Fast IDentity Online with Anonymous Credentials (FIDO-AC)

Wei-Zhu Yeoh*, Gunnar Heide, Lucjan Hanzlik

Michal Kepkowski*, Dali Kaafar,

*CISPA Helmholtz Center for Information Security
Macquarie University

*Equal contribution
Authentication

Proof

User

Resources

Relying Party
Authentication

User

Relying Party

Password

Passwordless

Resources
Authentication

User

Certified Attribute

Relying Party

Resources

Motivation
Authentication

Data Minimization ❌
Selective Disclosure  ❌

Raw Passport Data

User

Resources

Relying Party
Challenge

fido™ + Attr

Efficient

Compatible

Deployment

eID
ID: ABCD1234
NM: ABC
DOB: 1/1/2000
Contribution

fido™ +

eID
ID: ABCD1234
NM: ABC
DOB: 1/1/2000

DOB: 1/1/2000

Efficient
Compatible
Deployment
FIDO-AC Overview

FIDO-AC

- eID
- Data & Liveliness
- FIDO-AC Application
- Mediator

Standard FIDO2

- FIDO Client
- FIDO Assertion & AC Extension
- Relying Party
- FIDO2 Authenticator

FIDO2
1. Certified Attribute
In our instantiation, ICAO MRTD (e.g., eID, ePassport)

2. Mediator
For generically interfacing with different eID frameworks.

3. FIDO2 Extension
For binding with FIDO.
Identity Management: Over 140 Countries Bootstrapped with Certified Attributes

ICAO eID

Certified Attributes Sources
Anonymous Credential

Approach 1: Pure Server-side Verification

- Verify certificates
- Verify liveliness

Cross-domain User Linkability Violates Attributes Privacy
Anonymous Credential

Zero-Knowledge Proof

ZKP.Prove( CRS, statement, public-inputs, private-inputs) = Proof

ZKP.Verify( CRS, statement, proof, public-inputs) = True/False

Completeness: Verifier is convinced by correct proof.
Soundness: Prover cannot prove false statement.
Zero-Knowledge: Verifier learns nothing more than the statement.
Anonymous Credential

Approach 2: Local Proof-of-Interaction and Proof-of-Attributes

\[ \text{ZKP.Prove( } \text{PA( } \text{PASSPORT} \text{ ) } + \text{CA( } \text{PASSPORT} \text{ ) } = \text{ ZKP} \text{)} \]

Not Efficient + Deniable Transcript

\[ \text{ZKP.Verify( ZKP } ) \]
Introducing Mediator

- Don’t learn about server policy.
- Don’t learn about eID attributes.
- Generic Solution: eID agnostic.
Mediator

1. Verify(

   PA( Passport ) + CA ( Passport )

   Obtained from

2. Sign( \( H(\text{AttributesDigest} \ || \ \text{Randomness}) + \text{FIDO Challenge} \) )

   Sign( H(AttributesDigest || Randomness) + FIDO Challenge )

   Obtained from

   Mediator

   Mediator
Relying Party

1. \[\text{Sign.Verify(Data, Proof)}\]

Mediator Sign

2. \[\text{ZKP.Verify(Data, Proof)}\]

Extract Randomized Attribute Digest as Public Inputs
Zero-Knowledge Proof

**Prove**

\[
\text{Data} = H(\text{AttributesDigest} || \text{Randomness})
\]

and

\[
\text{AttributesDigest} = H(\text{bit}_0, ..., \text{bit}_n)
\]

and

\[
\text{Attribute} = \text{Parse}(\text{bit}_i, ..., \text{bit}_j)
\]

and

\[
\text{True} == \text{Policy-Sat}\text{(Attribute)}
\]
FIDO2 Extension

Client

FIDOChal = FIDOChal || AC

Append relying party challenge with the AC data

Certified Attribute

Certified Attribute

Authenticator

Authenticator

Attr

Attr
FIDO-AC Summary

FIDO-AC

- eID
- Data & Liveliness
- FIDO-AC Application
- Mediator

Standard FIDO2

- FIDO Client
- FIDO2 Authenticator
- FIDO Assertion & AC Extension
- Relying Party
Security Analysis

- Introduced passwordless authentication (PA) model with attribute (PAwA) based on PA model from Hanzlik' SP23.
- FIDO with attribute without mediator
- Due to compatibility and efficiency:
  - Introduced PAwA with mediator (PAwAM)

FIDO Security

- Impersonation Security
- Unlinkability
- Attribute Unforgeability
- Origin Privacy
- Attribute Privacy

Attribute Extension
## Mediator Thread Model

<table>
<thead>
<tr>
<th>Feature</th>
<th>Mediator-Verifier</th>
<th>Mediator-Prover</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unlinkability</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>None:</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>TEE:</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>C-TEE:</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Attribute Unforgeability</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* - For ICAO eID, other eID might achieve stronger property.
Proof-of-Concept Implementation

### Android Mobile App
- ePassport Interaction.
- Local Mediator backed by Android Key Attestation.
- Groth16’ ZKP using rust-arkwork library

### FIDO-AC Server
- Trusted Setup for Groth16’
- JavaScript fidoac.js that bridges communication between FIDO-AC Mobile App and Relying Party logics.

### Relying Party Server
- Standard FIDO verification with updated challenge state and dockerized FIDO-AC extension verification.
## Performance Evaluation

<table>
<thead>
<tr>
<th>Operation</th>
<th>Platform</th>
<th>Time (ms)</th>
<th>SD (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>eID Reading</td>
<td>Mobile</td>
<td>1059.4/0.0&lt;sup&gt;cached&lt;/sup&gt;</td>
<td>37.58</td>
</tr>
<tr>
<td>Liveliness Check</td>
<td>Mobile</td>
<td>738.92</td>
<td>47.06</td>
</tr>
<tr>
<td>ZK Verify</td>
<td>Cloud PC</td>
<td>8.19</td>
<td>0.29</td>
</tr>
<tr>
<td>ZK Prove</td>
<td>Mobile</td>
<td>3375.61&lt;sup&gt;*&lt;/sup&gt;</td>
<td>95.25</td>
</tr>
</tbody>
</table>

Mobile - Pixel 6 Pro
Cloud PC - a Standard D4s v3 Microsoft Azure Cloud Instance
* - Preprocessing possible
FIDO-AC

Fast Identity Online with Anonymous Credentials

Implementation demo
Summary

• FIDO-AC Framework
• Combining FIDO, eID and ZKP to create FIDO-AC
• Practical and privacy-preserving.
• Proof-of-Concept Implementation: