**Prior Reviews**

This manuscript was submitted to USENIX Security 2022 and received the “R2 Reject & Resubmit” review recommendation. Individual review recommendations are: 2 (Reject and resubmit), 4 (Minor revision), 2 (Reject and resubmit), 4(Minor revision) and 4 (Minor revision). Detailed review comments are attached in the end of this document.

In this manuscript, we have addressed all the issues pointed out by the reviewers. Here, we summarize the modifications that we have made in this submission.

1. We were suggested to “acknowledge the parallels between blind BLS signatures as used in for example PrivacyPass” and position our work wrt existing work.

In this submission, we extensively re-wrote Section 2 to introduce SSO login flows using OIDC implicit flow as an example (Section 2.1) and review prior approaches related to UPRESSO (Sections 2.2 and 2.3). We added a new subsection 2.2 to explain existing privacy-preserving SSO solutions (e.g., PPID, BrowserID, SPRESSO, PRIMA) and privacy-preserving identity federation protocols (e.g., PseudoID, EL PASSO, UnlimitID, Opaak, and Fabric Idemix).

As suggested by the reviewers, we compared UPRESSO with the existing approaches to explain the constraints and/or the additional requirements of prior work and the unique security protection supported by UPRESSO. The comparison was also summarized in the newly added Table 1.

1. We added a discussion about “privacy-preserving tokens/credentials” in Section 2.3 to review the approaches relying on the generation of anonymous tokens or credentials, including the PrivacyPass protocol and others using blind BLS signatures. These BLS-signature-based anonymous protocols hide or anonymize user identities from both the IdP (e.g., the signing server in PrivacyPass) and the RPs (e.g., the redemption servers). Therefore, they cannot support an important service that allows the RP to recognize the same user across multiple logins. As a result, PrivacyPass and others can only be used in use cases in which the user requests resource access, for example, in content delivery networks. However, they cannot support use cases with interactions, for example, solving a CAPTCHA challenge. However, most online systems require SSO to support such interaction between the user and the RP. Therefore, we consider this requirement an essential security property of SSO service and argue that a privacy-preserving SSO solution should provide it. More discussion is included in Section 3.2.
2. We reviewed privacy-preserving identity federation solutions including the EL PASSO protocol in Section 2.2 and compared UPPRESSO with them. Both privacy-preserving identity federation and UPPRESSO can protect user privacy against honest-but-curious IdPs and RPs, as well as IdP-RP collusion. However, privacy-preserving identity federation relies on long-term secrets and requires the user to maintain and manage the secrets, causing additional operations between the user and the RP and non-negligible secret recovery and renewal cost.
3. We thank the previous reviewers for their valuable suggestions. We made several changes in the design of the UPRESSO protocol to improve its security and efficiency.
   1. One reviewer pointed out that the “referrer” header would leak the RP’s identity to the IdP. In this submission, we re-design the user agent implementation in UPRESSO to prevent the use of the referrer header by setting the header *referrer-policy=no-referrer* in the HTTP response from the RP. We have tested this method in several browsers and confirmed there was no privacy leakage. This is discussed in the end of Section 4.4.
   2. One reviewer pointed out that the uniqueness requirement of PIDRPs restricted by IdP may not be necessary. This requirement requires the IdP to check if a newly-generated PIDRP is unique (i.e., has not been used in the system). We carefully checked this requirement and confirmed that it is not necessary. In this submission, we removed this requirement and added a proof in Section 5.1. We showed that the adversary cannot cause two different RPs to select the same PIDRP. So, the uniqueness checking is not necessary. This change simplifies the processing at the IdP and improves the protocol efficiency.
   3. One reviewer pointed out that H(t) sent to the IdP may break the privacy properties of PIDRP, where H(t) is the hash of the random number chosen by the IdP. We used H(t) originally to prevent PIDRP collisions. H() is a collision-free, one-way hash function such as SHA-256, which does not provide additional advantage for the IdP to break ECDLP. However, since we proved that the adversary cannot cause different RPs to use the same PIDRP, we do not need H(t) to check PIDRP collisions. Therefore, in this submission, we remove H(t) from the UPRESSO protocol. This reduces the information exchanged between the IdP, RP, and the user and improves the efficiency.
4. In this submission, we also clarify the concerns raised by previous reviewers about the privacy properties, the protocol scalability, and formal security proof.
   1. In Section 5.2, we formally defined the privacy properties as to distinguish the of login instances in UPPRESSO. We also added a proof of the indistinguishability of the system.
   2. Following the suggestion, we rewrote the formal proof of the security and privacy of UPRESSO in Sections 5.1 and 5.2, respectively. For security, we showed that UPPRESSO satisfies all the security requirements for the identity tokens in SSO services. For privacy, we showed that UPPRESSO effectively prevents both IdP-based login tracing and RP-based identity linkage privacy threats.
   3. We added a discussion about the scalability of the protocol in Section 7. We analyzed the probability that at least two PIDRPs are identical and formulated it as launching the birthday attack against EC-discrete-logarithm-based protocols. Our result shows that this probability is negligible.
   4. One reviewer suggested using H(RP-identifier) as IDRP instead of using a point constructed by the IdP. We explored this idea and added a discussion under “Alternative Way to Bind IDRP and EnptRP” in Section 7. First, it is possible to H(RP-identifier) as IDRP, for example, we can hash the RP’s domain to a point on an elliptic curve. The benefit is to avoid the use of RP certificate. Meanwhile, RP certificates also prevent unauthorized RPs from accessing the IdP’s SSO services. From this consideration, we still use RP certificates in our prototype implementation to verify if the target RP has already registered itself at the IdP. The advantages and disadvantages of this design option are discussed in Section 7.
5. We extended our evaluations to cover two typical scenarios in which privacy-preserving SSO protocols are used: (1) a browser (i.e., user agent) runs on the device that is connected to the servers with vSwitch (minimizing influence of network delays) and (2) a browser running locally visits the remote servers. We provided the detailed evaluation of the UPPRESSO protocol in Section 6.2.

We also open-sourced our prototype on GitHub at https://github.com/uppresso.

# Original Reviews

Review #344A

===========================================================================

Review recommendation

---------------------

2. Reject and resubmit

Reviewer expertise

------------------

3. Knowledgeable

Overall merit

-------------

1. Bottom 50% of submitted papers

Writing quality

---------------

3. Adequate

Paper summary

-------------

The authors propose UPPRESSO, a single-sign-on system that aims to allow users to engage in SSO services in a manner similar to how they do today, but such that the identity verifier doesn't learn what sites the user is authenticating on, and the sites the user is authenticating on does not learn the identify the user is asserting with the identify provider.

Strengths

---------

- Promised provided implementation

- Problem area selection

Weaknesses

----------

- Novelty

- Relationship with related work

Comments for author

-------------------

I appreciate the problem users have selected, and I particularly appreciate the promise to share the implementation, which is too rare in conference submissions. However, as is I do not think this paper would be a good fit for USENIX.

As best I can tell, the work is extremely similar to, through cryptographically wear then, the Privacy Pass protocol [1], versions of which are being discussed in standards bodies like IETF [2] and W3C [3], along with being promoted to developers [4]. The TrustToken / PrivacyPass protocol is also implemented in Chromium [5].

The main differences are that the UPPRESSO protocol uses an OPRF, while the existing privacy pass protocol uses a VOPRF (which seems strictly superior for this category of use case), and the UPPRESSO protocol uses the response to establish a persistent account, instead of as a point-in-time-assertion (which, protocol wise is a trivial difference). Given the similarity between UPPRESSO and existing work, its not clear to me whether the current work is a fundamental contribution over existing work, or an application of existing techniques to a slightly different problem space. Both are useful and important, but I don't think the latter would be a USENIX-tier contribution.

Further, the authors fail to cite or discuss this existing, very similar work, which is a weakness in and of itself, but further makes it very difficult to asses the current work's relationship to the prior work.

1. Davidson, Alex, et al. "Privacy Pass: Bypassing Internet Challenges Anonymously." Proc. Priv. Enhancing Technol. 2018.3 (2018): 164-180.

2. https://datatracker.ietf.org/meeting/108/materials/slides-108-pearg-trust-token-presentation

3. https://github.com/WICG/trust-token-api

4. https://web.dev/trust-tokens/

5. https://www.chromestatus.com/feature/5078049450098688

Requested Changes

-----------------

- Cite, discuss and clarify the papers contributions beyond existing work in research and industry

Questions for authors' response

-------------------------------

- What are the contributions the UPPRESSO protocol makes beyond the existing TrustToken / PrivacyPass protocols?

- Is there something unique about the SSO problem that makes the TT / PP approaches unsuited?

\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

Review #344B

===========================================================================

Review recommendation

---------------------

4. Minor revision

Reviewer expertise

------------------

2. Some familiarity

Overall merit

-------------

3. Top 25% but not top 10% of submitted papers

Writing quality

---------------

4. Well-written

Paper summary

-------------

The paper presents UPPRESSO, privacy-preserving single sign-on framework that can provide privacy in the face of a curious identity provider and offer security against collusive relying parties. UPPRESSO uses three transformations to hide users' and servers' ID. Consequently, identity provider does not learn about the services and services cannot derive the real users' identity even if they collude.

Strengths

---------

- Clear motivation and real world application

- Extensive study of related work

Weaknesses

----------

- Uniqueness of IDs needs to be explained further

- Breakdown of the timing evaluation

Comments for author

-------------------

I enjoyed reading this well-motivated paper. The problem the paper tries to address has real-world application. The technique seems to provide the desired security goals.

The timing does not show how much time each part of the protocol would take and refer to an overall time. The evaluation could be extended to include more details.

The paper refers to the uniqueness of IDs, but would this be scalable? What is the assumption on the number of registered accounts? This needs to be clarified in the analysis.

\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

Review #344C

===========================================================================

Review recommendation

---------------------

2. Reject and resubmit

Reviewer expertise

------------------

4. Expert

Overall merit

-------------

1. Bottom 50% of submitted papers

Writing quality

---------------

3. Adequate

Paper summary

-------------

The paper proposes UPPRESSO, a new single sign-on (SSO) scheme that has stronger privacy protections than existing SSO approaches. In SSO schemes users only have one authentication mechanism / account with a trusted Identity Provider (e.g., Facebook, Google, Apple) and then leverage this mechanism to log-in to replying parties (RPs). Traditional SSO schemes (e.g., SAML, OpenID, etc.) have two weaknesses that this paper aims to address. One, the IdP learns which RPs the user visits. Two, users' identities are the same accross RPs, even though pseudonyms would suffice.

UPRESSO addresses these two challenges. To address the first, the user and RP create blinded identity for the RP before sending it to the IdP for authentication. The IdP then uses this blinded RP identity to derive a blinded, RP-specific user identity, that the RP subsequently unblinds.

UPRESSO then provides protection against honest-but-curious IdPs that collude with RPs. Removing the HbC assumption, or the non-collusion assumption breaks the privacy properties.

Strengths

---------

+ The scheme is comparatively simple with respect to other proposals that rely on attribute-based credentials

+ Implementation of proposed scheme with performance measures

Weaknesses

----------

- Privacy properties only hold against non-colluding and HbC IdPs. Other work in this area provides better protection and is easily adapted to suit the needs of UPPRESSO

- The proposed scheme will still leak the RPs identity to the IdP as a result of the referral header when leading the IdP script.

- Privacy properties are not formally defined (e.g., in the form of a game), undoing the foundations of the privacy proof.

- The proposed scheme essentially uses blind BLS signatures and the fact that they are deterministic. Such a construction is, for example, more explicitly used in PrivacyPass.

- The two step process (registering PID registration and Identity-token regeneration) seem unnecessary: (adversarially) creating duplicate PIDs breaks the DL assumption.

- Unnecessary weakness in ID\_{RP} generation lets malicious IdP convert RP-specific identities.

- Unclear if implementation will be made open source, it should.

Comments for author

-------------------

### Overview

I really like the simplicity of the proposed scheme (especially when taking into account some of the simplifications listed below) for creating SSO scheme with better privacy properties than naive schemes. This is great. But existing privacy-friendly SSO replacements -- the same category that UPRESSO addresses, as it does make changes to both IdPs and RPs -- do measurably better. For example, EL-PASSO (as just one instantiation of a scheme that uses attribute-based credentials) completely avoid contacting the IdP in the first place. As such, EL-PASSO and related schemes provide protection against malicious IdPs as well as IdPs that collude with relying parties.

While some of my other concerns can probably be addressed, the lack of protection against IdPs is concerning, I am therefore proposing a Reject and Resubmit decision for the paper in its current form.

### Detailed comments

\*Relation with existing schemes\* The paper compares direction with EL PASS and UnlimitID in Section 2.3 and acknowledges that they solve the two identified privacy problems. Section 2.3 then identifies two challenges that I think can be solved using very standard mechanisms:

1. The user has to "locally manage pseudonyms for different RPs". This can easily be done using domain-specific pseudonyms (which is what I think the next sentence refers to) in software using the identity of the RP directly. There is no reason for the user to be involved to do any managing of identities.

2. There is an authentication step between User and RP that doesn't exist in "standard" SSO flows and this causes problems with credential compromise as users need to notify all RPs. This is not true. In UnlimitID, the IdP can deal with credential revocation checks by itself, no need to burden RPs. In EL PASSO, RPs could rely on global revocation service (as in essentially any ABC scheme) so that revocations are automatically applied everywhere. One can even combine both approaches (and protect against malicious and colluding servers) by leveraging something like Tandem [1].

In summary, I do not see why these existing approaches cannot solve the problem (possibly with tiny modifications).

\*Leakage via referrals\* In the protocol, step 1.2 / 1.3 the RP script will trigger a redirect to the IdP script. As far as I understand, this redirect will carry the RP's identity to the IdP in the form of a referral header. Why does this not happen?

\*Formalizing properties\* The privacy properties are defined in a rather narrow

fashion, and do not take into account scenarios that are realistic given the envisioned deployment. In particular, malicious RPs should be expected to see several user identities, maybe even across different RPs, and then be given the task to recognize an existing identity at a new RP. In some sense, this will then be a variant of the traditional unlinkability game. Neither the definitions in 3.3, nor the arguments in 5.2 capture this notion properly. They should.

Moreover, the protocol does not just send the blinded value PID to the IdP, but also the hash $H(t)$ of the blinding factor itself. The former alone is perfectly hiding, the combination of the two is not, and requires a more delicate argument and an assumption on $H$ to guarantee privacy wrt the IdP.

I would expect the proof to still work somewhat similar, but being precise here is important. Also notice that indeed, as per section 5.2 the proof will at the very least rely on the DDH assumption, and not, as stated earlier in section 4.3 on the DL assumption.

for example the definitions in 3.3 are very high-level and do not take into account deployment (e.g. that malicious RPs could see many instances).

Finally, it is not clear to me whether the Dolev-Yao model in the appendix correctly models collusions between users and several RPs in impersonating users. The example given in 5.1 is just that: an example of the base case.

\*Viewing as a BLS signature\* A BLS signature on a message $m$ with private key $x$ is given by $[x] H(m)$ where H is a hash function mapping messages to group elements. Such a scheme can easily be turned into a blind signature scheme by sending a randomized message $[r] H(m)$ to sign to the signer. This is essentially the core of the UPRESSO transformation. See for example the PrivacyPass [2] paper where this is used to great effect.

In fact, using H(RP-identifier) as base point instead of a point \_constructed\_ by the IdP will also mitigate the attack where the IdP is able to translate identities from one RP to another. This weakness is I think unnecessary, the "private key" corresponding to $ID\_{RP}$ values is in fact never used, so might as well replace it with a hash so that the IdP cannot cheat. This might also reduce the reliance on the RP certificate.

\*Uniqueness requirement of PID\_RPs\* I don't think this requirement is actually necessary, thus leading to a simplification of the scheme. Here is my analysis, if a user can find t\_1, t\_2, such that ID\_1 and ID\_2 are mapped to the same $PID = [t\_1] ID\_1 = [t\_2] ID\_2$ then $ID\_1 = [t\_1^{-1} t\_2] ID\_2$ and thus the user can find the discrete logarithm of ID\_1 wrt. ID\_2.

\*Evaluations\* It seems from the text that the measurements include network latency. How big was this latency? If these measurements were done in a LAN, how does higher WAN latency affect the measurements shown? It seems that UPRESSO has more roundtrips than other systems.

### Writing comments

\* Incorrect use of definitive article, for example "maintain the credential at the IdP" (this credential has not been introduced, should be "a"), "defined against either the curious IdP" (same, "a"), "the violations of identity" (fix by removing article), "the IdP-based login tracing" (fix by removing article)

\* Abstract / Intro: personally I think these would be more readable with less terminology and math.

\* p2 "All existing SSO protocols" -> this is false, see also the related work in this paper, EL-PASSO nor UnlimitID have this weakness

\* p3 "RP Dynamic Registration" this section did not make sense to me. If there is also manual or static registration, how does that work? As far as I can see they have not yet been described.

\* "Vulnerable SSO Implementations". This is a great overview that shows that implementing SSO correctly is difficult, but I am not quite sure what it brings to this paper

\* Figure 3: I think it would help to more clearly mark the two (or three!) scripts/webpages/windows that run in the user's browser in the activity diagram.

\* Step 4.3 "After obtaining the user's authorization to" -> does this also reveal the RP to the user? It seems to me that it is important that the user still checks which RPs they will be disclosing information to.

\* PEnpt\_U: It is unclear to me why this temporary endpoint is necessary. It seems to be transmitted, but then not subsequently used anywhere

### Nits, bits and pieces

\* p1 "collusive RPs" -> more common, and probably a bit more precise, "colluding RPs" (repeated)

\* p1 "across the RPs, to learn his" -> no comma (and, genders?)

\* p2 "they require modifications to .... that essential conflicts" -> conflic

\* p3 "a "pure" SSO protocol does not include any authentication step" -> but there is :), between user and IdP. Maybe rewrite

\* p3 "from different these tokens" -> error somewhere?

\* p4 "SSO system shall offer" -> should?

\* p4 "to satisfy the requirements ..., poses" -> doesn't work

\* p5 "appears a random variable" -> random variable has another meaning usually. Maybe "is indistinguishable from random" or something of the sort

\* p6 "curious-but-honest" -> more common is "honest-but-curious" I think

\* p6 "learning user privacy" -> no such think, maybe "breaking privacy" or "violating privacy"

\* p7 "RP certificates are designed to" -> by whom? maybe "we designed RP signatures"

[1] Wouter Lueks, Brinda Hampiholi, Greg Alpár, Carmela Troncoso:

Tandem: Securing Keys by Using a Central Server While Preserving Privacy. Proc. Priv. Enhancing Technol. 2020(3): 327-355 (2020)

[2] Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, Filippo Valsorda:

Privacy Pass: Bypassing Internet Challenges Anonymously. Proc. Priv. Enhancing Technol. 2018(3): 164-180 (2018)

Requested Changes

-----------------

The following modifications will help make the paper better:

\* Either provide better protection against colluding and malicious IdPs or very carefully argue why this paper provides an interesting design point. Once you start modifying existing SP and IdP code, one might as well go the full hog and use one of the ABC related schemes.

\* Fix / address Referral leakage via redirect

\* Rephrase proposed scheme in terms of (blind) BLS signatures. This will probably also help with eliciting the security properties

\* Proper modeling of privacy properties and clear proofs / statements of security.

Questions for authors' response

-------------------------------

\* Is referral leakage in the redirect correct? Counter measures possible?

\* Are there other advantages wrt EL PASSO and friends that we should consider?

\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

Review #344D

===========================================================================

Review recommendation

---------------------

4. Minor revision

Reviewer expertise

------------------

1. No familiarity

Overall merit

-------------

3. Top 25% but not top 10% of submitted papers

Writing quality

---------------

4. Well-written

Paper summary

-------------

This paper presents a privacy-preserving SSO system that protects users against colluding relying parties and curious identity providers by using ephemeral user pseudo-identities.

Strengths

---------

-The analysis of prior and related work is strong and thorough

-The protocol diagram in figure 3 is very helpful

-The analysis of the security and privacy properties are thorough and helpful

-I really like that you included the formal proofs, albeit in the appendix

Weaknesses

----------

-Leaving the security proofs to the appendix left a great deal of interesting detail out of the paper

-More discussion of the impact of the time required to run UPPRESSO in section 6.2 would be helpful

Comments for author

-------------------

This is an interesting paper about the prior work on SSO systems and the requirements of those systems. This provides a blueprint for anyone looking to implement an SSO system and looking to do so in a privacy-preserving manner. At times, I found the paper difficult to follow, but this may be because I have little familiarity with this area. I think what was most lacking for me was an analysis of the real-world impact--have we seen attacks like those your system defends against? What is the real impact of the performance of UPPRESSO? How can systems like this move from academic papers to 'the real world'?

Requested Changes

-----------------

More details on the impact of your system--are RPs actually colluding? Is this system too slow to actually perform in the real world? I would also prefer to see the formal proof included in the actual paper itself instead of relegated to the appendix.

\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

Review #344E

===========================================================================

Review recommendation

---------------------

4. Minor revision

Reviewer expertise

------------------

3. Knowledgeable

Overall merit

-------------

3. Top 25% but not top 10% of submitted papers

Writing quality

---------------

2. Needs improvement

Paper summary

-------------

The authors augment the mechanics of single sign-on to eliminate the identity provider's ability to link a user's identity/login activity across multiple login sites (relying parties) or sessions. They describe and benchmark prototype code to implement UPPRESSO.

Strengths

---------

\* Adds privacy via unlinkability to single sign-on flows

\* Well motivated from prior work, addresses a long-standing issue with a commonly used web component

\* Plausible performance with strategies to improve it (browser plugin or integration)

Weaknesses

----------

\* Lots of notation can make the argument challenging to follow

\* Unclear what parts of the ecosystem would have to change to adopt this method

Comments for author

-------------------

This is a good analysis of a privacy gap in currently common web protocols, with a strong proposal to address that weakness.

Since SSO is an ecosystem feature, requiring buy-in from multiple parties to function effectively, your proposal could be even stronger if it explained what a web-scale roll-out would require each participant (user, RP, and IdP) to change in its implementation -- you do this clearly for the end-user/browser.

The security and privacy analysis could also be enhanced by relating it to deployment scenarios, such as whether the linkages can be protected against a data breach or compelled disclosure at the IDP.

Will you make the source code available?

Editorial: Please proofread for grammar and flow. The paper's readability suffers from unclear or ungrammatical language.

Requested Changes

-----------------

\* Clarify implementation requirements on each party

\* Consider other questions raised above