Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 16, 2012

M. Blanchet
Viagenie
A. Sullivan
January 13, 2012

Stringprep Revision Problem Statement
draft-ietf-precis-problem-statement-04.txt

Abstract

Using Unicode codepoints in protocol strings that expect comparison with other strings requires preparation of the string that contains the Unicode codepoints. Internationalizing Domain Names in Applications (IDNA2003) defined and used Stringprep and Nameprep. Other protocols subsequently defined Stringprep profiles. A new approach different from Stringprep and Nameprep is used for a revision of IDNA2003 (called IDNA2008). Other Stringprep profiles need to be similarly updated or a replacement of Stringprep needs to be designed. This document outlines the issues to be faced by those designing a Stringprep replacement.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Internationalizing Domain Names in Applications (IDNA2003) [RFC3490], [RFC3491], [RFC3492], [RFC3454] described a mechanism for encoding Unicode labels making up Internationalized Domain Names (IDNs) as standard DNS labels. The labels were processed using a method called Nameprep [RFC3491] and Punycode [RFC3492]. That method was specific to IDNA2003, but is generalized as Stringprep [RFC3454]. The general mechanism can be used to help other protocols with similar needs, but with different constraints than IDNA2003.

Stringprep defines a framework within which protocols define their Stringprep profiles. Known IETF specifications using Stringprep are listed below:

- The Nameprep profile [RFC3490] for use in Internationalized Domain Names (IDNs);
- NFSv4 [RFC3530] and NFSv4.1 [RFC5661];
- The iSCSI profile [RFC3722] for use in Internet Small Computer Systems Interface (iSCSI) Names;
- EAP [RFC3748];
- The Nodeprep and Resourceprep profiles [RFC3920] for use in the Extensible Messaging and Presence Protocol (XMPP), and the XMPP to CPIM mapping [RFC3922] (the latter of these relies on the former);
- The Policy MIB profile [RFC4011] for use in the Simple Network Management Protocol (SNMP);
- The SASLprep profile [RFC4013] for use in the Simple Authentication and Security Layer (SASL), and SASL itself [RFC4422];
- TLS [RFC4279];
- IMAP4 using SASLprep [RFC4314];
- The trace profile [RFC4505] for use with the SASL ANONYMOUS mechanism;
- The LDAP profile [RFC4518] for use with LDAP [RFC4511] and its authentication methods [RFC4513];
- Plain SASL using SASLprep [RFC4616];
- NNTP using SASLprep [RFC4643];
- PKIX subject identification using LDAPprep [RFC4683];
- Internet Application Protocol Collation Registry [RFC4790];
- SMTP Auth using SASLprep [RFC4954];
- POP3 Auth using SASLprep [RFC5034];
- TLS SRP using SASLprep [RFC5054];
- IRI and URI in XMPP [RFC5122];
- PKIX CRL using LDAPprep [RFC5280];
- IAX using Nameprep [RFC5456];
- SASL SCRAM using SASLprep [RFC5802];
- Remote management of Sieve using SASLprep [RFC5804];

o The i;unicode-casemap Unicode Collation [RFC5051].

There turned out to be some difficulties with IDNA2003, documented in [RFC4690]. These difficulties led to a new IDN specification, called IDNA2008 [RFC5890], [RFC5891], [RFC5892], [RFC5893]. Additional background and explanations of the decisions embodied in IDNA2008 is presented in [RFC5894]. One of the effects of IDNA2008 is that Nameprep and Stringprep are not used at all. Instead, an algorithm based on Unicode properties of codepoints is defined. That algorithm generates a stable and complete table of the supported Unicode codepoints. This algorithm is based on an inclusion-based approach, instead of the exclusion-based approach of Stringprep/Nameprep.

This document lists the shortcomings and issues found by protocols listed above that defined Stringprep profiles. It also lists some early conclusions and requirements for a potential replacement of Stringprep.

2. Issues raised during newprep BOF

During IETF 77, a BOF discussed the current state of the protocols that have defined Stringprep profiles [NEWPREP]. The main conclusions from that discussion were as follows:

o Stringprep is bound to a specific version of Unicode: 3.2. Stringprep has not been updated to new versions of Unicode. Therefore, the protocols using Stringprep are stuck to Unicode 3.2.

o The protocols need to be updated to support new versions of Unicode. The protocols would like to not be bound to a specific version of Unicode, but rather have better Unicode agility in the way of IDNA2008. This is important partly because it is usually impossible for an application to require Unicode 3.2; the application gets whatever version of Unicode is available on the host.

o The protocols require better bidirectional support (bidi) than currently offered by Stringprep.

o If the protocols are updated to use a new version of Stringprep or another framework, then backward compatibility is an important requirement. For example, Stringprep is based on and may use NFKC [UAX15], while IDNA2008 mostly uses NFC [UAX15].

o Protocols use each other; for example, a protocol can use user identifiers that are later passed to SASL, LDAP or another authentication mechanism. Therefore, common set of rules or classes of strings are preferred over specific rules for each protocol.

Protocols that use Stringprep profiles use strings for different

purposes:

- o XMPP uses a different Stringprep profile for each part of the XMPP address (JID): a localpart which is similar to a username and used for authentication, a domainpart which is a domain name and a resource part which is less restrictive than the localpart.
- o iSCSI uses a Stringprep profile for the IQN, which is very similar to (often is) a DNS domain name.
- o SASL and LDAP uses a Stringprep profile for usernames.
- o LDAP uses a set of Stringprep profiles.

During the newprep BOF, it was the consensus of the attendees that it would be highly desirable to have a replacement of Stringprep, with similar characteristics to IDNA2008. That replacement should be defined so that the protocols could use internationalized strings without a lot of specialized internationalization work, since internationalization expertise is not available in the respective protocols or working groups.

3. Major Topics for Consideration

This section provides an overview of major topics that a Stringprep replacement needs to address. The headings correspond roughly with categories under which known Stringprep-using protocol RFCs have been evaluated. For the details of those evaluations, see Appendix A.

3.1. Comparison

3.1.1. Types of Identifiers

Following [I-D.iab-identifier-comparison], we can organize identifiers into three classes in respect of how they may be compared with one another:

Absolute Identifiers Identifiers that can be compared byte-by-byte for equality.

Definite Identifiers Identifiers that have a well-defined comparison algorithm on which all parties agree.

Indefinite Identifiers Identifiers that have no single comparison algorithm on which all parties agree.

Definite Identifiers include cases like the comparison of Unicode code points in different encodings: they do not match byte for byte, but can all be converted to a single encoding which then does match byte for byte. Indefinite Identifiers are sometimes algorithmically comparable by well-specified subsets of parties. For more discussion of these categories, see [I-D.iab-identifier-comparison].

The section on treating the existing known cases, Appendix A uses these categories.

3.1.2. Effect of comparison

The three classes of comparison style outlined in Section 3.1.1 may have different effects when applied. It is necessary to evaluate the effects if a comparison results in a false positive, and what the effects are if a comparison results in a false negative, especially in terms of the consequences to security and usability.

3.2. Dealing with characters

This section outlines a range of issues having to do with characters in the target protocols, and spends some effort to outline the ways in which IDNA2008 might be a good analogy to other protocols, and ways in which it might be a poor one.

3.2.1. Case folding, case sensitivity, and case preservation

In IDNA2003, labels are always mapped to lower case before the Punycode transformation. In IDNA2008, there is no mapping at all: input is either a valid U-label or it is not. At the same time, upper-case characters are by definition not valid U-labels, because they fall into the Unstable category (category B) of [RFC5892].

If there are protocols that require upper and lower cases be preserved, then the analogy with IDNA2008 will break down. Accordingly, existing protocols are to be evaluated according to the following criteria:

1. Does the protocol use case folding? For all blocks of code points, or just for certain subsets?
2. Is the system or protocol case sensitive?
3. Does the system or protocol preserve case?

3.2.2. Stringprep and NFKC

Stringprep profiles may use normalization. If they do, they use NFKC [UAX15]. It is not clear that NFKC is the right normalization to use in all cases. In [UAX15], there is the following observation regarding Normalization Forms KC and KD: "It is best to think of these Normalization Forms as being like uppercase or lowercase mappings: useful in certain contexts for identifying core meanings, but also performing modifications to the text that may not always be appropriate." For things like the spelling of users' names, then, NFKC may not be the best form to use. At the same time, one of the nice things about NFKC is that it deals with the width of characters

that are otherwise similar, by canonicalizing half-width to full-width. This mapping step can be crucial in practice. The WG will need to analyze the different use profiles and consider whether NFKC or NFC is a better normalization for each profile.

For the purposes of evaluating an existing example of Stringprep use, it is helpful to know whether it uses no normalization, NFKC, or NFC.

3.2.3. Character mapping

Along with the case mapping issues raised in Section 3.2.1, there is the question of whether some characters are mapped either to other characters or to nothing during Stringprep. [RFC3454], Section 3, outlines a number of characters that are mapped to nothing, and also permits Stringprep profiles to define their own mappings.

3.2.4. Prohibited characters

Along with case folding and other character mappings, many protocols have characters that are simply disallowed. For example, control characters and special characters such as "@" or "/" may be prohibited in a protocol.

One of the primary changes of IDNA2008 is in the way it approaches Unicode code points. IDNA2003 created an explicit list of excluded or mapped-away characters; anything in Unicode 3.2 that was not so listed could be assumed to be allowed under the protocol. IDNA2008 begins instead from the assumption that code points are disallowed, and then relies on Unicode properties to derive whether a given code point actually is allowed in the protocol.

Moreover, there is more than one class of "allowed in the protocol". While some code points are disallowed outright, some are allowed only in certain contexts. The reasons for the context-dependent rules have to do with the way some characters are used. For instance, the ZERO WIDTH JOINER and ZERO WIDTH NON-JOINER (ZWJ, U+200D and ZWNJ, U+200C) are allowed with contextual rules because they are required in some circumstances, yet are considered punctuation by Unicode and would therefore be DISALLOWED under the usual IDNA2008 derivation rules. The goal is to provide the widest possible repertoire of code points possible and consistent with the traditional DNS, trusting to the operators of individual zones to make sensible (and usually more restrictive) policies for their zones.

IDNA2008 may be a poor model for what other protocols ought to do in this case, because it is designed to support an old protocol that is designed to operate on the scale of the entire Internet. Moreover, IDNA2008 is intended to be deployed without any change to the base

DNS protocol. Other protocols may aim at deployment in more local environments, or may have protocol version negotiation built in.

3.2.5. Internal structure, delimiters, and special characters

IDNA2008 has a special problem with delimiters, because the delimiter "character" in the DNS wire format is not really part of the data. In DNS, labels are not separated exactly; instead, a label carries with it an indicator that says how long the label is. When the label is presented in presentation format as part of a fully qualified domain name, the label separator FULL STOP, U+002E (.) is used to break up the labels. But because that label separator does not travel with the wire format of the domain name, there is no way to encode a different, "internationalized" separator in IDNA2008.

Other protocols may include characters with similar special meaning within the protocol. Common characters for these purposes include FULL STOP, U+002E (.); COMMERCIAL AT, U+0040 (@); HYPHEN-MINUS, U+002D (-); SOLIDUS, U+002F (/); and LOW LINE, U+005F (_). The mere inclusion of such a character in the protocol is not enough for it to be considered similar to another protocol using the same character; instead, handling of the character must be taken into consideration as well.

An important issue to tackle here is whether it is valuable to map to or from these special characters as part of the Stringprep replacement. In some locales, the analogue to FULL STOP, U+002E is some other character, and users may expect to be able to substitute their normal stop for FULL STOP, U+002E. At the same time, there are predictability arguments in favour of treating names with FULL STOP, U+002E in them just the way they are treated under IDNA2008.

3.3. Where the data comes from and where it goes

3.3.1. User input and the source of protocol elements

Some protocol elements are provided by users, and others are not. Those that are not may presumably be subject to greater restrictions, whereas those that users provide likely need to permit the broadest range of code points. The following questions are helpful:

1. Do users input the strings directly?
2. If so, how? (keyboard, stylus, voice, copy-paste, etc.)
3. Where do we place the dividing line between user interface and protocol? (see [RFC5895])

3.3.2. User output

Just as only some protocol elements are expected to be entered directly by users, only some protocol elements are intended to be consumed directly by users. It is important to know how users are expected to be able to consume the protocol elements, because different environments present different challenges. An element that is only ever delivered as part of a vCard remains in machine-readable format, so the problem of visual confusion is not a great one. Is the protocol element published as part of a vCard, a web directory, on a business card, or on "the side of a bus"? Do users use the protocol element as an identifier (which means that they might enter it again in some other context)?

3.3.3. Operations

Some strings are useful as part of the protocol but are not used as input to other operations (for instance, purely informative or descriptive text). Other strings are used directly as input to other operations (such as cryptographic hash functions), or are used together with other strings to (such as concatenating a string with some others to form a unique identifier).

3.3.3.1. String classes

Strings often have a similar function in different protocols. For instance, many different protocols contain user identifiers or passwords. A single profile for all such uses might be desirable.

Often, a string in a protocol is effectively a protocol element from another protocol. For instance, different systems might use the same credentials database for authentication.

3.3.3.2. Community considerations

A Stringprep replacement that does anything more than just update Stringprep to the latest version of Unicode will probably entail some changes. It is important to identify the willingness of the protocol-using community to accept backwards-incompatible changes. By the same token, it is important to evaluate the desire of the community for features not available under Stringprep.

3.3.3.3. What to do about Unicode changes

IDNA2008 uses an algorithm to derive the validity of a Unicode code point for use under IDNA2008. It does this by using the properties of each code point to test its validity.

This approach depends crucially on the idea that code points, once valid for a protocol profile, will not later be made invalid. That is not a guarantee currently provided by Unicode. Properties of code points may change between versions of Unicode. Rarely, such a change could cause a given code point to become invalid under a protocol profile, even though the code point would be valid with an earlier version of Unicode. This is not merely a theoretical possibility, because it has occurred ([I-D.faltstrom-5892bis]).

Accordingly, a Stringprep replacement that intends to be Unicode version agnostic will need to work out a mechanism to address cases where incompatible changes occur because of new Unicode versions.

3.3.4. Some useful classes of strings

With the above considerations in hand, we can usefully classify strings into the following categories, inspired by those outlined in [I-D.saintandre-xmpp-i18n]:

Domainy strings Strings that are intended for use in a domain name slot, as defined in [RFC5890]. Note that domainy strings could be used outside a domain name slot: the question here is what the eventual intended use for the string is, and not whether the string is actually functioning as a domain name at any moment.

Namey strings Strings that are intended for use as identifiers but that are not domainy strings. Namey strings are normally public data within the protocol where they are used: these are intended as identifiers that can be passed around to identify something.

Secretish strings Strings that are intended for use as passwords or passphrases or other such type of token. Secretish strings are normally not public data within the protocol where they are used: they function as a token for authorization, and normally should not be shared publicly.

Protocolish strings Strings that are intended to be used by the protocol as free-form strings, but that have some significant handling within the protocol. For instance, a protocol slot that allows free-form text where case is not preserved would need to have case mapping rules applied; in this case, the string would be a protocolish string.

String blobs Elements of the protocol that look like strings to users, but that are passed around in the protocol unchanged and that cannot be used for comparison or other purposes. In effect, these are strings that are part of a protocol payload, and are not themselves part of the protocol at all.

4. Considerations for Stringprep replacement

The above suggests the following direction for the working group:

- o A stringprep replacement should be defined.
- o The replacement should take an approach similar to IDNA2008, in that it enables Unicode agility.
- o Protocols share similar characteristics of strings. Therefore, defining i18n preparation algorithms for a (small) set of string classes may be sufficient for most cases and provides the coherence among a set of protocol friends.
- o The sets of string classes need to be evaluated according to the considerations that make up the headings in Section 3
- o It is reasonable to limit scope to Unicode code points, and rule the mapping of data from other character encodings outside the scope of this effort.
- o Recommendations for handling protocol incompatibilities resulting from changes to Unicode are required.

Existing deployments already depend on Stringprep profiles. Therefore, the working group will need to consider the effects of any new strategy on existing deployments. By way of comparison, it is worth noting that some characters were acceptable in IDNA labels under IDNA2003, but are not protocol-valid under IDNA2008 (and conversely). Different implementers may make different decisions about what to do in such cases; this could have interoperability effects. The working group will need to trade better support for different linguistic environments against the potential side effects of backward incompatibility.

5. Security Considerations

This document merely states what problems are to be solved, and does not define a protocol. There are undoubtedly security implications of the particular results that will come from the work to be completed.

6. IANA Considerations

This document has no actions for IANA.

7. Discussion home for this draft

This document is intended to define the problem space discussed on the precis@ietf.org mailing list.

8. Acknowledgements

This document is the product of the PRECIS IETF Working Group, and participants in that Working Group were helpful in addressing issues with the text.

Specific contributions came from David Black, Alan DeKok, Bill McQuillan, Alexey Melnikov, Peter Saint-Andre, Dave Thaler, and Yoshiro Yoneya.

Dave Thaler provided the "buckets" insight in Section 3.1.1, central to the organization of the problem.

9. Informative References

[I-D.faltstrom-5892bis]
Faltstrom, P. and P. Hoffman, "The Unicode code points and IDNA - Unicode 6.0", draft-faltstrom-5892bis-05 (work in progress), June 2011.

[I-D.iab-identifier-comparison]
Thaler, D., "Issues in Identifier Comparison for Security Purposes", draft-iab-identifier-comparison-00 (work in progress), July 2011.

[I-D.saintandre-xmpp-i18n]
Saint-Andre, P., "Internationalized Addresses in XMPP", draft-saintandre-xmpp-i18n-03 (work in progress), March 2011.

[NEWPREP] "Newprep BoF Meeting Minutes", March 2010.

[RFC3454] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", RFC 3454, December 2002.

[RFC3490] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[RFC3491] Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)", RFC 3491, March 2003.

[RFC3492] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.

[RFC3530] Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M., and D. Noveck, "Network File System (NFS) version 4 Protocol", RFC 3530, April 2003.

[RFC3722] Bakke, M., "String Profile for Internet Small Computer Systems Interface (iSCSI) Names", RFC 3722, April 2004.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC3920] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 3920, October 2004.

[RFC3922] Saint-Andre, P., "Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)", RFC 3922, October 2004.

[RFC4011] Waldbusser, S., Saperia, J., and T. Hongal, "Policy Based Management MIB", RFC 4011, March 2005.

[RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", RFC 4013, February 2005.

[RFC4279] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[RFC4314] Melnikov, A., "IMAP4 Access Control List (ACL) Extension", RFC 4314, December 2005.

[RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[RFC4505] Zeilenga, K., "Anonymous Simple Authentication and Security Layer (SASL) Mechanism", RFC 4505, June 2006.

[RFC4511] Sermersheim, J., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, June 2006.

[RFC4513] Harrison, R., "Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms", RFC 4513, June 2006.

[RFC4518] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Internationalized String Preparation", RFC 4518, June 2006.

[RFC4616] Zeilenga, K., "The PLAIN Simple Authentication and Security Layer (SASL) Mechanism", RFC 4616, August 2006.

[RFC4643] Vinocur, J. and K. Murchison, "Network News Transfer Protocol (NNTP) Extension for Authentication", RFC 4643, October 2006.

[RFC4683] Park, J., Lee, J., Lee, H., Park, S., and T. Polk, "Internet X.509 Public Key Infrastructure Subject Identification Method (SIM)", RFC 4683, October 2006.

[RFC4690] Klensin, J., Faltstrom, P., Karp, C., and IAB, "Review and Recommendations for Internationalized Domain Names (IDNs)", RFC 4690, September 2006.

[RFC4790] Newman, C., Duerst, M., and A. Gulbrandsen, "Internet Application Protocol Collation Registry", RFC 4790, March 2007.

[RFC4954] Siemborski, R. and A. Melnikov, "SMTP Service Extension for Authentication", RFC 4954, July 2007.

[RFC5034] Siemborski, R. and A. Menon-Sen, "The Post Office Protocol (POP3) Simple Authentication and Security Layer (SASL) Authentication Mechanism", RFC 5034, July 2007.

[RFC5051] Crispin, M., "i;unicode-casemap - Simple Unicode Collation Algorithm", RFC 5051, October 2007.

[RFC5054] Taylor, D., Wu, T., Mavrogiannopoulos, N., and T. Perrin, "Using the Secure Remote Password (SRP) Protocol for TLS Authentication", RFC 5054, November 2007.

[RFC5122] Saint-Andre, P., "Internationalized Resource Identifiers (IRIs) and Uniform Resource Identifiers (URIs) for the Extensible Messaging and Presence Protocol (XMPP)", RFC 5122, February 2008.

[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.

[RFC5456] Spencer, M., Capouch, B., Guy, E., Miller, F., and K. Shumard, "IAX: Inter-Asterisk eXchange Version 2", RFC 5456, February 2010.

[RFC5661] Shepler, S., Eisler, M., and D. Noveck, "Network File

System (NFS) Version 4 Minor Version 1 Protocol", RFC 5661, January 2010.

[RFC5802] Newman, C., Menon-Sen, A., Melnikov, A., and N. Williams, "Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms", RFC 5802, July 2010.

[RFC5804] Melnikov, A. and T. Martin, "A Protocol for Remotely Managing Sieve Scripts", RFC 5804, July 2010.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[RFC5892] Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.

[RFC5893] Alvestrand, H. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.

[RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.

[RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, September 2010.

[UAX15] "Unicode Standard Annex #15: Unicode Normalization Forms", UAX 15, September 2009.

Appendix A. Protocols known to be using Stringprep

The known cases are here described in two ways. The types of identifiers the protocol uses is first called out in the ID type column (from Section 3.1.1), using the short forms "a" for Absolute, "d" for Definite, and "i" for Indefinite. Next, there is a column that contains an "i" if the protocol string comes from user input, an "o" if the protocol string becomes user-facing output, "b" if both are true, and "n" if neither is true. The remaining columns have an "x" if and only if the protocol uses that class, as described in Section 3.3.4. Values marked "-" indicate that an answer is not

useful; in this case, see detailed discussion in Appendix B.

RFC	IDtype	User?	Dom'y	Nam'y	Sec'ish	Pro'ish	Blob
3722	a	o		x	x	x	
3748	-	-	-	x	-	-	-
3920	a,d	b		x		x	
4314	a,d	b		x	x	x	

Table 1

[[anchor21: The table still needs to be filled in, I am aware.
--ajs@anvilwalrusden.com]]

Appendix B. Detailed discussion of protocols under consideration

Below are detailed reviews of the protocols under consideration (where such reviews are available). [[anchor22: These are to be cut and pasted from the wiki. --ajs@anvilwalrusden.com]]

Appendix C. Changes between versions

Note to RFC Editor: This section should be removed prior to publication.

C.1. 00

First WG version. Based on draft-blanchet-precis-problem-statement-00.

C.2. 01

- o Made clear that the document is talking only about Unicode code points, and not any particular encoding.
- o Substantially reorganized the document along the lines of the review template at <http://trac.tools.ietf.org/wg/precis/trac/wiki/StringprepReviewTemplate>.
- o Included specific questions for each topic for consideration.
- o Moved spot for individual protocol review to appendix. Not populated yet.

C.3. 02

- o Cleared up details of comparison classes
- o Added a section on changes in Unicode

C.4. 03

- o Aligned comparison discussion with identifier discussion from draft-iab-identifier-comparison-00
- o Added section on classes of strings ("Namey" and so on)

Authors' Addresses

Marc Blanchet
Viagenie
2600 boul. Laurier, suite 625
Quebec, QC G1V 4W1
Canada

Email: Marc.Blanchet@viagenie.ca
URI: http://viagenie.ca

Andrew Sullivan
519 Maitland St.
London, ON N6B 2Z5
Canada

Email: ajs@anvilwalrusden.com