

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: May 3, 2012

S. Sorce, Ed.  
Red Hat  
T. Yu, Ed.  
T. Hardjono, Ed.  
MIT Kerberos Consortium  
Oct 31, 2011

**A Generalized PAC for Kerberos V5  
draft-ietf-krb-wg-general-pac-01**

Abstract

This draft proposes a generalized authorization structure for the Kerberos V5 protocol. Such an authorization structure would allow for greater interoperability among directory services and other related Kerberos services across differing realms.

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 May 3, 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 . . . . .](#) [3](#)
- [2. Requirements Language . . . . .](#) [3](#)
- [3. Use-Case: Cross-Realm Directory Services . . . . .](#) [3](#)
- [4. A Generalized Authorization Structure for Kerberos V5 . . . . .](#) [4](#)
  - [4.1. Attributes . . . . .](#) [4](#)
  - [4.2. PAD-Realm . . . . .](#) [5](#)
  - [4.3. PAD-Principal . . . . .](#) [5](#)
  - [4.4. PAD-DNS-Domain . . . . .](#) [5](#)
  - [4.5. PAD-Short-Domain . . . . .](#) [5](#)
  - [4.6. PAD-UDID . . . . .](#) [6](#)
  - [4.7. PAD-Posix-Username . . . . .](#) [6](#)
  - [4.8. PAD-Posix-UID . . . . .](#) [6](#)
  - [4.9. PAD-Posix-GID . . . . .](#) [6](#)
  - [4.10. PAD-Posix-Gecos . . . . .](#) [6](#)
  - [4.11. PAD-Posix-Homedir . . . . .](#) [6](#)
  - [4.12. PAD-Posix-Shell . . . . .](#) [7](#)
  - [4.13. PAD-Fullname . . . . .](#) [7](#)
  - [4.14. PAD-AlternateNames . . . . .](#) [7](#)
  - [4.15. PAD-Groups . . . . .](#) [7](#)
  - [4.16. PAD Mapped Attributes . . . . .](#) [8](#)
  - [4.17. RFC2307 references for Directory Services backed KDCs . . . . .](#) [8](#)
    - [4.17.1. PAD-Posix-Username as 'uid' . . . . .](#) [8](#)
    - [4.17.2. PAD-Posix-UID as 'uidNumber' . . . . .](#) [9](#)
    - [4.17.3. PAD-Posix-GID as 'gidNumber' . . . . .](#) [9](#)
    - [4.17.4. PAD-Posix-Gecos as 'gecos' . . . . .](#) [9](#)
    - [4.17.5. PAD-Posix-Homedir as 'homeDirectory' . . . . .](#) [9](#)
    - [4.17.6. PAD-Posix-Shell as 'loginShell' . . . . .](#) [9](#)
- [5. Encoding . . . . .](#) [9](#)
  - [5.1. PAD Format . . . . .](#) [9](#)
- [6. Data Structures and Extensions . . . . .](#) [11](#)
  - [6.1. SignedPrincipalAuthorizationData . . . . .](#) [11](#)
  - [6.2. GSS-API Authenticator Extension . . . . .](#) [13](#)
- [7. Assigned numbers . . . . .](#) [14](#)
- [8. Timeouts Considerations . . . . .](#) [14](#)
- [9. IANA Considerations . . . . .](#) [15](#)
- [10. Security Considerations . . . . .](#) [15](#)
- [11. Acknowledgements . . . . .](#) [15](#)
- [12. References . . . . .](#) [15](#)
  - [12.1. Normative References . . . . .](#) [15](#)
  - [12.2. Informative References . . . . .](#) [16](#)
- [Appendix A. Additional Stuff . . . . .](#) [16](#)
- [Authors' Addresses . . . . .](#) [16](#)



## **1. Introduction**

There is an increasing need today for Kerberos to support the delivery and processing of authorization information pertaining to the principals seeking access to the servers. Kerberos today is used extensively for authentication to directory services within the Enterprise. In many cases, a directory service is implemented as a distributed database system organized across multiple realms. As such, when a client in one realm seeks access to a directory service component located within a different realm, information regarding both the identity of the client and the permissions associated with that client must be communicated across the realms. Currently there does not exist a common and standardized structure in Kerberos (V5) for conveying access control or authorization information.

This draft proposes a general authorization structure for Kerberos that identifies a base set of common data elements or fields within the authorization structure, as well as the format of that structure. We refer to this data structure as the Principal Authorization Data (PAD) structure in order to distinguish it from existing structures, such as the Privilege Attribute Certificate defined by Microsoft in [\[MS-PAC\]](#).

## **2. Requirements Language**

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](#)].

## **3. Use-Case: Cross-Realm Directory Services**

In this section we discuss one of the primary use-case scenarios for the Principal Authorization Data (PAD) structure within Kerberos V5. In this use-case a client principal is seeking to access a service in a different realm. Since the remote service does not have authorization information regarding the client, it needs to obtain it either from querying the directory service in its own realm or the directory service located in the client's realm. It is here that a common PAD structure becomes necessary and invaluable in order to achieve a high-degree of interoperability between directory services in distinct realms.

In this use-case a client principal C1 in realm R1 is seeking access to services (or servers) located in a different realm R2. In accessing local service S1 in realm R1 the client must first be authenticated by KDC1 in that realm. A directory service (e.g.



LDAP) called D1 is used in realm R1 to perform authorization of the client, after the client has been authenticated by KDC1.

When the client principal later seeks to access services or resources S2 in realm R2, following the usual Kerberos flow the client must first obtain a cross-realm TGT from KDC1 (in realm R1) and then present it to KDC2 (in realm R2) in order to obtain a service-ticket for S2. However, one immediate issue is the fact that service S2 does not have authorization information regarding the permissions or privileges of client C1 in realm R1. The service S2 could query its own directory service D2 to obtain authorization information pertaining to client C1. In the absence of such information in D2, the service S2 could then perform a cross-realm query to the directory services D1 operating in realm R1.

However, this cross-realm query from S2 to D1 is not only inefficient, but it also implies knowledge of multiple heterogeneous systems by all actors. Two different realms may rely on completely different infrastructures for user information storage, ranging from different LDAP implementations with different schema conventions to NIS, SQL databases, flat files, and so on. Every service in the realm R2 would have to know what information system is in use in R1, how to reach it, how to read and eventually how to map data from it. Moreover security related aspects on the authentication of S2 by the directory D1, the authorization of S2 to make such a query, the protection of responses from D1 to S2, and so on, would have to be addressed.

This use-case illustrates the need for a common PAD structure to address this cross-realm authorization problem. In particular, the PAD structure for the cross-realm access to remote services needs to be contained or carried within cross-realm TGTs and service-tickets. Such a PAD structure needs to carry enough authorization information such that a decision can be made by service S2 in realm R2 regarding the access request originating from the client principal C1 within realm R1.

## **4. A Generalized Authorization Structure for Kerberos V5**

### **4.1. Attributes**

The following attributes are defined in this document:

- o PAD-Realm
- o PAD-Principal



- o PAD-DNS-Domain
- o PAD-Short-Domain
- o PAD-UDID
- o PAD-Posix-Username
- o PAD-Posix-UID
- o PAD-Posix-GID
- o PAD-Posix-Gecos
- o PAD-Posix-Homedir
- o PAD-Posix-Shell
- o PAD-Fullname
- o PAD-AlternateNames
- o PAD-Groups

These are each defined and discussed further below.

#### **4.2. PAD-Realm**

The full Realm Name of the Realm the authorization information belongs to.

#### **4.3. PAD-Principal**

The name of the principal. Joined with the PAD-Realm component it MUST match the full principal name of the owner of the ticket.

#### **4.4. PAD-DNS-Domain**

The DNS Domain name associated to the Realm.

#### **4.5. PAD-Short-Domain**

A short domain name that uniquely identifies, within the set of trusted realms, the domain the principal belongs to. The short Domain name is useful for representation purposes in the OS. A KDC is allowed to change this field during validation. This may be done to resolve name conflicts in large trust relationships.





#### **4.6. PAD-UDID**

A UDID is a Unique Domain Identifier. Ideally it universally identifies the domain as the one the following local identifiers belongs to. This is used to differentiate between local identifiers belonging to different domains/realms.

The UDID size can be dependent on the specific Domain type and implementation. However it SHOULD be not less than 96 bits long so that chances of conflicts are relatively low. A 96 bit long identifier allows to construct a 128bit account identifier by concatenating the UDID to the local account Identifier (32bit quantity in POSIX).

For the purpose of this document the UDID is a completely opaque number and implementations SHOULD not try to perform any enforcement on the format of this number on receiving it.

#### **4.7. PAD-Posix-Username**

This is the user name that correspond to the kerberos principal, this is the name that SHOULD be used by the OS to represent the user. The OS may decide to prefix or suffix this name with the PAD-Domain or PAD-Realm names in case of name conflicts with local accounts.

#### **4.8. PAD-Posix-UID**

This is the UID Number associated to the user. This number is local to the domain identified by PAD-Domain-UUID.

#### **4.9. PAD-Posix-GID**

This is the Primary GID Number associated to the user. This number is local to the domain identified by PAD-Domain-UUID.

#### **4.10. PAD-Posix-Gecos**

The Gecos field for the User associated to the Principal if available. Can be omitted. If not available PAD-Fullname can be used instead.

#### **4.11. PAD-Posix-Homedir**

The home directory path relative to the local system, if available. If not available local defined defaults apply.



#### **4.12. PAD-Posix-Shell**

The default shell for the user, defined as the path of the binary relative to the local filesystem, if available. If not available local defined defaults apply.

#### **4.13. PAD-Fullname**

The full name of the user if available.

#### **4.14. PAD-AlternateNames**

Alternate names can be used by application to identify a user by means that differ from the user principal. Names are in string form and utf8 encoded [UTF-8]. In order to allow applications to recognize the name type without guesswork, alternate names are prefixed with a string followed by the colon ':' character and the name, without any space or other separation character. The following Alternate names are currently recognized: EMAIL, OS, OPENID, OAUTH It is allowed to include multiple alternate names of the same type. The order in which they are provided represent the priority within the same name type, if applications need to choose between names.

(TODO: need discussion on whether these needs labeled prefixes or explicit attributes for each alternate representation etc...)

#### **4.15. PAD-Groups**

This is a structured attribute and defines the groups the principal is member of.

The first value in the structure represents the domain UDID and is optional. If missing the domain UDID is assumed to be the one defined in the PAD-UDID attribute.

Then an array of values that define the groups as follows. Each group value includes 3 subvalues:

- o (1) Name: This is the name of the group.
- o (2) Type: Optional, type of group
- o (3) ID: group ID.

If the type is missing it is assumed that the group is of type "Posix Group" and the following ID is required and represents the gid number. The type is represented through a simplified OID like type where only 2 levels are defined. 0.0 Is reserved for posix groups, and the 0



prefix is reserved to official RFX use. Additional Prefixes can be assigned to organizations that request it for their purposes. Assignment TBD.

Multiple PAD-Groups attributes can be present at the same time. A trusting KDC can augment the original user's set of groups by adding a new PAD-Groups structure that contains groups local to the trusting domain. In this case the domain UDID is required. The domain UDID is used for gid number conflict resolution when the PAC is transmitted between services of different realms.

PAD-Groups are optional attributes and the KDC, upon PAC revalidation, may decide to remove the original attributes that do not belong to the KDC security domain in order to save space or to censor information to avoid disclosing data to services.

#### **4.16. PAD Mapped Attributes**

In POSIX, users and groups ID are not universally unique, and different Realms (even different machines within an authorization realm actually) may have overlapping and conflicting IDs. If this is the case, a trusting KDC may decide to re-map IDs coming from a foreign Realm to help services with uid/gid mapping and avoid ID conflicts that can lead to serious security issues. The original IDs are generally preserved.

If multiple PAD buffers are received and one of them contains a PAD-UDID that is recognized by the application to be the local security domain identifier, then only the mapped attributes in this buffer SHOULD be used for authorization purposes.

#### **4.17. [RFC2307](#) references for Directory Services backed KDCs**

A few attributes contain the keyword 'Posix' in their name. These attributes are usually represented by [RFC2307](#) in Directory Services. If the primary store for these attributes is a Directory the following equivalence with [RFC2307](#) defined attributes can be used.

##### **4.17.1. PAD-Posix-Username as 'uid'**

The PAD-Posix-Username is the User ID, and its syntax is equivalent to the attribute named 'uid' in [RFC 2307](#). This attribute is defined in [RFC 4519](#) (2.39). The attribute is defined as multivalued in [RFC 4519](#) but in this context only a single value is allowed. To define aliases refer to the attribute PAD-AlternateNames.



#### **4.17.2. PAD-Posix-UID as 'uidNumber'**

The PAD-Posix-UID is the User's Unique Identifier Number, and its syntax is equivalent to the attribute named 'uidNumber' in [RFC 2307](#).

#### **4.17.3. PAD-Posix-GID as 'gidNumber'**

The PAD-Posix-GID is the User's Primary Group Identifier Number, and its syntax is equivalent to the attribute named 'gidNumber' in [RFC 2307](#).

#### **4.17.4. PAD-Posix-Gecos as 'gecos'**

The PAD-Posix-Gecos is the User's Common Name, although, traditionally, this field has been used to convey additional information beyond the user's full name. Its syntax is equivalent to the attribute named 'gecos' in [RFC 2307](#).

#### **4.17.5. PAD-Posix-Homedir as 'homeDirectory'**

The PAD-Posix-Homedir is the User's LOCAL home directory. Its syntax is equivalent to the attribute named 'homeDirectory' in [RFC 2307](#).

#### **4.17.6. PAD-Posix-Shell as 'loginShell'**

The PAD-Posix-Shell is the User's preferred login shell. Its syntax is equivalent to the attribute named 'loginShell' in [RFC 2307](#).

## **5. Encoding**

The Kerberos protocol is defined in [[RFC4120](#)] using Abstract Syntax Notation One (ASN.1) [X680]. As such, this specification also uses the ASN.1 syntax for specifying both the abstract layout of the PAD attributes, as well as their encodings.

### **5.1. PAD Format**

The information carried in the PAD needs to be augmented by some control information and packaged in a way that makes it possible to devise future extensions.

Additional information needed to validate the PAD:

- o The expiration time (must be the same as the ticket expiration time).





- o The principal name (must be the same principal that owns the ticket).
- o The KDC signature (for re-validation purposes).
- o The Service Signature (in order to trust the PAD has not been tampered with).
- o Optional Trusted Service Key Signature (for use by trusted services on a host)
- o Optional PUBKEY KDC Signature

This information is needed to validate the PAD and make sure it is not modified, outdated, or contains information for a different principal.

In order to make the PAD extensible and at the same time always verifiable we propose that the PAD is embedded in a ASN.1 structure that can contain multiple optional buffers identified by numbers (how to assign numbers TBD).

Buffer number 0 is an ASN.1 structure that includes all attributes described in paragraph 4. This buffer is itself optional.

The whole structure with all its buffers is what is signed with the KDC and the service keys.

The final structure to be included in AD-IF-RELEVANT container and looks loosely like the following diagram.



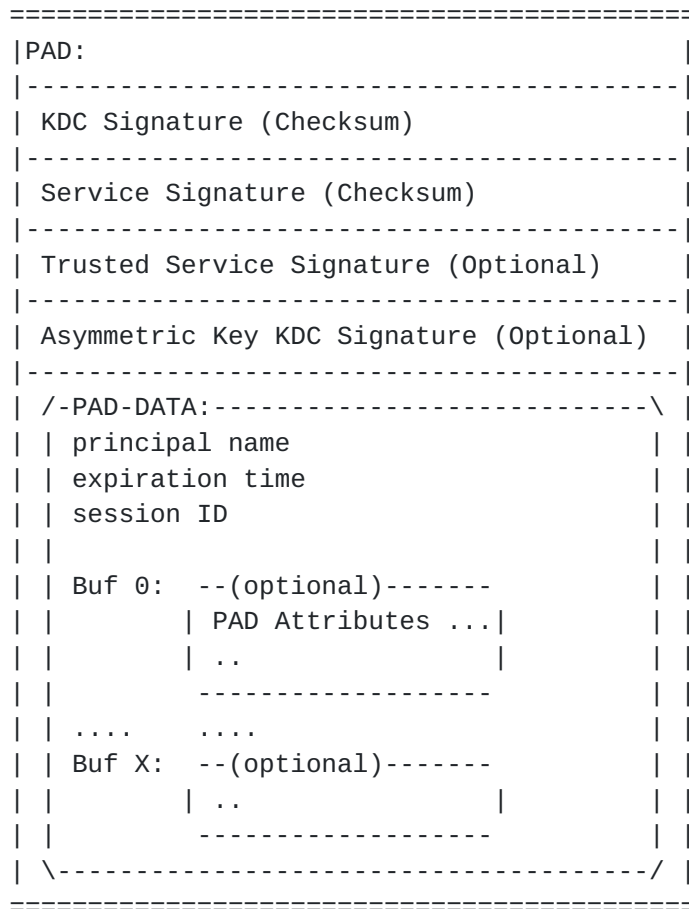


Figure 1: PAD Format

**6. Data Structures and Extensions**

**6.1. SignedPrincipalAuthorizationData**



```
AD-PAD ::= SEQUENCE {
    kdc-signature      [0] Checksum,
    svc-signature      [1] Checksum,
    trusted-svc-signature [2] PAD-OPT-Checksum OPTIONAL,
    pubkey-signature   [2] PAD-OPT-Checksum OPTIONAL,
    pad-data           [3] PAD-DATA
}
```

```
PAD-OPT-Checksum ::= SEQUENCE {
    Identifier [0] Name,
    Signature  [1] Checksum
}
```

#### kdc-signature

A cryptographic checksum computed over the encoding of the pad-data field, keyed with the krbtgt key.  
Checksum type TBD.

#### svc-signature

A cryptographic checksum computed over the encoding of the pad-data field, keyed with the service long term key.  
Checksum type TBD.

#### Trusted-svc-signature

A principal name and a cryptographic checksum computed over the encoding of the pad-data field, keyed with the long term key of the principal name specified in the Name field. Unless otherwise explicitly administratively configured, the key SHOULD be found by substituting the service name component of the principal name of the service with 'host'.

If the service is 'host' this checksum is redundant and can be omitted.

If the resulting host/<name>@REALM or the administratively configured service is not found in the KDC database this checksum can be omitted.

Checksum type TBD.

#### pubkey-signature

A name identifying the asymmetric key-pair used.

A checksum computed over the encoding of the pad-data field using the Private Key identified in the Name field.

If an asymmetric key is not available this checksum MUST be omitted.

Signature type TBD.



```
PAD-DATA ::=SEQUENCE {  
  p-realm      [0] Realm,  
  p-name       [1] PrincipalName,  
  expiration   [2] Date,  
  session-id   [3] TBD,  
  elements     [4] SEQUENCE OF AuthorizationData  
}
```

p-realm, p-name

The realm and name of the principal the authorization data elements apply to.

expiration

The Expiration Date of the Authorization Data. Normally this is the same as the original TGT expiration date.

session-id

A random number that uniquely ties any following ticket this PAD Data is associated to with the original TGT Released to the user

elements

A sequence of authorization data elements issued by the KDC.

The AD-PAD data is intended to provide a means for a Kerberos principal credentials to carry authorization data that the receiving service can use to perform authorization decisions.

The KDC signature is required to allow the KDC to validate the data without having to recompute the contents at every TGS request.

The SVC signature is required so that the Service can verify that the authorization data has been validated by the KDC.

Both the Trusted Service Checksum and the asymmetric KDC Signature are useful to verify the PAD authenticity on the same host, when the PAD is received by a less trusted service and passed to a more trusted service on the same host without the need for additional roundtrips to the KDC.

The ad-type for AD-SIGNED-PAD is (TBD).

## **6.2. GSS-API Authenticator Extension**

The Authenticator Checksum as defined in [RFC 4121](#) limit the size of delegated credentials in the KRB\_CRED message to a size of 64KiB.





In order to be able to transfer larger messages an extension is defined. This extension is used instead of the Dlight/Deleg fields, and the Dlight and Deleg fields MUST not be included when this extension is appended to the authenticator.

The extension SHALL have the following format which is drafted according to [[draft-ietf-krb-wg-gss-cb-hash-agility](#)]:

| Octet | Name   | Description   |
|-------|--------|---|
| 0..3  | ExtN   | A 16bit value identifying the extension. Represented in big-endian order; Contains the hex value 0XXXXXXXX. |
| 4..7  | Length | The length of the Extended Delegation field. Represented in big-endian order;                               |
| 8..N  | Data   | A KRB_CRED message (N = Length + 8)   |

A new flag GSS\_C\_EXT\_DELEG\_FLAG with Value X is also defined. This flag is used instead of GSS\_C\_DELEG\_FLAG when the delegated credentials are larger than 64KiB and cannot fit in the standard Deleg field.

Implementors SHOULD use this Extensions and this flag only if the KRB\_CRED message is larger than 64KiB and use the standard Deleg field otherwise.

## 7. Assigned numbers

TBD

## 8. Timeouts Considerations

Current implementations depend on very strict timeouts on obtaining AS Replies. In popular implementations the client will timeout if it doesn't receive a reply within 1 second. Adding authorization data may involve lookups to external (to the KDC) data sources. Implementors should consider whether the current timeout is still reasonable in light of the additional processing KDCs may be required to do.



## **9. IANA Considerations**

TBD.

## **10. Security Considerations**

Although it is anticipated that the PAD structure itself will be carried within a ticket and thereby protected using the existing encryption methods on that ticket, there are a number of issues that have bearings on the security of the entire Kerberos realm as a whole. Some of these issues are as follows:

- o UID and GID Collisions: There is always the possibility of collision of numbers representing a UID and a GID. This problem can be remedied to a large degree by realms using an appropriate range selection policy and algorithms.
- o When collisions are detected the KDC or, alternatively, the receiving Service MUST be able to remap IDs so that they do not conflict with locally defined IDs
- o Transit-domain issues: The PAC must be signed by the KDC that is attaching it to a ticket with 2 different signatures. The service signature so that the service can verify its KDC validated the contents. The KDC signature, so that the OS can ask the KDC to confirm the PAD has not been modified by a less trusted service. An optional asymmetric key signature is also allowed if Keys are available in order to avoid additional roundtrips. For cross-realm tickets the "service" signature is made with the cross-realm key. When a KDC receives a PAD it is allowed to modify it in any way. It can filter out information or add information (like group memberships defined locally). A KDC may also decide to change information in different ways depending on what service it is targeted to.

## **11. Acknowledgements**

TBD.

## **12. References**

### **12.1. Normative References**

- [RFC3961] Raeburn, K., "Encryption and Checksum Specifications for Kerberos 5", [RFC 3961](#), February 2005.



- [RFC3962] Raeburn, K., "Advanced Encryption Standard (AES) Encryption for Kerberos 5", [RFC 3962](#), February 2005.
- [RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", [RFC 4120](#), July 2005.

## **12.2. Informative References**

- [MIT-Athena]  
Steiner, J., Neuman, B., and J. Schiller, "Kerberos: An Authentication Service for Open Network Systems. In Proceedings of the Winter 1988 Usenix Conference. February.", 1988.
- [MS-PAC] Microsoft, "Microsoft MS-PAC: Privilege Attribute Certificate Data Structure (v20100711)", July 2010.
- [POSIX] The Open Group, "Portable Operating System Interface (POSIX.1-2008)", 2008.
- [RFC1510] Kohl, J. and B. Neuman, "The Kerberos Network Authentication Service (V5)", [RFC 1510](#), September 1993.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2307] Howard, L., "An Approach for Using LDAP as a Network Information Service", [RFC 2307](#), March 1998.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", [BCP 72](#), [RFC 3552](#), July 2003.
- [X.690] ISO, "ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) - ITU-T Recommendation X.690 (ISO/IEC International Standard 8825-1:1998)", 1997.

## **Appendix A. Additional Stuff**

This becomes an Appendix.



Authors' Addresses

Simo Sorce (editor)  
Red Hat

Email: [ssorce@redhat.com](mailto:ssorce@redhat.com)

Tom Yu (editor)  
MIT Kerberos Consortium

Email: [tlyu@mit.edu](mailto:tlyu@mit.edu)

Thomas Hardjono (editor)  
MIT Kerberos Consortium

Email: [hardjono@mit.edu](mailto:hardjono@mit.edu)



