Prior Reviews and Revisions

This manuscript was submitted to USENIX Security 2022 and received the "R2 Reject & Resubmit" recommendation. Individual review recommendations are: 2 (Reject and resubmit), 4 (Minor revision), 2 (Reject and resubmit), 4 (Minor revision) and 4 (Minor revision). It was then submitted to IEEE S&P 2023 and received the "Reject" recommendation. Individual review recommendations are: 1 (Reject) and 1 (Reject). 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. We summarize the modifications that we have made in this submission as below.

- **Technical contribution and comparison with related work, especially on the collusive attacks by an IdP and RPs**

In Section 2.2 we analyze and compare existing privacy-preserving solutions of SSO and identity federation. These solutions are summarized in Table 1. In this manuscript, we use the term "single sign-on (SSO)" to describe the scenario where a user is authenticated to *only* an IdP and visits multiple RPs, and "identity federation" for the scenario where a user is authenticated to an IdP and also presents some credentials to RPs, which are derived from his secret (i.e., some authentication steps are actually involved between a user and an RP).

In the comparison, privacy-preserving SSO solutions are OIDC with PPID [5], BrowserID [7], and SPRESSO [6], while privacy-preserving schemes of identity federation include PRIMA [17], PseudoID [11], EL PASSO [12], UnlimitID [13], Opaak [14], Idemix [16, 18], and U-Prove [15]. We also discussed different applications of anonymous tokens or credentials, including PrivacyPass [26], TrustToken [27], ZKlaims [28], Crypto-Book [29], and Tandem [32].

Compared with existing privacy-preserving solutions of SSO [5-7], our improvement is remarkable because a user's online profile is protected against both a curious IdP and colluding RPs while the performance overhead is reasonable. Only one type of privacy threat is prevented in the existing privacy-preserving SSO schemes [5-7].

As discussed in Section 7 Discussions "Collusive Attack by an IdP and RPs", compared with privacy-preserving schemes of identity federation [11-18], UPPRESSO intentionally eliminates the actual authentication steps between a user and an RP. All privacy-preserving schemes of identity federation actually needs some authentication steps between a user and an RP (see Table 1 and Section 2.2), and then a user needs to hold a long-term secret verified by RPs and locally manage the accounts at different RPs. Some schemes require the user to manage independent accounts locally by himself [11], which is very inconvenient. The others deterministically derive the account from an RP domain and the user secret [12-15, 18], to reduce the user's burden; however, if the user secret is lost or leaked, a user has to notify all RPs to update his accounts which are derived from this secret.

In principle, to prevent user privacy against the collusive attacks by an IdP and RPs, a privacy-preserving scheme needs the extra authentication steps between a user and an RP; that is, a user has to (*a*) *manage independent accounts by himself* or (*b*) *derive the accounts from a secret held by the user only*. If the accounts are not masked by a user secret managed by the user himself, the colluding IdP and RPs can eventually link them. It is very inconvenient to manage independent accounts or update accounts when the secret is lost or leaked. Additionally, for web applications, a user needs to install a browser extension to handle the long-term user secret.

One of our design goals is to eliminate the user-managed accounts and the authentication steps between a user and an RP, to keep the convenience. This convenience is a desirable feature of the widely-used SSO protocols. Thus, the cost is that UPPRESSO does not protect user privacy against the collusive attacks by an IdP and RPs. Moreover, because in UPPRESSO a user is authenticated to only the IdP by any appropriate means (e.g., password, or one-time password), there is no such a long-term user secret so that UPPRESSO works with commercial-off-the-shelf (COTS) browsers. It is another desirable convenient feature.

Besides, in order to avoid disputes, in this submission we do not declare that UPPRESSO is the first SSO protocol preventing both the IdP-based login tracing and the RP-based identity linkage.

The cryptographic technologies used in UPPRESSO is not so significant, and similar cryptographic skills (e.g., blind signature and OPRF) have been applied in different scenarios in addition to sign-on. Our main contributions are *to explicitly and comprehensively consider the relationships of all five (pseudo-)identities in the SSO login flow*, and to find out cryptographic technologies to construct the identity transformation functions satisfying desirable requirements. This is mentioned in Section 2.3 Extended Related Works “Privacy-Preserving Token or Credential”.

In Section 2.3 Extended Related Work, related works on SSO Implementation Vulnerabilities are still kept in this submission. Although they are unrelated to the privacy-preserving designs, these vulnerabilities are closely related to the four security requirements of SSO identity tokens. The vulnerabilities result from one or more violation of these four security requirements.

- **Performance evaluation**

We finished more extensive performance evaluation in Section 6.2. The performance in two scenarios is evaluated, where all entities are deployed in a virtual private cloud and a local user browser remotely visits the servers in the cloud, respectively. The cloud scenario expresses the measurement in LAN without network delays, while the remotely-visiting scenario works similarly to the real-world deployment.

The evaluation shows that UPPRESSO outperforms SPRESSO but requires only a few more overheads than OIDC with PPID (i.e., MITREID Connect). The overall times of an SSO login instance for MITREid Connect, UPPRESSO, and SPRESSO are (*a*) 63 ms, 179 ms, and 190 ms, respectively, when all entities are deployed on Alibaba Cloud, and (*b*) 312 ms, 471 ms, and 510 ms, respectively, when the user browser runs locally to remotely visit the servers.

We divide an SSO login flow into three parts, namely identity-token requesting, identity-token generation, and identity-token acceptance, to analyze the overheads in details in Section 6.2.

- **Scalability or accommodation**

We discuss the scalability (or the accommodation of users and RPs) in Section 7 Discussions

“Scalability”. UPPRESSO accommodates *n* users and *n* RPs, where *n* is the order of *G*. For the NIST P256 elliptic curve, *n* is approximately 2^{256}; or for a stronger elliptic curve, e.g., *n* is approximately 2^{384} for the NIST P384 elliptic curve.
Adversaries cannot exhaust PID_{RP}, either. We ensure PID_{RP} is unique in unexpired tokens. When the system serves 10^8 requests per second and the validity period of tokens is 10 minutes, the PID_{RP}-collision probability is less than 2^{-183}, which is negligible, for the NIST P256 curve.

- **Design improvement**

Thanks very much for the anonymous reviewer’s suggestions. We improve the designs of UPPRESSO: some useless steps are removed as below, resulting in much better performance. Based on the improved designs, we finish the proofs of security and privacy, implement the prototype system, and evaluate the performance.
The improved designs are described in Section 4.5 with details. The steps of RP dynamic registration are removed, and $H(t)$ and $PEnpt_U$ are removed accordingly. In Section 5.1 Security “RP Designation”, we prove that PID_{RP} collision is negligible (or computationally impossible) with the improved designs, based on the elliptic curve discrete logarithm problem (ECDLP). So the steps of RP dynamic registration to check PID_{RP} uniqueness are removed.

In Section 7 Discussions “Alternative Way to Generate ID_{RP} and Bind $Enpt_{RP}$”, we discuss different ways to generate ID_{RP}. The design of RP certificates binding ID_{RP} and $Enpt_{RP}$, ensures the target RP has already registered itself at the IdP, which prevents unauthorized RPs from accessing the IdP’s services.
As suggested by the reviewer, an alternative way to generate ID_{RP} and bind $Enpt_{RP}$ is as follows: ID_{RP} is deterministically calculated based on the RP’s unambiguous name. For example, *Hs*() encodes an RP’s domain (or the RP script’s origin, e.g., https://RP.com) to a point on the elliptic curve as ID_{RP}, where hashing to elliptic curves *Hs*() [68] provides collision resistance and does not reveal the discrete logarithm of the output. This way removes RP certificates, decrease the size of postMessage messages, and replaces the signature verification by the hashing to elliptic curves. It produces approximate performance, but it needs special operations by each user to migrate his account to the updated RP system if an RP updates its domain.

In Section 4.4, the referer leakage is eliminated, by setting the referrer-policy=no-referrer header in the HTTP response from the RP, when it is redirected to the IdP to download the IdP script. This completely prevents the possible referer leakage of the target RP’s origin to the IdP, when a user browser downloads the IdP script. This method is specified by W3C [64] and widely supported. We have tested it in browsers including Chrome, Safari, Edge, Opera and Firefox, and confirmed no referer leakage.

- **Open source**

We have open sourced the prototype system at https://github.com/uppresso/.

- **Formal proofs of security and privacy**

We improved the formalized proofs of security and privacy of UPPRESSO. As for security, in Section 5.1 we prove four sufficient and necessary properties, namely RP Designation, User

Identification, Confidentiality, and Integrity, which have been analyzed [38, 39, 41]. In particular, we prove that there is no *PID_{RP}* collision, based on the elliptic curve discrete logarithm problem (ECDLP).

As for privacy, in Section 5.2, we prove that UPPRESSO prevents the IdP-based login tracing and the RP-based identity linkage. In particular, indistinguishability of *PID_{RP}* to the IdP is described more rigorously. To prove the prevention against the RP-based identity linkage, we consider the more general condition that *c* colluding RPs collect the information of login instances by *v* users, but not only two RPs with information of two login instances in the previous version. We prove it based on the elliptic curve decision Diffie-Hellman (ECDDH) assumption.

As mentioned in Section 5.1, we develop a Dolev-Yao style model to analyze which processes are involved in the lifecycle of an identity token in UPPRESSO. This model formally proves confidentiality and integrity of identity tokens.

On the other hand, in the proof of the prevention against the IdP-based login tracing by a Dolev-Yao style model of BrowserID [7], D. Fett *et al.* actually proved that the IdP cannot access or retrieve the parameters to generate identity assertions, which ensures its privacy property. We follow a similar way in the Dolev-Yao style model, to prove no adversary able to retrieve or manipulate the parameters in the identity token. However, because we utilize a very different approach to protect user privacy, this model cannot be used to prove the privacy properties of UPPRESSO: in BrowserID the protected privacy identities are “kept out of reach of the IdP” (i.e., kept confidential), while in UPPRESSO the protected privacy identities are transformed among entities. Note that we have proved the privacy properties of UPPRESSO based on the cryptographic features of an elliptic curve.

- **Threat model and remaining attack surface**

In Section 4.1, we assume an honest-but-curious IdP, and this is consistent with the widely-used SSO services [1-5]. We also assume a script downloaded from honest entities is also honest, for HTTPS is adopted to secure the communications.

As mentioned in Section 4.2, UPPRESSO is designed for users who really care about privacy, so a user never authorizes the IdP to enclose any distinctive attributes in identity tokens, such as telephone number, Email address, etc. A user does not configure distinctive attributes at any RP, either. Thus, the privacy leakage due to re-identification by distinctive attributes across RPs, is out of the scope. UPPRESSO provides an option for such users.

As mentioned in Section 4.2, the active account linkage through web documents by colluding RPs, is not considered in our work. When a user visits multiple RPs concurrently from one browser, an RP might actively redirect his account to another RP server by carefully-crafted web documents. We focus on the privacy threats introduced by SSO services, but such attacks exist in all web applications as well as the network traffic analysis that tracks a user’s activities from packets. Such active attacks can be detected based on the abnormal behaviors of web documents. They should be prevented by other defenses, and are not considered in our work.

- **SSO service in the real world and privacy attack**

The compatibility with OIDC is discussed in Section 4.6. It will be easy to update an OIDC system to support UPPRESSO, by (*a*) mapping each existing user identity at the OIDC IdP to a unique random integer u, (*b*) assigning unique random ID_{RP} to each RP, and (*c*) mapping every existing account at an RP to $[u]ID_{RP}$. Because UPPRESSO works with COTS browsers, the users will smoothly continue to access the services of UPPRESSO.

The RP-based identity linkage is discussed in NIST Special Publication 800-63C: Digital identity guidelines: Federation and assertions, and pairwise pseudonymous identifiers (PPIDs) are recommended in SSO services to prevent this privacy threat. Such identity linkages are widely discussed and reported; for example, when WeChat SSO and AliPay SSO are very popular in China, some service providers (or RPs) cooperate to build their users' profiles by linking user accounts, sometimes using the protocols of private set intersection (PSI). The service providers (or RPs) then push advertisements based on a user's visit history.

In widely-used PPID-enhanced SSO services, the IdP-based login tracing is always possible and several IdP operators push advertisements based on a user's visit history. If there is a data breach at the IdP, the RP-based identity linkage also becomes possible: the mapping of PPIDs to a user identity is disclosed to RPs, and then they are able to link the accounts across RPs.
However, since UPPRESSO does not prevent the collusive attacks by an IdP and RPs, compelled data disclosure at the IdP and RPs will learn a user's login activities and online profile.

- **Presentation, writing, grammar errors and typos**

In Figure 3, two vertical lines split the user operations into two groups (i.e., in two browser windows), one of which is to communicate with the IdP, and the other is with the target RP.

In Section 4.5, the IdP script *locally* obtains the user's authorization to enclose the requested attributes in Step 3.3 of the UPPRESSO protocol. We clarify that this authorization is finished locally, so the RP identity is not disclosed to the IdP.

We have corrected the grammar errors and typos. We will invite a native speaker to help us to improve the writing.

Reference, mentioned in this document

[1] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, OpenID Connect core 1.0 incorporating errata set 1, The OpenID Foundation, 2014.

[2] D. Hardt, RFC 6749: The OAuth 2.0 authorization framework, Internet Engineering Task Force, 2012.

[3] J. Hughes, S. Cantor, J. Hodges, F. Hirsch, P. Mishra, R. Philpott, and E. Maler, Profiles for the OASIS security assertion markup language (SAML) V2.0, OASIS, 2005.

[4] T. Hardjono and S. Cantor, SAML V2.0 subject identifier attributes profile version 1.0, OASIS, 2018.

[5] P. Grassi, E. Nadeau, J. Richer, S. Squire, J. Fenton, N. Lefkovitz, J. Danker, Y.-Y. Choong, K. Greene, and M. Theofanos, SP 800-63C: Digital identity guidelines: Federation and

assertions, National Institute of Standards and Technology (NIST), 2017.

[6] D. Fett, R. Küsters, and G. Schmitz, “SPRESSO: A secure, privacy-respecting single sign-on system for the Web,” in 22nd ACM Conference on Computer and Communications Security (CCS), 2015, pp. 1358–1369.

[7] D. Fett, R. Küsters, and G. Schmitz, “Analyzing the BrowserID SSO system with primary identity providers using an expressive model of the Web,” in 20th European Symposium on Research in Computer Security (ESORICS), 2015, pp. 43–65.

[11] A. Dey and S. Weis, “PseudoID: Enhancing privacy for federated login,” in 3rd Hot Topics in Privacy Enhancing Technologies (HotPETs), 2010.

[12] Z. Zhang, M. Król, A. Sonnino, L. Zhang, and E. Rivière, “EL PASSO: Efficient and lightweight privacy-preserving single sign on,” Privacy Enhancing Technologies, vol. 2021, no. 2, pp. 70–87, 2021.

[13] M. Isaakidis, H. Halpin, and G. Danezis, “UnlimitID: Privacy-preserving federated identity management using algebraic MACs,” in 15th ACM Workshop on Privacy in the Electronic Society (WPES), 2016, pp. 139–142.

[14] G. Maganis, E. Shi, H. Chen, and D. Song, “Opaak: Using mobile phones to limit anonymous identities online,” in 10th International Conference on Mobile Systems, Applications, and Services (MobiSys), 2012.

[15] C.Paquin, U-Prove technology overviewv1.1, Microsoft Corporation, 2013.

[16] Hyperledger Fabric, “MSP implementation with Identity Mixer,” https://hyperledger-fabric.readthedocs.io/en/release-2.2/idemix.html, Accessed July 20, 2022.

[17] M. R. Asghar, M. Backes, and M. Simeonovski, “PRIMA: Privacy-preserving identity and access management at Internet-scale,” in 52nd IEEE International Conference on Communications (ICC), 2018.

[18] J. Camenisch and E. V. Herreweghen, “Design and implementation of the Idemix anonymous credential system,” in 9th ACM Conference on Computer and Communications Security (CCS), 2002.

[26] A. Davidson, I. Goldberg, N. Sullivan, G. Tankersley, and F. Valsorda, “Privacy Pass: Bypassing Internet challenges anonymously,” Privacy Enhancing Technologies, vol. 2018, no. 3, pp. 164–180, 2018.

[27] Web Incubator CG, “TrustToken API,” https://github.com/WICG/trust-token-api, Accessed July 20, 2022.

[28] M. Schanzenbach, T. Kilian, J. Schutte, and C. Banse, “ZKlaims: Privacy-preserving attribute-based credentials using non-interactive zero-knowledge techniques,” in 16th International Joint Conference on e-Business and Telecommunications (ICETE),Volume 2: SECRYPT, 2019.

[29] J. Maheswaran, D. I. Wolinsky, and Bryan Ford, “Crypto-book: An architecture for privacy preserving online identities,” in 12th ACM Workshop on Hot Topics in Networks (HotNets), 2013.

[32] W. Lueks, B. Hampiholi, G. Alpar, and C. Troncoso, “Tandem: Securing keys by using a central server while preserving privacy,” Privacy Enhancing Technologies, vol. 2020, no. 3, pp. 327–355, 2020.

[38] D. Fett, R. Küsters, and G. Schmitz, “A comprehensive formal security analysis of OAuth 2.0,” in 23rd ACM Conference on Computer and Communications Security (CCS), 2016, pp.

1204–1215.

[39] D. Fett, R. Küsters, and G. Schmitz, “The Web SSO standard OpenID Connect: In-depth formal security analysis and security guidelines,” in 30th IEEE Computer Security Foundations Symposium (CSF), 2017, pp. 189–202.

[41] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, and L. Tobarra, “Formal analysis of SAML 2.0 web browser single sign-on: Breaking the SAML-based single sign-on for Google Apps,” in 6th ACM Workshop on Formal Methods in Security Engineering (FMSE), 2008.

[64] J. Eisinger and E. Stark, W3C candidate recommendation: Referrer policy, World Wide Web Consortium (W3C), 2017.

[68] A. Faz-Hernandez, S. Scott, N. Sullivan, R. Wahby, and C. Wood, draft-irtf-cfrg-hash-to-curve-16: Hashing to elliptic curves, Internet Engineering Task Force, 2022.

Original Reviews

USENIX Security '22 Fall Paper #344 Reviews and Comments

===

Paper #344 UPPRESSO: Untraceable and Unlinkable Privacy-PREserving Single Sign-On Services

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?

Reviewer feedback on authors' response and online discussion

Thanks for taking the time to respond to the reviews. Please carefully reconsider how BLS signatures and PrivacyPass relate to your proposed scheme. They are really very similar in the _techniques they use_ even though they are used to achieve different effects. I really think that acknowledging these similarities or shared ideas will make the paper clearer and easier to read, while at the same time making it easier to highlight your contribution.

About the proofs. The heavily compressed of unlinkability with several background instances is a little bit too compact to easily expand in the rebuttal. I appreciate the level of detail, but maybe the rebuttal didn't quite have enough space to make that work. I do trust that you'll be able to extend the security model and proofs to reflect the more realistic situation.

I am more skeptical about the value `H(t)` that needs to be include. Yes, conceptually, the proof should work, because `t` is high entropy, but I'd expect some assumption on `H` will be required to make the actual proof work. For example, a random oracle assumption. This should be reflected in the paper.

Finally, to improve the paper, I think it would help a lot to step back a little bit and state clearly the properties that you'd like to achieve. Focussing on properties will let you clearly specify that you'd want easy recovery of secrets (maybe via the IdP). And maybe from there it can be argued that the existing approaches really cannot work? Or that Recovery of Secrets (or maybe no-on-device secrets) means the IdP must be honest but curious?

FWIW, I am less interested in the specific SSO flow or maintaining that. I think this is a false requirement. The flow is replacable, once you accept that you can change code client, IdP, and RP.

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

Comment @A1 by Reviewer C

Thanks for submitting this paper to USENIX Security. The paper caused quite a bit of discussion among the reviewers. Reviewers liked the practical aspects of the work, aiming to provide better privacy wrt an IdP. At the same time, reviewers were concerned about a lack of novelty and positioning wrt existing work as well as a reduction in security guarantees compared to existing work. Ultimately, we decided on an R&R decision for this paper.

To improve the paper, we recommend two big changes. One, acknowledge the parallels between blind BLS signatures as used in for example PrivacyPass. Even though the final schemes are different, they share a lot of ideas. Two, ideally, UPRESSO should also provide privacy when the IdP and RP collude. In case that is not possible, the paper should be more explicit about these security and privacy properties in existing work (e.g., UnlimitID / ESPRESSO) to show both where UPRESSO makes advantages, where it does not, and explain why that is a reasonable tradeoff.

===

IEEE Symposium on Security & Privacy 2023 Paper #254 Reviews and Comments

#254 UPPRESSO: Untraceable and Unlinkable Privacy-PREserving Single Sign-On Services

Review #254A

===

Overall merit

----

1. Reject

Reviewer expertise

----

4. Expert

Reviewer confidence

----

3. High

Paper summary

This paper proposes a new (Web) Single Sign-On scheme (SSO) called UPPRESSO. This SSO aims to prevent identity providers (IdPs, parties that attest users' identities) to track where users login and further provides relying parties (RPs, parties where users log in) only with user identifiers that cannot be linked across RPs. The core of the scheme is based on an identity transformation scheme based on elliptic curves. This transformation scheme is used to create ephemeral pseudo-ids for the RP and for the user during log in and allows the RP to use a trap-door function to calculate a permanently unique RP-specific user id. The identity transformation scheme is embedded in a web-based protocol. As a proof-of-concept, the authors adapt the widely-used Web SSO OpenID Connect by replacing its core mechanism with their protocol.

Strengths
----
* This paper aims to create a simple-to-use (Web) Single Sign-On protocol that prevents IdP-based login tracking as well as RP-based identity linkage at the same time.
* The paper discusses the security of the proposed identitiy transformation scheme and mentiones that the security of the web protocol has been anlysed with formal methods.

Weaknesses
----
* UPPRESSO relies on central parts of the IdP logic to be fully trusted in order to protect against IdP-based privacy attacks.

Comments for author

My main concern regarding this paper is the very strong assumption that the IdP script, which runs in the user's browser needs to be fully trusted. The authors state that they assume an "curious-but-honest IdP [.. that ..] strictly follows the protocol [..] it might store all received messages [..]". In a web setting, however, the control realm of protocol participants extends into the user's web browser. All scripts running inside a browser window are under full control of that window's origin. In UPPRESSO, the IdP script (which runs in a window under the IdPs origin) performs key steps of the protocol which require and necessarily imply full knowledge of the RP's identity. Hence, the assumption of the authors is that this part of the IdP logic is fully honest and not just curious-but-honest (including all other scripts running under the same origin). An IdP can easily replace that script with a script that strictly follows the protocol, but in addition reveals the RP's identity to the IdP server.

Another concern is about the goal of unlinkability of user ids across RPs. The authors assume that their approach protects this privacy property also against colluding malicious RPs. Colluding RPs however can easily link user identities via the user's browser. Hence, this privacy property cannot hold under such a strong attaker model. (This property rather aims to protect against user database leakage at RPs, i.e., an attack in which RPs do not act maliciously towards the user.)

The authors state that they have analysed the security of UPPRESSO using the formal methods introduced by [8]. This paper, however, also introduces a method to analyse privacy (IdP-based login tracking) of an SSO system. Did the authors also analyse the privacy of UPPRESSO using this formal method?

Review #254B
===

Overall merit

----

1. Reject

Reviewer expertise

----

2. Some familiarity

Reviewer confidence

----

2. Medium

Paper summary

This paper is about privacy in single sign-on (SSO) protocols. SSO protocols like OIDC and OAuth are ubiquitous on the modern internet. They work by having the conventional user-website login flow include an additional identity provider. When the user wants to login to a website, the user asks its identity provider for a (signed) assertion about its identity and some of its attributes. It can give this assertion to the website in lieu of running (say) a password-based authentication protocol. A key problem with this architecture is privacy: in particular, the identity provider learns information about the websites the user is visiting, and the website may learn unnecessary information about the user from the identity provider.

This paper seeks to build new SSO protocols that protect the user's browsing information from both the idp and (possibly colluding) websites. The paper presents several identity transformation functions. The functions use multiplicative blinding of elliptic curve points to derive ephemeral identifiers from long-lived ones, obscuring the linkage between logins.

Strengths

+ Good explanation of SSO basics
+ Polished, very clearly written

Weaknesses

- RP-based identity linkage does not seem like a real problem
- Minimal technical contribution

Comments for author

I thank the authors for their submission. Privacy in SSO is an important question, and this paper does a good job of explaining the challenges and subtleties in SSO privacy. However, I do not think this paper should appear at the conference, for two main reasons.

The first is that I don't understand why the RP-based identity linkage problem is important to solve.

Given that solving this specific problem is a key focus of the paper, this is a crucial question that isn't really answered. If I understand it, the intent is to prevent colluding RPs from reconstructing user browsing history from SSO interactions. Is this meaningful, given that other data (e.g., login name or the account recovery email address) will likely be correlated across RPs? If the user's accounts are easy to link across RPs (mine certainly are - I use just 2-3 email addresses for dozens of accounts) why is it important to eliminate this kind of linkage attack?

The second reason I do not support publication is that this paper's technical contribution is minimal. The identity transformation functions are all based on simple multiplicative blinding/unblinding of elliptic curve points. I get the sense that a blind signature and/or OPRF could solve these privacy problems very generically - if they can't, the authors should explain why. A related concern is that the security analysis is very oversimplified: only a few very simple security properties of the scheme are proven; it's not clear why these properties are the most important ones for security. The authors should explain why the results they prove in section 5 give a relatively complete argument for the security and privacy of UPPRESSO.