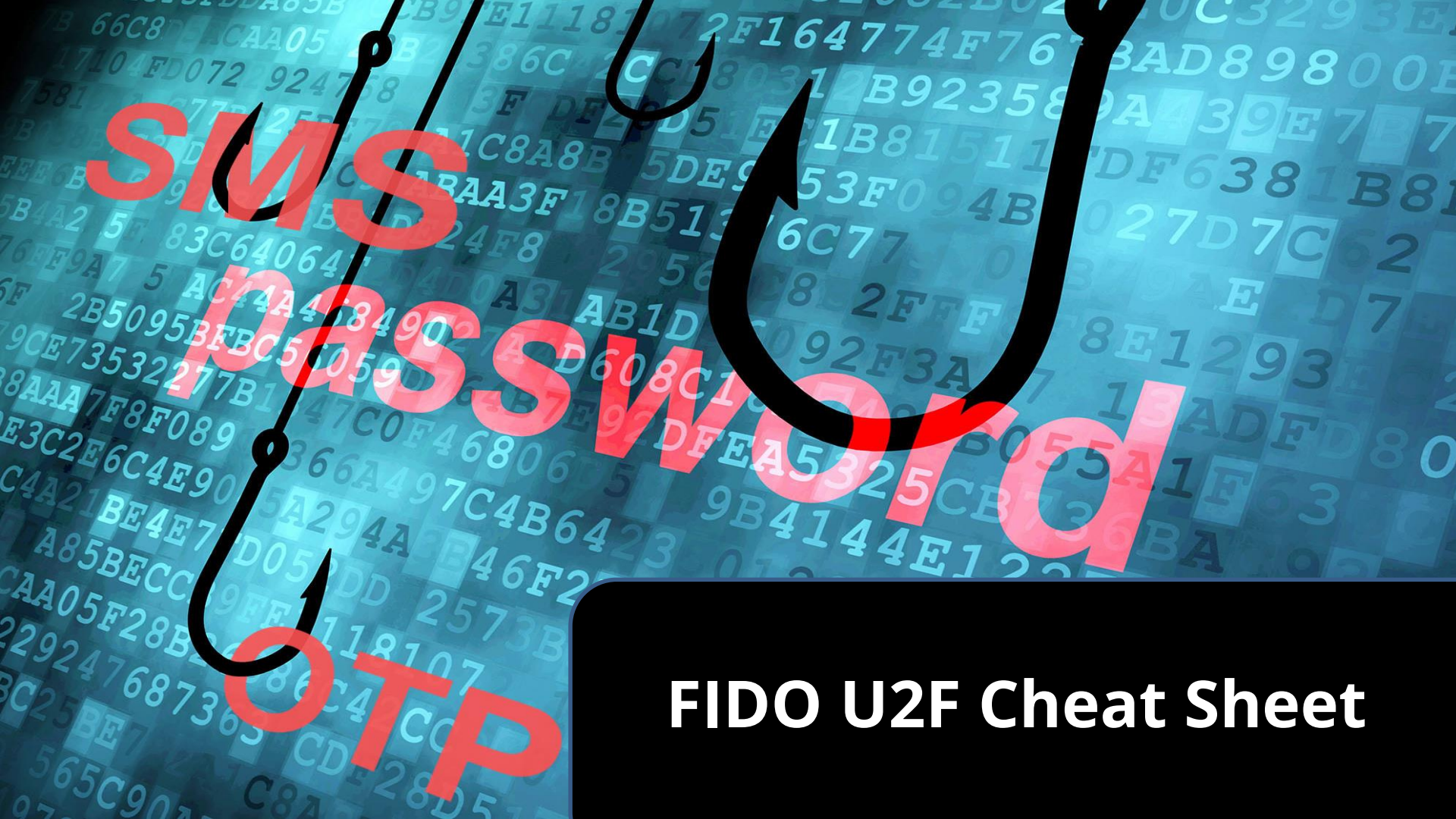
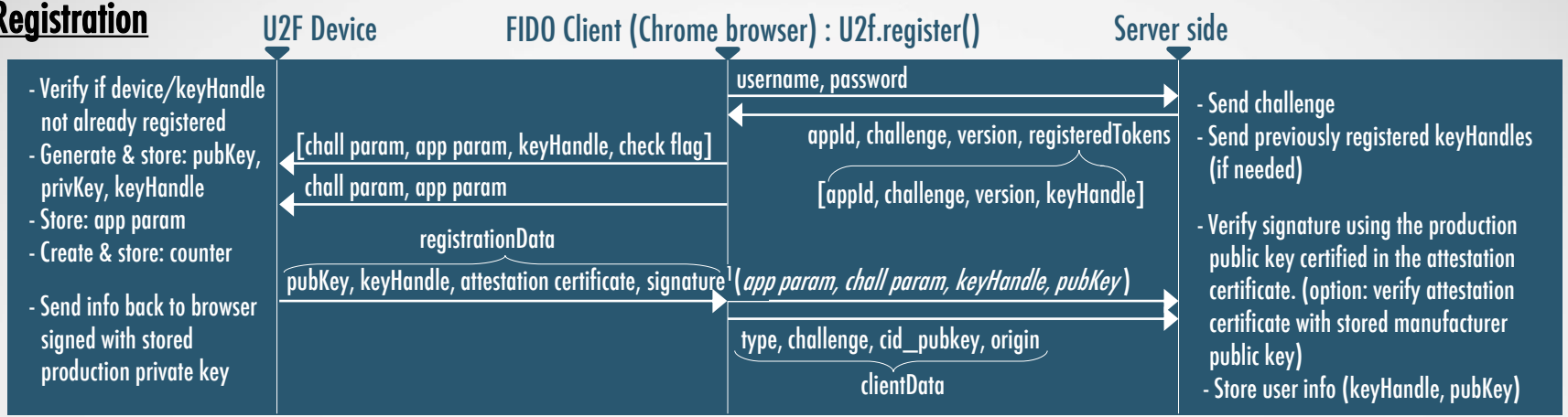


FIDO U2F Cheat Sheet

U2F Registration



pubKey: user public Key [65 bytes], stored on U2F device and authentication server
privKey: user private Key [32 bytes], stored on U2F device
keyHandle: unique Key index [up to 255 bytes[2]], generated by U2F device, can be purely random or used to wrap private keys related information (can be unsafe)
appId: application ID, also called origin, server domain name, i.e.: localhost, xyz.com [32 bytes]
app param (application parameter): SHA-256 hash of application ID [32 bytes]
challenge: random string, generated by the server [32 bytes here – no recommended default length]
chall param (challenge parameter): SHA-256 hash of clientData = type, challenge, origin (and cid_pubkey if optional TLS Channel ID protection is available – see details below) [32 bytes]
registeredTokens: contains challenges for every keyHandles a user already registered, inside an array of (appId, challenge, keyHandle, version)
check flag: All already registered keyHandles are used to simulate an authentication with U2F device. This flag means: just check if this keyHandle is not already present inside U2F device.
version: selected version of U2F (String "U2F_V2")
counter: per key pair [4 bytes], incremented every time an authentication occurs (U2F specifications let manufacturers choose to have a global counter or per key pair counters[3])

registrationData:
- **user public key**: [65 bytes]. (uncompressed) x,y-representation of a curve point on a P-256 NIST elliptic curve
- **keyHandle length byte** [1 byte], which specifies the length of the key handle
- **keyHandle** [length specified in previous field, 64 bytes by default here]. Unique ID of the generated key pair.
- **attestation certificate**: [max size: 2048 bytes] X.509 DER certificate. Same for all -or at least a large batch/number of- devices from a same manufacturer)(production public key + product info) signed by manufacturer private key

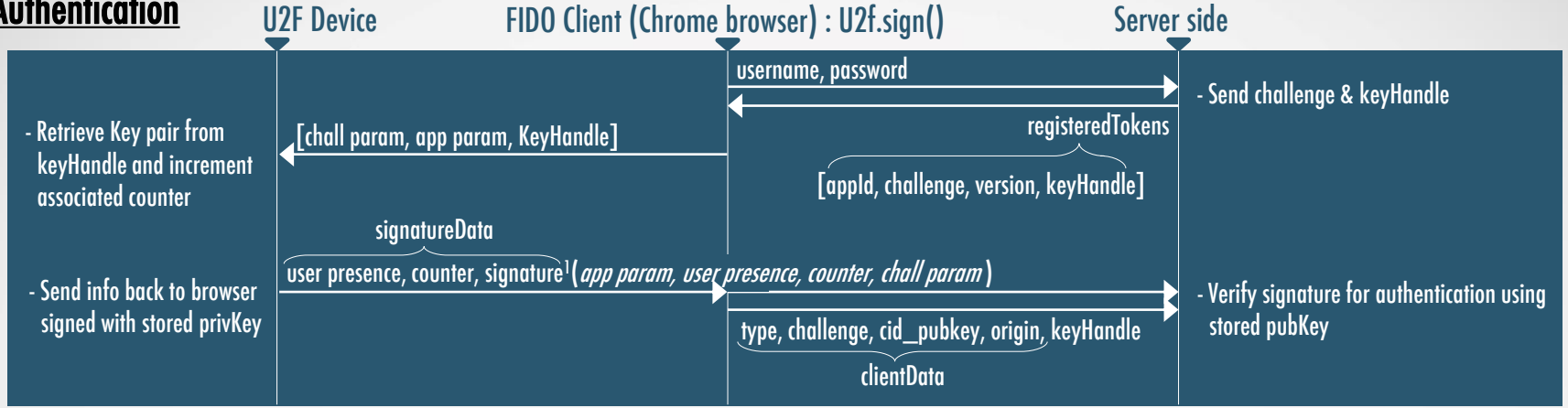
clientData:
- **type**: constant 'navigator.id.finishEnrollment' for registration
- **challenge**: random encoded string provided by the server
- **origin**: same as appId
- **cid_pubkey** (Optional, if browser and server support ID TLS extension): Channel ID public key used by browser to communicate with origin

[1] - **signature** : [max size: 72 bytes] ECDSA signature of (app param, chall param, keyHandle, pubKey) → signed with production private key

https://linkedin.com/in/frederic2

2 - For Keydo U2F product: Random 64 Bytes
3 - For Keydo U2F product: per key pair counters

U2F Authentication



pubKey: user public Key [65 bytes], stored on both U2F device and authentication server sides
privKey: user private Key [32 bytes], stored on U2F device side only
keyHandle: unique Key index [up to 255 bytes[2]], generated by U2F device, can be purely or used to wrap private keys related information (can be unsafe)
appId: application ID, also called origin, server domain name, i.e.: localhost, xyz.com [32 bytes]
app param (application parameter): SHA-256 hash of application ID [32 bytes]
challenge: random string, generated by the server [32 bytes here – no recommended default length]
chall param (challenge parameter): SHA-256 hash of clientData = type, challenge, origin (and cid_pubkey if optional TLS Channel ID protection is available – see details below) [32 bytes]
registeredTokens: contains challenges for every keyHandles a user already registered, inside an array of (appId, challenge, keyHandle, version)
version: selected version of U2F (String "U2F_V2")
counter: per key pair [4 bytes], incremented every time an authentication occurs (U2F specifications let manufacturers choose to have a global counter or per key pair counters[3])

signatureData:
- **user presence**: [1 byte]. Bit 0 is set to 1, which means that user presence was verified.
- **counter**: [4 bytes]. big-endian representation of U2F device authentication counter
[1] - **signature**: [max size: 72 bytes] ECDSA signature of (app param, user presence, counter, chall param)
→signed with privKey

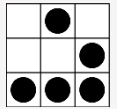
clientData:
- **type**: constant 'navigator.id.getAssertion' for authentication
- **challenge**: random encoded string provided by the server
- **origin**: same as appId
- **cid_pubkey** (Optional, if browser and server support ID TLS extension): Channel ID public key used by browser to communicate with origin

https://linkedin.com/in/frederic2

2- For Keydo U2F product: Random 64 Bytes
3- For Keydo U2F product: per key pair counters

Last version of this document is always available here:
https://www.neowave.fr/pdfs/FIDO-U2F-CHEAT-SHEET.pdf

More information on official FIDO U2F specifications:
https://fidoalliance.org/download/



Frédéric MARTIN
System & Security Architect

https://linkedin.com/in/frederic2
frederic.martin@neowave.fr

Document version 170707