ENUM -- Telephone Number Mapping Working Group
Internet-Draft
Expires: August 26, 2006

A. Mayrhofer
enum.at
February 22, 2006

Telephone Number Mapping and Domain Keys as a Distributed Identity Infrastructure
draft-mayrhofer-enum-domainkeys-00

Status of this Memo

This Internet-Draft will expire on August 26, 2006.

Copyright Notice

Abstract

This document creates a decentralized indentity infrastructure by combining technology from E.164 Number Mapping (ENUM) and DomainKeys Identified Mail (DKIM). This infrastructure uses E.164 numbers as identities, ENUM DNS for key distribution, and leverages the trust relations from ENUM validation to actual messages signed by the number holder.

Table of Contents

1. Introduction

E.164 numbers [2] serve as well known addressing elements for communication on various networks (voice calls, instant text messages, video calls). Besides their primary addressing role, those numbers also serve as identities for the owner, and are sometimes even part of an authentication process.

E.164 Number Mapping (ENUM) [1] associates each E.164 number with a DNS domain. Since ENUM validation [3] ensures that only the holder of a certain E.164 number can aquire and control the respective ENUM domain, the contents of such a domain is under the descretion of the number holder.

DomainKeys Identified Mail (DKIM) FIXME-ref describes a mechanism where owners of a domain publish the public part of a cryptographic key in the DNS and sign messages with the private part. Entities receiving suche messages verify the signature by discovering and fetching the public key directly from the DNS without prior contact with the sender.

By combining selected parts of ENUM and DKIM technology, any owner of a phone number can potentially convey the identity his number reflects to any other entity on the Internet in a decentralized way. For the purpose of the following sections, the proposed infrastructure is abbreviated as "ENDI" (ENUM Distributed Identity).

Please note that this document is currently just aimed at conveying the idea - most parts of the proposed infrastructure need to be described in more detail in upcoming versions of this draft.

2. Infrastructure Components

The proposed identity infrastructure "ENDI" uses only certain parts of the ENUM and DKIM specifications. Most of the definitions in DKIM focus on email, and do not apply to the infrastructure described. ENUM, on the other hand, has a lot of features which are not neccessary for the service described, but would make implementations fully incompatible with existing DKIM tools. The following section describes the components used in this approach.

2.1. The Identifier

The identifier used to convey an identity MUST be a fully qualified E.164 number, including the leading "plus" sign. Numbers which are not valid E.164 numbers MUST NOT be used as an identifier, as well as numbers which are local to a certain network, and may therefore

introduce collisions in the identity space (they wouldn't work anyways). ENDI applications SHOULD NOT attempt to convert such local numbers into E.164 space, since guessing identities is always bad. Clients MAY, however, apply "dial plan" style normalizations as well as whitespace stripping, removal of non-numeric characters to the input string. Applications SHOULD give feedback to the user about normalizations applied.

Example of an identifier string (office identity of the author):

```
+431505641634
```

2.2. Identifier Domain Mapping

The mapping from an identifier to a domain name is to be performed as described in RFC3761, section 2.1 ("Application Unique String"), and section 2.4 ("Valid databases"). However, processing should be stopped at list item 4. of RFC3761, section 2.4 since ENDI is not a full Dynamic Delegation Discovery System (DDDS) application. The final output of this step is a full qualified domain name, as outlined there.

Example of the domain mapping (for the identifier listed above):

```
4.3.6.1.4.6.5.0.5.1.3.4.e164.arpa
```

A client application SHOULD NOT confuse the user by displaying the domain - it SHOULD always display/request the identifier instead.

2.3. Key Management and Storage

Key management is to be done according to section 3.6 of draft-ietf-dkim-base-00. The identifier domain mapping described above is to be used as "domain".

In contrary to to a full DDDS application, a "TXT" type resource record is proposed, which makes the propsed infrastructure compatible with existing Domainkeys toolkits. The "selector" mechanism as described in draft-ietf-dkim-base-00, section 3.1 may be used for the reasons outlined in said document, especially if different applications sharing a single E.164 number each require a seperate key.

Example of public key (stored for the identifier above):

```
@ORIGIN _domainkey.4.3.6.1.4.6.5.0.5.1.3.4.e164.arpa.
@ IN TXT "p=MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAM2r/
A7PvMlW7p6imaNlwjwTRp/
```

xvIsbbSpLF7fyBMv2PWdy0dCwrkEpfLQGfKS6P+cLPSw6OTjlgt3IK7Jr5KcCAwEAAQ"

Note: line breaks introduced due to format limits

2.4. Signing Actions

Generally, signing follows the principles outlined in draft-ietf-dkim-base-00. However, there are a few differences to due to the facts that only a subset of DKIM is used, and this subset is potentially applied to protocols different from email:

- Relation between Users and Keys: Most DKIM domains will be shared by many users of the same domain, which means that the private key part will be shared by many users as well. Contrary to that, it is envisaged that the domain mapped identifier proposed in this document is not shared (unless the underlying E.164 number is shared as well), and therefore one ENDI key pair per user is set up.
- Signing and key availability: In DKIM, signing is expected to occur on the outbound SMTP server rather than in the user's application itself, which makes the entity operating the server the Signer. In ENDI signing is expected to take place in the application itself, which makes the Signer identical to the User, and in turn qualifies the protocol for end user authentication.
- Since the Signer needs access to the private key DKIM requires the key to be present on the outbound SMTP server. Since ENDI applications sign messages by themselves the private key needs to reside in the application.

2.5. Verification Actions

Again, verification principles follow the principles of DKIM. However, the actual implementation of the verification steps is specific to the application, as is the interpretation of the verification result. Section 3 lists a few examples.

Verification steps:

1. Extracting the signature
2. Extracting the identity
3. Fetching the public key
4. Verifying the signature
5. Interpreting the results

3. Application scenarios

The following section tries to outline a few potential applications of ENDI. Obviously, the examples should be treated as rough conceptual sketches rather than finished protocols.

3.1. Peer to Peer Communications

Many communication services are set up in the way of distributed peer to peer networking - most often with the drawback that for addressing and authentication a centralized server is still necessary in most cases. Even if no authentication is required, the mechanism for assigning unique identifiers to each user still mandates a central component.

Some P2P applications leverage existing addressing schemes as like as email addresses, and attach new credentials to those addresses. However, a centralized component caching/managing those credentials is still required.

By using the proposed architecture, P2P networks could replace those central functions in the following way:

- o Identity allocation: The user uses his existing E.164 number as his identifier. This solves the problem of unique identifiers, since E.164 numbers can only be assigned once, and the allocation mechanism is already in place. In addition, phone numbers are well established addressing elements - using them to be reachable on just another network requires no new address book entry.
- o Authentication: Without any prior knowledge, any node in the P2P network is able to verify an authentication request signed by the owner of the E.164 number. Credential verification by a central component is replaced by an ad hoc verification of signed messages.
- o Usability: Re-using a well known identifier (the E.164 number) for just another service without risking of giving any more information is good. Other users will easily accomodate to this identifier, since they are already using it on other networks to contact the user (eg. on the Public Switched Telephony Network).

3.2. Trusted Caller ID

Voice over IP (VoIP) gateways need to signal a calling party number for calls which traverse from the Internet to the PSTN. Even if the call itself would be free (as free Beer), authentication is required to ensure that the caller ID presented resembles the number allocated to the user. A gateway which does not have access to respective data sources would be unable to provide a trusted caller ID to the PSTN.

If, however, the gateway receives (and sucessfully verifies) a call request signed by ENDI technology, it can safely assume that the E.164 number presented is under control of the calling party (because the public key used for verification is under the control of the E.164 number holder). Subsequently, it can safely signal the E.164 number presented as calling party ID to the PSTN.

3.3. Spam over Internet Telephony (SPIT) Prevention

It is safe to assume that once open VoIP endpoint deployment take up, spammers will closely follow. One concept of fighting SPIT is whitelisting, however, for SIP this approach has the same drawbacks as email whitelisting - spammers regularly use addresses which are likely to be already whitelisted.

By using an E.164 number as identifier for the whitelist (probably auto-generated from the user's address book) together with a mechanism to verify the E.164 number of a caller, SPIT potential could be greatly reduced. ENDI would provide a mechanism to securely convey the E.164 identity of a caller.

3.4. Secure Real Time Transport Protocol (SRTP) key exchange

SRTP provides encryption of the media stream to VoIP applications. Before encrypted communication takes places, keys have to be exchanged. If the call parties know the E.164 numbers of the other party from signaling, they could use the ENDI public keys for media encryption instead

4. IANA considerations

There are no considerations for IANA (yet)

5. Security Considerations

Obviously, a protocol dealing with cryptographic keys, authentication/authorization has to be analyzed in depth for security concerns. Most of the security concerns from DKIM apply to this protocol as well, and with no doubt more issues will come up due to the combination with another technology.

[analysis required here]

6. Acknowledgements

This specification contains information that was derived from the original DKIM and ENUM documents.

The Author wishes to thank Klaus Darilion for his ideas.

7. References

7.1. Normative References

[1] Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 3761, April 2004.

[2] ITU-T, "The international public telecommunication numbe ring plan", Recommendation E.164, May 1997.

7.2. Informative References

[3] Mayrhofer, A. and B. Hoeneisen, "ENUM Validation Architecture", draft-ietf-enum-validation-arch-01 (work in progress), February 2006.

Author's Address

Alexander Mayrhofer
enum.at GmbH
Karlsplatz 1/9
Wien A-1010
Austria

Phone: +43 1 5056416 34
Email: alexander.mayrhofer@enum.at
URI: http://www.enum.at/

Intellectual Property Statement

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.