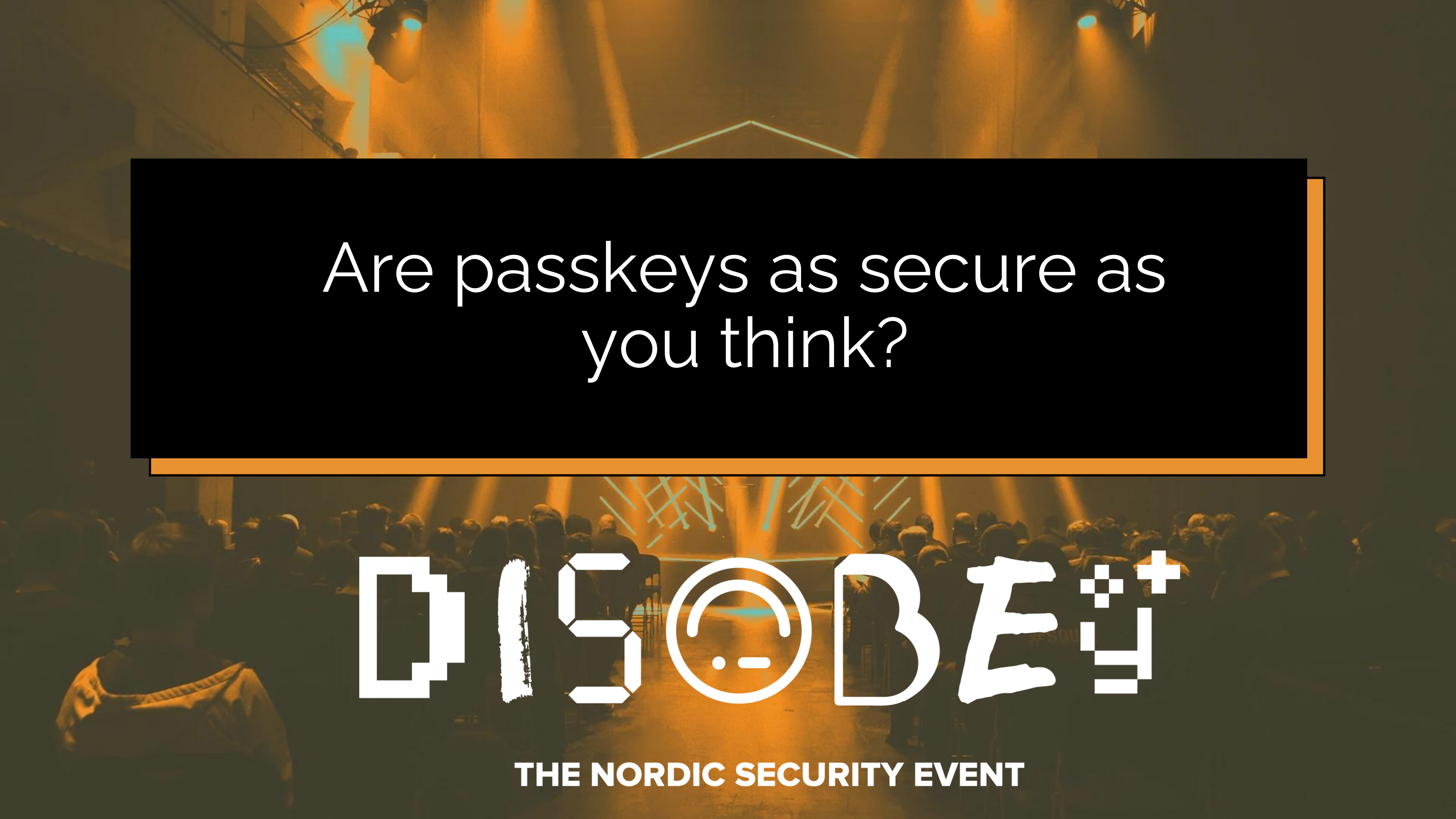
Are passkeys as secure as you think?



Intro – Fabian Bader

- Lives in Germany
- Cyber Security Architect @ glueck kanja
- Microsoft MVP (Security / Azure)
- Organizer of "Purple Elbe Security User Group"
- Author of
  - TokenTacticsV2
  - entrascopes.com
  - XDRInternals
  - SentinelARConverter

Socials

Blog/talks:	cloudbrothers.info
Twitter:	@fabian_bader
BlueSky:	@fabian.bader.cloud

glueck kanja

Talk Agenda

- What is a passkey?
- Synced vs. Device-bound passkeys
- Security of passkey providers
- Why does attestation matter?
- Threat modeling for enterprises & Currently known attacks
- Attack mitigation

DISOBEY
THE NORDIC SECURITY EVENT

glueck kanja

What is a passkey?

glueck kanja

What is a passkey?

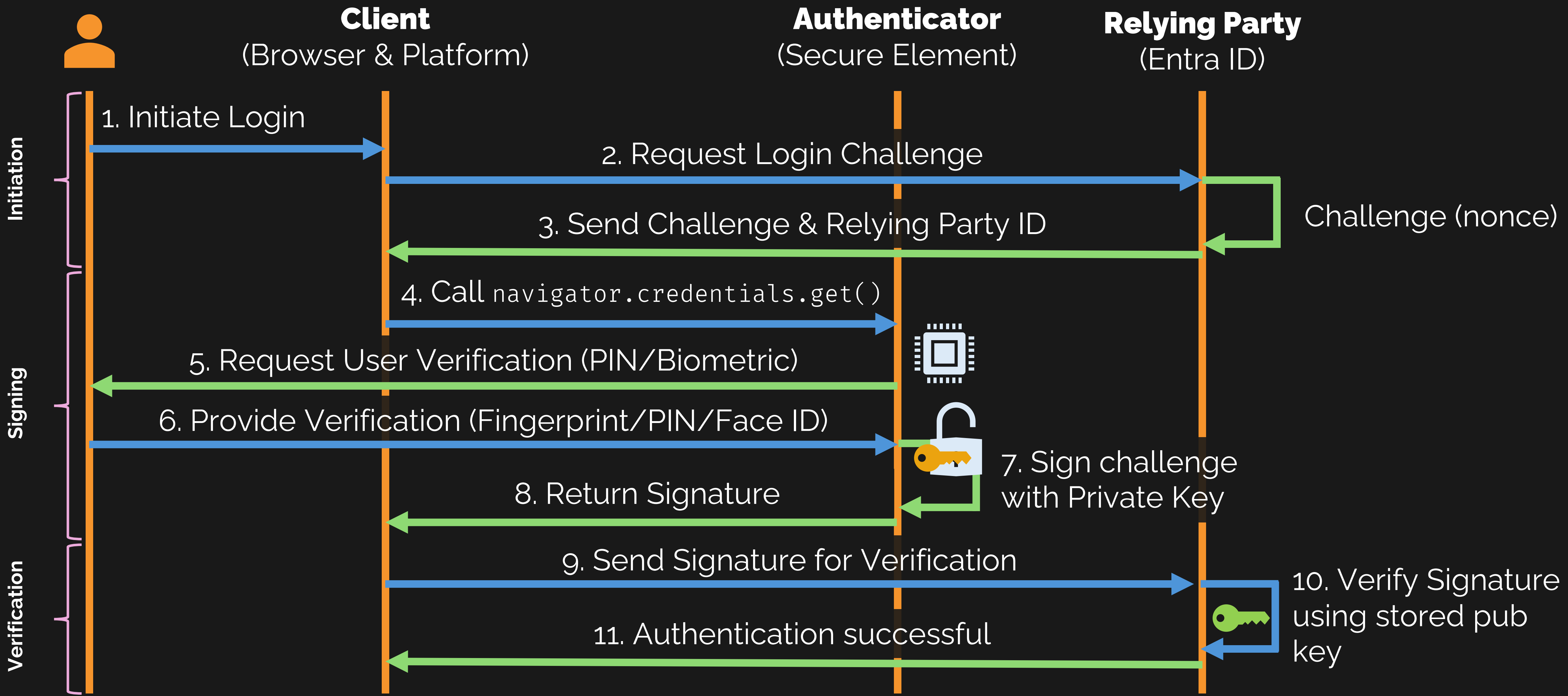
- FIDO2/WebAuthn Discoverable Credential
  - DC: Stores Relying Party Id & User ID

replyingPartyId	Username
login.microsoft.com	maija@c4a8korriban.com

- Based on a cryptographic foundation
  - Private key stored in the authenticator
  - Public key stored by the relying party
- Enforced origin binding

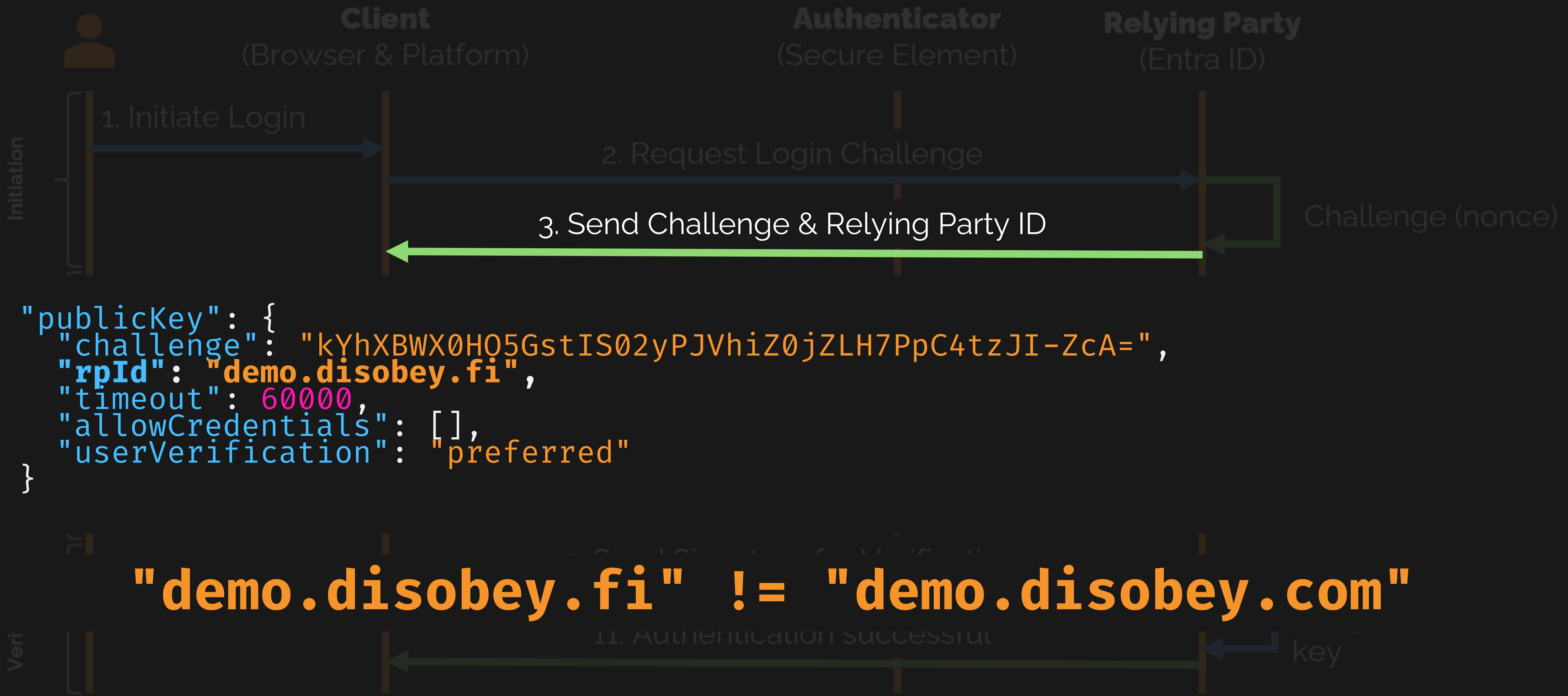
DISOBEY
THE NORDIC SECURITY EVENT

Authentication with a passkey



Client
(Browser & Platform)
Authenticator
(Secure Element)
Relying Party
(Entra ID)
Initiation
Signing
Verification
1. Initiate Login
2. Request Login Challenge
Challenge (nonce)
3. Send Challenge & Relying Party ID
4. Call navigator.credentials.get()
5. Request User Verification (PIN/Biometric)
6. Provide Verification (Fingerprint/PIN/Face ID)
7. Sign challenge with Private Key
8. Return Signature
9. Send Signature for Verification
10. Verify Signature using stored pub key
11. Authentication successful

Why phishing resistant?

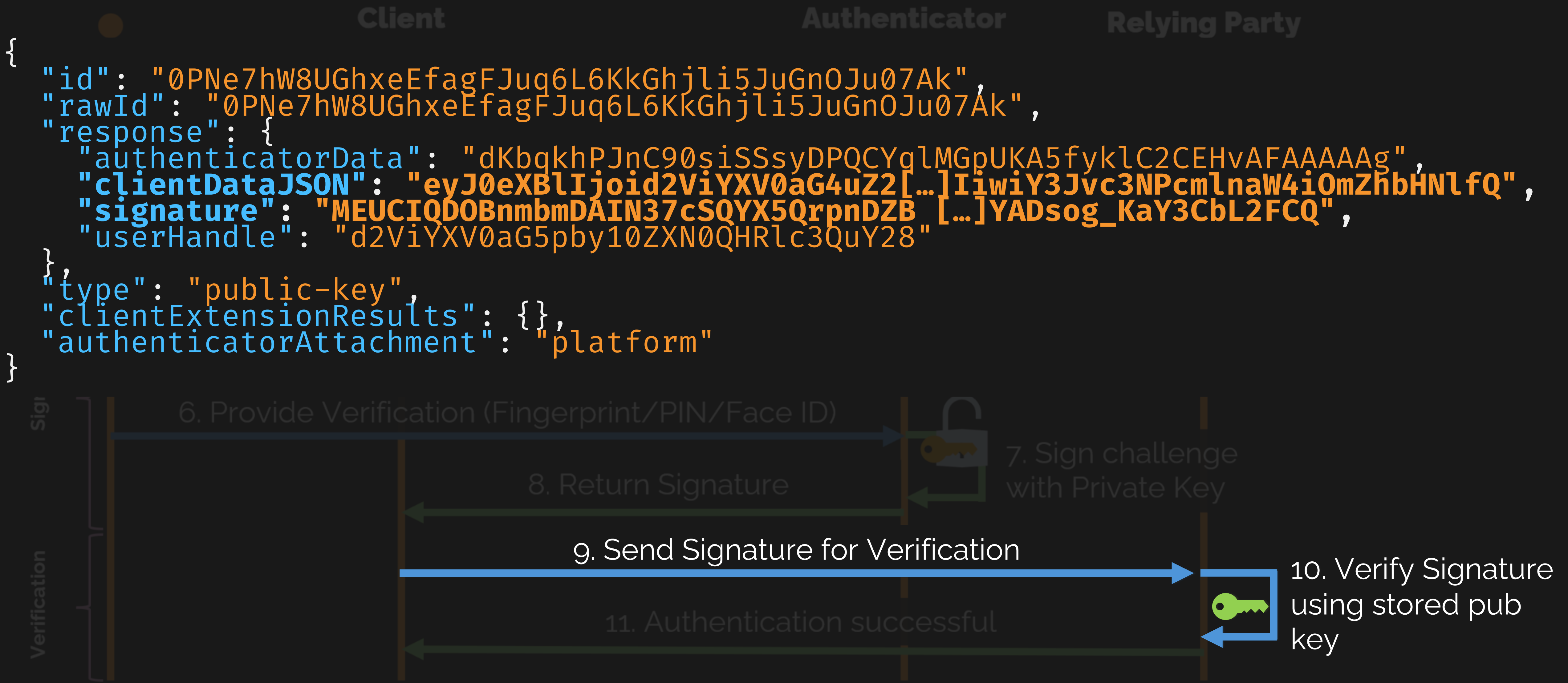


```
{
  "publicKey": {
    "challenge": "kYhXBWX0HO5GstIS02yPJVhiZ0jZLH7PpC4tzJI-ZcA=",
    "rpId": "demo.disobey.fi",
    "timeout": 60000,
    "allowCredentials": [],
    "userVerification": "preferred"
  }
}
```

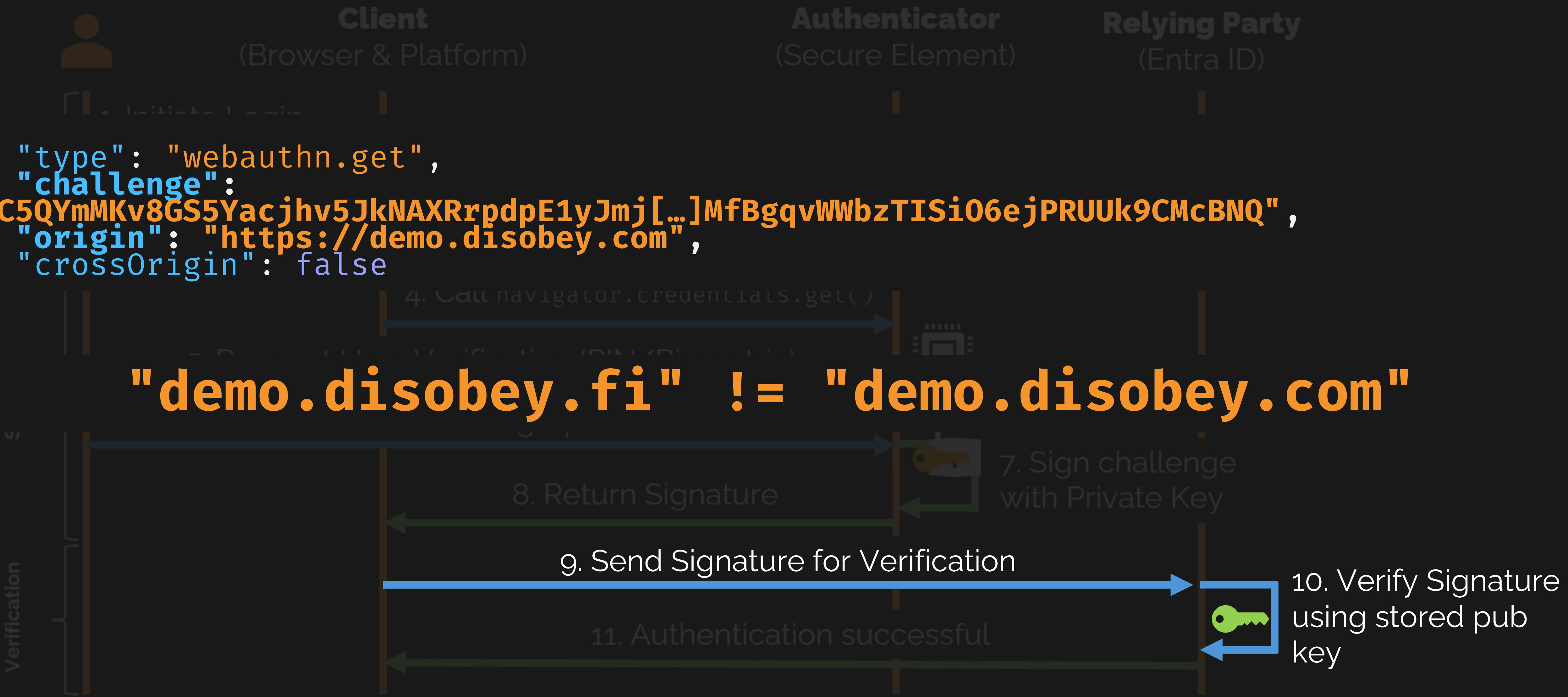
"demo.disobey.fi" != "demo.disobey.com"

Why phishing resistant?

```
{
  "id": "0PNe7hW8UGhxeEfagFJuq6L6KkGhjli5JuGnOJu07Ak",
  "rawId": "0PNe7hW8UGhxeEfagFJuq6L6KkGhjli5JuGnOJu07Ak",
  "response": {
    "authenticatorData": "dKbqkhPJnC90siSSsyDPQCYqlMGpUKA5fyklC2CEHvAFAAAAAg",
    "clientDataJSON": "eyJ0eXBlIjoid2ViYXV0aG4uZ2[…]IiwiY3Jvc3NPcmlnaW4iOmZhbHNlfQ",
    "signature": "MEUCIQDOBnmbmDAIN37cSQYX5QrpnDZB […]YADsog_KaY3CbL2FCQ",
    "userHandle": "d2ViYXV0aG5pby10ZXN0QHRlc3QuY28"
  },
  "type": "public-key",
  "clientExtensionResults": {},
  "authenticatorAttachment": "platform"
}
```


Client
Authenticator
Relying Party
Sign
6. Provide Verification (Fingerprint/PIN/Face ID)
7. Sign challenge with Private Key
8. Return Signature
Verification
9. Send Signature for Verification
10. Verify Signature using stored pub key
11. Authentication successful

Why phishing resistant?


Client
(Browser & Platform)
Authenticator
(Secure Element)
Relying Party
(Entra ID)
4. Call navigator.credentials.get()
7. Sign challenge with Private Key
8. Return Signature
9. Send Signature for Verification
10. Verify Signature using stored pub key
11. Authentication successful
Verification

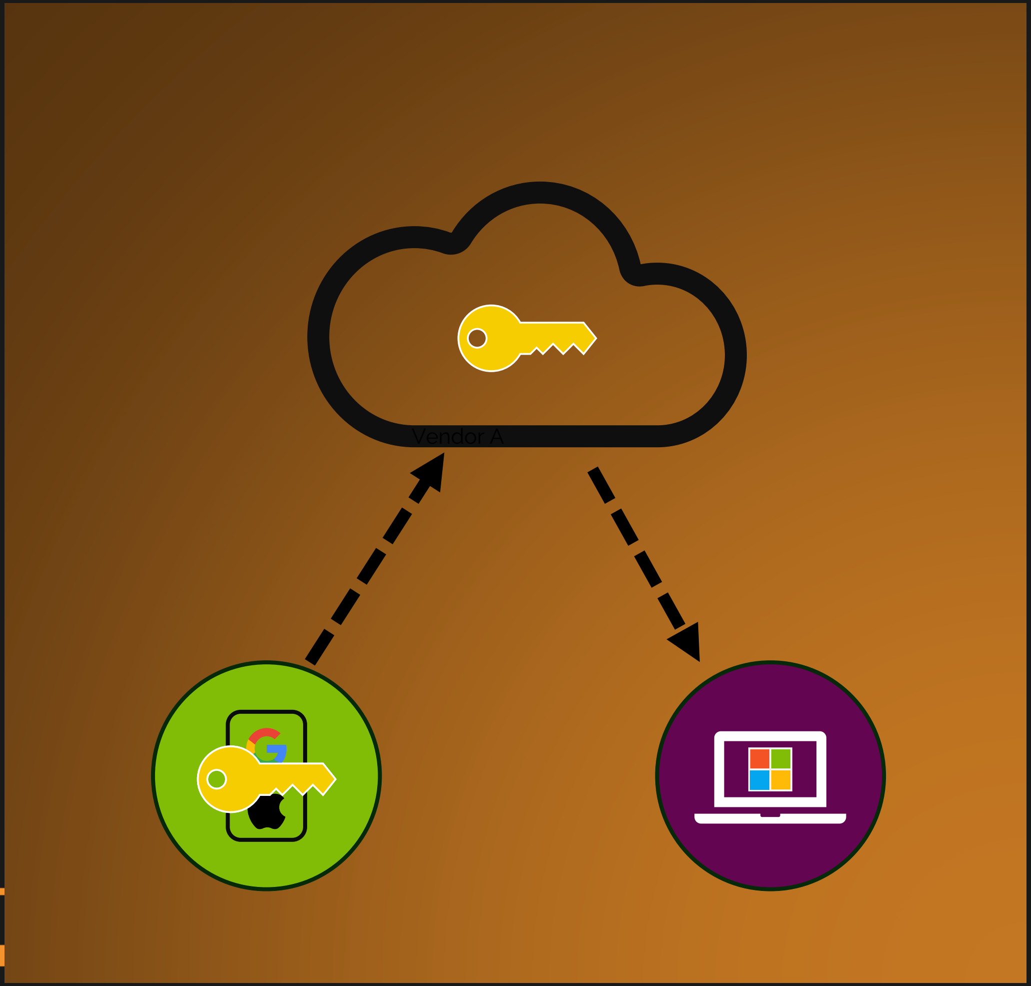
```
{
  "type": "webauthn.get",
  "challenge":
"C5QYmMKv8GS5Yacjhv5JkNAXRrpdpE1yJmj[…]MfBgqvWWbzTISiO6ejPRUUk9CMcBNQ",
  "origin": "https://demo.disobey.com",
  "crossOrigin": false
}
```

"demo.disobey.fi" != "demo.disobey.com"

glueck kanja

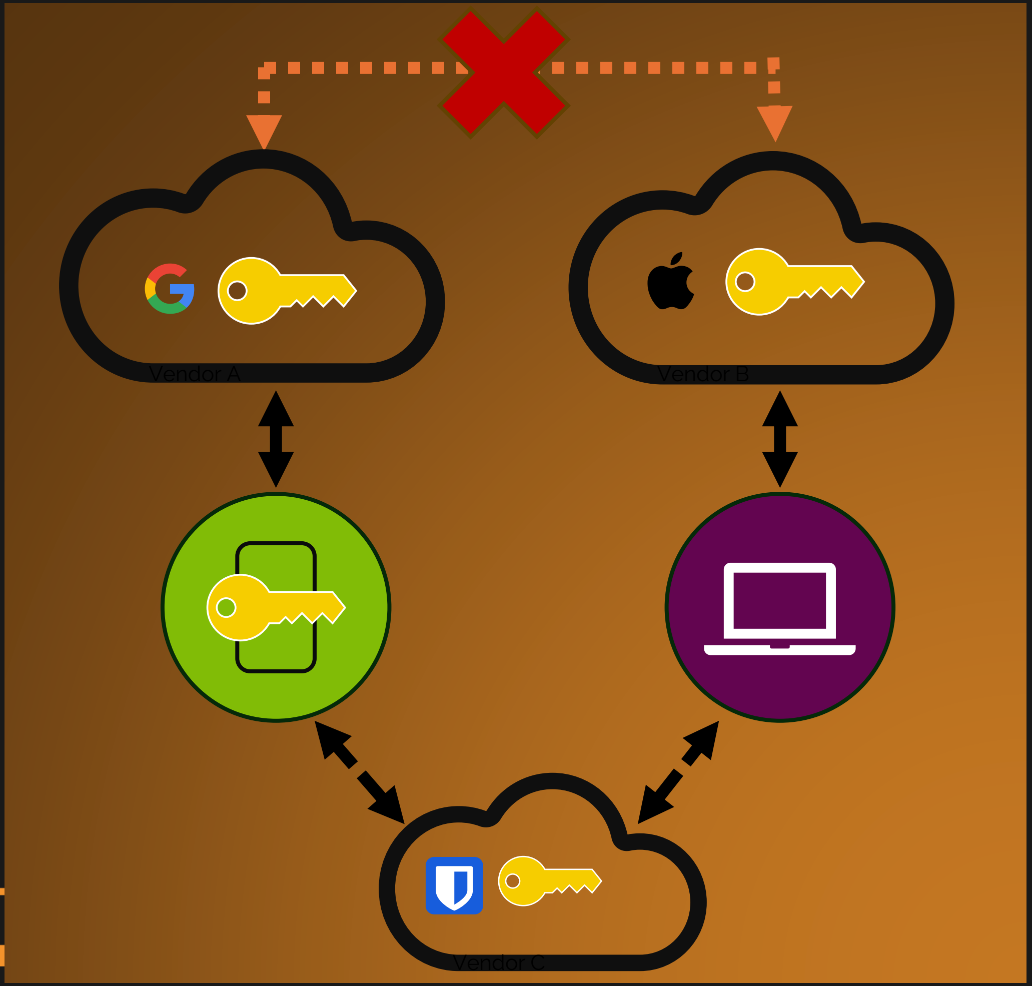
Synced vs. device-bound passkeys

Synced vs. Device-bound passkeys



- Passkeys are synced by **default**
- Private key is sent to mobile device vendor or third-party passkey provider
- Restore and usage on other devices possible

Synced vs. Device-bound passkeys



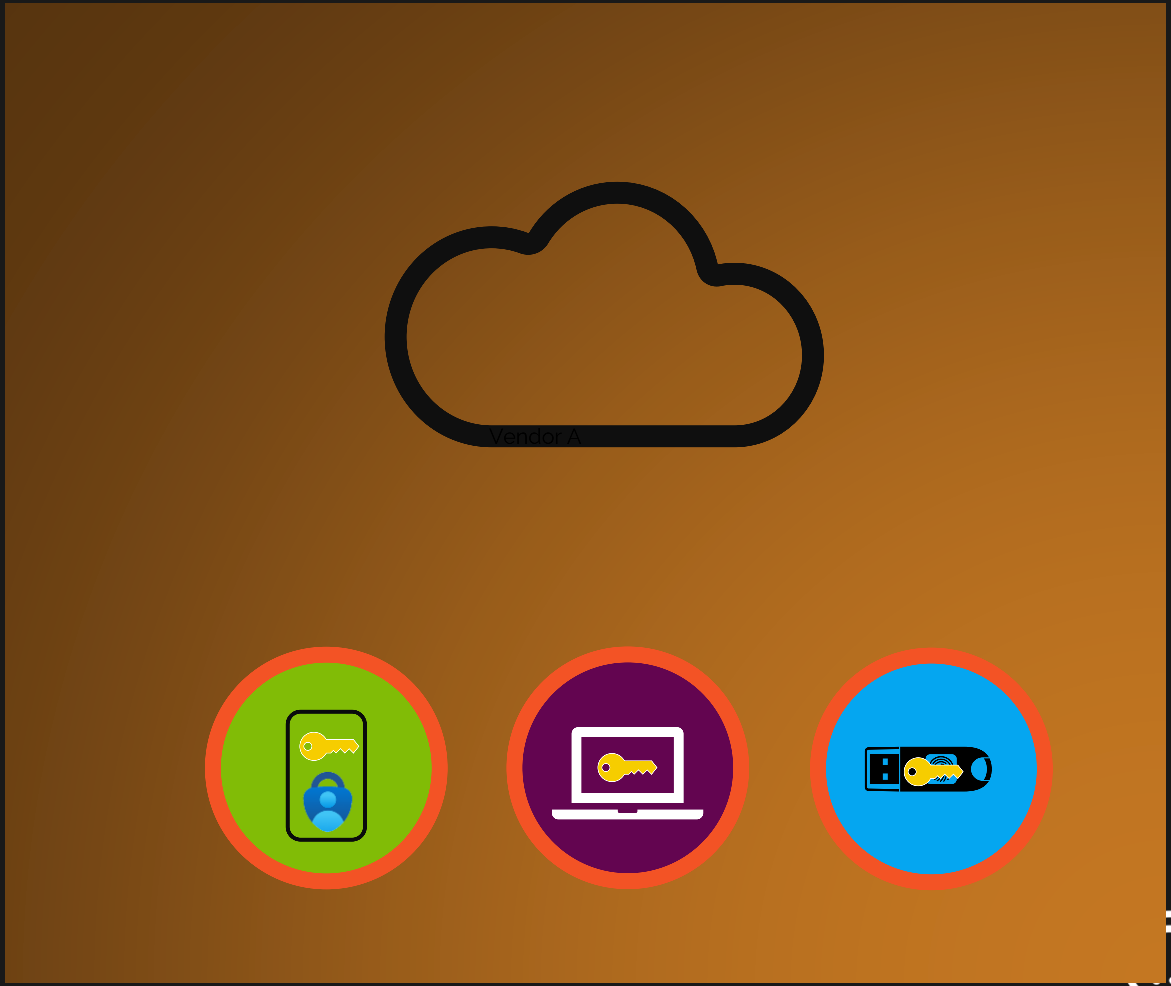
- Native cross vendor sync is not (yet) possible
- Workarounds
  - Cross-Device Authentication
  - Third-party passkey provider
  - Export/Import (e.g. JSON)
- The future
  - Credential Exchange Protocol (CXP)

glueck kanja

Synced vs. Device-bound passkeys

- The private key cannot leave the device
- FIDO2 security keys are device-bound passkeys
- Some apps create a device-bound passkey (e.g. Microsoft Authenticator)

- Recovery = New Setup



DISOBEY
THE NORDIC SECURITY EVENT

glueck kanja

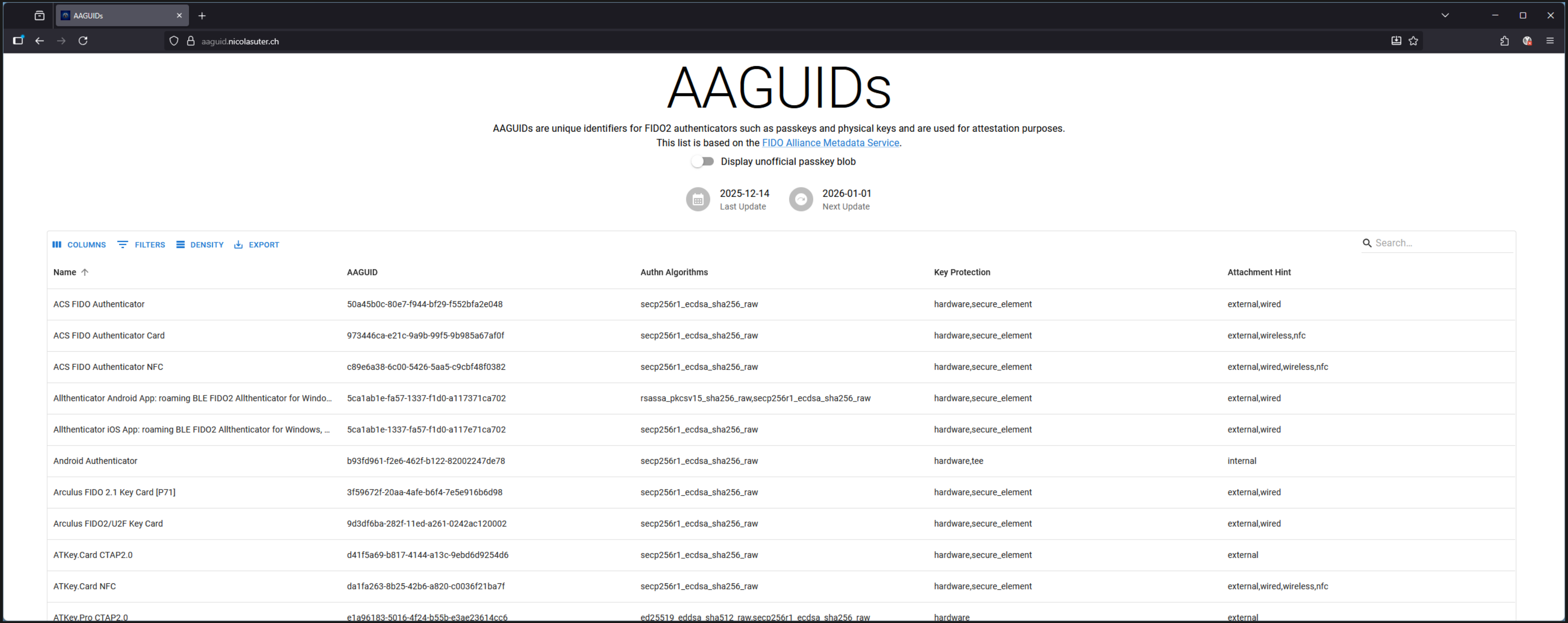
Does attestation matter?

Attestation

- Each make and model is identified by an AAGUID
  - 128-bit identifier (UUID)
  - YubiKey 5C Firmware 5.1 = cb69481e-8ff7-4039-93ec-0a2729a154a8
  - YubiKey 5C Firmware 5.7 = 19083c3d-8383-4b18-bc03-8f1c9ab2fd1b
- Burnt into the device by the manufacturer
- Small impact to user privacy
- Synced passkeys do not support attestation
- Enterprise attestation allows to identify certain FIDO2 keys instead of just a make and model

Attestation

MDS Browser e.g. https://aaguid.nicolasuter.ch/



AAGUIDs

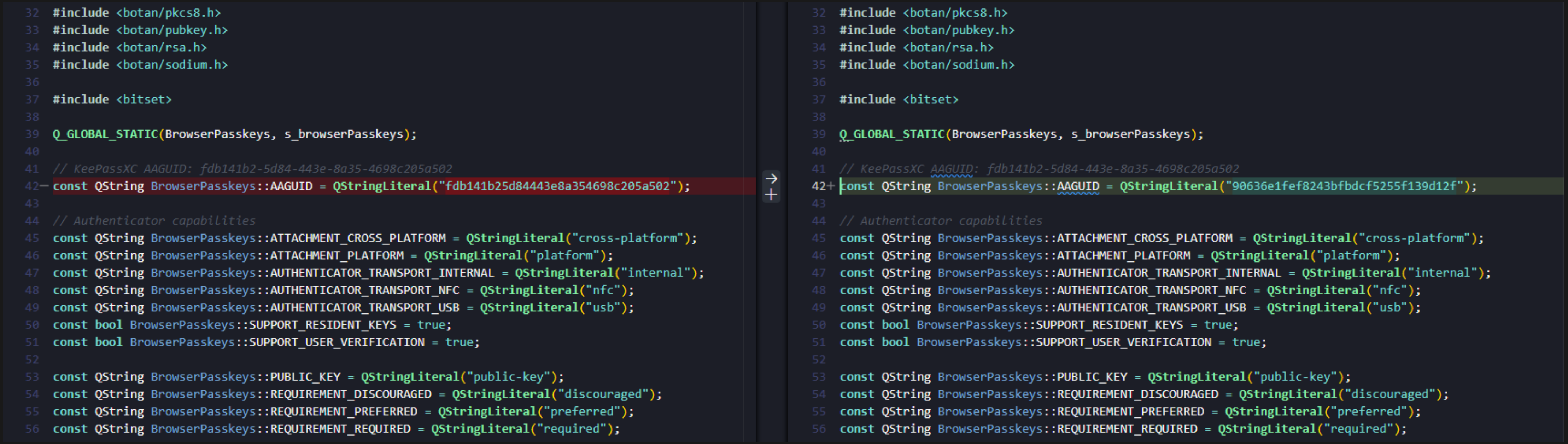
AAGUIDs are unique identifiers for FIDO2 authenticators such as passkeys and physical keys and are used for attestation purposes.
This list is based on the FIDO Alliance Metadata Service.

Display unofficial passkey blob

2025-12-14 Last Update
2026-01-01 Next Update

Name ↑	AAGUID	Authn Algorithms	Key Protection	Attachment Hint
ACS FIDO Authenticator	50a45b0c-80e7-f944-bf29-f552bfa2e048	secp256r1_ecdsa_sha256_raw	hardware,secure_element	external,wired
ACS FIDO Authenticator Card	973446ca-e21c-9a9b-99f5-9b985a67af0f	secp256r1_ecdsa_sha256_raw	hardware,secure_element	external,wireless,nfc
ACS FIDO Authenticator NFC	c89e6a38-6c00-5426-5aa5-c9cbf48f0382	secp256r1_ecdsa_sha256_raw	hardware,secure_element	external,wired,wireless,nfc
Allthenticator Android App: roaming BLE FIDO2 Allthenticator for Windo…	5ca1ab1e-fa57-1337-f1d0-a117371ca702	rsassa_pkcsv15_sha256_raw,secp256r1_ecdsa_sha256_raw	hardware,secure_element	external,wired
Allthenticator iOS App: roaming BLE FIDO2 Allthenticator for Windows, …	5ca1ab1e-1337-fa57-f1d0-a117e71ca702	secp256r1_ecdsa_sha256_raw	hardware,secure_element	external,wired
Android Authenticator	b93fd961-f2e6-462f-b122-82002247de78	secp256r1_ecdsa_sha256_raw	hardware,tee	internal
Arculus FIDO 2.1 Key Card [P71]	3f59672f-20aa-4afe-b6f4-7e5e916b6d98	secp256r1_ecdsa_sha256_raw	hardware,secure_element	external,wired
Arculus FIDO2/U2F Key Card	9d3df6ba-282f-11ed-a261-0242ac120002	secp256r1_ecdsa_sha256_raw	hardware,secure_element	external,wired
ATKey.Card CTAP2.0	d41f5a69-b817-4144-a13c-9ebd6d9254d6	secp256r1_ecdsa_sha256_raw	hardware,secure_element	external
ATKey.Card NFC	da1fa263-8b25-42b6-a820-c0036f21ba7f	secp256r1_ecdsa_sha256_raw	hardware,secure_element	external,wired,wireless,nfc
ATKey.Pro CTAP2.0	e1a96183-5016-4f24-b55b-e3ae23614cc6	ed25519_eddsa_sha512_raw,secp256r1_ecdsa_sha256_raw	hardware	external

Passkeys = Easily faked AAGUID

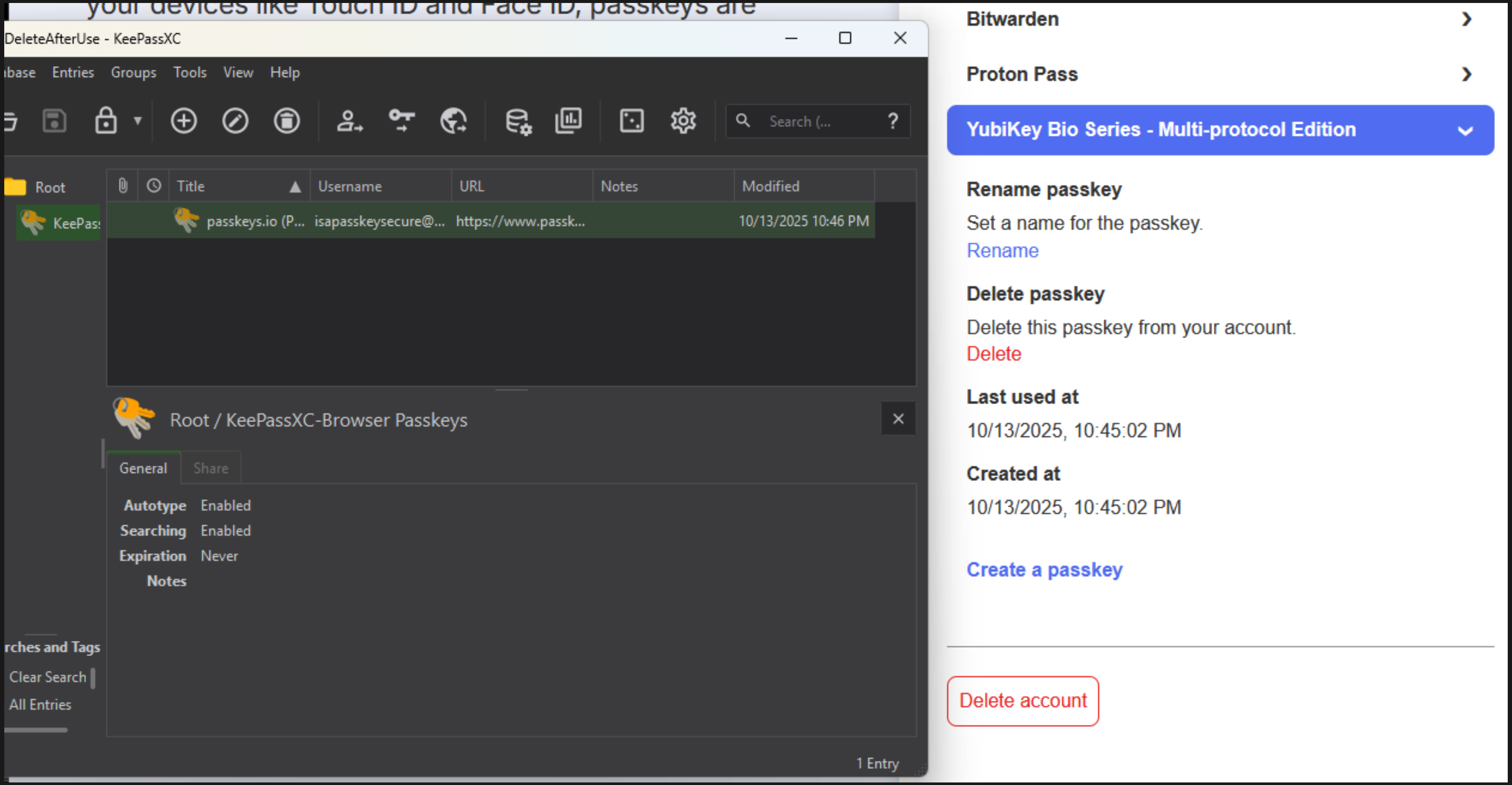


```
#include <botan/pkcs8.h>
#include <botan/pubkey.h>
#include <botan/rsa.h>
#include <botan/sodium.h>

#include <bitset>

Q_GLOBAL_STATIC(BrowserPasskeys, s_browserPasskeys);

// KeePassXC AAGUID: fdb141b2-5d84-443e-8a35-4698c205a502
-const QString BrowserPasskeys::AAGUID = QStringLiteral("fdb141b25d84443e8a354698c205a502");

// Authenticator capabilities
const QString BrowserPasskeys::ATTACHMENT_CROSS_PLATFORM = QStringLiteral("cross-platform");
const QString BrowserPasskeys::ATTACHMENT_PLATFORM = QStringLiteral("platform");
const QString BrowserPasskeys::AUTHENTICATOR_TRANSPORT_INTERNAL = QStringLiteral("internal");
const QString BrowserPasskeys::AUTHENTICATOR_TRANSPORT_NFC = QStringLiteral("nfc");
const QString BrowserPasskeys::AUTHENTICATOR_TRANSPORT_USB = QStringLiteral("usb");
const bool BrowserPasskeys::SUPPORT_RESIDENT_KEYS = true;
const bool BrowserPasskeys::SUPPORT_USER_VERIFICATION = true;

const QString BrowserPasskeys::PUBLIC_KEY = QStringLiteral("public-key");
const QString BrowserPasskeys::REQUIREMENT_DISCOURAGED = QStringLiteral("discouraged");
const QString BrowserPasskeys::REQUIREMENT_PREFERRED = QStringLiteral("preferred");
const QString BrowserPasskeys::REQUIREMENT_REQUIRED = QStringLiteral("required");
```

```
#include <botan/pkcs8.h>
#include <botan/pubkey.h>
#include <botan/rsa.h>
#include <botan/sodium.h>

#include <bitset>

Q_GLOBAL_STATIC(BrowserPasskeys, s_browserPasskeys);

// KeePassXC AAGUID: fdb141b2-5d84-443e-8a35-4698c205a502
+const QString BrowserPasskeys::AAGUID = QStringLiteral("90636e1fef8243bfbdcf5255f139d12f");

// Authenticator capabilities
const QString BrowserPasskeys::ATTACHMENT_CROSS_PLATFORM = QStringLiteral("cross-platform");
const QString BrowserPasskeys::ATTACHMENT_PLATFORM = QStringLiteral("platform");
const QString BrowserPasskeys::AUTHENTICATOR_TRANSPORT_INTERNAL = QStringLiteral("internal");
const QString BrowserPasskeys::AUTHENTICATOR_TRANSPORT_NFC = QStringLiteral("nfc");
const QString BrowserPasskeys::AUTHENTICATOR_TRANSPORT_USB = QStringLiteral("usb");
const bool BrowserPasskeys::SUPPORT_RESIDENT_KEYS = true;
const bool BrowserPasskeys::SUPPORT_USER_VERIFICATION = true;

const QString BrowserPasskeys::PUBLIC_KEY = QStringLiteral("public-key");
const QString BrowserPasskeys::REQUIREMENT_DISCOURAGED = QStringLiteral("discouraged");
const QString BrowserPasskeys::REQUIREMENT_PREFERRED = QStringLiteral("preferred");
const QString BrowserPasskeys::REQUIREMENT_REQUIRED = QStringLiteral("required");
```

glueck kanja

Passkeys = Easily faked AAGUID


your devices like Touch ID and Face ID, passkeys are

DeleteAfterUse - KeePassXC

		Title	Username	URL	Notes	Modified
		passkeys.io (P...	isapasskeysecure@...	https://www.passk...		10/13/2025 10:46 PM

Root / KeePassXC-Browser Passkeys

Autotype Enabled
Searching Enabled
Expiration Never
Notes

1 Entry

Rename passkey
Set a name for the passkey.
Rename

Delete passkey
Delete this passkey from your account.
Delete

Last used at
10/13/2025, 10:45:02 PM

Created at
10/13/2025, 10:45:02 PM

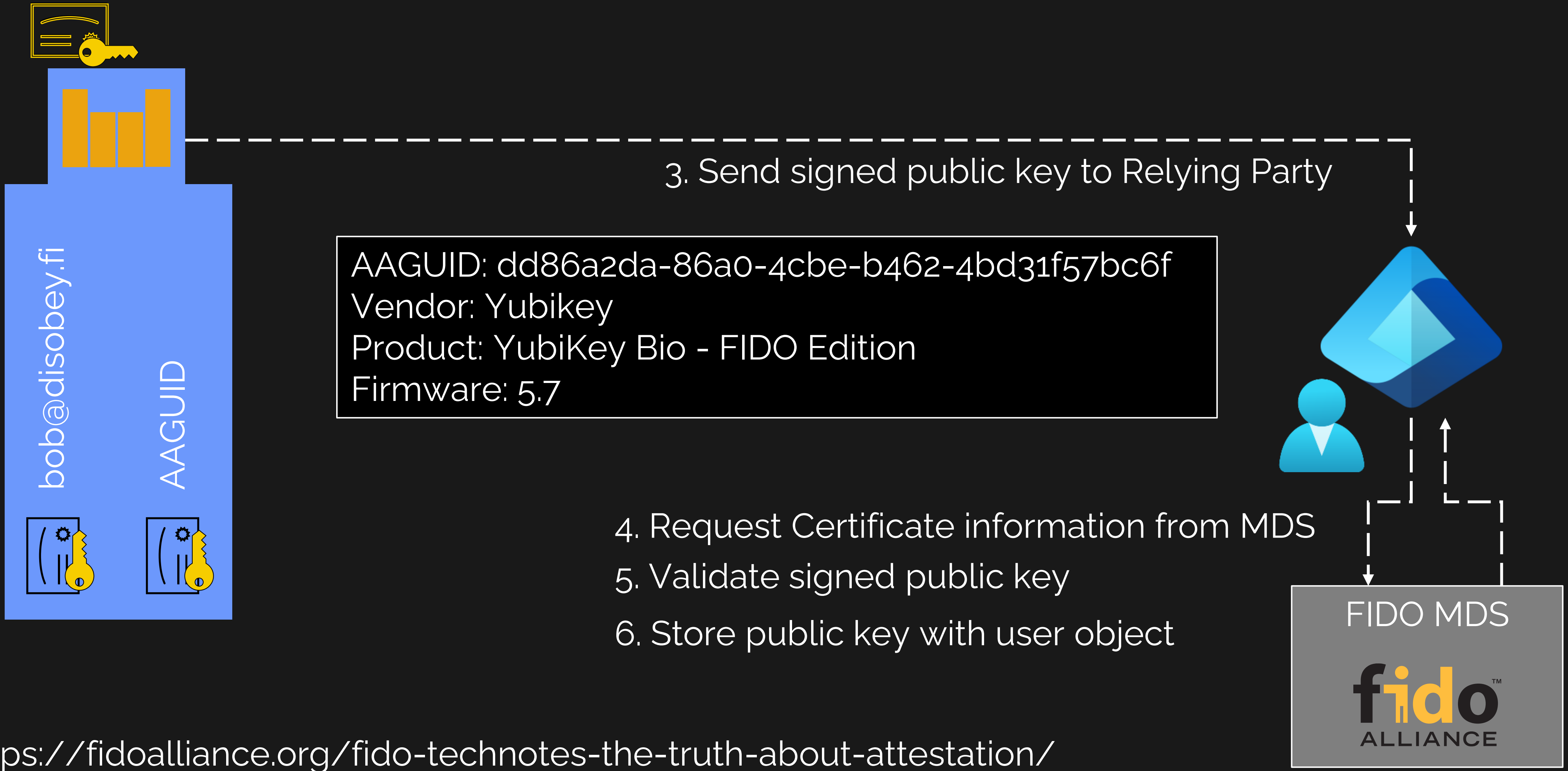
Create a passkey

Delete account

DISOBEY THE NORDIC SECURITY EVENT

Attestation

1. Credential key pair generated
2. Sign public key with private attestation key



AAGUID: dd86a2da-86a0-4cbe-b462-4bd31f57bc6f
Vendor: Yubikey
Product: YubiKey Bio - FIDO Edition
Firmware: 5.7

4. Request Certificate information from MDS
5. Validate signed public key
6. Store public key with user object

https://fidoalliance.org/fido-technotes-the-truth-about-attestation/

glueck kanja

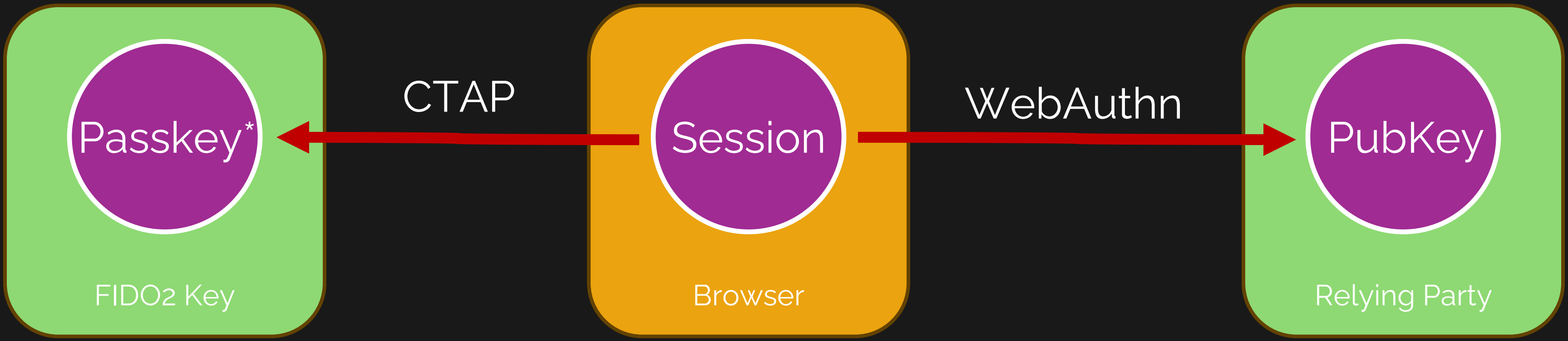
Syncable Passkeys do not support attestation

DISOBEY
THE NORDIC SECURITY EVENT

glueck kanja

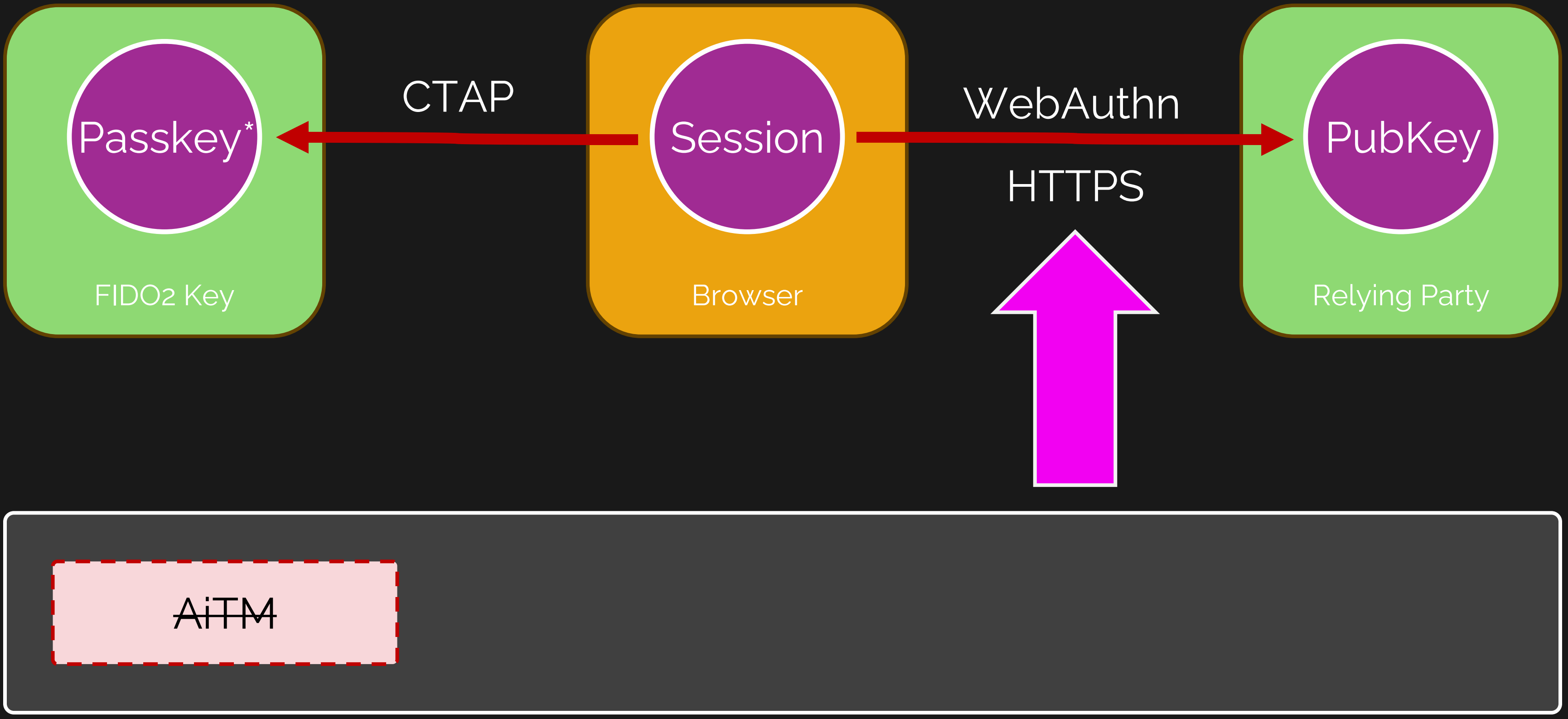
Threat modeling for enterprises

Attack vectors: Device-bound passkeys



*Passkey = Private Key

Attack vectors: Device-bound passkeys

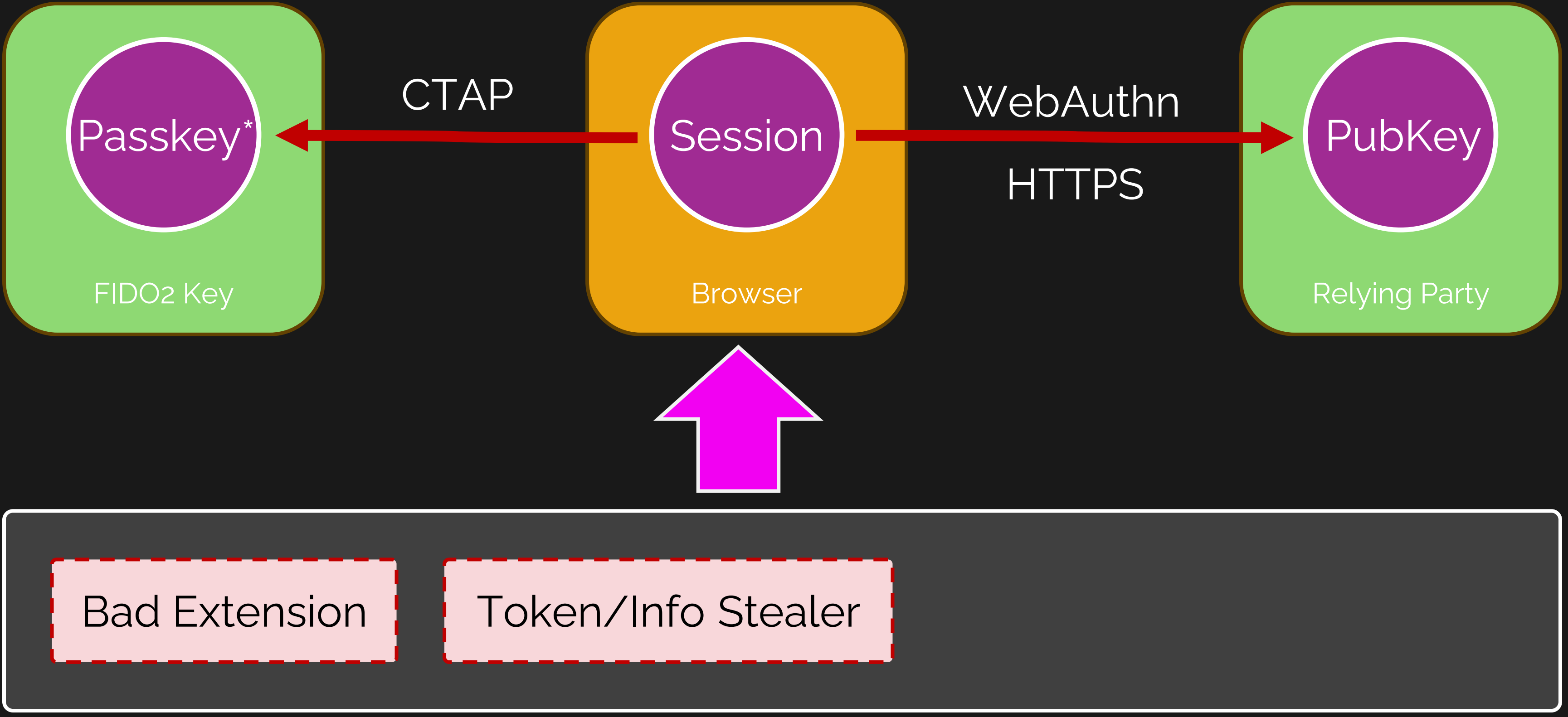


*Passkey = Private Key

Downgrade attacks

- AiTM "hide" the Passkey sign-in option from the victim
- If already selected the AiTM plugin will force a fallback to another, phishable method
- Security of the Passkey credential is not impacted
- Not a "real" attack on passkey, but still relevant

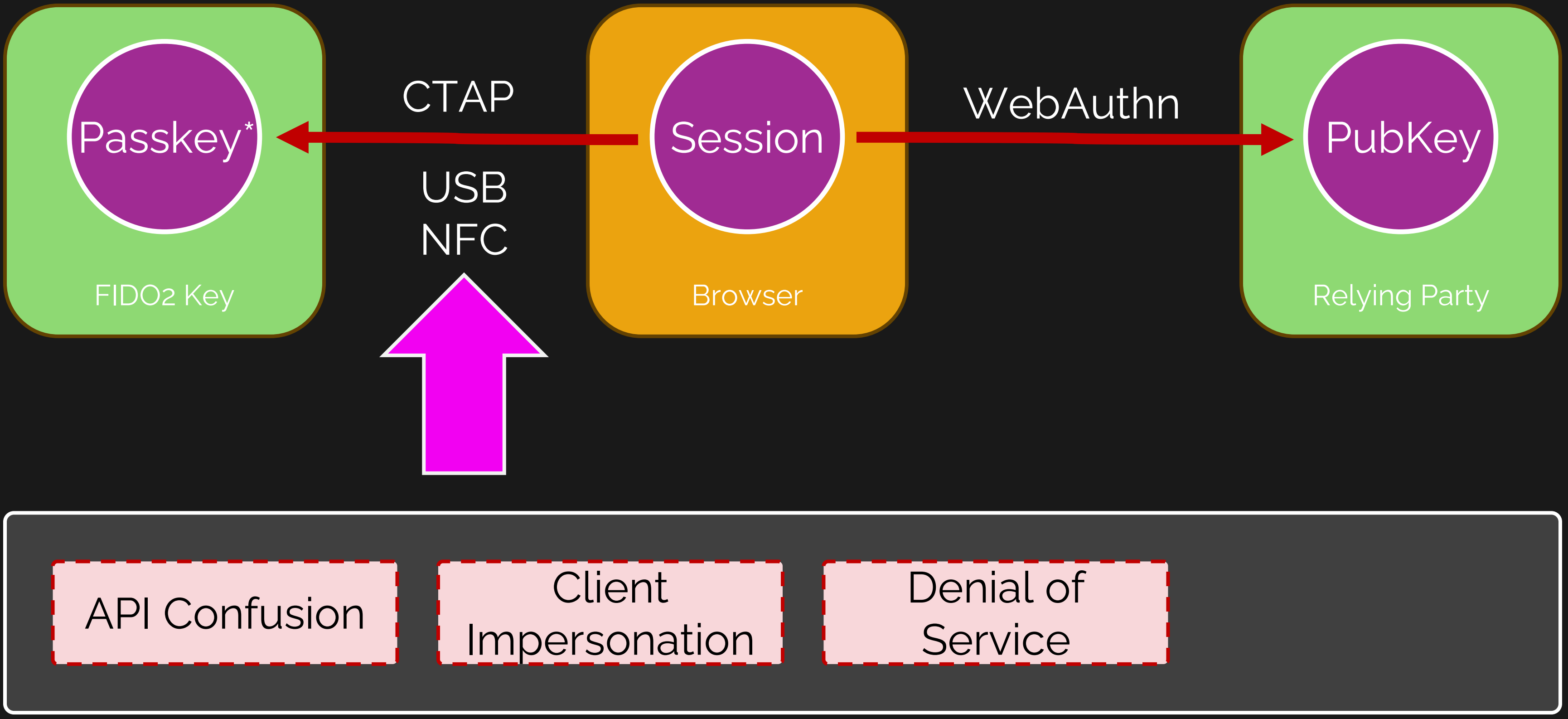
Attack vectors: Device-bound passkeys



*Passkey = Private Key

glueck kanja

Attack vectors: Device-bound passkeys



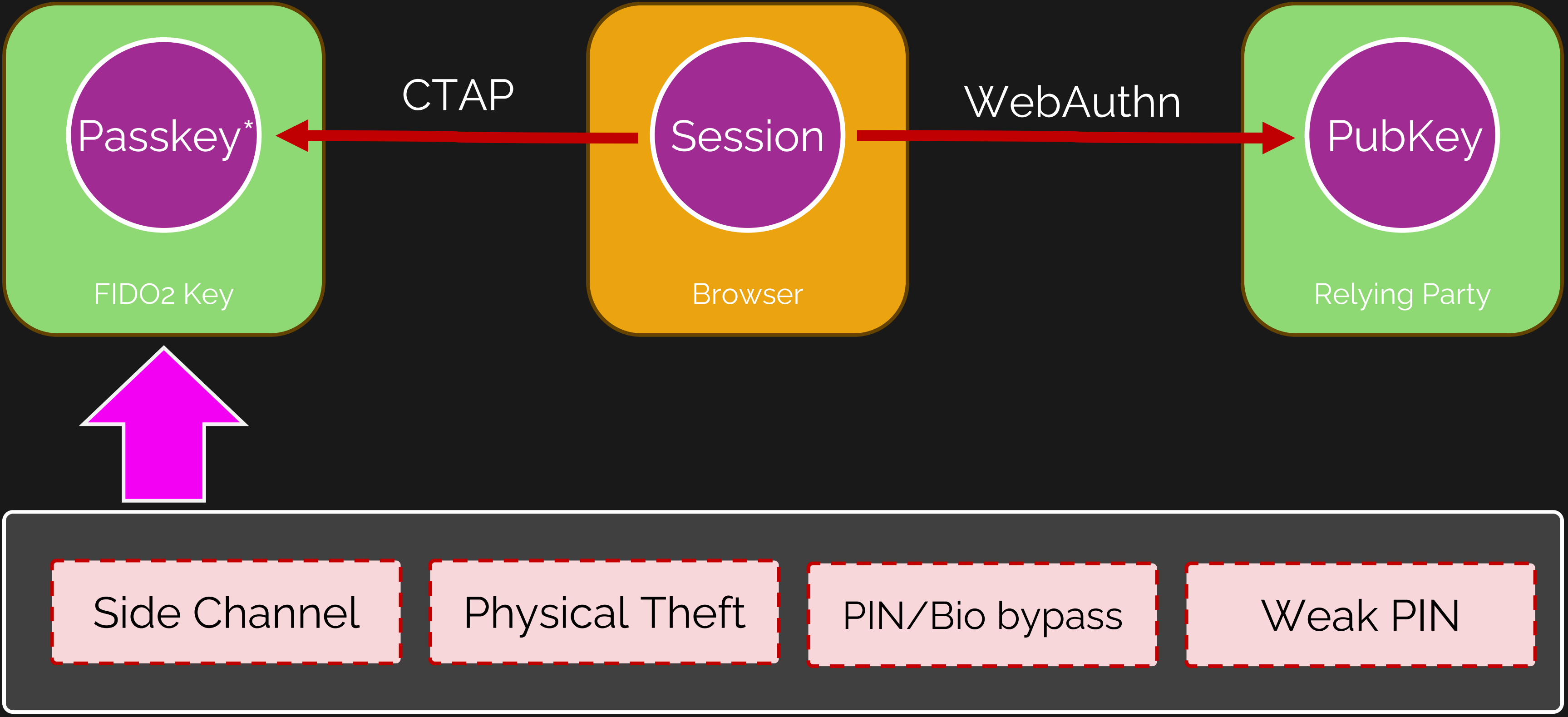
*Passkey = Private Key

DISOBEY
THE NORDIC SECURITY EVENT

Known attacks - CTRAPS

- CTRAPS: CTAP Client Impersonation and API Confusion on FIDO2
  - Marco Casagrande, EURECOM & Daniele Antonioli, EURECOM
- Attacks
  - Delete Credentials (NFC)
  - List Credentials
- Sources:
  - https://arxiv.org/pdf/2412.02349
  - https://www.youtube.com/watch?v=07B0etOq7OM

Attack vectors: Device-bound passkeys

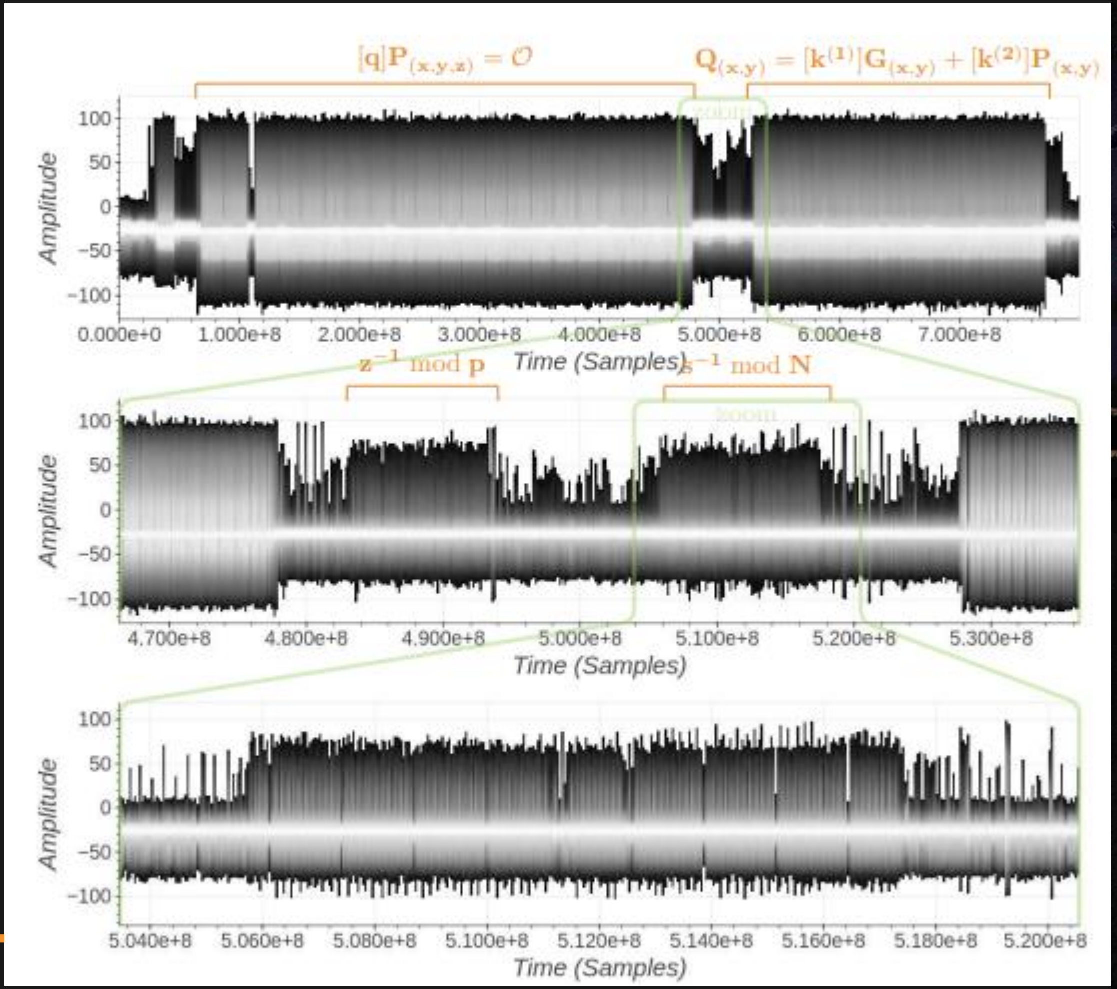


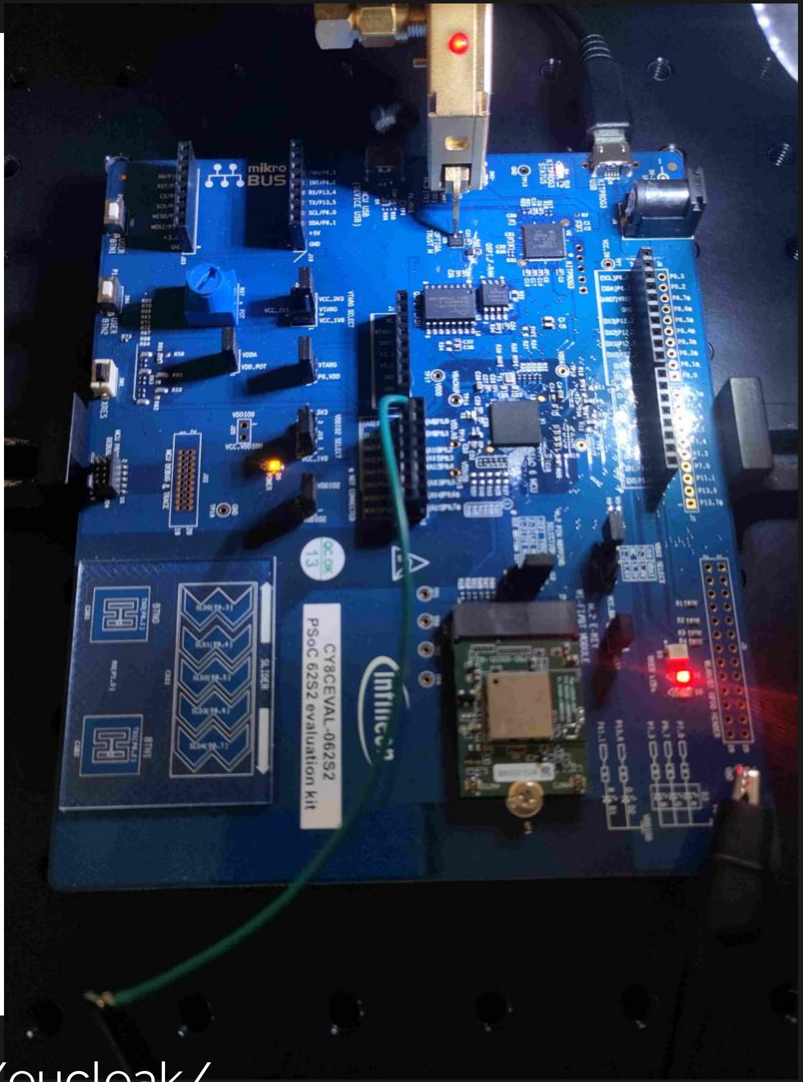
*Passkey = Private Key

Known attacks - CVE-2024-45678

- Side-Channel attack on Infineon SLE78
- Only when used with Infineon cryptographic library
- Allows for extraction of private keys
- Affected: All YubiKey 5 Series < 5.7
- Requirements:
  - Physical access
  - PIN for the device
- Research published: https://ninjalab.io/eucleak/

Known attacks - CVE-2024-45678





Source: https://ninjalab.io/eucleak/

DISOBEY THE NORDIC SECURITY EVENT

Known attacks - CVE-2024-45678

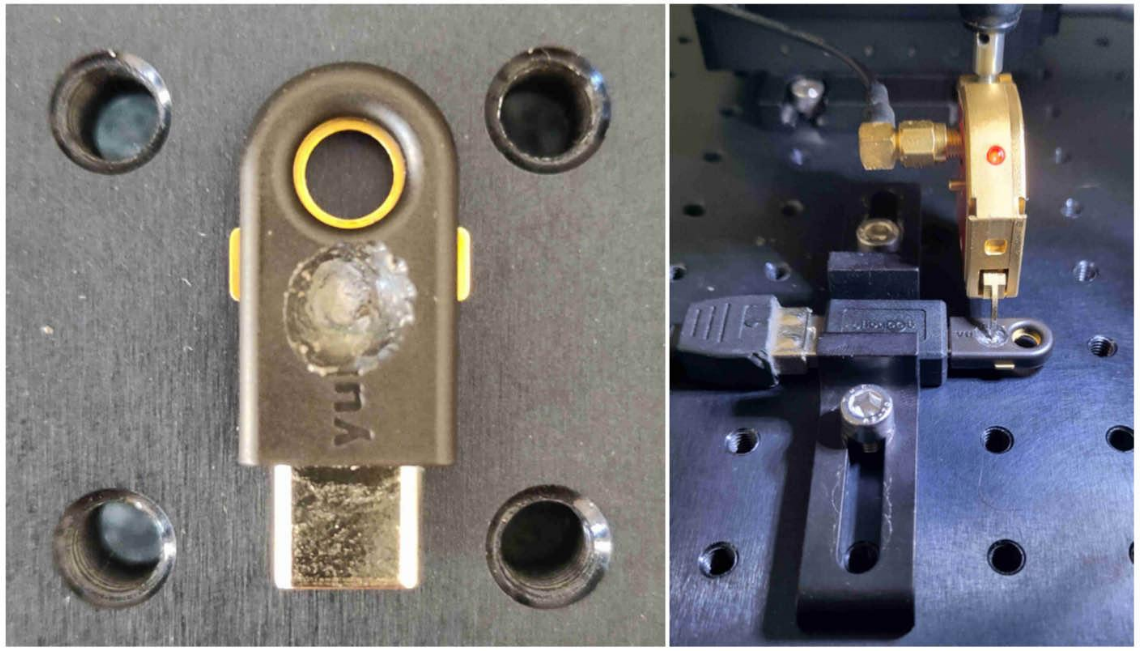


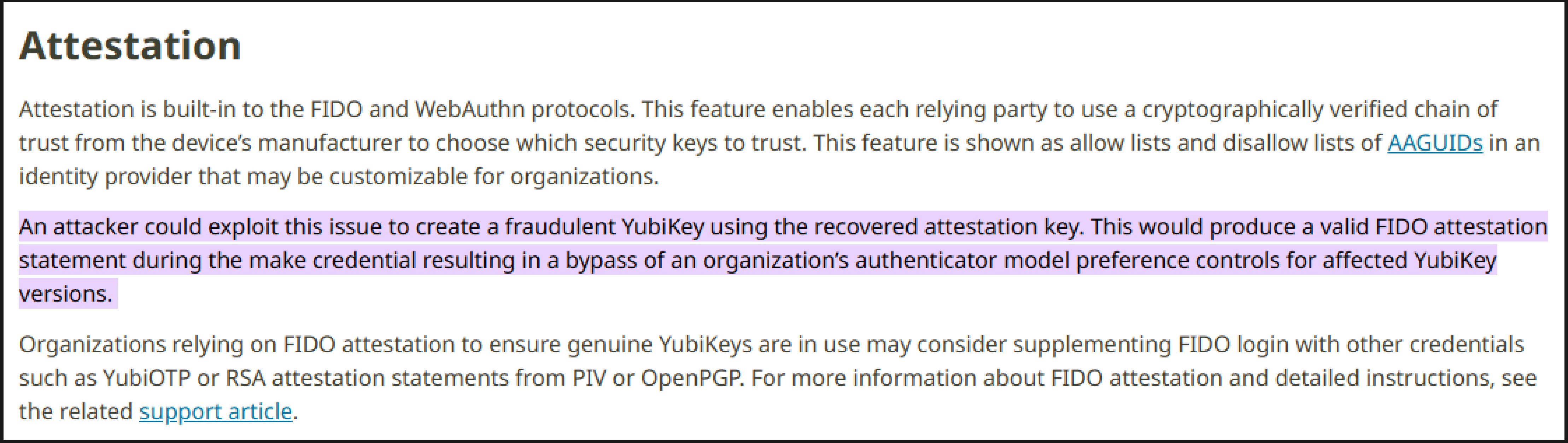
Figure A.3: `YubiKey 5C` – Second Opening

In both cases however, the device needs to be re-packaged if the adversary wants to give it back to legitimate user without him noticing. We did not study further this issue.

Source: https://ninjalab.io/eucleak/

glueck▪kanja

Impact on attestation



Attestation

Attestation is built-in to the FIDO and WebAuthn protocols. This feature enables each relying party to use a cryptographically verified chain of trust from the device's manufacturer to choose which security keys to trust. This feature is shown as allow lists and disallow lists of AAGUIDs in an identity provider that may be customizable for organizations.

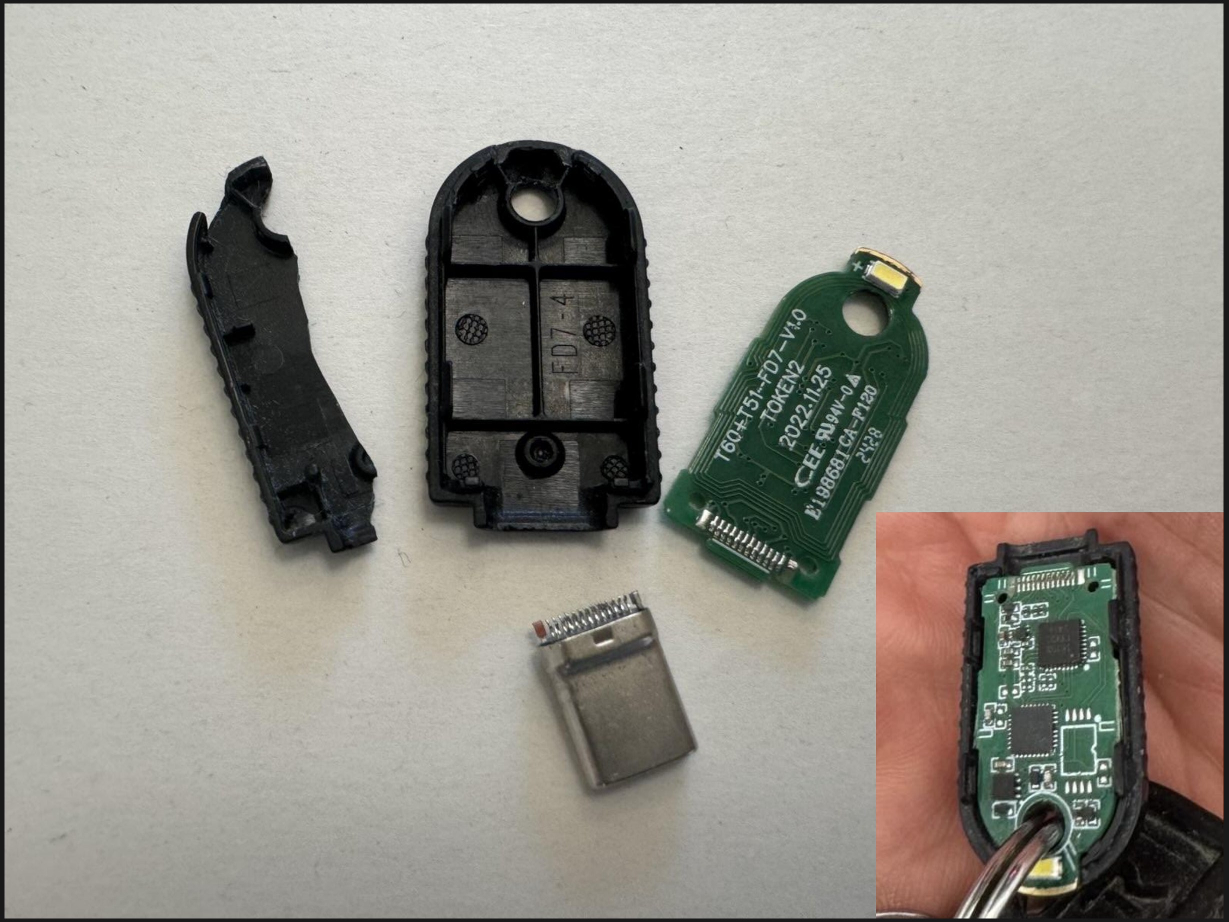
An attacker could exploit this issue to create a fraudulent YubiKey using the recovered attestation key. This would produce a valid FIDO attestation statement during the make credential resulting in a bypass of an organization's authenticator model preference controls for affected YubiKey versions.

Organizations relying on FIDO attestation to ensure genuine YubiKeys are in use may consider supplementing FIDO login with other credentials such as YubiOTP or RSA attestation statements from PIV or OpenPGP. For more information about FIDO attestation and detailed instructions, see the related support article.

Source:
https://www.yubico.com/support/security-advisories/ysa-2024-03/

DISOBEY THE NORDIC SECURITY EVENT

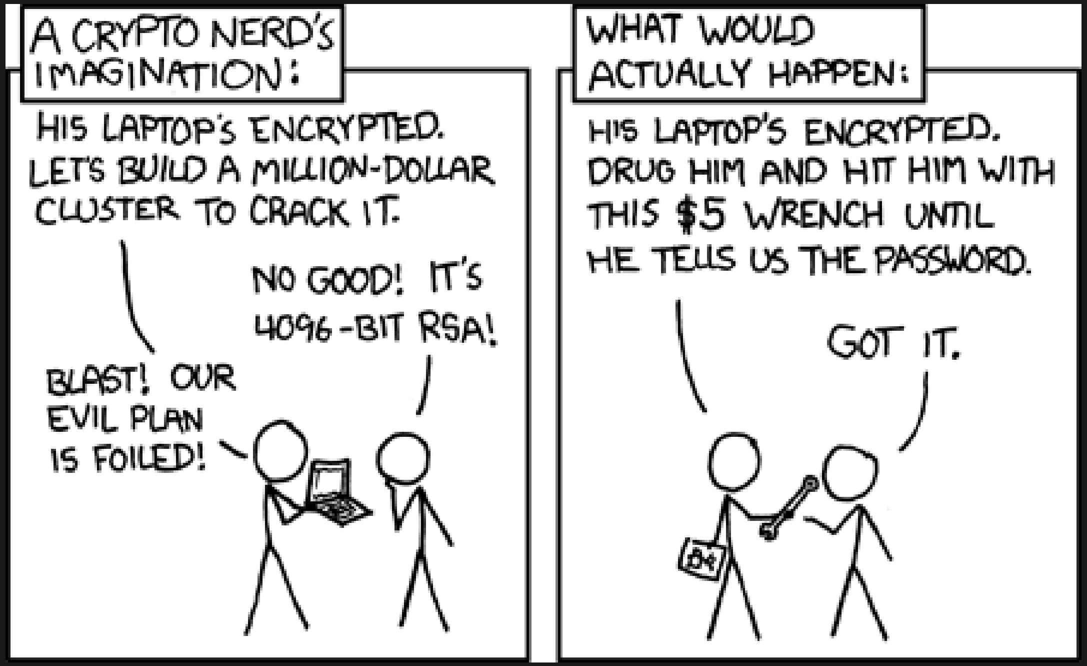
Other hardware token



Photos by Jakob Schaefer

glueck▪kanja

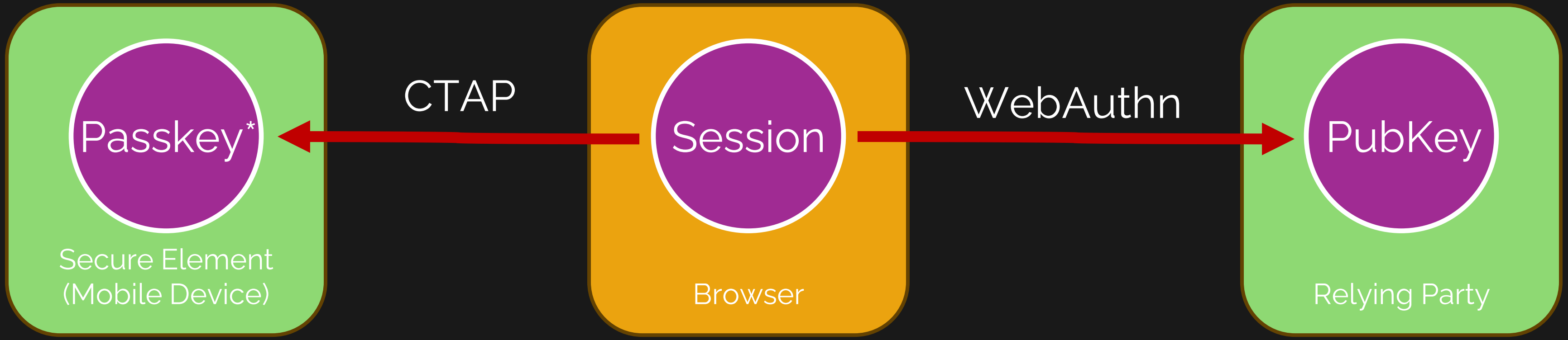
Easier method


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.
B4

Source: https://xkcd.com/538/

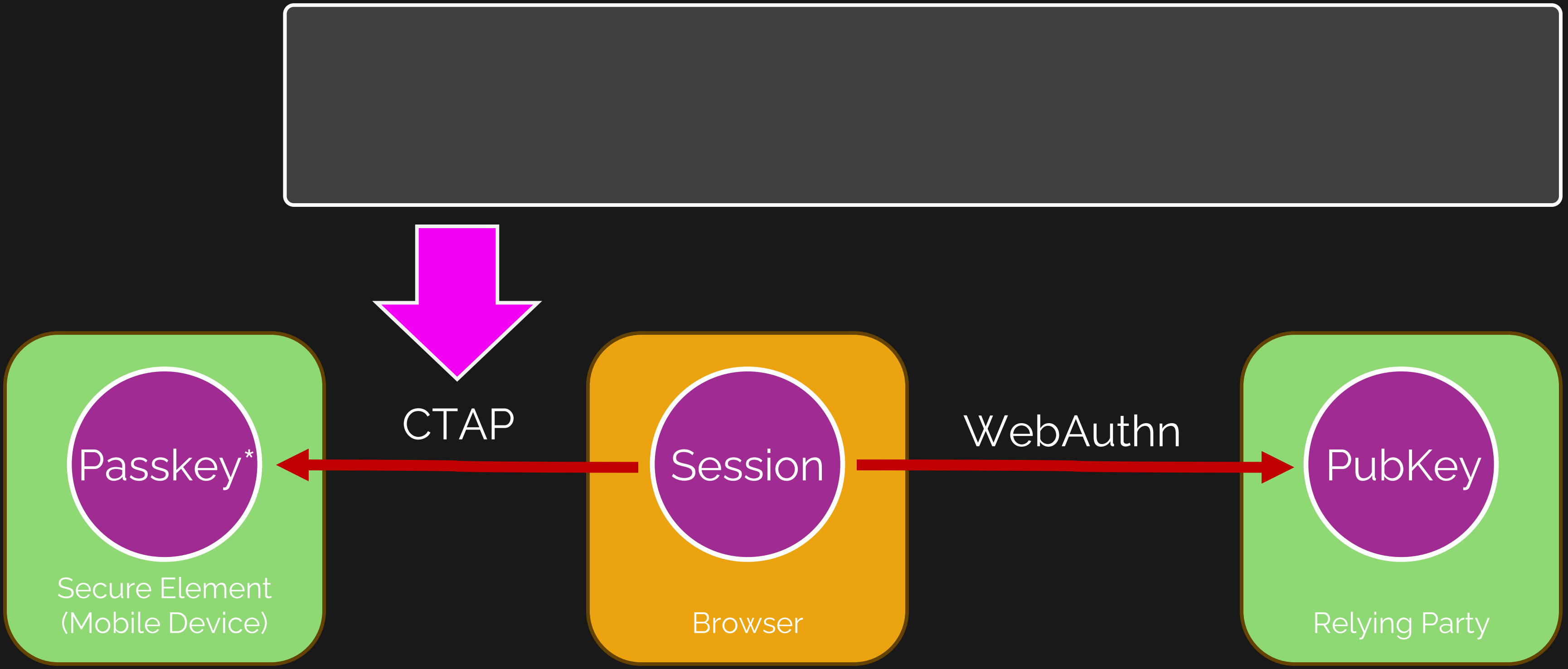
DISOBEY
THE NORDIC SECURITY EVENT

Attack vectors: Syncable passkeys



*Passkey = Private Key

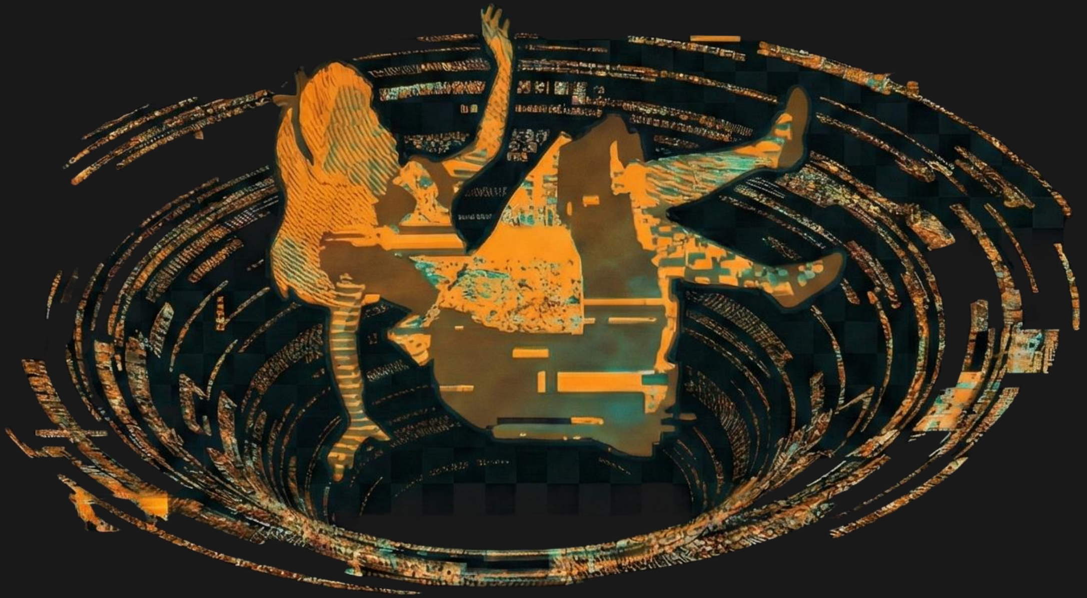
Attack vectors: Syncable passkeys



Same device
Cross-device (CDA)

*Passkey = Private Key

glueck kanja

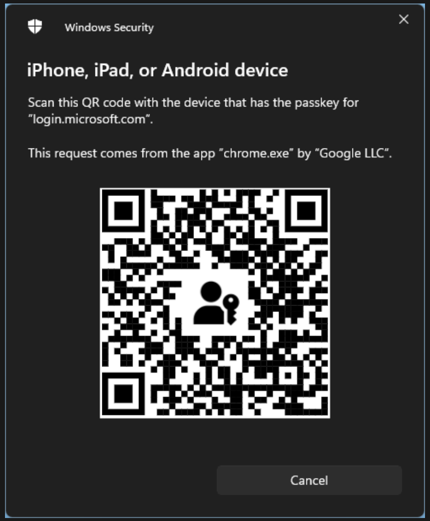


Cross Device Authentication

Cross Device Authentication

- Bluetooth (BLE) on both devices for proximity check

- Internet access for data transfer
  - https://cable.ua5v.com (Android)
  - https://cable.auth.com (Apple)


Windows Security

iPhone, iPad, or Android device

Scan this QR code with the device that has the passkey for "login.microsoft.com".

This request comes from the app "chrome.exe" by "Google LLC".

Cancel

```
FIDO:/088521772645746
304256629196898023805
213791974887885159969
946751928771388701793
485401070923423366366
303159168738737767290
060661159865120837177
```

glueck kanja

QR Code unraveled

```
FIDO:/088521772645746
3042566291968980238805
213791974887885159969
946751928771388701793
485401070923423366366
303159168738737767290
060661159865120837177
011010667266107096654
083332
```

- Base10 encoded string
- Concise Binary Object Representation (CBOR) data format

DISOBEY
THE NORDIC SECURITY EVENT

QR Code unraveled

```
FIDO:/088521772645746
304256629196898023805
213791974887885159969
946751928771388701793
485401070923423366366
303159168738737767290
060661159865120837177
011010667266107096654
083332
```

- Base10 encoded string
- Concise Binary Object Representation (CBOR) data format

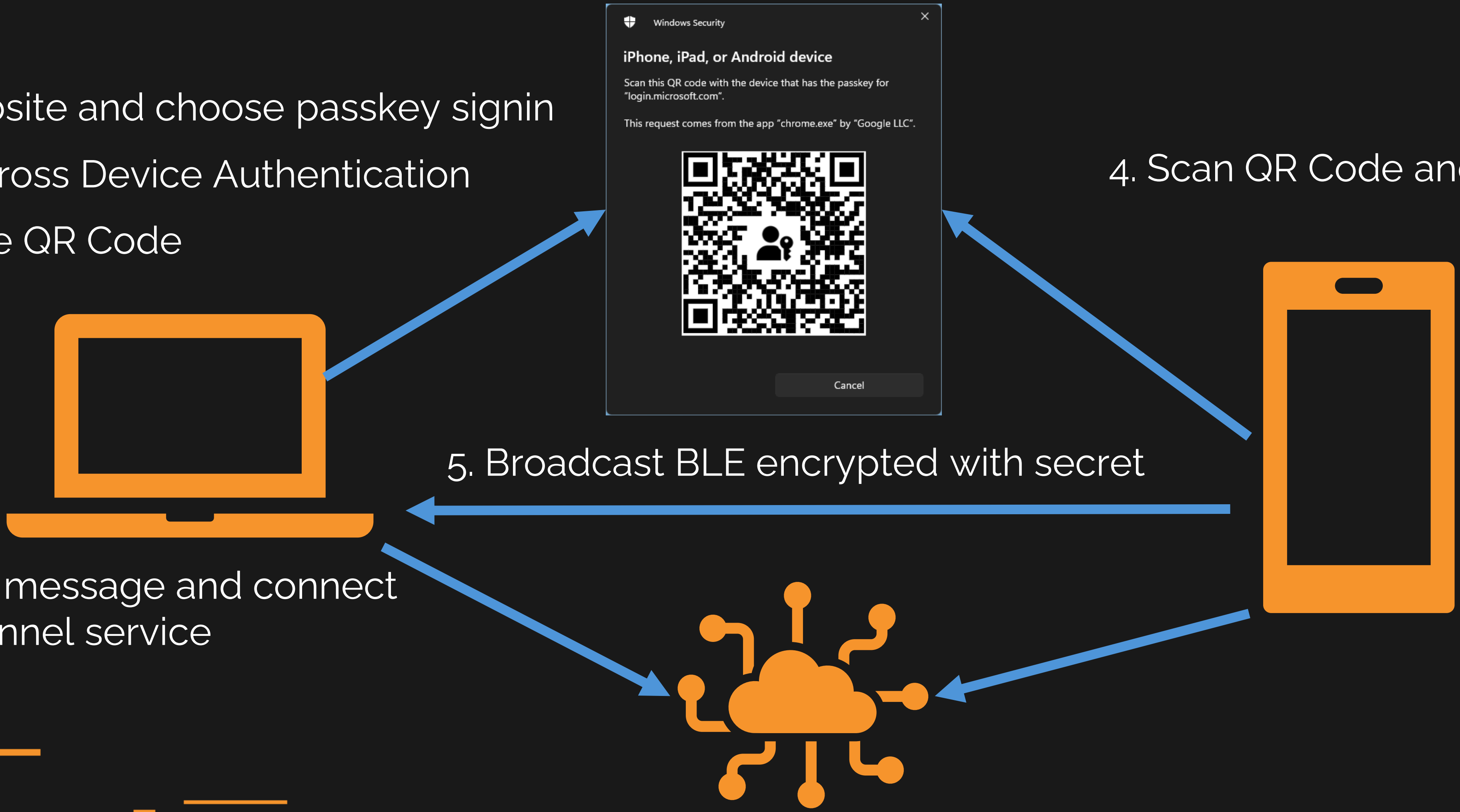
QR Code unraveled

```
A6 00 58 21 02 73 1F
00 0E 75 37 28 D1 39
97 00 CD 91 98 8A EA
85 12 00 2D B4 16 91
2E D5 38 00 7A 17 FF
52 2B 56 31 00 5F C4
01 50 7A F8 FB 00 59
60 2E FD 4F 8C 38 00
48 AF DA C4 B5 27 02
00 02 03 1A 67 4B 14
84 00 04 F5 05 62 67
61 00 00
```

```
// Compressed public key
0: h'02731F0E7537[...]315FC4'
// Shared secret
1: h'7AF8FB59602E[...]C4B527'
// decodeTunnelServerDomain
2: 2
// Current epoch time
3: 1732973700
// State-assisted transactions
4: true
// getAssertion or makeCredential
5: "ga"
```

Cross Device Authentication

1. Visit website and choose passkey signin
2. Select Cross Device Authentication
3. Generate QR Code



Windows Security

iPhone, iPad, or Android device

Scan this QR code with the device that has the passkey for "login.microsoft.com".

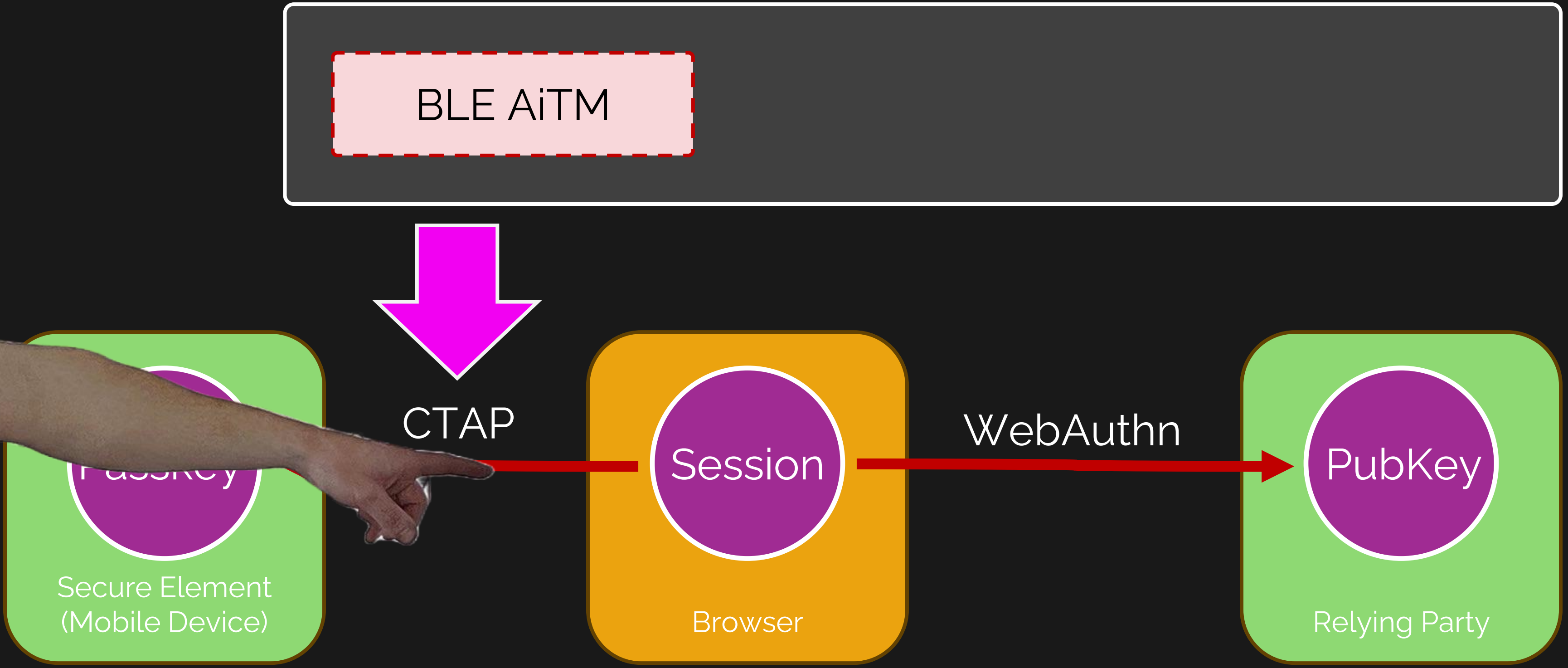
This request comes from the app "chrome.exe" by "Google LLC".

4. Scan QR Code and retrieve secret

5. Broadcast BLE encrypted with secret

6. Receive message and connect through tunnel service

Attack vectors: Syncable passkeys



Same device
Cross-device (CDA)

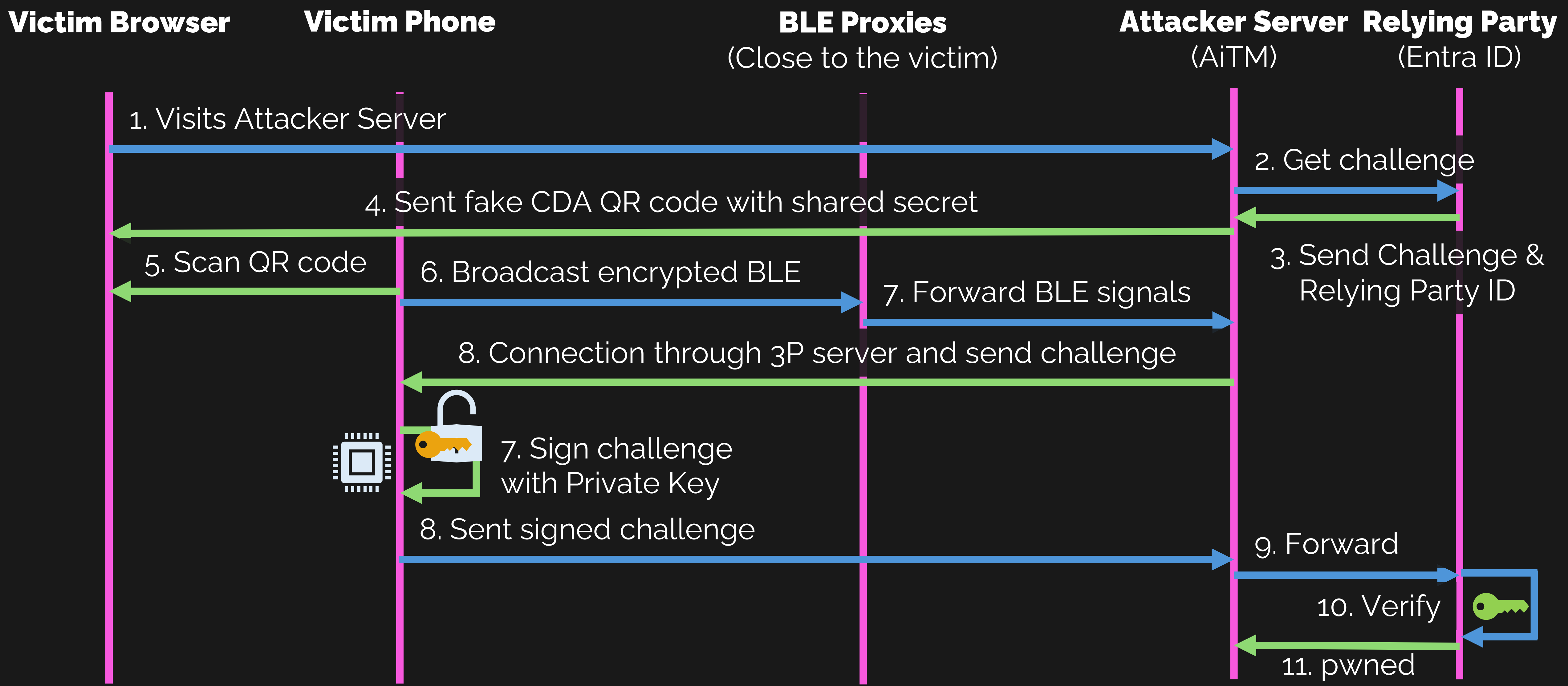
*Passkey = Private Key

Known attacks

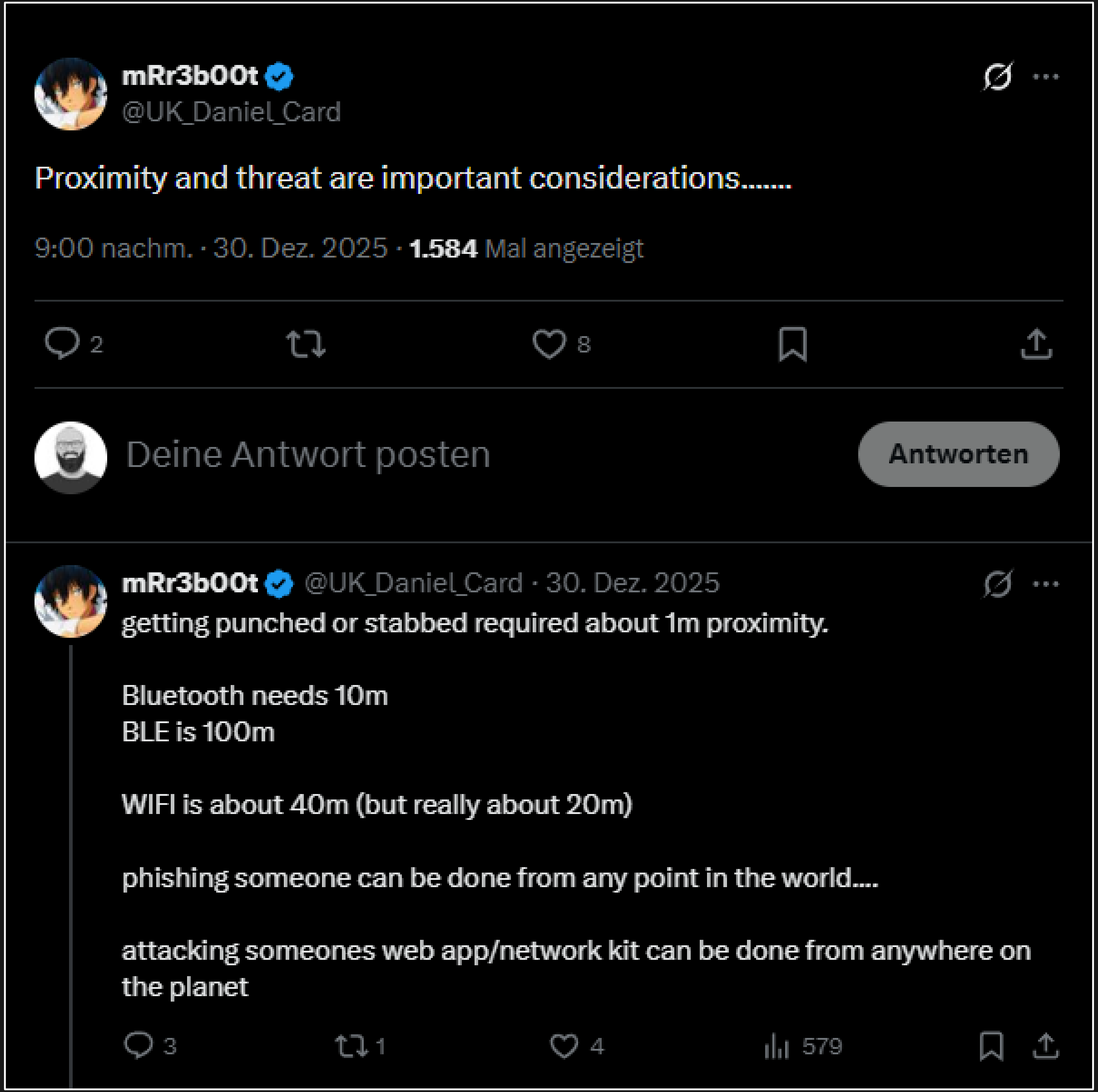
- BLE AiTM aka Cross Device Phishing
- Sources:
  - https://mastersplinter.work/research/passkey/#cve-2024-9956
  - https://www.inovex.de/de/blog/phishing-for-passkeys-an-analysis-of-webauthn-and-ctap/
  - https://denniskniep.github.io/posts/14-fido-cross-device-phishing/

glueck kanja

Passkey phising


Victim Browser
Victim Phone
BLE Proxies
(Close to the victim)
Attacker Server
(AiTM)
Relying Party
(Entra ID)
1. Visits Attacker Server
2. Get challenge
4. Sent fake CDA QR code with shared secret
3. Send Challenge & Relying Party ID
5. Scan QR code
6. Broadcast encrypted BLE
7. Forward BLE signals
8. Connection through 3P server and send challenge
7. Sign challenge with Private Key
8. Sent signed challenge
9. Forward
10. Verify
11. pwned

glueck kanja



mRr3b00t
@UK_Daniel_Card

Proximity and threat are important considerations.......

9:00 nachm. · 30. Dez. 2025 · **1.584** Mal angezeigt

2 8

Deine Antwort posten

Antworten

mRr3b00t @UK_Daniel_Card · 30. Dez. 2025

getting punched or stabbed required about 1m proximity.

Bluetooth needs 10m
BLE is 100m

WIFI is about 40m (but really about 20m)

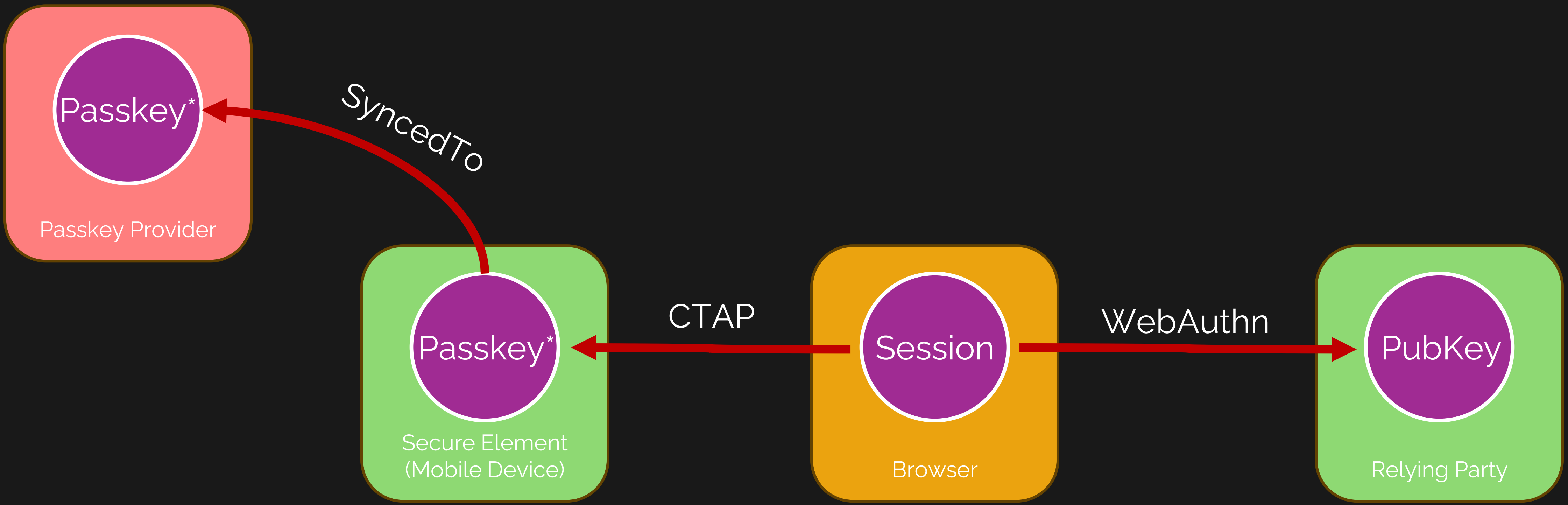
phishing someone can be done from any point in the world....

attacking someones web app/network kit can be done from anywhere on the planet

3 1 4 579

DISOBEY
THE NORDIC SECURITY EVENT

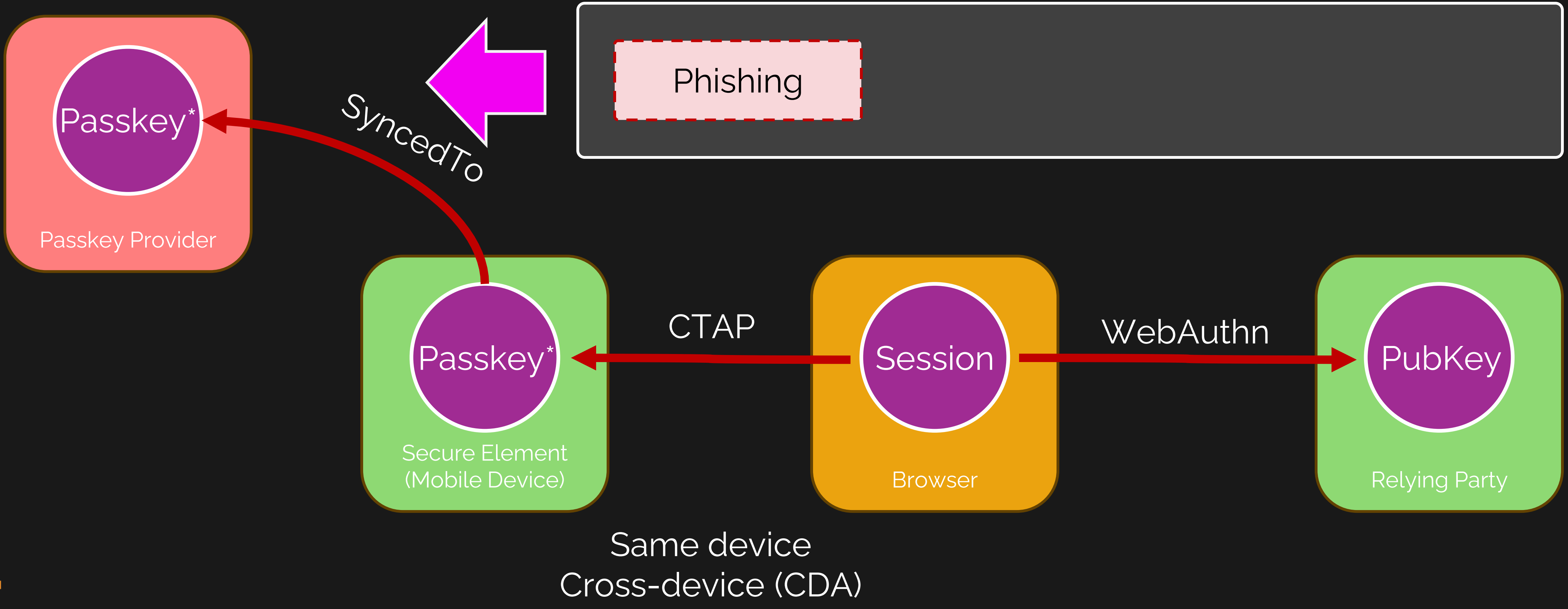
Attack vectors: Syncable passkeys



Same device
Cross-device (CDA)

*Passkey = Private Key

Attack vectors: Syncable passkeys



*Passkey = Private Key

Attacks on syncable passkeys

- Apple
  - Recovery: iCloud Account, Password, SMS + Device Passcode
  - 10 attempts until locked
  - https://support.apple.com/en-us/102195
- Google
  - Recovery: Google Account, Password + Device Passcode or PIN
  - https://security.googleblog.com/2022/10/SecurityofPasskeysintheGooglePasswordManager.html
- Third parties
  - It's the wild west

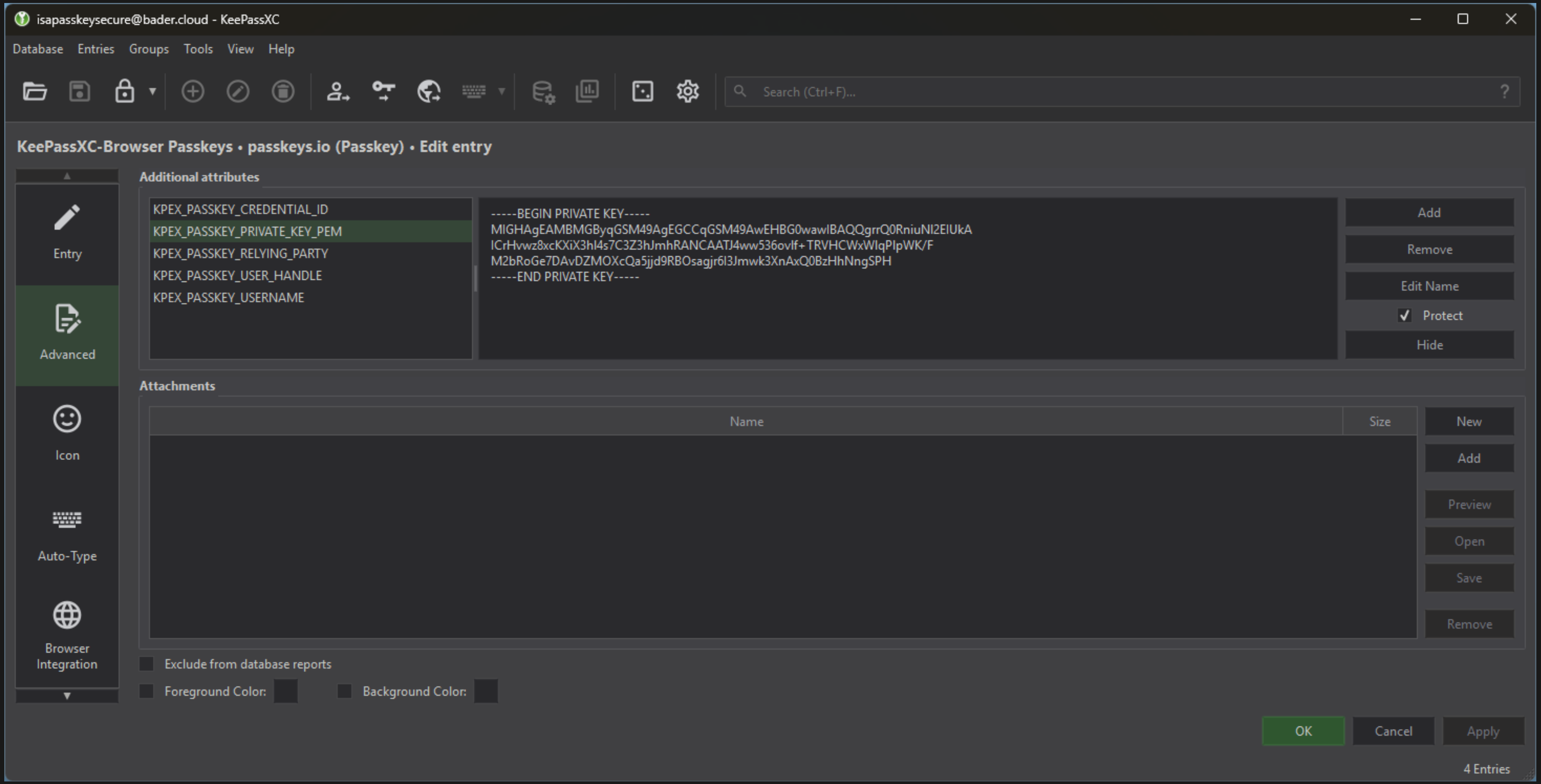
Phishing of main account

Provider	MFA Required for Passkey creation	Export/ Import	New device verification	New device notification
Bitwarden	No	Yes/Yes	E-Mail	Yes
ProtonPass	No	Yes/Yes	No	No
KeePassXC	No	Yes/Yes	No	No
Keeper	No	Yes/Yes	E-Mail*	No

*Based on source IP address

glueck▬kanja

Phishing of main account

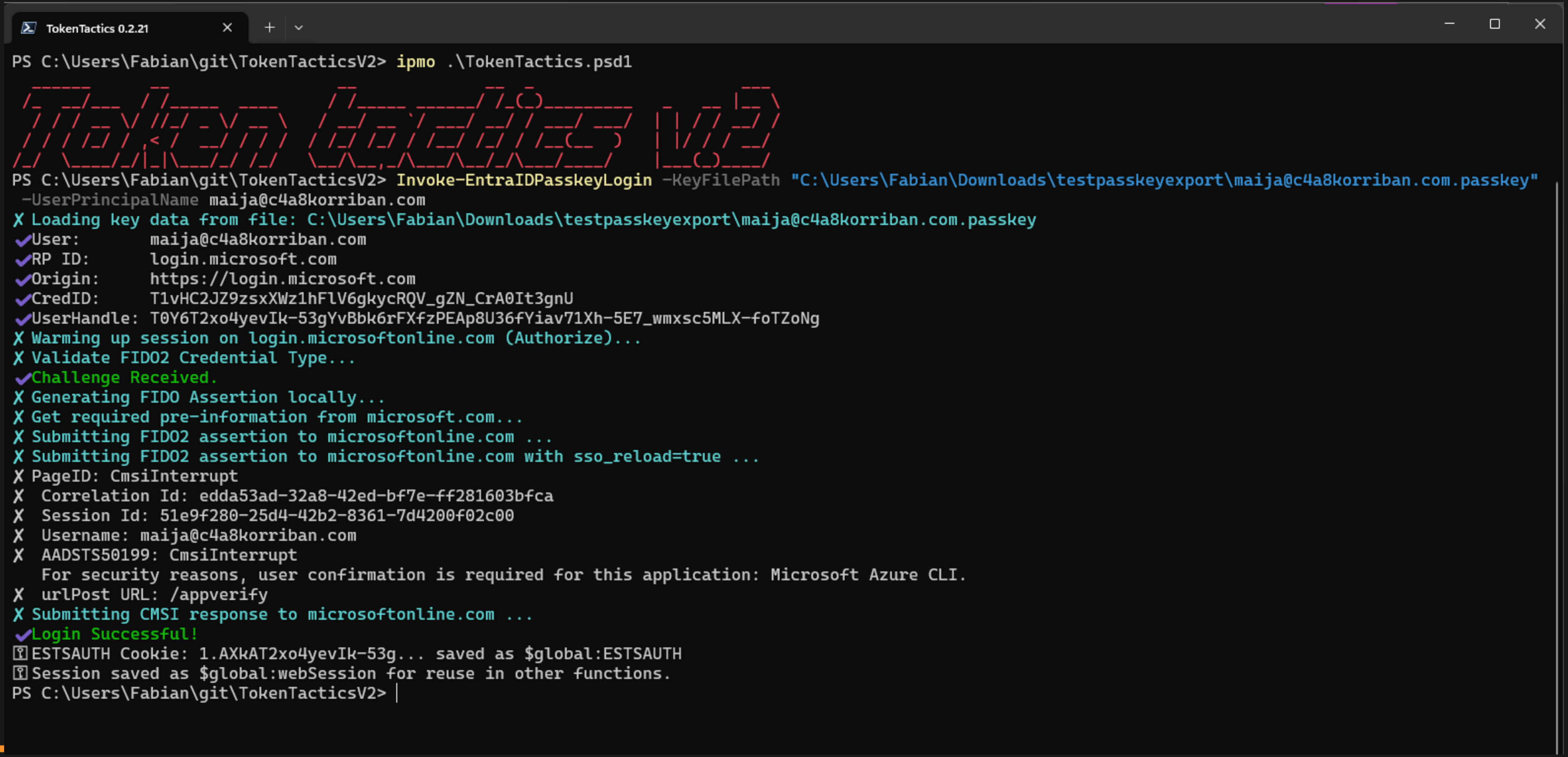


KPEX_PASSKEY_CREDENTIAL_ID
KPEX_PASSKEY_PRIVATE_KEY_PEM
KPEX_PASSKEY_RELYING_PARTY
KPEX_PASSKEY_USER_HANDLE
KPEX_PASSKEY_USERNAME

```
-----BEGIN PRIVATE KEY-----
MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgrrQ0RniuNI2EIUkA
lCrHvwz8xcKXiX3hl4s7C3Z3hJmhRANCAATJ4ww536ovlf+TRVHCWxWIqPIpWK/F
M2bRoGe7DAvDZMOXcQa5jjd9RBOsagjr6l3Jmwk3XnAxQ0BzHhNngSPH
-----END PRIVATE KEY-----
```

DISOBEY
THE NORDIC SECURITY EVENT

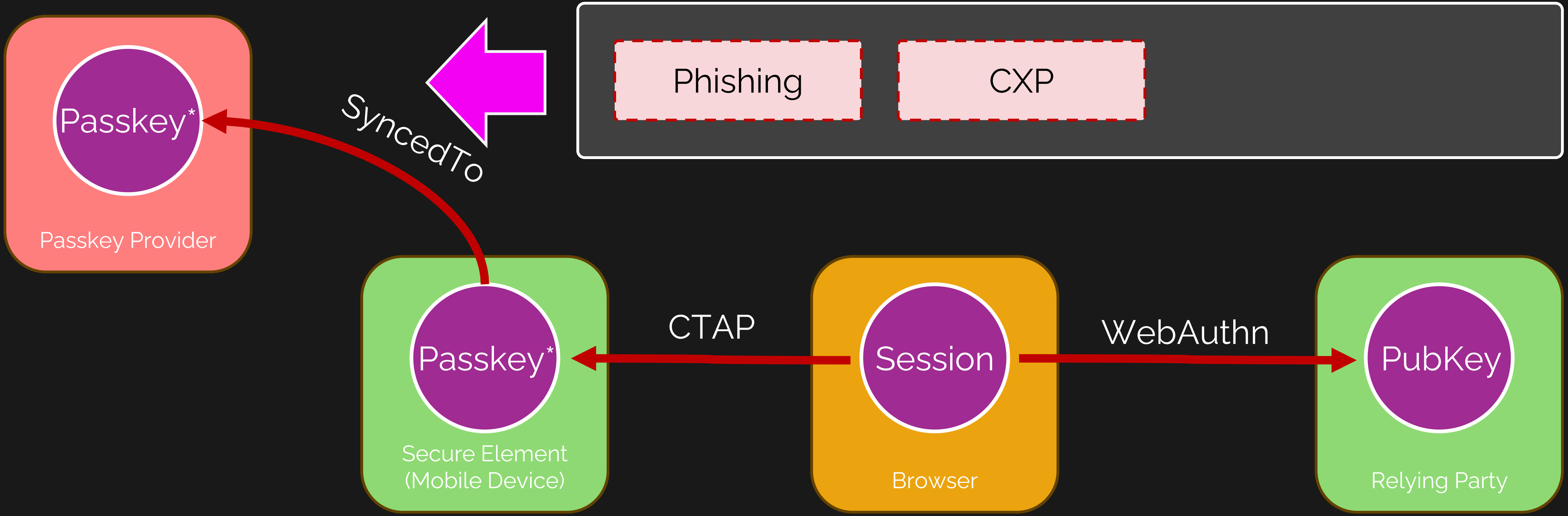
(Ab)use of exported passkeys



TokenTactics 0.2.21

```
PS C:\Users\Fabian\git\TokenTacticsV2> ipmo .\TokenTactics.psd1
PS C:\Users\Fabian\git\TokenTacticsV2> Invoke-EntraIDPasskeyLogin -KeyFilePath "C:\Users\Fabian\Downloads\testpasskeyexport\maija@c4a8korriban.com.passkey"
 -UserPrincipalName maija@c4a8korriban.com
✗ Loading key data from file: C:\Users\Fabian\Downloads\testpasskeyexport\maija@c4a8korriban.com.passkey
✓User:       maija@c4a8korriban.com
✓RP ID:      login.microsoft.com
✓Origin:     https://login.microsoft.com
✓CredID:     T1vHC2JZ9zsxXWz1hFlV6gkycRQV_gZN_CrA0It3gnU
✓UserHandle: T0Y6T2xo4yevIk-53gYvBbk6rFXfzPEAp8U36fYiav71Xh-5E7_wmxsc5MLX-foTZoNg
✗ Warming up session on login.microsoftonline.com (Authorize)...
✗ Validate FIDO2 Credential Type...
✓Challenge Received.
✗ Generating FIDO Assertion locally...
✗ Get required pre-information from microsoft.com...
✗ Submitting FIDO2 assertion to microsoftonline.com ...
✗ Submitting FIDO2 assertion to microsoftonline.com with sso_reload=true ...
✗ PageID: CmsiInterrupt
✗  Correlation Id: edda53ad-32a8-42ed-bf7e-ff281603bfca
✗  Session Id: 51e9f280-25d4-42b2-8361-7d4200f02c00
✗  Username: maija@c4a8korriban.com
✗  AADSTS50199: CmsiInterrupt
   For security reasons, user confirmation is required for this application: Microsoft Azure CLI.
✗  urlPost URL: /appverify
✗ Submitting CMSI response to microsoftonline.com ...
✓Login Successful!
⚿ESTSAUTH Cookie: 1.AXkAT2xo4yevIk-53g... saved as $global:ESTSAUTH
⚿Session saved as $global:webSession for reuse in other functions.
PS C:\Users\Fabian\git\TokenTacticsV2>
```

Attack vectors: Syncable passkeys

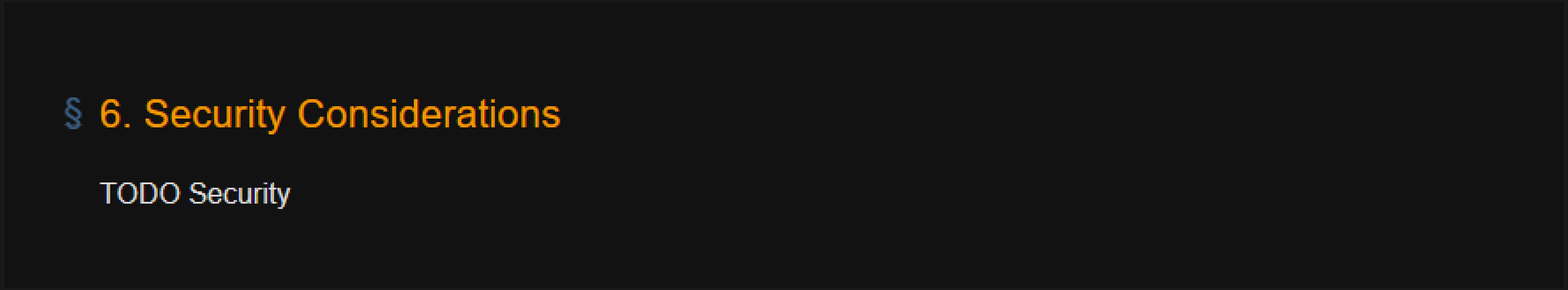


Same device
Cross-device (CDA)

*Passkey = Private Key

Credential Exchange Protocol

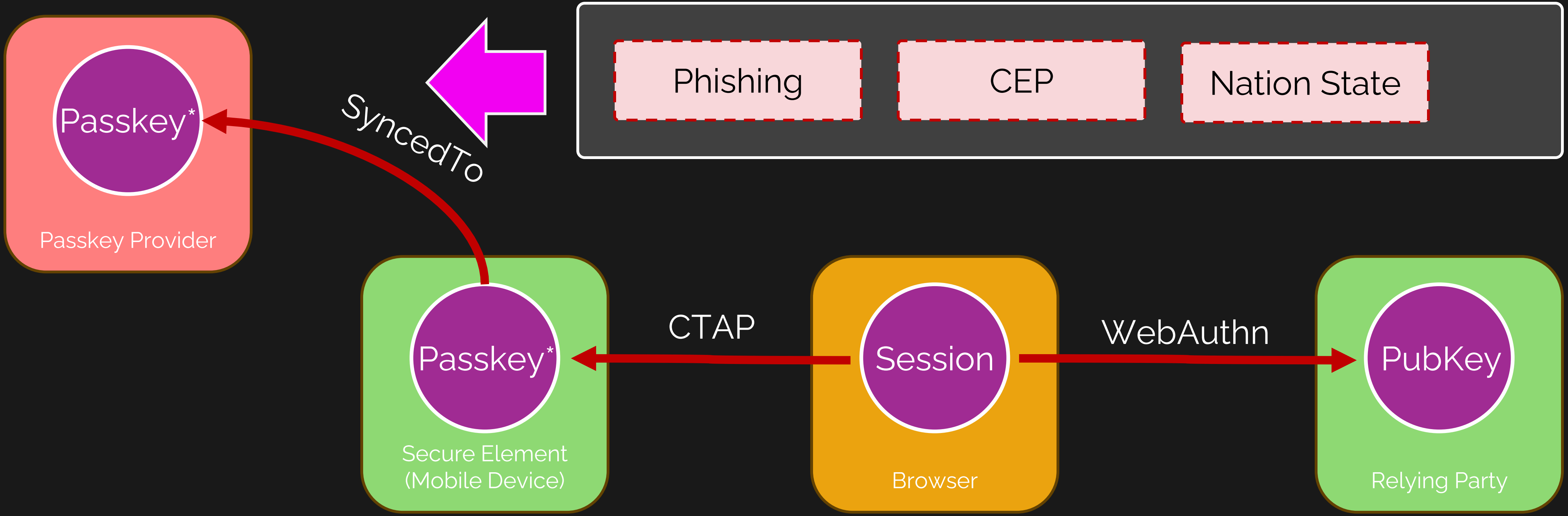
- Official method to transfer passkeys between providers
- Currently in draft



§ 6. Security Considerations

TODO Security

Attack vectors: Syncable passkeys



Same device
Cross-device (CDA)

*Passkey = Private Key

Mitigations

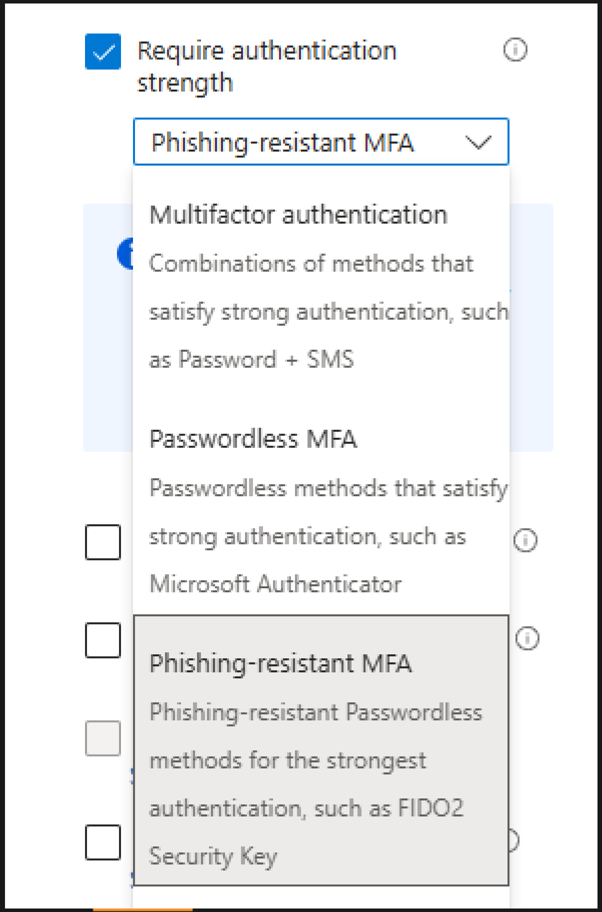
glueck kanja

Mitigations in Entra ID

- Enforce Conditional Access Authentication Strength to only allow Passkeys
  - Recovery: Allow Temporary Access Pass from trusted location
- Use Passkey profiles to target specific audiences
  - Device-bound passkey for Administrators and targeted accounts
  - Syncable passkey for others
- Enforce attestation for device bound passkeys
- Limit device bound passkeys to verified AAGUIDs

DISOBEY
THE NORDIC SECURITY EVENT

Mitigations in Entra ID



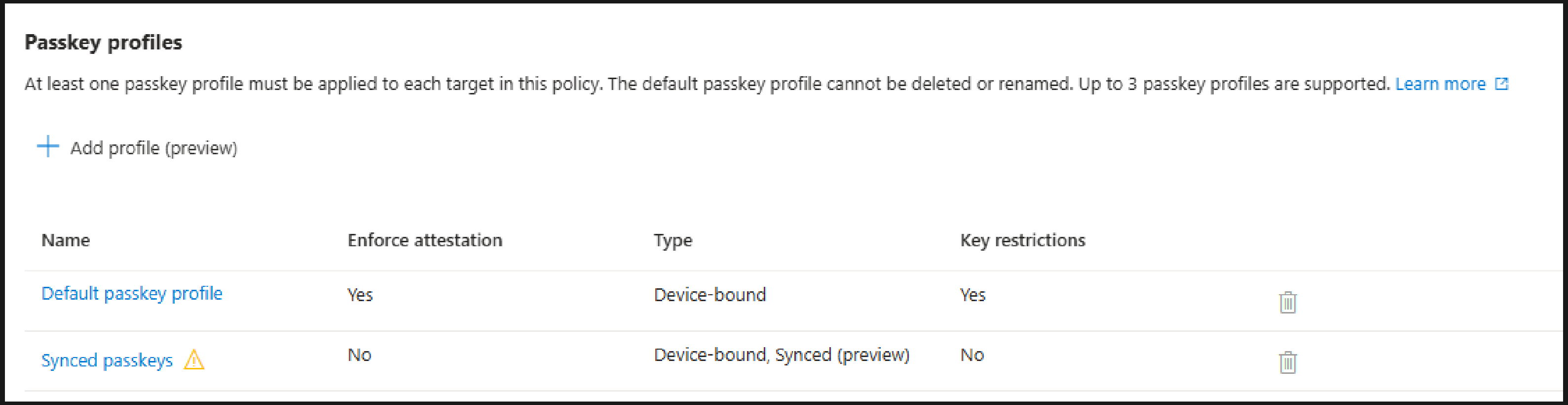
☑ Require authentication strength

Phishing-resistant MFA

Multifactor authentication
Combinations of methods that satisfy strong authentication, such as Password + SMS

Passwordless MFA
Passwordless methods that satisfy strong authentication, such as Microsoft Authenticator

Phishing-resistant MFA
Phishing-resistant Passwordless methods for the strongest authentication, such as FIDO2 Security Key

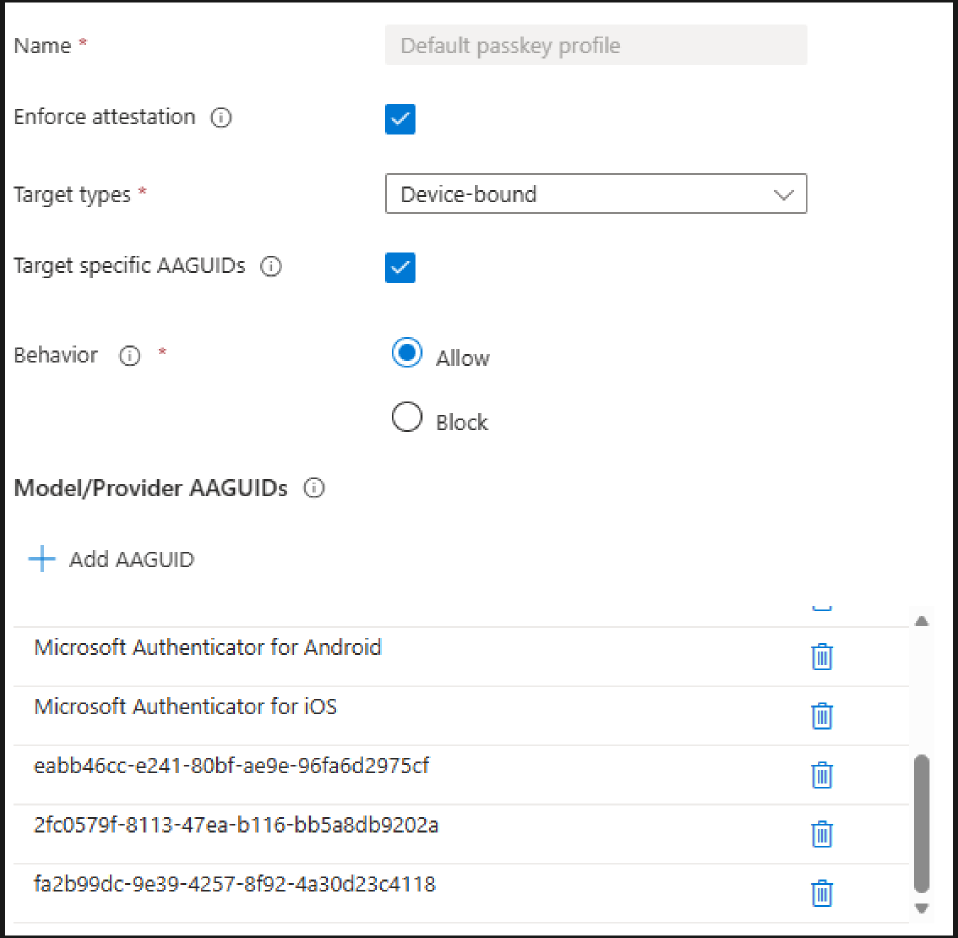


Passkey profiles

At least one passkey profile must be applied to each target in this policy. The default passkey profile cannot be deleted or renamed. Up to 3 passkey profiles are supported. Learn more

Name	Enforce attestation	Type	Key restrictions
Default passkey profile	Yes	Device-bound	Yes
Synced passkeys ⚠	No	Device-bound, Synced (preview)	No

Mitigations in Entra ID



Name *	Default passkey profile
Enforce attestation	✓
Target types *	Device-bound
Target specific AAGUIDs	✓
Behavior *	[✓ Allow] Block

Model/Provider AAGUIDs

+ Add AAGUID

- Microsoft Authenticator for Android
- Microsoft Authenticator for iOS
- eabb46cc-e241-80bf-ae9e-96fa6d2975cf
- 2fc0579f-8113-47ea-b116-bb5a8db9202a
- fa2b99dc-9e39-4257-8f92-4a30d23c4118

User mitigations

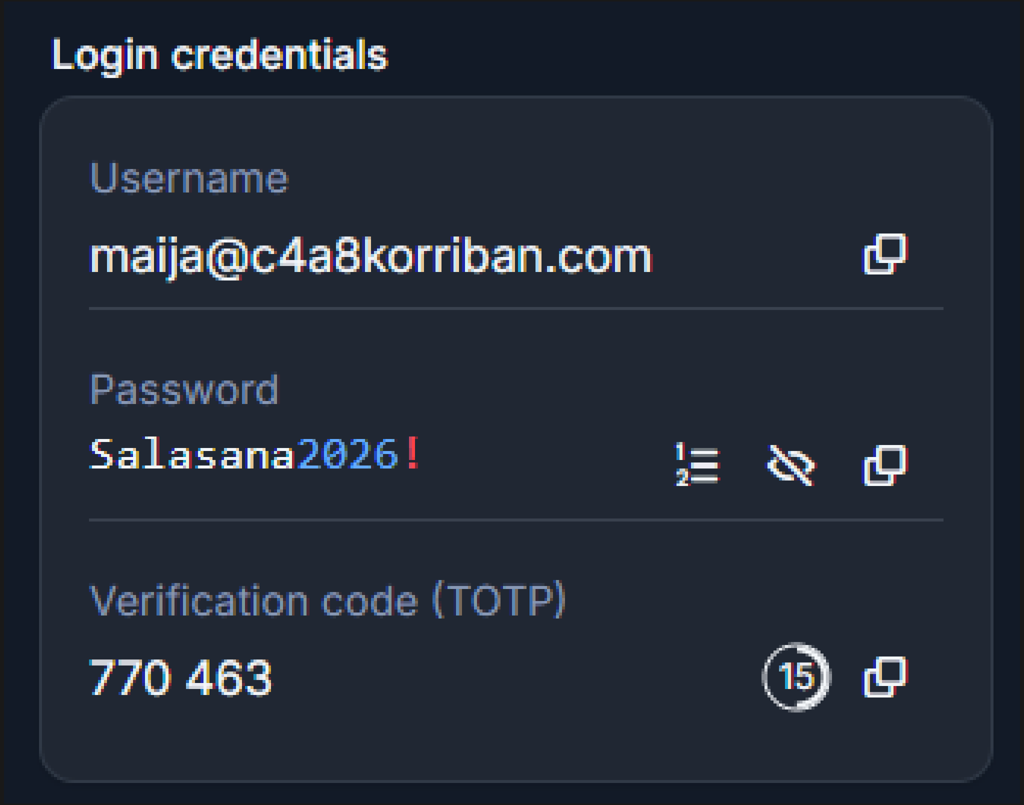
- Apple Advanced Data Protection
  - Fully end-to-end encryption of all data
  - Generated recovery key for own safe keeping
  - Stolen Device Protection forces biometrics outside of trusted places
- Google Advanced Protection Program
  - Requires >= 2 security key or passkey for sign-in
  - "Extra steps" for account recovery
  - Identity Check forces biometrics outside of trusted places

glueck kanja

Conclusion

glueck kanja

It's still 100x better than this...


Login credentials

Username
maija@c4a8korriban.com

Password
Salasana2026!

Verification code (TOTP)
770 463

DISOBEY
THE NORDIC SECURITY EVENT

Further information

- Your Passkey is Weak: Phishing the Unphishable
  - Chad Spensky, Ph D
  https://www.youtube.com/watch?v=xdl08cPDgtE
- Passkeys Pwned:Turning WebAuthn Against Itself
  - S Pratap Singh, J Lin, D Seetoh
  https://www.youtube.com/watch?v=GG4gAhbhPH8
- Google on attestation
  https://groups.google.com/a/fidoalliance.org/g/fido-dev/c/nhpxExcofb8/m/pd_SAJsnAwAJ
- https://www.corbado.com/blog/android-16-passkeys

Further information

- Manage passkeys in Chrome
  https://support.google.com/chrome/answer/13168025
- Entra ID: Enable passkeys (FIDO2) for your organization
  https://learn.microsoft.com/entra/identity/authentication/how-to-enable-passkey-fido2
- Assign a passkey or security key in the AWS Management Console
  https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable_fido.html

Further information

- https://github.com/f-bader/TokenTacticsV2/
- Passkeys: they're not perfect but they're getting better https://www.ncsc.gov.uk/blog-post/passkeys-not-perfect-getting-better

Thank You