We thank the reviewers for their time and constructive comments.

1. Reviewer-A: “The uniqueness of IDs could cause an issue depending on the number of users registered with the ID provider… How many registration efforts are needed by an attacker to impersonate a user?”

The collision probability of PID\_RP increases with the number of users (more precisely, the number of requests), hence, we measured it based on the number of unexpired identity proofs in the system, which is upper bounded by IdP’s throughput. In our current setting, the collision probability is negligibly small: 2^-183 (Section-VIII). As pointed out by the reviewer, a malicious RP can impersonate the user and generate many PID\_RPs to trigger a collision. This may cause a small increase in the collision rate, since the IdP’s throughput constraint still applies. We’ll extend our security analysis to discuss the effort for registering PID\_RPs and assess its impact to the collision probability.

2. Reviewer-A: “Technical work … is limited.”

Reviewer-B: “The second concern is about the novelty of this paper...”

Existing solutions only prevent either RP-based or IdP-based threats. More importantly, they are mutually exclusive and cannot be simply combined. This work identified the key challenge towards a comprehensive privacy solution, formalized it as an id-transformation problem, and proposed the trapdoor-based transformation solution. Therefore, we consider this work novel.

3. Reviewer-B: “(IdPs) tend to control which RPs they would like to cooperate with.” “... they also want to select which RPs they can provide users information to.”

We agree that IdPs may want to know the RPs and they even have this knowledge by default in some SSO systems, which raised privacy concerns about IdP-based tracking and caught attention from both academia (SPRESSO [1]) and industry (Mozilla’s BrowserID [2]). Moreover, we argue that this knowledge of RP may be useful but is not necessary for IdP to provide SSO services. For example, in OIDC, when an RP requests user information using the identity request’s “scope” field, IdP should consult user to obtain an explicit consent. But the IdP doesn’t need to know the RP’s identity as the user is already in the loop. Hence, the concern about privacy risk may outweigh the potential benefit.

4. Reviewer-B: “if the tackled privacy issue is a real threat ... (users) are expecting IdP to provide their true ID\_U to the RP...”

Reviewer-D: “How practical a threat is RP-based identity linkage?”

Some users may willingly share personal information with RPs, but identifiable information such as email is commonly considered as privacy, especially by privacy-savvy users. Therefore, a solution against RP-based linkage is expected. In fact, the RP-based linkage threat is widely recognized in the literature on federated identity management and SSO, and PPID is well-accepted to prevent this threat. For example, OIDC and SAML include PPID in their specifications to prevent possible correlation among RPs; Active Directory Federation Services and Oracle Access Management support the use of PPID [3,4]; identity service providers such as NORDIC APIS and CURITY suggest adopting PPID in SSO to protect user privacy [5,6]. However, with rising concerns about the new IdP-based tracing threat, new challenges arise as PPID cannot be directly integrated into existing solutions to prevent both threats.

5. Reviewer-B: “Can a malicious RP trick users by letting them download a script ...”

The script is downloaded from the honest-but-curious IdP, hence, a malicious RP cannot directly trick users to download a malicious script. Presumably, the RP could pretend to be an IdP and trick users to download the script from it. However, this malicious script cannot obtain any information (e.g., identity proof) from the honest IdP because of the same origin policy.

6. Reviewer-B: “UPPRESSO triples the processing time of MITREid.”

Reviewer-D: “whether the load of the SSO server is significantly affected …”

We agree on the tradeoff between processing time and privacy protection. Our experiments showed that UPRESSO took 492ms to complete a request. This latency is still acceptable, considering users rated the QoS level of Internet applications with delays less than 5 seconds as “High” [7] and the average Internet round-trip time is 200ms. Moreover, UPPRESSO purposely shifts most of the time-consuming operations to the client-side to avoid overloading the server. The IdP and RPs are required to perform only one and two additional modpow operations (on average less than 10ms each), respectively. We are working on additional experiments to evaluate the server-side overhead. We’ll report the results in the revision if granted the opportunity.

7. Reviewer-C: “you use residue classes over the integers for the discrete log problem, instead of elliptic curves…”

We thank the reviewer for this constructive suggestion. We chose residue classes over the integers simply for fast implementation considerations. We agree the solution based on elliptic curves would be more efficient while achieving the same level of security, so we plan to improve our solution on elliptic curves in the revision of our work. We estimate this would reduce the overall processing time up to 10%. Meanwhile, we consider the main contribution of our work is the new protocol against two SSO privacy threats and the proposed approach achieves this design goal.

8. Reviewer-C: “... the situation here may be different… And here's the attack which is not covered by your analysis ...”

We use the computational hardness assumption about the discrete logarithm problem to prove that g^x “looks” random to the adversary who has no knowledge about x. This is slightly different from the DDH assumption. Hence, in definition 1, the equivalence of the modpow functions with different exponents and different bases holds, if the adversary cannot control or infer the exponent. In particular, (1) PID\_RP=ID\_RP^N\_U looks random to a curious IdP, because a random N\_U is chosen for each request by the user who has no intention to manipulate or leak N\_U; (2) both PID\_U= ID\_RP^(N\_U\*ID\_U) and Account= ID\_RP^ID\_U are random to collusive RPs, because ID\_RP and ID\_U are chosen by the honest-but-curious IdP and ID\_U is never disclosed to the user nor RPs. Therefore, the attack pointed out here won’t succeed. We’ll add this discussion to the security analysis.

9. Reviewer-C: “… hard to understand what security guarantees your formal analysis gives.”

We used the discrete logarithm problem as the basis to prove the security of the transformation functions. As this property is used in many cryptographic protocols, we only briefly included it in the paper but did not give formal analysis. Instead, our security analysis focuses on the security of the proposed protocol using the Dolev-Yao model, which has been widely used in SSO system analysis [1,8,9]. We will improve the security analysis to clarify the security guarantees provided by our design.

[1] Fett et. al., “SPRESSO: A secure, privacy-respecting single sign-on system for the web,” in ACM CCS, 2015.

[2] <https://github.com/mozilla/id-specs/blob/prod/browserid/index.md.>

[3] <https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/technical-reference/the-role-of-claims>

[4] <https://docs.oracle.com/cd/E40329_01/admin.1112/e27239/oif_1.htm>

[5] <https://nordicapis.com/build-gdpr-compliant-apis-with-openid-connect/>

[6] <https://curity.io/resources/architect/openid-connect/ppid-intro>

[7] Bouch et. al., “Quality is in the eye of the beholder: meeting users' requirements for Internet quality of service,” in SIGCHI 2000.