Internet-Draft
Intended Status: Best Current Practice
Expires: 20 June 2014

R. Housley
Vigil Security

20 December 2013

Guidelines for Cryptographic Algorithm Agility
<draft-housley-crypto-alg-agility-00.txt>

Abstract

Many IETF protocols may use of cryptographic algorithms to provide confidentiality, integrity, or non-repudiation. Communicating peers must support the same cryptographic algorithm or algorithms for these mechanisms to work properly. This memo provides guidelines for ensuring that such a protocol has the ability to migrate from one algorithm to another over time.

1. Introduction

Many IETF protocols may use of cryptographic algorithms to provide confidentiality, integrity, or non-repudiation. For the mechanisms to work properly,communicating peers must support the same cryptographic algorithm or algorithms. Yet, cryptographic algorithms become weaker with time. As new cryptanalysis techniques are developed and computing performance improves, the work factor to break a particular cryptographic algorithm will reduce. For the protocol implementer, this means that implementations should be modular to easily accommodate the insertion of new algorithms. For the protocol designer, this means that one or more algorithm identifier must be carried, the set of mandatory to implement algorithms will change over time, and an IANA registry of algorithm identifiers will be needed.

1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

2. Guidelines

These guidelines are for use by IETF working groups and protocol authors for IETF protocols that make use of cryptographic algorithms.

2.1. Algorithm Identifiers

IETF protocols that make use of cryptographic algorithms MUST carry one or more algorithm identifier.

Some approaches carry one identifier for each algorithm that is used. Other approaches carry one identifier for a suite of algorithms. Either approach is acceptable; however, designers are encouraged to pick one of these approaches and use it consistently throughout the protocol.

An IANA registry SHOULD be used for these algorithm identifiers.

2.2. Mandatory to Implement Algorithms

For interoperability, communicating peers must support the same cryptographic algorithm or algorithms. For this reason, the protocol SHOULD specify one or more mandatory to implement algorithm. This is not done for protocols that are embedded in other protocols. For example, S/MIME [RFC5751] makes use of CMS [RFC5652]. Other protocols also make use of CMS. S/MIME specifies the mandatory to implement algorithms, not CMS.

The IETF must be able to change the mandatory to implement algorithms over time. It is highly desirable to make this change without updating the base protocol specification. Therefore the base protocol specification SHOULD reference a companion algorithms document, allowing the update of one document without necessarily requiring an update to the other. This division also facilitates the advancement of the base protocol specification on the maturity ladder even if the algorithm document changes frequently.

Some cryptographic algorithms are inherently tied to a specific key size, but others allows many different key sizes. When more than one key size is available, the algorithm specification MUST identify the specific sizes that are to be supported.

Guidance on cryptographic key size for public keys can be found in BCP 86 [RFC3766].

Symmetric keys used for protection of long-term values SHOULD be at least 128 bits.

2.3. Balance Security Strength

When selecting a suite of cryptographic algorithms, the strength of each algorithm MUST be considered.

In CMS [RFC5652], a previously distributed symmetric key-encryption key can be used to encrypt a content-encryption key, which is in turn used to encrypt the content. The key-encryption and content-encryption algorithms are often different. If, for example, a message content is encrypted with 168-bit Triple-DES key and the Triple-DES content-encryption key is wrapped with a 40-bit RC2 key, then at most 40 bits of protection is provided. Thus, a trivial search to determine the value of the 40-bit RC2 key will recover Triple-DES key, and then the recovered Triple-DES key can be used to decrypt the content. In this situation, the algorithm and key size selections should ensure that the key encryption is at least as

strong as the content encryption.

3. Algorithm Agility Considerations

Some attempts at algorithm agility have not been completely successful. This section provides some of the insights based on protocol designs and deployments.

3.1. Algorithm Identifiers

If a protocol does not carry an algorithm identifier, then the protocol version number or some other major change is needed to transition from one algorithm to another. The inclusion of an algorithm identifier is a minimal step toward cryptographic algorithm agility. In addition, an IANA registry is needed to pair the identifier with an algorithm specification.

Sometimes application layer protocols can make use of transport layer security protocols, such as TLS or DTLS. This insulates the application layer protocol from the cryptography altogether, but it may still necessary to handle the transition to from unprotected to protected use of the the application layer protocol.

3.2. Migration Mechanisms

When a protocol specifies a single mandatory-to-implement algorithm for integrity and authentication algorithm, eventually that algorithm will be found to be weak. Perhaps there will be a flaw found in the algorithm that greatly shortens its expected life. Regardless, all algorithms age, and the advances in computing power available to the attacker will eventually make them obsolete. For this reason, protocols need mechanisms to migrate from one algorithm to another over time.

Extra care is needed when a mandatory-to-implement algorithm is used to provide integrity protection for the negotiation of other cryptographic algorithms used by the protocol. In this situation, a flaw in the mandatory-to-implement algorithm may allow an attacker to influence the choices of other algorithms.

3.3. Cryptographic Key Management

Traditionally, protocol designers have avoided a more than one approach to key management because it makes the security analysis of the overall protocol more difficult. However, with the increasing deployment of frameworks such as EAP and GSSAPI, the key management is very flexible, often hiding many of the details from the application. As a result, more and more protocols support multiple

key management approaches. In fact, the key management approach may be negotiable, which creates a design challenge to protect the negotiation of the key management approach before it is used to produce cryptographic keys for the cryptographic algorithm.

Protocols can negotiate a key management approach, derive an initial cryptographic key, and then authenticate the negotiation. However, if the authentication fails, the only recourse is to start the negotiation over from the beginning.

Some environments will restriction the key management approaches by policy. Such policies tend to improve interoperability within a particular environment, but they cause problems for individuals that need to work in multiple incompatible environments.

4. Security Considerations

This document provides guidance to working groups and protocol designers. The security of the Internet is improved when broken or weak cryptographic algorithms can be easily replaced with strong ones.

5. References

This section contains normative and informative references.

5.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3766] Orman, H. and P. Hoffman, "Determining Strengths For Public Keys Used For Exchanging Symmetric Keys", BCP 86, RFC 3766, April 2004.

5.2. Informative References

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, September 2009.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

Acknowledgements

Thanks to Bernard Aboba for his review and insightful comments.

Authors' Addresses

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170
USA
EMail: housley@vigilsec.com