TOC

Network Working Group	J. Lindberg
Internet-Draft	Google
Intended status: Standards Track	S. Farrell
Expires: July 18, 2010	NewBay Software
	January 14, 2010

Domain Name Assertions
draft-ietf-xmpp-dna-00

Abstract

An application-level approach to asserting and proving the delegated ownership of a domain name for XMPP.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

TOC

As large hosting providers begin providing XMPP services for multiple domains, several issues with previous mechanisms for server-to-server federation have become apparent.

A large number of sockets are currently required between hosting providers. Although servers may attempt to piggyback whenever possible, the possibility exists that 2*N*M sockets will be created (where N is the number of domains on one hosting provider, and M is the number of domains on another hosting provider). The goal would be that the number of sockets was dependent on load or deployment considerations.

In order to enable or require encryption, the hosting provider must create a separate socket for each hostname pair and have access to a X.509 certificate that is signed by a widely-trusted CA and includes both the public and private keys. Customers of hosting providers might be uncomfortable with the level of trust this requires.

This document lays out an approach known as Domain Name Assertions (DNA) that allows providers to effectively host XMPP services on behalf of other companies, and might be expanded later to support other protocols.

2. Acknowledgements

TOC

The original version of this specification was written by Joe Hildebrand and Sean Turner.

3. Glossary

TOC

Hosted domain An XMPP domain whose network services are delegated to a third party.

Hosting provider A business entity that provides services for one or more domains that it does not directly and fully control.

Self-hosted domain A domain whose owner acts as its hosting provider.

Delegation A ceremony that acts as proof of the intent of the owner of a hosted domain to cede control to a hosting provider for a given protocol.

Widely-trusted CA For open communities, a Certificate Authority whose certificate that is trusted by multiple web browsers by

> default. For closed communities, a Certificate Authority that is trusted by all members of that community.
>
> **Sender Domain** The domain associated with the 'from' address on a stanza to be sent across a federation boundary.
>
> **Target Domain** The domain associated with the 'to' address on a stanza to be sent across a federation boundary.
>
> **Originating Server** The machine that wants to send a message from an entity at the Sender Domain to an entity at the Target Domain and thus is attempting to establish a connection between the two servers.
>
> **Receiving Server** The machine to which the Originating Server has opened a connection for the purpose of sending a message from the Sender Domain to the Target Domain.
>
> **Asserting Entity** A system element (such as a server) asserting a given domain name as an identity.
>
> **Validating Entity** A system element (such as a client or server) that checks a given identity asserted by an asserting entity.
>
> **Asserted Domain** A domain name asserted by either side of a conversation. validating entities may require assertions to be backed up with proof.
>
> **Proof** A secure mechanism through which a validating entity can ascertain that an asserting entity has authority for the asserted domain, either directly or indirectly (by delegation).

4. Requirements

TOC

1. A hosting provider MUST be able to service domains for which it cannot obtain certificates signed by a widely-trusted CA.

2. All network traffic (except for initial handshakes) MUST be encrypted in a manner not subject to man-in-the-middle attacks.

3. The number of socket connections between hosting providers MUST NOT be a function of the number of domains hosted by either provider.

4. Connections MUST be usable in either direction, if allowed by policy and deployment considerations.

5. The owners of the hosted domain MUST NOT be required to give out private keying material associated with any certificate they own that has been signed by a widely-trusted CA.

6. The owners of the hosted domain MUST be allowed to choose the frequency with which they wish to perform the delegation ceremony.

7. The owners of the hosted domain MUST be allowed to revoke their delegation at any time.

8. Multiple mechanisms for proving delegation MUST be possible.

9. It MUST be possible for new assertions to be added to a stream at any point after the stream is fully established, but before the stream is closed

5. Generic Use Cases

TOC

The DNA mechanism can be used for multiple different protocols. In particular, client-to-server XMPP and server-to-server XMPP are discussed herein, but the general approach could be used for non-XMPP protocols such as SMTP or IMAP. As such, the protocol syntax offered here is normative for XMPP, but merely illustrative for other protocols, which will need their own protocol bindings.
The following domain names are used in the examples in this section:

asserted.tld The domain name being asserted.

5.1. Assertions

TOC

The asserting entity asserts a domain name through the use of an "assert" element. Assertions MUST contain a "from" attribute naming the domain name that is being asserted.

```
<assert xmlns='urn:ietf:params:xml:ns:dna' from='asserted.tld'/>
```

TOC

5.2. Validation

When the validating entity has been satisified that the asserting entity is authoritative for the domain name asserted, it MUST send a "valid" element. At this point, the asserting entity MAY send stanzas to the validating entity containing from addresses in the asserted and validated domain.

Validating entities SHOULD respond to a domain name assertion without asking for further proof when the domain name asserted is represented in the certificate offered by the asserting entity according to the rules set out in [rfc3920bis] (Saint-Andre, P., “Extensible Messaging and Presence Protocol (XMPP): Core,” September 2009.).

Validating entities MAY respond to a domain name assertion without asking for further proof when the domain name asserted is known to be associated with the asserting entity through some other secure means such as DNSSEC, caching, or local knowledge. In the DNSSEC case, the server hostname in the SRV record used to connect SHOULD be looked for in the certificate offered by the asserting entity, according to the rules set out in [rfc3920bis] (Saint-Andre, P., “Extensible Messaging and Presence Protocol (XMPP): Core,” September 2009.).

```
     <valid xmlns='urn:ietf:params:xml:ns:dna' to='asserted.tld'/>
```

5.3. Invalidation

TOC

When the validating entity does not accept proof offered by the asserting entity, or it no longer trusts the proof (for example due to the proof timing out or being revoked), it sends the asserting entity an "invalid" element. The asserting entity MUST NOT send any stanzas to the validating entity containing from addresses in the invalidated domain without performing another validation.

Invalid responses MAY be given in direct response to an assertion if the validating entity has reason to believe that no proof is possible. Examples that would cause this response include a blocking list or a negative cache.

```
     <invalid xmlns='urn:ietf:params:xml:ns:dna' to='asserted.tld'/>
```

5.4. Requesting proof with a challenge

TOC

If an assertion cannot be validated immediately, the validating entity may ask for further proof. Inside the "challenge" element, at least one form of proof that will be acceptable MUST be given to the asserting entity. If an acceptable proof format is not available for the asserted domain, the validating entity MUST return an invalid response proactively.

A "proof" element designates a type of proof that would be acceptable to the verifying entity. The "type" attribute of the "proof" element MUST be a valid URI. A registry of proof types will be created with the IANA (see Section 9 (IANA Considerations)). Standard proof types will begin with "urn:ietf:params:dna:proof:". Custom proofs should be signaled with a "type" attribute value containing a full URI under the control of the defining party. Proof types MUST be compared for equality using the rules for comparing URIs.

Some proof types MAY require information or nonces from the validating entity. If so, the specification for that proof type MUST specify extensions to the "proof" element in a new namespace.

In some protocols, a challenge MAY be sent without an assertion, if the validating entity has reason to believe that the entity with which it is talking is authoritative for a given domain.

```
<challenge xmlns='urn:ietf:params:xml:ns:dna' to='asserted.tld'>
  <proof xmlns='urn:ietf:params:dna:proof:attribute-cert'/>
  < type='http://example.com/proof/custom'/>
</challenge>
```

5.5. No proof possible

TOC

If the validating entity requires proof, but will not accept proof by a means that the asserting entity has available for the asserted domain, the asserting entity MUST respond with an "impossible" element. The validating entity MUST NOT send a "valid" or "invalid" element in response, and MUST NOT accept stanzas from the asserted domain on this connection unless some other assertion works in the future.

The "impossible" element MAY be sent after full validation, if the asserting entity would like to retract the assertion.

```
<impossible xmlns='urn:ietf:params:xml:ns:dna' from='asserted.tld'/>
```

5.6. Proving a challenge

TOC

If an asserting entity thinks it can prove a given assertion when challenged, it sends that proof in a "proof" element. The REQUIRED "type" attribute specifies the chosen proof type, and the REQUIRED "from" attribute specifies the domain being proved. Each proof type MUST specify the format of the contents of the "proof" element. Suggestions for formats include Base64-encoded binary as character data or structured XML in a new namespace.

```
     <proof xmlns='urn:ietf:params:xml:ns:dna'
            from='asserted.tld'
            type='urn:ietf:params:dna:proof:attribute-cert'>
       (Base64-encoded attribute certificate)
     </proof>
```

6. Attribute Certificate Proof

TOC

When an asserting entity has been delegated responsibility for hosting a given domain, an Attribute Certificate MUST be used to prove the delegation. The proof type URI associated with attribute certificates SHALL be 'urn:ietf:params:dna:proof:attribute-cert' (EDITOR'S NOTE: We will work with IANA to come up with a good URN. This is just a placeholder.)

The certificate that signed the attribute certificate MUST have been acceptable as proof of ownership of a given domain for the protocol in question, according to the rules in [rfc3920bis] (Saint-Andre, P., “Extensible Messaging and Presence Protocol (XMPP): Core,” September 2009.). Validating entities SHOULD try prepending the asserted domain with "www." and re-checking for name matches before rejecting the signing certificate, in order to allow for easier deployments using existing web certificates as proof.

Each protocol that is delegated will be assigned its own OID to identify a service and whether the entity can act as a server or client. These values will be included in the Access Identifier attribute, from [I-D.ietf-pkix-3281update] (Housley, R., Farrell, S., and S. Turner, “An Internet Attribute Certificate Profile for Authorization,” April 2009.). This document defines the OIDs for XMPP. Other documents may specify additional OIDs.

The remaining paragraphs in this section profile the attribute certificate issuers public key certificate and the attribute certificate.

6.1. Attribute Certificate Issuer Profile

TOC

The following is a profile of the attribute certificate issuer's public key certificate, which is as per [I-D.ietf-pkix-3281update] (Housley, R., Farrell, S., and S. Turner, “An Internet Attribute Certificate Profile for Authorization,” April 2009.):

- *The issuer's certificate MUST conform to [RFC5280] (Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” May 2008.).

- *The issuer's certificate MUST have a keyUsage extension with the digitalSignature bit set.

- *The issuer's certificate MUST NOT have a basicConstraints extension with the cA BOOLEAN set to TRUE.

In addition to the [I-D.ietf-pkix-3281update] (Housley, R., Farrell, S., and S. Turner, “An Internet Attribute Certificate Profile for Authorization,” April 2009.) requirements, the subject name MUST be non-NULL in the attribute certificate issuer's public key certificate.

6.2. Attribute Certificate Profile

TOC

The attribute certificate issued MUST conform to [I-D.ietf-pkix-3281update] (Housley, R., Farrell, S., and S. Turner, “An Internet Attribute Certificate Profile for Authorization,” April 2009.). There are options in that profile and the following profiles those options:

- *The holder field MUST be the baseCertificateID.

- *The attributes field MUST include the Access Identity attribute, as specified in Section 4.4.2 of [I-D.ietf-pkix-3281update] (Housley, R., Farrell, S., and S. Turner, “An Internet Attribute Certificate Profile for Authorization,” April 2009.). Both the service and ident fields' GeneralName choice MUST be registeredID. The service and ident fields MUST be as defined in Section 5.3. Other attributes MAY be included.

- *The extensions field MUST include the non-critical noRevAvail extension, as defined in Section 4.3.6 of [I-D.ietf-pkix-3281update] (Housley, R., Farrell, S., and S. Turner, “An Internet Attribute Certificate Profile for Authorization,” April 2009.), to indicate that no revocation information is available from the attribute certificate issuer.

*The extensions field MAY include:

-The authorityKeyIdentifier extension if the issuer has more than one key pair.

-The issuerAltName extension if the issuer's certificate includes the subjectAltName extension. If included issuerAltName MUST be marked non-critical.

TOC

6.3. Access Identity Values

The following paragraphs define the service and ident values for the delegated protocols. Currently, only values for XMPP are defined. A later version of this document or another document may specify additional values for other protocols.

TOC

6.3.1. XMPP

When XMPP is delegated the following procedures MUST be followed. The service field MUST be id-xmpp. The following object identifier identifies that the AC holder supports XMPP:

```
id-xmpp OBJECT IDENTIFIER ::= { TBD }
```

The ident field MUST be either id-xmpp-client or id-xmpp-server. Both id-xmpp-client and id-xmpp-server MAY appear in the same attribute certificate. Note that the Access Identity attribute will be multi-valued when both id-xmpp-client and id-xmpp-server are present. The following object identifier identifies the AC holder as the XMPP client:

```
id-xmpp-client OBJECT IDENTIFIER ::= { id-xmpp 0 }
```

The following object identifier identifies the AC holder as the XMPP server:

```
id-xmpp-server OBJECT IDENTIFIER ::= { id-xmpp 1 }
```

TOC

6.4. Attribute Certificate Signature Algorithms

The issuer MUST support signing attribute certificate with the PKCS #1 version 1.5 signature algorithm with SHA-256, as specified in [RFC4055] (Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," June 2005.).

6.5. Proof Encoding

TOC

The attribute certificate, the issuer's certificate, and all of the CA certificates in the trust chain of the signing certificate back to the trust anchor are encoded as a "certs-only" SMIME message, as per [I-D.ietf-smime-3851bis] (Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification," May 2009.) (i.e, a degenerate SignedData with no content just certificates). The resulting message is then Base64 encoded, as per Section 6.8 of [RFC2045] (Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," November 1996.). The end result is then transmitted as the character data of a "proof" element.

7. DNA for XMPP Federation

TOC

Two XMPP servers, each of which hosts multiple domains that they do not directly control, desire to connect in order to exchange traffic for at least two of those domains (one on either side).
The following domain names are used in the examples:

target.tld The domain portion of the JID in the to address of the stanza that caused this connection to be initiated.

othertarget.tld The domain portion of the JID in the to address of a stanza being sent across a stream that was originally started to talk to target.tld.

targetprovider.tld The hosting provider for target.tld.

server.targetprovider.tld A server with XMPP federation services at the target's hosting provider.

originator.tld The domain portion of the JID in the from address of the stanza that caused this connection to be initiated.

originatingprovider.tld
The hosting provider for target.tld.

server.originatingprovider.tld A server with XMPP federation services at the originator's hosting provider.

7.1. DNS SRV lookups

TOC

In a delegated hosting scenario, DNS SRV records are REQUIRED, since otherwise the hosting provider will never be contacted for the target domain. As specified by [rfc3920bis] (Saint-Andre, P., “Extensible Messaging and Presence Protocol (XMPP): Core,” September 2009.) the originating server looks up the target domain to find a list of receiving servers. If the originating server already has a connection to the IP address represented by one of these servers (perhaps because it is communicating with another domain hosted by this provider), it MAY reuse that stream (see Stream Reuse). If the originating server does not have a connection it wants to reuse, it performs the SRV algorithm to select an SRV record and makes a TCP connection to the server and port specified by the selected SRV record.

Unless assured by a mechanism such as DNSSEC, the originating server MUST NOT trust the information received from the DNS SRV as proof that the target domain has been delegated to the receiving server.

```
% dig +short -t SRV _xmpp-server._tcp.target.tld
0 1 5269 server.targetprovider.tld
```

7.2. Certificates during Start-TLS

TOC

The first step during stream negotiation MUST be Start-TLS. The receiving server MUST offer a certificate signed by a widely-trusted CA. The receiving server MUST require a client certificate. The certificate offered by the originating server MUST be signed be a widely-trusted CA. Both sides MUST check the certificate offered to it for validity (e.g. time period, signatures, and trust anchor), but MUST NOT disconnect when the certificate received does not contain a name matching its expectations.

The names on these certificates SHOULD be associated with the relevant hosting provider, and need not be related to the domains being hosted. If the certificates have the name of the server offered in the SRV record, it MAY be possible to use DNSSEC for proof in the future.

```
     CN=server.targetprovider.tld


     CN=server.originatingprovider.tld
```

7.3. Discovering Support

TOC

To and from addresses are REQUIRED in the stream:stream tag. These represent the first domain pair associated with this stream, and are the domain names from the stanza that caused this connection to be established.
To announce its support for DNA, the receiving server asserts its identity in the stream features following TLS negotiation.

```
     <stream:features>
       <assert xmlns='urn:ietf:params:xml:ns:dna' from='target.tld'/>
     </stream:features>
```

7.4. Turning on DNA

TOC

If the originating server supports DNA, it looks for an assertion in the stream features. If it finds none, it MAY fall back on another means of verifying the identiy of the target server, if allowed by local policy.
Originating servers that support DNA talking to target servers that declare support for DNA MUST NOT send protocol other than DNA negotations until they are able to validate the assertion offered by the target server in the stream features. The first validation proves to the originating server that it is talking to a server authoritative for the target domain, so that it is safe to use this domain in "to" addresses on this stream.
Once an originating server completes this first validation it signals that it is willing and able to participate in bi-directional XMPP federation traffic, as long as all of the domains required have been asserted and validated at least once on this stream.
If the originating server does not require more proof (due to a certificate match or DNSSEC-verified delegation), it may send a "valid" element without requesting proof first, as in all DNA interactions.

```
<challenge xmlns='urn:ietf:params:xml:ns:dna' to='target.tld'>
  <proof type='urn:ietf:params:dna:proof:attribute-cert'/>
</challenge>
```

```
<proof xmlns='urn:ietf:params:xml:ns:dna'
       from='target.tld'
       type='urn:ietf:params:dna:proof:attribute-cert'>
  (Base64-encoded attribute certificate)
</proof>
```

```
<valid xmlns='urn:ietf:params:xml:ns:dna' to='target.tld'/>
```

7.5. Asserting new domains

TOC

Before either side sends stanzas on a given stream, it MUST ensure that the other side will accept those stanzas by asserting the domain in the "from" attribute of those stanzas, and waiting for a "valid" response before sending the stanzas in question.
The originating server MUST therefore send its own assertion after accepting the target domain's assertion.

```
<assert xmlns='urn:ietf:params:xml:ns:dna' from='originator.tld'/>
```

7.6. Proactive challenges

TOC

Before either side sends stanzas on a given stream, it MUST ensure that the other side is authoriative for the domain in the "to" attribute on those stanzas. If the sender has already accepted an assertion on this stream, and that assertion has not been revoked with an "impossible" element, no action is required. Otherwise, the sender can proactively request proof for that domain by sending a challenge even though the other side has not sent an assertion for that domain yet.

```
<challenge xmlns='urn:ietf:params:xml:ns:dna' to='othertarget.tld'>
  <proof type='urn:ietf:params:dna:proof:attribute-cert'/>
</challenge>
```

7.7. Proactive validation

TOC

When two hosting providers connect, they may have previous knowledge (perhaps from a cache) of which domains they will trust on the new connection. If so, either side MAY send as many "valid" elements as desired, even though the other side has not sent assertions for those domain.

The server receiving these proactive validations MUST NOT change its self-image (which domains it thinks it is authoritative for), but SHOULD NOT send assertions for these domains on this stream. If the server receiving a proactive validation is no long authoritative for a given domain, it MUST send an "impossible" element, at which point the sender MUST remove the receiver from any cache and not send any stanzas on this stream to the given domain.

Any cache of DNA information SHOULD be associated with the certificate offerred by the relevant server, and SHOULD be checked for revocation if possible, according to local policy.

```
<valid xmlns='urn:ietf:params:xml:ns:dna' to='target1.tld'/>
<valid xmlns='urn:ietf:params:xml:ns:dna' to='target2.tld'/>
<valid xmlns='urn:ietf:params:xml:ns:dna' to='target3.tld'/>
```

7.8. Reusing streams

TOC

DNA streams are bi-directional, and may have an aribtrary number of domains validated in either direction, at any point in the lifetime of the stream. Before sending a stanza on a given stream, the sender MUST ensure that "valid" elements have been exchanged according to the above rules for both the "to" and "from" address, and that no "invalid" or "impossible" element has revoked an assertion.

An "impossible" or "invalid" element SHOULD NOT cause the rest of the stream to become invalidated in either directions. When these elements are seen, they SHOULD merely change the list of domains that are valid on that stream. If no domains are valid on the stream, the stream MAY be closed immediately, or MAY be left open if desired. If left open, the stream MUST NOT be used for stanza traffic until domains are asserted as needed for the desired domains.

Domains that are marked as "invalid" or "impossible" SHOULD NOT be retried on the same stream unless new information has become available, in order to prevent assertion storms.

TOC

7.9. Implementation notes

If the first server-to-server validation exchange fails, the parties MAY keep the connection open (perhaps for a shorter than is usual) in case another domain pair would need a connection between these servers. Ensure that only one challenge is outstanding on a given connection for a given domain. Ensure that only one assertion or one proof is outstanding on a given connection for a given domain.

8. DNA for XMPP client connections

TOC

Hosting providers have a similar problem for client to server connections. Clients need to ensure that they are talking to an authoritative server for the domain they intend to log in to. Typically, this is done by examining the certificate offered by the server during TLS negotiation, according to the rules in [rfc3920bis] (Saint-Andre, P., “Extensible Messaging and Presence Protocol (XMPP): Core,” September 2009.). However, hosting providers will typically not have access to a valid certificate for the target domain and its associated private key. DNA can be used for the hosting provider to prove that hosting services have been delegated to it.

8.1. Announcing Support

TOC

To announce its support for DNA, the server asserts its identity in the stream features following TLS negotiation. The server MUST offer the identity of the domain specified in the client's stream header "to" attribute.

```
<stream:features>
  <assert xmlns='urn:ietf:params:xml:ns:dna' from='target.tld'/>
</stream:features>
```

8.2. Client challenges for proof

TOC

To utilize the server's DNA assertion, the client performs Start-TLS per [rfc3920bis] (Saint-Andre, P., “Extensible Messaging and Presence Protocol (XMPP): Core,” September 2009.), however, if the client does not find a name match on the offered certificate, it does not disconnect immediately. Instead, if the server offers an assertion, it can use the name from that assertion to ask the server for proof of delegation.

Subsequent protocol follows the generic use cases above, with the exception that alternate or additional domain names MUST NOT be asserted. If the server returns an "impossible" element, the server MUST terminate the stream. If the client sends an "invalid" element, the client MUST terminate the stream.

```
<challenge xmlns='urn:ietf:params:xml:ns:dna' to='asserted.tld'>
  <proof type='urn:ietf:params:dna:proof:attribute-cert'/>
</challenge>
```

9. IANA Considerations

TOC

9.1. XML Namespace Name for DNA

TOC

A URN sub-namespace for Domain Name Assertion (DNA) negotiation data in the Extensible Messaging and Presence Protocol (XMPP) is defined as follows. (This namespace name adheres to the format defined in .)

> **URI:** urn:ietf:params:xml:ns:dna
>
> **Specification:** XXXX
>
> **Description:** This is the XML namespace name for Domain Name Assertion (DNA) negotiation data in the Extensible Messaging and Presence Protocol (XMPP) as defined by XXXX.
>
> **Registrant Contact:** IETF, XMPP Working Group, <xmppwg@xmpp.org>

9.2. URN space for standard DNA Proof Types

TOC

A URN sub-namespace for DNA is defined as follows. (This namespace name adheres to the format defined in .)

> **URI:** urn:ietf:params:dna:proof
>
> **Specification:** XXXX

Description:
This is the sub-namespace for standardized Domain Name Assertion (DNA) proof types as defined by XXXX.

Registrant Contact: IETF, XMPP Working Group, <xmppwg@xmpp.org>

9.3. DNA Proof Registry

TOC

The URNs inside urn:ietf:params:dna:proof

9.4. Object Identifiers

TOC

The following OIDs are defined in Section 5.3 of this document:

- *id-xmpp
- *id-xmpp-client
- *id-xmpp-server

10. Internationalization Considerations

TOC

The domains offered MUST conform to all of the rules for "Domain Identitifiers", as specified in &rfc3920bis;, including (but not limited to) the rules for syntax, cannonicalization and comparison.

11. Security Considerations

TOC

TBD.

12. Normative References

TOC

[rfc3920bis]	Saint-Andre, P., “Extensible Messaging and Presence Protocol (XMPP): Core,” draft-ietf-xmpp-3920bis-02 (work in progress), September 2009 (TXT).

[I-D.ietf-pkix-3281update]	Housley, R., Farrell, S., and S. Turner, “An Internet Attribute Certificate Profile for Authorization,” draft-ietf-pkix-3281update-05 (work in progress), April 2009 (TXT).
[I-D.ietf-smime-3851bis]	Ramsdell, B. and S. Turner, “Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification,” draft-ietf-smime-3851bis-11 (work in progress), May 2009 (TXT).
[RFC2045]	Freed, N. and N. Borenstein, “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies,” RFC 2045, November 1996 (TXT).
[RFC4055]	Schaad, J., Kaliski, B., and R. Housley, “Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” RFC 4055, June 2005 (TXT).
[RFC5280]	Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” RFC 5280, May 2008 (TXT).

TOC

Appendix A. RELAX NG XML Schema

```
default namespace = "urn:ietf:params:xml:ns:dna"

# Intent: Internationalized Domain Name (simplisitic view)
domain = xsd:string { pattern = "(\p{L}|\p{N})(\p{L}|\p{N}|\p{M}|-)*"
"(\.(\p{L}|\p{N})(\p{L}|\p{N}|\p{M}|-)*)*"}

assert = element assert {
  attribute from { domain }
}

valid = element valid {
  attribute to { domain }
}

invalid = element valid {
  attribute to { domain }
}

proof = element proof {
 attribute type { xsd:anyURI },
 attribute from { domain }?,
 text?
}

challenge = element challenge {
  proof+
}

start = assert | valid | invalid | proof | challenge
```

Authors' Addresses

TOC

	Jonas Lindberg
	Google
Email:	jonasl@google.com
	Stephen Farrell
	NewBay Software
Email:	sfarrell@newbay.com