SANDBOXAQ™

Introduction to Passkeys

Dr. Nina Bindel
Staff Research Scientist

Nina Bindel
she/her

ninabindel.de
nbindel
@NinaBindel

Affiliation

SANDBOXAQ

since July 2022

To attest or not to attest, this is the question – Provable attestation in FIDO2

Nina Bindel[1], Nicolas Gama[1], Sandra Guasch[1], and Eyal Ronen[2]

[1] SandboxAQ, Palo Alto, CA, USA, {first.second}@sandboxaq...
[2] Tel Aviv University, Tel Aviv, Israel, eyal.ronen@cs...

When Cryptography Needs a Hand: Practical Post-Quantum Authentication for V2V Communications

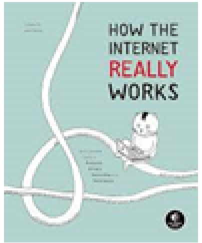
Geoff Twardokus, Nina Bindel, Hanif Rahbari, Sarah McCarthy

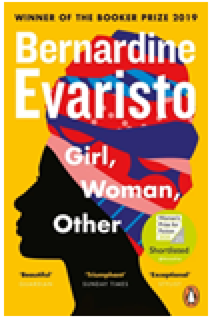
Quantum Lattice Enumeration in Limited Depth

Nina Bindel[1], Xavier Bonnetain[2], Marcel Tiepelt[3], and Fernando Virdia[4]

Latest published papers

Book recommendation

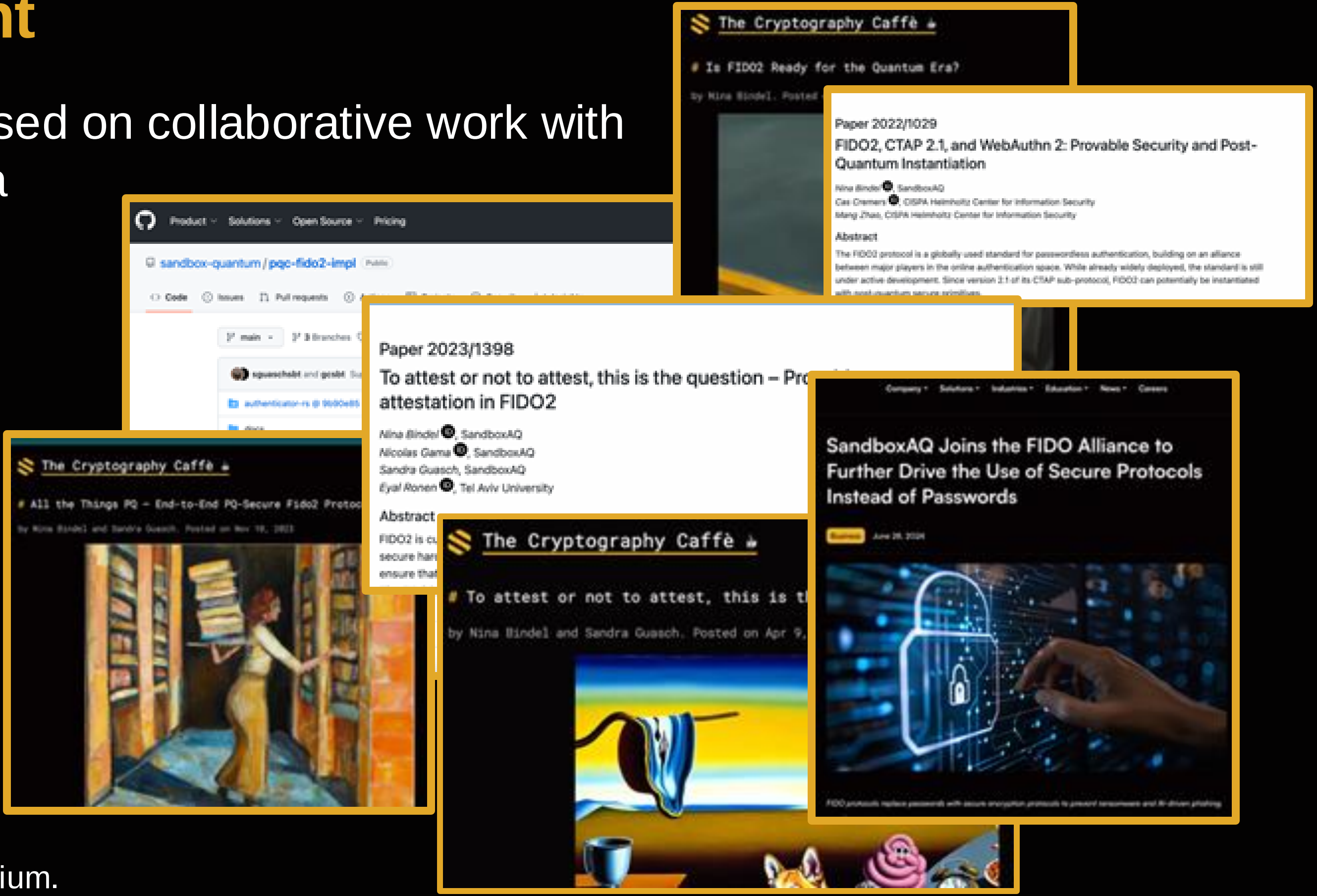




Acknowledgment

This presentation is based on collaborative work with

- Gabriel Campagna
- Cas Cremers
- Nicolas Gama
- Sandra Guasch
- James Howe
- Kyle Kotowick
- Duc Nguyen
- Eyal Ronen
- Spencer Wilson
- Tarun Yadav
- Mang Zhao



The Cryptography Caffè

Is FIDO2 Ready for the Quantum Era?

by Nina Bindel. Posted

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.

sandbox-quantum / pqc-fido2-impl Public

sguaschabt and gesbt

authenticator-rs @ 9b90e85

docs

The Cryptography Caffè

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

by Nina Bindel and Sandra Guasch. Posted on Nov 16, 2023

Paper 2023/1398

To attest or not to attest, this is the question – Pro attestation in FIDO2

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

Abstract

FIDO2 is cu secure har ensure tha

The Cryptography Caffè

To attest or not to attest, this is t

by Nina Bindel and Sandra Guasch. Posted on Apr 9,

SandboxAQ Joins the FIDO Alliance to Further Drive the Use of Secure Protocols Instead of Passwords

Business June 26, 2024

FIDO protocols replace passwords with secure encryption protocols to prevent ransomware and AI-driven phishing

All icons are from flaticon premium.

AGENDA

01 Passkeys/FIDO2 – a protocol for a passwordless Internet

02 The FIDO2 Protocol Flow

03 Post-Quantum FIDO2

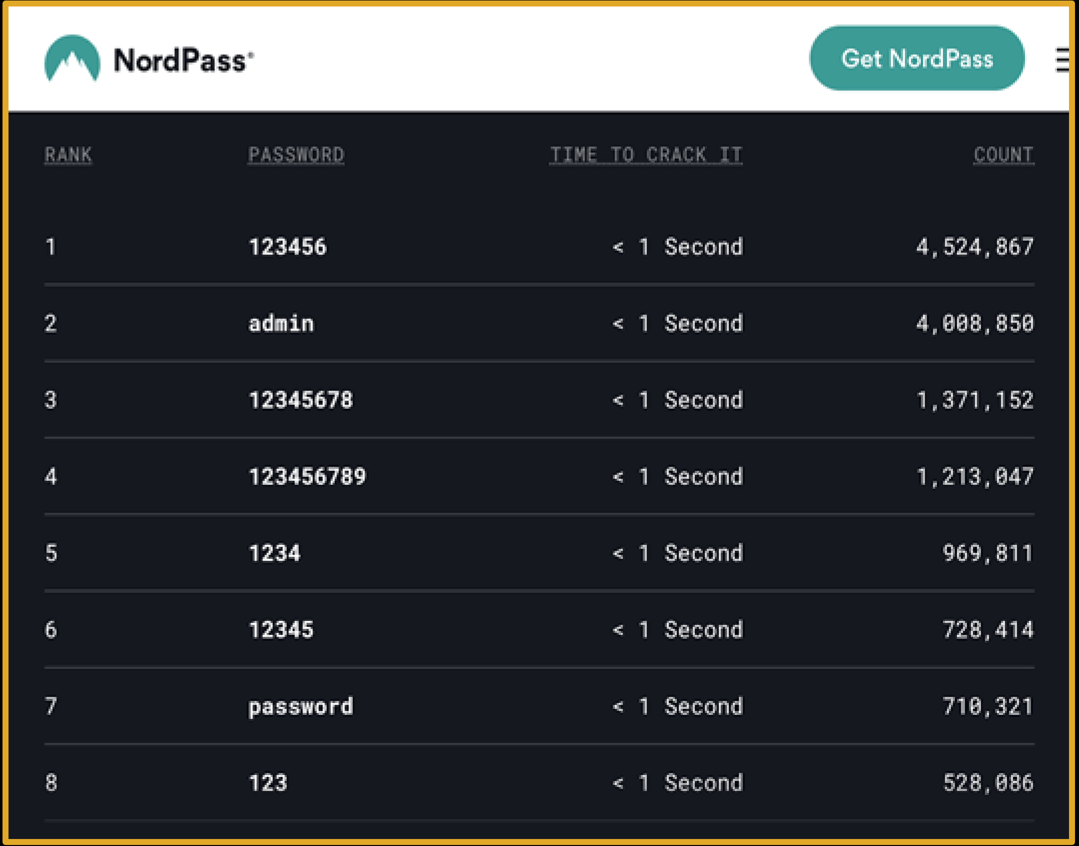
04 Challenges

01

FIDO2 – a protocol for a passwordless Internet

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

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

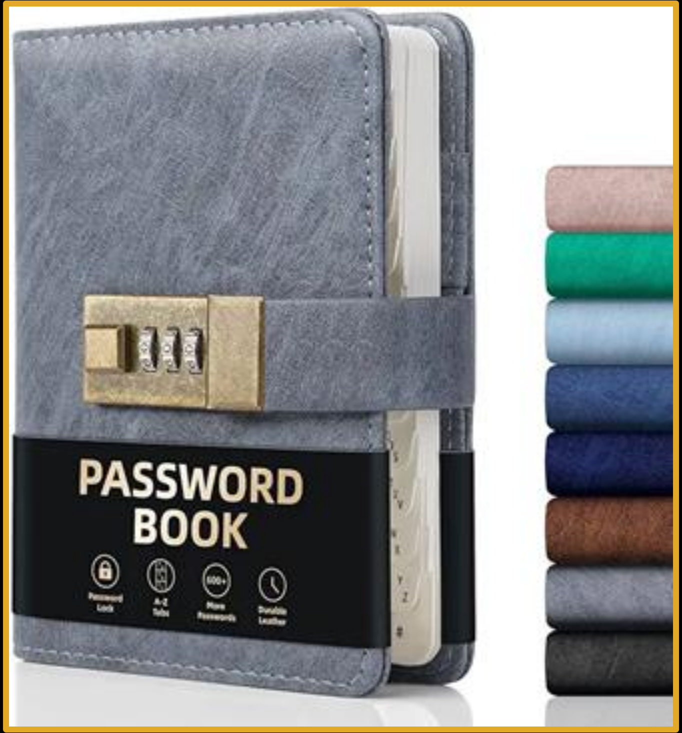


RANK	PASSWORD	TIME TO CRACK IT	COUNT
1	123456	< 1 Second	4,524,867
2	admin	< 1 Second	4,008,850
3	12345678	< 1 Second	1,371,152
4	123456789	< 1 Second	1,213,047
5	1234	< 1 Second	969,811
6	12345	< 1 Second	728,414
7	password	< 1 Second	710,321
8	123	< 1 Second	528,086

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].



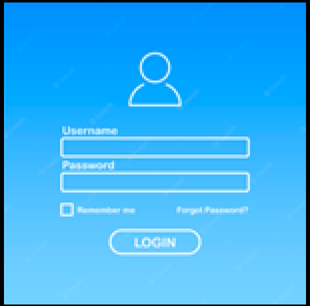
[1] https://us.norton.com/blog/privacy/password-statistics

Problem statement

Classic authentication solutions for web are not working.

Passwords

- Hard to remember or weak
- Vulnerable to phishing attacks
- Synchronisation across devices can be challenging (pwd managers)



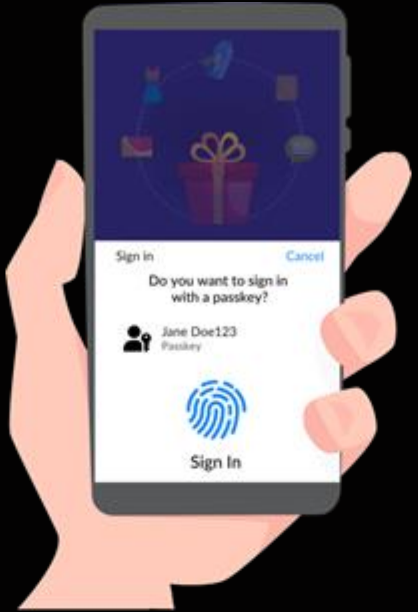
Multi-factor authentication / OTPs

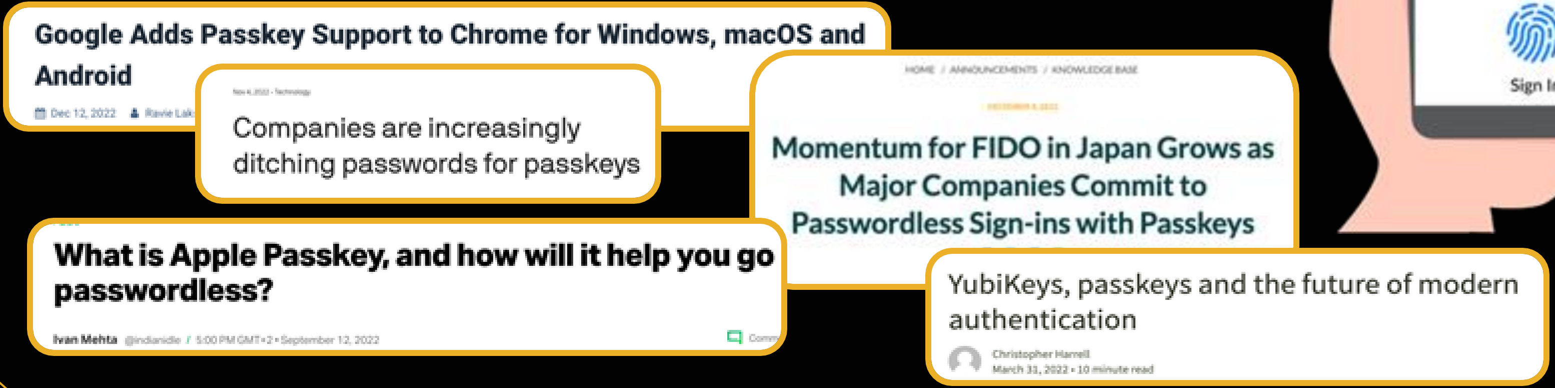
- Low usability
- Still rely on passwords
- Still vulnerable to phishing

Passwordless Alternative: FIDO Authentication

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 Lak

Nov 4, 2022 · Technology

Companies are increasingly ditching passwords for passkeys

HOME / ANNOUNCEMENTS / KNOWLEDGE BASE

DECEMBER 8, 2022

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

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

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

YubiKeys, passkeys and the future of modern authentication

Christopher Harrell
March 31, 2022 • 10 minute read

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

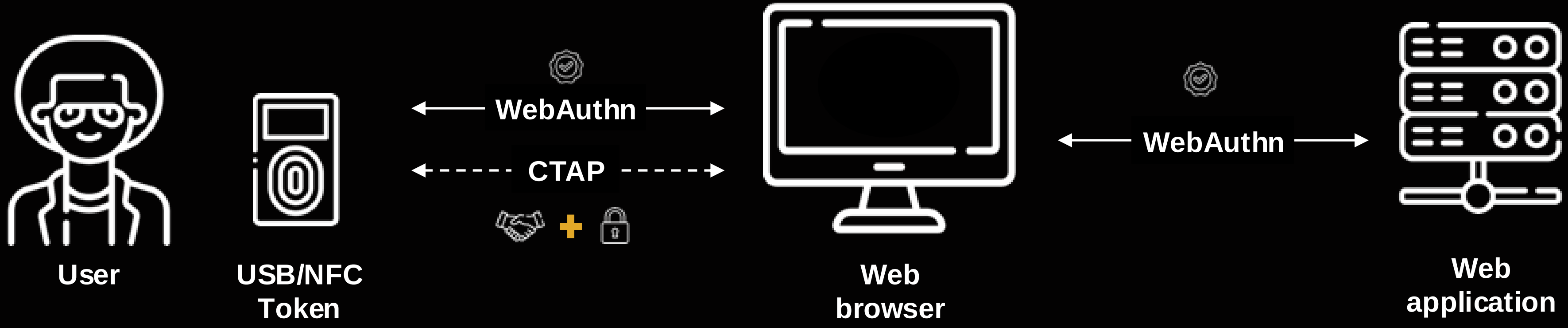
White Paper: Addressing FIDO Alliance’s ‘Technologies in Post Quantum World’

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

02

The FIDO2 Protocol Flow

FIDO2 = WebAuthn + CTAP



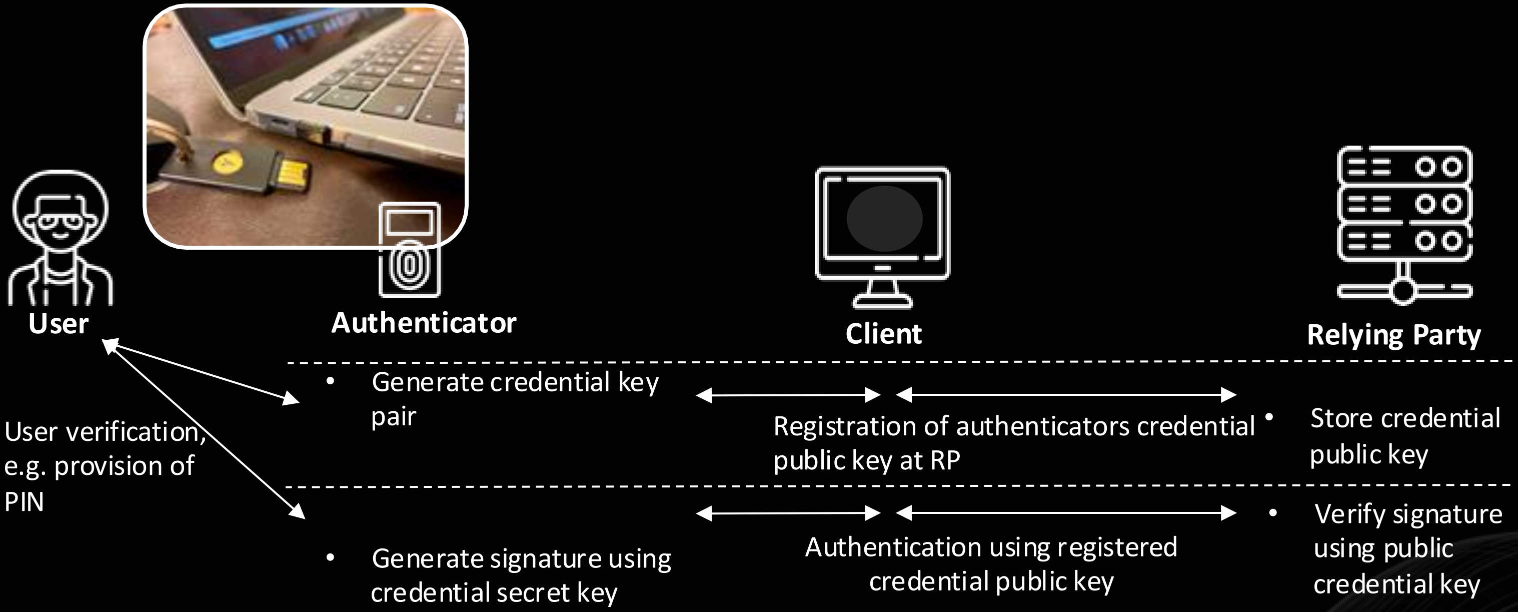
WebAuthn

Sub-protocol between the client and the server to let the user authenticate into the web service with the hardware token

CTAP (Client To Authenticator Protocol)

Sub-protocol between the token and the client to also ensure only browsers trusted by the user can communicate directly with the token

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

- Generate signature using credential secret key

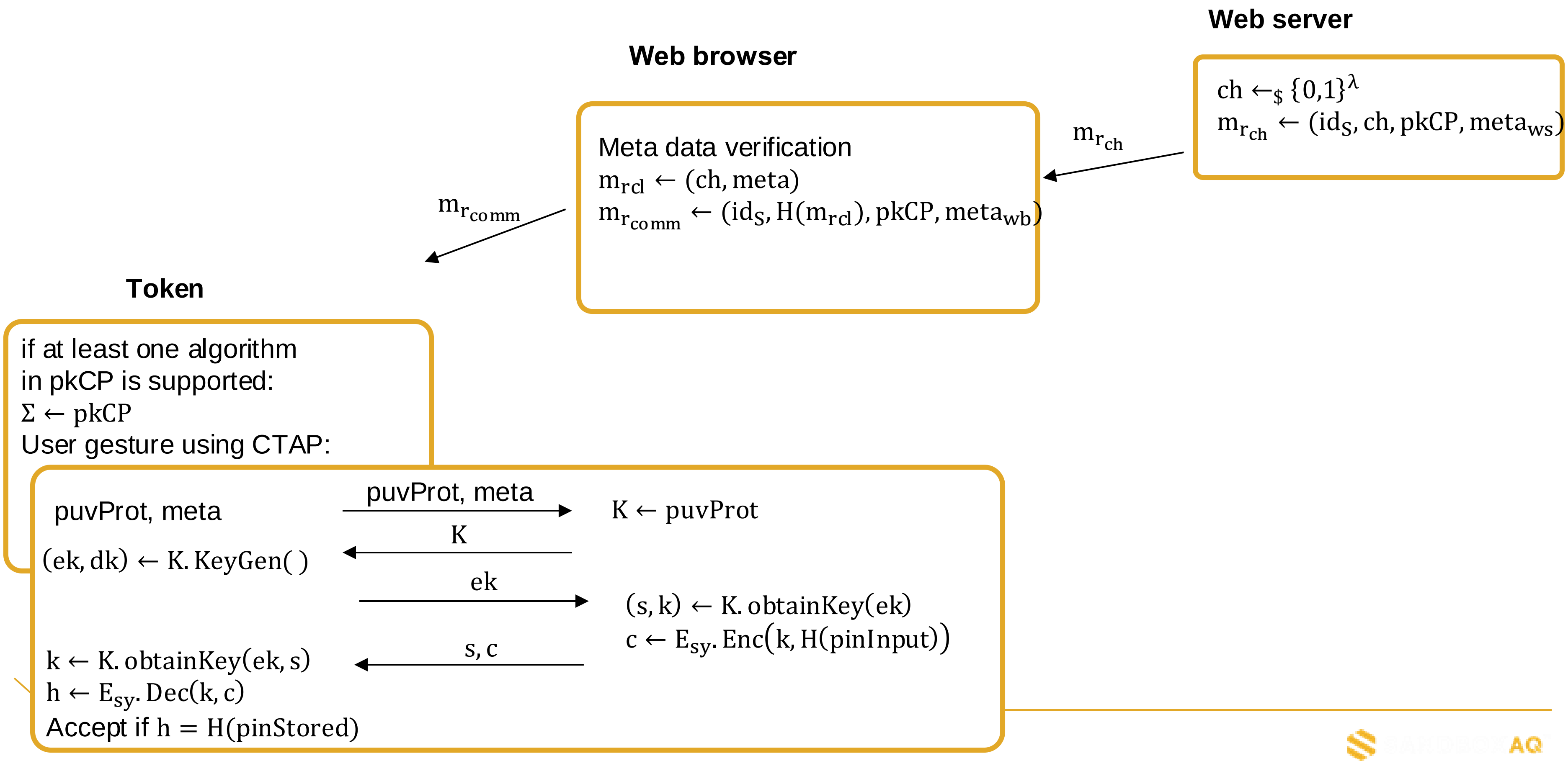
Authentication using registered credential public key

- Verify signature using public credential key

© SandboxAQ

SandboxAQ Proprietary Material

Cryptographic details of registration flow



Web server

$ch \leftarrow_{\$} \{0,1\}^{\lambda}$

$m_{r_{ch}} \leftarrow (id_S, ch, pkCP, meta_{ws})$

$m_{r_{ch}}$

Web browser

Meta data verification

$m_{rcl} \leftarrow (ch, meta)$

$m_{r_{comm}} \leftarrow (id_S, H(m_{rcl}), pkCP, meta_{wb})$

$m_{r_{comm}}$

Token

if at least one algorithm in pkCP is supported:

$\Sigma \leftarrow pkCP$

User gesture using CTAP:

puvProt, meta

$\xrightarrow{\text{puvProt, meta}}$

$K \leftarrow puvProt$

$\xleftarrow{K}$

$(ek, dk) \leftarrow K.KeyGen(\,)$

$\xrightarrow{ek}$

$(s, k) \leftarrow K.obtainKey(ek)$

$c \leftarrow E_{sy}.Enc\big(k, H(pinInput)\big)$

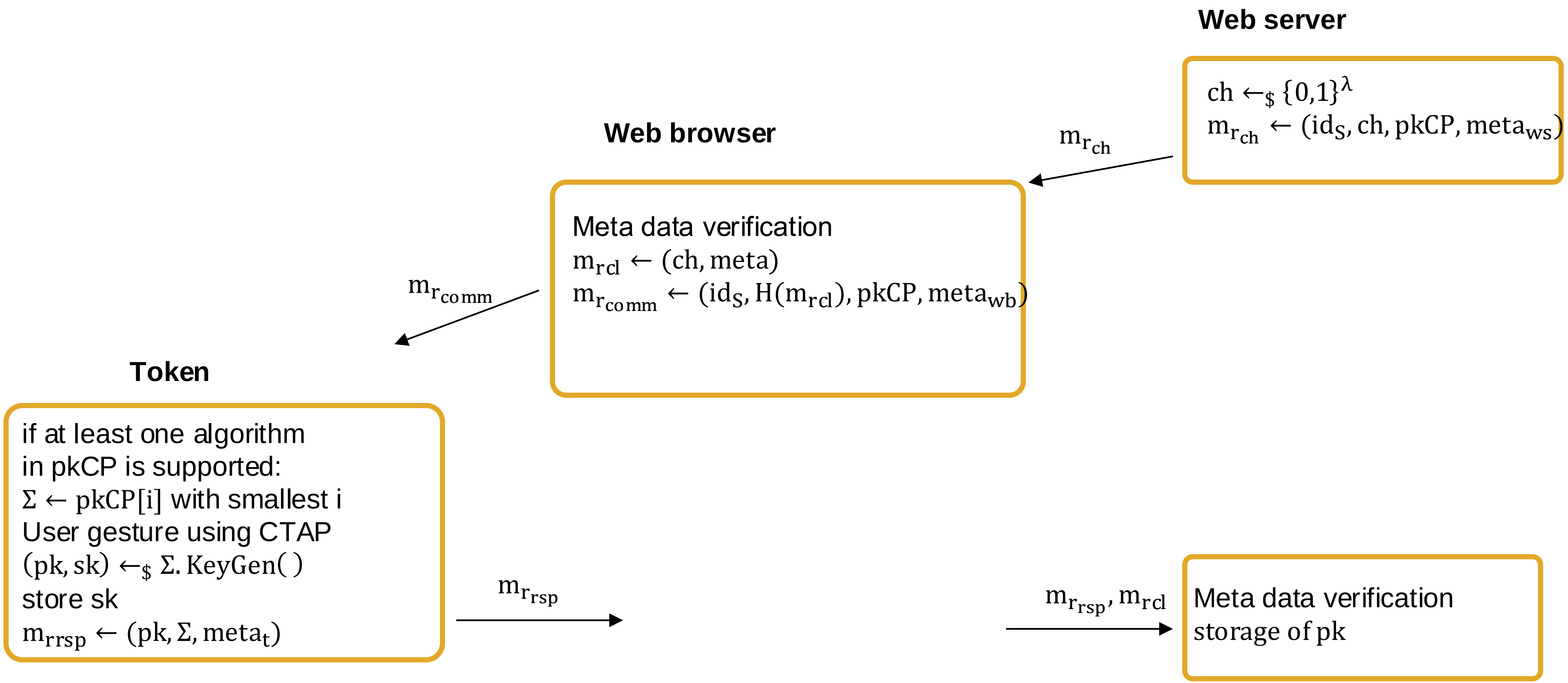
$\xleftarrow{s, c}$

$k \leftarrow K.obtainKey(ek, s)$

$h \leftarrow E_{sy}.Dec(k, c)$

Accept if $h = H(pinStored)$

Cryptographic details of registration flow



Web server

$\mathrm{ch} \leftarrow_{\$} \{0,1\}^{\lambda}$

$\mathrm{m}_{\mathrm{r}_{\mathrm{ch}}} \leftarrow (\mathrm{id}_{\mathrm{S}}, \mathrm{ch}, \mathrm{pkCP}, \mathrm{meta}_{\mathrm{ws}})$

$\mathrm{m}_{\mathrm{r}_{\mathrm{ch}}}$

Web browser

Meta data verification

$\mathrm{m}_{\mathrm{rcl}} \leftarrow (\mathrm{ch}, \mathrm{meta})$

$\mathrm{m}_{\mathrm{r}_{\mathrm{comm}}} \leftarrow (\mathrm{id}_{\mathrm{S}}, \mathrm{H}(\mathrm{m}_{\mathrm{rcl}}), \mathrm{pkCP}, \mathrm{meta}_{\mathrm{wb}})$

$\mathrm{m}_{\mathrm{r}_{\mathrm{comm}}}$

Token

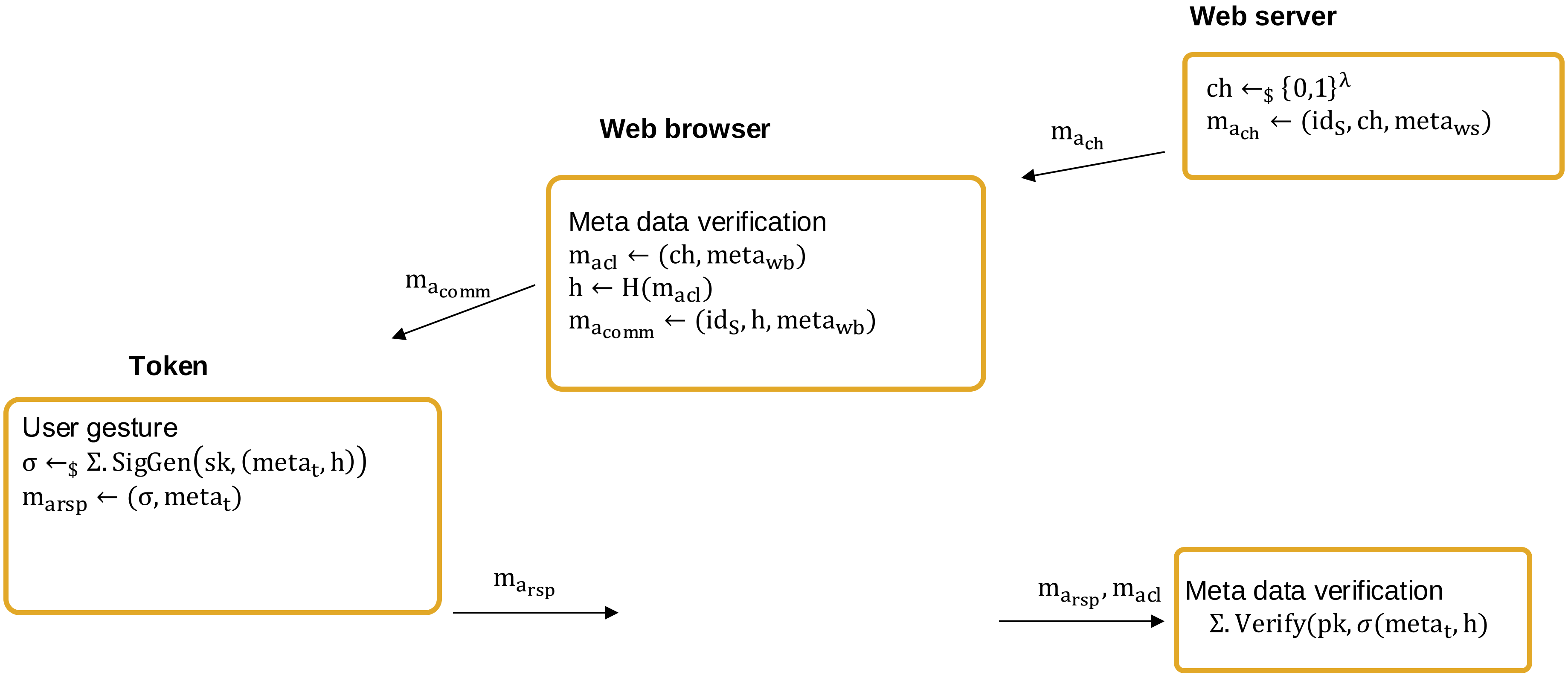
if at least one algorithm
in pkCP is supported:
$\Sigma \leftarrow \mathrm{pkCP}[\mathrm{i}]$ with smallest i
User gesture using CTAP
$(\mathrm{pk}, \mathrm{sk}) \leftarrow_{\$} \Sigma.\mathrm{KeyGen}(\,)$
store sk
$\mathrm{m}_{\mathrm{rrsp}} \leftarrow (\mathrm{pk}, \Sigma, \mathrm{meta}_{\mathrm{t}})$

$\mathrm{m}_{\mathrm{r}_{\mathrm{rsp}}}$

$\mathrm{m}_{\mathrm{r}_{\mathrm{rsp}}}, \mathrm{m}_{\mathrm{rcl}}$

Meta data verification
storage of pk

Cryptographic details of authentication flow



Web server

$ch \leftarrow_{\$} \{0,1\}^{\lambda}$

$m_{a_{ch}} \leftarrow (id_S, ch, meta_{ws})$

$m_{a_{ch}}$

Web browser

Meta data verification

$m_{acl} \leftarrow (ch, meta_{wb})$

$h \leftarrow H(m_{acl})$

$m_{a_{comm}} \leftarrow (id_S, h, meta_{wb})$

$m_{a_{comm}}$

Token

User gesture

$\sigma \leftarrow_{\$} \Sigma.\mathrm{SigGen}\big(sk, (meta_t, h)\big)$

$m_{a_{rsp}} \leftarrow (\sigma, meta_t)$

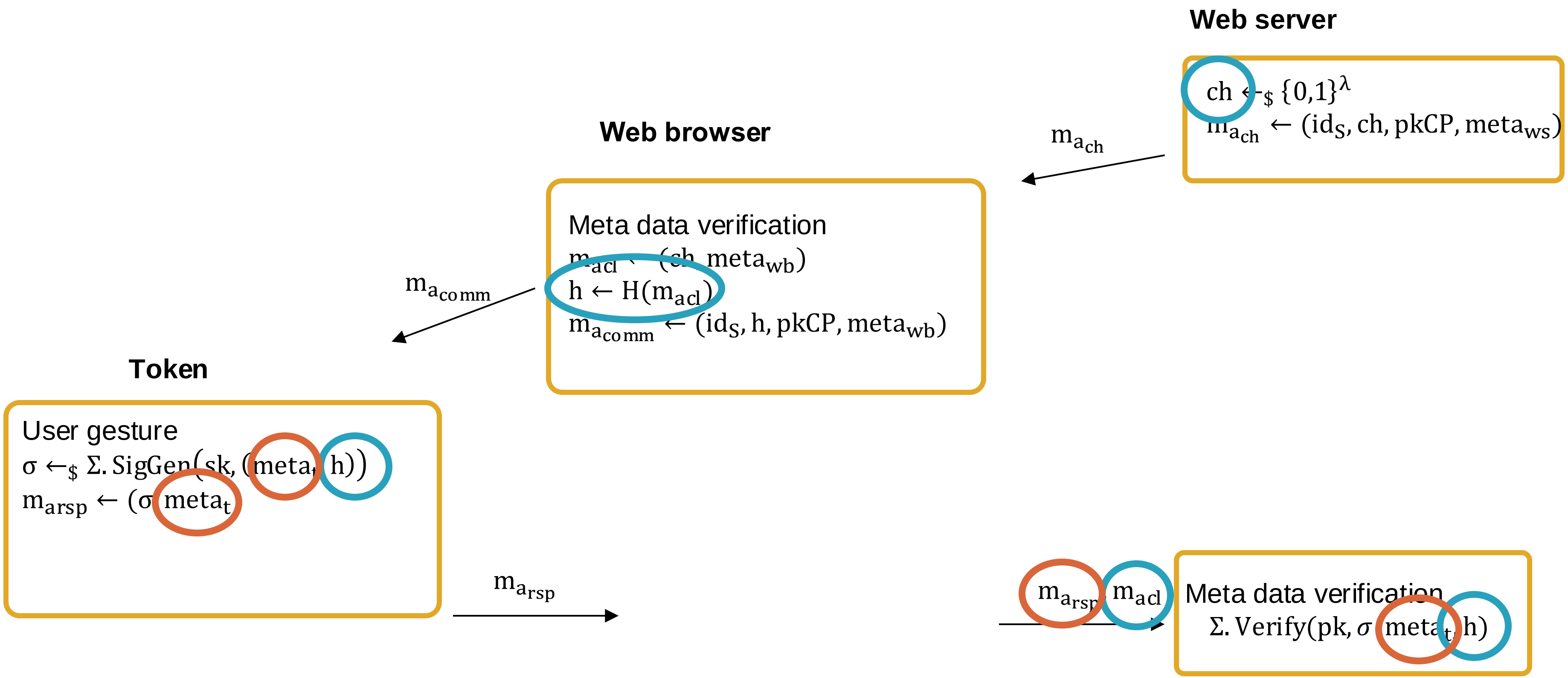
$m_{a_{rsp}}$

$m_{a_{rsp}}, m_{acl}$

Meta data verification

$\Sigma.\mathrm{Verify}(pk, \sigma(meta_t, h)$

Cryptographic details of authentication flow



Web server

$ch \leftarrow_{\$} \{0,1\}^{\lambda}$

$m_{a_{ch}} \leftarrow (id_S, ch, pkCP, meta_{ws})$

$m_{a_{ch}}$

Web browser

Meta data verification

$m_{a_{cl}} \leftarrow (ch, meta_{wb})$

$h \leftarrow H(m_{a_{cl}})$

$m_{a_{comm}} \leftarrow (id_S, h, pkCP, meta_{wb})$

$m_{a_{comm}}$

Token

User gesture

$\sigma \leftarrow_{\$} \Sigma.\mathrm{SigGen}(sk, (meta_t, h))$

$m_{a_{rsp}} \leftarrow (\sigma, meta_t)$

$m_{a_{rsp}}$

$m_{a_{rsp}}, m_{a_{cl}}$

Meta data verification

$\Sigma.\mathrm{Verify}(pk, \sigma, meta_t, h)$

- For more details and a security reduction of FIDO2 see https://eprint.iacr.org/2020/756, https://eprint.iacr.org/2022/1029

FIDO2 protocol options

We have seen so far…

- FIDO2 base protocol (WebAuthn + CTAP) = FIDO2 w/o attestation

Next …

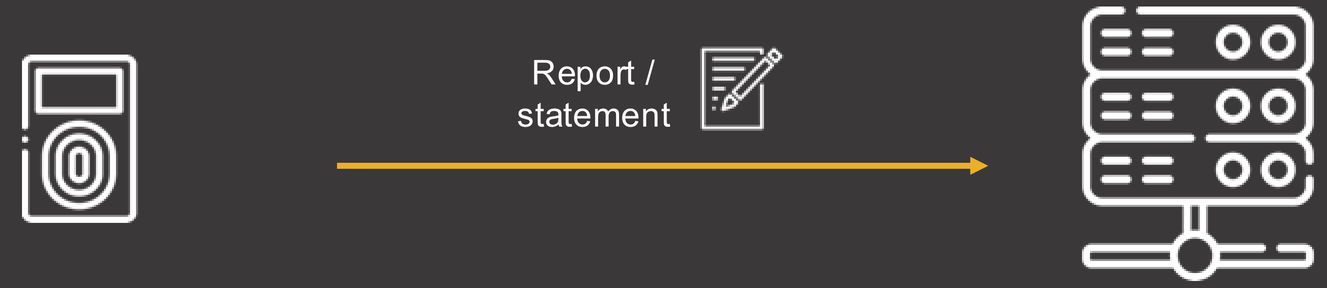
- FIDO2 + attestation

© SandboxAQ

SandboxAQ Proprietary Material

Remote attestation

Basic setup



Report / statement

Host / Client / Device

Challenger / Verifier

Report / statement
software running, boot sequence, hardware specifications, system integrity, device model..

Challenger / verifier
verifies characteristics of the client, based on the attestation report , determines the level of trust on the system, and makes authorization decisions.

The integrity and authenticity of the attestation report is preserved with a **digital signature**

© SandboxAQ

SandboxAQ Proprietary Material

Where is attestation used?

Trusted environments:
hosts, cloud services and virtualisation

V2X communications

DRM rights protection

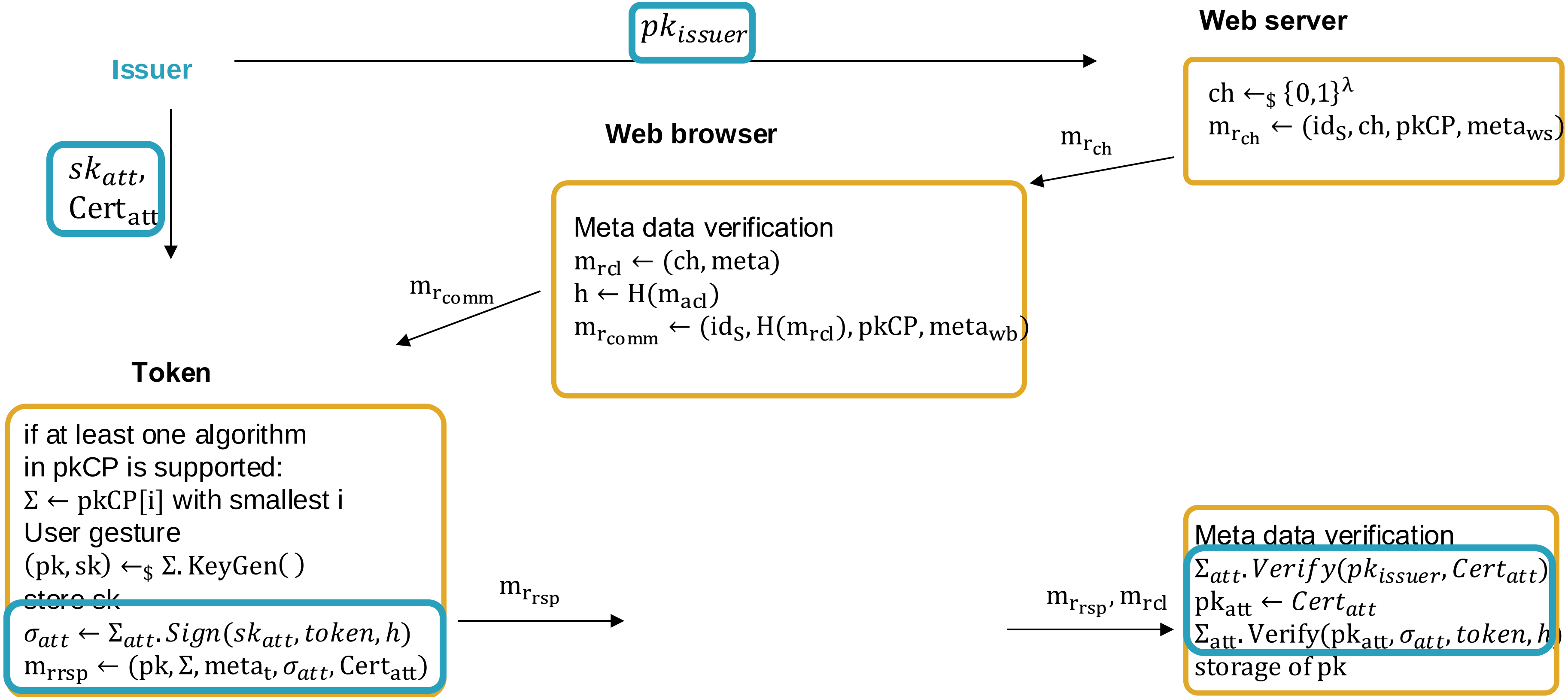
Secure authentication

Hardware security verification

© SandboxAQ

SandboxAQ Proprietary Material

Registration flow w/ attestation



pk_{issuer}
Issuer
Web server
Web browser
Token
sk_{att},
Cert_{att}
$\mathrm{m}_{\mathrm{r}_{\mathrm{ch}}}$
$\mathrm{m}_{\mathrm{r}_{\mathrm{comm}}}$
$\mathrm{m}_{\mathrm{r}_{\mathrm{rsp}}}$
$\mathrm{m}_{\mathrm{r}_{\mathrm{rsp}}}, \mathrm{m}_{\mathrm{rcl}}$

$\mathrm{ch} \leftarrow_{\$} \{0,1\}^{\lambda}$
$\mathrm{m}_{\mathrm{r}_{\mathrm{ch}}} \leftarrow (\mathrm{id}_S, \mathrm{ch}, \mathrm{pkCP}, \mathrm{meta}_{\mathrm{ws}})$

Meta data verification
$\mathrm{m}_{\mathrm{rcl}} \leftarrow (\mathrm{ch}, \mathrm{meta})$
$\mathrm{h} \leftarrow \mathrm{H}(\mathrm{m}_{\mathrm{acl}})$
$\mathrm{m}_{\mathrm{r}_{\mathrm{comm}}} \leftarrow (\mathrm{id}_S, \mathrm{H}(\mathrm{m}_{\mathrm{rcl}}), \mathrm{pkCP}, \mathrm{meta}_{\mathrm{wb}})$

if at least one algorithm
in pkCP is supported:
$\Sigma \leftarrow \mathrm{pkCP}[\mathrm{i}]$ with smallest i
User gesture
$(\mathrm{pk}, \mathrm{sk}) \leftarrow_{\$} \Sigma.\mathrm{KeyGen}(\,)$
store sk
$\sigma_{att} \leftarrow \Sigma_{att}.Sign(sk_{att}, token, h)$
$\mathrm{m}_{\mathrm{rrsp}} \leftarrow (\mathrm{pk}, \Sigma, \mathrm{meta}_{\mathrm{t}}, \sigma_{att}, \mathrm{Cert}_{\mathrm{att}})$

Meta data verification
$\Sigma_{att}.Verify(pk_{issuer}, Cert_{att})$
$\mathrm{pk}_{\mathrm{att}} \leftarrow Cert_{att}$
$\Sigma_{\mathrm{att}}.\mathrm{Verify}(\mathrm{pk}_{\mathrm{att}}, \sigma_{att}, token, h)$
storage of pk

For more details on different attestation modes in FIDO2 and an analysis of their privacy and security guarantees, see https://eprint.iacr.org/2022/084, https://eprint.iacr.org/2023/1398

FIDO2 protocol options

We have seen so far…

- FIDO2 base protocol (WebAuthn + CTAP) = FIDO2 w/o attestation
- FIDO2 + attestation

Next …

- FIDO2 with different credential storing options
  - non-residential / non-discoverable
  - residential / discoverable
    - Passkeys

FIDO2 discoverable vs non-discoverable credentials

Discoverable (residential) credentials

Private keys are stored in the token.

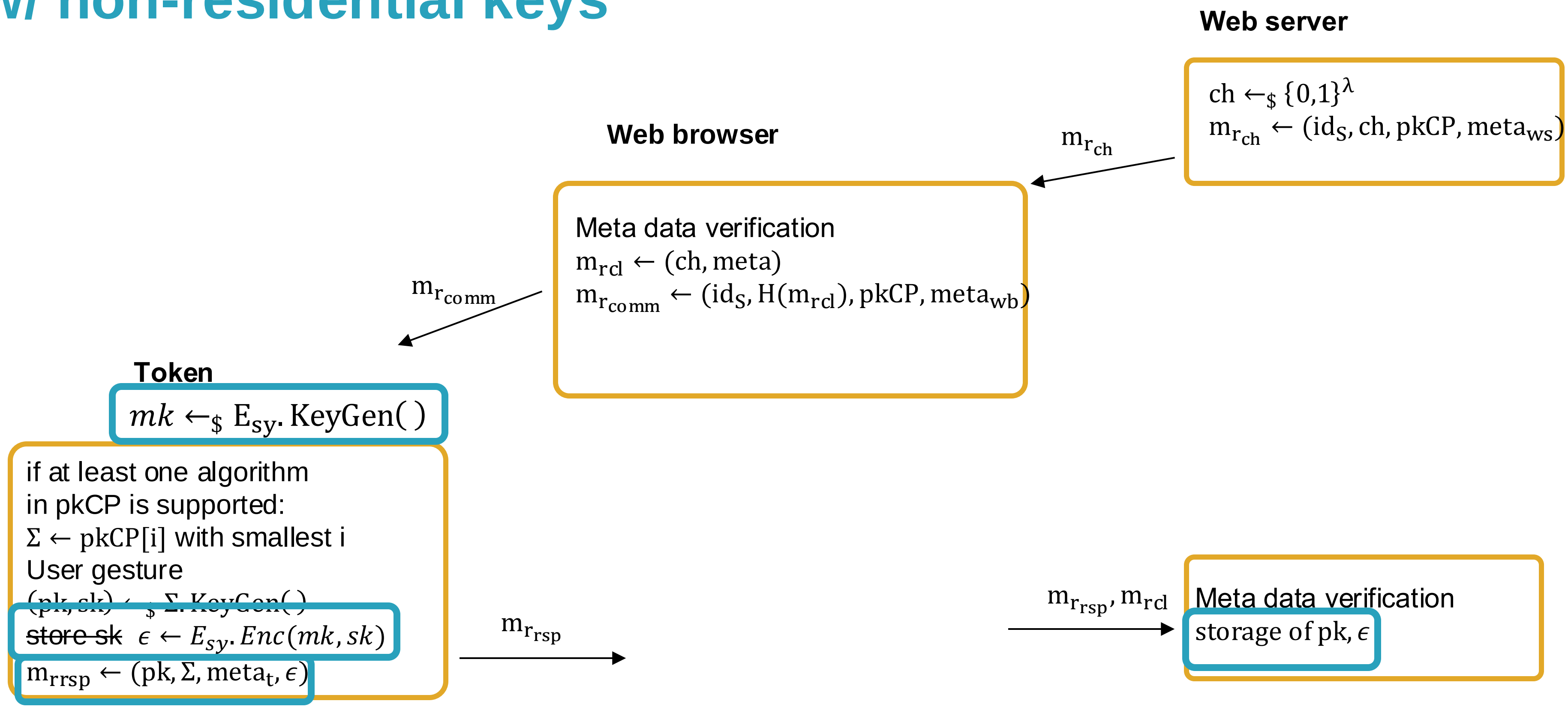
# of servers to register with is limited by token storage space.

Non-discoverable (non-residential) credentials

Private keys are stored in the remote servers, encrypted with a token master key.

# of servers to register with is potentially unlimited.

Cryptographic details of registration flow
w/ non-residential keys



Web server

$\mathrm{ch} \leftarrow_{\$} \{0,1\}^{\lambda}$
$\mathrm{m}_{\mathrm{r_{ch}}} \leftarrow (\mathrm{id_S}, \mathrm{ch}, \mathrm{pkCP}, \mathrm{meta_{ws}})$

$\mathrm{m}_{\mathrm{r_{ch}}}$

Web browser

Meta data verification
$\mathrm{m_{rcl}} \leftarrow (\mathrm{ch}, \mathrm{meta})$
$\mathrm{m}_{\mathrm{r_{comm}}} \leftarrow (\mathrm{id_S}, \mathrm{H}(\mathrm{m_{rcl}}), \mathrm{pkCP}, \mathrm{meta_{wb}})$

$\mathrm{m}_{\mathrm{r_{comm}}}$

Token

$mk \leftarrow_{\$} \mathrm{E_{sy}}.\mathrm{KeyGen}(\,)$

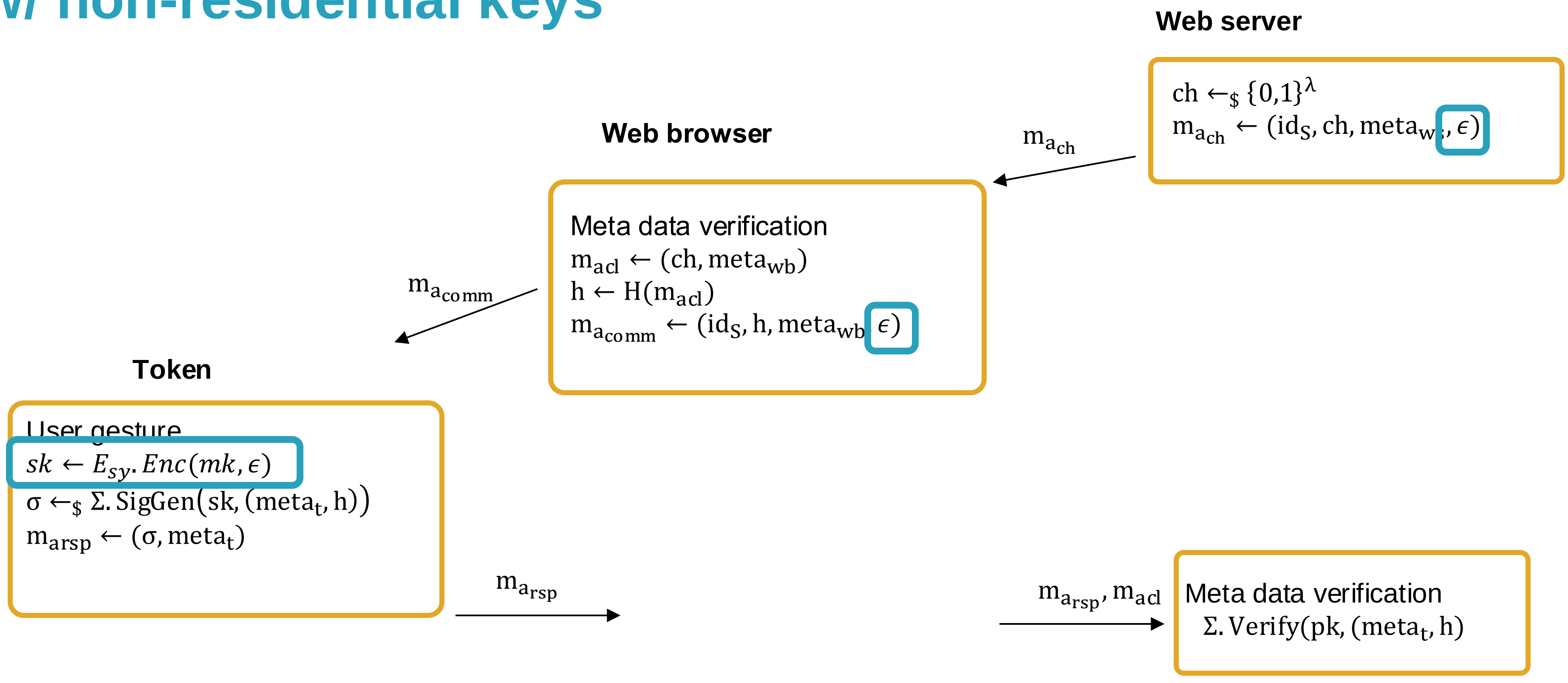
if at least one algorithm
in pkCP is supported:
$\Sigma \leftarrow \mathrm{pkCP[i]}$ with smallest i
User gesture
$(\mathrm{pk}, \mathrm{sk}) \leftarrow_{\$} \Sigma.\mathrm{KeyGen}(\,)$
~~store sk~~ $\epsilon \leftarrow E_{sy}.Enc(mk, sk)$
$\mathrm{m_{rrsp}} \leftarrow (\mathrm{pk}, \Sigma, \mathrm{meta_t}, \epsilon)$

$\mathrm{m}_{\mathrm{r_{rsp}}}$

$\mathrm{m}_{\mathrm{r_{rsp}}}, \mathrm{m_{rcl}}$

Meta data verification
storage of pk, ϵ

Cryptographic details of authentication flow
w/ non-residential keys



Web server

$\text{ch} \leftarrow_{\$} \{0,1\}^{\lambda}$

$\text{m}_{\text{a}_{\text{ch}}} \leftarrow (\text{id}_{\text{S}}, \text{ch}, \text{meta}_{\text{ws}}, \boxed{, \epsilon)}$

$\text{m}_{\text{a}_{\text{ch}}}$

Web browser

Meta data verification

$\text{m}_{\text{acl}} \leftarrow (\text{ch}, \text{meta}_{\text{wb}})$

$\text{h} \leftarrow \text{H}(\text{m}_{\text{acl}})$

$\text{m}_{\text{a}_{\text{comm}}} \leftarrow (\text{id}_{\text{S}}, \text{h}, \text{meta}_{\text{wb}} \boxed{\epsilon)}$

$\text{m}_{\text{a}_{\text{comm}}}$

Token

User gesture

$\boxed{sk \leftarrow E_{sy}.Enc(mk, \epsilon)}$

$\sigma \leftarrow_{\$} \Sigma.\text{SigGen}(\text{sk}, (\text{meta}_{\text{t}}, \text{h}))$

$\text{m}_{\text{arsp}} \leftarrow (\sigma, \text{meta}_{\text{t}})$

$\text{m}_{\text{a}_{\text{rsp}}}$

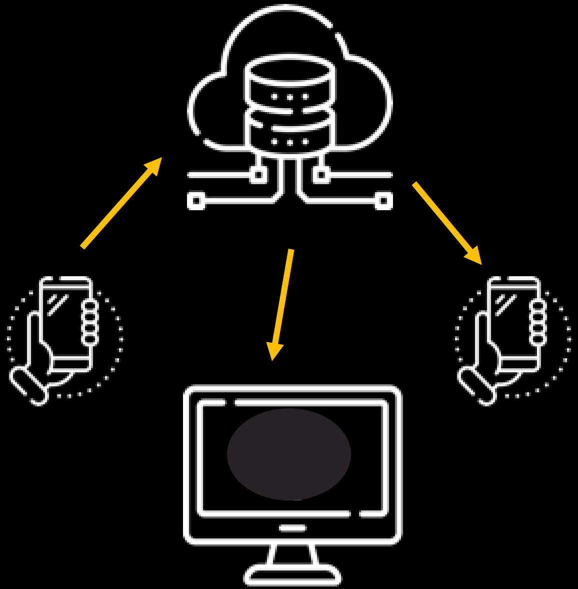
$\text{m}_{\text{a}_{\text{rsp}}}, \text{m}_{\text{acl}}$

Meta data verification

$\Sigma.\text{Verify}(\text{pk}, (\text{meta}_{\text{t}}, \text{h}))$

Passkeys = FIDO2 + credential synchronisation

- They are **discoverable / resident** FIDO credentials
- **Cross-device authentication**
- **Synced passkeys:**
  - E2E across all devices on Passkey provider
  - Since very recently, proposal to sync over different platform possible
- Hardware-bound credentials can still be enforced for critical applications
- Attestation can become crucial to understand how a credential is managed



FIDO2 protocol options

We have seen so far…

- FIDO2 base protocol (WebAuthn + CTAP) = FIDO2 w/o attestation
- FIDO2 + attestation
- FIDO2 with different credential storing options
  - non-residential / non-discoverable
  - residential / discoverable
    - Passkeys

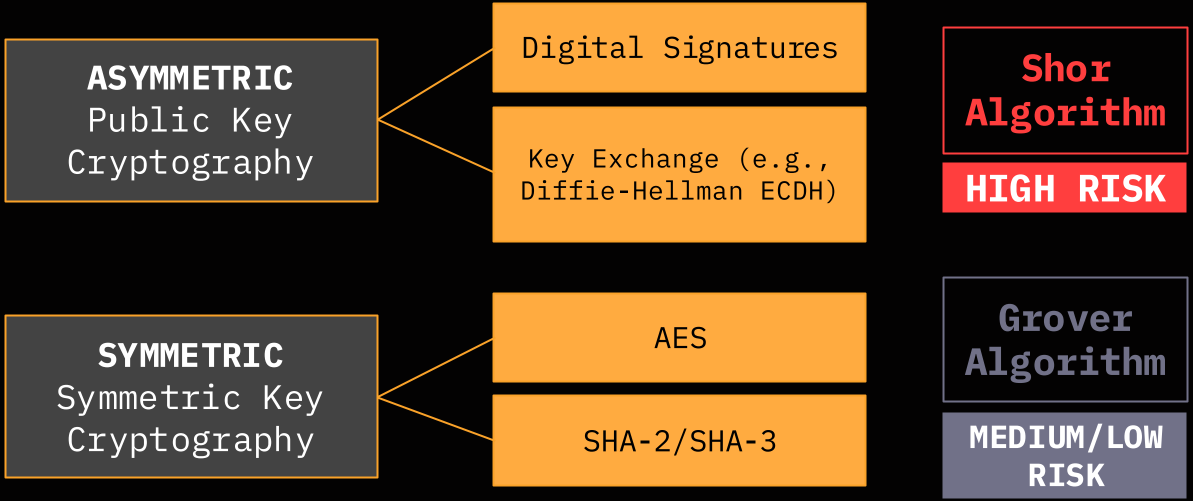
Next …

- Is FIDO2 Post-Quantum secure?

03

PQ FIDO2

Cryptography at risk



ASYMMETRIC
Public Key
Cryptography

Digital Signatures

Key Exchange (e.g., Diffie-Hellman ECDH)

Shor Algorithm

HIGH RISK

SYMMETRIC
Symmetric Key
Cryptography

AES

SHA-2/SHA-3

Grover Algorithm

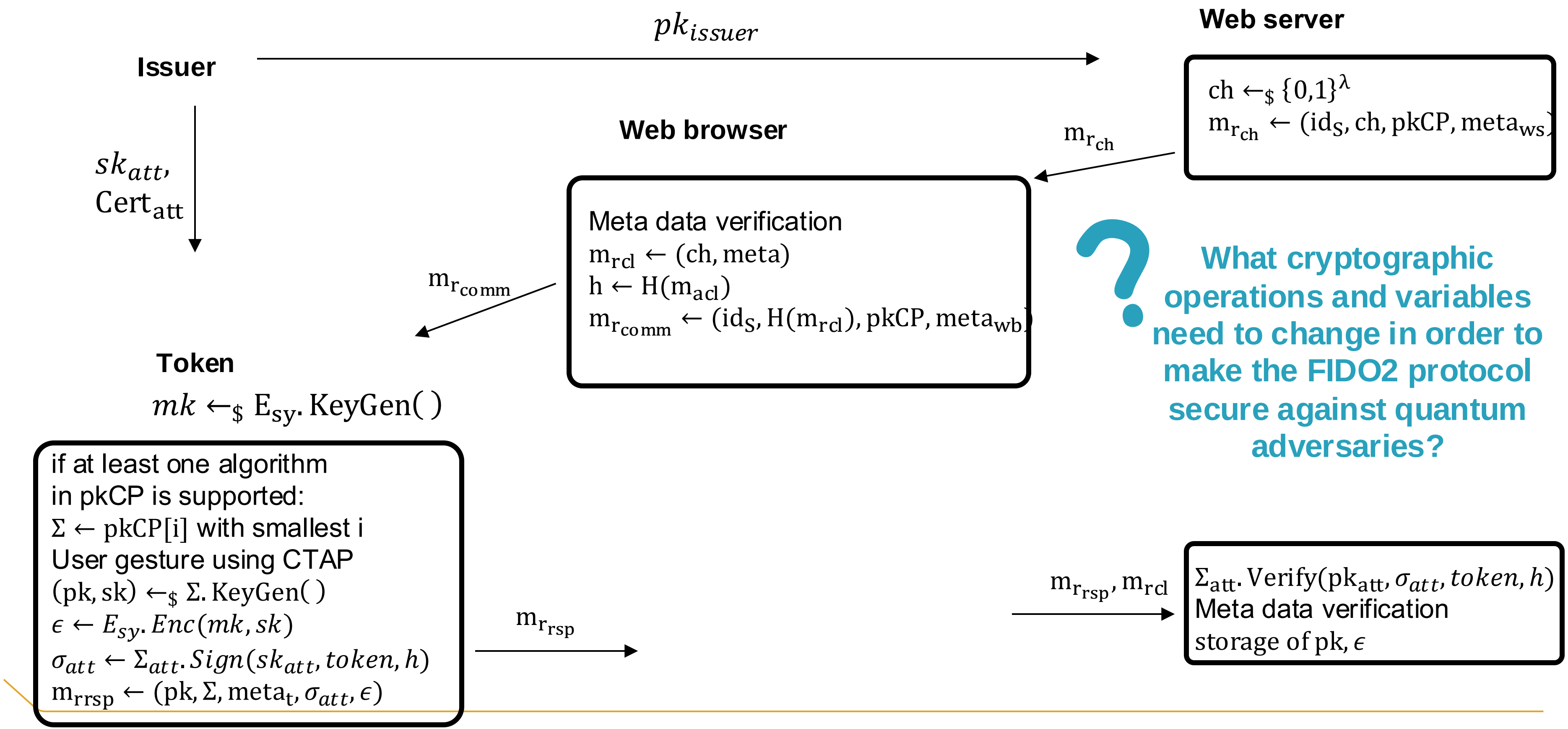
MEDIUM/LOW RISK

~~Double the size of your AES keys and SHA hashes~~

All AES, SHA2 and SHA3 parameters are considered safe

Grover attacks are not realistic against them

Registration flow w/ attestation and non-residential keys



pk_{issuer}

Issuer

sk_{att},
Cert_{att}

Web server

$\mathrm{ch} \leftarrow_{\$} \{0,1\}^{\lambda}$
$\mathrm{m}_{\mathrm{r_{ch}}} \leftarrow (\mathrm{id_S}, \mathrm{ch}, \mathrm{pkCP}, \mathrm{meta_{ws}})$

$\mathrm{m}_{\mathrm{r_{ch}}}$

Web browser

Meta data verification
$\mathrm{m_{rcl}} \leftarrow (\mathrm{ch}, \mathrm{meta})$
$\mathrm{h} \leftarrow \mathrm{H}(\mathrm{m_{acl}})$
$\mathrm{m}_{\mathrm{r_{comm}}} \leftarrow (\mathrm{id_S}, \mathrm{H}(\mathrm{m_{rcl}}), \mathrm{pkCP}, \mathrm{meta_{wb}})$

$\mathrm{m}_{\mathrm{r_{comm}}}$

Token

$mk \leftarrow_{\$} \mathrm{E_{sy}}.\mathrm{KeyGen}(\,)$

if at least one algorithm
in pkCP is supported:
$\Sigma \leftarrow \mathrm{pkCP[i]}$ with smallest i
User gesture using CTAP
$(\mathrm{pk}, \mathrm{sk}) \leftarrow_{\$} \Sigma.\mathrm{KeyGen}(\,)$
$\epsilon \leftarrow E_{sy}.Enc(mk, sk)$
$\sigma_{att} \leftarrow \Sigma_{att}.Sign(sk_{att}, token, h)$
$\mathrm{m_{rrsp}} \leftarrow (\mathrm{pk}, \Sigma, \mathrm{meta_t}, \sigma_{att}, \epsilon)$

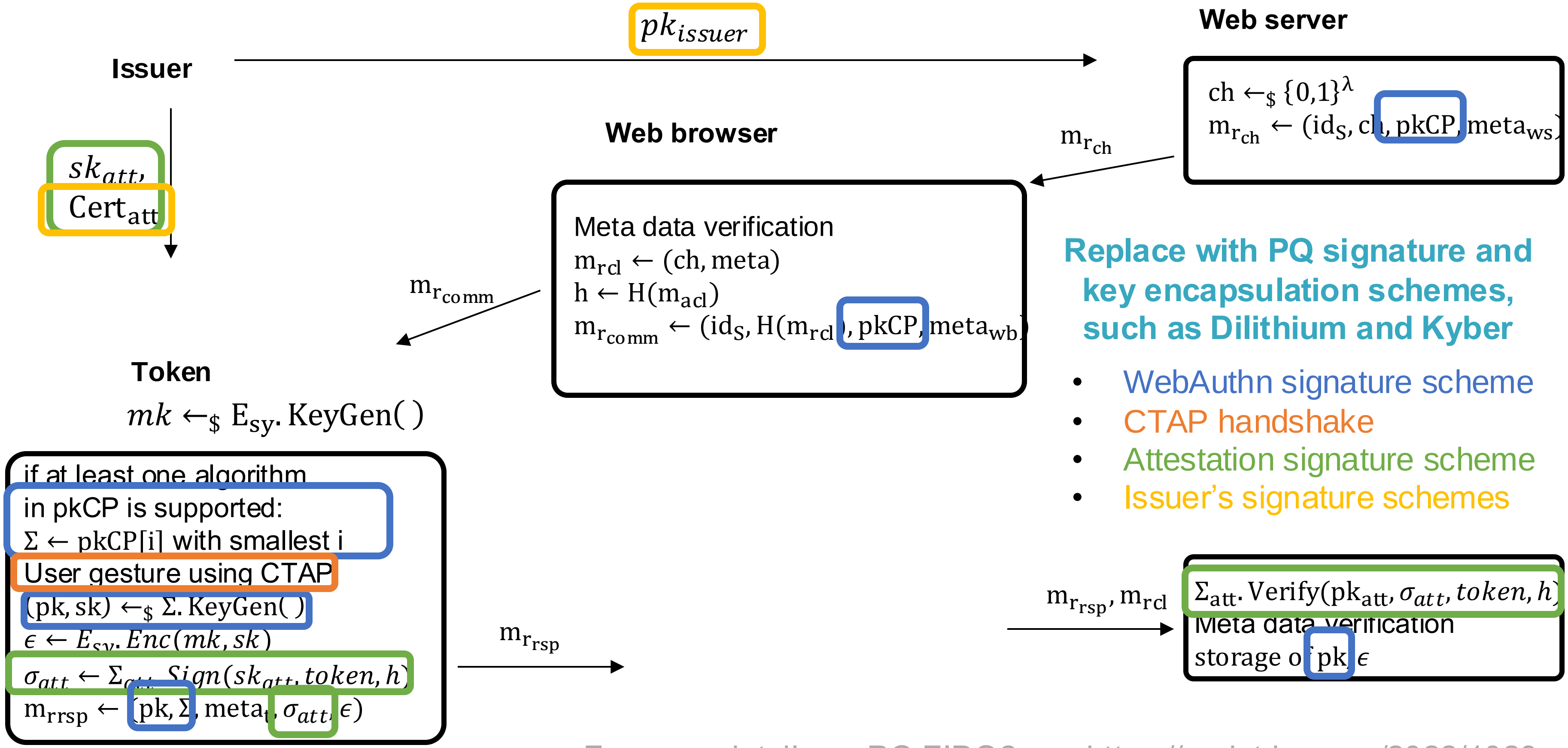
$\mathrm{m}_{\mathrm{r_{rsp}}}$

$\mathrm{m}_{\mathrm{r_{rsp}}}, \mathrm{m_{rcl}}$

$\Sigma_{\mathrm{att}}.\mathrm{Verify}(\mathrm{pk_{att}}, \sigma_{att}, token, h)$
Meta data verification
storage of pk, ϵ

What cryptographic operations and variables need to change in order to make the FIDO2 protocol secure against quantum adversaries?

Registration flow w/ attestation and non-residential keys



pk_{issuer}

Issuer

sk_{att},
Cert_{att}

Web server

$\mathrm{ch} \leftarrow_{\$} \{0,1\}^{\lambda}$
$\mathrm{m}_{r_{ch}} \leftarrow (\mathrm{id}_S, \mathrm{ch}, \mathrm{pkCP}, \mathrm{meta}_{ws})$

$\mathrm{m}_{r_{ch}}$

Web browser

Meta data verification
$\mathrm{m}_{rcl} \leftarrow (\mathrm{ch}, \mathrm{meta})$
$\mathrm{h} \leftarrow \mathrm{H}(\mathrm{m}_{acl})$
$\mathrm{m}_{r_{comm}} \leftarrow (\mathrm{id}_S, \mathrm{H}(\mathrm{m}_{rcl}), \mathrm{pkCP}, \mathrm{meta}_{wb})$

$\mathrm{m}_{r_{comm}}$

Token

$mk \leftarrow_{\$} \mathrm{E}_{sy}.\mathrm{KeyGen}(\,)$

if at least one algorithm
in pkCP is supported:
$\Sigma \leftarrow \mathrm{pkCP}[i]$ with smallest i
User gesture using CTAP
$(\mathrm{pk}, \mathrm{sk}) \leftarrow_{\$} \Sigma.\mathrm{KeyGen}(\,)$
$\epsilon \leftarrow E_{sy}.Enc(mk, sk)$
$\sigma_{att} \leftarrow \Sigma_{att}.Sign(sk_{att}, token, h)$
$\mathrm{m}_{rrsp} \leftarrow (\mathrm{pk}, \Sigma, \mathrm{meta}_t, \sigma_{att}, \epsilon)$

$\mathrm{m}_{r_{rsp}}$

$\mathrm{m}_{r_{rsp}}, \mathrm{m}_{rcl}$

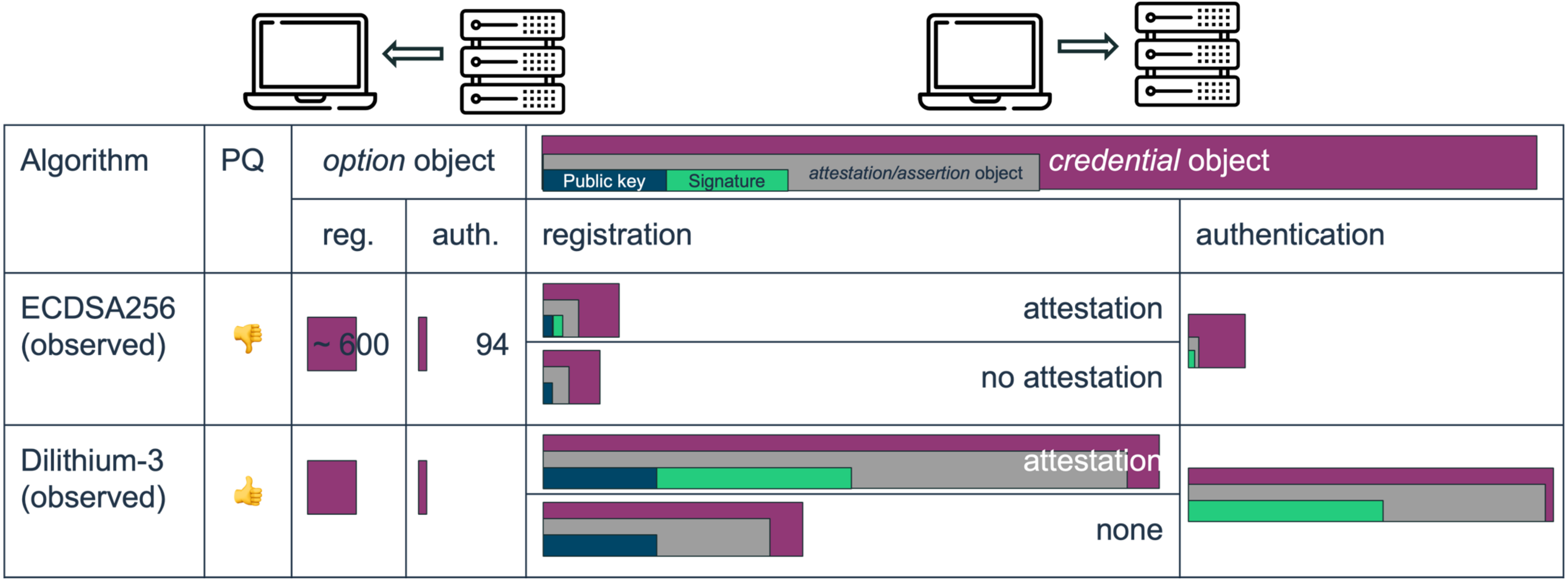
$\Sigma_{att}.\mathrm{Verify}(\mathrm{pk}_{att}, \sigma_{att}, token, h)$
Meta data verification
storage of pk, ϵ

Replace with PQ signature and key encapsulation schemes, such as Dilithium and Kyber

- WebAuthn signature scheme
- CTAP handshake
- Attestation signature scheme
- Issuer's signature schemes

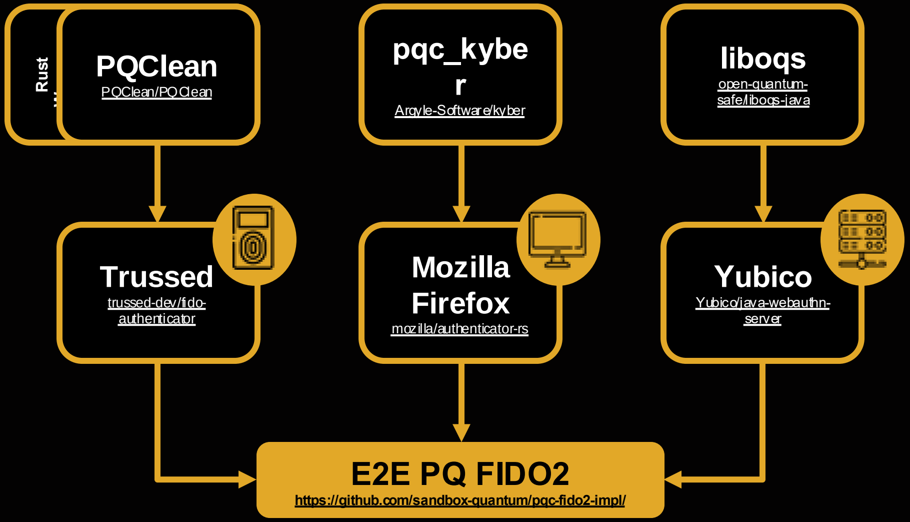
For more details on PQ FIDO2 see https://eprint.iacr.org/2022/1029

Object sizes (w/ CTAP)



Algorithm	PQ	*option* object		*credential* object: Public key, Signature, *attestation/assertion* object		
		reg.	auth.	registration		authentication
ECDSA256 (observed)	👎	~ 600	94		attestation	
					no attestation	
Dilithium-3 (observed)	👍				attestation	
					none	

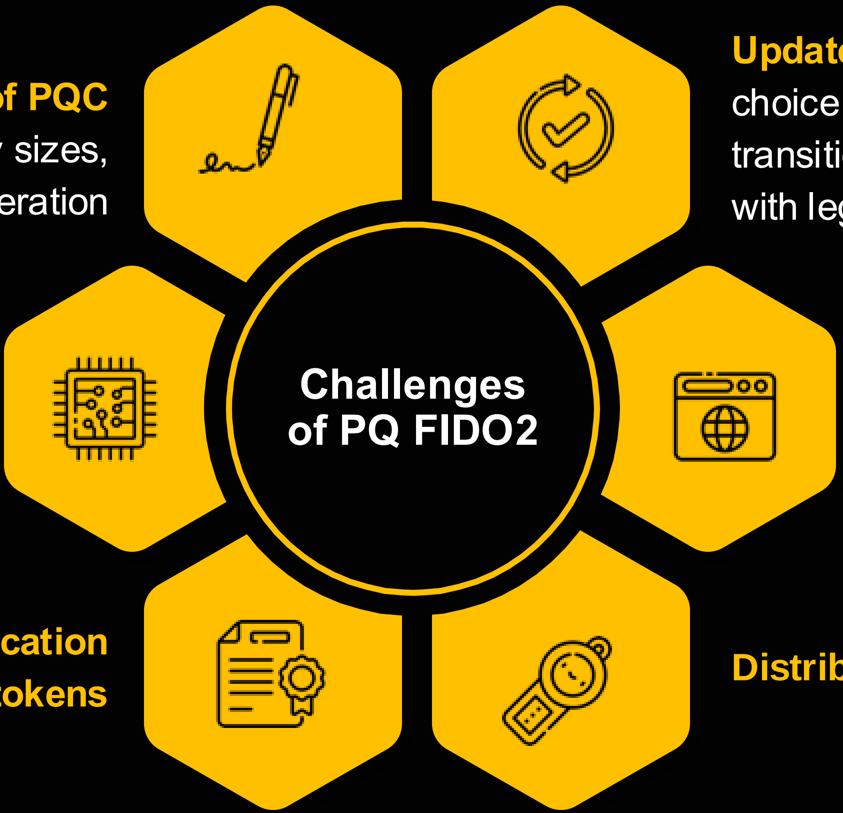
End-to-end open source PQC Fido2 implementation



- E2E PQ FIDO2 OSS using Kyber and Dilithium on Git
- Blog post
- Presentation

04

Challenges to make Passkeys quantum secure



Challenges of PQ FIDO2

Efficiency of PQC
large key sizes,
slower signature generation

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

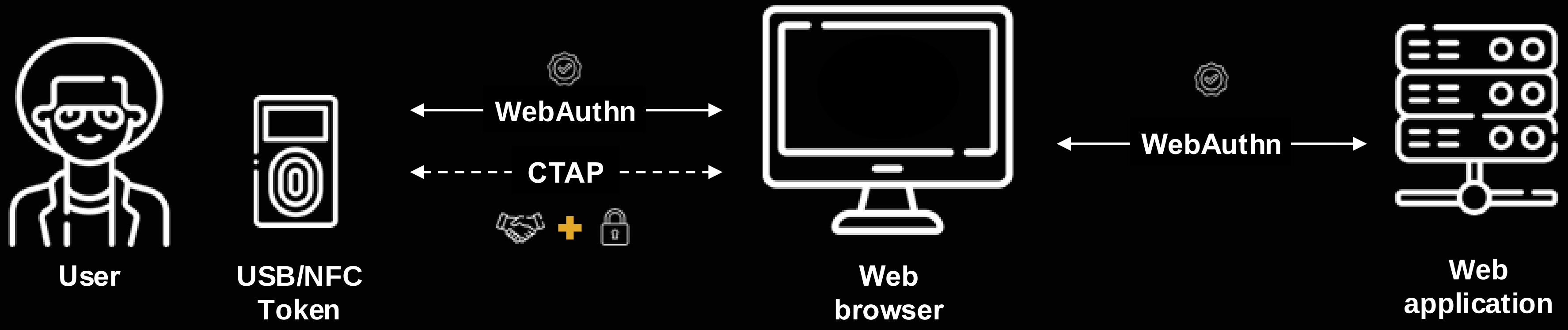
Change of hardware

Update CAs, browsers and web applications

Re-certification of hardware tokens

Distribute new hardware

Summary



- FIDO2/Passkeys protocol alternative to password-based log-in

- Cryptographic protocol flow of registration and authentical (with variants):
  - Digital signature schemes (authentication and attestation)
  - Key agreement (DH key exchange, of Kyber KEM)
  - Hash functions
  - Symmetric encryption

  Update for PQ migration

- Discussion of challenges of PQC migration of the FIDO2 system