DANCE - CIRA

Leveraging DNSSEC in Digital Identity

IETF Jul 29th 2022

Presented By
Jacques Latour and Jesse Carter
Problem Statement:
We’re missing DNS/DNSSEC support for finding, identifying and authenticating “Digital Identity Trust Registries”
There’s a story here, so here it goes, where to start?

Disclaimer:
Using our iotregistry.ca as an example of an issuer
Using an IoT Device as example of a wallet
Using a ASP as an example of a verifier
May use incorrect terminology – not an expert yet ;)
Trying to explain the use case of trust registries
Some references to SSI/ToIP/Decentralized ID/W3C DID
CERT=certificate, not CERT RRTYPE ;)
And you should know PKI and DNSSEC better than me
WHERE’S THE BEGINNING?

Let’s start with verifiable credentials

• The IoT device acts as a wallet that holds verifiable credentials
• A verifiable credential in this example is based on a signed Digital Certificate
• The issuer issued the verifiable credentials to the IoT Device
• Signed Digital Certificate Details
  • SAN: 1 or more FQDN unique identifier
  • i.e. SAN: uuid._device.iotregistry.ca
  • Signed by the issuer ← this is the important part
  • i.e. SAN: iotregistry.ca

• At least one SAN can be used to verify the credential status and authenticity (TLSA or NXdomain)
• The (self or not) Signed Certificate is used to establish the identity of the IoT device and facilitates the connection to the ASP (TLS, eventually using dane_clientid, DANCE proposal work here! 😃)
THE ROLE OF THE ISSUER

We need TLSA records to track the issuers public keys and TLSA records of issued verifiable credentials

- These TLSA records for the root and sub cert can be used to verify the authenticity of the issuer
  - TLSA record for iotregistry.ca subCertificate (0 0 0) – public key
  - TLSA record for iotregistry.ca rootCertificate (0 0 0) – public key

- The verifiable credentials TLSA records can be used to verify their authenticity and status (NXDOMAIN = revoked)
  - TLSA record for the uuid._device.iotregistry.ca (3 0 1) – hash of public key

DANCE: no new stuff, this works so far, right?
THE ROLE OF THE VERIFIER

To verify the authenticity ‘technical trust’ of the verifiable credentials

• Against the digital certificate chain
  • i.e. SAN: uuid._device.iotregistry.ca
  • Signed by the iotregistry.ca SUB Certificate
  • (but they need to find those root/sub certificate in trust store)

• Over DNS/DNSSEC with TLSA
  • TLSA record for uuid._device.iotregistry.ca (3 0 1) (certificate hash)
  • TLSA record for iotregistry.ca subCertificate with public key ( 0 0 0 )
  • TLSA record for iotregistry.ca rootCertificate with public key ( 0 0 0 )
Please hang on
I’m going somewhere here!
(I hope 😊)
I SIMPLIFIED IT, LARGELY BASED ON SSI/TOIP/DECENTRALIZED ID

In the, let’s call it “Digital Identity World”, here’s how I see it!

This world is about Trust: do I trust the issuer, do I trust the verifier, do I trust the wallet holder, do I trust the Trust Registry, do I trust that Digital Identity ecosystem
But how do you know the **issuer** of verifiable credentials is (wait for the famous word!) **trusted**
IN THE UP-SIDE-DOWN WORLD, THIS IS WHERE DNSSEC IS IMPORTANT

Human/Legal Trust in the **front**, Technical Trust in the **back**

How do you find a trust registry?
The Role of Trust Registries

To register issuers, verifiers and other trust registries

- Trust Registry have a governance model and framework that defines the characteristic of a registration (what the registrant can and can’t do, in a DID...)

- But we see a need for a trust registry to prove via TLSA that an issuer or verifier or another trust registry is part of its ecosystem using:
  - `<_trustregistry>` label

- And a need for an issuer, a verifier or another trust registry to prove their trust registry affiliation using:
  - `<TR>` RRTYPE (urg, another ;)

WWW.CIRA.CA
ISSUER REGISTRATION IN A TRUST REGISTRY

Let’s look at the iotregistry.ca issuer registration in that context

- The issuer iotregistry.ca should have one or more TR (trust registry affiliation) records to point to the trust registries they belong to:
  
  iotregistry.ca TLSA ( 0 0 0 ) rootCertificate
  iotregistry.ca TR trustregistry.ca

- Trust registry trustregistry.ca should publish TLSA (0 0 1) _trustregistry record for the issuers that matches it’s TLSA/Certificate public key

  trustregistry.ca TLSA ( 0 0 0 ) rootCertificate
  iotregistry.ca._trustregistry.trustregistry.ca TLSA (0 0 1)

  - TLSA of iotregistry.ca root CERT (like a DS record)
  - This record with would provide with authenticity the affiliation to a trust registry (but nothing about the human trust itself!!)

Yes “_trustregistry” can be shortened ;)
PROPOSING THIS AS A GLOBAL CHAIN OF TRUST FOR TRUST REGISTRIES

From a verifiable credential, you can find the issuer and associated trust registries

- The issuer iotregistry.ca should have one or more TR trust registry affiliation record to point to the trust registries they belong to
  - iotregistry.ca TLSA (0 0 0) rootCertificate
  - iotregistry.ca TR trustregistry.ca

- Canada’s Trust Registry (is affiliated with the IANA trust registry)
  - trustregistry.ca TLSA (0 0 0) rootCertificate
  - trustregistry.ca TR trustregistry.iana.org
  - iotregistry.ca._trustregistry.trustregistry.ca TLSA (0 0 1) hash of rootCertificate

- IANA Trust Registry: (Would have an entry for Canada’s trust registry)
  - trustregistry.iana.org TLSA (0 0 0) rootCertificate
  - trustregistry.ca._trustregistry.trustregistry.iana.org TLSA (0 0 1) rootCertificate

- ROOT ZONE: (global trust registry anchor)
  - Root zone: trustregistry.iana.org._trustregistry. TLSA (0 0 1) rootCertificate
I went too far, didn’t I? 😊
LEVERAGING DNSSEC IN DIGITAL IDENTITY

There’s a good story here, hope it makes sense now ;)

- Something along the line presented here would provide a method and system for finding, identifying and authenticating “Digital Identity Trust Registries”
- Anchoring the Digital Identity world into the existing IANA ROOT ZONE DNSSEC trust anchor makes sense
- Allows for unique identifiers
- Allows for a single global trust anchor (for those who wish to use)
- Allows for real time verifiable credentials management real time using DNS (non existence = revoked)
- Alex worked on https://tools.ietf.org/id/draft-mayrhofer-did-dns-03.html to link DID to DNS
- Check our DANCE GitHub repository

Question?

https://github.com/CIRALabs/dance
Thank You

https://www.cira.ca/labs