Leveraging DNS in Digital Trust: Credential Exchanges and Trust Registries

Oct 31, 2023

Presented By
Jesse Carter – CIRA

cira

Workgroup: Network Working Group
Internet-Draft: draft-latour-dns-and-digital-trust-latest
Published: 4 October 2023
Intended Status: Informational
Expires: 6 April 2024
Authors: J. Carter CIRA
J. Latour CIRA
M. Glaude NorthernBlock

Leveraging DNS in Digital Trust: Credential Exchanges and Trust Registries

Abstract

This memo describes an architecture for digital credential verification and validation using Decentralized Identifiers (DIDs), distributed ledgers, trust registries, and the DNS. This architecture provides a verifier with a simple process by which to cryptographically verify the credential they are being presented with, verify and resolve the issuer of that credential to a domain, and verify that issuer's membership in a trust registry.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at

Table of Contents

1. Introduction
   1.1. Note
2. Conventions and Definitions
3. Terminology
4. Mapping a DID to the DNS
   4.1. URI record scoping
   4.2. Issuer Handles
5. DID Public Keys in the DNS
   5.1. TLSA Record Scoping, Selector Field
   5.2. Issuer Handles
   5.3. Instances of Multiple Key Pairs
   5.4. Benefits of Public Keys in the DNS
6. Digital Credential Verification using DIDs and the DNS
7. Role of DNSSEC for Assurance and Revocation
8. The Role of Trust Registries in Bidirectional Credential Verification
   8.1. Issuer's Membership Claim in a Trust Registry
      8.1.1. URI Record Name Scoping
   8.2. Trust Registry Membership Proof

This is accomplished in 3 steps:

1. **A substantiated and verifiable association between a DID and a Domain via the DNS**
   - DID claims an association via the alsoKnownAs/Services fields to a domain name.
   - The domain substantiates that claim using URI and TLSA records in its DNS zone.
   - That information is discoverable on publicly available and widely supported infrastructure (the DNS).

2. **The ability to verify Verifiable Credential Proofs using information in the DNS**
   - Using the URI and TLSA records created above, a verifier has the option to use the public key associated with a TLSA record in the DNS to verify a credential proof without having to interface with a distributed ledger or unfamiliar/unsupported DID method.

3. **The ability for an entity to claim membership in a Trust Registry via the DNS, and the Trust Registry to support that claim via the DNS**
   - An entity can present a claim that they are a member of a trust registry using a URI record.
   - The trust registry can substantiate that claim by hosting the associated public keys of that entity in their DNS zone using appropriately named TLSA records.

Why is this helpful?

1. **Portability**
   - DIDs can remain rooted in decentralized ledgers or ledger-based implementations while making their PKI and associations available over easily accessible public infrastructure through DNS.

2. **Authenticity**
   - If a DID claims to be associated with a domain, and the domain is also claiming to be associated with the DID, that's a pretty strong indicator the claim is true.
   - With DNSSEC, DNS records can be confidently obtained from any resolver with cryptographic assurance.

3. **Flexibility and Support**
   - Supporting existing or new implementations with long standing and well governed infrastructure facilitates adoption and integration for users.

How does it work?

Importance of DNSSEC

- **DNSSEC brings a layer of cryptographic verifiability to the answers resulting from a DNS query via the form of a proof/signature in an RRSIG.**

- **This provides assurance that however and with whoever you resolved your query, you can be verify that the results are accurately representing what was placed there by the zone owner.**

```
jesse@CIRA-20220055:~$ dig ciralabs.ca +dnssec

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> ciralabs.ca +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57122
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4000
;; QUESTION SECTION:
;ciralabs.ca.                   IN      A

;; ANSWER SECTION:
ciralabs.ca.            3600    IN      A       3.209.122.207
ciralabs.ca.            3600    IN      RRSIG   A 13 2 3600 20231109000000 20231019000000 58624 cirala
bs.ca. ExNo0m9xakiJkS/YqkEixpVjTw6ejiKSm8W74bkutMZzM9CfRz1w8SMK 2FmtYu+NQ2w/pzyy0cMi4i5LFBZWLA==

;; Query time: 20 msec
;; SERVER: 10.4.91.26#53(10.4.91.26) (UDP)
;; WHEN: Mon Oct 30 17:54:45 EDT 2023
;; MSG SIZE  rcvd: 163
```

HOW DO WE LINK A DID TO A DOMAIN?

DID to DNS name mapping

§ 5.1.3 Also Known As

A DID subject can have multiple identifiers for different purposes, or at different times. The assertion that two or more DIDs (or other types of URI) refer to the same DID subject can be made using the `alsoKnownAs` property.

alsoKnownAs

The `alsoKnownAs` property is *OPTIONAL*. If present, the value *MUST* be a set where each item in the set is a URI conforming to [RFC3986].

This relationship is a statement that the subject of this identifier is also identified by one or more other identifiers.

https://www.w3.org/TR/did-core/#also-known-as

- As specified in the W3C DID Core spec v1.0, the “alsoKnownAs” field can contain a URI, making the assertion that the URI in the field refers to the same DID subject.

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2018/v1",
    "https://w3id.org/security/suites/x25519-2019/v1"
  ],
  "id": "did:example:1234abc",
  "alsoKnownAs": ["ciralabs.ca"],
  "verificationMethod": [
    {
      "type": "Ed25519VerificationKey2018",
      "id": "did:example:1234abcM#key-1",
      "controller": "did:example:1234abc",
      "publicKeyBase58": "HdXo5kegxgPze3tAw6QYU7vvJg4gbxztqSidt8LsB6eS"
    }
  ],
  "authentication": [
    "did:example:1234abc#key-1"
  ],
  "assertionMethod": [
    "did:example:1234abc#key-1"
  ],
  "keyAgreement": [
    "did:example:1234abc#key-1"
  ]
}
```

HOW DO WE LINK A DID TO A DOMAIN?

DID to DNS name mapping

§ 5.4 Services

Services are used in DID documents to express ways of communicating with the DID subject or associated entities. A service can be any type of service the DID subject wants to advertise, including decentralized identity management services for further discovery, authentication, authorization, or interaction.

https://www.w3.org/TR/did-core/#services

- As specified in the W3C DID Core spec v1.0, the “services” property can be leveraged to indicate a way to communicate with the DID subject or an associated entity for a variety of reasons.

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2018/v1",
    "https://w3id.org/security/suites/x25519-2019/v1"
  ],
  "id": "did:example:1234abc",
  "service": [{
    "id": "did:example:1234abc",
    "type": "LinkedDomains",
    "serviceEndpoint": "https://ciralabs.ca"
  }],
  "verificationMethod": [
    {
      "type": "Ed25519VerificationKey2018",
      "id": "did:example:1234abcM#key-1",
      "controller": "did:example:1234abc",
      "publicKeyBase58": "HdXo5kegxgPze3tAw6QYU7vvJg4gbxztqSidt8LsB6eS"
    }
  ],
  "authentication": [
    "did:example:1234abc#key-1"
  ],
  "assertionMethod": [
    "did:example:1234abc#key-1"
  ],
  "keyAgreement": [
    "did:example:1234abc#key-1"
  ]
}
```

HOW DO WE LINK A DID TO A DOMAIN?

But there's a problem.

- Malicious actors can create DIDs and associate them with any domains they want using the previous methods.
- Ex: Someone gets a fake or invalid credential and sees the Issuer is claiming an association with a real entity or organization when they resolve the DID doc.
    - Does this devalue the authority and credibility of the actual Organization?
    - What about the credential itself?
    - How would someone verify this claim?
- What if the evidence to substantiate the claim lived in the actual subject of the claim itself?
    - The DNS zone of the domain. Ex: **Ontario.ca**, **nist.gov**, **uscis.gov, driverslicenses.org, etc.**

HOW DO WE LINK A DOMAIN TO A DID?

How do we link the domain name of an Entity and their DID in DNS?

- A **URI record** allows us to associate a domain name (like **ciralabs.ca**) to a **URI**, like **did:example:1234abc**.
- Through this association, a domain can show which DIDs are under their jurisdiction, in a publicly accessible and resolvable manner.
- Ex: URI Record
  - **_did.ciralabs.ca 1 0 “did:example:1234abc”**
- But this association with URI records only substantiates a claim. We turn to cryptography and the help of TLSA records to provide verification.

```
$ dig _did.ciralabs.ca URI +dnssec
```

HOW DO WE LINK A DOMAIN TO A DID?

TLSA records provide the cryptographic glue holding the association between a domain and a DID together.

- **A TLSA record allows us to host cryptographic information in the DNS.**
  - TLSA records will hold the public half of a **verificationMethod** used by the DID.
  - Ex: TLSA Records
    - **_did.ciralabs.ca 3 1 0 F716B82BD54…(public key un-hashed)**
    - **_did.ciralabs.ca 3 1 1 8A9E530067D…(sha256 public key hash)**
  - By hosting the public keys in the DNS, the Verifier can not only verify any signature against the verificationMethod in the DID document, but also against the associated TLSA record via DNS as well.
  - In a similar manor, a cryptographic challenged can be done against both the domain and the DID to ensure they both have access to the associated private key.

```
$ dig _did.ciralabs.ca TLSA +dnssec
```

How is this beneficial?

1. **Portability. Strengthened association between DIDs and Entities/Orgs domains while keeping architectures and implementations separate.**

2. **Authenticity. Acts as a form of 2FA/MFA for the DID.**

3. **Flexibility and Support. Provides increased accessibility and interoperability with existing and well supported infrastructure.**

WWW.CIRA.CA

What about Verifiable Credentials?

cira

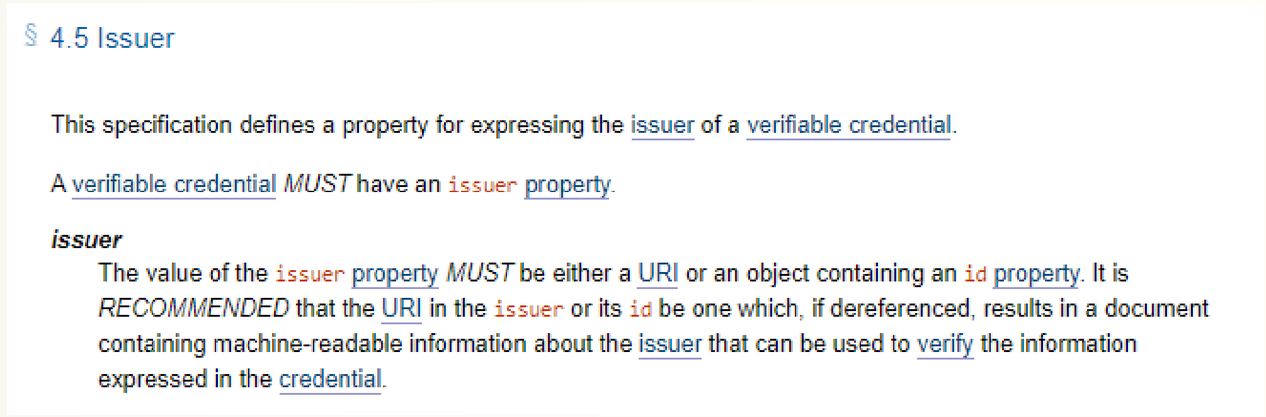
What happens when you're presented with a credential and can't resolve the Issuer or verificationMethod?

- There's hundreds of DID methods (and counting)
  - That's a lot of varying infrastructure to interoperate with and support.
- Issues with a "Universal Resolver"
  - Centralization
  - How do you know the results are accurate?
- Accessibility
  - What is the technical barrier to participation in this ecosystem?

What if you had the !!***OPTION***!! to get the required verification material via DNS?

HOW DO WE LINK A VERIFIABLE CREDENTIAL TO A DOMAIN?

From the Verifiable Credential:



§ 4.5 Issuer

This specification defines a property for expressing the issuer of a verifiable credential.

A verifiable credential *MUST* have an `issuer` property.

issuer

The value of the `issuer` property *MUST* be either a URI or an object containing an `id` property. It is *RECOMMENDED* that the URI in the `issuer` or its `id` be one which, if dereferenced, results in a document containing machine-readable information about the issuer that can be used to verify the information expressed in the credential.

https://www.w3.org/TR/vc-data-model/#issuer

- As specified in the W3C Verifiable Credential Data Model spec v2.0, the “issuer” field can either be a URI or an object containing a URI.
- This means we can include associations like a domain name to this property.
- That leaves the “verificationMethod” to point to the DID.

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.w3.org/ns/credentials/examples/v2",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ],
  "id": "https://ciralabs.ca/credentials/3732",
  "type": [
    "VerifiableCredential",
    "ExampleDegreeCredential"
  ],
  "issuer": "https://ciralabs.ca",
  "validFrom": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "degree": {
      "type": "ExampleBachelorDegree",
      "name": "Bachelor of Science and Arts"
    }
  },
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2023-10-21T19:54:59Z",
    "verificationMethod": "did:example:1234abc#key1",
    "proofPurpose": "assertionMethod",
    "proofValue": "z4WpCbBwMvPLqEHe9PCtbc3BAuKBa3mAcaX3rD9o7t2ca..."
  }
}
```

VERIFYING CREDENTIALS USING DNS

A verifier can verify a credential proof using the information stored in the same URI and TLSA records from earlier.

- **The credential indicates the domain to query against and the verificationMethod used to generate the proof.**
  - Ex: {“issuer”: “https://**ciralabs.ca**”}
  - Ex: { “verificationMethod”: “**did:example:1234abc**#key1”}
- **The verifier queries the domain for a URI record containing the DID corresponding with the verificationMethod used in the proof.**
  - Ex: _did.**ciralabs.ca** 1 0 “**did:example:1234abc**”
- **Once they have confirmed the correct DID is also being represented in the zone, they can use the key in the TLSA record to verify the credential signature.**
  - This proof: {“proofValue”: "z4WpCbBwMvPLqEHe9PCtbc3BAuKBa3mAcaX3rD9o7t2ca..."}
  - Can be verified using this public key:
    - _did.**ciralabs.ca** 3 1 0 F716B82BD54…(public key un-hashed)

How is this beneficial?

1. **Portability. DNS is supported everywhere. All DID methods/distributed ledgers are not. You can verify a credential proof from any DID method without having to directly support it.**

2. **Authenticity. Thanks to DNSSEC, you can be sure the records you are being shown are as intended by the zone owner.**

3. **Choice for implementers. The verifier determines the level of assurance they need, in some cases this may be adequate, in cases where it isn't this doesn't prevent the verifier from diving deeper.**

WWW.CIRA.CA

The Trust Registry Part…

SUPPORTING TRUST REGISTRIES WITH DNS

What we just outlined is useful for determining the authenticity of DIDs and verifying credentials, but can we also determine an Issuer's membership in a Trust Registry the same way?

By leveraging the same URI and TLSA records we used before to verify the credential, we can also verify the Issuer's membership in a Trust Registry.

- **ciralabs.ca hosts a URI record, pointing to a trust registry it is claiming membership in.**
  - Ex: URI Record
    - **_tr.ciralabs.ca 1 0 "trustregistry.ca"**
  - This record indicates that ciralabs.ca is claiming it is registered under trustregistry.ca, a Trust Registry in Canada.

```
$ dig _tr.ciralabs.ca URI +dnssec
```

SUPPORTING TRUST REGISTRIES WITH DNS

A verifier can now reference that the public key/s used by ciralabs.ca are hosted in trustregistry.ca's zone.

The verifier can check trustregistry.ca for proof of ciralabs.ca's membership. We use TLSA records to provide a layer of cryptographic verifiability to that association by hosting the public key of ciralabs.ca in trustregistry.ca's zone.

- **Ex: TLSA Record**
  - ciralabs.ca._tr.trustregistry.ca 3 1 0 (public key un-hashed)
  - ciralabs.ca._tr.trustregistry.ca 3 1 1 (sha256 hash of public key)
- **Now the verifier has the option to verify the signature of anything signed by ciralabs.ca's key, as well as perform cryptographic challenges against the domain or its associated DIDs as outlined previously.**

```
$ dig ciralabs.ca._tr.trustregistry.ca TLSA +dnssec
```

How is this beneficial?

1. **Discoverability. It's possible to explore Trust Registries and their associations by traversing the series of DNS pointers through URI and TLSA records.**

2. **Authenticity. DNSSEC provides confidence in your DNS queries. DNS queries substantiate claims made from a DID document about Trust Registry membership.**

3. **Choice for implementers. The verifier determines the level of assurance they need, in some cases this may be adequate, in cases where it isn't this doesn't prevent the verifier from diving deeper.**

Thank You

https://www.cira.ca/labs

cira

Northern Block

Digital Identity LABORATORY