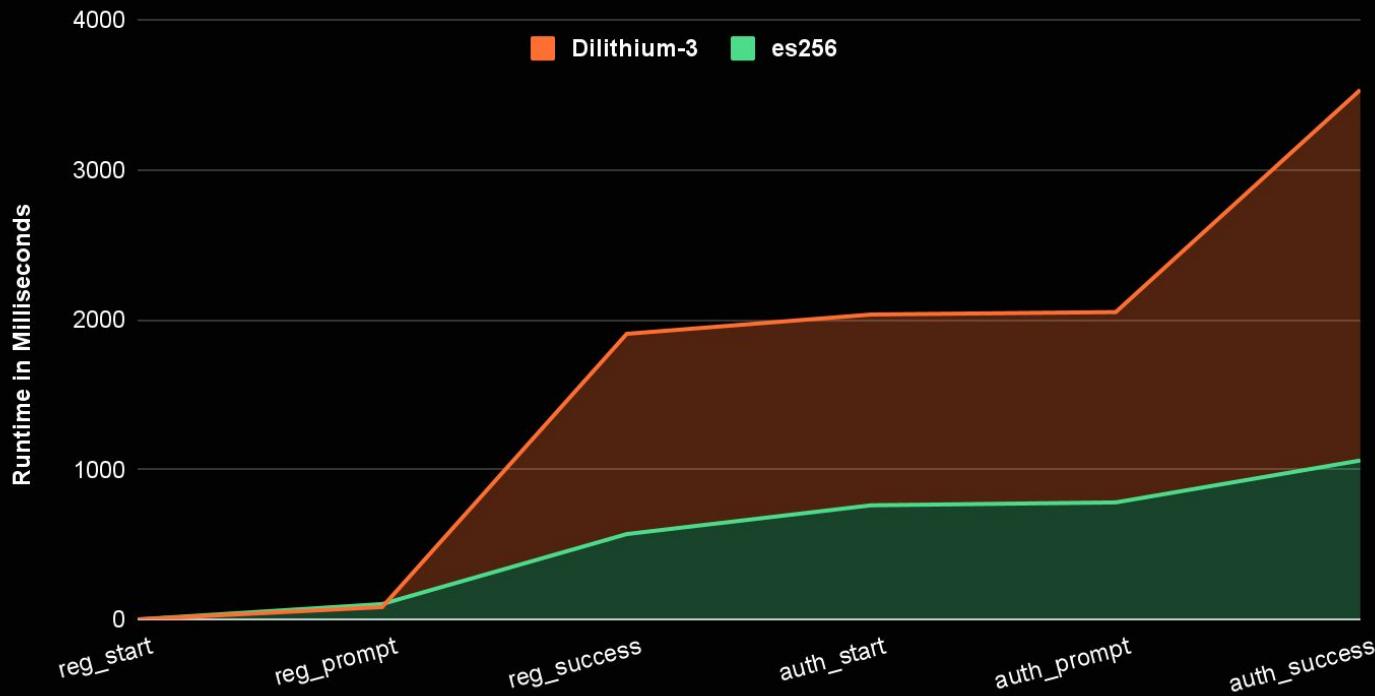
Performance of FIDO2

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

The screenshot shows a laptop screen with two main windows. On the left is a web browser window titled "WebAuthn Demo" displaying a demo application for a Java WebAuthn server. The page has a green header "JAVA-WEBAUTHN-SERVER DEMO". Below it are input fields for "Username", "Display name", "Credential nickname", "Credential ID", and buttons for "Create account with non-discoverable credential", "Create account with passkey", "Authenticate with username", and "Authenticate with passkey". There are also "Deregister" and "Log out" buttons. On the right is a terminal window showing the command `./gradlew run` and its output log. The log includes debug messages from the Java code, information about the Yubico WebAuthn configuration, and various system logs and warnings related to Jersey and Jetty servers.

```
Java-webauthn-server -- ./gradlew run -- ./gradlew --java -Xmx84m -Xms84m -Dorg.gradle.appname=gradlew -classpath ...  
+-- java:webauthn-server git+(ff021a87) x ./gradlew run  
|  
+-- Task :weauthn-server-demo run  
17:09:08.442+0100 [main] DEBUG demo.webauthn.Config - YUBICO_WEBAUTHN_ALLOWED_ORIGINS: null  
17:09:08.444+0100 [main] INFO demo.webauthn.Config - Origins: [https://localhost:8443]  
17:09:08.444+0100 [main] DEBUG demo.webauthn.Config - RP name: null  
17:09:08.444+0100 [main] DEBUG demo.webauthn.Config - RP ID: null  
17:09:08.445+0100 [main] DEBUG demo.webauthn.Config - RP name not given - using default.  
17:09:08.445+0100 [main] DEBUG demo.webauthn.Config - RP ID not given - using default.  
17:09:08.445+0100 [main] INFO demo.webauthn.Config - RP identity: RelyingPartyIdentity{name=Yubico WebAuthn demo, id=localhos...}  
17:09:08.445+0100 [main] INFO demo.webauthn.WebAuthnServer - Using only Yubico JSON file for attestation metadata.  
17:09:08.660+0100 [main] INFO org.eclipse.jetty.util.log - Logging initialized @416ms to org.eclipse.jetty.util.log.Slf4jLog  
17:09:08.791+0100 [main] INFO org.eclipse.jetty.server.Server - jetty-9.4.9.+20180320; built: 2018-03-20T13:21:10+01:00; git: 1f8159b1ea...  
Mar 21, 2024 5:09:08 PM org.glassfish.jersey.message.internal.MessagingBinders$EnabledProviders$bindToBinder  
WARNING: A class javax.activation.DataSource for a default provider MessageBodyWriter<javax.activation.DataSource> was not found. The provider is not available.  
Mar 21, 2024 5:09:08 PM org.glassfish.jersey.server.wadl.WadlFeature configure  
WARNING: JAX-B API not found . WADL feature is disabled.  
Mar 21, 2024 5:09:08 PM org.glassfish.jersey.internal.inject.Providers checkProviderRuntime  
WARNING: A provider demo.webauthn.WebAuthnRestResource registered in SERVER runtime does not implement any provider interfaces applicable in the SERVER runtime. Due to constraint configuration problems the provider demo.webauthn.WebAuthnRestResource will be ignored.  
17:09:08.952+0100 [main] INFO o.e.j.server.handler.ContextHandler - Started o.e.j.s.ServletContextHandler@267517e4{/file:///Users/sandra.guasch/Documents/pqfido_test_140224/pqc-fido2-impl/java-webauthn-server/weauthn-server-demo/src/main/webapp/AVAILABLE}  
17:09:08.964+0100 [main] INFO o.e.jettty.util.ssl.SslContextFactory - x509=X509@777a281fc(serverKey,=null,=null) for SslContextFactory@4912d525(provider=null, keyStore=null, keyStoreFile=null, trustStore=null)  
17:09:09.026+0100 [main] INFO o.e.jettly.server.AbstractConnector - Started ServerConnector@2761f037(SSL,[ssl, http/1.1])(127.0.0.1:8443)  
17:09:09.026+0100 [main] INFO org.eclipse.jetty.server.Server - Started @777ms  
<=====> 95% EXECUTING (65s)  
> :weauthn-server-demo:run  
|  
+-- Sandra Guasch Castello
```

AGENDA

01

FIDO2

Introduction to the FIDO2 protocol

02

PQ-readiness of FIDO2

Analysis of WebAuthn and CTAP

03

E2E PQ FIDO2 OSS

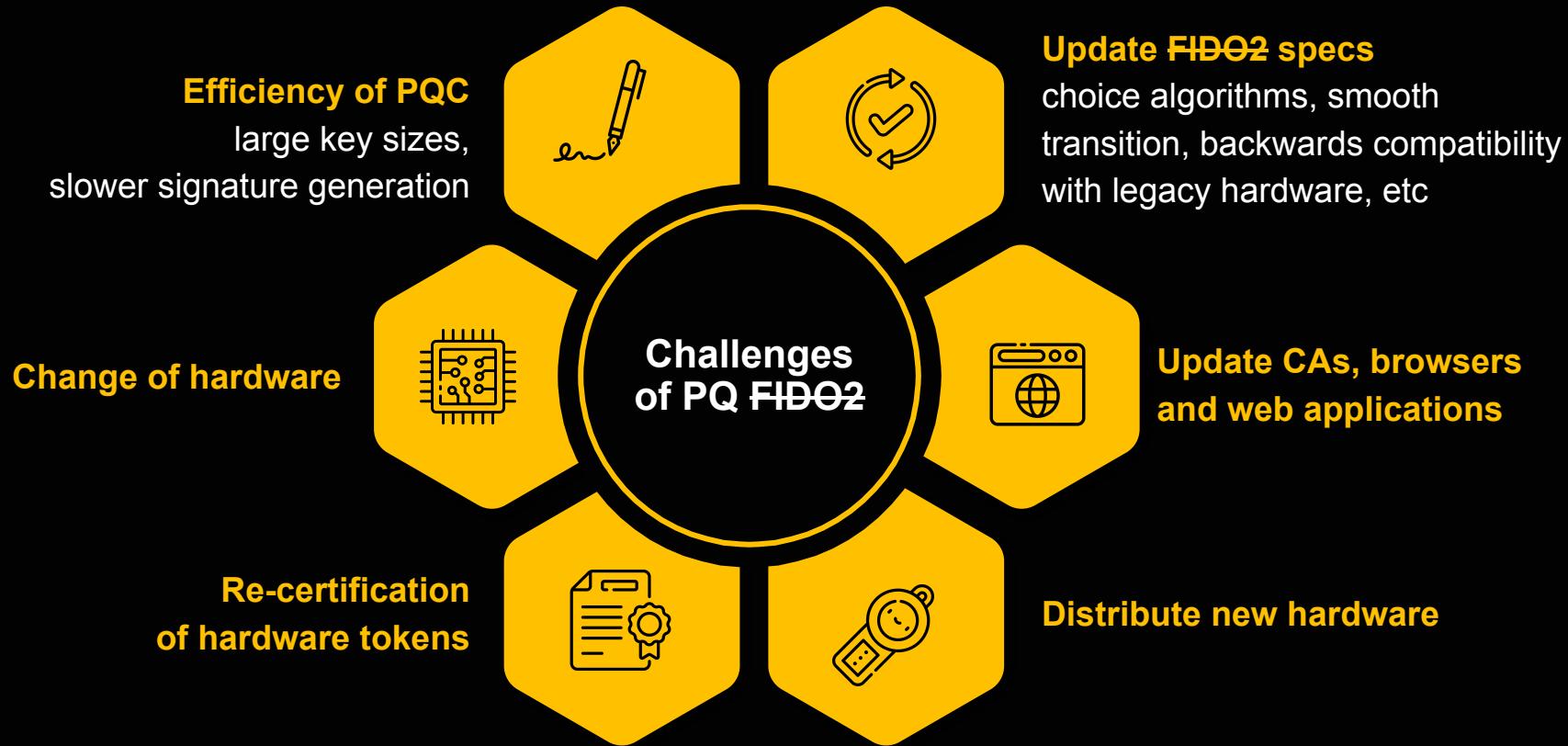
Implementation details

04

Challenges and future work

Additional modes to be considered in the PQ migration





Change of hardware

Re-certification
of hardware tokens

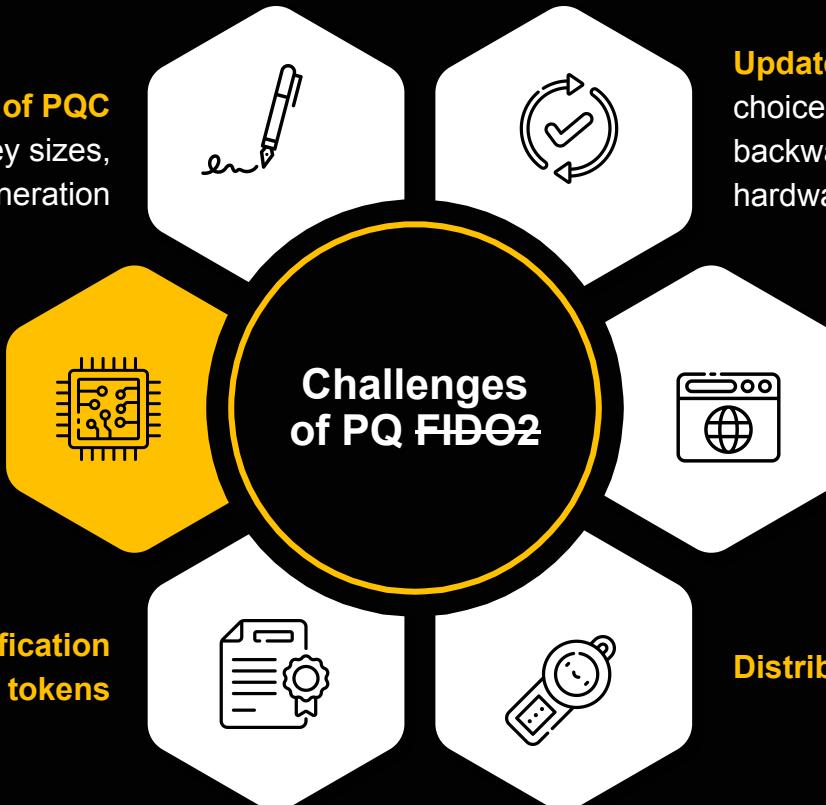
Challenges of PQ FIDO2

Efficiency of PQC
large key sizes,
slower signature generation

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

**Update CAs, browsers and
web applications**

Distribute new hardware



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.

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](#)

We are hiring

Check out sandboxaq.com/careers

Thank you