Resource Integrity Proofs

cryptographic linking provides discoverability, integrity, and scheme agility

a white paper from Rebooting the Web of Trust VII

by Ganesh Annan and Kim Hamilton Duffy
with Manu Sporny, Dave Longley, David Lehn, and Bohdan Andriyiv

ABSTRACT

Currently, the Web provides a simple yet powerful mechanism for the dissemination of information via links. Unfortunately, there is no generalized mechanism that enables verifying that a fetched resource has been delivered without unexpected manipulation. Would it be possible to create an extensible and multipurpose cryptographic link that provides discoverability, integrity, and scheme agility?

This paper proposes a linking solution that decouples integrity information from link and resource syntaxes, enabling verification of any representation of a resource from any type of link. We call this approach Resource Integrity Proofs (RIPs). RIPs provide a succinct way to link to resources with cryptographically verifiable content integrity. RIPs can be combined with blockchain technology to create discoverable proofs of existence to off-chain resources.

civic
Protocol Labs
sovrin
EXODUS™
htc
interlinked
connecting intelligence
PTBVENTURES
TIERION
Microsoft

Sponsors for the Rebooting the Web of Trust VII Design Workshop

INTRODUCTION

Cryptographic linking solutions today have yet to provide a generalized mechanism for creating tamper-evident links. The Subresource Integrity standard limits this guarantee to script and link resources loaded on Web pages via the use of HTML attributes. IPFS provides a verification mechanism that is constrained to hash-based, content-addressable links, with no ability to complete content negotiation. RFC6920 proposes another mechanism that cannot be applied to existing links: it recommends the use of named information hashes and a resolution method that creates a content addressable URL [1]. Resource Integrity Proofs incorporates ideas from these standards and solutions to provide a new data format for cryptographic links that is fit for the open world.

This paper describes use cases benefitting from RIPs, such as enabling Verifiable Displays and meeting regulatory compliance.

FEATURES

Integrity

Resource Integrity Proofs use the representation of a resource as the input to a cryptographic hash function to generate a digest value. We can reproduce the digest value because the RIP data model requires recording of the content type and digest algorithm. Third parties can easily verify data received by 1) dereferencing the URL of the desired resource and 2) using the digest algorithm on the data to generate a matching digest value, ensuring that the data was not unexpectedly manipulated. The content received is now tamper-evident. RIPs may be placed on blockchains to simultaneously enable discoverability of off-chain resources and establish a proof of existence.

Discoverability

Resource Integrity Proofs allow any party to find a given resource. This is achieved simply by including the URL of the resource in the data model. RIPs may be placed on blockchains to enable discovery of data whilst keeping sensitive data off-chain (i.e., private and secure).

Scheme Agility

Resource Integrity Proofs make no assumptions on the URL scheme used. This scheme agility means that one can enable verification of the integrity of a resource using any URL scheme with any content type.

DATA MODEL

The Resource Integrity Proof (RIP) is a data model built using the Linked Data Proofs specification. It can be represented using many different syntaxes; examples are given here in JSON-LD, N-Quads, and in a simple table.

- id: The location of the resource.
- proof: The Linked Data digital proof.
  - type: The identifier for the digital proof suite.

- contentType: The content type for the resource.
- multiDigest: The multibase encoded multihash [2].

JSON-LD Syntax

```
{
  "@context": "https://w3id.org/security/v2",
  "id": "https://example.com/storage/ndBRHU8gqjRzkcRdrPC2XQ",
  "proof": {
    "type": "Multihash2018",
    "contentType": "application/json",
    "multiDigest": "zQmUvZSaVzgjVHCDDDAoNNBgpiAkN6wKmCcD37vvnmoKq6e"
  }
}
```

N-Quads Syntax

```
<https://example.com/storage/ndBRHU8gqjRzkcRdrPC2XQ>
<https://w3id.org/security#proof> _:b0 .
_:b0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<https://w3id.org/security#Multihash2018> .
_:b0 <http://schema.org/contentType> "application/json" .
_:b0 <https://w3id.org/security#multiDigest>
"zQmUvZSaVzgjVHCDDDAoNNBgpiAkN6wKmCcD37vvnmoKq6e" .
```

Table

Subject	Predicate	Object	Graph
https://example.com/storage/ndBRHU8gqjRzkcRdrPC2XQ	https://w3id.org/security#proof	_:b0	
_:b1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	https://w3id.org/security#Multihash2018	_:b0
_:b1	http://schema.org/contentType	application/pdf	_:b0
_:b1	https://w3id.org/security#multiDigest	zQmUvZSaVzgjVHCDDDAoNNBgpiAkN6wKmCcD37vvnmoKq6e	_:b0

USE CASES

There are many compelling applications of RIPs in a decentralized ecosystem. We will first dive into the problem of Verifiable Displays, which seeks to ensure that the rendering of the Verifiable Credential content matches what the issuer intended. Next, we will envision a new age regulatory compliance system built on top of Decentralized Identifiers (DID), Verifiable Credentials (VC), and Object Capabilities (OCAP).

Verifiable Displays

In the Educational/Occupational Credentials space, RIPs allow issuers to specify a set of approved visual renderings associated with a signed claim. This enables any viewer of the claim to determine if the visual rendering differs from what was intended by the issuer -- an ability that's critical for detecting social engineering attacks introduced by tampering with the rendered image. The "verifiability" of a Verifiable Credential applies to the content of the claim, not necessarily the human-readable display. As described in Verifiable Displays, this risk has been generally been addressed in an ad-hoc, use-case-dependent way. But there is no clear standard or convention for tamper detection across different credential schemas and use cases.

This example shows how we might use a RIP to address the problem of proving that a PNG file hashes to the value expected by a referencing Verifiable Credential:

```
{
  "@context": ["https://w3id.org/credentials/v1", "https://w3id.org/security/v2"],
  "id": "http://credentials.example.org/credentials/3732",
  "type": ["VerifiableCredential", "EmployeeOfTheMonthCredential"],
  "issuer": "did:example:12345678",
  "issuanceDate": "2014-01-02",
  "expirationDate": "2014-02-02",
  "claim": {
    "id": "did:example:10011872",
    "accomplishment": "Employee of the Month Demonstrating Excellent Leadership 
Skills"
  },
  "verifiableDisplay": {
    "id": "https://raw.githubusercontent.com/WebOfTrustInfo/rwot7/master/draft-
documents/images/exampleVerifiableDisplay.png",
    "proof": {
      "type": ["ResourceIntegrityProof", "Multihash2018"],
      "contentType: "image/png",
      "multiDigest": "zQmUvZSaVzgjVHCDDDAoNNBgpiAkN6wKmCcD37vvnmoKq6e"
    }
  },
  "proof": { ... }
}
```

In this example, we leverage the **`ResourceIntegrityProof`** and **`Multihash2018`** type to say that the image identified by **`id`** is expected to have a multibase encoded multihash that matches the value in **`multiDigest`**.

Extensions to General Linked Data

Expanding on linked visual data examples, this method could enable a pharmacist to ensure the prescription they are viewing matches the associated machine-readable content. If the credential contained sensitive data, we wouldn't want the image to be publicly-hosted. But this is also supported: **`id`** can be any URI, so the referenced visual rendering could be stored offline.

RIPs enable snapshot integrity proofs for general linked data; this can be used for credentials bridging legacy systems where data is stored in a mutable store.

Meeting Regulatory Compliance

Organizations must provide documentation to regulators in order to maintain compliance.

We can implement software to meet the full requirements of this use case by adding RIPs to the already composable Lego-like ecosystem of interoperable decentralized technologies such as DIDs, VCs, and OCAPs -- and by combining this ecosystem with a cryptographically auditable system, such as a blockchain.

When an organization is preparing supporting documentation to meet compliance, they can post one or more RIPs and an OCAP for accessing each resource to a blockchain. This OCAP only grants access to the regulator and only to the specific items and for the duration that they need. Posting the RIP to a blockchain enables discoverability of the resource and establishes a proof of existence. The tamper-evident characteristics of the blockchain prove that the data existed at some point in the past, establishing trust via the cryptosystem rather than requiring it in the organization. The regulator then uses the delegated OCAP to dereference the url in the RIP and to ensure the data was not changed since the time of submission.

APPENDIX

[1] Example from Naming Things with Hashes (RFC6920)

Using an Authority of **`example.com`** and the sha-256 hash of the text **`"Hello World!"`** we can generate the following **`ni`** URI:

```
ni://example.com/sha-256;f4OxZX_x_FO5LcGBSKHWXfwtSx-j1ncoSt3SABJtkGk
```

The generated `ni` URI takes advantage of the `.well-known` URI (RFC5785) format so that we can dereference the information using HTTP(S):

```
http://example.com/.well-known/ni/sha-256/f4OxZX_x_FO5LcGBSKHWXfwtSx-
j1ncoSt3SABJtkGk
```

[2] Generating a Multibase encoded Multihash

An example using Node.js v8.12.0

```
const multibase = require("multibase"); // v0.5.0
const multihash = require("multihashes"); // v0.4.14
const crypto = require("crypto");

const hash = crypto.createHash('sha256').update('Hello World!', 'utf8').digest();
// hash is 7f83b1657ff1fc53b92dc18148a1d65dfc2d4b1fa3d677284addd200126d9069

const mh = multihash.encode(hash, 'sha2-256');
// mh is 12207f83b1657ff1fc53b92dc18148a1d65dfc2d4b1fa3d677284addd200126d9069

const mb = multibase.encode('base58btc', mh);
// mb is zQmWvQxTqbG2Z9HPJgG57jjwR154cKhbtJenbyYTWkjgF3e
```

ADDITIONAL CREDITS

Lead Authors: Ganesh Annan and Kim Hamilton Duffy

Contributors: Manu Sporny, Dave Longley, David Lehn, and Bohdan Andriyiv

About Rebooting the Web of Trust

This paper was produced as part of the Rebooting the Web of Trust VII design workshop. On September 26th through 28th, 2018, over 40 tech visionaries came together in Mississauga, Ontario to talk about the future of decentralized trust on the internet with the goal of writing 3-5 white papers and specs. This is one of them.

Leadership Team: Christopher Allen, Joe Andrieu, Kim Hamilton Duffy, Manu Sporny, and Heather Vescent

Gold Sponsors: Civic, Protocol Labs, Sovrin, an anonymous donor

Silver Sponsors: HTC, Microsoft, PTB Ventures, Spherity, Tierion

Community Sponsors: Blockchain Commons, Learning Machine, Legendary Requirements, Purple Tornado, Veres One

Workshop Credits: Christopher Allen (Founder), Joe Andrieu (Producer and Facilitator), Shannon Appelcline (Editor-in-chief), and Claire Rumore (Graphical Recorder)

Thanks to our other contributors and sponsors!

What's Next?

The design workshop and this paper are just starting points for Rebooting the Web of Trust. If you have any comments, thoughts, or expansions on this paper, please post them to our GitHub issues page:

https://github.com/WebOfTrustInfo/rwot7/issues

The next Rebooting the Web of Trust design workshop is scheduled for the week of February 27th to March 1st, , 2019. If you'd like to be involved or would like to help sponsor the event, email:

rwot-leadership@googlegroups.com