

SECDSA: Mobile signing and authentication under classical “sole control”

Eric Verheul

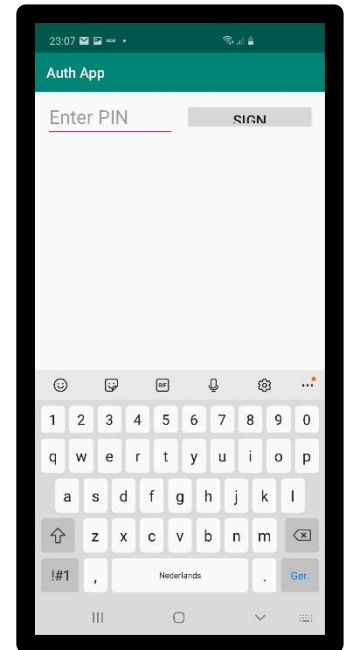
Eric.Verheul@keycontrols.nl

July 19 2021

Full paper: <https://eprint.iacr.org/2021/910> short url: <https://ia.cr/2021/910>.

Agenda

- Introduction and problem description
- Split-ECDSA (SECDSA) outline
- SECDSA applications
- Impression of cryptographic techniques



Introduction and problem description

The *highest* requirements in the eIDAS regulation 910/2014 are:

- A two-factor **Authenticator** ('authentication means') resistant to High attack potential ('eIDAS assurance level High'),
- A qualified Signature Creation Device (**SCD**) under *sole control* of the user (for optimal *non-repudiation*).

Draft eIDAS regulation (3-6-2021) introduces a **European Digital Identity Wallet**, an eIDAS *High* Authenticator/SCD allowing for:

- flexible and selective personal attribute disclosure under user control,
- qualified ('legally binding') signing.

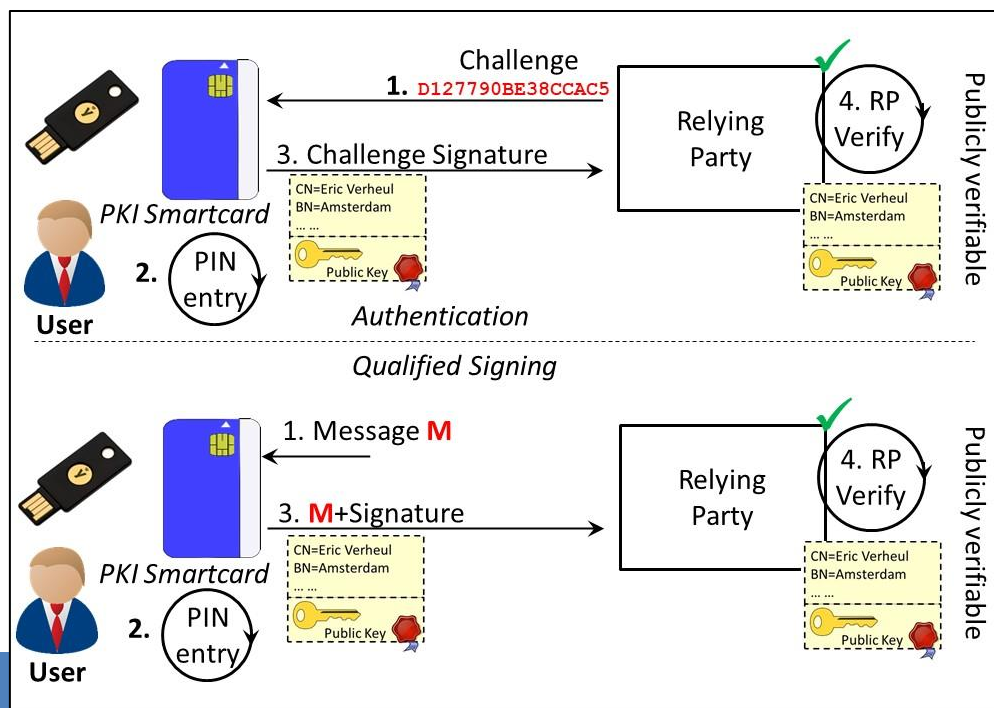
Introduction and problem description

Highest requirements in the eIDAS regulation 910/2014 :

- A two-factor Authenticator ('authentication means') resistant to High attack potential ('eIDAS assurance level High'),
- A qualified Signature Creation Device (SCD) under *sole control* of the user (for optimal *non-repudiation*).

Draft eIDAS regulation (3-6-2021) introduces a **European Digital Identity Wallet**, an eIDAS *High* Authenticator/SCD allowing for:

- flexible and selective personal attribute disclosure under user control,
- qualified ('legally binding') signing.
- A **PKI smartcard** is the *classical* solution for this; strong security/sole control, but expensive, user unfriendly and inflexible.



"Classical sole control and non-repudiation"

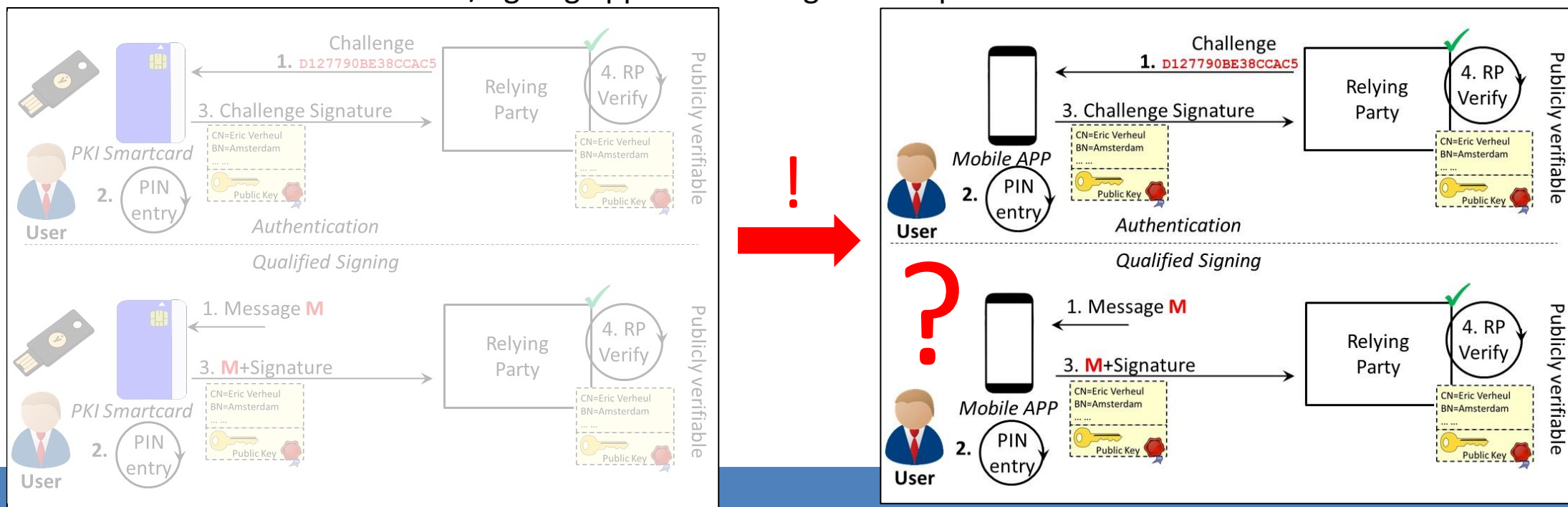
Introduction and problem description

Highest requirements in the eIDAS regulation 910/2014:

- A two-factor Authenticator ('authentication means') resistant to High attack potential ('eIDAS assurance level High'),
- A qualified Signature Creation Device (SCD) under *sole control* of the user (for optimal *non-repudiation*).

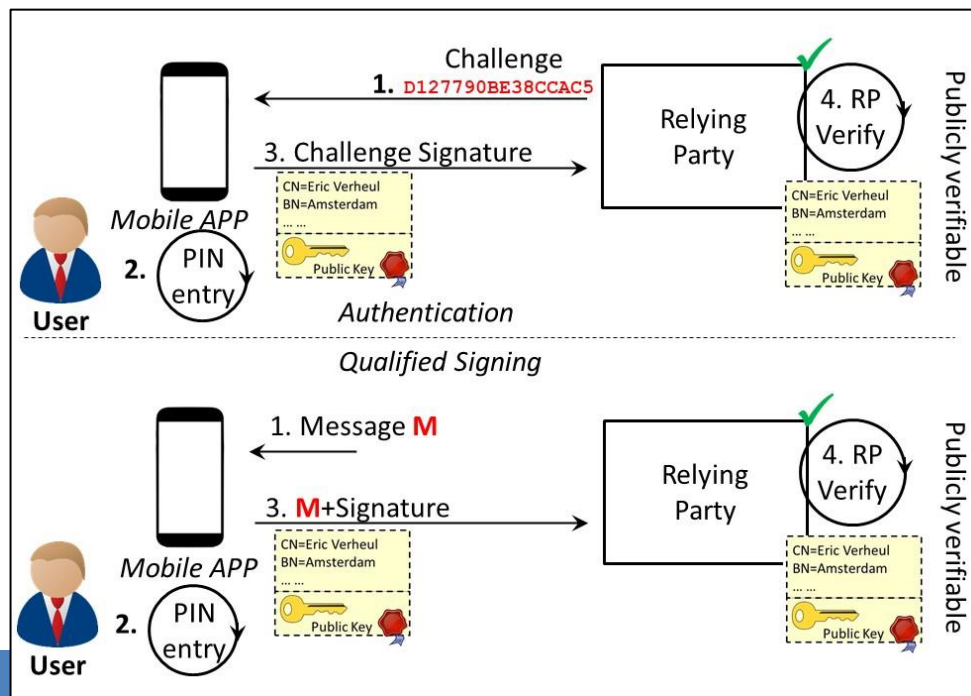
Draft eIDAS regulation (3-6-2021) introduces a **European Digital Identity Wallet**, an eIDAS *High* Authenticator/SCD allowing for:

- flexible and selective personal attribute disclosure under user control,
- qualified ('legally binding') signing.
- A **PKI smartcard** is the classical solution for this; strong security/sole control, but expensive, user unfriendly and inflexible.
- Users want *mobile* authentication/signing applications: High attack potential resistance and sole control is challenging!

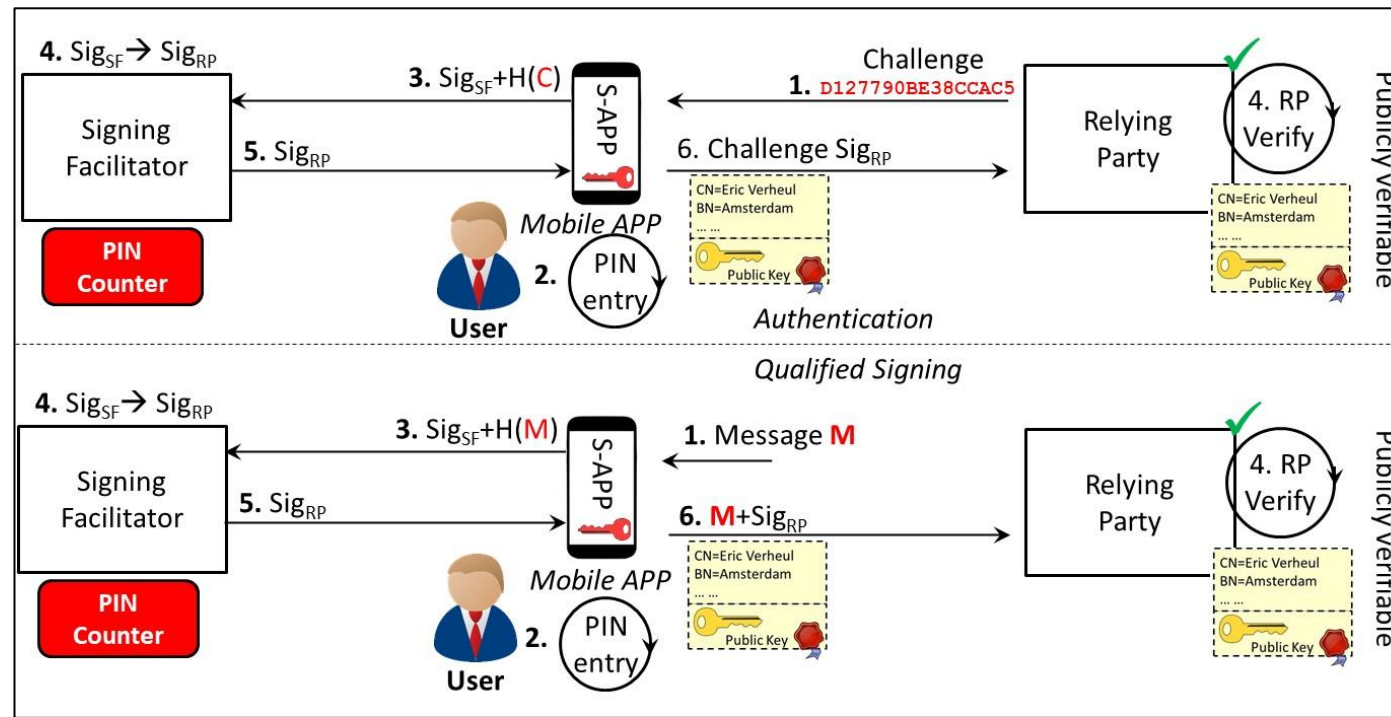


Introduction and problem description

- >90% mobile devices have cryptographic *hardware* support, but no support for authentication/signing APP PINs
- eIDAS High mobile authentication/ qualified signing application seems feasible judging from eIDAS notifications
- That is, mobile cryptographic *hardware* is not the bottleneck in meeting eIDAS High requirements!
- However, notified eIDAS High/qSCD APPs (Netherlands, Latvia and Belgium) are:
 - proprietary and security/sole control cannot be publicly assessed
 - based on a ‘man-in-the-middle’ design and do not have the same cryptographic security/sole control as PKI smartcards
- *Challenge: can we construct an eIDAS High mobile authentication/ qualified signing application with equivalent security/sole control as a PKI smartcard?*
- *SECDSA can accomplish this*



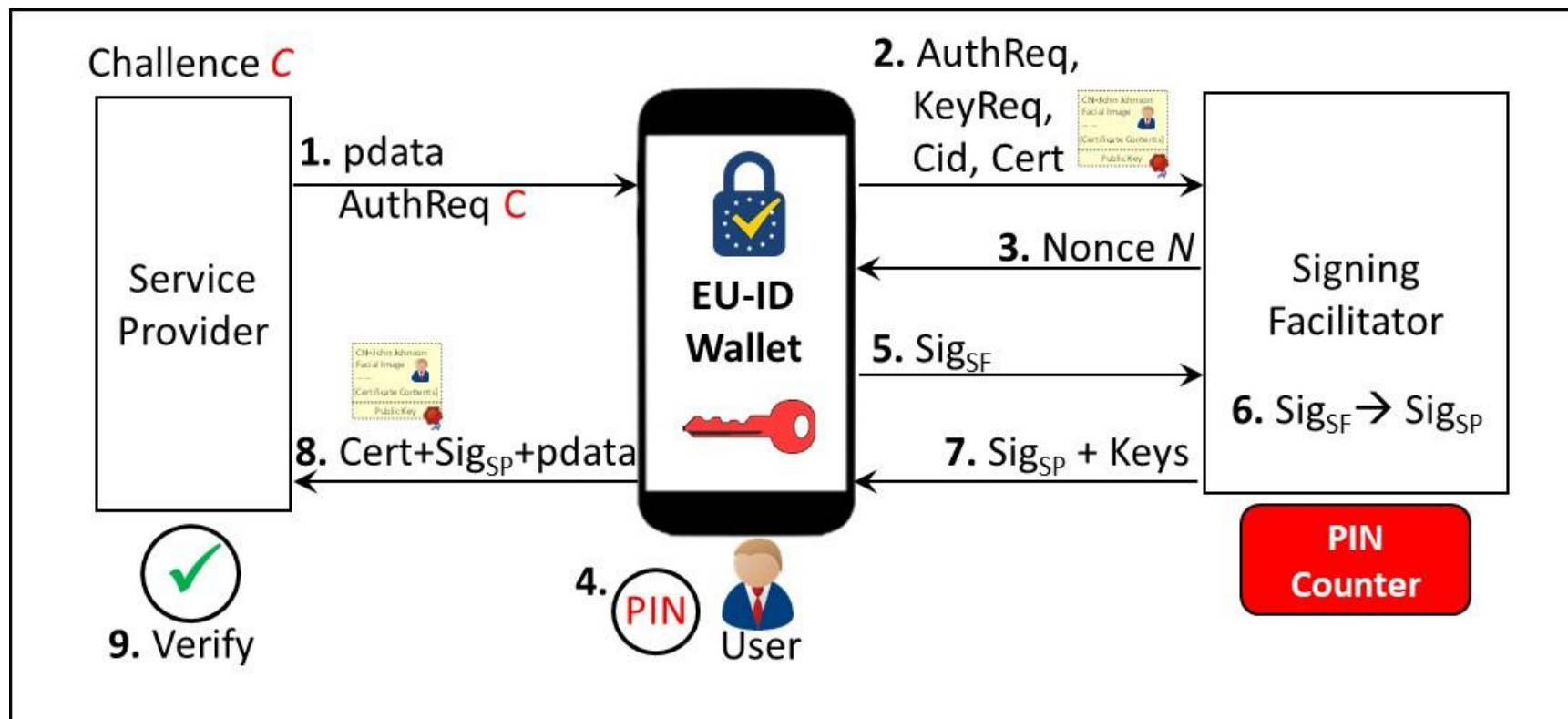
Split-ECDSA (SECDSA) outline



- SECDSA signatures are regular (ECDSA) signatures based on a signing key $k = k_1 \cdot k_2$, i.e. product of two private keys k_1, k_2 :
 - One key k_1 managed inside the (limited) device cryptographic hardware in non-exportable fashion ('possession key')
 - One key k_2 derived on the PIN entered by the user and another key inside the device cryptographic hardware in non-exportable fashion ('PIN/knowledge key')
- Using a secret key the SF can verify SECDSA signatures generated by APP but cannot generate them itself
- The SF can also transform SECDSA signatures generated by S-APP into public verifiable form as with a PKI smartcard
- The PIN is not stored at Signing Facilitator (SF) nor in the APP nor is information on which the PIN can be brute-forced
- SECDSA provides classical sole control as a PKI smartcard
- The SF is not a man-in-the-middle as in existing (notified) mobile solutions

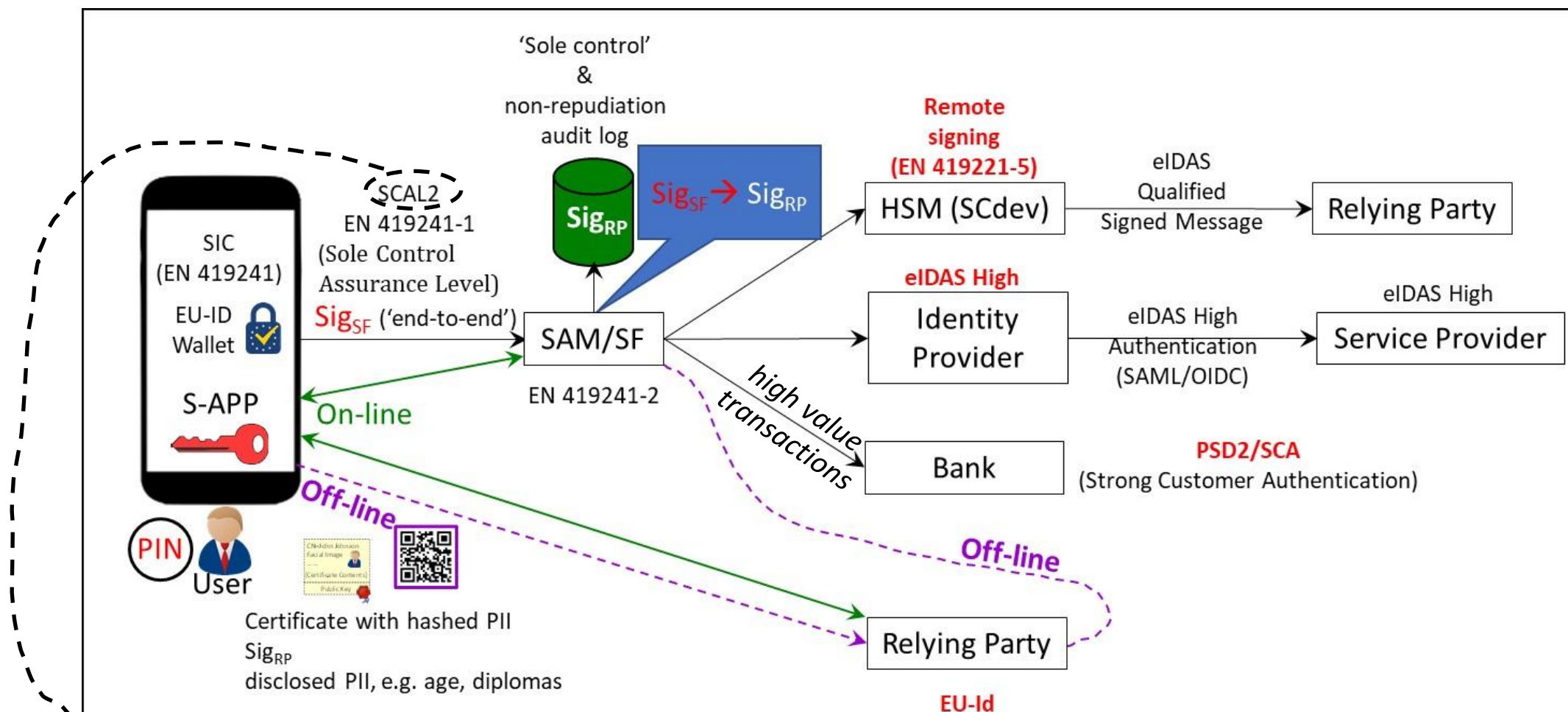
SECDSA applications: example

- SECDSA based EU-ID wallet (example): flexible and selective personal attribute disclosure under user control
- Crux:
 - use digital certificates binding the user SECDSA public key to *hashes* of attributes only
 - store attributes encrypted in the EU-ID wallet; keys retrievable from SF based on strong (SECDSA) authentication
- We can also let SF produce ‘conventional’ ECDSA user signatures; storing the SECDSA signatures at the SF for dispute handling only; probably best from a usability perspective.



SECDSA applications: overview

- A publicly verifiable SECDSA signature consists of two EC-Schnorr or six regular (ECDSA) signatures.
- In practice SECDSA signatures might be registered in a log file and used for dispute handling only.



“is aimed to achieve the same sole control assurance level as what would be achieved by a stand-alone QSCD as defined in the eIDAS [4] Regulation” (European norm EN 419241-1)

Impression of cryptographic techniques

Algorithm 6 Split-ECDSA (SECDSA) signature generation

Input: message M , SCE-key $u \in \mathbb{F}_q^*$, PIN-key $\sigma \in \mathbb{F}_q^*$

Output signature (r, s) .

- 1: Compute $\mathcal{H}(M)$ and convert this to an integer e .
 - 2: Compute $e' = \sigma^{-1} \cdot e \bmod q$
 - 3: Select random $k \in \{1, \dots, q-1\}$
 - 4: Compute $kG = (x, y)$ and convert x to integer $\bar{x}$
 - 5: Compute $r = \bar{x} \bmod q$. If $r = 0$ go to Line 1
 - 6: If $r \bmod q = 0$ then go to Line 1
 - 7: Compute $s = k^{-1}(e' + u \cdot r) \bmod q$. If $s = 0$ go to Line 1
 - 8: Compute $s' = \sigma \cdot s \bmod q$
 - 9: Return (r, s')
-

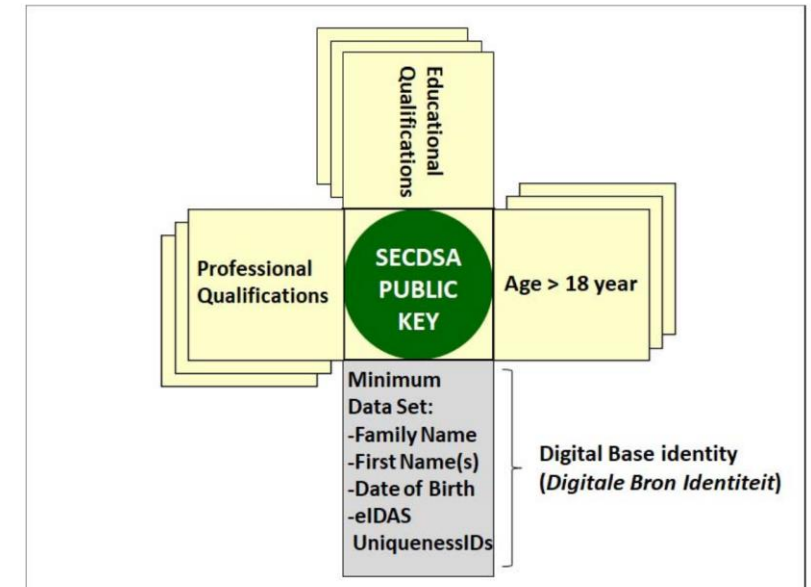


Figure 10. Flexible SECDSA certificates by shared public key use