Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

A. Brotman
Comcast, Inc
S. Farrell
Trinity College Dublin

July 8, 2019

Related Domains By DNS
draft-brotman-rdbd-02

Abstract

This document outlines a mechanism by which a DNS domain can publicly document the existence or absence of a relationship with a different domain, called "Related Domains By DNS", or "RDBD".

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Table of Contents

1. Introduction

[[Discussion of this draft is taking place on the dbound@ietf.org mailing list. There's a github repo for this draft at <https://github.com/abrotman/related-domains-by-dns> - issues and PRs are welcome there.]]

Determining relationships between registered domains can be one of the more difficult investigations on the Internet. It is typical to see something such as "example.com" and "dept-example.com" and be unsure if there is an actual relationship between those two domains, or if one might be an attacker attempting to impersonate the other. In some cases, anecdotal evidence from the DNS or WHOIS/RDAP may be sufficient. However, service providers of various kinds may err on the side of caution and treat one of the domains as untrustworthy or abusive because it is not clear that the two domains are in fact related. This specification provides a way for one domain to explicitly document a relationship with another, utilizing DNS records.

Possible use cases include:

- where a company has websites in different languages, and would like to correlate their ownership more easily, consider "example.de" and "example.ie" registered by regional offices of the same company;

- following an acquisition, a domain holder might want to indicate that example.net is now related to example.com in order to make a later migration easier;

- when doing Internet surveys, we should be able to provide more accurate results if we have information as to which domains are related.

Similarly, a domain may wish to declare that no relationship exists with some other domain, for example "good.example" may want to declare that it is not associated with "g00d.example" if the latter is currently being used in some cousin-domain style attack. In such cases, it is more likely that there can be a larger list of names (compared to the "positive" use-cases) for which there is a desire to disavow a relationship.

It is not a goal of this specification to provide a high-level of assurance as to whether or not two domains are definitely related, nor to provide fine-grained detail about the kind of relationship that may exist between domains.

Using "Related Domains By DNS", or "RDBD", it is possible to declare that two domains are related, or to disavow such a relationship.

We include an optional digital signature mechanism that can somewhat improve the level of assurance with which an RDBD declaration can be handled. This mechanism is partly modelled on how DKIM [RFC6376] handles public keys and signatures - a public key is hosted at the relating-domain (e.g., "example.com") and a reference from the related-domain (e.g., "dept-example.com") contains a signature (verifiable with the "example.com" public key) over the text representation ('A-label') of the two domain names (plus a couple of other inputs).

RDBD is intended to declare or disavow a relationship between registered domains, not individual hostnames. That is to say that the relationship should exist between "example.com" and "dept-example.com", not "foo.example.com" and "bar.dept-example.com" (where those latter two are hosts).

There already exists Vouch By Reference (VBR) [RFC5518], however this only applies to email. RDBD could be a more general purpose solution that could be applied to other use cases, as well as for SMTP transactions.

This document describes the various options, how to create records, and the method of validation, if the option to use digital signatures is chosen.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following terms are used throughout this document:

- Relating-domain: this refers to the domain that is declarating a relationship exists. (This was called the "parent/primary" in -00).

- Related-domain: This refers to the domain that is referenced by the relating-domain, such as "dept-example.com". (This was called the "secondary" in -00.)

2. New Resource Record Types

We define two new RRTYPES, an optional one for the relating-domain (RDBDKEY) to store a public key for when signatures are in use and one for use in related-domains (RDBD).

2.1. RDBDKEY Resource Record Definition

The RDBDKEY record is published at the apex of the relating-domain zone.

The wire and presentation format of the RDBDKEY resource record is identical to the DNSKEY record. [RFC4034]

[[All going well, at some point we'll be able to say...]] IANA has allocated RR code TBD for the RDBDKEY resource record via Expert Review.

The RDBDKEY RR uses the same registries as DNSKEY for its fields. (This follows the precedent set for CDNSKEY in [RFC7344].)

No special processing is performed by authoritative servers or by resolvers, when serving or resolving. For all practical purposes, RDBDKEY is a regular RR type.

The flags field of RDBDKEY records MUST be zero. [[Is that correct/ ok? I've no idea really:-)]]

There can be multiple occurrences of the RDBDKEY resource record in the same zone

2.2. RDBD Resource Record Definition

To declare a relationship exists an RDBD resource record is published at the apex of the related-domain zone.

To disavow a relationship an RDBD resource record is published at the apex of the relating-domain zone.

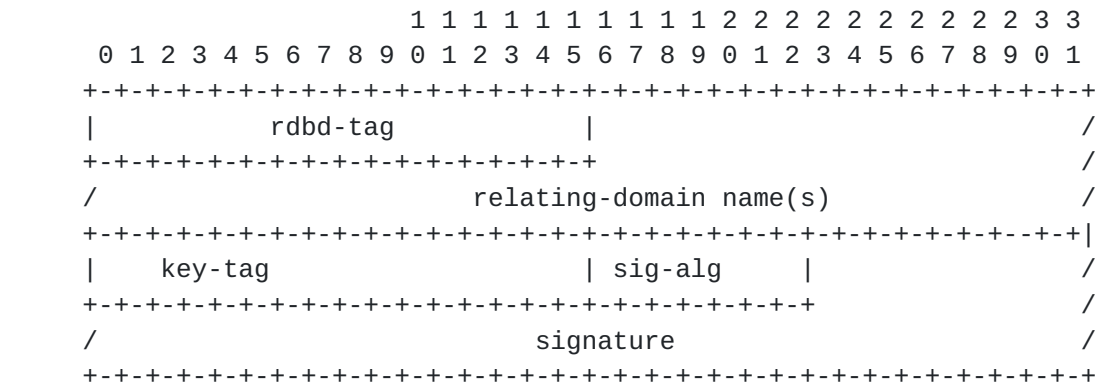
[[All going well, at some point we'll be able to say...]] IANA has allocated RR code TBD for the RDBD resource record via Expert Review.

The RDBD RR is class independent.

The RDBD RR has no special Time to Live (TTL) requirements.

There can be multiple occurrences of the RDBD resource record in the same zone.

The wire format for an RDBD RDATA consists of a two octet rdbd-tag, the relating-domain name(s), and the optional signature fields which are: a two-octet key-tag, a one-octet signature algorithm, and the digital signature bits.



We define two possible values for the rdbd-tag in this specification, later specifications can define new rdbd-tag values:

- 0: states that no relationship exists between the domains

- 1: states that some relationship exists between the domains

The relating-domain name(s) field contains either a single domain name, or an HTTPS URL. In the latter case, successfully de-referencing that URL results in a JSON object that contains the list of domain names, such as is shown in the figure below.

```
[
    "example.com",
    "example.net",
    "foo.example"
]
```

If an optional signature is included, the sig-alg field MUST contain the signature algorithm used, with the same values used as would be used in an RRSIG. The key-tag MUST match the RDBDKEY RR value for the corresponding public key.

If the optional signature is omitted, then the presentation form of the key-tag, sig-alg and signature fields MAY be omitted. If not omitted then the sig-alg and key-tag fields MUST be zero and the signature field MUST be a an empty string. [[Is that the right way to have optional fields in RRs? Not sure.]]

The input to signing ("to-be-signed" data) is the concatenation of the following linefeed-separated (where linefeed has the value '0x0a') lines:

```
relating=<relating-domain name>
related=<related-domain name or URL>
rdbd-tag=<rdbd-tag value>
key-tag=<key-tag>
sig-alg=<sig-alg>
```

The relating-domain and related-domain values MUST be the 'A-label' representation of these names.

The trailing "." representing the DNS root MUST NOT be included in the to-be-signed data, so a relating-domain value above might be "example.com" but "example.com." MUST NOT be used as input to signing.

A linefeed MUST be included after the "sig-alg" value in the last line.

[[Presentation syntax and to-be-signed details are very liable to change.]]

See the examples in the Appendix for further details.

3. Directionality and Cardinality

RDBD relationships are uni-directional. If bi-directional relationships exist, then both domains can publish RDBD RRs and optionally sign those.

If one domain has relationships with many others, then the relevant RDBD RRs (and RDBDKEY RRs) can be published to represent those or one RDBD RR can contain an HTTPS URL at which one can provide a list of names.

4. Required Signature Algorithms

Consumers of RDBD RRs MAY support signature verification. They MUST be able to parse/process unsigned or signed RDBD RRs even if they cannot cryptographically verify signatures.

Implementations producing RDBD RRs SHOULD support optional signing of those and production of RDBDKEY RRs.

Implementations of this specification that support signing or verifying signatures MUST support use of RSA with SHA256 (sig-alg==8) with at least 2048 bit RSA keys. [RFC5702]

RSA keys SHOULD use a 2048 bit or longer modulus.

Implementations of this specification that support signing or verifying signatures SHOULD support use of Ed25519 (sig-alg==15). [RFC8080][RFC8032]

5. Validation

A validated signature is solely meant to be additional evidence that the relevant domains are related, or that one disavows such a relationship. The existence or disavowal of a relationship does not by itself mean that data or services from any domain should be considered as more or less trustworthy.

6. Security Considerations

6.1. Efficiacy of signatures

The optional signature mechanism defined here offers no protection against an active attack if both the RDBD and RDBDKEY values are accessed via an untrusted path.

If the RDBDKEY value has been cached, or is otherwise known via some sufficiently secure mechanism, then the RDBD signature does confirm that the holder of the private key (presumably the relating-domain) considered that the relationship, or lack thereof, with a related-domain was real at some point in time.

6.2. DNSSEC

RDBD does not require DNSSEC. Without DNSSEC it is possible for an attacker to falsify DNS query responses for someone investigating a relationship. Conversely, an attacker could delete the response that would normally demonstrate the relationship, causing the investigating party to believe there is no link between the two domains. An attacker could also replay an old RDBD value that is actually no longer published in the DNS by the related-domain.

Deploying signed records with DNSSEC should allow for detection of these kinds of attack.

If the relating-domain has DNSSEC deployed, but the related-domain does not, then the optional signature can (in a sense) extend the DNSSEC chain to cover the RDBD RR in the related-domain's zone.

If both domains have DNSSEC deployed, and if the relating-domain public key has been cached, then the the signature mechanism provides additional protection against active attacks involving a parent of one of the domains. Such attacks may in any case be less likely and detectable in many scenarios as they would be generic attacks against DNSSEC-signing (e.g. if a regisgtry injected a bogus DS for a relating-domain into the registry's signed zone). If the public key from the relevant RDNDKEY RRs is read from the DNS at the same time as a related RDBD RR, then the signature mechanism provided here may provide litle additional value over and above DNSSEC.

6.3. Lookup Loops

It's conceivable that an attacker could create a loop of relationships, such as a.com->b.com->c.com->a.com or similar. This could cause a resource issue for any automated system. A system SHOULD only perform three lookups from the first domain

(a.com->b.com->c.com->d.com). The related and relating-domains SHOULD attempt to keep links direct and so that only the fewest number of lookups are needed, but it is understood this may not always be possible.

7. IANA Considerations

This document introduces two new DNS RR types, RDBD and RDBDKEY. [[Codepoints for those are not yet allocated by IANA, nor have codepoints been requested so far.]]

[[New rdbd-tag value handling wll need to be defined if we keep that field. Maybe something like: 0-255: RFC required; 256-1023: reserved; 1024-2047: Private use; 2048-65535: FCFS.]]

8. Acknowledgements

Thanks to all who commented on this on the dbound and other lists, in particular to the following who provided comments that caused us to change the draft: Bob Harold, John Levine, Andrew Sullivan, Suzanne Woolf, and Paul Wouters. (We're not implying any of these fine folks actually like this draft btw, but we did change it because of their comments:-) Apologies to anyone we missed, just let us know and we'll add your name here.

9. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC5518] Hoffman, P., Levine, J., and A. Hathcock, "Vouch By Reference", RFC 5518, DOI 10.17487/RFC5518, April 2009, <https://www.rfc-editor.org/info/rfc5518>.

[RFC5702] Jansen, J., "Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5702, DOI 10.17487/RFC5702, October 2009, <https://www.rfc-editor.org/info/rfc5702>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/info/rfc6376>.

[RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, DOI 10.17487/RFC7344, September 2014, <https://www.rfc-editor.org/info/rfc7344>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[RFC8080] Sury, O. and R. Edmonds, "Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC", RFC 8080, DOI 10.17487/RFC8080, February 2017, <https://www.rfc-editor.org/info/rfc8080>.

Appendix A. Examples

This appendix provides examples of RDBD-related values. The following names and other values are used in these examples.

- Relating domain: my.example
- Related domain: my-way.example
- Unrelated domain: my-bad.example
- URL for other related domains: <https://example.com/related-names>
- URL for other unrelaed domains: <https://example.com/unrelateds>

The github repo <https://github.com/abrotman/related-domains-by-dns> has a script in sample/mk_samples.sh that generated this appendix.

A.1. Unsigned Examples

```
;ZONE FILE FRAGMENT STARTS
; assertion that my-way.example claims to be related
; to my.example
my-way.example. 3600 IN RDBD 1 my.example.

; assertion that my-way.example claims not to be
; related to my-bad.example
my-way.example. 3600 IN RDBD 0 my-bad.example.

; assertion that my-way.example claims to be related
; to whatever is at https://example.com/related-names
my-way.example. 3600 IN RDBD 1 https://example.com/related-names

; assertion that my-way.example claims not to be
; related to whatever is at https://example.com/related-names
my-way.example. 3600 IN RDBD 0 https://example.com/unrelateds

;ZONE FILE FRAGMENT ENDS
```

A.2. RSA-signed Example

```
# BASH SNIPPET STARTS
# HOWTO generate RSA key pair
$ openssl genrsa -out rsa.private 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................+++++
.....................+++++
e is 65537 (0x010001)
writing RSA key
$ openssl rsa -in rsa.private -out rsa.public -pubout \
     -outform PEM
$ cat rsa.private
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA8qpGa2i6O76MdGiFVYGgd6UDYfUNl4qHpXfzmrqawUqb8JHM
5m8hjLTbKDGRpZJA5m9sQlaM41hQxpAPuW3mJXYvVOiqxtRCBlVTvUCR+Gkah7LU
FT54U8PT+ciF2or2ArL7O1LIT1PnEL0AZsbM6FwYLsRE8RX89PqeL9N/U4xJ7jQq
SRK88bnRRFs+W7F+iQuA5erHfIZHftq+yzRG8rtgTN7HvTOewkDa2XS2Tn/z0tRV
wNUX4Mi4e/DaEZU0u3kn7HqL0hyBXz/EEHUq9+57BURUT/ZQ41ORGgJ/4+tv2Gp8
mp1ydOJaAhzgqky2zGBL1RS+mBFCio9dXBIJywIDAQABAoIBAQDwE7ITtcr6LKy8
xmOTkul1NVZBZbYKxU0qUaA65n8Q2IWq3jR/jlb85DkmbNQRoL6AvJ+4ifRdQBS6
PfCwnZ/iVCjDsmSyzXB835I3XFiOET3kHvJgCiv1g3qGVvLGolB9nyGbMW1nvjSO
hM6O4AP9po9uRVOHyR84J3K1EmOX/PgmITPel61CEe+IdYwX3K3w1Mmm9C3ZSitx
rx8gpWHknpAG9Z+DA+f406TFTFtuTQe6xCDjKl0/uUGsT62UQqEiw00wekm8e+2b
rDdtqu61Lqq1aakhtGlozXIT7ED+oobp6cAnM1QbV09z7zZevXl95yrXt9xZ3odS
HNcHnMCRAoGBAP6N00vMmplk76OOJ738Twf6fsaL+zkwNmL74Rv85LJn96p6xQUm
wJdMsa7HoD7LwuxhJkhDbdTiq+oV+Mc4J+FAe5xRK5vD7wVmPNWtUcvwdTZXnpNr
```

```
6fU28PqL76feU22G/V2mthgKOXPvYehcteAJAEGbLwQoo1Q7qZa7Q9i/AoGBAPQL
KSqcOzGHJ57iYN+HbN0cqXpBburA7dZstl6WiwQO0U4+KsJ3eyyuUE0Mz6epXeQD
Ry1W8yoH9ypfLDEP3YwP3wp7dqISyGAzsRospGyGJ0tHkQ3orW+/6PgSl9W2aJV2
LqyfOPutrjfIRUkgALMq8cTRxR1xpx0HpVfOMCX1AoGAJi9SPfGgU1hf1lIRxh8e
H91EvTXsZqTD089i8lbaW6Ta8xjdiytIAqo/kS9i62iXgewE2Rw8Uo36KfBH1GKp
INISeN14RDJ9HXs7rvYD6irU+mTkZcrvWph2R69MMQtZynlQcob6k9qcybZkIn4d
zlCrWCwWPnJ2JcGZbAIFaHMCgYBXIdD95LABu/a6dKsPxANrYrtj6g7XBDEmuMPY
O7nApiW24N1Vd2FkD4yeJe/SNddO/JiiKIRDQnrOBxL5JWf9hQEmdfRiY4BlUK9v
3/aIxNEswI2awLODzao5QDIz3J+0lXCOs36d5WHpiriqJiH51mBh3F+bZqO66qrv
EbABLQKBgDUhcsEyIMI6oygKO/L+iVOZM+ao+64TtKxqUmgHU+gBgvdkY8h9GvxY
kL7L1hBOB9bAyP8ObU18R5CMbQgO7PITACRoU+uYIiQ67UREDKjBRtoDcgK0JrRX
kqy4v6LuuFCpS5OJKQL6rKfk2XNSjDD1ZwLuBZjHQDdc/gnn7XRx
-----END RSA PRIVATE KEY-----
$ cat rsa.public
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8qpGa2i6O76MdGiFVYGg
d6UDYfUNl4qHpXfzmrqawUqb8JHM5m8hjLTbKDGRpZJA5m9sQlaM41hQxpAPuW3m
JXYvVOiqxtRCBlVTvUCR+Gkah7LUFT54U8PT+ciF2or2ArL7O1LIT1PnEL0AZsbM
6FwYLsRE8RX89PqeL9N/U4xJ7jQqSRK88bnRRFs+W7F+iQuA5erHfIZHftq+yzRG
8rtgTN7HvTOewkDa2XS2Tn/z0tRVwNUX4Mi4e/DaEZU0u3kn7HqL0hyBXz/EEHUq
9+57BURUT/ZQ41ORGgJ/4+tv2Gp8mp1ydOJaAhzgqky2zGBL1RS+mBFCio9dXBIJ
ywIDAQAB
-----END PUBLIC KEY-----
# To-be-signed data for RSA
$ cat  to-be-signed-8.txt
relating=my.example
related=my-way.example
rdbd-tag=1
key-tag=38501
sig-alg=8
# Sign that
$ openssl dgst -sha256 -sign rsa.private -out rsa.sig \
        to-be-signed-8.txt
# Hexdump of signature
$ hexdump rsa.sig
0000000 8664 bd57 8cbf a8e1 9182 1b5f a4fc 5eb9
0000010 49b4 fe21 f1c7 8097 ed90 44a5 bcb1 543c
0000020 f784 c190 e1d9 2f2b 18ca d3c2 640f 3823
0000030 7f8a e446 d0eb bd14 6077 0597 6015 a82b
0000040 42d7 8677 b1a3 37fa a1e8 8109 07ec ff62
0000050 16b8 3895 66de d992 dc4d ed99 9ec3 0a62
0000060 6a07 3baa 45f2 d528 1e83 a147 60ce 9b25
0000070 a967 4ba0 3fb5 98db 5ff3 b070 058b 4d8f
0000080 f198 6c1f e6b6 7a6c 1e8c ad42 237f 5440
0000090 7856 caac f96c f87d e79c 4dc5 b833 bc03
00000a0 c52e 5603 46a7 59b5 9fe3 fccd 04ee e908
00000b0 71e7 21f8 47ad fea8 40bf 14a5 9e6b b3d4
00000c0 c61a 5b96 c559 3491 4dfa 91a0 4c0b f3ff
```

```
00000d0 e460 484c 7e49 5368 85e3 16be fe6b 809a
00000e0 117d c2cb be19 c5ba 7594 2f60 16ad 1132
00000f0 f978 6ca1 5448 180f 8ca7 e73d 1137 7064
0000100
# BASH SNIPPET ENDS

; ZONE FILE FRAGMENT STARTS
; The RDBDKEY RR for my.example is...
my.example. 3600 IN RDBDKEY 0 3 8 (
            LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlJQklqQU5C
            Z2txaGtpRzl3MEJBUUVGQUFPQ0FROEFNSUlCQ2dLQ0FRRUE4
            cXBHYTJpNk83Nk1kR2lGVllHZwpkNlVEWWZVTmw0cUhwWGZ6
            bXJxYXdVcWI4SkhNNW04aGpMVGJLREdScFpKQTVtOXNRbGFN
            NDFoUXhwQVB1VzNtCkpYWXZWT2lxeHRSQ0JsVlR2VUNSK0dr
            YWg3TFVGVDU0VThQVCtjaUYyb3IyQXJMN08xTElUMVBuRUww
            QVpzYk0KNkZ3WUxzUkU4Ulg4OVBxZUw5Ti9VNHhKN2pRcVNS
            Szg4Ym5SUkZzK1c3RitpUXVBNWVySGZJWkhmdHEreXpSRwo4
            cnRnVE43SHZUT2V3a0RhMlhTMlRuL3owdFJWd05VWDRNaTRl
            L0RhRVpVMHUza243SHFMMGh5Qlh6L0VFSFVxCjkrNTdCVVJV
            VC9aUTQxT1JHZ0ovNCt0djJHcDhtcDF5ZE9KYUFoemdxa3ky
            ekdCTDFSUyttQkZDaW85ZFhCSUoKeXdJREFRQUIKLS0tLS1F
            TkQgUFVCTElDIEtFWS0tLS0tCg== )
; The RDBD RR to be published by my-way.example is...
my-way.example. 3600 IN RDBD 1 my.example 38501 8 (
            ZIZXvb+M4aiCkV8b/KS5XrRJIf7H8ZeAkO2lRLG8PFSE95DB
            2eErL8oYwtMPZCM4in9G5OvQFL13YJcFFWArqNdCd4ajsfo3
            6KEJgewHYv+4FpU43maS2U3cme3DnmIKB2qqO/JFKNWDHkeh
            zmAlm2epoEu1P9uY819wsIsFj02Y8R9stuZseoweQq1/I0BU
            Vnisymz5ffic58VNM7gDvC7FA1anRrVZ45/N/O4ECOnncfgh
            rUeo/r9ApRRrntSzGsaWW1nFkTT6TaCRC0z/82DkTEhJfmhT
            44W+Fmv+moB9EcvCGb66xZR1YC+tFjIRePmhbEhUDxinjD3n
            NxFkcA== )
; ZONE FILE FRAGMENT ENDS
```

A.3. Ed25519-signed Example

```
# BASH SNIPPET STARTS
# HOWTO generate an Ed25519 key pair...
$ ./ed25519-signer.py -s rdbd-example0001rdbd-example0002 \
   -r my.example -d my-way.example
private:b'726462642d6578616d706c6530303031726462642d6578616d
706c6530303032'
public:b'353fc31e1168c91f0af65d6c26fd441fb7df9671a23a746bb3e
c86be8d35b648'
b64pubkey: NT/DHhFoyR8K9l1sJv1EH7fflnGiOnRrs+yGvo01tkg=
keyid: 35988
to-be-signed:|relating=my.example
related=my-way.example
rdbd-tag=1
key-tag=35988
sig-alg=15
|
sig:b'64bc444ce759fb9435fe9c1875eb241c4ec6d0995cd8138a372782
32fc8e79f53cb8f88059f6040054c61be8cfd73fd44521f73994628fc7c3
0135fa929ab00f'
# hex dump of Ed25519 private
$ hexdump ed25519.priv
0000000 6472 6462 652d 6178 706d 656c 3030 3130
0000010 6472 6462 652d 6178 706d 656c 3030 3230
0000020
# hex dump of Ed25519 public
$ hexdump ed25519.pub
0000000 3f35 1ec3 6811 1fc9 f60a 6c5d fd26 1f44
0000010 dfb7 7196 3aa2 6b74 ecb3 be86 358d 48b6
0000020
# hex dump of Ed25519 signature
$ hexdump ed25519.sig
0000000 bc64 4c44 59e7 94fb fe35 189c eb75 1c24
0000010 c64e 99d0 d85c 8a13 2737 3282 8efc f579
0000020 b83c 80f8 f659 0004 c654 e81b d7cf d43f
0000030 2145 39f7 6294 c78f 01c3 fa35 9a92 0fb0
0000040
# BASH SNIPPET ENDS

; ZONE FILE FRAGMENT STARTS
; The RDBDKEY RR for my.example is...
my.example. 3600 IN RDBDKEY 0 3 15 (
            NT/DHhFoyR8K9l1sJv1EH7fflnGiOnRrs+yGvo01tkg= )
; The RDBD RR to be published by my-way.example is...
my-way.example. 3600 IN RDBD 1 my.example 35988 15 (
            ZLxETOdZ+5Q1/pwYdeskHE7G0Jlc2BOKNyeCMvyOefU8uPiA
            WfYEAFTGG+jP1z/URSH3OZRij8fDATX6kpqwDw== )
; ZONE FILE FRAGMENT ENDS
```

Appendix B. Ed25519 Signing Code

Since OpenSSL does not yet support Ed25519 signing via its command line tool, we generate our example using the python script below, which is called as "ed25519-signer.py" above. This uses the python library from Appendix A of [RFC8032].

```
#!/usr/bin/env python3
# CODE_BEGINS
import argparse, sys, binascii
from eddsa2 import Ed25519

# from https://gist.github.com/wido/4c6288b2f5ba6d16fce37dca3fc2cb4a
def calc_keyid(flags, protocol, algorithm, dnskey):
    st=struct.pack('!HBB', int(flags), int(protocol), int(algorithm))
    st+=base64.b64decode(dnskey)
    cnt = 0
    for idx in range(len(st)):
        s = struct.unpack('B', st[idx:idx+1])[0]
        if (idx % 2) == 0:
            cnt += s << 8
        else:
            cnt += s
    return ((cnt & 0xFFFF) + (cnt >> 16)) & 0xFFFF

def main():
    parser=argparse.ArgumentParser(description='Ed25519 signing')
    parser.add_argument('-s','--secret',
                        dest='secret', help='secret key')
    parser.add_argument('-r','--relating',
                        dest='relating', help='relating domain')
    parser.add_argument('-d','--related',
                        dest='related', help='related domain')
    parser.add_argument('-n','--negative',
                        dest='negative', help='negative assertion')
    args=parser.parse_args()

    if args.secret is None:
        print("You do need a secret... - exiting")
        sys.exit(1)
    # secret has to be 32 octets funnily enuugh:-)
    # e.g. secret="rdbd-example0001rdbd-example0002".encode('utf-8')
    if len(args.secret)!=32:
        print("Secret has to be 32 octets... - exiting")
        sys.exit(1)
    if args.relating is None:
        print("You do need a relating domain... - exiting")
```

```
            sys.exit(1)
        if args.related is None:
            print("You do need a related domain... - exiting")
            sys.exit(1)
        secret=args.secret.encode('utf-8')
        privkey,pubkey = Ed25519.keygen(secret)
        print("private:"+ str(binascii.hexlify(privkey)))
        print("public:"+ str(binascii.hexlify(pubkey)))

        b64pubkey=binascii.b2a_base64(pubkey).rstrip().decode("utf-8")
        print("b64pubkey: " + b64pubkey)

        keyid=calc_keyid("0","3","15",b64pubkey)
        print("keyid: " + str(keyid))

        rdbdtag="1"
        if args.negative:
            rdbdtag="0"

        tbs="relating="+args.relating+"\nrelated="+\
              args.related+"\nrdbd-tag="+rdbdtag+\
              "\nkey-tag="+str(keyid)+"\nsig-alg=15\n"
        print("to-be-signed:|" + str(tbs)+"|")

        with open("ed25519.priv", "wb") as privf:
            privf.write(privkey)
        with open("ed25519.pub","wb") as pubf:
            pubf.write(pubkey)
        with open("to-be-signed-15.txt","wb") as tbsf:
            tbsf.write(tbs.encode('utf-8'))

        msg=tbs.encode('utf-8')
        signature = Ed25519.sign(privkey, pubkey, msg)
        print("sig:"+ str(binascii.hexlify(signature)))
        with open("ed25519.sig", "wb") as sigf:
            sigf.write(signature)
        return

    if __name__ == "__main__":
        main()

    # CODE_ENDS
```

Appendix C. Changes and Open Issues

[[RFC editor: please delete this appendix]]

C.1. Changes from -01 to -02

- Added negative assertions based on IETF104 feedback
- Added URL option based on IETF104 feedback
- Made sample generation script
- Typo fixes etc.

C.2. Changes from -00 to -01

- Changed from primary/secondary to relating/related (better suggestions are still welcome)
- Moved away from abuse of TXT RRs
- We now specify optional DNSSEC-like signatures (we'd be fine with moving back to a more DKIM-like mechanism, but wanted to see how this looked)
- Added Ed25519 option
- Re-worked and extended examples

C.3. Open Issues

Current open github issues include:

- #5: specify input for signing more precisely - e.g. is there a CR or NULL or not
- #6: what, if anything, does rdbd for example.com mean for foo.example.com?

These can be seen at: <https://github.com/abrotman/related-domains-by-dns/issues>

Authors' Addresses

Alex Brotman
Comcast, Inc

Email: alex_brotman@comcast.com

Stephen Farrell
Trinity College Dublin

Email: stephen.farrell@cs.tcd.ie