Dear Shepherd,

We have submitted the revised manuscript for your review. This revision follows the suggested “revision requirements” and our proposed revision plan. We summarize the main changes as follows.

**Revision requirements:**

1. **Remove or clean-up section 3.2.**

*Modifications:*

1. We removed Section 3.2 “The Identity dilemma of Privacy-Preserving SSO” in the original manuscript.
2. In this manuscript, we highlighted the security requirements of SSO in Section 3.1 and explained the requirements of identity transformation in Section 3.2.
3. **Refine the security analysis: define proper notions to prove if you do a formal proof and provide better intuitions on the challenges in proving each notion w.r.t the chosen adversarial model. Make clear which assumptions each property relies on (e.g., pull them all into a theorem)**

*Modifications:* We heavily revised the security and privacy analysis in Section 5.

1. The threat model in Section 4.1 considers four types of adversaries, an honest-but-curious IdP, malicious RPs, malicious users, and colluding RPs and users. Following this threat model, we added a new subsection 5.1 to define three adversarial scenarios.
   1. We considered colluding RPs and users who aim to intrude an honest user’s login flow in two different cases. The attack goal is to break the security of SSO. We did not consider the case with only malicious users or only malicious RPs, as it is straightforward to 扩展原有的安全结论. We prove the security of UPPRESSO in this adversarial scenario.
   2. We consider the honest-but-curious IdP that aims to infer the identity of the RPs. We prove that UPPRESSO prevent IdP-based login tracing in this adversarial scenario.
   3. We consider the colluding RPs and users who aim to associate the logins to colluding RPs to link an honest user’s accounts across these RPs. UPPROSSO is proved against RP-based identity linkage in this adversarial scenario.
2. In Table 3, we defined the notations used in the manuscript. We clarified the assumptions about the random numbers used in the UPPRESSO protocol.
3. We improved Section 5.2 to describe how to develop a Dolev-Yao style model to formally model the processing of messages and their attributes in the UPPRESSO protocol. Applying this model, we drew six conclusions to show that the identity tokens, pseudo-identities, random variables, etc. cannot be accessed or modified by unauthorized entities in the proposed protocol.
4. We revised Section 5.3 to prove the security of UPPRESSO. First, we defined four theorems to prove that UPPRESSO supports each of the four security requirements defined in Section 3.1, respectively. Then, we proved Theorem 5 to show that any SSO protocol satisfying these four requirements is considered secure.
5. We revised Section 5.3 to prove that UPPRESSO prevents two privacy threats. We proved Theorem 6 and Theorem 7 to show its protection against IdP-based login tracing and RP-based identity linkage, respectively.

1. **Clarify the protocol design in the write-up (i.e., the check in 3.3, and the double computation of PID\_RP)**

*Modifications:*

1. We clarified the statement in Step 3.3 by adding a sentence: “the IdP checks if the received $PID\_{RP} is a point on E”. How to verify if a point is on E or not? 放到椭圆曲线方程中，验证一下y2=x3+ax+b即可；应该不需要写？
2. We removed the calculation of PID\_RP in Step 2.2. However, this calculation cannot be completely avoided in the protocol, because in Step 4.2, the RP is required to “verify if PID\_RP in the received identity token matches the pseudo-identity negotiated in Step 2.2”. In this manuscript, we moved this calculation of PID\_RP to Step 4.2: “The RP extracts PID\_RP from the token and checks if it equals [t]ID\_RP”, to make the protocol logic clearer to follow.
3. It is important for the RP to check if the received PID\_RP is associated with the current login (i.e., t) and for itself (i.e., ID\_RP). Otherwise, the attacker can manipulate PID\_RP to break RP designation or user identification. We added two paragraphs in Section 4.3 to explain two possible attacks.
4. Elaborate on the weakness of having t in the IdP context or integrating a suitable MPC scheme, as noted in the response.

*Modifications:*

1. We added a paragraph in Section 4.4 to explain that t is critical for the RP and the user (using the IdP script) to calculate the PID\_RP independently. Based on our threat model, the IdP is honest-but-curious, so the IdP script is also considered honest. We explained that the IdP script should be trusted because it knows the RP’s domain which reveals the RP’s identity. Therefore, MPC-based solutions are not necessary.
2. We discussed an alternative solution of using browser extension instead of the IdP script. We do not use it because it requires users’ agreement and involvement to install the extension for each new RP and IdP (假设协议已经标准化，原则上，应该for each IdP；需要预装该IdP的公钥；事实上，我们在性能实验中，有说过这个extension；之前的投稿版本就有，在page 13的footnote).
3. Illustrate the differences compared to PrivacyPass [27] and Trust-Tokens [26], and highlight where the protocol actually shows significant novelty.

*Modifications:*

1. We revised Sections 2.2 to provide a conceptual categorization of the existing work related to privacy-preserving identity management systems into privacy-preserving SSO, privacy-preserving identity federation, and anonymous identity federation. While…
2. In Section 2.3 we compared the cryptographic technique used in PrivacyPass/TrustToken and our work.

We plan to improve the discussion of related work in Section 2.3 and add comparison with recent related work, including PrivacyPass [27] and Trust-Tokens [26].

\textbf{Revision 5.1:} In particular, we consider PrivacyPass and Trust-Tokens as “anonymous tokens”. PrivacyPass [27] and Trust-Tokens [26], similar as UPRESSO, adopt cryptographic techniques proposed in [1] for OPRFs. We will add discussion about this comparison in terms of cryptographic constructs and techniques in Section 2.3.

\textbf{Revision 5.2:} Then, we plan to elaborate the novelty of UPRESSO, compared with PrivacyPass and Trust-Tokens, from three aspects: (1) UPPRESSO identifies each user at an RP, while PrivacyPass and TrustToken anonymous tokens support anonymous SSO service, which rely on one consistent private key to serve all users. (2) UPPRESSO utilizes the cryptographic technique differently from PrivacyPass and Trust-Tokens, by using them to transform identities in SSO. (3) UPPRESSO supports more privacy requirements, i.e., the unlinkability across RPs, than anonymous SSO enabled using PrivacyPass and Trust-Tokens. This property of the cryptographic skills is not considered in either OPRFs or anonymous tokens.

We will add this new reference to the revised manuscript.

[1] S. Jarecki,A. Kiayias,H. Krawczyk,and J. Xu, "Highly-efficient and composable password-protected secret sharing (or: How to protect your Bitcoin wallet online)," in 1st IEEE European Symposium on Security and Privacy (EuroSP), 2016, pp. 276–291.

这个第一次仓促放上的参考文献

M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. Journal of the ACM, 51(2):231–262, 2004.

S.Jarecki,A.Kiayias,andH.Krawczyk. Round-optimal password-protected secret sharing and T-PAKE in the password-only model. In 20th International Conference on the Theory and Application of Cryptology and Information Security (AsiaCrypt), pages 233–253, 2014.

更准确的，应该是上面的2篇。原来写上去的那篇，其实是Round-optimal password-protected secret sharing and T-PAKE in the password-only model的应用方案。

1. **We carefully polished the writing of the manuscript.**

*Modifications:*

We made small edits throughout the paper to improve its presentation. To avoid distractions, we did not mark these edits in color blue.