A privacy-preserving authentication service using mobile devices

Mihai Togan
Security Software Architect
certSIGN
Context

Mobile devices
- Present everywhere

High performance
- Computing power
- Security features

2 factors authentication
- Good candidates
- Easy to use

What to consider

Credentials transfer
- QR codes

Authentication protocols
- UProve
- FIDO
Agenda

1. QR-based Authentication
   • QR-based Authentication using PKI

2. U-Prove & FIDO protocols

3. Privacy-preserving Authentication
   • FIDO Attribute-based Authentication
   • FIDO Authentication with Privacy-preserving

4. Use-cases

5. Conclusions
QR-based Authentication

- **The aim**: authentication and authorization for the user on the Service Provider’s web application
- Credentials stored on the mobile device
- Identity transfer from the user’s mobile phone to his desktop by using QR codes
- **Authentication phase** (user’s mobile – QR Authenticator)
- Multiple solutions:
  - **PKI**, TLS, FIDO, custom
- A *Honest* QR Authenticator adds privacy-preserving for the user
QR-based Authentication using PKI

- **PKI authentication of the user to QR Authenticator**
  - Secure key storage (Android/iOS key store, hardware secure element)
  - Digital signatures on the smartphone

- **Authentication process**
  1. The user signs the QR code content using **his private key** and **certificate** stored **on the mobile phone**
  2. The phone sends the signed content and his certificate to the QR Authenticator server using a special connection
  3. The QR Authenticator server verifies the digital signature, the content and the digital certificate of the user
     - If verification succeeds, the user is granted an access token which will be sent to the Service Provider
QR-based Authentication using PKI, cont.

- Two-factor authentication mechanism
  - The smartphone (something the user has)
  - The password to access the certificate from the smartphone (something the user knows)
    - A hardware secure element can be used to protect the private key

- Problems...
  - A PKI infrastructure needed
  - CA to manage the users’ certificates
  - Requires digital certificates on mobile device
  - Users’ enrollment to get certificates
  - Users’ certificates path validation
  - Users’ certificates revocation status checking (OCSP service)

- Best fit in PKI enabled environments
U-Prove

- Attribute-based cryptographic protocol providing user’s privacy
  - Maintained by Microsoft

- Three entities involved
  - The user (the prover)
  - The issuer – issues attribute containers
  - The verifier – verifies user’s proofs (attributes)

- Two main protocols
  - **Issuing protocol** (issuer ↔ user)
    - Issuing the *Token Information* (TI) including user’s attributes
  - **Presentation protocol** (user ↔ verifier)
    - Proving user’s attributes validity & the user’s private key ownership
    - Proof generation sub-protocol (user – device)
    - Proof verification sub-protocol (user – verifier)

- Main idea: disclose only the required attributes to verifier
  - *Unlinkability*
  - *Untraceability*
U-Prove (cont.)

• On the server-side
  • Issuer (web-app)
  • Verifier (web-app)
  • REST API interface
  • WS-Trust Serialization [Paq11]

• On the user-side
  • Android application
  • U-Prove attributes stored as blobs in the application database

FIDO

• Passwordless authentication framework
  • FIDO Alliance (great support)
  • UAF, U2F

• FIDO entities
  • FIDO server (server-side)
  • FIDO client (client-side)
  • FIDO authenticator (client-side, trusted HW device)

• FIDO protocols: registration, authentication, deregistration
  • Generate user RSA key-pair
  • Challenge-response protocol
    • User unlocks his private key using various protection mechanisms

• Protocol messages
  • Extensions (used in our work to include attributes in FIDO)

• FIDO extension (not Extensions!)
  • FIDO Attribute-based Authentication
  • FIDO Authentication with Privacy-preserving
FIDO Attribute-based Authentication

• Combine the FIDO and U-Prove
  • With FIDO: user authentication
  • With U-Prove: user authorization (based on attributes)
    • Improved security layer on the server side
    • Granular access

• FIDO extended version
  • FIDO UAF standard messages (not modified)
  • Usage of FIDO extensions to carries user’s attribute info
  • Server asks the required attributes using AuthenticationRequest
  • The client responds with U-Prove proofs in AuthenticationResponse
    • Attributes are embedded in Response extensions
FIDO Attribute-based Authentication

Dictionary AuthenticationRequest {
  required OperationHeader header;
  required ServerChallenge challenge;
  Transaction[ ] transaction;
  required Policy policy;
}

Authentication Request
FIDO server → User

Dictionary AuthenticatorSignAssertion {
  required DOMString assertionScheme;
  required DOMString assertion;
  Extension[ ] exts; /* Serialized U-Prove proof */
}

Authentication Response
FIDO ASM → FIDO server

Dictionary Extension {
  required DOMString id; /* Bind to 'U-Prove - attribute' */
  required DOMString data; /* Required attribute encoded as base64 */
  required boolean fail_if_unknown; /* Bind to true */
}
FIDO Attribute-based Authentication

- Does not provide:
  - Unlinkability
  - Untraceability
FIDO Authentication with Privacy-preserving

- The user doesn't trust the FIDO server
- Unlinkability and Untraceability are required
- FIDO and U-Prove logic are separated
  - Step 1 (UP-Issuing), after FIDO authentication. User receives:
    - "authenticated" attribute
    - "validity timeframe" attribute
  - Step 2 (UP-Presentation): U-Prove authorization
    - User presents attributes to U-Prove verifier
  - Get a K-anonymity scheme
Use cases

• Pilot implementation – ReCRED project
  – Access to campus resources
    • Registered professors and students
    • Granting access to guests
  – Access to on-line restricted content
    • 18+
    • Legislation can be enforced

www.recred.eu
Conclusion

• Mobile devices are used for 2 factor authentication
  – Credential transfer – QR codes
  – Authentication protocols
    • U-Prove
    • FIDO
• Combination of authentication protocols
  – Easy to use (FIDO)
  – Privacy preserving (U-Prove)
    • Untraceability
    • Unlinkability
• Pilot implementation – ReCRED project

• Next steps
  – Implementation using TEE equipped hardware
Thank you!