Anonymous Credentials and the EUDI Wallet

Anna Lysyanskaya
Brown University

Based on the paper "Cryptographers' Feedback on the EU Digital Identity's ARF" by Baum, Blazy, Camenisch, Hoepman, Lee, Lehmann, Lysyanskaya, Mayrhofer, Montgomery, Nguyen, Preneel, shelat, Slamanig, Tessaro, Thomsen, Troncoso

EU Digital Identity Regulation

- https://eur-lex.europa.eu/eli/reg/2024/1183/oj
- “Fully mobile, secure and user-friendly” identity app.
- “§4. European Digital Identity Wallets shall enable the user, in a manner that is user-friendly, transparent, and traceable by the user, to:
  - (a) securely [..] authenticate to relying parties [..] while ensuring that selective disclosure of data is possible;
  - (b) generate pseudonyms and store them encrypted and locally within the European Digital Identity Wallet;
- “The technical framework of the European Digital Identity Wallet shall:
  - (a) not allow […any...] party […] to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, […] unless explicitly authorised by the user;
  - (b) enable privacy preserving techniques which ensure unlinkability, where the attestation of attributes does not require the identification of the user.”
- All member states must provide such an app to their citizens by 2026.

Cryptographers Get Involved

- June 5&6, 2024: EUDI Wallet Team of the European Commission held a (virtual) presentation of a proposed architecture (“ARF”) to cryptographers
- Spoiler alert: we didn’t like it!
- Our proposal: use anonymous credentials instead
- See our “Cryptographers’ Feedback” paper

The Original ARF and Why It Falls Short

- Try 1 (no privacy):
  - An Identity Provider (IdP) is associated with a signature verification key VK.
  - A user is associated with a public key PK of his device (SK is stored in secure hardware), and has identity attributes $a_1,\ldots,a_n$.
    (Identity attributes are, for example, name, date of birth, address, etc.)
  - A credential is the IdP's signature σ on $(PK,a_1,\ldots,a_n)$
  - A verifier ("relying party," or "RP") verifies σ
  - Nice feature: device binding – RP can verify that the user has possession of the device by requiring evidence of possession of SK

The Original ARF and Why It Falls Short

- The ARF is a modification in an attempt to achieve privacy:
  - An Identity Provider (IdP) is associated with a signature verification key VK.
  - A user is associated with a public key PK of his device (SK is stored in secure hardware), and has identity attributes $a_1,\ldots,a_n$.
    (Identity attributes are, for example, name, date of birth, address, etc.)
  - A credential is the IdP's signature σ on ~~$(PK,a_1,\ldots,a_n)$~~ $(h(PK,salt_0), h(a_1,salt_1), \ldots, h(a_n,salt_n))$
    - For unlinkability, σ can only be used once!
    - So need to issue a batch of single-use credentials, each with different random $(salt_0,\ldots,salt_n)$
  - A verifier ("relying party," or "RP") verifies σ on $h(PK,salt_0), h(a_1,salt_1), \ldots, h(a_n,salt_n)$
  - User can reveal whatever subset of attributes it wants
  - ~~Nice feature: device binding – RP can verify that the user has possession of the device by requiring evidence of possession of SK~~
- What's not to like?
  - Fails to ensure unlinkability between IdP and RP
  - Batch issuance is cumbersome, in practice apps might fail to do it

Anonymous Credentials

- June 5&6, 2024: EUDI Wallet Team of the European Commission held a (virtual) presentation of a proposed architecture (“ARF”) to cryptographers
- Spoiler alert: we didn’t like it!
- Our proposal: use anonymous credentials instead

- Anonymous credentials [Chaum84,…,CL01,Lys02,CamenischLysyanskaya02,CL04,…] consist of
    - (1) A commitment scheme with appropriate protocols
    - (2) A digital signature scheme with appropriate protocols

Anonymous Credentials

- (1) A commitment scheme with appropriate protocols
  - A non-interactive cryptographic commitment scheme Commit(attributes;$\text{rand}_{\text{attr}}$)
    - Hiding: Commit(attributes;$\text{rand}_{\text{attr}}$) reveals nothing about attributes
    - Binding: infeasible to find attributes ≠ attributes’, $\text{rand}_{\text{attr}}$, $\text{rand'}_{\text{attr}}$ such that
      Commit(attributes,$\text{rand}_{\text{attr}}$) = Commit(attributes’, $\text{rand'}_{\text{attr}}$)
  - Efficient proof protocols for committed values:

Let **P** = {P(attributes)} be a family of predicates that correspond to access control policies.

For example, age or residency verification.

For each P in **P**, we need a zero-knowledge proof of knowledge of the witness for the relation

R_P = {(C,w) | w = (attributes, $\text{rand}_{\text{attr}}$) such that
C = Commit(attributes, $\text{rand}_{\text{attr}}$) AND P(attributes) = TRUE}

Anonymous Credentials

- (2) A digital signature scheme with appropriate protocols
  - A digital signature scheme (KeyGen, Sign, VerifySig)
  - A secure issuing protocol between User(VK,attributes,$\text{rand}_{\text{attr}}$) and Signer(SK,C) where
    - IF SK corresponds to VK and C = Commit(attributes,$\text{rand}_{\text{attr}}$)
    - THEN User's output is σ = Sign(SK,attributes), Signer's output is Accept
    - ELSE both output Reject

Secure = each party just learns their output and nothing else

- The ZK-show protocol: A zero-knowledge proof of knowledge of the witness for the relation
  R = {((C,VK),w) | w=(attributes, $\text{rand}_{\text{attr}}$, σ) such that
  C=Commit(attributes, $\text{rand}_{\text{attr}}$) AND VerifySig(VK,attributes,σ) = TRUE}

Plugging in Anonymous Credentials

- An Identity Provider (IdP) is associated with a signature verification key VK.
- A user is associated with a public key PK of his device (SK is stored in secure hardware), and has identity attributes $a_1,\ldots,a_n$.
  (Identity attributes are, for example, name, date of birth, address, etc.)
- A credential is the IdP's signature σ on (~~PK,~~SK,$a_1,\ldots,a_n$). It is issued via the secure issuing protocol where IdP's input is C=Commit((SK,$a_1,\ldots,a_n$), rand).
- A verifier ("relying party," or "RP") ~~verifies σ~~ takes as input C' and runs the ZK proof protocols with the user to verify that user knows attributes=(SK,$a_1,\ldots,a_n$,rand') and σ such that
  (0) C'=Commit(attributes, rand')
  (1) attributes satisfy RP's access control policy P (using the ZK proof for R_P)
  (2) VerifySig(VK,attributes,σ) = TRUE (using the ZK-show protocol)
- Nice feature: device binding – RP can verify that the user has possession of the device ~~by requiring evidence of possession of SK~~ because ZK proof of knowledge of SK is included

The Fine Print

- Which commitment scheme, signature scheme, and protocols to plug in?
- How to make them compatible with existing technology for device binding?

Which Commitment, Signature, and Protocols?

- For any commitment scheme, there exist appropriate secure protocols that turn them into anonymous credentials. Can use general ZK proofs [GMW87,…,Ligero22,Testudo23]

- In practice, we might want to use something else:
  - A solution created for this specific application can be more efficient
  - Want a standardized approach

- ”Cryptographers’ Feedback” paper suggests using BBS+ [BBS+CL04,…,TZ23]
  - Known for 20+ years, a lot of people attention and peer review
  - Reasonably efficient, small overhead over our “Try 1”
  - IETF draft standard (community input would be helpful)
  - Challenge: how to migrate from “Try 1” to BBS+ based credentials without upgrading hardware for device binding. I.e. currently SK residing in hardware is an EC-DSA SK.

- Other efforts:
  (1) Use EC-DSA and customize a system like Ligero22 or Testudo23 to work for it [Google].
  (2) Modify BBS to accommodate an EC-DSA-based secure element [Orange].

Finally the Math for BBS [TessaroZhu23]

- Bilinear setup: groups $G_1 = <g_1>$, $G_2 = <g_2>$ of order q, bilinear map e into G_T, other generators $h_1,\dots,h_k$ of G_1 for signing k attributes
- Key generation: secret key x <- Z_q, VK = g_2^x
- $C = g_1 h_1^{a1} \dots h_k^{ak}$ is a compact representation of all the attributes.
  If a_k is random, then it's a non-interactive commitment (Pedersen commitment)

- Signature on attributes $(a_1,\dots,a_k)$ is (A,ε) such that $e(A,X) = e(B,g_2)$ where $B=CA^{-\varepsilon}$

- Can issue the signature without learning attributes: signer receives the commitment C (user's proved knowledge of opening), picks ε, computes $A = C^{1/(x+\varepsilon)}$

- ZK proof of knowledge: NOTE: if $A' = A^r$ and $B' = B^r$ then $e(A',X) = e(B',g_2)$
  ZK protocol: (1) reveal A' and B' to the verifier
  (2) prove knowledge of $r, a_1,\dots,a_k, \varepsilon$ such that $B' = g_1^r (h_1^{a1} \dots h_k^{ak})^r (A')^{-\varepsilon}$
  using standard (Schnorr-type) proof of knowledge of representation

Conclusions

- The future is now! And we are in a position to shape it.
  - The EC’s approach for soliciting feedback works
  - Similar efforts in the US – thank you, NIST, for staying in touch!
  - If we don’t weigh in, policy makers will adopt bad solutions ☹
  - Even if we do, there are still challenges

- Hard, but not unsolvable problems for Digital Identity
  - Device binding, either with EC-DSA or by upgrading hardware
  - Standardization
  - Public awareness and understanding that it’s possible to ensure identity proofing even while guaranteeing privacy