Self-Verifying Web Resource Representations using Solid, RDF-star and Signed URIs

Christoph H.-J. Braun[0000−0002−5843−0316]* and Tobias Käfer[0000−0003−0576−7457]

Institute AIFB, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
`braun@kit.edu, tobias.kaefer@kit.edu`

Abstract. Our demo showcases a Solid-based Web where the integrity of a Web resource's representation is directly verifiable using its content and its identifier: A Web resource is available at some URI described in RDF. Each such representation includes a Linked Data Signature, which we model using RDF-star. In addition, each Web resource's URI includes the signature's value as a suffix, which we call *Signed URI*. In a Web of such resource representations, modifications to a resource are detectable unless all resources that transitively reference the original are updated as well. We present a Solid-based Web application where such a Web of resources with self-verifying representations can be created and verified.

Website `http://people.aifb.kit.edu/co1683/2022/eswc/`
Code `https://github.com/uvdsl/solid-web-ldsig`

1 Introduction

Anybody can publish anything on the Web; and later modify or delete it. No commonly accepted mechanism ensures that the published data is not altered after it was first made accessible on the Web. In addition, declaring authorship of information on the Web is not required by design.

However, with the recent trend of Self-Sovereign Identity [1] and the growing Solid ecosystem [9], users claim more control over their digital life. The ability to express verifiable information on the Web in a self-sovereign manner is becoming evermore important. While some may sense a possible blockchain use-case, as previously presented in [2], we argue: The Web is all we need.

Our demo is built with the following components: We rely on the Solid Protocol [3]. Web resource representations in RDF are stored in the user's personal online data storage (Pod). These representations are signed by the user using a Linked Data Signature, which models all information necessary for verification. To this end, we propose to use RDF-star [5] for modeling Linked Data Signatures. Inspired by Trusty URIs [8], we use a suffix in the Web resource's URI, where we include the signature value, instead of the content hash as with Trusty

* Corresponding author

URIs. We call such a URI a *Signed URI*. When accessing a Signed URI, a user can expect a specific content with matching signature value to be served. In other words, Signed URIs allow users to reference a resource together with its state. This way, a graph of signed resource representations is created. A change in resource state thus becomes detectable, unless all resource representations that transitively reference the changed resource, are updated as well. In this demo, we showcase:

- Creation and Verification of Linked Data Signatures using a Solid App.
- Modeling of Linked Data Signatures using RDF-star.
- Signed URIs, which allow to reference a resource in a specific state.

This paper is structured as follows: First, we give a short overview on related work. Next, we provide a very basic walkthrough of our demo. Then, we touch on modeling Linked Data Signatures. Hereafter, we ponder the implications of Signed URIs. Finally, we conclude.

2 Related work

We briefly survey related work in the realm of Linked Data integrity and Solid. An early description about the Solid project is provided in [9].

The integrity of Linked Data is typically verified by calculating and comparing cryptographic hashes of the underlying RDF graphs. We use the algorithm of Hogan [6] for graph canonicalisation as it handles blank nodes most gracefully.

The W3C recently released the Verifiable Credential (VC) data model [10] as a recommendation for sharing verifiable claims. We do not use the VC data model as we concentrate more on the signature itself than on the claim. Mentioned by the VC recommendation as a valid signature scheme, the draft specification of Linked Data Proofs has been renamed Data Integrity[1], now noting the usage of Linked Data only as an optional feature. Also in recent discussion, a proposal for standardization of Linked Data Signatures[2] has sparked fierce discourse within the community, albeit recognizing the need for standradized RDF canonicalization and hashing.

Trusty URIs [8] aim to make digital resources verifiable, immutable and permanent by by extending the usual URI scheme with a cryptographic hash of the resource as a URI suffix. Our approach builds on and extends the conceptional idea of Trusty URIs with digital signatures to *Signed URIs*.

Nanopublications [7] aim at publishing data on the Web using Trusty URIs such that links among nanopublications contribute to data integrity. With only hash values ensuring integrity of data, authorship of publications is not preserved. The centralised yet distributed nanopublication-server-network ensures publications' discoverability, permanence and immutability.

Web Publishing using Named Graphs has first been proposed by Carroll et al. [4]. The approach relies on ontology-defined terms to indicate if a graph is to be interpreted as assertative or non-assertative by an information consumer. We

[1] `https://w3c-ccg.github.io/data-integrity-spec/`
[2] `https://lists.w3.org/Archives/Public/semantic-web/2021Oct/0020.html`

use RDF-star [5] instead of Named Graphs, thereby allowing the information creator to normatively define which triples are (non-)assertative.

3 Basic Demo Walkthrough

Adhering to the Solid Protocol, users are identified by a WebID and store their data on a personal online data storage (Pod) under access control. A user logs in to our demo app with their WebID. The user then creates an RDF graph and signs it with a Linked Data Signature: First, the created RDF graph is canonicalised using the algorithm of Hogan [6]. Next, SHA-256 is applied as message digest and the Elliptic Curve Digital Signature Algorithm (ECDSA) with curve P-256 creates the signature value. The app takes care of creating, storing and handling the user's cryptographic keys in the user's Solid Pod under access control. The signature value is appended to the URI to create a *Signed URI.* The user stores the signed resource representation on her Pod at that URI.

Upon dereferencing a URI, the app validates the resource in three steps: First, the app checks if all triples quoted by the Linked Data Signature are in fact asserted. Second, the app checks if the signature value is verifiable using the quoted triples and the specified algorithms. Here, the signature suffix is removed prior to resolving relative URIs. Third, the app checks if the signature value matches with the signature suffix from the URI.

The user can not only verify the content from its Linked Data Signature but also expect it specifically to be served from its Signed URI. Moreover, the user is able to reference a resource in a specific state from another resource representation, thereby creating a Web of self-verifying resource representations.

4 Modeling Linked Data Signatures using RDF-star

Linked Data Signatures (LDS) are a way of modelling a cryptographic signature of an RDF graph or dataset. LDS are listed as one possible signature scheme in the Verifiable Credentials (VC) data model [10]. However, the VC data model is heavily influenced by the JSON syntax. Examining the data model from a pure RDF perspective, we took issue (1) in the use of RDF datasets, which have underspecified semantics [11], and (2) in the assertion of claimed statements, which are a result of the use of Named Graphs [4]. To avoid these issues, we model LDS using RDF-star. For an example of our data model, we recommend the inclined reader to take a look at our website which is linked on the first page. RDF-star allows for quoting triples, i. e. referencing without asserting. We argue that the LDS only provides meta-information on the signed triples and does not necessarily have to assert those triples. This may be useful, for instance, when the truth-value of the signed triples can change over time: a digital student id card becomes invalid once the student graduates. The signature, however, is always valid as it makes no statement about the truth-value of the signed triples.

5 Pondering the Implications of Signed URIs

Inspired by Trusty URIs [8, 7], a *Signed URI* is a URI that includes the hexadecimal value of the content's signature as a suffix, delimited by two underscores, e. g. *http://ex.org/test__0x12*. Specifically, the suffix ends the hierarchical part of a URI that does not have a query part. In a Web of resource representations with Linked Data Signatures available at Signed URIs, links among representations contribute the integrity of the resources' representations. When a resource's state changes, the signature value changes, the existing Signed URI mismatches, and the modification becomes detectable. For an undetectable modification, a new Signed URI is to be used and all resources that transitively reference the changed resource have to be updated with respective new Sigend URIs. Since on the Web, resources are typically under control of many different users, all those users have to agree on such updates. Otherwise, *some* evidence will remain.

6 Conclusion

In this demo, we showcased a Web of self-verifying resource representations using Linked Data Signatures and Signed URIs. In a Web of such resource representation, modifications to a resource's state become detectable unless all other resource representations that transitively reference the modified state are also changed. We envision a Web where such resources are under decentralised control of many different users, thereby providing a basis for trust in tamper-evident information.

Acknowledgements

This work is partially supported by the German federal ministry of education and research (BMBF) in MANDAT (FKZ 16DTM107B).

References

1. Allen, C.: The Path to Self-Sovereign Identity, `http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html`. (2016)
2. Braun, C., and Käfer, T.: Verifying the Integrity of Hyperlinked Information Using Linked Data and Smart Contracts. In: Proceedings of the 15th International Conference on Semantic Systems (SEMANTiCS) (2019)
3. Capadisli, S., Berners-Lee, T., Verborgh, R., and Kjernsmo, K.: Solid Protocol. Version 0.9.0, `https://solidproject.org/TR/protocol`. W3C Solid Community Group (2021)
4. Carroll, J.J., Bizer, C., Hayes, P.J., and Stickler, P.: Named graphs. J. Web Semant. 3(4) (2005)
5. Hartig, O., Champin, P.-A., Kellogg, G., and Seaborne, A.: RDF-star and SPARQL-star. W3C Draft Community Group Report, `https://w3c.github.io/rdf-star/cg-spec/editors_draft.html`. W3C (2022)

6. Hogan, A.: Canonical Forms for Isomorphic and Equivalent RDF Graphs: Algorithms for Leaning and Labelling Blank Nodes. ACM Trans. Web 11(4) (2017)
7. Kuhn, T., Chichester, C., Krauthammer, M., Queralt-Rosinach, N., Verborgh, R., Giannakopoulos, G., Ngomo, A.N., Viglianti, R., and Dumontier, M.: Decentralized provenance-aware publishing with nanopublications. PeerJ Comput. Sci. 2 (2016)
8. Kuhn, T., and Dumontier, M.: Trusty URIs: Verifiable, Immutable, and Permanent Digital Artifacts for Linked Data. In: Proceedings of the 11th European Semantic Web Conference (ESWC) (2014)
9. Mansour, E., Sambra, A.V., Hawke, S., Zereba, M., Capadisli, S., Ghanem, A., Aboulnaga, A., and Berners-Lee, T.: A Demonstration of the Solid Platform for Social Web Applications. In: Proceesings of Posters & Demos at the 25th International Conference on World Wide Web (WWW) (2016)
10. Sporny, M., Noble, G., Longley, D., Burnett, D.C., Zundel, B., and Den Hartog, K.: Verifiable Credentials Data Model. W3C Recommendation, `https://www.w3.org/TR/vc-data-model/`. W3C (2021)
11. Zimmermann, A.: RDF 1.1: On Semantics of RDF Datasets. W3C Working Group Note, `https://www.w3.org/TR/rdf11-datasets/`. W3C (2014)