Reviews from the Submission to USENIX Security 2021 and Our Improvements

This manuscript was submitted to USENIX Security 2021 under the title "UPPRESSO: An Unlinkable Privacy-PREserving Single Sign-On System", and the decision was Reject and Resubmit.

The following are the weaknesses extracted from the reviews. We have improved the designs and implementations greatly, to solve these weaknesses.

Review 1

1.1 Related work

A number of proposals for such privacy-friendly SSO exist already, but most target only one of the two goals. One recent exception is the ElPasso protocol [1], which is not mentioned in the paper. The idea of using $ID_{RP}^{ID_U}$ to create domain-specific pseudonyms has been widely used in several pseudonym schemes (anonymous credentials, pseudonym systems, group signatures, DAA ..) but also in the context of SSO (e.g. [2,1]).

[1] EL PASSO: Efficient and lightweight privacy-preserving single sign on. 2021.

[2] UnlimitID: Privacy-preserving federated identity management using algebraic MACs. 2016.

[3] PseudoID: Enhancing privacy for federated login. 2010.

Our improvement While there are a number of proposals claiming to provide privacy-preserving single sign-on services, they follow a login flow different from the commonly-used SSO systems such as OIDC and OAuth 2.0. In EL PASSO, UnlimitID or PseudoID, a user is authenticated by the RP with his credentials to prove that he owns the long-term secret.

a user needs to notify each RP one by one when a credential is lost or compromised, because the user is authenticated by the RP with his credentials to prove that he owns the secret. On the contrary, in the commonly-used SSO systems [**?**, **?**] and privacy-preserving SSO solutions such as BrowserID [**?**], SPRESSO [**?**] and also UPPRESSO, because (a) authentication happens between a user and the IdP and (b) an RP verifies only tokens generated by the IdP, the user only needs to renew his credential at the IdP if it is compromised. Although EL PASSO calls itself an SSO scheme [**?**] and the service signing tokens or credentials in EL PASSO, UnlimitID and PseudoID is also called the IdP [**?**, **?**, **?**], the authentication steps between the user and an RP do not exist in the common SSO flows.

1.2 Prototype implementation and evaluation Thus, while the exact application of the DDH idea to this particular type of SSO seems new, the cryptographic novelty is rather limited. This is not an issue, if the paper properly integrates this simple idea into SSO and demonstrate its viability for the considered use case. Unfortunately, this is not done in a satisfactory manner.

There is also a lot of room for improvement regarding the implementation and evaluation. In particular, discussing the user and RP integration of the new

functionality, as the construction now shifts significant parts of the protocol to the user.

3. Proofs of security and privacy - claims provable security, but there are no proper security models or proofs There are already a number of troubling high-level claims on the security of UPPRESSO:

- abstract and Sec 5.3 claim that UPPRESSO achieves the same security guarantees as OIDC and "UPPRESSO does not introduce any new role nor change the security assumptions for each role" This is not true: in standard OIDC, the user does not have to perform any crucial computations or elaborate certificate checks (beyond standard TLS authentication). However, UPPRESSO crucially relies on the user to verify certificates that the IdP issued to the RP, implicitly comparing it with the RP's TLS certificate to check it is the right RP, and extracting ID_{RP} from it. This is a critical part of the protocol in order to avoid phishing attacks. In standard OIDC, the user relies on the IdP to "authenticate" the RP and bind the id token to it.

The security analysis focuses on the unlinkability of pseudonyms which is expressed in three games that I found hardly intelligible. There is no proper description of the games and the figure is barely readable. From what I could extract, it looks rather static though, i.e., with only limited possibilities for the adversary to interact with honest parties.

The brief privacy analysis is then complemented by a security analysis, that discusses cookies and seems misplaced. What is missing is a clear description of the desired security and privacy properties and the respective corruption setting and sound justifications and assumptions for the underlying building blocks. In particular, the binding of id tokens to a designated RP now requires much more care and should be analyzed.

There is also a worrisome mismatch with related DDH-based OPRF/pseudonym constructions: All simple DDH-based OPRF constructions require a second layer of hashing and a Gap One-More type of assumption for the security proof – and a security model that takes the one-more-type of "forgeries" into account. OPRFs or blind pseudonym systems that achieve stronger security require zero-knowledge proofs of well-formed inputs. Neither of these approaches is used by UPPRESSO, but I would expect that a proper security analysis would reveal similar challenges here too.

2. Usability

- contains several unsubstantiated claims regarding usability of the proposed solution - compatibility with OIDC: "we require minimal modifications to the IdP and RP servers" OIDC outputs standard signatures to the RP, whereas UPPRESSO requires the RP to participate in an interactive protocol to blind and unblind the pseudonym, and check the related signatures. Overall, it is still a relatively lightweight protocol, but it is substantially different from OIDC and not compatible with any standard.

- UPPPRESSO & authorization code flow: "UPPRESSO can also support the authorization code flow of OIDC with small modifications" / "can be integrated into OIDC authorization code flow directly"

In the authorization code flow, the IdP server and the RP communicate

directly, which renders the authorization code flow unsuitable for any protocol where the IdP is not supposed to learn the RP's identity. The authors suggest to use TOR, but this would not be helpful as the IdP still need to know to whom it is talking.

In particular the first item is troubling, as this reliance on the user to verify and extract ID_{RP} is essential for the security of UPPRESSO, yet is hardly discussed in the construction or security analysis.

Review 1

1. Motivation

Lack of motivation or justification on the hardness of combining existing solutions that separately solves IdP-based login tracing and RP-based identifier linkage problems.

1. Contribution and related work Contributions are not clear for the individual building blocks.

This may not be the first protocol that solves both privacy issues. An existing work is not compared with.

Even though the privacy problems, which are being targeted to solve, are separately addressed by the prior art, it seems achieving privacy against both problems is not trivial because of the privacy dilemma behind the existing solutions. However, this point is not well-justified in the paper. Furthermore, it is not clear that if the individual solutions are also novel compared to the existing work.

The authors claim that this is the first system that solves both privacy problems at the same time. However, the following work also claims the same. I would like to see a comparison with this system.

"EL PASSO: Privacy-preserving, Asynchronous Single Sign-On"

How hard is combining existing solutions, SPRESSO and PPID, to prevent both privacy problems? Or are these two protocols not compatible with each other?

1. Writing Some sections are hard to follow due to excessive use of notations, and figures are not self-explanatory.

Some clarification on the text could improve the readability. For instance, it is not clear that PIDrp= NuIDrp at page 7, bullet 2.3, is a regular multiplication or an operation on elliptic curve cryptography.

1. Usability The paper also is not discussing about the user acceptance of this new design. Okay, it does not require major updates to build the whole system, but still requires modifications on IdP and RP sides. It would be very useful to include such an analysis about how the new protocol affects the usability of existing OIDC. Or, how easy it will be to deploy/convince RPs and IdPs for the new design.

What is the user acceptance and usability of your system compared to existing OIDC protocols?