XEP-0210: Requirements for Encrypted Sessions

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

2007-05-30
Version 0.2

Status	Type	Short Name
Deferred	Standards Track	N/A

This document describes the requirements for an XMPP end-to-end encrypted session protocol.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction 1

2 Scope 2

3 Security Requirements 2

3.1 Confidentiality . . . 3
3.2 Integrity . . . 3
3.3 Replay Protection . . . 3
3.4 Perfect Forward Secrecy . . . 3
3.5 PKI Independence . . . 4
3.6 Authentication . . . 4
3.7 Identity Protection . . . 4
3.8 Repudiability . . . 4
3.9 Robustness . . . 4
3.10 Upgradability . . . 5

4 Application Requirements 5

4.1 Generality . . . 5
4.2 Implementability . . . 6
4.3 Usability . . . 6
4.4 Efficiency . . . 6
4.5 Flexibility . . . 6
4.6 Offline Sessions . . . 6
4.7 Interoperability . . . 7
4.8 Object Encryption . . . 7

5 Security Considerations 7

6 IANA Considerations 7

7 XMPP Registrar Considerations 7

1 Introduction

Existing approaches to encryption of XMPP communications have generally assumed that each stanza to be encrypted is a standalone storeable object; the term ”object encryption” well captures this assumption. Both Current Jabber OpenPGP Usage (XEP-0027) [1] and RFC 3923 [2] assume that no interactive session exists, and that XMPP communications are similar to the exchange of files or email messages - where the receiver is typically not connected to its server at the time the message is sent. Although Current Jabber OpenPGP Usage uses ”old-style” PGP object encryption and RFC 3923 [3] uses ”new-style” S/MIME object encryption, both specify the use of object encryption. Any new protocol based on XML Encryption [4] and XML Signature [5], would also be an ”object encryption” protocol.

However, encryption schemes that are appropriate for less dynamic Internet technologies are not appropriate for session-oriented communication technologies like XMPP. With XMPP the receiver is typically connected to its server at the time the message is sent, so XMPP can take advantage of much more secure session-oriented approaches to encryption - approaches that are not feasible for less dynamic technologies like email. Most importantly, XMPP can benefit from Perfect Forward Secrecy and Identity Protection.

Therefore, for XMPP, the focus should be on ”session encryption” rather than ”object encryption”. The paradigm should be something closer to the widely-deployed Secure Shell technology (see RFC 4253 [6]) or RFC 6189 [7] (an acclaimed SRTP - RFC 3711 [8] - key agreement protocol) or TLS (see RFC 5246 [9]) or IPsec (see RFC 4301 [10]) than to the traditional encryption of files and email messages.

The session metaphor applies to communication between any two XMPP endpoints. For instance, in IM applications, most instant messaging exchanges occur in bursts within limited time periods (e.g., two people may send a fairly large number of messages during a five-minute chat and then not exchange messages again for hours or even days). The XML stanzas exchanged during such a session may not be limited to <message/> stanzas; for instance, the session may be triggered by a change in one of the parties' presence status (e.g., changing from away to available) and the session may involve the exchange of <iq/> stanzas (e.g., to transfer a file as specified in SI File Transfer (XEP-0096) [11]).

Note: The encryption of archived messages is necessarily less secure than session encryption. The encryption of such stored messages is described in Message Archiving (XEP-0136) [12] and

[1] XEP-0027: Current Jabber OpenPGP Usage <https://xmpp.org/extensions/xep-0027.html>.
[2] RFC 3923: End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP) <http://tools.ietf.org/html/rfc3923>.
[3] RFC 3923: End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP) <http://tools.ietf.org/html/rfc3923>.
[4] XML Encryption Syntax and Processing <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>.
[5] XML Signature Syntax and Processing <http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/>.
[6] RFC 4253: The Secure Shell (SSH) Transport Layer Protocol <http://tools.ietf.org/html/rfc4253>.
[7] RFC 6189: ZRTP: Media Path Key Agreement for Unicast Secure RTP <http://tools.ietf.org/html/rfc6189>.
[8] RFC 3711: The Secure Real-time Transport Protocol (SRTP) <http://tools.ietf.org/html/rfc3711>.
[9] RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2 <http://tools.ietf.org/html/rfc5246>.
[10] RFC 4301: Security Architecture for the Internet Protocol <http://tools.ietf.org/html/rfc4301>.
[11] XEP-0096: SI File Transfer <https://xmpp.org/extensions/xep-0096.html>.
[12] XEP-0136: Message Archiving <https://xmpp.org/extensions/xep-0136.html>.

is therefore out-of-scope for this document.

2 Scope

The XMPP communications described above exist in the context of a one-to-one communication session between two entities. However, several forms of XMPP communication exist outside the context of one-to-one communication sessions:

- Many-to-many sessions, such as a text conference in a chatroom as specified in Multi-User Chat (XEP-0045) [13].
- One-to-many "broadcast", such as undirected presence stanzas sent from one user to many contacts (see RFC 3921 [14]) and data syndication implemented using Publish-Subscribe (XEP-0060) [15].
- One-to-one communications that are stored for later delivery rather than delivered immediately, such as so-called "offline messages".

Ideally, any technology for end-to-end encryption in XMPP could be extended to cover all the scenarios above as well as one-to-one communication sessions. However, both many-to-many sessions and one-to-many broadcast are deemed out-of-scope for this document.
Communications where the receiving entity is offline should ideally be handled via a simple extension to the protocol for one-to-one sessions between two entities that are online simultaneously. This approach enables code reuse, minimises the points of failure and significantly increases the security (for example, by providing Perfect Forward Secrecy).

3 Security Requirements

This document stipulates the following security requirements for end-to-end encryption of XMPP communications:

- Confidentiality
- Integrity
- Replay protection
- Perfect forward secrecy

[13] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[14] RFC 3921: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc3921>.
[15] XEP-0060: Publish-Subscribe <https://xmpp.org/extensions/xep-0060.html>.

- PKI Independence
- Authentication
- Identity Protection
- Repudiability
- Robustness
- Upgradability

Each of these requirements is explained in greater depth below.

3.1 Confidentiality

The one-to-one XML stanzas exchanged between two entities MUST NOT be understandable to any other entity that might intercept the communications. The encrypted stanzas should be understood by an intermediate server only to the extent required to route them. (One complicating factor is that routing information may include not only the stanza's 'to', 'from', 'type, and 'id' attributes, but also Advanced Message Processing (XEP-0079) [16] extensions.)

3.2 Integrity

Alice and Bob MUST be sure that no other entity may change the content of the XML stanzas they exchange, or remove or insert stanzas into the ESession undetected.

3.3 Replay Protection

Alice or Bob MUST be able to identify and reject any communications that are copies of their previous communications resent by another entity.

3.4 Perfect Forward Secrecy

The encrypted communication MUST NOT be revealed even if long-lived keys are compromised in the future (e.g., Steve steals Bob's computer). For long-lived sessions it MUST be possible to periodically change the decryption keys. [17]

[16] XEP-0079: Advanced Message Processing <https://xmpp.org/extensions/xep-0079.html>.

[17] Long-lived keys are typically used for a few years, whereas Offline ESession keys are destroyed as soon as the stanza is decrypted - they typically exist for just a few hours. So Perfect Forward Secrecy should significantly enhance the security even of Offline ESessions.

3.5 PKI Independence

The protocol MUST NOT rely on any public key infrastructure (PKI), certification authority, web of trust, or any other trust model that is external to the trust established between Alice and Bob. However, if external authentication or trust models are available then Alice and Bob MUST be able to use them to enhance any trust that exists between them.

3.6 Authentication

Each party to a conversation MUST know that the other party is who they want to communicate with (Alice must be able to know that Bob really is Bob, and vice versa). [18]

3.7 Identity Protection

No other entity should be able to identify Alice or Bob. The JIDs they use to route their stanzas are unavoidably vulnerable to interception. So, even if Alice and Bob protect their identities by using different JIDs for each session, it MUST be possible for their clients to authenticate them transparently, without any other entity identifying them via an active ("man-in-the-middle") attack, or even linking them to their previous sessions. If that is not possible because Alice and Bob choose to authenticate using public keys instead of retained shared secrets, then the public keys MUST NOT be revealed to other entities using a passive attack. Bob MUST also be able to choose between protecting either his public key or Alice's public key from disclosure through an active attack.

3.8 Repudiability

Alice and Bob MUST be able to repudiate any stanza that occurs within an ESession. After an ESession has finished, it MUST NOT be possible to *prove cryptographically* that any transcript has not been modified by a third party. [19]

3.9 Robustness

The protocol SHOULD provide more than one difficult challenge that has to be overcome before an attack can succeed (for example, by generating encryption keys using as many shared secrets as possible - like retained secrets or optional passwords).

[18] Authentication is not identification, authentication may be as simple as Alice confirming that Bob is the same Bob that she communicated with yesterday or that she talked to on the telephone. The reliable association between an entity and its public keys is "identification" and therefore beyond the scope of this document.

[19] Naturally, it is possible that Alice or Bob may retain cleartext versions of the exchanged communications; however, that threat is out-of-scope for this document.

3.10 Upgradability

The protocol MUST be upgradable so that, if a vulnerability is discovered, a new version can fix it. Alice MUST tell Bob which versions of the protocol she is prepared to support. Then Bob MUST either choose one or reject the ESession. [20]

4 Application Requirements

In addition to the foregoing security profile, this document also stipulates the following application-specific requirements for encrypted communication in the context of Jabber/XMPP technologies:

- Generality
- Implementability
- Usability
- Efficiency
- Flexibility
- Offline ”sessions”
- Interoperability
- Object encryption

Each of these is explained in greater depth below.

4.1 Generality

The solution MUST be generally applicable to the full content of any XML stanza type (<message/>, <presence/>, <iq/>) sent between two entities. It is deemed acceptable if the solution does not apply to many-to-many stanzas (e.g., groupchat messages sent within the context of multi-user chat) or one-to-many stanzas (e.g., presence ”broadcasts” and pubsub notifications); end-to-end encryption of such stanzas may require separate solutions.

[20] It is exceptionally difficult to design a truly secure authenticated key-exchange protocol. Weaknesses are often only discovered after years of expert cryptographic analysis. In many cases, only the widespread use of a protocol will motivate experts to undertake exhaustive analyses and recommend enhancements.

4.2 Implementability

The only good security technology is an implemented security technology. The solution SHOULD be one that client developers can implement in a relatively straightforward and interoperable fashion.

4.3 Usability

The requirement of usability takes implementability one step further by stipulating that the solution MUST be one that organizations may deploy and humans may use with 100% transparency (with the ease-of-use of https:). Experience has shown that: solutions requiring a full public key infrastructure do not get widely deployed, and solutions requiring any user action are not widely used. If, however, Alice and/or Bob are prepared to verify the integrity of their copies of each other's keys (thus enabling them to discover targeted active attacks or even the mass surveilance of a population), then the actions necessary for them to achieve that MUST be minimal (requiring no more effort than a one-time out-of-band verification of a string of up to 6 alphanumeric characters).

4.4 Efficiency

Cryptographic operations are highly CPU intensive, particularly public key and Diffie-Hellman operations. Cryptographic data structures can be relatively large, especially public keys and certificates. Network round trips can introduce unacceptable delays, especially over high-latency wireless connections. The solution MUST perform efficiently even when CPU and network bandwidth are constrained. The number of stanzas required for ESession negotiation MUST be minimized.

4.5 Flexibility

The solution MUST be compatible with a variety of existing (and future) cryptographic algorithms and identity certification schemes (including X.509 and PGP). The protocol MUST also be able to evolve to correct the weaknesses that are inevitably discovered once any cryptographic protocol is in widespread use.

4.6 Offline Sessions

It SHOULD be possible to encrypt one-to-one communications that are stored for later delivery (instead of being delivered immediately - so-called "offline messages") and still benefit from Perfect Forward Secrecy (with a slightly longer period of vulnerability than if both parties were online simultaneously). However, any vulnerabilities introduced into the solution in order to enable such offline communications MUST NOT make online communications more

vulnerable.

4.7 Interoperability

Ideally, it would be possible for an XMPP user to exchange encrypted messages (and, potentially, presence information) with users of non-XMPP messaging systems.

4.8 Object Encryption

Ideally, it would be possible in cases where a session is not desired, to encrypt, sign and send a single stanza in isolation, so-called ”object encryption”.

5 Security Considerations

Security issues are discussed throughout this document.

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [21].

7 XMPP Registrar Considerations

This document requires no interaction with the XMPP Registrar [22].

[21] 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/>.

[22] 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/>.