Revocation and VC

Ahamed Azeem, azeem.ahamed@danubetec.com

1 Introduction

Revocation is an essential requirement for a widely usable and acceptable identity management system and an essential self-sovereign Identity (SSI) feature. Verifiable credentials (VC) are digital versions of physical credentials with additional features such as tamper-proof and privacy-preserving [1]. Revocation used in VCs should preserve privacy.

Problem

Different revocation methods are used within SSI, such as Revocation List 2020 [4], Hyper-Ledger Indy revocation [3], BBF18- cryptographic accumulator based on RSA [2], Non-Revocation Token [8], and Ether Status Registry 2019 [7]. Some revocation methods are designed for the VC ecosystem and some specific Anoncred [3]. These revocation methods use underline technologies such as accumulators, smart contracts, and bit string compression. Due to the different underlining technologies, each revocation method can suit different use cases.

VC uses the "credentialStatus" field to present and verify the revocation status in a VC. The "credentialStatus" is a claim asserted by the issuer during the issuance of a VC. Hence, this is not possible to alter or derive a different value during the presentation of the VC. The revocation method, such as revocation list 2020 and ethr status registry 2019, uses the value issued by the issuer during the presentation. The values related to revocation proofs are user-specific because these values denote the revocation status in a registry. A static value for revocation can lead to privacy violations such as linkability and identifiability.

Approach towards the solution.

A solution to overcome the privacy issue caused by the revocation status values is dynamic values during the presentation. Therefore, the dynamic value is hard to correlate or identify with the same holder.

The revocation methods, such as Indy Revocation, and Non-Revocation Token, provide the dynamic presentation of revocation status. However, with the VC data model, it is hard to derive a dynamic value for "credentialStatus".

Dynamic revocation status proof generation with holder binding will enable privacy-preserving non revocation proof with zkp. A solution using BBS+ signature is possible because BBS+ signature supports zkp-based presentation.

Acknowledgements

Part of Danube Tech's work on DIDs, SSI, etc. has been supported by eSSIF-Lab.

References

[1] David Chadwick Manu Sporny Dave Longley. “W3C standard for Verifiable Credentials Data Model 1.0”. In: (2021). url: https://www.w3.org/TR/vcdata-model/ [2] Mike Lodder. Privacy Preserving Revocation Proposal. 2021-07-31. url: https : / / hackmd . io / O4c3wiLZQLeXuXirm7dl9A # Math - explaination (visited on 04/14/2022). [3] Andrew Tobin. Sovrin: What Goes on the Ledger. 2018. url: https:// www.evernym.com/wp-content/uploads/2017/07/What-Goes-On-TheLedger.pdf (visited on 11/14/2021). [4] Manu Sporny and Dave Longley. Revocation List 2020. 2021. url: https: //w3c-ccg.github.io/vc-status-rl-2020/ (visited on 02/22/2022). [5] Kim Wuyts, Riccardo Scandariato, and Wouter Joosen. “LIND (D) UN privacy threat tree catalog”. In: Department of Computer Science, KU Leuven (2014). [6] Matt Raffel Stephen Curran. “AnonCreds Specification”. In: (2021). url: https://anoncreds-wg.github.io/anoncreds-spec/. [7] Mircea Nistor. Revocation Registry 2019. 2020. url: https://github.com/ uport-project/revocation-registry (visited on 02/22/2022). [8] Stephan Curran and Andrew Whitehead. Non-Revocation Token. 2021. url: https://hackmd.io/kj223D1ZQN29WiusmnPFMA.