Attribute-based Access Control on Solid Pods using Privacy-friendly Credentials

Christoph H.-J. Braun[1,*], Tobias Käfer[1]

[1]Institute AIFB, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, 76131 Karlsruhe, Germany

Abstract

Our demo showcases how a user is granted access to resources stored on a Solid Pod, i. e., a web server that adheres to the Solid Protocol, using Web-based Verifiable Credentials. To protect the privacy of the user, we rely on the BBS+ signatures scheme allowing for selective disclosure of only those attributes necessary. We present a PWA where a user can (a) request a Verifiable Credential from another user, (b) store it on their own Solid Pod, and (c) use it to gain access to a resource on a third user's Solid Pod.

Keywords

Linked Data, Solid, Attribute-based Access Control, Verifiable Credentials, Selective Disclosure

Website https://purl.org/uvdsl/semantics22demo
Code https://github.com/uvdsl/solid-vc-pwa/, https://github.com/uvdsl/solid-vc-module/

1. Introduction

With the recent trend of Self-Sovereign Identity [1], digital credentials see increasing efforts of adoption from industry and government, e. g., for digital drivers licenses [2] or digital health certificates [3], while often controversially relying on distributed ledger technologies [4]. At the same time, the ecosystem around the Web-based project Solid[1] is rapidly evolving: A new way of defining access control policies (ACPs) [5] in Solid is being specified, already mentioning Verifiable Credentials (VCs) [6] as one potential component.

Our demo showcases how VCs can be used for attribute-based authentication and authorization, i. e., access control, on Solid Pods[2]. We focus on ensuring the user's privacy with regards to data minimisation by allowing for selective disclosure of only those attributes that are necessary for authentication and authorization. Our demo relies on the Solid Protocol [7]: We use WebIDs [8], i. e., a URI that identifies a user, Linked Data Notifications (LDNs) [9] for communication between users, and Solid Pods for storing personal information, credentials and so on. We follow the W3C recommendation *Verifiable Credentials data model* [6] and showcase the application of the BBS+ signature scheme [10] to allow for selective disclosure of attributes. We present a proof-of-concept server module and a Progressive Web App (PWA) where the user

SEMANTICS 2022 EU: 18th International Conference on Semantic Systems, September 13-15, 2022, Vienna, Austria
*Corresponding author.
braun@kit.edu (C. H.-J. Braun); tobias.kaefer@kit.edu (T. Käfer)
0000-0002-5843-0316 (C. H.-J. Braun); 0000-0003-0576-7457 (T. Käfer)

[1]https://solidproject.org
[2]Pod (personal online data storage), a web server adhering to the Solid Protocol

- requests a Verifiable Credential from another Solid user,
- stores the credential in their own Solid Pod,
- uses it to gain access to a resource stored on a third user's Solid Pod,
- while only disclosing the necessary attributes.

This paper is structured as follows: First, we discuss related work. Next, we illustrate our demo with an example. Then, we present a demo walkthrough. Hereafter, we discuss the current system architecture and outline future work to improve our current prototype.

2. Discussion of Related Work

We briefly survey related work in the realm of Verifiable Credentials (VCs) and Solid. An early description about the Solid project is provided in [11]. The Verifiable Credential data model [6] is a recent W3C recommendation for sharing verifiable claims. Linked Data Proofs, which had been mentioned by the VC specification as a valid signature scheme, have been renamed Data Integrity[3], now noting the usage of Linked Data only as an optional feature.

Combining Solid and VCs, Ezike present a system for issuance, handling and revocation of VCs [12]. Ezike envision an ecosystem of applications where access to restricted services and resources would be granted relying on such VCs. As credential signatures, however, only simple JSON-LD signatures are used. This poses a privacy issue regarding data minimisation when more information than necessary would be disclosed with a presented credential. In our work, we take a step towards the declared vision by showcasing how a user can be authenticated and subsequently granted access to resources under access control. At the same time, we address data minimisation by supporting selective disclosure via BBS+ signatures [10].

The BBS+ signature scheme [10] is based on the strong Diffie-Hellman assumption for cryptographic hardness, as first presented by Boneh et al. in [13]. Similar in goal but based on a different hardness assumption, Camenisch and Lysyanskaya presented CL signatures [14], which are typically used in "anonymous credentials" to achieve anonymity of the holder [15]. Camenisch and Lysyanskaya also describe an approach to anonymous credentials using BBS where modifications are necessary [15], which are provided by BBS+ [16]. Thus, BBS+ is also suitable to create anonymous credentials which we have not yet implement in our system.

Access control in Solid is implemented using Access Control Lists (ACL) [17]. With ACLs, different modes of access to resources can be granted for specific agents or agent groups. A new alternative is introduced with Access Control Policies (ACP) [5], whose specification is still under development. The current ACP draft specification mentions VCs in the context of access requests. Our prototype showcases how such VCs can be used for access control on Solid Pods.

SSIBAC [18] was presented as a system for access control based on Self-Sovereign Identities. The system fundamentally relies on distributed ledger technology for managing identifiers and associated cryptographic keys to provide attribute-based access control on resources in the information system. Their credentials specifically rely on the Hyperledger Indy[4] framework. In contrast, our system relies on Web standards and the Solid Protocol.

[3] https://w3c-ccg.github.io/data-integrity-spec/
[4] https://www.hyperledger.org/use/hyperledger-indy

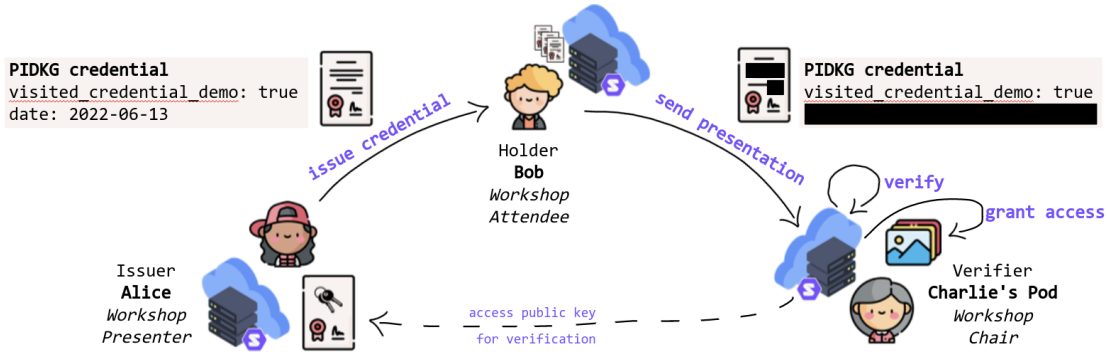

Figure 1: A high-level demo overview illustrating the provided functionality. (attribution: see footnote[5])

3. Illustrating Example

Consider the setting[6] depicted in Figure 1: Charlie is organising a public workshop with many attendees. Alice is one of the demo presenters at the workshop. Whoever attended the demo may access a digital badge, provided by Charlie. As the attendees can choose on-site which demo to attend, Charlie cannot know beforehand who attended, e. g., Alice's demo. Instead, Alice issues a corresponding VC to her attendees, one of which is Bob. Bob can then present this VC to Charlie's Pod. However, the VC contains the time when the attendee visited Alice's demo, but Charlie only requires to know if (and not when) Alice's demo was attended. Bob can selectively disclose only the necessary attributes and is granted access to the digital badge. A more serious example would be that both the birth date of Bob and some health information are stored in the same RDF graph, but Bob only wants to disclose the former to provide proof of age to a doorman.

4. Basic Demo Walkthrough

Adhering to the Solid Protocol, users are identified by a WebID and store their data, e. g., credentials and associated keys, on a Solid Pod under access control. In the demo, a user takes the role of *Bob* from Figure 1. Our PWA and server module provide the functionality described.

The user logs in to our PWA with their WebID. Access to the demo resource, i. e., the digital badge from chapter 3, is denied by the Pod serving the resource. As no credentials are stored in the user's wallet, a new credential to "unlock" the demo resource is requested: An LDN with a corresponding request is sent to a (for the demo predefined) agent, i. e., *Alice* from Figure 1.

Alice processes the request LDN, and creates a new credential containing her WebID as the issuer, the URI of the public key for verification, and claims about the user (identified by their WebID). The credential is then issued to the inbox of the requesting user's Pod.

Upon receiving the credential, the user saves it to their wallet. To unlock access to the demo resource, the user selects the credential and specifies which attributes should be disclosed to

[5]Comic icons from flaticon.com; created by Freepik, except for the cloud server which was created by vectorsmarket15.
[6]A visitor to our booth at the conference will experience this example first-hand.

the agent controlling the demo resource, i. e., *Charlie*. The newly derived credential[7], only containing the selected attributes and additionally signed by the user, is sent to Charlie's Pod.

The LDN is processed by our proof-of-concept server module checking if (a) the credential received is valid and verifiable, (b) the necessary attributes are disclosed and (c) if the credential was actually issued to and signed by the agent sending the access request. To this end, all WebIDs are dereferenced and the associated public keys for verification are retrieved. If the user can thus be authenticated, access to the resource is authorized. The user is notified via an LDN that the resource is now accessible.

5. Discussion of System Architecture

With the current architecture, we aim to provide an early proof-of-concept for attribute-based access control with focus on data minimisation rather than a production-ready implementation:

For example, access to a resource is granted asynchronously to the actual resource access. We envision that the outlined verification procedure is directly tied to HTTP requests. Instead of LDNs, the required information is exchanged within the HTTP requests and responses, e. g., using headers. This way, any relevant information regarding access control can directly be exchanged between the agents. Moreover, any inconsistency regarding the user's authorization, e. g. in case of expiring or revoked credentials, are avoided. Modelling access control rules to express, e. g., which attributes of a credential are required, is left for future research, as even ACPs [5] are still evolving.

An additional issue poses the secure storage of credentials on the Solid Pod. Typically, credentials are stored on the user's device outside of the control of others. Providing Solid-based cloud-stored credentials should be similarly secure and reliable, especially when considering that most users do not host the Solid Pod themselves but rely on third party Pod providers. Corresponding trade-offs need to be considered when deciding where to store the credentials.

To improve users' privacy beyond selective disclosure, we also look at providing anonymous credentials [15]. Currently, the WebID of the holder needs to be disclosed for authentication. With anonymous credentials, the holder can prove the attribute required for resource access without revealing his identity. Enabling additional Zero-Knowledge Proofs, e. g., range proofs or set membership proofs, for Linked Data-based credentials poses also promising future research.

6. Conclusion

In this demo, we showcased a proof-of-concept for attribute-based access control on a Solid Pod using Verifiable Credentials. Moreover, those credentials are Web-based and allow for selective disclosure to provide data minimisation. We presented a server module and a PWA showcasing such privacy-friendly credentials for access control on a Solid Pod. We identified promising directions for future research and hope to contribute to a privacy-friendly Solid ecosystem.

[7]For an example, the interested reader may take a look at our website linked on the first page.

Acknowledgments

This work is supported in part by the German federal ministry of education and research (BMBF) in MANDAT (FKZ 16DTM107B). We thank Leo Floegel for help in an early stage of the work.

References

[1] C. Allen, The path to self-sovereign identity, 2016. URL: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.

[2] Federal Ministry for Digital Affairs and Transport of Germany, Technik für digitalen führerschein steht, 2021. URL: https://www.bmvi.de/goto?id=486282.

[3] Lissi, Datev, Case study, 2022. URL: https://link.medium.com/tyJEdTJhrsb.

[4] H. Halpin, Vision: A critique of immunity passports and W3C decentralized identifiers, in: Proc. of the 6th SSR, volume 12529 of *LNCS*, Springer, 2020, pp. 148–168.

[5] M. Bosquet, Access Control Policy (ACP), Editor's Draft, W3C Solid CG, 2022. URL: https://solid.github.io/authorization-panel/acp-specification/.

[6] M. Sporny, G. Noble, D. Longley, D. C. Burnett, B. Zundel, K. Den Hartog, Verifiable Credentials Data Model, Recommendation, W3C, 2021. URL: https://www.w3.org/TR/vc-data-model/.

[7] S. Capadisli, T. Berners-Lee, R. Verborgh, K. Kjernsmo, Solid Protocol, Version 0.9.0, W3C Solid CG, 2021. URL: https://solidproject.org/TR/protocol.

[8] A. Sambra, H. Story, T. Berners-Lee, WebID 1.0 - Web Identity and Discovery, W3C Editor's Draft, W3C, 2014. URL: https://www.w3.org/2005/Incubator/webid/spec/identity/.

[9] S. Capadisli, A. Guy, Linked Data Notifications, Recommendation, W3C, 2017. URL: https://www.w3.org/TR/ldn/.

[10] T. Looker, O. Steele, BBS+ Signatures 2020, Draft CG Report, W3C Credentials CG, 2022. URL: https://w3c-ccg.github.io/ldp-bbs2020/#the-bbs-signature-suite-2020.

[11] E. Mansour, A. V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga, T. Berners-Lee, A demonstration of the solid platform for social web applications., in: Proc. of Posters & Demos at the 25th WWW, ACM, 2016, pp. 223–226.

[12] K. Y. Ezike, SolidVC : a decentralized framework for Verifiable Credentials on the web, Master's thesis, MIT EECS, 2019. URL: https://hdl.handle.net/1721.1/121667.

[13] D. Boneh, X. Boyen, H. Shacham, Short group signatures, in: Proc. of the 24th CRYPTO, volume 3152 of *LNCS*, Springer, 2004, pp. 41–55.

[14] J. Camenisch, A. Lysyanskaya, A signature scheme with efficient protocols, in: Revised Papers of the 3rd SCN, volume 2576 of *LNCS*, Springer, 2002, pp. 268–289.

[15] J. Camenisch, A. Lysyanskaya, Signature schemes and anonymous credentials from bilinear maps, in: Proc. of the 24th CRYPTO, volume 3152 of *LNCS*, Springer, 2004, pp. 56–72.

[16] M. H. Au, W. Susilo, Y. Mu, Constant-size dynamic k-taa, in: Proc. of the 5th SCN, 2006.

[17] S. Capadisli, Web Access Control, Editor's Draft, W3C Solid CG, 2022. URL: https://solid.github.io/web-access-control-spec/.

[18] R. Belchior, B. Putz, G. Pernul, M. Correia, A. Vasconcelos, S. Guerreiro, SSIBAC: self-sovereign identity based access control, in: Proc. 19th TrustCom, IEEE, 2020, pp. 1935–1943.