High Assurance DIDs with DNS

Proof of Concept Results
Discussion and Next Steps

Problem and Solution

Problem:

Identifiers today are EITHER:

EASY to recognize, but HARD to verify

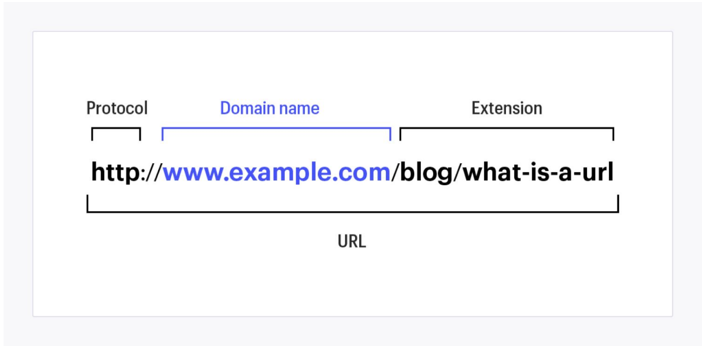
OR:

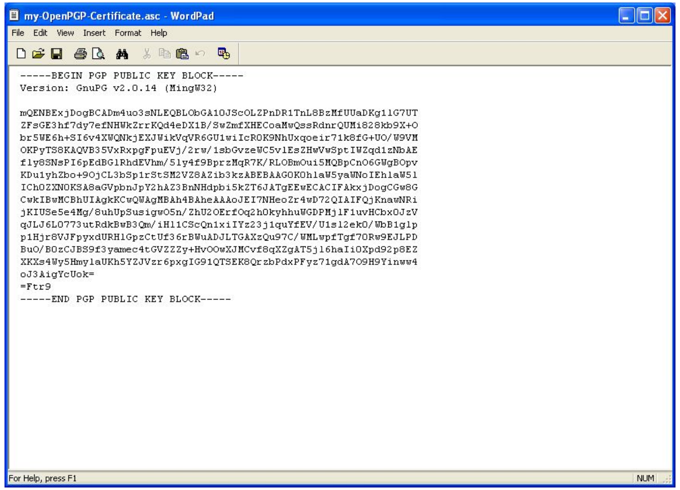
EASY to verify, but HARD to recognize

There is no formal guidance to publish an identifier that is easy to recognize AND easy to verify.

Solution:

Develop formal guidance to publish an identifier that is easy to recognize AND easy to verify (with high assurance)




my-OpenPGP-Certificate.asc - WordPad
File Edit View Insert Format Help
For Help, press F1
NUM

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.14 (MingW32)

mQENBExjDogBCADm4uo3sNLEQBLObGA1OJScOLZPnDR1TnL8BzMfUUaDKg11G7UT
ZFsGE3hf7dy7efNHWkZrrKQd4eDX1B/SwZmfXHECoaMwQssRdnrQUMi828kb9X+O
br5WE6h+SI6v4XWQNkjEXJWikVqVR6GU1wiIcROK9NhUxqoeir71k8fG+UO/W9VM
OKPyTS8KAQVB35VxRxpgFpuEVj/2rw/1sbGvzeWC5vlEsZHwVwSptIWZqd1zNbAE
fly8SNsPI6pEdBGlRhdEVhm/5ly4f9BprzMqR7K/RLOBmOui5MQBpCnO6GWgBOpv
KDu1yhZbo+9OjCL3bSp1rStSM2VZ8AZib3kzABEBAAG0KOhlaW5yaWNoIEhlaW5l
IChOZXNOKSA8aGVpbnJpY2hAZ3BnNHdpbi5kZT6JATgEEwECACIFAkxjDogCGw8G
CwkIBwMCBhUIAgkKCwQWAgMBAh4BAheAAAoJEI7NHeoZr4wD72QIAIFQjKnawNRi
jKIUSe5e4Mg/8uhUpSusigwO5n/ZhU2OErfOq2hOkyhhuWGDPMjlF1uvHCbxOJzV
qJLJ6LO773utRdkBwB3Qm/iHl1CScQn1xiIYz23j1quYfEV/U1sl2ekO/WbB1glp
p1Hjr8VJFpyxdURHlGpzCtUf36rBWuADJLTGAXzQu97C/WMLwpfTgf7ORw9EJLPD
BuO/BOzCJBS9f3yamec4tGVZZZy+HvOOwXJMCvf8qXZgAT5jl6haIiOXpd92p8EZ
XKXs4Wy5HmylaUKh5YZJVzr6pxgIG91QTSEK8QrzbPdxPFyz71gdA7O9H9Yinww4
oJ3AigYcUok=
=Ftr9
-----END PGP PUBLIC KEY BLOCK-----
```

What is a High-Assurance DID with DNS?

Human Trust

Human-Readable Identifier

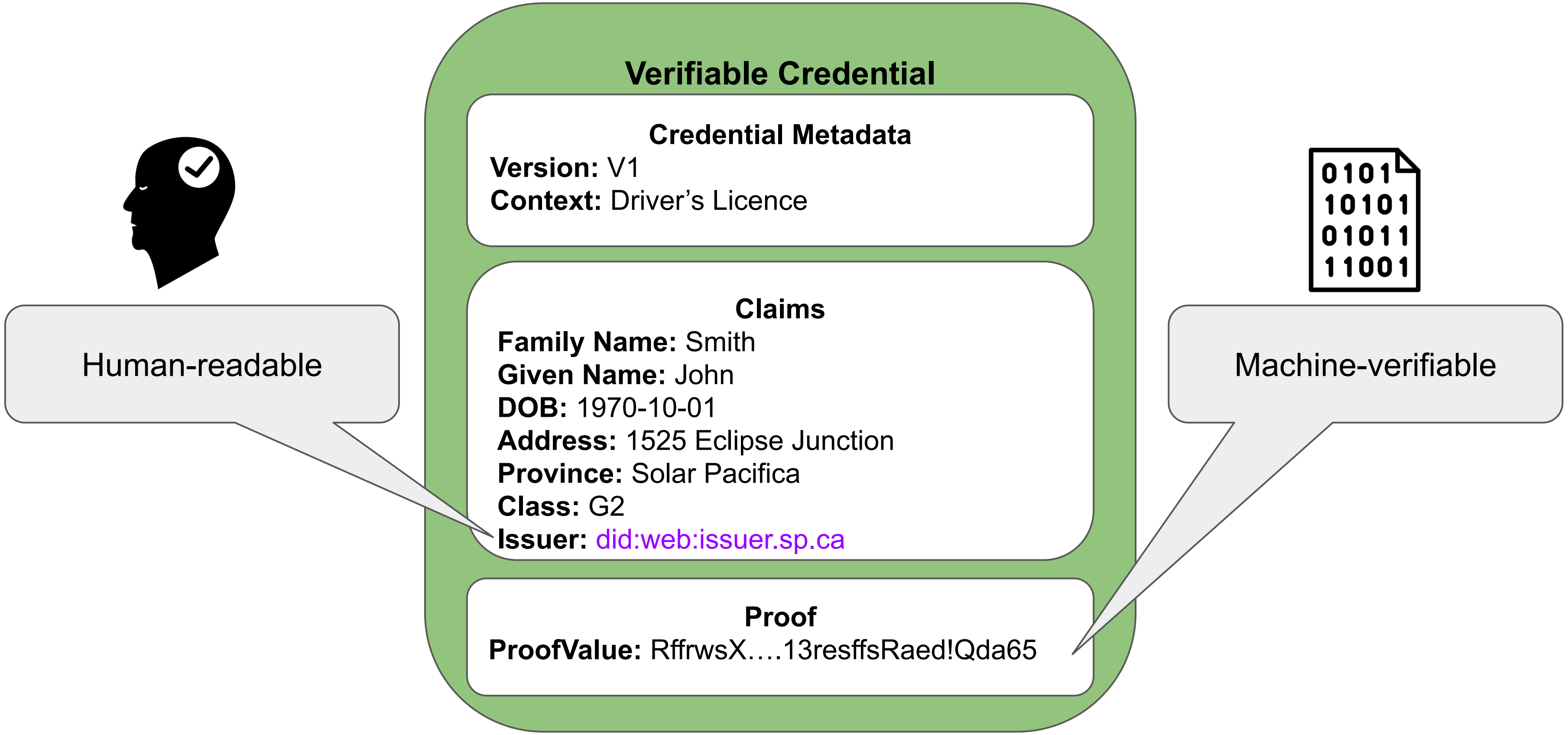
did:web:issuer.trustregistry.ca

Machine-Verifiable Assertions

- Digital Certificates and Signatures
- DID Document DataIntegrityProof
- DNS/DNSSEC Validation
  - DANE TLS Check
  - DID URI Record Match
  - TLSA Self-Issued Certificate Record Match
- Member of Trust Registry

Technical Trust

Verifiable Credential using a High-Assurance DID



Verifiable Credential

Credential Metadata
Version: V1
Context: Driver's Licence

Claims
Family Name: Smith
Given Name: John
DOB: 1970-10-01
Address: 1525 Eclipse Junction
Province: Solar Pacifica
Class: G2
Issuer: did:web:issuer.sp.ca

Proof
ProofValue: RffrwsX….13resffsRaed!Qda65

Successful Proof of Concept

Proved that the **did:web** method can be made **high assurance**.

- Leverage DNSSEC for **cryptographically-assured binding** of identifier (i.e.,domain name) to certificates (X.509 TLS)
- Enable a **digitally-signed** DID document

Repo:

https://github.com/CIRALabs/high-assurance-dids-with-dns

Implementations:

https://trustregistry.ca, https://trustroot.ca, trustregistry.nborbit.ca, godiddy.com

Draft RFC:

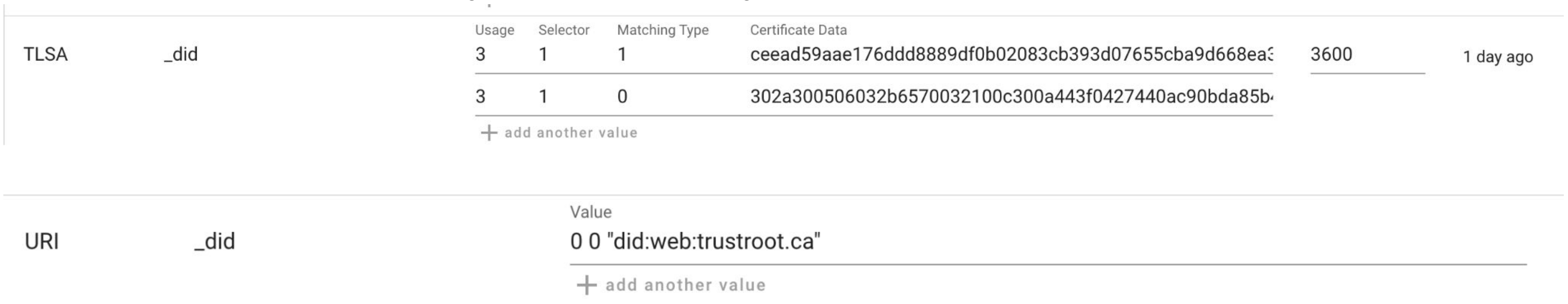
https://www.ietf.org/archive/id/draft-carter-high-assurance-dids-with-dns-03.html

Key Findings and Conclusions

- Proof of Concept (POC) proves high-assurance did:web is possible
    - No new standards, specifications, methods, or technologies are required; only well-documented operating procedures.
    - Can be easily incorporated into existing infrastructure (little to no investment is required)
    - Cryptographic assurances enables the did:web method to be used for high-assurance use cases
        - Government-issued credentials, signature verification, etc.
- High-assurance did:web concept is ready to pilot
    - Key security mechanisms have been proven to work
    - A high-assurance did:web pilot project can now be considered with the knowledge that the security mechanisms are sound.
    - Approach is independent of CA/B, EU/QWAC, Mobile and Proprietary Platforms

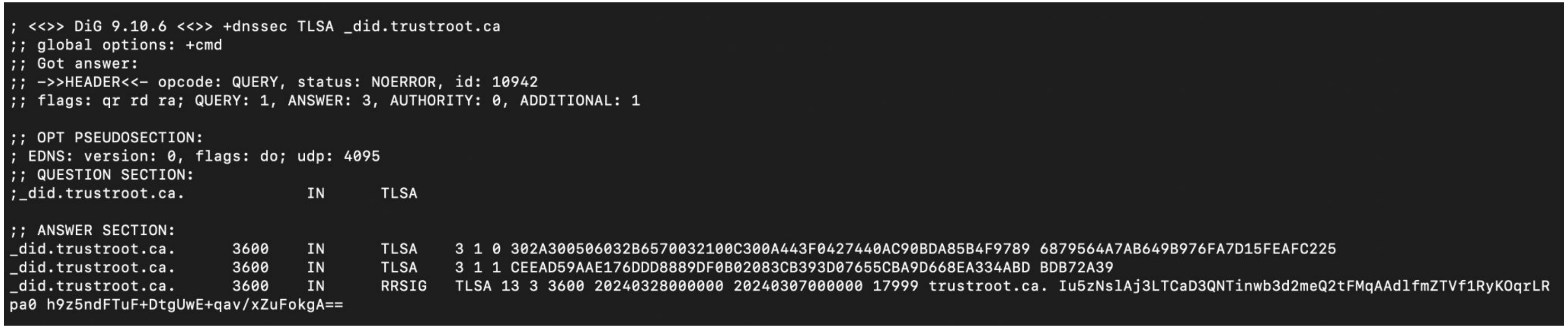
Cryptographic Validation using DNS/DNSSEC

DNS/DNSSEC controlled by domain authority



		Usage	Selector	Matching Type	Certificate Data		
TLSA	_did	3	1	1	ceead59aae176ddd8889df0b02083cb393d07655cba9d668ea3	3600	1 day ago
		3	1	0	302a300506032b6570032100c300a443f0427440ac90bda85b		

		Value
URI	_did	0 0 "did:web:trustroot.ca"



```
; <<>> DiG 9.10.6 <<>> +dnssec TLSA _did.trustroot.ca
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10942
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4095
;; QUESTION SECTION:
;_did.trustroot.ca.             IN      TLSA

;; ANSWER SECTION:
_did.trustroot.ca.      3600    IN      TLSA    3 1 0 302A300506032B6570032100C300A443F0427440AC90BDA85B4F9789 6879564A7AB649B976FA7D15FEAFC225
_did.trustroot.ca.      3600    IN      TLSA    3 1 1 CEEAD59AAE176DDD8889DF0B02083CB393D07655CBA9D668EA334ABD BDB72A39
_did.trustroot.ca.      3600    IN      RRSIG   TLSA 13 3 3600 20240328000000 20240307000000 17999 trustroot.ca. Iu5zNslAj3LTCaD3QNTinwb3d2meQ2tFMqAAdlfmZTVf1RyKOqrLR
pa0 h9z5ndFTuF+DtgUwE+qav/xZuFokgA==
```

dig command above shows RRSIG records for DNSSEC

Methods/Controls Used

No.	TLS Website (https://trustregistry.ca)	High Assurance DNS (did:web:trustregistry.ca)	Verification Method
1	W1. Trusted Certificate Authority		Browser Validation
2	W2. Domain Name Association		Browser Validation
3	W3. Certificate Validity (expiry, revocation)	Already In Place	Browser Validation
4	W4. Public Key (signing, encryption verification)		Browser Validation
5	W5. Browser Root Store Check		Browser Validation
6		H0. DANE Check	Cryptographic Validation
7	W7. Website Resource Control	H1. DID Resource Control	Policy Verification
8	W8. Website Page Origin	H2. DID Document Management	Policy Verification
9		H3. DID Document Data Integrity (1)	Cryptographic Validation
10		H4. DID Document Key Control	Policy Verification
11	POC	H5. DID Document Key Generation	Policy Verification
12		H6. DID Domain Name Control (DNSSEC)	Policy Verification
13		H7. Domain Name Association (DNSSEC)	Cryptographic Validation
14		H8. Domain Name Signing (DNSSEC)	Cryptographic Validation
15		H9. Domain Name Key Control (DNSSEC)	Policy Verification
16		H10. Domain Name Key Generation (DNSSEC)	Policy Verification
17		H11. Hardware Security Module	Policy Verification

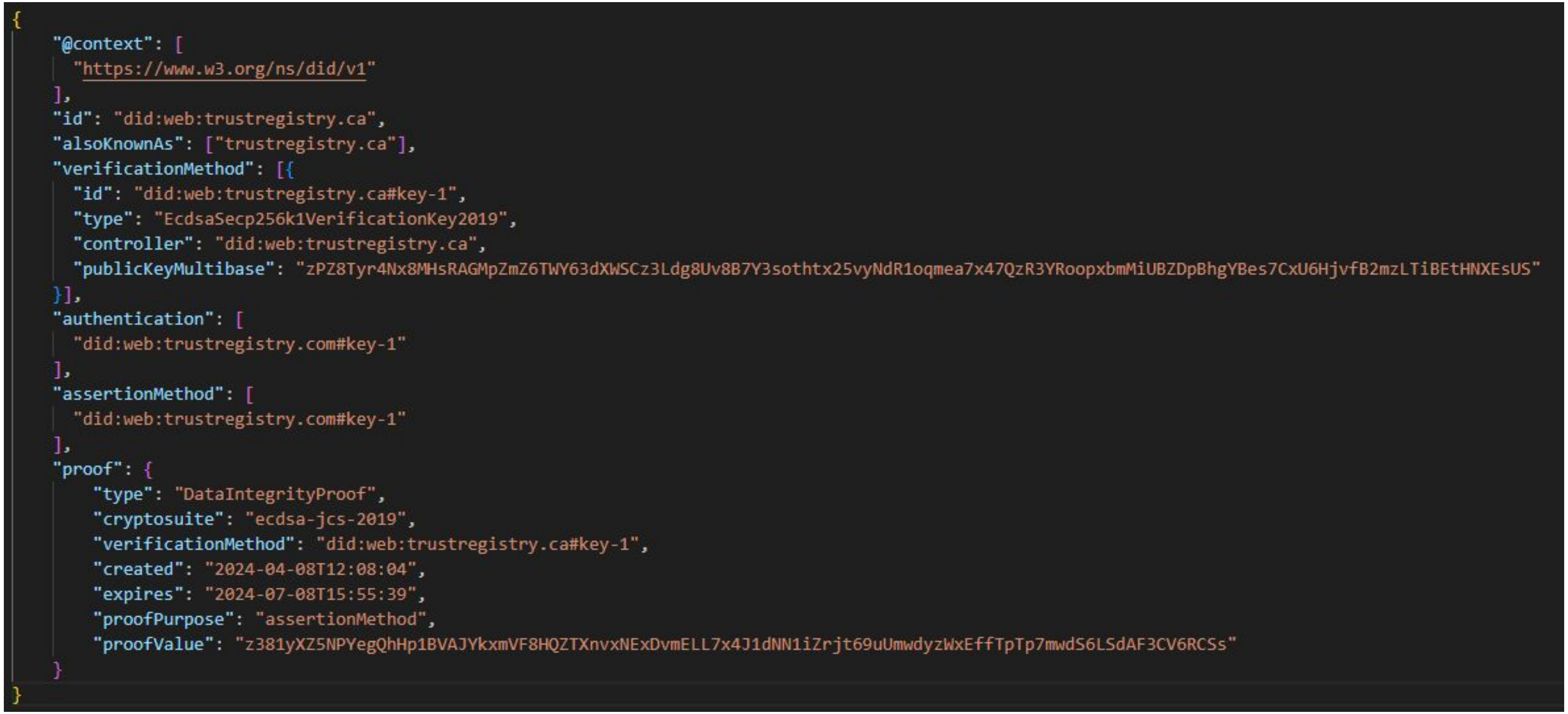
Browser Validation: Validated by browser implementation
Policy Verification: Verified by self-attestation, third party, or assessment body.
Cryptographic Validation: Validated by cryptographic algorithms

(1) Includes certificate validity

Demonstration of Cryptographic Validation Methods

1. DID Verification Methods
2. DID document proof (DataIntegrityProof)
3. DNSSEC Signed TLSA and URI Records
4. Validation of 1-3

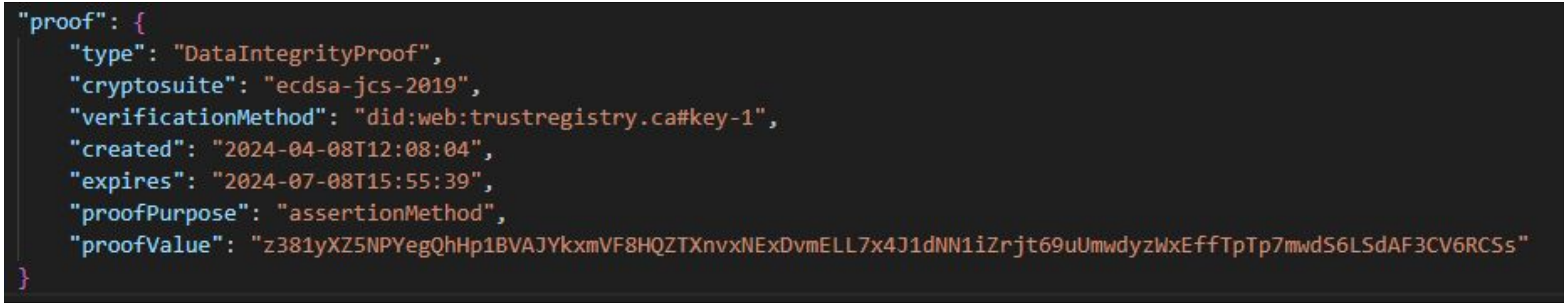
DID Doc



```
{
    "@context": [
      "https://www.w3.org/ns/did/v1"
    ],
    "id": "did:web:trustregistry.ca",
    "alsoKnownAs": ["trustregistry.ca"],
    "verificationMethod": [{
      "id": "did:web:trustregistry.ca#key-1",
      "type": "EcdsaSecp256k1VerificationKey2019",
      "controller": "did:web:trustregistry.ca",
      "publicKeyMultibase": "zPZ8Tyr4Nx8MHsRAGMpZmZ6TWY63dXWSCz3Ldg8Uv8B7Y3sothtx25vyNdR1oqmea7x47QzR3YRoopxbmMiUBZDpBhgYBes7CxU6HjvfB2mzLTiBEtHNXEsUS"
    }],
    "authentication": [
      "did:web:trustregistry.com#key-1"
    ],
    "assertionMethod": [
      "did:web:trustregistry.com#key-1"
    ],
    "proof": {
        "type": "DataIntegrityProof",
        "cryptosuite": "ecdsa-jcs-2019",
        "verificationMethod": "did:web:trustregistry.ca#key-1",
        "created": "2024-04-08T12:08:04",
        "expires": "2024-07-08T15:55:39",
        "proofPurpose": "assertionMethod",
        "proofValue": "z381yXZ5NPYegQhHp1BVAJYkxmVF8HQZTXnvxNExDvmELL7x4J1dNN1iZrjt69uUmwdyzWxEffTpTp7mwdS6LSdAF3CV6RCSs"
    }
}
```

DID Doc Data Integrity Proof

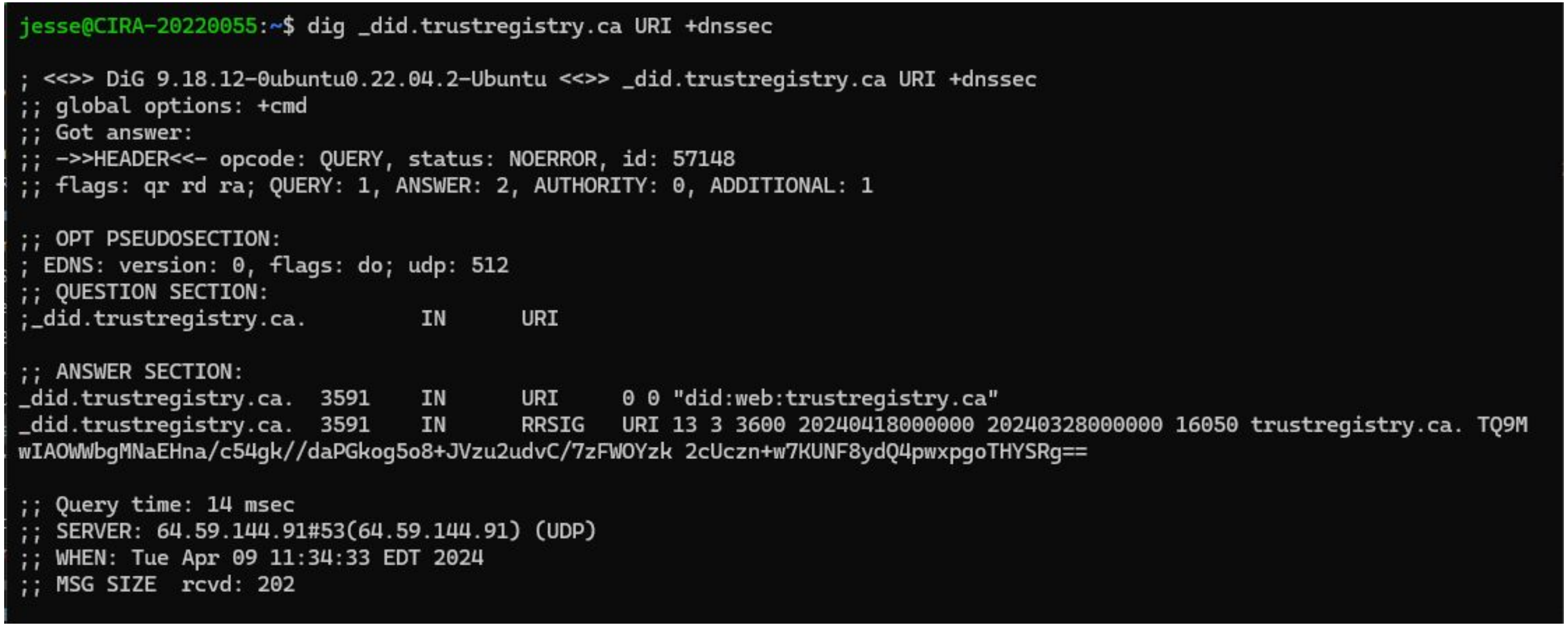
DataIntegrityProof signed by private key of #key-1 generated from previous step



```
"proof": {
    "type": "DataIntegrityProof",
    "cryptosuite": "ecdsa-jcs-2019",
    "verificationMethod": "did:web:trustregistry.ca#key-1",
    "created": "2024-04-08T12:08:04",
    "expires": "2024-07-08T15:55:39",
    "proofPurpose": "assertionMethod",
    "proofValue": "z381yXZ5NPYegQhHp1BVAJYkxmVF8HQZTXnvxNExDvmELL7x4J1dNN1iZrjt69uUmwdyzWxEffTpTp7mwdS6LSdAF3CV6RCSs"
}
```

URI Record Matching

DNS/DNSSEC records need to be added by domain owner

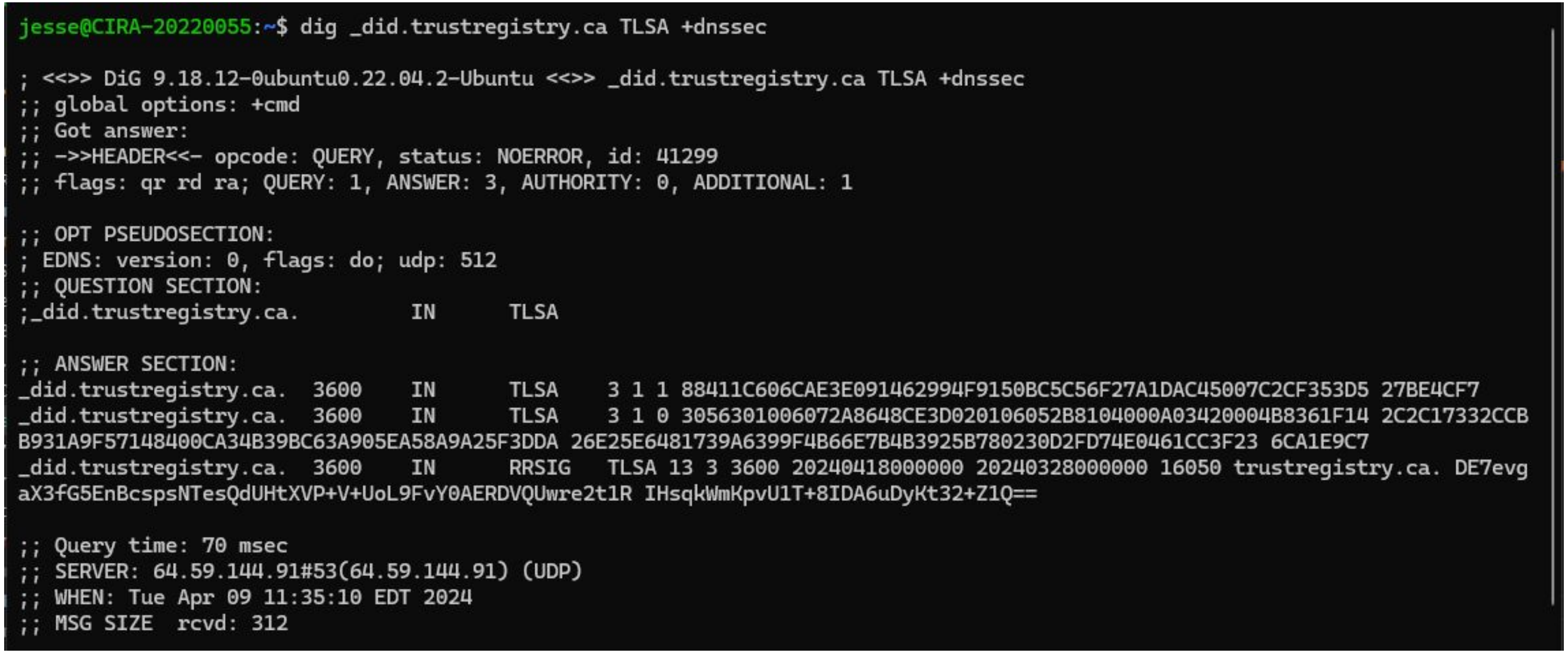


```
jesse@CIRA-20220055:~$ dig _did.trustregistry.ca URI +dnssec

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> _did.trustregistry.ca URI +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57148
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;_did.trustregistry.ca.         IN      URI

;; ANSWER SECTION:
_did.trustregistry.ca.  3591    IN      URI     0 0 "did:web:trustregistry.ca"
_did.trustregistry.ca.  3591    IN      RRSIG   URI 13 3 3600 20240418000000 20240328000000 16050 trustregistry.ca. TQ9M
wIAOWWbgMNaEHna/c54gk//daPGkog5o8+JVzu2udvC/7zFWOYzk 2cUczn+w7KUNF8ydQ4pwxpgoTHYSRg==

;; Query time: 14 msec
;; SERVER: 64.59.144.91#53(64.59.144.91) (UDP)
;; WHEN: Tue Apr 09 11:34:33 EDT 2024
;; MSG SIZE  rcvd: 202
```

dig command above shows URI and RRSIG records for DNSSEC

TLSA Record Matching

DNS/DNSSEC records need to be added by domain owner



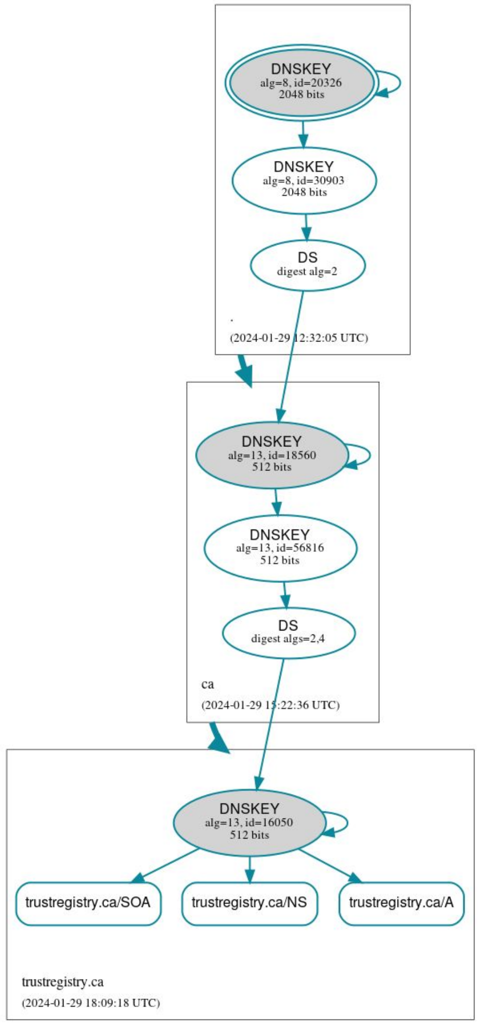
```
jesse@CIRA-20220055:~$ dig _did.trustregistry.ca TLSA +dnssec

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> _did.trustregistry.ca TLSA +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41299
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;_did.trustregistry.ca.         IN      TLSA

;; ANSWER SECTION:
_did.trustregistry.ca.  3600    IN      TLSA    3 1 1 88411C606CAE3E091462994F9150BC5C56F27A1DAC45007C2CF353D5 27BE4CF7
_did.trustregistry.ca.  3600    IN      TLSA    3 1 0 3056301006072A8648CE3D020106052B8104000A03420004B8361F14 2C2C17332CCB
B931A9F57148400CA34B39BC63A905EA58A9A25F3DDA 26E25E6481739A6399F4B66E7B4B3925B780230D2FD74E0461CC3F23 6CA1E9C7
_did.trustregistry.ca.  3600    IN      RRSIG   TLSA 13 3 3600 20240418000000 20240328000000 16050 trustregistry.ca. DE7evg
aX3fG5EnBcspsNTesQdUHtXVP+V+UoL9FvY0AERDVQUwre2t1R IHsqkWmKpvU1T+8IDA6uDyKt32+Z1Q==

;; Query time: 70 msec
;; SERVER: 64.59.144.91#53(64.59.144.91) (UDP)
;; WHEN: Tue Apr 09 11:35:10 EDT 2024
;; MSG SIZE  rcvd: 312
```

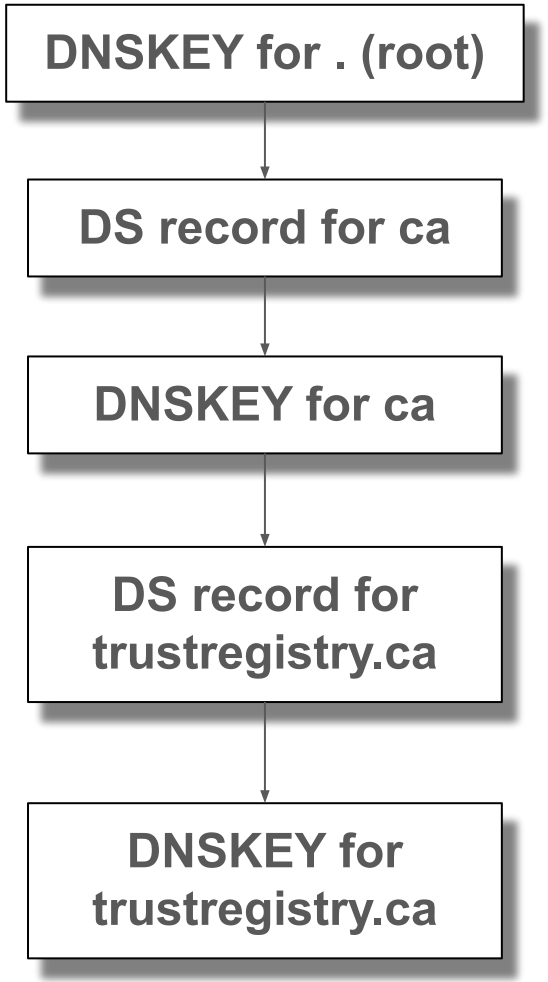
dig command above shows TLSA and RRSIG records for DNSSEC

Verifying DNSSEC Chain of Trust

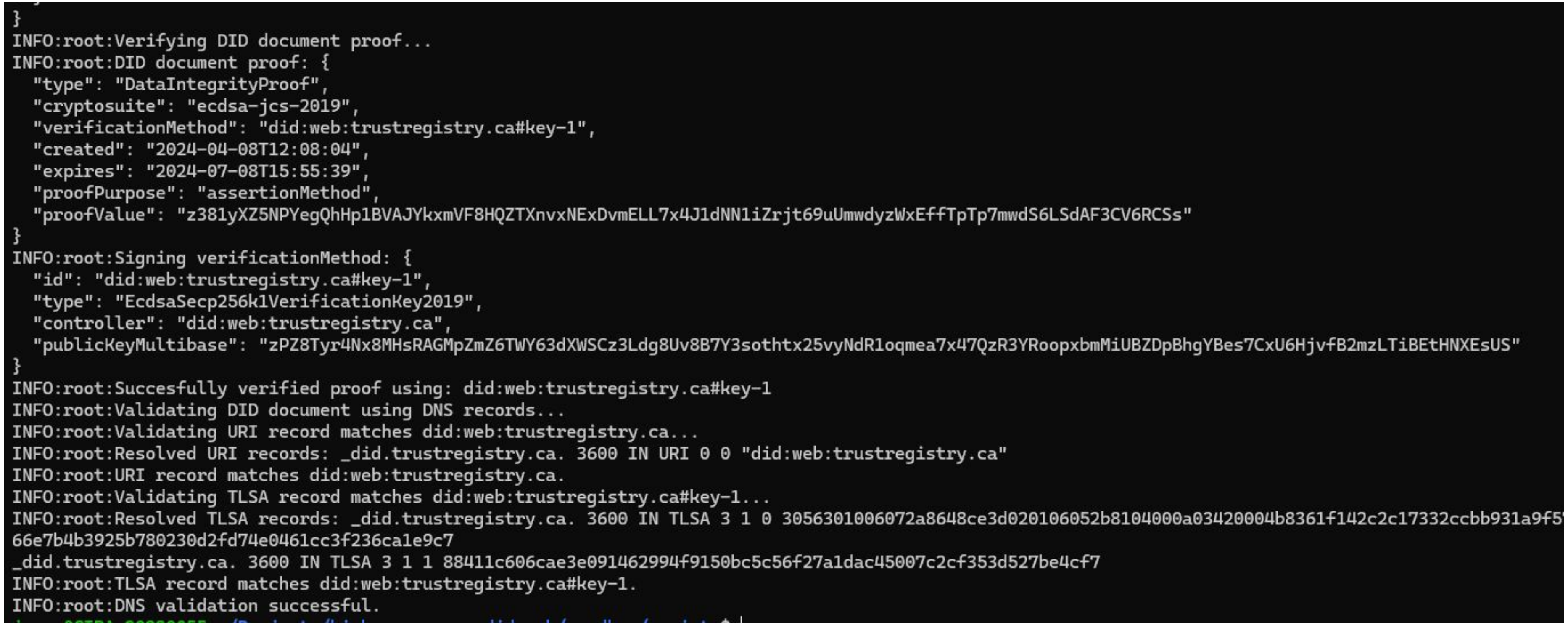
Publicly available tool for visual analysis of the DNSSEC authentication chain for a domain name and its resolution path in the DNS namespace

https://dnsviz.net/



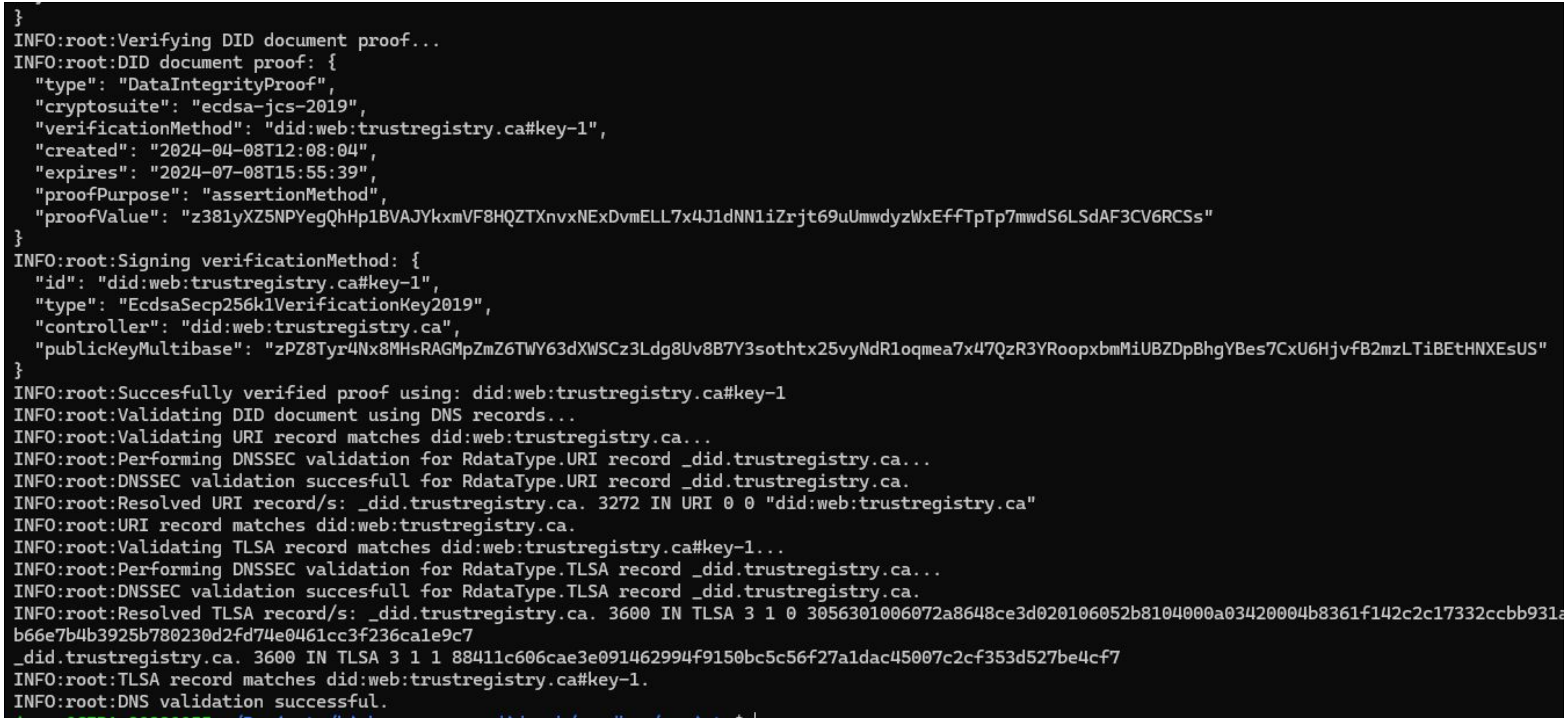


Example verification of did:web:trustregistry.ca using DNS



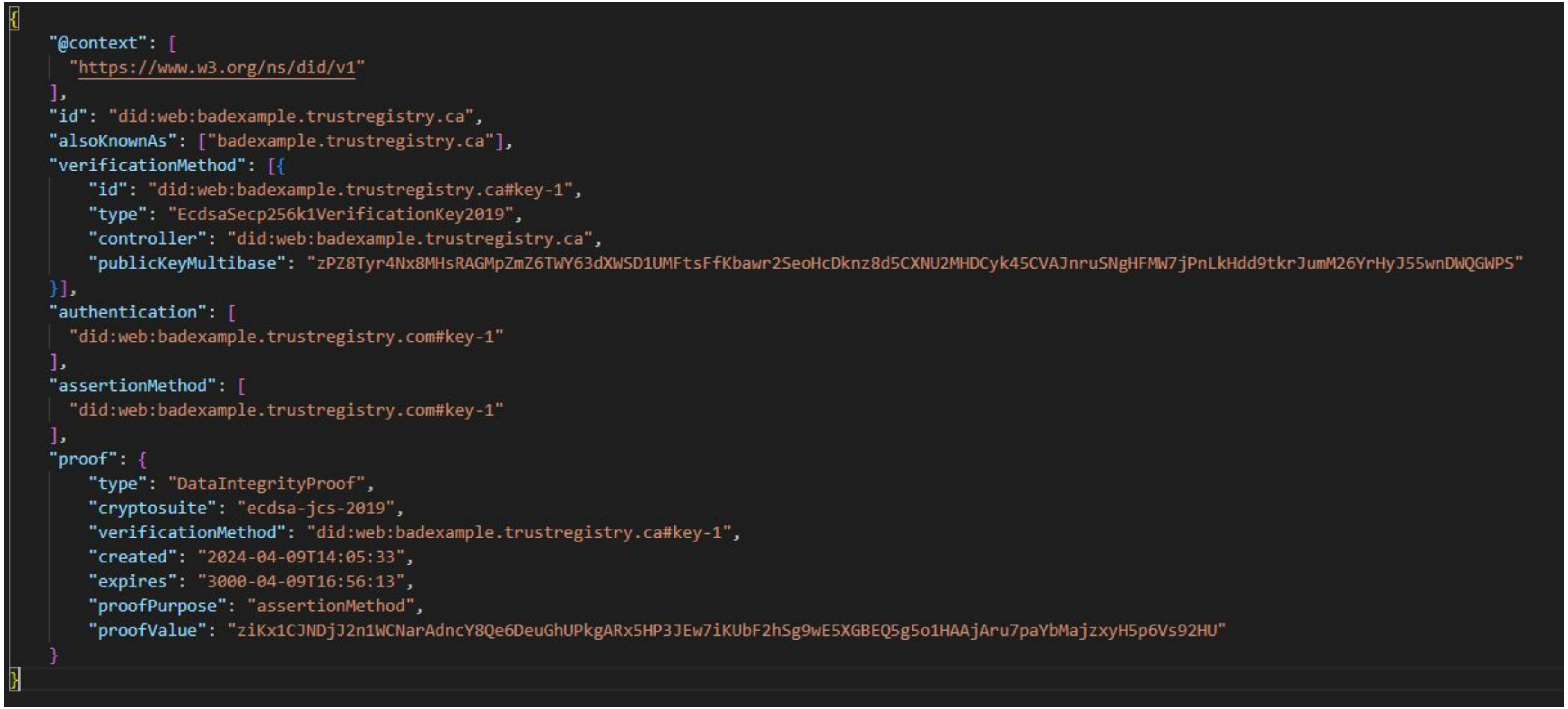
```
}
INFO:root:Verifying DID document proof...
INFO:root:DID document proof: {
  "type": "DataIntegrityProof",
  "cryptosuite": "ecdsa-jcs-2019",
  "verificationMethod": "did:web:trustregistry.ca#key-1",
  "created": "2024-04-08T12:08:04",
  "expires": "2024-07-08T15:55:39",
  "proofPurpose": "assertionMethod",
  "proofValue": "z381yXZ5NPYegQhHp1BVAJYkxmVF8HQZTXnvxNExDvmELL7x4J1dNN1iZrjt69uUmwdyzWxEffTpTp7mwdS6LSdAF3CV6RCSs"
}
INFO:root:Signing verificationMethod: {
  "id": "did:web:trustregistry.ca#key-1",
  "type": "EcdsaSecp256k1VerificationKey2019",
  "controller": "did:web:trustregistry.ca",
  "publicKeyMultibase": "zPZ8Tyr4Nx8MHsRAGMpZmZ6TWY63dXWSCz3Ldg8Uv8B7Y3sothtx25vyNdR1oqmea7x47QzR3YRoopxbmMiUBZDpBhgYBes7CxU6HjvfB2mzLTiBEtHNXEsUS"
}
INFO:root:Succesfully verified proof using: did:web:trustregistry.ca#key-1
INFO:root:Validating DID document using DNS records...
INFO:root:Validating URI record matches did:web:trustregistry.ca...
INFO:root:Resolved URI records: _did.trustregistry.ca. 3600 IN URI 0 0 "did:web:trustregistry.ca"
INFO:root:URI record matches did:web:trustregistry.ca.
INFO:root:Validating TLSA record matches did:web:trustregistry.ca#key-1...
INFO:root:Resolved TLSA records: _did.trustregistry.ca. 3600 IN TLSA 3 1 0 3056301006072a8648ce3d020106052b8104000a03420004b8361f142c2c17332ccbb931a9f5
66e7b4b3925b780230d2fd74e0461cc3f236ca1e9c7
_did.trustregistry.ca. 3600 IN TLSA 3 1 1 88411c606cae3e091462994f9150bc5c56f27a1dac45007c2cf353d527be4cf7
INFO:root:TLSA record matches did:web:trustregistry.ca#key-1.
INFO:root:DNS validation successful.
```

Example verification of did:web:trustregistry.ca using DNSSEC



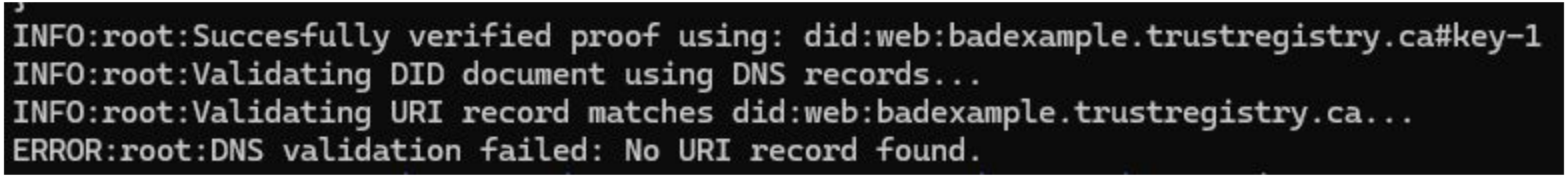
```
}
INFO:root:Verifying DID document proof...
INFO:root:DID document proof: {
  "type": "DataIntegrityProof",
  "cryptosuite": "ecdsa-jcs-2019",
  "verificationMethod": "did:web:trustregistry.ca#key-1",
  "created": "2024-04-08T12:08:04",
  "expires": "2024-07-08T15:55:39",
  "proofPurpose": "assertionMethod",
  "proofValue": "z381yXZ5NPYegQhHp1BVAJYkxmVF8HQZTXnvxNExDvmELL7x4J1dNN1iZrjt69uUmwdyzWxEffTpTp7mwdS6LSdAF3CV6RCSs"
}
INFO:root:Signing verificationMethod: {
  "id": "did:web:trustregistry.ca#key-1",
  "type": "EcdsaSecp256k1VerificationKey2019",
  "controller": "did:web:trustregistry.ca",
  "publicKeyMultibase": "zPZ8Tyr4Nx8MHsRAGMpZmZ6TWY63dXWSCz3Ldg8Uv8B7Y3sothtx25vyNdR1oqmea7x47QzR3YRoopxbmMiUBZDpBhgYBes7CxU6HjvfB2mzLTiBEtHNXEsUS"
}
INFO:root:Succesfully verified proof using: did:web:trustregistry.ca#key-1
INFO:root:Validating DID document using DNS records...
INFO:root:Validating URI record matches did:web:trustregistry.ca...
INFO:root:Performing DNSSEC validation for RdataType.URI record _did.trustregistry.ca...
INFO:root:DNSSEC validation succesfull for RdataType.URI record _did.trustregistry.ca.
INFO:root:Resolved URI record/s: _did.trustregistry.ca. 3272 IN URI 0 0 "did:web:trustregistry.ca"
INFO:root:URI record matches did:web:trustregistry.ca.
INFO:root:Validating TLSA record matches did:web:trustregistry.ca#key-1...
INFO:root:Performing DNSSEC validation for RdataType.TLSA record _did.trustregistry.ca...
INFO:root:DNSSEC validation succesfull for RdataType.TLSA record _did.trustregistry.ca.
INFO:root:Resolved TLSA record/s: _did.trustregistry.ca. 3600 IN TLSA 3 1 0 3056301006072a8648ce3d020106052b8104000a03420004b8361f142c2c17332ccbb931a
b66e7b4b3925b780230d2fd74e0461cc3f236ca1e9c7
_did.trustregistry.ca. 3600 IN TLSA 3 1 1 88411c606cae3e091462994f9150bc5c56f27a1dac45007c2cf353d527be4cf7
INFO:root:TLSA record matches did:web:trustregistry.ca#key-1.
INFO:root:DNS validation successful.
```

Verification Failure Example

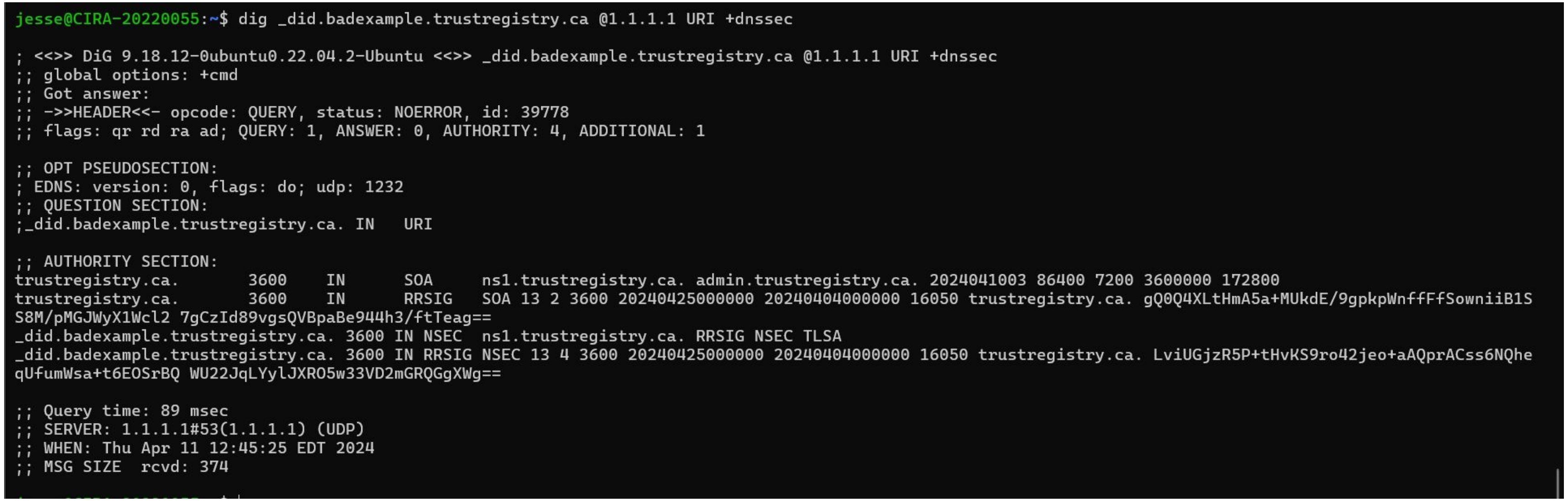


```
{
    "@context": [
      "https://www.w3.org/ns/did/v1"
    ],
    "id": "did:web:badexample.trustregistry.ca",
    "alsoKnownAs": ["badexample.trustregistry.ca"],
    "verificationMethod": [{
        "id": "did:web:badexample.trustregistry.ca#key-1",
        "type": "EcdsaSecp256k1VerificationKey2019",
        "controller": "did:web:badexample.trustregistry.ca",
        "publicKeyMultibase": "zPZ8Tyr4Nx8MHsRAGMpZmZ6TWY63dXWSD1UMFtsFfKbawr2SeoHcDknz8d5CXNU2MHDCyk45CVAJnruSNgHFMW7jPnLkHdd9tkrJumM26YrHyJ55wnDWQGWPS"
    }],
    "authentication": [
      "did:web:badexample.trustregistry.com#key-1"
    ],
    "assertionMethod": [
      "did:web:badexample.trustregistry.com#key-1"
    ],
    "proof": {
        "type": "DataIntegrityProof",
        "cryptosuite": "ecdsa-jcs-2019",
        "verificationMethod": "did:web:badexample.trustregistry.ca#key-1",
        "created": "2024-04-09T14:05:33",
        "expires": "3000-04-09T16:56:13",
        "proofPurpose": "assertionMethod",
        "proofValue": "ziKx1CJNDjJ2n1WCNarAdncY8Qe6DeuGhUPkgARx5HP3JEw7iKUbF2hSg9wE5XGBEQ5g5o1HAAjAru7paYbMajzxyH5p6Vs92HU"
    }
}
```

Failed verification of did:web:badexample.trustregistry.ca: No URI Record

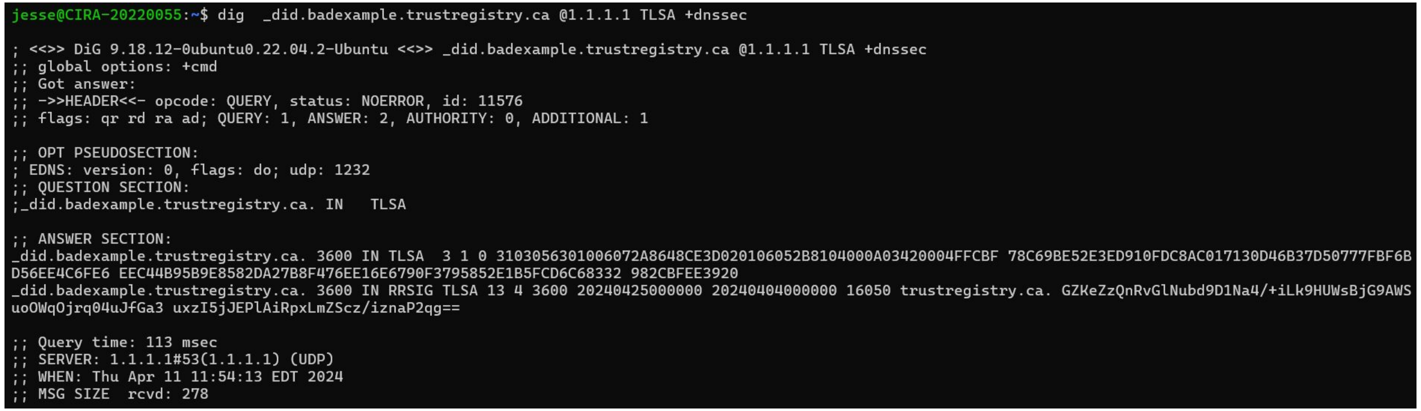


```
INFO:root:Succesfully verified proof using: did:web:badexample.trustregistry.ca#key-1
INFO:root:Validating DID document using DNS records...
INFO:root:Validating URI record matches did:web:badexample.trustregistry.ca...
ERROR:root:DNS validation failed: No URI record found.
```

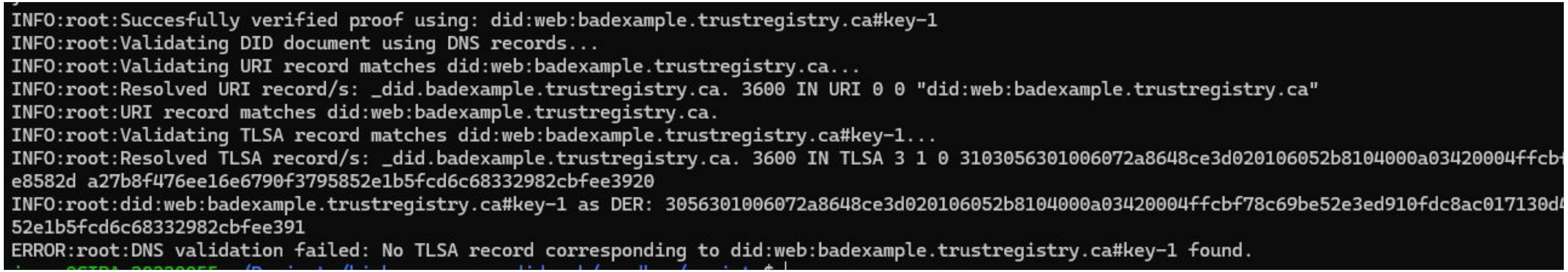


```
jesse@CIRA-20220055:~$ dig _did.badexample.trustregistry.ca @1.1.1.1 URI +dnssec

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> _did.badexample.trustregistry.ca @1.1.1.1 URI +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39778
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
;; QUESTION SECTION:
;_did.badexample.trustregistry.ca. IN   URI

;; AUTHORITY SECTION:
trustregistry.ca.       3600    IN      SOA     ns1.trustregistry.ca. admin.trustregistry.ca. 2024041003 86400 7200 3600000 172800
trustregistry.ca.       3600    IN      RRSIG   SOA 13 2 3600 20240425000000 20240404000000 16050 trustregistry.ca. gQ0Q4XLtHmA5a+MUkdE/9gpkpWnffFfSowniiB1S
S8M/pMGJWyX1Wcl2 7gCzId89vgsQVBpaBe944h3/ftTeag==
_did.badexample.trustregistry.ca. 3600 IN NSEC  ns1.trustregistry.ca. RRSIG NSEC TLSA
_did.badexample.trustregistry.ca. 3600 IN RRSIG NSEC 13 4 3600 20240425000000 20240404000000 16050 trustregistry.ca. LviUGjzR5P+tHvKS9ro42jeo+aAQprACss6NQhe
qUfumWsa+t6EOSrBQ WU22JqLYylJXRO5w33VD2mGRQGgXWg==

;; Query time: 89 msec
;; SERVER: 1.1.1.1#53(1.1.1.1) (UDP)
;; WHEN: Thu Apr 11 12:45:25 EDT 2024
;; MSG SIZE  rcvd: 374
```

Failed verification of did:web:badexample.trustregistry.ca: Invalid TLSA Record



```
jesse@CIRA-20220055:~$ dig  _did.badexample.trustregistry.ca @1.1.1.1 TLSA +dnssec

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> _did.badexample.trustregistry.ca @1.1.1.1 TLSA +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11576
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
;; QUESTION SECTION:
;_did.badexample.trustregistry.ca. IN   TLSA

;; ANSWER SECTION:
_did.badexample.trustregistry.ca. 3600 IN TLSA  3 1 0 3103056301006072A8648CE3D020106052B8104000A03420004FFCBF 78C69BE52E3ED910FDC8AC017130D46B37D50777FBF6B
D56EE4C6FE6 EEC44B95B9E8582DA27B8F476EE16E6790F3795852E1B5FCD6C68332 982CBFEE3920
_did.badexample.trustregistry.ca. 3600 IN RRSIG TLSA 13 4 3600 20240425000000 20240404000000 16050 trustregistry.ca. GZKeZzQnRvGlNubd9D1Na4/+iLk9HUWsBjG9AWS
uoOWqOjrq04uJfGa3 uxzI5jJEPlAiRpxLmZScz/iznaP2qg==

;; Query time: 113 msec
;; SERVER: 1.1.1.1#53(1.1.1.1) (UDP)
;; WHEN: Thu Apr 11 11:54:13 EDT 2024
;; MSG SIZE  rcvd: 278
```



```
INFO:root:Succesfully verified proof using: did:web:badexample.trustregistry.ca#key-1
INFO:root:Validating DID document using DNS records...
INFO:root:Validating URI record matches did:web:badexample.trustregistry.ca...
INFO:root:Resolved URI record/s: _did.badexample.trustregistry.ca. 3600 IN URI 0 0 "did:web:badexample.trustregistry.ca"
INFO:root:URI record matches did:web:badexample.trustregistry.ca.
INFO:root:Validating TLSA record matches did:web:badexample.trustregistry.ca#key-1...
INFO:root:Resolved TLSA record/s: _did.badexample.trustregistry.ca. 3600 IN TLSA 3 1 0 3103056301006072a8648ce3d020106052b8104000a03420004ffcb
e8582d a27b8f476ee16e6790f3795852e1b5fcd6c68332982cbfee3920
INFO:root:did:web:badexample.trustregistry.ca#key-1 as DER: 3056301006072a8648ce3d020106052b8104000a03420004ffcbf78c69be52e3ed910fdc8ac017130d
52e1b5fcd6c68332982cbfee391
ERROR:root:DNS validation failed: No TLSA record corresponding to did:web:badexample.trustregistry.ca#key-1 found.
```

Discussion

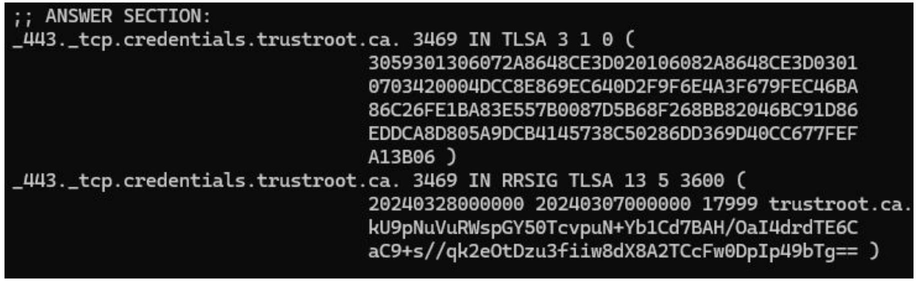
1. Technical
   a. Clarification of any technical detail of POC and demo
2. Implications
   a. Potential input into W3C Recommendations
   b. Independence from trust schemes: browser vendor root lists (CA/B) and EU Qualification Website Authentication Certificates (EU QWAC)
   c. Better enabling domain owners to assert verifiable information.
3. Next Steps
   a. Pilot Project
   b. Further outreach and engagement

Annex Slides

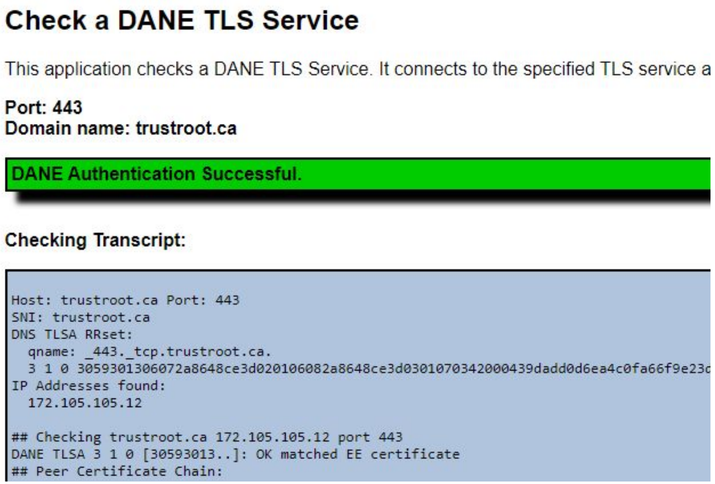
Annex: Independence from CA/B Forum and EU/QWAC.

Trust in the self-signed web site certificates can be achieved via a TLSA record for the web service, port 443 on TCP. No need for expensive CERTs when the issuers already have proper crypto equipment.

https://www.huque.com/bin/danecheck



```
;; ANSWER SECTION:
_443._tcp.credentials.trustroot.ca. 3469 IN TLSA 3 1 0 (
                                3059301306072A8648CE3D020106082A8648CE3D0301
                                0703420004DCC8E869EC640D2F9F6E4A3F679FEC46BA
                                86C26FE1BA83E557B0087D5B68F268BB82046BC91D86
                                EDDCA8D805A9DCB4145738C50286DD369D40CC677FEF
                                A13B06 )
_443._tcp.credentials.trustroot.ca. 3469 IN RRSIG TLSA 13 5 3600 (
                                20240328000000 20240307000000 17999 trustroot.ca.
                                kU9pNuVuRWspGY50TcvpuN+Yb1Cd7BAH/OaI4drdTE6C
                                aC9+s//qk2eOtDzu3fiiw8dX8A2TCcFw0DpIp49bTg== )
```



Check a DANE TLS Service

This application checks a DANE TLS Service. It connects to the specified TLS service a

Port: 443
Domain name: trustroot.ca

DANE Authentication Successful.

Checking Transcript:

```
Host: trustroot.ca Port: 443
SNI: trustroot.ca
DNS TLSA RRset:
  qname: _443._tcp.trustroot.ca.
  3 1 0 3059301306072a8648ce3d020106082a8648ce3d030107034200043 9dadd0d6ea4c0fa66f9e23c
IP Addresses found:
  172.105.105.12

## Checking trustroot.ca 172.105.105.12 port 443
DANE TLSA 3 1 0 [30593013..]: OK matched EE certificate
## Peer Certificate Chain:
```