Internet Engineering Task Networking Group
IP Security Working Group
INTERNET-DRAFT
Category: Standards Track
Expires November 1999

Scott Judy
Sandra MacGregor
Mykotronx, Inc.

May 1999

The ESP SKIPJACK-CBC Cipher Algorithm With Implicit IV
<draft-ietf-ipsec-skipjack-cbc-00.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This document is a submission to the IETF Internet Protocol Security (IPSEC) Working Group. Comments are solicited and should be addressed to the working group mailing list (ipsec@tis.com) and/or to mfurusawa@myko.rainbow.com, Telephone: +1 (301) 533-8100 x6307.

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This protocol describes the SKIPJACK symmetric block cipher algorithm. The SKIPJACK algorithm is a confidentiality mechanism used, with other mechanisms, to provide secure messaging.

This protocol describes the use of SKIPJACK in Cipher Block Chaining (CBC) mode with an Implicit IV within the context of the IP Encapsulating Security Payload [ESP].

Table of Contents

1. Introduction

The Encapsulating Security Payload (ESP) provides confidentiality for IP datagrams by encrypting the payload data to be protected. This protocol is intended to provide the encryption portion of this confidentiality service by use of the SKIPJACK algorithm, and be used between between special purpose units such as terminal servers or routers and a monitoring host, and also between clients and servers on host computers. Typically the clients are on workstation hosts and the servers are on mainframe hosts.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119].

2. Overview of the SKIPJACK Algorithm

SKIPJACK is a 64-bit "electronic codebook" symmetric block algorithm that transforms a 64-bit input block into a 64-bit output block. Each input and output result in the same number of octets, which facilitates in-place encryption and decryption.

The transformation is parameterized by an 80-bit cryptovariable(key) and a 64-bit Initialization Vector. The algorithm can be used in any one of the four DES operating modes defined in [skip/kea], [FIPS 81], [Denning]. This protocol addresses only the Cipher Block Chaining (CBC) mode. [Schneier96] provides a provides a general description of Cipher Block Chaining Mode, a mode which is applicable to several encryption algorithms.

The SKIPJACK algorithm was developed by NSA and implemented by Mykotronx, Inc. in the Clipper and Capstone chips, and by other hardware manufacturers. SKIPJACK is intended to protect sensitive but unclassified information [valid]. The algorithm is described in [skip/kea] and [FIPS-185].

SKIPJACK encrypts an 8-octet data block by alternating between the two stepping rules (A and B) performed 32 times. Each step is an iteration of a complex, nonlinear function. Each stepping rule includes the use of a G-permutation, a 4-round Feistel structure based on a fixed byte-substitution table called the F-table. Each round of G also incorporates 1 byte of the 10-byte cryptovariable (key) used in its natural order.

2.1 Cryptovariable (Key)

The secret SKIPJACK-CBC cryptovariable (key) of 80 bits (10 bytes) MUST be randomly generated.

[arch] describes the general mechanism to derive keying material for the ESP transform. The key does not differ between the manually- and automatically-keyed security associations. This random generation MUST produce an 80-bit key value for use by this cipher.

When implemented in FORTEZZA(R) compatible applications the SKIPJACK cryptovariable (key) is known by convention as the Message Encryption Key (MEK). The MEK must be encrypted (or wrapped) by another cryptovariable, which is generated by the Key Exchange Algorithm (KEA). This second cryptovariable (key) is known by convention as the Token Encryption Key (TEK).

A strong random number generator (randomizer) or pseudo-random function MUST be used to generate the required cryptovariable (key). For a discussion on this topic, reference [RFC1750].

2.2 Initialization Vector (IV)

The Cipher Block Chaining (CBC) mode of SKIPJACK requires an Initialization Vector (IV) of 8 octets (64 bits). The IV MUST be a random value.

The IV is XOR'ed with the first input block of data. After encryption, the output block of encrypted data is XOR'ed with the next input block; this logically extends (chains) the encrypted IV across the packets. The receiver MUST NOT assume any meaning for this value, other than that it is an IV.

A strong pseudo-random function MUST be used to generate the required Initialization Vector. For a discussion on this topic, reference [RFC1750].

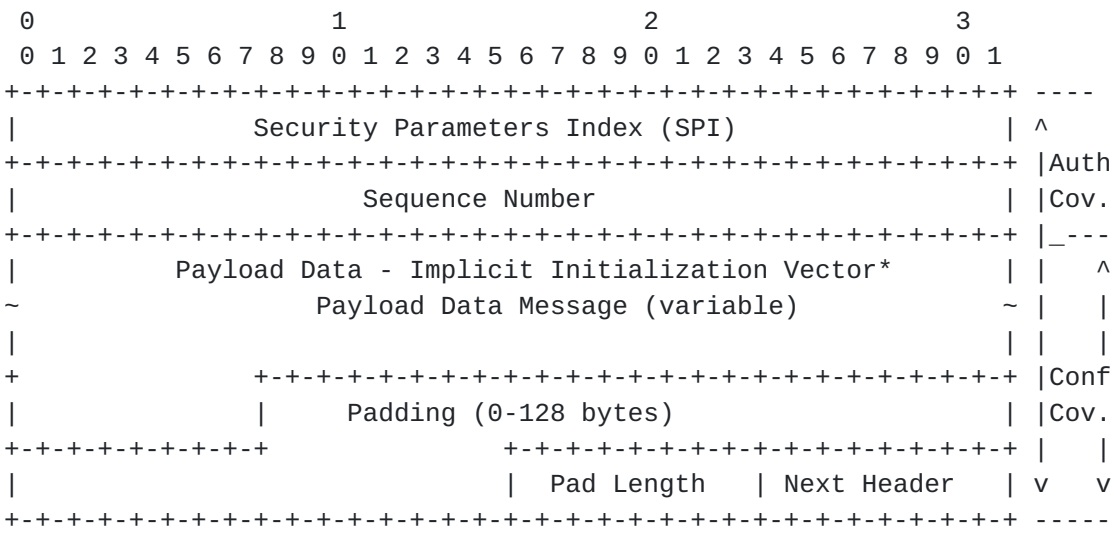
To avoid ECB mode encryption of very similar plaintext blocks in different packets, implementations MUST NOT use a counter or other low-Hamming distance source for IVs.

2.3 Performance

At the time of writing, the SKIPJACK algorithm has been implemented primarily in hardware and firmware, refer to [valid]. These solutions can encrypt and decrypt at data rates of approximately 50 mbps.

3. ESP and SKIPJACK Payload Packet Format

The SKIPJACK payload data field within the ESP packet format, as defined in [ESP], is broken down according to the following diagram:



* The Implicit Initialization Vector occurs only in the first packet.

3.1 Security Parameters Index (SPI)

A 32-bit value identifying the Security Parameters for this datagram. The value MUST NOT be zero.

3.2 Sequence Number

This unsigned 32-bit field contains a monotonically increasing counter value (sequence number). It is mandatory and is always present even if the receiver does not elect to enable the anti-replay service for a specific SA. Processing of the Sequence Number field is at the discretion of the receiver, i.e., the sender MUST always transmit this field, but the receiver need not act upon it. [ESP]

3.3 Initialization Vector

The CBC mode of SKIPJACK requires an implicit Initialization Vector (IV) of 8 octets (64 bits). The size of the IV is identical to the block size. The IV immediately precedes the data in the first payload packet.

Octets are sent in network order (most significant octet first)

[RFC-1700].

3.4 Payload Data

The SKIPJACK-CBC algorithm described in this document MUST use a block size of 8 octets (64 bits). The input data MUST be padded to this block size of 64 bits; i.e., the size of the complete payload must be a multiple of 64 bits.

The number of blocks is variable.

Prior to encryption and after decryption, this field contains the information described by the Next Header field. Note that in the

case of IP-in-IP encapsulation (Payload Type 4), this will be another IP header. The Payload Data field is mandatory and is an integral number of bytes in length. [ESP]

3.5 Padding

The SKIPJACK-CBC algorithm operates on full blocks of eight octets. This often requires padding after the end of the original data. The padding begins immediately after the last datum and extends to the end of the octet which the data occupies; i.e., the size of the complete payload data must be a multiple of 64 bits.

For the purpose of ensuring that the bits to be encrypted are a multiple of the algorithm's block size, the padding computation applies to the Payload Data exclusive of the Pad Length, and Next Header fields.

For the purposes of ensuring that the Authentication Data is aligned on a 4-byte boundary, the padding computation applies to the Payload Data inclusive of the IV, the Pad Length, and Next Header fields.

When padding is required, it MUST be done according to the conventions specified in [ESP]. An all 0 pattern, or the [ESP] default of bytes of increasing numerical sequence may be used.

After decryption, the padding MUST be ignored.

3.6 Pad Length

This field contains the size of the padding in bytes. It does not include the IV, Pad Length and Payload Type fields. The value typically is either 0 or 8, but may be up to 255 to permit hiding of the actual data length.

3.7 Next Header

This field indicates the contents of the Payload Data field, using the IP Protocol/Payload value. Up-to-date values of the IP Protocol/Payload are specified in the most recent "Assigned Numbers". [RFC-1700]

This field is opaque. That is, the value is set prior to encryption, and is examined only after decryption.

4. Security Considerations

"The strength of any encryption algorithm depends on its ability to withstand an attack aimed at determining either the key or the unencrypted ("plaintext") communications. There are basically two types of attack, brute-force and shortcut." [Denning]

4.1 Susceptibility to Brute Force Attack by Exhaustive Search

"In a brute-force attack (also called "exhaustive search"), the adversary essentially tries all possible keys until one is found that decrypts the intercepted communications into a known or meaningful plaintext message. The resources required to perform an exhaustive search depend on the length of the keys, since the number of possible keys is directly related to key length. In particular, a key of length N bits has 2^N possibilities. SKIPJACK uses 80-bit keys, which means there are 2^80 (approximately 10^24) or more than 1 trillion trillion possible keys." [Denning]

"Another way of looking at the problem is by comparing a brute force attack on SKIPJACK with one on DES, which uses 56-bit keys. Since SKIPJACK keys are 24 bits longer than DES keys, there are 2^24 times more possibilities. " [Denning] Assuming that the cost of processing power is halved every eighteen months, then it will not be for another (2^24)/(2^1.5x) = 1; x = 16 years before the cost of breaking SKIPJACK is equal to the cost of breaking DES today.

"Conclusion 1: Under an assumption that the cost of processing power is halved every eighteen months, it will be 16 years before the cost of breaking SKIPJACK by exhaustive search will be equal to the cost of breaking DES today." [Denning]

4.2 Susceptibility to Shortcut Attacks

"In a shortcut attack, the adversary exploits some property of the encryption algorithm that enables the key or plaintext to be

determined in much less time than by exhaustive search. For example, the RSA public-key encryption method is attacked by factoring a public value that is the product of two secret primes into its primes." [Denning]

"Most shortcut attacks use probabilistic or statistical methods that exploit a structural weakness, unintentional or intentional (i.e., a "trapdoor"), in the encryption algorithm. In order to determine whether such attacks are possible, it is necessary to thoroughly examine the structure of the algorithm and its statistical properties. In the time available for this review, it was not feasible to conduct an evaluation on the scale that NSA has conducted or that has been conducted on the DES. Such review would require many man-years of effort over a considerable time interval. Instead, we concentrated on reviewing NSA's design and evaluation process. In addition, we conducted several of our own tests." [Denning]

4.3 NSA's Design and Evaluation Process

"SKIPJACK was designed using building blocks and techniques that date back more than forty years. Many of the techniques are related to work that was evaluated by some of the world's most accomplished and famous experts in combinatorics and abstract algebra. SKIPJACK's more immediate heritage dates to around 1980, and it's initial design to 1987." [Denning]

"SKIPJACK was designed to be evaluatable, and the design and evaluation approach was the same used with algorithms that protect the country's most sensitive classified information. The specific structures included in SKIPJACK have a long evaluation history, and the cryptographic properties of those structures had many prior years of intense study before the formal process began in 1987. Thus, an arsenal of tools and data was available. This arsenal was used by dozens of adversarial evaluators whose job was to break SKIPJACK. Many spent at least a full year working on the algorithm. Besides highly experienced evaluators, SKIPJACK was subjected to cryptanalysis by less experienced evaluators who were untainted by past approaches. All known methods of attacks were explored, including differential cryptanalysis. The goal was a design that did not allow a shortcut attack." [Denning]

"The design underwent a sequence of iterations based on feedback from the evaluation process. These iterations eliminated properties which, even though they might not allow successful attack, were related to properties that could be indicative of vulnerabilities.

The head of the NSA evaluation team confidently concluded "I believe

that SKIPJACK can only be broken by brute force, there is no better way." [Denning]

"In summary, SKIPJACK is based on some of NSA's best technology. Considerable care went into its design and evaluation in accordance with the care given to algorithms that protect classified data." [Denning]

5. Independent Analysis and Testing

"Our own analysis and testing increased our confidence in the strength of SKIPJACK and its resistance to attack." [Denning]

5.1 Randomness and Correlation Tests

"A strong encryption algorithm will behave like a random function of the key and plaintext so that it is impossible to determine any of the key bits or plaintext bits from the ciphertext bits (except by exhaustive search). We ran two sets of tests aimed at determining whether SKIPJACK is a good pseudo-random number generator. These tests were run on a Cray YMP at NSA. The results showed that SKIPJACK behaves like a random function and that ciphertext bits are not correlated with either key bits or plaintext bits." [Denning]

5.2 Differential Cryptanalysis

"Differential cryptanalysis is a powerful method of attack that exploits structural properties in an encryption algorithm. The

method involves analyzing the structure of the algorithm in order to determine the effect of particular differences in plaintext pairs on the differences of their corresponding ciphertext pairs, where the differences are represented by the exclusive-or of the pair. If it is possible to exploit these differential effects in order to determine a key in less time than with exhaustive search, an encryption algorithm is said to be susceptible to differential cryptanalysis. However, an actual attack using differential cryptanalysis may require substantially more chosen plaintext than can be practically acquired." [Denning]

"We examined the internal structure of SKIPJACK to determine its susceptibility to differential cryptanalysis. We concluded it was not possible to perform an attack based on differential cryptanalysis in less time than with exhaustive search." [Denning]

5.3 Weak Key Test

"Some algorithms have "weak keys" that might permit a shortcut solution. We saw no pattern of symmetry in the SKIPJACK algorithm which could lead to weak keys. We also experimentally tested the all "0" key (all 80 bits are "0") and the all "1" key to see if they were weak and found they were not." [Denning]

5.4 Symmetry Under Complementation Test

"An algorithm may satisfy the property that for a given plaintext-ciphertext pair and associated key, encryption of the one's complement of the plaintext with the one's complement of the key yields the one's complement of the ciphertext. This "complementation property" shortens an attack by exhaustive search by a factor of two since half the keys can be tested by computing complements in lieu of performing a more costly encryption. We tested SKIPJACK for this property and found that it did not hold." [Denning]

5.5 Comparison with Classified Algorithms

"We compared the structure of SKIPJACK to that of NSA Type I algorithms used in current and near-future devices designed to protect classified data. This analysis was conducted with the close assistance of the cryptographer who developed SKIPJACK and included an in-depth discussion of design rationale for all of the algorithms involved. Based on this comparative, structural analysis of SKIPJACK against these other algorithms, and a detailed discussion of the similarities and differences between these algorithms, our confidence in the basic soundness of SKIPJACK was further increased." [Denning]

"Conclusion 2: There is no significant risk that SKIPJACK can be broken through a shortcut method of attack." [Denning]

6. Randomness of the Initialization Variable

"The case for using random values for IVs has been refined with the following summary provided by Steve Bellovin. Refer to [Bell97] for further information." [Denning]

> "The problem arises if you use a counter as an IV, or some other source with a low Hamming distance between successive IVs, for encryption in CBC mode. In CBC mode, the "effective plaintext" for an encryption is the XOR of the actual plaintext and the ciphertext of the preceding block. Normally, that's a random value, which means that the effective plaintext is quite random. That's good, because many blocks of actual plaintext don't change very much from packet to packet, either.
>
> For the first block of plaintext, though, the IV takes the place of the previous block of ciphertext. If the IV doesn't differ much from the previous IV, and the actual plaintext block doesn't

differ much from the previous packet's, then the effective plaintext won't differ much, either. This means that you have pairs of ciphertext blocks combined with plaintext blocks that differ in just a few bit positions. This can be a wedge for assorted cryptanalytic attacks."

"The discussion on IVs has been updated to require that an implementation not use a low-Hamming distance source for IVs." [Denning]

7. References

[Denning] Denning, D. E.; Brickell, E. F.; Kent, S. T.; Maher, D. P.; Tuchman, W. ; "SKIPJACK Review Interim Report The SKIPJACK Algorithm", July 28, 1993, http://www.vortex.com/privacy/SKIPJACK.1

[FIPS-185] National Institute of Standards and Technology, "Escrowed Encryption Standard (EES)", Federal Information Processing Standard (FIPS) Publication 185, February 1994, available at http://www.itl.nist.gov/div897/pubs/fip185.htm.

[DSS] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", Federal Information Processing Standard (FIPS) Publication 186, May 1994, available at http://www.itl.nist.gov/div897/pubs/fip185.htm.

[valid] National Institute of Standards and Technology, "SKIPJACK Validation List", April 1999, available at http://www-08.nist.gov/cryptval/des/skipval.htm.

[skip/kea] "SKIPJACK and KEA Algorithm Specifications" version 2.0, 29 May 1998, available at http://csrc.nist.gov/encryption/SKIPJACK-kea.htm.

[Bell96] Bellovin, S., "Problem Areas for the IP Security Protocols", Proceedings of the Sixth Usenix Security Symposium, July 1996.

[Bell97] Bellovin, S., "Probable Plaintext Cryptanalysis of the IP Security Protocols", Proceedings of the Symposium on Network and Distributed System Security, San Diego, CA, pp. 155-160, February 1997 (also http://www.research.att.com/~smb/papers/probtxt.{ps, pdf}).

[BS93] Biham, E., and Shamir, A., "Differential Cryptanalysis of the Data Encryption Standard", Berlin: Springer-Verlag, 1993.

[Blaze96] Blaze, M., Diffie, W., Rivest, R., Schneier, B.,

Shimomura, T., Thompson, E., Wiener, M., "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security", currently available at http://www.bsa.org/policy/encryption/cryptographers.html.

[RFC-1750] Eastlake, D., Crocker, S., Schiller, J., "Randomness Recommendations for Security", RFC 1750, December, 1994.

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

[Schneier96] Schneier, B., "Applied Cryptography Second Edition", John Wiley & Sons, New York, NY, 1996. ISBN 0-471-12845-7.

[ESP] Kent, S., Atkinson, R., "IP Encapsulating Security Payload (ESP)", draft-ietf-ipsec-esp-v2-06.txt, work in progress, July 1998.

[AH] Kent, S., Atkinson, R., "IP Authentication Header (AH)", draft-ietf-ipsec-auth-header-04.txt, work in progress, February 1998.

[arch] Kent, S., Atkinson, R., "Security Architecture for the Internet Protocol", draft-ietf-ipsec-arch-sec-03.txt, work in progress, February 1998.

[road] Thayer, R., Doraswamy, N., Glenn, R., "IP Security Document Roadmap", draft-ietf-ipsec-doc-roadmap-03.txt, work in progress, February, 1998.

[RANDOM] D. Eastlake, 3rd; S. Crocker; J. Schiller; "Randomness Recommendations for Security" RFC 1750, December 1994

[RFC 1700] Reynolds, J.; Postel, J.; "Assigned Numbers", STD 2, RFC 1700, USC/Information Sciences Institute, October 1994.

8. Acknowledgments

The information provided on security consideration and independent analysis and testing were originated in articles authored by Dorothy Denning and others in [Denning].

9. Author Information

Scott Judy
Mykotronx, Inc.
9861 Broken Land Parkway, #258
Columbia, Md. 21046
USA

Phone: +1 (410) 290-5730
Fax: +1 (410) 290-9546
EMail: sjudy@myko.rainbow.com

Sandra MacGregor
Mykotronx, Inc.
357 Van Ness Way, #200
Torrance, CA 90501
USA

Phone: +1 (301) 533-8100
Fax: +1 (310) 533-0527
EMail: smacgreg@myko.rainbow.com

10. Security Considerations

The SKIPJACK algorithm provides a method for digitally encrypting data.

Implementations must protect the shared private cryptovariable key. Compromise of users's private key permits masquerade.

Implementations must randomly generate initialization vectors (IVs), and padding when randomly padding is used. Also, the generation of the key relies on random numbers. The use of inadequate pseudo-random number generators (PRNGs) to generate cryptographic keys can result in little or no security. An attacker may find it much easier to reproduce the PRNG environment that produced the keys, searching the resulting small set of possibilities, rather than brute force searching the whole key space. The generation of quality random numbers is difficult. RFC 1750 [RANDOM] offers important guidance in this area, and Appendix 3 of FIPS Pub 186 [DSS] provides one quality PRNG technique.

When using previously distributed symmetric key-encryption keys, a key-encryption key is used to encrypt the content-encryption key. If the key-encryption and content-encryption algorithms are different, the effective security is determined by the weaker of the two algorithms. If, for example, a message content is encrypted with

80-bit SKIPJACK key and the SKIPJACK content-encryption key is wrapped with a 40-bit RC2 key, then at most 40 bits of protection is provided. A trivial search to determine the value of the 40-bit RC2 key can recover SKIPJACK key, and then the SKIPJACK key can be used to decrypt the content. Therefore, implementers must ensure that key-encryption algorithms are as strong or stronger than content-encryption algorithms.

Implementers should be aware that cryptographic algorithms become weaker with time. As new cryptoanalysis techniques are developed and computing performance improves, the work factor to break a particular cryptographic algorithm will reduce. Therefore, cryptographic algorithm implementations should be modular allowing new algorithms to be readily inserted. That is, implementers should be prepared for the set of mandatory requirements to implement algorithms to change over time.

11. Intellectual Property Rights

"The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to

be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat."

"The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director."

12. Copyright Section