

NAME

gss_get_mic, **gss_sign** - Calculate a cryptographic message integrity code (MIC) for a message; integrity service

SYNOPSIS

```
#include <gssapi/gssapi.h>
```

OM_uint32

```
gss_get_mic(OM_uint32 *minor_status, const gss_ctx_id_t context_handle, gss_qop_t qop_req,  
            const gss_buffer_t message_buffer, gss_buffer_t msg_token);
```

OM_uint32

```
gss_sign(OM_uint32 *minor_status, const gss_ctx_id_t context_handle, gss_qop_t qop_req,  
         gss_buffer_t message_buffer, gss_buffer_t msg_token);
```

DESCRIPTION

Generates a cryptographic MIC for the supplied message, and places the MIC in a token for transfer to the peer application. The *qop_req* parameter allows a choice between several cryptographic algorithms, if supported by the chosen mechanism.

Since some application-level protocols may wish to use tokens emitted by **gss_wrap()** to provide "secure framing", implementations must support derivation of MICs from zero-length messages.

The **gss_sign()** routine is an obsolete variant of **gss_get_mic()**. It is provided for backwards compatibility with applications using the GSS-API V1 interface. A distinct entrypoint (as opposed to #define) is provided, both to allow GSS-API V1 applications to link and to retain the slight parameter type differences between the obsolete versions of this routine and its current form.

PARAMETERS

minor_status Mechanism specific status code.

context_handle Identifies the context on which the message will be sent.

qop_req Specifies requested quality of protection. Callers are encouraged, on portability grounds, to accept the default quality of protection offered by the chosen mechanism, which may be requested by specifying `GSS_C_QOP_DEFAULT` for this parameter. If an unsupported protection strength is requested, **gss_get_mic()** will return a *major_status* of `GSS_S_BAD_QOP`.

message_buffer Message to be protected.

`msg_token` Buffer to receive token. The application must free storage associated with this buffer after use with a call to `gss_release_buffer()`.

RETURN VALUES

`GSS_S_COMPLETE` Successful completion

`GSS_S_CONTEXT_EXPIRED` The context has already expired

`GSS_S_NO_CONTEXT` The `context_handle` parameter did not identify a valid context

`GSS_S_BAD_QOP` The specified QOP is not supported by the mechanism

SEE ALSO

`gss_release_buffer(3)`, `gss_wrap(3)`

STANDARDS

RFC 2743 Generic Security Service Application Program Interface Version 2, Update 1

RFC 2744 Generic Security Service API Version 2 : C-bindings

HISTORY

The `gss_get_mic` function first appeared in FreeBSD 7.0.

AUTHORS

John Wray, Iris Associates

COPYRIGHT

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.