Internet Engineering Task Force	P. M. Hallam-Baker
Internet-Draft	Comodo Group Inc.
Intended status: Standards Track	R. N. Stradling
Expires: April 26, 2012	Comodo CA Ltd.
	S. Farrell
	Trinity College Dublin
	D. Kutscher
	NEC
	B. Ohlman
	Ericsson
	October 24, 2011

The Named Information (ni) URI Scheme: Parameters
draft-hallambaker-decade-ni-params-00

Abstract

This document specifies some optional algorithms and parameters that may be used in the query string of ni URIs.

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 http://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 April 26, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

The ni URI scheme [nischeme] supports extensibility in terms of the algorithm used to derive a value (normally, but not always a strong digest algorithm) and via support for a query-string thay can contain a list of key=value pairs. This document defines some uses for both of these extensibility points and creates IANA registries that can be used to register additional algorithms and key strings for use in the query part of an ni name.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 *[RFC2119]*.

[[Comments are included in double-square brackets, like this.]]

[[Note that the features here are less mature than the specification in the [nischeme] document. The intent is to develop these as required for the various use-cases as we go. If something from here appears to be as widely useful as the core ni scheme, then the authors are willing to move features from this document to the core document. We are also happy to incoroporate features to handle additional use-cases here if those arise.]]

2. Additional Algorithms

This section specifies two additional algorithms that MAY be used to handle truncated hashes and hashes calculated over dynamically changing objects.
For these optional algorithms, we establish a new IANA algorithm registry in Section 5.

2.1. Truncated Hashes

Message Digest algorithms are designed to provide protection against a collision attack. Due to the birthday paradox, this requires that the digest length be twice the length of a related encryption or authentication key to achieve the same work factor. Generally, hash function outputs will therefore be long, of the order of 256 bits (32 bytes raw, 43 bytes base64 encoded) or more. However, in some applications, strong collision resistance is not required, and ni names with shorter-hashes can be used without affecting security.
Different hash algorithm identifiers MUST be used for truncated hashes, that is, implementations MUST NOT accept digest values that are shorter than the (encoded) length for the specified hash algorithm.
We define the sha-256-32 algorithm as being the leftmost 32 bits of output of the sha-256 algorithm and this algorithm is registered in Section 5.
[[Note: we probably need some discussion to pick a good truncated hash.]]
The following example includes an authority and uses a truncated variant of SHA-256.
ni://example.com/sha-256-32;B_K97zTtFuOhug27fke4_Q
[[Note: examples need to be checked.]]

2.2. Hashed Dynamic Content

The ni scheme involves calculating digest values over content objects. That works fine with static objects but is problematic for objects whose value is dynamically generated. In this section we define an algorithm that supports the same core "name-data integrity" service for dynamic objects. The basic idea is simply to include a hash of a public key in the ni name, and then for the dynamic object to be digitally signed with the corresponding private key. With a little work to handle the various useful formats, this allows a client that is presented with the ni name and the signed object to verify the binding between the name and the object data.
Note that the signature scheme used might or might not provide additional information, e.g. a name for the signer. Applications might benefit from that, but it is not required in order to provide the core "name-data integrity" service for dynamically generated objects.
Since there are a number of digital signing schemes that might be used, our approach is to define a new algorithm for the ni scheme that takes

as input a specific public key encoded in a specific way, and runs that through a digest function. That is, the ni algorithm fields will specify both a public key algorithm and a digest algorithm, just as is done with digital signature algorithm identifiers.

Since it is possible that an ni algorithm might also be defined where the value contains an actual digital signature we need to be careful to ensure there is no ambiguity. However, since the lengths of signatures and hash outputs are (with current algorithms) always different, we could use that fact to disambigute between rsa-with-sha256 meaning the value is a sha256 hash of an rsa public key and the alternative meaning the the value is an rsa-with-sha256 signature. However, we prefer to use a new registry (see Section 5 to ensure disambiguate these. The basic ni URI scheme requires algorithms to be chosen from the RFC 5698 registry, [RFC5698] for dynamic content, an algorithm from the registry defined here MUST be used.

We define one such algorithm, "pk-rsa-with-sha256" that takes an RSA public key as input, with the input bits formatted as a SubjectPublicKeyInfo as defined by RFC 5280. [RFC5280] Note that this does not mean that one cannot use e.g. PGP to sign the actual object. It means that if you do use PGP then in order to verify the name-data integrity, the client needs to extract the signer's PGP public key, then reformat that as a SubjectPublicKeyInfo and then run that through the sha-256 algorithm and make the relevant comparison.

3. Query String Paramaeters

This section defines query string parameters that MAY be used to indicate the type of content hashed or to specify additional locations from which the named content can be retrieved. We also define a way to specify how an encryption key MAY be included in an ni URI that allows for decryption of object content.

3.1. Digest with Content Type

The semantics of a digest being used to establish a secure reference from an authenticated source to an external source may be a function of associated meta data such as the content type. This data MAY be specified by means of a parameter:

ni:sha-256;B_K97zTtFuOhug27fke4_Zgc4Myz4b_lZNgsQjy6fkc?ct=text/plain

The Content Type "ct" parameter specifies the MIME Content Type of the associated data as defined in [RFC4288]

3.2. Additional Locators

In addition to the algorithm for mapping an ni URI to an HTTP(S) URL specified in the ni scheme definition [nischeme], an ni name MAY provide information about additional locations from which the referenced content might be available. This is done via the inclusion of an "alt" or "alts" key in the query string that can supply more

values for the authority field when constructing the HTTP or HTTPS URL.
For example:
ni://example.com/sha-256;B_K97zTtFuOhug27fke4_Zgc4Myz4b_lZNgsQjy6fkc?alt=ni.example.net
The corresponding content might then also be retrieved from the URL:
http://ni.example.net/.well-known/ni/sha-256/B_K97zTtFuOhug27fke4_Zgc4Myz4b_lZNgsQjy6fkc
A ni name MAY specify multiple locations from which the content MAY be obtained:
ni:///sha-256;B_K97zTtFuOhug27fke4_Zgc4Myz4b_lZNgsQjy6fkc?alt=one.example.com&alt=two.example.com
The above example asserts that the content might be retrieved from either of the following URIs:
http://one.example.com/.well-known/ni/sha-256/B_K97zTtFuOhug27fke4_Zgc4Myz4b_lZNgsQjy6fkc
http://two.example.com/.well-known/ni/sha-256/B_K97zTtFuOhug27fke4_Zgc4Myz4b_lZNgsQjy6fkc
The "alt" parameter means "use HTTP" and the "alts" parameter means use "HTTPS".
The alt and alts parameters are used to specify a possible means of resolving the referenced content. Multiple locator parameters MAY be used to specify alternative sources for accessing the content.
The alt and alts parameters take a single argument, the authority to be used for resolution. To permit the use of ni URIs in ASCII-only environments, the ASCII encoding (aka punycode) of the domain name MUST be used. [[Note sure if this is needed/correct.]]

3.3. Digest with Decryption Key

An ni name MAY provide a key for decrypting the referenced data. The use-case here is where the referenced data has be distributed (somehow) in ciphertext form, probably with little or no access control required (since the data is strongly encrypted) and where a client wishing to decrypt that data subsequently acquires an ni name for that data that provides the required decryption key.
Clearly, to be of any benefit, access to the ni name that includes the decryption key MUST be controlled so that only the appropriate clients get access to the ni name and of course this ni name MUST be strongly protected via some (probably mutual) authentication and confidentiality service such as can be provided by TLS. [RFC5246]
ni///:sha-256;B_K97zTtFuOhug27fke4_Zgc4Myz4b_lZNgsQjy6fkc?enc=aes-cbc:Fw3x20nEKfq6FDGzq7ttIQ
The "enc" specifier is used when the encrypted object consists of the ciphertext alone. The "menc" spcifier is used when the encrypted object consists of a MIME header containing metadata followed by the binary object encoding. [[Note: there may be more needed here.]]
The encryption specifiers both take an agrument of the form:
algorithm ":" base64url (key) [":" base64url (iv)]
Where

algorithm
Is the algorithm used to encrypt the associated content

key Is the value of the cryptographic key

iv (optional) Is the value of the cryptographic Initialization Vector.

If the IV is not spcified for a block cipher mode that requires one, the IV MUST be prepended to the encrypted content.

[[Note: Actually the IV does not provide any additional security for this application but explaining the reason might be more effort than it is worth and what we really care about is saving bytes in the identifier, not the resulting data package.]

4. Security Considerations

[[We need to say when its safe, or not, to use truncated hashes.]]
[[More TBD no doubt.]]

5. IANA Considerations

5.1. Creation of ni additional algorithms registry

This specification creates a new IANA registry entitled "ni additional algorithms."
The policy for future assignments to the registry is "RFC Required".
The initial contents of the registry are:

Parameter	Meaning	Reference
sha-256-32	SHA-256 truncated to 32 bits	[RFC-THIS]
pk-rsa-with-sha256	Public key input to SHA-256	[RFC-THIS]

5.2. Creation of ni parameter registry

This specification creates a new IANA registry entitled "Named Information URI Parameter Definitions".
The policy for future assignments to the registry is "RFC Required".
The initial contents of the registry are:

```
Parameter    Meaning                                       Reference
-----------  --------------------------------------------  ---------
ct           Content Type                                  [RFC-THIS]
alt          Additional HTTP Locator                       [RFC-THIS]
alts         Additional HTTPS Locator                      [RFC-THIS]
enc          Encryption Key                                [RFC-THIS]
menc         Encryption Key                                [RFC-THIS]
```

6. References

[nischeme]	Farrell, S, Kutscher, D, Ohlman, B and P Hallam-Baker, "The Named Information (ni) URI Scheme: Core Syntax", Internet-Draft draft-farrell-decade-ni-00, October 2011.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC4288]	Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.
[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.
[RFC5246]	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
[RFC5698]	Kunz, T., Okunick, S. and U. Pordesch, "Data Structure for the Security Suitability of Cryptographic Algorithms (DSSC)", RFC 5698, November 2009.

Authors' Addresses

Phillip Hallam-Baker Hallam-Baker Comodo Group Inc. EMail: philliph@comodo.com

Rob Stradling Stradling Comodo CA Ltd. EMail: rob.stradling@comodo.com

Stephen Farrell Farrell Trinity College Dublin Dublin, 2 Ireland Phone: +353-1-896-2354 EMail: stephen.farrell@cs.tcd.ie

Dirk Kutscher Kutscher NEC Kurfuersten-Anlage 36 Heidelberg, Germany EMail: kutscher@neclab.eu

Boerje Ohlman Ohlman Ericsson Stockholm, S-16480 Sweden EMail: Borje.Ohlman@ericsson.com