We thank the reviewers for their constructive comments and will try our best to answer their questions.

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?*”

We agree with the reviewer that the collision probability of PID\_RP increases with the number of users (or more precisely the number of total requests). So, in our scalability discussion, we measured this probability based on the number of unexpired identity proofs in the system, which is upper bounded by the throughput of the IdP. In our current setting, the collision probability is negligibly small, i.e. 2^(-183). A malicious RP may impersonate or collude with a malicious user to generate many PID\_RP to increase the collision rate, but the IdP throughput constrain still applies. We thank the reviewer for pointing this out and will extend our security analysis to discuss the registration effort of a malicious RP and its impact.

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

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

While solutions to prevent either RP-based or IdP-based threats have been proposed, we consider this work novel because existing solutions against two threats are mutually exclusive and cannot be simply integrated. 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.

1. 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. This raised privacy concerns about IdP-based tracking and caught attention from both academia (SPRESSO [1]) and industry (Mozilla’s BrowserID protocol [2]). Moreover, we argue that this knowledge of RP could be useful but is not necessary for the IdP to provide SSO services. For example, in OIDC, when RP requests user information in the “scope” field of an identity request, the IdP should consult the user to obtain an explicit consent. The IdP is not required to know RP’s identity because the user is in the loop. So, the concern about privacy risk may outweigh the potential benefit.

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

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

We agree with the reviewers that 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. So, it is important to provide a solution against RP-based linkage. 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 protect user from possible correlation among RPs; Active Directory Federation Services (AD FS) 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]. The problem is that, with the raising concerns about the new IdP-based tracing threat (e.g., SPRESSO and BrowserID), PPID cannot be directly integrated into existing solutions for IdP-based tracing.

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

The script is downloaded from honest-but-curious IdPs, so a malicious RP cannot directly trick users to download a malicious script. To do so, the RP may pretend to be an IdP and trick users to download a malicious script from it. However, this malicious script cannot obtain any information (e.g., identity proof) from honest IdPs due to the same origin policy.

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

Reviewer-D: “*whether the load of the SSO server is significantly affected for operating such UPPRESSO?*”

We agree with the reviewers on the tradeoff between processing time and privacy protection. Our experiments showed UPPRESSO took 492ms to complete a request. We think this is still an affordable latency for Internet applications, considering the average round trip time on the Internet is about 200ms and users rated the QoS level as “High” for delay less than 5 seconds [7].

Also, UPPRESSO purposely shifts most of the time-consuming operations to the client-end 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 IdP and RP servers’ throughput and will report the results in the revision if allowed.

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

We thank the reviewer for this very 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. Adopting elliptic curves in the id-transformation module could reduce the processing time of modpow operations and improve the performance by 10% as we estimate. 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.

1. Reviewer-C: “*... the situation here may be different because the values may be under adversarial control.”*

This concern is valid only if the attacker can set the exponent or base to a few special values and thus break the randomness of g^x. However, in UPRESSO, any single adversary cannot independently control the selection of the exponents and bases. For example, collusive RPs, who cannot infer and control IdP-chosen ID\_U, may want to tamper with ID\_RP to correlate a user’s Account at these RPs. However, ID\_RP is in the RP’s certificate issued by semi-honest IdP and verifiable by the user. Meanwhile, a curious IdP may want to derive ID\_RP from PID\_RP, however, the exponent N\_U is determined by the user, who has no intention to manipulate N\_U. A malicious RP may pretend to be the user to infer ID\_U, however, semi-honest IdP never leaks ID\_U to users either.

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

We use the computational hardness assumption about the discrete logarithm problem as the basis to prove the security of the transformation functions -- it is computationally infeasible for a malicious RP to compute ID\_U from ID\_RP^( ID\_U) and for a curious IdP to compute ID\_RP from ID\_RP^N\_U. As this property is adopted in many cryptographic protocols, we only briefly discussed it in the paper but did not formally prove it. Instead, our security analysis focuses on the security of the proposed protocol based on 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 and add discussions on potential attack scenarios.

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

[2] Fett et al., “Analyzing the BrowserID SSO system with primary identity providers using an expressive model of the web,” in ESORICS 2015.

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

[4] Introducing Identity Federation in Oracle Access Management. https://docs.oracle.com/cd/E40329\_01/admin.1112/e27239/oif\_1.htm#AIAAG6499

[5] [Build GDPR Compliant APIs with OpenID Connect](https://nordicapis.com/build-gdpr-compliant-apis-with-openid-connect/). https://nordicapis.com/build-gdpr-compliant-apis-with-openid-connect/

[6] Pairwise Pseudonymous Identifiers. 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.

[8] Fett et. al., “A comprehensive formal security analysis of OAuth 2.0,” in ACM CCS 2016.

[9] Fett et.al., “The web SSO standard OpenID Connect: In-depth formal security analysis and security guide- lines,”in IEEE CSF 2017.