XEP-0218: Bootstrapping Implementation of Encrypted Sessions

Peter Saint-Andre
mailto:stpeter@stpeter.im
xmpp:stpeter@jabber.org
https://stpeter.im/

Ian Paterson
mailto:ian.paterson@clientside.co.uk
xmpp:ian@zoofy.com

2007-05-30
Version 0.1

Status	Type	Short Name
Deferred	Informational	N/A

This document provides guidelines to client and library developers for bootstrapping implementation of the encrypted sessions technology.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

The XMPP Standards Foundation (XSF) [1] has defined a technology for end-to-end encryption of XMPP communications, called "Encrypted Sessions" (ESessions). This document describes ways for client and library developers to approach the task of implementing ESessions.
In essence, implementation of ESessions proceeds in two directions:

1. From the client/interface level down.
2. From the library/API level up.

If client developers implement the "frontend" specifications, they should be able to integrate the "backend" code developed by library developers, enabling the two sets of developers to "meet in the middle" and offer complete implementations.

2 Approach

When working from the client/interface level down to the library/API level, it makes sense to implement the relevant specifications in the following order:

1. Best Practices for Message Threads (XEP-0201) [2]
   This document describes what a "stanza session" is, i.e., it is defined by the life of a message thread. Clients that implement this specification are well on their way to implementing encrypted sessions, since it is necessary to have a clear session "object" in a client before implementing an encrypted version of such a session.

2. Stanza Session Negotiation (XEP-0155) [3]
   Because this document describes how to negotiate a stanza session, it is a building block for developing how to negotiate an encrypted stanza session.

3. Stanza Encryption (XEP-0200) [4]
   By hardcoding the initial parameters during an early phase of development, implementors can use this specification as a starting point for testing of encrypted sessions. A later phase would address rekeying (Section 9) and negotiation of the initial parameters (see below).

[1] The XMPP Standards Foundation (XSF) is an independent, non-profit membership organization that develops open extensions to the IETF's Extensible Messaging and Presence Protocol (XMPP). For further information, see <https://xmpp.org/about/xmpp-standards-foundation>.
[2] XEP-0201: Best Practices for Message Threads <https://xmpp.org/extensions/xep-0201.html>.
[3] XEP-0155: Stanza Session Negotiation <https://xmpp.org/extensions/xep-0155.html>.
[4] XEP-0200: Stanza Encryption <https://xmpp.org/extensions/xep-0200.html>.

4. Simplified Encrypted Session Negotiation (XEP-0217) [5]
   This specification (to be published concurrently) defines a simple subset of the process for negotiating the initial parameters used in an encrypted session.

5. Encrypted Session Negotiation (XEP-0116) [6]
   This specification defines the ”heavy lifting” involved in going from a cleartext state to an encrypted state, i.e., it specifies how to the initial parameters for an encrypted session.

6. Cryptographic Design of Encrypted Sessions (XEP-0188) [7]
   Strictly speaking, a developer does not implement this specification, since it describes the theory behind the process of upgrading from cleartext to encryption that is defined in Encrypted Session Negotiation (XEP-0116) [8]. However it is useful background knowledge for developers who implement XEP-0116.

Naturally, when developing from the library/API level up to the client/interface level, the order should be (roughly) reversed.

3 Optional Add-Ons

Once a library or client has implemented the specifications listed above, it may choose to implement the following additional specifications, which supplemented the core encrypted sessions specifications.

1. Public Key Publishing (XEP-0189) [9]
   This specification defines a precondition for implementation of the specifications that follow.

2. Offline Encrypted Sessions (XEP-0187) [10]
   We should be able to encrypt so-called ”offline messages” (see Best Practices for Handling Offline Messages (XEP-0160) [11]) using the same basic principles used to encrypted messages sent while online.

[5] XEP-0217: Simplified Encrypted Session Negotiation <https://xmpp.org/extensions/xep-0217.html>.
[6] XEP-0116: Encrypted Session Negotiation <https://xmpp.org/extensions/xep-0116.html>.
[7] XEP-0188: Cryptographic Design of Encrypted Sessions <https://xmpp.org/extensions/xep-0188.html>.
[8] XEP-0116: Encrypted Session Negotiation <https://xmpp.org/extensions/xep-0116.html>.
[9] XEP-0189: Public Key Publishing <https://xmpp.org/extensions/xep-0189.html>.
[10] XEP-0187: Offline Encrypted Sessions <https://xmpp.org/extensions/xep-0187.html>.
[11] XEP-0160: Best Practices for Handling Offline Messages <https://xmpp.org/extensions/xep-0160.html>.

3. Message Archiving (XEP-0136) [12]
   This specification enables secure archiving of the messages sent and received in an encrypted session.

4 Security Considerations

Incomplete implementations of the Encrypted Sessions technology will not have the same security properties as complete implementations.

5 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [13].

6 XMPP Registrar Considerations

This document requires no interaction with the XMPP Registrar [14].

[12] XEP-0136: Message Archiving <https://xmpp.org/extensions/xep-0136.html>.

[13] The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

[14] The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.