Grid Security Infrastructure Message Specification

Edited by Von Welch (vwelch@ncsa.uiuc.edu)

Version 2: February 9th, 2004

Abstract

This document provides a description of the messages created and exchanged by the Globus Toolkit Grid Security Infrastructure (GSI). I t is intended for implementers and those who need to understand its messages in detail.

1 Introduction

This document describes the order and content of the messages generated by the Globus Toolkit Grid Security Infrastructure (GSI) [GSI] as supplied in version 3.0.x of the Globus Toolkit (GT3). This document does not attempt to define the full content of all these messages since their content is based on messages defined by the Secure Socket Layer version 3 (SSLv3), instead it defines only the differences and extensions that GSI makes to SSLv3.

Depending on the application using GSI, the messages specified in this document may be carried directly over a TCP connection (and look very much like SSLv3) or they may be carried in higher level protocols, such as GridFTP ADAT packages [ADAT] or SOAP messages as done by GT3 [GSI-SecConv]. This document does not attempt to define full protocol specifications for all these scenarios, simply the flow and contents of the GSI messages carried in those protocols. Readers are directed to the referenced documents to see how GSI messages are used in those contexts.

Section 2 covers prerequisite reading for this document. Section 3 covers terminology used in this document. Section 4 contains the description of the contents of the GSI messages. Section 5 contains a brief description of the early (pre-standard) version of the Proxy Certificates as used in version 2 of the Globus Toolkit. Section 7 contains references.

2 Prerequisites

It is assume that the reader is familiar with the following technologies:

- Public Key Infrastructure and X.509 Certificates [RFC 3280]: GSI bases its credentials on X.509 certificates.
- Secure Socket Layer version 3 (SSLv3) [SSL] protocol: GSI is built on SSLv3 and the GSI messages are described by reference to SSLv3 in a number of places in this document.
- Generic Security Services API (GSS-API) [GSSAPI]. GSI is accessed via the GSS-API and understanding GSS-API will help a reader understand how GSI functions. Note however that the messages generated by GSI do not fully comply with the GSSAPI specification in that they lack an identifying preamble.

- X.509 Proxy Certificates [Proxies]: Proxy certificates are used as an extension to X.509 End Entity Certificates by GSI to support delegation and single sign-on.

3 Terminology

The following terms are used freely in this document:

- *Client, Server*: The terms *client* and *server* are used throughout this document to describe the two parties exchanging GSI messages. The client is defined as the party which initiates the GSI message exchange. This is usually, but not always, the party which initiates the underlying network connection over which the GSI messages are being exchanged.
- *End Entity Certificate chain (EEC chain)*: A standard certificate chain for a user or server as defined by RFC 3280 [RFC3280].
- *GT2:* Globus Toolkit version 2.x
- *GT3:* Globus Toolkit version 3.0.x
- *GT2 Proxy Certificate Chain*: A X.509 certificate chain with one or more pre-specification proxy certificates as described in Section 5. These were predominately used in GT2, but are still supported in GT3.
- *Internet Draft Proxy Certificate Chain:* A X.509 certificate chain with one or more Proxy Certificates as defined by [Proxies].
- *Proxy Certificate Chain:* When used without qualification, this term means either a GT2 Proxy Certificate chain or an Internet Draft Proxy Certificate chain.
- *SSLv3:* Secure Socket Layer version 3 [SSL]
- *SSL Compatibility Mode:* Normally GSI extends the SSL protocol to allow for delegation and better performance. GSI can be run in SSL compatibility mode, which turns off these features and allows for GSI messages to be transmitted over a TCP connection to look identical to SSL over TCP. This mode is not used in general and is not covered in this document.

4 Specification of Messages

GSI messages may be conceptually thought of as having three phases, which occur in the given order[1]:

- Context establishment, where the two parties authenticate to each other and establish a context to protect further message exchange;
- Delegation, where the client may delegate credentials to the service for later use of their behalf; and

[1] In the context of the GSSAPI, the authentication and delegation phases are accomplished by calls to the gss_init_sec_context() and gss_accept_sec_context() calls and application data protection is accomplished by the gss_wrap() call.

- Application-specific, in which application data is exchanges, optionally protected by the context established in the first phase. Further delegations may also take place during this phase

The following subsections describe each phase of messages in detail.

4.1 Context Establishment Messages

In this phase the client and server are authenticated to each other and security information is exchanged in order to provide message protection for further data exchanges. In terms of the wire protocol, the context establishment phase is nothing more than normal SSLv3 handshake messages being exchanged.

The following attributes apply to the exchange:

- Either client or server may use standard an EEC certificate chain or a Proxy Certificate cert chain for authentication.
- Client authentication is optional in terms of the protocol, though most GSI-based services require it.
- GSI supports all the ciphers of the underlying SSL implementation it is built on. At the time of this writing that list is:
    - Encryption: DES, Triple-DES, IDEA, RC4, RC2
    - MAC: SHA1, MD5
    - The default cipher set is triple-DES with SHA1.
    - Encryption is optional, but integrity protection is required.

4.2 Delegation Messages

In this phase the client may, at the client discretion, delegate credentials to the server. These credentials can then be used by the server (or processes initiated by the server) on the client's behalf. GSI delegation consists of three messages:

1. A delegation flag from client to server indicating desire to delegate.
2. A PKCS10 certificate request [PKCS10] from server to client. This primarily serves to contain the public key of a newly generated key pair by the server.
3. A Proxy Certificate chain from client to server binding the public key provided by the server to an identity derived from the client identity.

While GSI does support delegation in either direction later during the exchange of application data (as described in Section 4.4), delegation during the delegation phase of the protocol may only occur from client to server.

Delegation messages are protected by the SSLv3 context established in the first phase of message exchanges. From the perspective of the SSLv3 protocol, they are treated as any other application data. This means they are, at a minimum, integrity protected and may also be encrypted, depending on the ciphers chosen during context establishment.

4.2.1 Delegation Flag

The first message is sent from client to server and indicates whether delegation will take place. This message consists of a single octet, which has the following legal values:

- "D" (ASCII code 68): Indicates that the client wishes to perform delegation.
- "0" (ASCII code 48): Indicates that the client does not wish to perform delegation.

No codes other than those above should be sent. Behavior upon receiving a code other than those above is undefined.

If the client does not wish to perform delegation, no further delegation messages are exchanged and further messages are application data as described in Section 4.3

If the client does wish to perform delegation, the rest of the messages described in this section are exchanged.

4.2.2 Certificate Request

In response to a client indicating that it wishes to perform delegation, the server should send an ASN.1 encoded certificate request.

The following components of the certificate request are meaningful:

- *Public key:* This will be the public key placed in the returned Proxy Certificate.
- *Proxy Policy:* If a ProxyCertInfo extension, as defined in [Proxies], is present. The Proxy Policy and Policy Languages fields will be duplicated in the returned Proxy Certificate unless the client specifically overrides them.

All other components of the certificate request are ignored by the client.

4.2.3 Delegated Certificate Chain

In response to the certificate request from the server, the client should respond by sending an ASN.1-encoded Proxy Certificate chain. This chain should conform to the following:

- It must be a series of DER-encoded certificates. Order of the certificates is unimportant.
- The chain must include at least the new Proxy Certificate, but may include all certificates in the Proxy Certificate chain.
- If the client used an End Entity Certificate or Internet Draft Proxy Certificate for authentication during context establishment, any Proxy Certificates in the delegated chain shall be Internet Draft Proxy Certificates. These certificates shall be formatted as specified in [Proxies].

4.3 Application Data Protection

In this phase application-specific data is exchanged. It may be protected by GSI to provide integrity and confidentiality. Delegation also occur during this phase as described in Section 4.4, but its framing will be completely application dependent.

There are three basic ways application data can be protected during this phase:

- *Using SSL-formated messages.* These application messages are simply the application data protected by SSLv3 using the context and ciphers derived during context establishment (Section 4.1). GSI does not add to or modify these messages as defined by SSL.
- *Using a separate signature.* In this mode the message format is completely application-specific, but GSI provides signature elements which provide integrity protection for these messages (via the GSSAPi getmic() call). The signature element is composed of the sequence of an 8 octet sequence number, a 4 octet message length and then the normal digest (whose length depends on the digest mechanism selected during context establishment). How these signature elements are conveyed is application-specific. An example of this is a GT3 SOAP message where the application data and signature element appear in different parts of the SOAP message.
- *Unprotected.* Messages may be completely unprotected by GSI and formatted in any manner the application desires.

It is possible, though not customary, to mix these various types of protection during an application session. How these changes in quality of protection are negotiated between the two parties is completely application dependent.

4.4 Delegation after Context Establishment

If an application wishes it may use the GGF-defined GSS-Extensions [GSS-Ext] to perform delegation at any time during application data exchange. This is typically only used in applications that do credential management (e.g. MyProxy [MyProxy]).

Note that in these cases it is up to the application to appropriate frame the delegation so that both sides are aware of the delegation.

Delegation in this context may occur in either direction (client to server or server to client) as framed by the application protocol.

The messages exchanged are identical to the messages exchanged in the delegation that happens immediately after context establishment (Section 4.2). Note that the delegation flag contained in the first message must always be a “D” indicating the desire to delegate.

4.5 Supported Keys

GSI should, in theory, support any type and size of key supported by SSLv3 as part of either an end entity or proxy certificate chain. At this time the keys in general use are RSA keys (512, 1024, and 2048 bits).

4.6 Error Handling

During the context establishment phase, SSLv3 alert messages may be exchanged by either side to indicate that a local error has occurred. Upon receiving such an alert message, the recipient should discontinue the current attempt at context establishment.

In other phases of message exchange, GSI does not have a defined method for error handling.

5 GT2 Proxy Certificates

With the release of GT3, the Internet Draft Proxy Certificates are now the default in GSI. However, there still exists a fairly large GT2 install base, so we describe the deprecated GT2 proxy certificates briefly.

GT2 proxy certificate have the following characteristics:

- There is no extension identifying them as proxy certificates as there is with the IETF-standardized proxy certificates, so they look very much like end entity certificates.
- The issuer is either an end entity certificate or another GT2 proxy certificate.
- The subject name is the subject name of the issuer with a proxy identifier appended.

The proxy identifier is a CommonName (CN) component with either the value "Proxy" or "LimitedProxy". "Proxy" specifies that the proxy is a full proxy, indicating a complete delegation of rights from the issuer to the proxy bearer was intended. "LimitedProxy" indicates that the issuer intended only a limited delegation, namely that the proxy must not be used for process invocation.

Unlike GT3 implementations as described in 4.2.2, some GT2 implementations treat the Subject Name in the certificate request as meaningful. To accommodate this a server expecting a client to delegate a GT2 Proxy Certificate (which should only happen if the client authenticated with a GT2 Proxy Certificate) should fill in the Subject Name in the certificate request with the subject name it expects to see in the delegated certificate (i.e. the client's subject name with a "CN=Proxy" component appended, see Section 4.5).

If the client used a GT2 Proxy Certificate for authentication during context establishment, the delegated Proxy Certificate shall be a GT2 Proxy Certificate. An Internet Draft proxy certificate may not be used to issue a GT2 proxy certificate or vice versa.

6 Acknowledgements

Contributions to the GSI concepts and code and this document have been made by (in alphabetical order with apologies to anyone missed): Doug Engert, Ian Foster, Jarek Gawor, Carl Kesselman, Sam Lang, Sam Meder, Olle Mulmo, Laura Pearlman, Frank Siebenlist, Steve Tuecke, Von Welch.

We also thank Matt Crawford for comments made on an earlier version of this document.

Numerous parties contributed to the design of the IETF Proxy Certificate specification and are listed in that document [Proxies].

7 References

[ADAT] Horowitz, M. and Lunt S. FTP Security Extensions. Internet RFC 2228, 1997.

[GSI] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke. Security for Grid Services. *Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*, IEEE Press, to appear June 2003.

[GSSAPI] Linn, J. Generic Security Service Application Program Interface, Version 2. *INTERNET RFC 2078*, 1997.

[GSS-Ext] Meder, S., Welch, V., Tucke, S., and Engert, D. GSS-API Extensions. GGF Draft, 2003.

[MyProxy] J. Novotny, S. Tuecke, V. Welch. An Online Credential Repository for the Grid: MyProxy. *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*, IEEE Press, August 2001.

[Proxies] Tuecke, S., Welch, V., Engert, D., Pearlman, L., and Thompson, M. Internet X.509 Public Key Infrastructure Proxy Certificate Profile, IETF, 2003. http://www.ietf.org/internet-drafts/draft-ietf-pkix-proxy-10.txt

[PKCS10] Kaliski, B., PKCS #10: Certification Request Syntax v1.5, RFC 2314, October 1997.

[RFC 3280] Houseley, R., Polk, W., Ford, W., and Solo, D. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Internet RFC 3280, 2002.

[SSL] Dierks, T. and Allen, C. The TLS Protocol Version 1.0, IETF, 1999. http://www.ietf.org/rfc/rfc2246.txt.