Privacy-Preserving Single Sign-On

Attributes & Blindness Workshop @ Eurocrypt 2024
26.5.2024

Anja Lehmann
Hasso Plattner Institute | University of Potsdam
Single Sign-On | Convenient User Authentication

- Authentication outsourced to Online Identity Provider
- User → IdP: password/2FA. Single pwd, no credentials/keys!
- User → RP: relayed ID token signed by IdP

1) Access request
2) Redirect to IdP
3) Request Auth to RP & User Auth to IdP
4) ID Token
5) ID Token

Only needs public key of IdP

Only needs single pwd/2FA towards IdP → authentication to many RPs
Single Sign-On | Strong User Authentication

- ID Token signed by IdP → security through unforgeable signatures
  - attests necessary user information
  - bound to session & RP

Properties | SSO
--- | ---
Usability | ✓
Strong Authentication | ✓
IdP knows several verified user attributes & attests only **minimally necessary** user information.

**Properties**

<table>
<thead>
<tr>
<th></th>
<th>SSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>✔️</td>
</tr>
<tr>
<td>Strong Authentication</td>
<td>✔️</td>
</tr>
<tr>
<td>Selective Disclosure</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Single Sign-On | Unlinkability through Pseudonyms (PPID)

- Unlinkability through “Pairwise Pseudonymous Identifier” (ppid in OIDC)
  - Dedicated pseudonyms per RP → unlinkable across RPs
    (+ fresh signatures)

### Table: Pseudonym Assignments

<table>
<thead>
<tr>
<th>Name</th>
<th>Movies</th>
<th>Mail</th>
<th>Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>nym_{A,1}</td>
<td>nym_{A,2}</td>
<td>nym_{A,3}</td>
</tr>
<tr>
<td>Bob</td>
<td>nym_{B,1}</td>
<td>nym_{B,2}</td>
<td>nym_{B,3}</td>
</tr>
<tr>
<td>Carol</td>
<td>nym_{C,1}</td>
<td>nym_{C,2}</td>
<td>nym_{C,3}</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Properties</th>
<th>SSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>✔</td>
</tr>
<tr>
<td>Strong Authentication</td>
<td>✔</td>
</tr>
<tr>
<td>Selective Disclosure</td>
<td>✔</td>
</tr>
<tr>
<td>Unlinkability (RP)</td>
<td>✔</td>
</tr>
</tbody>
</table>

**Diagram:**

1. Access request
2. Redirect to IdP
3. Request Auth to RP & User Auth to IdP
4. ID Token \(nym_{A,2}, \text{GER}, >18\)
5. ID Token \(nym_{A,2}, \text{GER}, >18\)

**User**

**Relying Party (RP)**

**Unobservability (IdP)**

\(nym_{A,2} = nym_{A,3}\)?
IdP needs to know the RP the user wants to authenticate to:
binds token to specific RP $rid \rightarrow$ phishing prevention
$rid$ needed for RP-specific pseudonyms

User

1) Access request to RP $rid$

2) Redirect to IdP

3) Request Auth to RP $rid$

4) ID Token $nym_{A,2}$, $nym_{B,2}$, $nym_{C,2}$, $nym_{A,3}$, $nym_{B,3}$, $nym_{C,3}$

No Privacy towards IdP -- can track users online behaviour

Properties

SSO

Usability

Strong Authentication

Unlinkability (RP)

Unobservability (IdP)
**Single Sign-On | Achieving Unobservability**

- IdP needs to know the RP the user wants to authenticate to:  
  - **Blindly** binds token to specific RP $\text{rid}$  
  - **Blindly derives** RP-specific pseudonyms

1) Access request to RP $\text{rid}$  
2) Redirect to IdP

3) Request for $c = H(\text{rid}, r)$  
4) ID Token $nym_{A,z}, \text{GER, >18, c}$  
5) ID Token $nym_{A,z}, \text{GER, >18, c}, r$

$c := H(\text{rid}, r)$ for random $r$

![Properties Table]

<table>
<thead>
<tr>
<th>Properties</th>
<th>SSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>✓</td>
</tr>
<tr>
<td>Strong Authentication</td>
<td>✓</td>
</tr>
<tr>
<td>Selective Disclosure</td>
<td>✓</td>
</tr>
<tr>
<td>Unlinkability (RP)</td>
<td>✓</td>
</tr>
<tr>
<td>Unobservability (IdP)</td>
<td>✓</td>
</tr>
</tbody>
</table>

[HSB’20] Hammann, Sasse, Basin  
Privacy-Preserving OpenID Connect  
AsiaCCS’20  
Sven Hammann  
Ralf Sasse  
David Basin  
ETH Zurich  
ETH Zurich  
ETH Zurich
**Single Sign-On | Are we done?**

- No! **RP Authentication** is missing → only registered RPs must be allowed to use SSO service

---

**User**

1) Access request to RP $\text{rid}$

2) Redirect to IdP

3) Request for $c$

4) ID Token $nym_{A,2}, \text{GER}, >18, c$

5) ID Token $nym_{A,2}, \text{GER}, >18, c, r$

$c := H(\text{rid}, r)$ for random $r$

Verifies token and that $c = H(\text{rid}, r)$

---

<table>
<thead>
<tr>
<th>Properties</th>
<th>SSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>✓</td>
</tr>
<tr>
<td>Strong Authentication</td>
<td>✓</td>
</tr>
<tr>
<td>Selective Disclosure</td>
<td>✓</td>
</tr>
<tr>
<td>Unlinkability (RP)</td>
<td>✓</td>
</tr>
<tr>
<td>Unobservability (IdP)</td>
<td>✓</td>
</tr>
<tr>
<td>RP Authentication</td>
<td>✗</td>
</tr>
</tbody>
</table>
**Single Sign-On | Are we done?**

- No! **RP Authentication** is missing → only registered RPs must be allowed to use SSO service
- Easy to add – RP has membership certificate from IdP & authenticates with every request

**Is RP Authentication that important?**

<table>
<thead>
<tr>
<th>Properties</th>
<th>SSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>✓</td>
</tr>
<tr>
<td>Strong Authentication</td>
<td>✓</td>
</tr>
<tr>
<td>Selective Disclosure</td>
<td>✓</td>
</tr>
<tr>
<td>Unlinkability (RP)</td>
<td>✓</td>
</tr>
<tr>
<td>Unobservability (IdP)</td>
<td>✓</td>
</tr>
<tr>
<td>RP Authentication</td>
<td>✓</td>
</tr>
</tbody>
</table>

→ But breaks privacy!

Verifies token and that $c = H(\text{rid}, r)$

$c := H(\text{rid}, r)$ for random $r$
Single Sign-On | Reality Check → Need for RP Authentication

Front Channel (aka Implicit Flow) vs. Back Channel (aka Authorization Code Flow)

**Implicit Flow is deprecated from OAuth 2.1**
→ w/o Implicit Flow, no chance for privacy-preserving SSO

**Specification(s)**

- OIDC ?

- **Draft:**
  - OAuth 2.1
  - Version 11:
    - May ‘24

**RP Authentication by default**
- Privacy/Unobservability impossible

---

"The [IdP] issuing access tokens to the client after successfully authenticating the [RP] and obtaining authorization."

eIDAS 2.0 §8:
[...], relying parties should provide the information necessary to allow for their identification and authentication towards the European Digital Identity Wallets
European Digital Identity Wallet

- eIDAS 2.0 published in December 2023:

§7: The technical framework of the European Digital Identity Wallet (EDIW) shall:

(a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows for tracking, linking, correlating or otherwise obtain knowledge of transactions or user behaviour unless explicitly authorised by the user.

(b) enable privacy preserving techniques which ensure unlinkability, ...

§9c: EDIWs should include a functionality to generate user chosen and managed pseudonyms, to authenticate when accessing online services.

§29: The EDIW should technically enable the selective disclosure of attributes to relying parties.

Annex 11(c)
The use of the wallet [...] should not result in the processing of data beyond what is necessary for the provision of wallet services. To ensure privacy, EDIW providers should ensure unobservability by not collecting data and not having insight into the transactions of the users of the Wallet.

§8: [...], relying parties should provide the information necessary to allow for their identification and authentication towards the European Digital Identity Wallets.
Our Work: Privacy-Preserving SSO with RP Authentication

- SSO with RP Authentication and
  - Unobservability: IdP doesn’t learn \( rid \)
  - RP Binding (part of Strong Auth): Tokens are bound to \( rid \)
  - Unlinkability: IdP derives \( rid \)-specific pseudonym

Save The Implicit Flow? Enabling Privacy-Preserving RP Authentication in OpenID Connect
Maximillian Kroschewski, Anja Lehmann
PETS 2023

OPPID: Single Sign-On with Oblivious Pairwise Pseudonyms
Maximillian Kroschewski, Anja Lehmann, Cavit Özbay
work in progress, on ePrint soon

<table>
<thead>
<tr>
<th>Properties</th>
<th>Our Work(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>✔</td>
</tr>
<tr>
<td>Strong Authentication</td>
<td>✔</td>
</tr>
<tr>
<td>Selective Disclosure</td>
<td>✔</td>
</tr>
<tr>
<td>Unlinkability (RPs)</td>
<td>✔</td>
</tr>
<tr>
<td>Unobservability (IdP)</td>
<td>✔</td>
</tr>
<tr>
<td>RP Authentication</td>
<td>✔</td>
</tr>
</tbody>
</table>
Privacy-Preserving Single-Sign On with RP Authentication

Anonymous Credentials to the rescue → but for the RP!

Key pair $isk := (isk_{RP}, isk_τ)$ $ipk := (ipk_{RP}, ipk_τ)$

RP Registration:
IdP issues anonymous credential to RP on its $rid$

$cred := \text{Sign}(isk_{RP}, rid)$

signature scheme with efficient proofs
Privacy-Preserving SSO | User & RP Authentication

- Requirements – IdP must:
  - Verify that request comes from registered RP
  - Bind token to the RP \( rid \)
  - Not learn RPs \( rid \)

- Idea similar to [HSB20]: IdP signs committed \( rid \)
- But we use Pedersen commitment & NIZK proof to authenticate the hidden \( rid \)

\[ \tau := \text{Sign}(\text{isk}_\tau, (c, \text{uid}, \text{sid})) \]

\[ cred := \text{Sign}(\text{isk}_{RP}, \text{rid}) \]

Verify that \( c = \text{Com}(\text{rid}, o) \)

\[ \pi := \text{NIZK} \left\{ (\text{rid}, \text{cred}) : Vf(\text{ipk}_{RP}, \text{rid}, \text{cred}) = 1 \land c = \text{Com}(\text{rid}, o) \right\} (\text{sid}) \]
Privacy-Preserving SSO | User & RP Authentication

- Final token should be self-contained & verifiable for \((uid, rid, sid)\)

1) Access request to RP \(rid, c, o\)

2) RP Auth: \(sid, \pi\)

3) Request + RP Auth \(sid, uid, (c, \pi)\)

4) ID Token \(sid, \tau\)

5) ID Token \(sid, uid, \tau_{\text{fin}} := (\tau, c, o)\)

\(c := \text{Com}(rid, o)\) for random \(o\)

\(\tau := \text{Sign}(isk_\tau, (c, uid, sid))\)

\(\text{standard signature}\)

\(cred := \text{Sign}(isk_{RP}, rid)\)

Verify that \(\tau_{\text{fin}}\) is valid for \((uid, rid, sid)\): Parse \(\tau_{\text{fin}} = (\tau, c, o)\)

\(Vf(ipk_\tau, (c, uid, sid), \tau) = 1\) and \(c = \text{Com}(rid, o)\)
Privacy-Preserving SSO | Security & Privacy

Only registered RPs can provide valid $\pi$
(Soundness of NIZK & unforgeability of Anon Cred) $\rightarrow$ RP Authentication

Verify $\pi$ w.r.t $ipk_{RP}, sid, c$
Compute ID token as $\tau := Sign(isk_{\tau}, (c, uid, sid))$

IdP learns nothing about $rid$ due to hiding commitment & ZK property of $\pi$
$\rightarrow$ RP Hiding / Unobservability

IdP blindly binds token to $rid$ by signing commitment $c$ $\rightarrow$ RP Binding

Verify that $\tau_{fin}$ is valid for $(uid, rid, sid)$:
Parse $\tau_{fin} = (\tau, c, o)$
$Vf(ipk_{\tau}, (c, uid, sid), \tau) = 1$ and $c = Com(rid, o)$
Challenge: how can IdP compute RP-specific pseudonyms without learning \( \text{rid} \)?

Verify \( \pi \) w.r.t \( ipk_{RP}, sid, c \)

Compute ID token as

\[
\tau := \text{Sign}(isk_\tau, (c(\text{uid}, sid))
\]

\[nym = F_k(\text{uid}, \text{rid})\]

1) Access request to RP \( \text{rid}, c, o \)

2) RP Auth: \( sid, \pi \)

3) Request + RP Auth \( \text{sid}, \text{uid}, (c, \pi) \)

4) ID Token \( \text{sid}, \tau \)

5) ID Token \( \text{sid}, \text{uid}, \tau_{fin} := (\tau, c, o) \)

\[c := \text{Com}(\text{rid}, o)\]

\(\text{for random } o\)

Verify that \( \tau_{fin} \) is valid for \((\text{uid}, \text{rid}, \text{sid})\):

\[\text{Parse } \tau_{fin} = (\tau, c, o)\]

\[Vf(ipk_\tau, (c, \text{uid}, sid), \tau) = 1 \text{ and } c = \text{Com}(\text{rid}, o)\]
Privacy-Preserving SSO | Pseudonyms

- Focus just on pseudonyms for now...
  - Unique per user & RP
  - Unlinkable across RPs
  - Blindly computable

Blindly compute ID Token $\tau$ for $sid, rid$ and

$$nym = F_k(uid, rid)$$

$$= H(rid)^{PRF(k, uid)}$$

OPRF-ish: User sends $\overline{rid} := H(rid)^r$

Verify that $\tau_{fin}$ is valid for $(nym, rid, sid)$
How to ensure that pseudonym is computed for correct \( rid \)?

Blindly compute ID Token \( \tau \) for \( sid, rid \) and

\[
nym = F_k(uid, rid)
\]

1) RP binds it’s NIZK \( \pi \) to verified \( \overline{rid} \)
2) IdP signs blinded \( \overline{rid} \) and \( \overline{nym} \) (together with \( c \)) in ID Token \( \tau \)
3) Final verification checks that commitment \( c \) and \( \overline{rid} \) are for the same \( rid \)
\( nym \) is correctly derived from the signed \( \overline{nym} \)

Verify that \( \tau_{fin} \) is valid for \( (nym, rid, sid) \)
# Privacy-Preserving Single Sign-On | Summary

<table>
<thead>
<tr>
<th>Properties</th>
<th>SSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>✔️</td>
</tr>
<tr>
<td>Strong Authentication</td>
<td>✔️</td>
</tr>
<tr>
<td>Selective Disclosure</td>
<td>✔️</td>
</tr>
<tr>
<td>Unlinkability (RPs)</td>
<td>✔️</td>
</tr>
<tr>
<td>Unobservability (IdP)</td>
<td>✔️</td>
</tr>
<tr>
<td>RP Authentication</td>
<td>✔️</td>
</tr>
<tr>
<td>Untraceability (RP &amp; IdP)</td>
<td></td>
</tr>
</tbody>
</table>

- Efficient protocol from simple building blocks
  - Standard signatures (→ RSA)
  - Signatures with efficient proofs (→ PS)
  - Commitments (→ Pedersen)
  - Pseudonyms: DDH Group, (HMAC)-SHA-256
  - Running time of 2-20ms per party

- Limitation: No Privacy against colluding IdP & RP!
  - Deterministic pseudonyms, linkage via timing information (& $sid$)
  - Inherent in solutions with single IdP and no keys/creds on user side
### Privacy-Preserving Single Sign-On | Comparison

<table>
<thead>
<tr>
<th>Properties</th>
<th>SSO</th>
<th>Anon Cred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Strong Authentication</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Selective Disclosure</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Unlinkability (RPs)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Unobservability (IdP)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RP Authentication</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Untraceability (RP &amp; IdP)</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>

- **Limitation:** No Privacy against colluding IdP & RP!
  - Deterministic pseudonyms, linkage via timing information (& *sid*)
  - Inherent in solutions with single IdP and no keys/creds on user side

![Privacy-Preserving Single Sign-On Diagram](image)

Are we „privacy-washing“ an inherently bad solution?
- No – open for discussion ;)
- We need usable solutions ➔ convenience is key
- Ideally both approaches co-exist: user’s choice