Network Working Group
Request for Comments: 4107
BCP: 107
Category: Best Current Practice

S. Bellovin
Columbia University
R. Housley
Vigil Security

June 2005

Guidelines for Cryptographic Key Management

Status of This Memo

Copyright Notice

Abstract

The question often arises of whether a given security system requires some form of automated key management, or whether manual keying is sufficient. This memo provides guidelines for making such decisions. When symmetric cryptographic mechanisms are used in a protocol, the presumption is that automated key management is generally but not always needed. If manual keying is proposed, the burden of proving that automated key management is not required falls to the proposer.

1. Introduction

The question often arises of whether or not a given security system requires some form of automated key management, or whether manual keying is sufficient.

There is not one answer to that question; circumstances differ. In general, automated key management SHOULD be used. Occasionally, relying on manual key management is reasonable; we propose some guidelines for making that judgment.

On the other hand, relying on manual key management has significant disadvantages, and we outline the security concerns that justify the preference for automated key management. However, there are situations in which manual key management is acceptable.

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 RFC 2119 [B].

2. Guidelines

These guidelines are for use by IETF working groups and protocol authors who are determining whether to mandate automated key management and whether manual key management is acceptable. Informed judgment is needed.

The term "key management" refers to the establishment of cryptographic keying material for use with a cryptographic algorithm to provide protocol security services, especially integrity, authentication, and confidentiality. Automated key management derives one or more short-term session keys. The key derivation function may make use of long-term keys to incorporate authentication into the process. The manner in which this long-term key is distributed to the peers and the type of key used (pre-shared symmetric secret value, RSA public key, DSA public key, and others) is beyond the scope of this document. However, it is part of the overall key management solution. Manual key management is used to distribute such values. Manual key management can also be used to distribute long-term session keys.

Automated key management and manual key management provide very different features. In particular, the protocol associated with an automated key management technique will confirm the liveness of the peer, protect against replay, authenticate the source of the short-term session key, associate protocol state information with the short-term session key, and ensure that a fresh short-term session key is generated. Further, an automated key management protocol can improve interoperability by including negotiation mechanisms for cryptographic algorithms. These valuable features are impossible or extremely cumbersome to accomplish with manual key management.

For some symmetric cryptographic algorithms, implementations must prevent overuse of a given key. An implementation of such algorithms can make use of automated key management when the usage limits are nearly exhausted, in order to establish replacement keys before the limits are reached, thereby maintaining secure communications.

Examples of automated key management systems include IPsec IKE and Kerberos. S/MIME and TLS also include automated key management functions.

Key management schemes should not be designed by amateurs; it is almost certainly inappropriate for working groups to design their own. To put it in concrete terms, the very first key management protocol in the open literature was published in 1978 [NS]. A flaw and a fix were published in 1981 [DS], and the fix was cracked in 1994 [AN]. In 1995 [L], a new flaw was found in the original 1978 version, in an area not affected by the 1981/1994 issue. All of these flaws were obvious once described -- yet no one spotted them earlier. Note that the original protocol (translated to employ certificates, which had not been invented at that time) was only three messages.

Key management software is not always large or bloated. Even IKEv1 [HC] can be done in less than 200 Kbytes of object code, and TLS [DA] in half that space. Note that this TLS estimate includes other functionality as well.

A session key is used to protect a payload. The nature of the payload depends on the layer where the symmetric cryptography is applied.

In general, automated key management SHOULD be used to establish session keys. Strong justification is needed in the security considerations section of a proposal that makes use of manual key management.

2.1. Automated Key Management

Automated key management MUST be used if any of these conditions hold:

A party will have to manage n^2 static keys, where n may become large.

Any stream cipher (such as RC4 [TK], AES-CTR [NIST], or AES-CCM [WHF]) is used.

An initialization vector (IV) might be reused, especially an implicit IV. Note that random or pseudo-random explicit IVs are not a problem unless the probability of repetition is high.

Large amounts of data might need to be encrypted in a short time, causing frequent change of the short-term session key.

Long-term session keys are used by more than two parties. Multicast is a necessary exception, but multicast key management standards are emerging in order to avoid this in the future. Sharing long-term session keys should generally be discouraged.

The likely operational environment is one where personnel (or device) turnover is frequent, causing frequent change of the short-term session key.

2.2. Manual Key Management

Manual key management may be a reasonable approach in any of these situations:

The environment has very limited available bandwidth or very high round-trip times. Public key systems tend to require long messages and lots of computation; symmetric key alternatives, such as Kerberos, often require several round trips and interaction with third parties.

The information being protected has low value.

The total volume of traffic over the entire lifetime of the long-term session key will be very low.

The scale of each deployment is very limited.

Note that assertions about such things should often be viewed with skepticism. The burden of demonstrating that manual key management is appropriate falls to the proponents -- and it is a fairly high hurdle.

Systems that employ manual key management need provisions for key changes. There MUST be some way to indicate which key is in use to avoid problems during transition. Designs SHOULD sketch plausible mechanisms for deploying new keys and replacing old ones that might have been compromised. If done well, such mechanisms can later be used by an add-on key management scheme.

Lack of clarity about the parties involved in authentication is not a valid reason for avoiding key management. Rather, it tends to indicate a deeper problem with the underlying security model.

2.3. Key Size and Random Values

Guidance on cryptographic key size for public keys that are used for exchanging symmetric keys can be found in BCP 86 [OH].

When manual key management is used, long-term shared secret values SHOULD be at least 128 bits.

Guidance on random number generation can be found in BCP 106 [ESC].

When manual key management is used, long-term shared secrets MUST be unpredictable "random" values, ensuring that an adversary will have no greater expectation than 50% of finding the value after searching half the key search space.

3. Security Considerations

This document provides guidance to working groups and protocol designers. The security of the Internet is improved when automated key management is employed.

The inclusion of automated key management does not mean that an interface for manual key management is prohibited. In fact, manual key management is very helpful for debugging. Therefore, implementations ought to provide a manual key management interface for such purposes, even if it is not specified by the protocol.

4. References

This section contains normative and informative references.

4.1. Normative References

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

[ESC] Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

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

4.2. Informative References

[AN] M. Abadi and R. Needham, "Prudent Engineering Practice for Cryptographic Protocols", Proc. IEEE Computer Society Symposium on Research in Security and Privacy, May 1994.

[DA] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[DS] D. Denning and G. Sacco. "Timestamps in key distributed protocols", Communication of the ACM, 24(8):533--535, 1981.

[HC] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[L] G. Lowe. "An attack on the Needham-Schroeder public key authentication protocol", Information Processing Letters, 56(3):131--136, November 1995.

[NIST] National Institute of Standards and Technology. "Recommendation for Block Cipher Modes of Operation -- Methods and Techniques," NIST Special Publication SP 800-38A, December 2001.

[NS] R. Needham and M. Schroeder. "Using encryption for authentication in large networks of computers", Communications of the ACM, 21(12), December 1978.

[TK] Thayer, R. and K. Kaukonen. "A Stream Cipher Encryption Algorithm", Work in Progress.

[WHF] Whiting, D., Housley, R., and N. Ferguson , "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.

Authors' Addresses

Steven M. Bellovin
Department of Computer Science
Columbia University
1214 Amsterdam Avenue, M.C. 0401
New York, NY 10027-7003

Phone: +1 212-939-7149
EMail: bellovin@acm.org

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170

Phone: +1 703-435-1775
EMail: housley@vigilsec.com

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.