

**Extended Generic Security Service Mechanism Inquiry APIs**  
**draft-ietf-kitten-extended-mech-inquiry-01.txt**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 19, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document introduces new application programming interfaces (APIs) to the Generic Security Services API (GSS-API) for extended mechanism attribute inquiry. These interfaces are primarily intended for use in mechanism composition, but also to reduce instances of hardcoding of mechanism identifiers in GSS applications.

These interfaces include: mechanism attributes and attribute sets, a function for inquiring the attributes of a mechanism, a function for indicating mechanisms that possess given attributes, and a function

for displaying mechanism attributes.

## Table of Contents

<a href="#">1.</a>	Conventions used in this document . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">3.</a>	New GSS-API Interfaces . . . . .	<a href="#">3</a>
<a href="#">3.1.</a>	Mechanism Attributes and Attribute Sets . . . . .	<a href="#">3</a>
<a href="#">3.2.</a>	Determination of Attribute Sets of Composite Mechs . . . . .	<a href="#">4</a>
<a href="#">3.3.</a>	List of Known Mechanism Attributes . . . . .	<a href="#">4</a>
<a href="#">3.4.</a>	Mechanism Attribute Sets of Existing Mechs . . . . .	<a href="#">6</a>
<a href="#">3.5.</a>	New GSS-API Function Interfaces . . . . .	<a href="#">8</a>
<a href="#">3.5.1.</a>	GSS_Indicate_mechs_by_attr() . . . . .	<a href="#">8</a>
<a href="#">3.5.2.</a>	GSS_Inquire_attrs_for_mech() . . . . .	<a href="#">9</a>
<a href="#">3.5.3.</a>	GSS_Display_mech_attr() . . . . .	<a href="#">10</a>
<a href="#">3.5.4.</a>	New Major Status Values . . . . .	<a href="#">10</a>
<a href="#">3.5.5.</a>	C-Bindings . . . . .	<a href="#">10</a>
<a href="#">4.</a>	Requirements for Mechanism Designers . . . . .	<a href="#">11</a>
<a href="#">5.</a>	IANA Considerations . . . . .	<a href="#">11</a>
<a href="#">6.</a>	Security considerations . . . . .	<a href="#">11</a>
<a href="#">7.</a>	References . . . . .	<a href="#">12</a>
<a href="#">7.1.</a>	Normative . . . . .	<a href="#">12</a>
<a href="#">7.2.</a>	Normative . . . . .	<a href="#">12</a>
	Author's Address . . . . .	<a href="#">13</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">14</a>



## **1. Conventions used in this document**

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 [[RFC2119](#)].

## **2. Introduction**

## **3. New GSS-API Interfaces**

GSS-API applications face, today, the problem of how to select from multiple GSS-API mechanisms that may be available. For example, applications that support mechanism negotiation directly often have to be careful not to use SPNEGO to avoid two-layer mechanism negotiation, but since SPNEGO may be indicated by `GSS_Indicate_mechs()` and since there's no way to know that a mechanism negotiates mechanisms other than to hardcode the OIDs of such mechanisms, such applications must hardcode the SPNEGO OID. This problem is likely to be exacerbated by the introduction of composite mechanisms.

To address this problem we introduce a new concept: that of mechanism attributes. By allowing applications to query the set of attributes associated with individual mechanisms and to find out which mechanisms support a given set of attributes we allow applications to select mechanisms based on their attributes yet without having to hardcode mechanism OIDs.

[Section 3.1](#) describes the mechanism attributes concept. Sections 3.5.1, 3.5.2 and 3.5.3 describe three new interfaces that deal in mechanisms and attribute sets:

- o `GSS_Indicate_mechs_by_attrs()`
- o `GSS_Inquire_attrs_for_mech()`
- o `GSS_Display_mech_attr()`

### **3.1. Mechanism Attributes and Attribute Sets**

An abstraction for the features provided by pseudo-mechanisms is needed in order to facilitate the programmatic selection of mechanisms as well as for the programmatic composition of mechanisms.

Two data types are needed: one for individual mechanism attributes and one for mechanism attribute sets. To simplify the mechanism attributes interfaces we reuse the 'OID' and 'OID set' data types and model individual mechanism attribute types as OIDs.

Williams

Expires April 19, 2006

[Page 3]

To this end we define an open namespace of mechanism attributes and assign them arcs off of this OID:

<TBD> [1.3.6.1.5.5.12 appears to be available, registration w/ IANA TBD]

### **3.2. Determination of Attribute Sets of Composite Mechs**

Each mechanism, composite or otherwise, has a set of mechanism attributes that it supports as specified.

The mechanism attribute set of a composite mechanism is to be determined by the top-most stackable pseudo-mechanism of the composite according to its own attribute set and that of the remainder of the composite mechanism stack below it.

It may well be that some composite mechanisms' attribute sets consist of the union of those of their every component, however this need not be the case and MUST NOT be assumed.

Every stackable pseudo-mechanism's specification MUST specify the rules for determining the mechanism attribute set of mechanisms composed by it.

### **3.3. List of Known Mechanism Attributes**

Mech Attr Name	OID	Arc Name
	Arc	
GSS_C_MA_MECH_CONCRETE	(1)	concrete-mech
GSS_C_MA_MECH_STACKABLE	(2)	pseudo-mech
GSS_C_MA_MECH_COMPOSITE	(3)	composite-mech
GSS_C_MA_MECH_NEGO	(4)	mech-negotiation-mech
GSS_C_MA_MECH_GLUE	(5)	mech-glue
GSS_C_MA_NOT_MECH	(6)	not-mech
GSS_C_MA_DEPRECATED	(7)	mech-deprecated
GSS_C_MA_NOT_DFLT_MECH	(8)	mech-not-default
GSS_C_MA_ITOK_FRAMED	(9)	initial-is-framed
GSS_C_MA_AUTH_INIT	(10)	auth-init-princ
GSS_C_MA_AUTH_TARG	(11)	auth-targ-princ
GSS_C_MA_AUTH_INIT_INIT	(12)	auth-init-princ-initial
GSS_C_MA_AUTH_TARG_INIT	(13)	auth-targ-princ-initial
GSS_C_MA_AUTH_INIT_ANON	(14)	auth-init-princ-anon
GSS_C_MA_AUTH_TARG_ANON	(15)	auth-targ-princ-anon
GSS_C_MA_DELEG_CRED	(16)	deleg-cred
GSS_C_MA_INTEG_PROT	(17)	integ-prot
GSS_C_MA_CONF_PROT	(18)	conf-prot



GSS_C_MA_MIC	(19)	mic
GSS_C_MA_WRAP	(20)	wap
GSS_C_MA_PROT_READY	(21)	prot-ready
GSS_C_MA_REPLAY_DET	(22)	replay-detection
GSS_C_MA_OOS_DET	(23)	oos-detection
GSS_C_MA_CBINDINGS	(24)	channel-bindings
GSS_C_MA_CBINDINGS_BIDI	(25)	channel-bindings-bidirectional
GSS_C_MA_CBINDINGS_NEGO	(26)	channel-bindings-negotiate
GSS_C_MA_PFS	(27)	pfs
GSS_C_MA_COMPRESS	(28)	compress
GSS_C_MA_CTX_TRANS	(29)	context-transfer
<reserved>	(30..)	

Table 1

Mech Attr Name	Purpose
GSS_C_MA_MECH_CONCRETE	Indicates that a mech is neither a pseudo- mechanism nor a composite mechanism.
GSS_C_MA_MECH_STACKABLE	Indicates that a mech is a pseudo-mechanism.
GSS_C_MA_MECH_COMPOSITE	Indicates that a mech is a composite mechanism.
GSS_C_MA_MECH_NEGO	Indicates that a mech negotiates other mechs (e.g., SPNEGO has this attribute).
GSS_C_MA_MECH_GLUE	Indicates that the OID is not for a mechanism but for the GSS-API itself.
GSS_C_MA_NOT_MECH	Indicates that the OID is known, yet also known not to be the OID of any GSS-API mechanism (or the GSS-API itself).
GSS_C_MA_DEPRECATED	Indicates that a mech (or its OID) is deprecated and MUST NOT be used as a default mechanism.
GSS_C_MA_NOT_DFLT_MECH	Indicates that a mech (or its OID) MUST NOT be used as a default mechanism.
GSS_C_MA_ITOK_FRAMED	Indicates that the given mechanism's initial context tokens are properly framed as per- <a href="#">section 3.1 of rfc2743</a> .
GSS_C_MA_AUTH_INIT	Indicates support for authentication of initiator to acceptor.
GSS_C_MA_AUTH_TARG	Indicates support for authentication of acceptor to initiator.
GSS_C_MA_AUTH_INIT_INIT	Indicates support for initial





	authentication of initiator to acceptor.
GSS_C_MA_AUTH_TARG_INIT	Indicates support for initial authentication of acceptor to initiator.
GSS_C_MA_AUTH_INIT_ANON	Indicates support for initiator anonymity.
GSS_C_MA_AUTH_TARG_ANON	Indicates support for acceptor anonymity.
GSS_C_MA_DELEG_CRED	Indicates support for credential delegation.
GSS_C_MA_INTEG_PROT	Indicates support for per-message integrity protection.
GSS_C_MA_CONF_PROT	Indicates support for per-message confidentiality protection.
GSS_C_MA_MIC	Indicates support for MIC tokens.
GSS_C_MA_WRAP	Indicates support for WRAP tokens.
GSS_C_MA_PROT_READY	Indicates support for per-message protection prior to full context establishment.
GSS_C_MA_REPLAY_DET	Indicates support for replay detection.
GSS_C_MA_OOS_DET	Indicates support for out-of-sequence detection.
GSS_C_MA_CBINDINGS	Indicates support for channel bindings.
GSS_C_MA_CBINDINGS_BIDI	Indicates that acceptors unconditionally indicate to initiators whether their channel bindings were matched the acceptors', even when the acceptor applications use GSS_C_NO_CHANNEL_BINDINGS..
GSS_C_MA_CBINDINGS_NEGO	Indicates that the mech acts as a signal for application support for and willingness to use channel bindings.
GSS_C_MA_PFS	Indicates support for Perfect Forward Security.
GSS_C_MA_COMPRESS	Indicates support for compression of data inputs to GSS_Wrap().
GSS_C_MA_CTX_TRANS	Indicates support for security context export/import.

Table 2

### 3.4. Mechanism Attribute Sets of Existing Mechs

The Kerberos V mechanism [[RFC1964](#)] [CFX] provides the following mechanism attributes:

Williams

Expires April 19, 2006

[Page 6]

- o GSS\_C\_MA\_MECH\_CONCRETE
- o GSS\_C\_MA\_ITOK\_FRAMED
- o GSS\_C\_MA\_AUTH\_INIT
- o GSS\_C\_MA\_AUTH\_TARG
- o GSS\_C\_MA\_DELEG\_CRED
- o GSS\_C\_MA\_INTEG\_PROT
- o GSS\_C\_MA\_CONF\_PROT
- o GSS\_C\_MA\_MIC
- o GSS\_C\_MA\_WRAP
- o GSS\_C\_MA\_PROT\_READY
- o GSS\_C\_MA\_REPLAY\_DET
- o GSS\_C\_MA\_OOS\_DET
- o GSS\_C\_MA\_CBINDINGS
- o GSS\_C\_MA\_CTX\_TRANS (some implementations, using implementation-specific exported context token formats)

The Kerberos V mechanism also has a deprecated OID which has the same mechanism attributes as above, and GSS\_C\_MA\_DEPRECATED.

[The mechanism attributes of the SPKM family of mechanisms will be provided in a separate document as SPKM is current being reviewed for possibly significant changes due to problems in its specifications.]

The LIPKEY mechanism offers the following attributes:

- o GSS\_C\_MA\_MECH\_CONCRETE (should be stackable, but does not compose)
- o GSS\_C\_MA\_ITOK\_FRAMED
- o GSS\_C\_MA\_AUTH\_INIT\_INIT
- o GSS\_C\_MA\_AUTH\_TARG (from SPKM-3)
- o GSS\_C\_MA\_AUTH\_TARG\_ANON (from SPKM-3)
- o GSS\_C\_MA\_INTEG\_PROT
- o GSS\_C\_MA\_CONF\_PROT
- o GSS\_C\_MA\_REPLAY\_DET
- o GSS\_C\_MA\_OOS\_DET
- o GSS\_C\_MA\_CTX\_TRANS (some implementations, using implementation-specific exported context token formats)

(LIPKEY should also provide GSS\_C\_MA\_CBINDINGS, but SPKM-3 requires clarifications on this point.)

The SPNEGO mechanism [SPNEGO] provides the following attributes:

- o GSS\_C\_MA\_MECH\_NEGO
- o GSS\_C\_MA\_ITOK\_FRAMED

The attributes of mechanisms negotiated by SPNEGO are not modified by the use of SPNEGO.

All other mechanisms' attributes will be described elsewhere.



### **3.5. New GSS-API Function Interfaces**

Several new interfaces are given by which, for example, GSS-API applications may determine what features are provided by a given mechanism, what mechanisms provide what features and what compositions are legal.

These new interfaces are all OPTIONAL.

In order to preserve backwards compatibility with applications that do not use the new interfaces `GSS_Indicate_mechs()` MUST NOT indicate support for any stackable pseudo-mechanisms. `GSS_Indicate_mechs()` MAY indicate support for some, all or none of the available composite mechanisms; which composite mechanisms, if any, are indicated through `GSS_Indicate_mechs()` SHOULD be configurable. `GSS_Acquire_cred()` and `GSS_Add_cred()` MUST NOT create credentials for composite mechanisms not explicitly requested or, if no desired mechanism or mechanisms are given, for composite mechanisms not indicated by `GSS_Indicate_mechs()`.

Applications SHOULD use `GSS_Indicate_mechs_by_attr()` instead of `GSS_Indicate_mechs()` wherever possible.

Applications can use `GSS_Indicate_mechs_by_attr()` to determine what, if any, mechanisms provide a given set of features.

`GSS_Indicate_mechs_by_attr()` can also be used to indicate (as in `GSS_Indicate_mechs()`) the set of available mechanisms of each type (concrete, mechanism negotiation pseudo-mechanism, stackable pseudo-mechanism and composite mechanisms).

Applications may use `GSS_Inquire_attrs_for_mech()` to test whether a given composite mechanism is available and the set of features that it offers.

#### **3.5.1. `GSS_Indicate_mechs_by_attr()`**

Inputs:

- o `desired_mech_attrs` SET OF OBJECT IDENTIFIER -- set of `GSS_C_MA_*` OIDs that the mechanisms indicated in the `mechs` output parameter MUST offer.
- o `except_mech_attrs` SET OF OBJECT IDENTIFIER -- set of `GSS_C_MA_*` OIDs that the mechanisms indicated in the `mechs` output parameter MUST NOT offer.

Outputs:

- o `major_status` INTEGER,
- o `minor_status` INTEGER,



- o mechs SET OF OBJECT IDENTIFIER -- set of mechanisms that support -- the desired\_mech\_attrs but not the except\_mech\_attrs.

Return major\_status codes:

- o GSS\_S\_COMPLETE indicates success; the output mechs parameter MAY be the empty set (GSS\_C\_NO\_OID\_SET).
- o GSS\_BAD\_MECH\_ATTR indicates that at least one mechanism attribute OID in desired\_mech\_attrs or except\_mech\_attrs is unknown to the implementation.
- o GSS\_S\_FAILURE indicates that the request failed for some other reason.

GSS\_Indicate\_mechs\_by\_mech\_attrs() returns the set of mechanism OIDs that offer at least the desired\_mech\_attrs but none of the except\_mech\_attrs.

When desired\_mech\_attrs and except\_mech\_attrs are the empty set this function acts as a version of GSS\_indicate\_mechs() that outputs the set of all supported mechanisms of all types. By setting the desired\_mechs input parameter to a set of a single GSS\_C\_MA\_MECH\* feature applications can obtain the list of all supported mechanisms of a given type (concrete, stackable, etc...).

### **3.5.2. GSS\_Inquire\_attrs\_for\_mech()**

Inputs:

- o mech OBJECT IDENTIFIER -- mechanism OID

Outputs:

- o major\_status INTEGER,
- o minor\_status INTEGER,
- o mech\_attrs SET OF OBJECT IDENTIFIER -- set of mech\_attrs OIDs (GSS\_C\_MA\_\*)

Return major\_status codes:

- o GSS\_S\_COMPLETE indicates success; the output mech\_attrs parameter MAY be the empty set (GSS\_C\_NO\_OID\_SET).
- o GSS\_S\_BAD\_MECH indicates that the mechanism named by the mech parameter does not exist or that mech is GSS\_C\_NO\_OID and no default mechanism could be determined.
- o GSS\_S\_FAILURE indicates that the request failed for some other reason.

GSS\_Inquire\_mech\_attrs\_for\_mech() indicates the set of mechanism attributes supported by a given mechanism.

Because the mechanism attribute sets of composite mechanisms need not be the union of their components', when called to obtain the feature





set of a composite mechanism `GSS_Inquire_mech_attrs_for_mech()` obtains it by querying the mechanism at the top of the stack. See [Section 3.1](#).

### **3.5.3. `GSS_Display_mech_attr()`**

Inputs:

- o `mech_attr` OBJECT IDENTIFIER -- mechanism attribute OID

Outputs:

- o `major_status` INTEGER,
- o `minor_status` INTEGER,
- o `name` OCTET STRING, -- name of mechanism attribute (e.g., `GSS_C_MA_*`)
- o `short_desc` OCTET STRING, -- a short description of the mechanism attribute
- o `long_desc` OCTET STRING -- a longer description of the mechanism attribute

Return `major_status` codes:

- o `GSS_S_COMPLETE` indicates success.
- o `GSS_S_BAD_MECH_ATTR` indicates that the mechanism attribute referenced by the `mech_attr` parameter is unknown to the implementation.
- o `GSS_S_FAILURE` indicates that the request failed for some other reason.

This function can be used to obtain human-readable descriptions of GSS-API mechanism attributes.

### **3.5.4. New Major Status Values**

A single new major status code is added for `GSS_Display_mech_attr()`:

- o `GSS_S_BAD_MECH_ATTR` roughly corresponding to `GSS_S_BAD_MECH`, but applicable to mechanism attribute OIDs, rather than to mechanism OIDs.

For the C-bindings `GSS_S_BAD_MECH_ATTR` shall have a routine error number of 19 (this is shifted to the left by `GSS_C_ROUTINE_ERROR_OFFSET`).

### **3.5.5. C-Bindings**

```
#define GSS_S_BAD_MECH_ATTR (19u1 << GSS_C_ROUTINE_ERROR_OFFSET)
```

```
OM_uint32 gss_inquire_mechs_for_mech_attrs(  
    OM_uint32          *minor_status,
```



```
const gss_OID_set desired_mech_attrs,  
gss_OID_set      *mechs);  
  
OM_uint32 gss_inquire_mech_attrs_for_mech(  
    OM_uint32      *minor_status,  
    const gss_OID  mech,  
    gss_OID_set    *mech_attrs);  
  
OM_uint32 gss_display_mech_attr(  
    OM_uint32      *minor_status,  
    const gss_OID  mech_attr,  
    gss_buffer_t   name,  
    gss_buffer_t   short_desc,  
    gss_buffer_t   long_desc);
```

Figure 1

#### **4. Requirements for Mechanism Designers**

Stackable pseudo-mechanisms specifications MUST:

- o list the set of GSS-API mechanism attributes associated with them
- o list their initial mechanism composition rules
- o specify a mechanism for updating their mechanism composition rules

All other mechanism specifications MUST:

- o list the set of GSS-API mechanism attributes associated with them

#### **5. IANA Considerations**

The namespace of programming language symbols with names beginning with GSS\_C\_MA\_\* is reserved for allocation by the IANA.

Allocation of arcs in the namespace of OIDs relative to the base mechanism attribute OID specified in [Section 3.1](#) is reserved to the IANA.

#### **6. Security considerations**

...

#### **7. References**

##### **[7.1. Normative](#)**



- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", [RFC 2743](#), January 2000.
- [RFC2744] Wray, J., "Generic Security Service API Version 2 : C-bindings", [RFC 2744](#), January 2000.

## **[7.2.](#) Normative**

- [RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", [RFC 1964](#), June 1996.
- [RFC2478] Baize, E. and D. Pinkas, "The Simple and Protected GSS-API Negotiation Mechanism", [RFC 2478](#), December 1998.



Author's Address

Nicolas Williams  
Sun Microsystems  
5300 Riata Trace Ct  
Austin, TX 78727  
US

Email: [Nicolas.Williams@sun.com](mailto:Nicolas.Williams@sun.com)



## Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## 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

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

## Acknowledgment

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

