Network Working Group   P Metzger
Internet Draft   [Piermont]
                                                             W A Simpson
                                                            [DayDreamer]
expires in six months   July 1997

The ESP DES-CBC Transform
draft-ietf-ipsec-ciph-des-derived-00.txt

Status of this Memo

Follows draft-simpson-esp-des1-v2-00.txt.

Abstract

This document describes the DES-CBC block cipher transform interface used with the IP Encapsulating Security Payload (ESP). It provides compatible migration from RFC-1829.

1. Introduction

The Encapsulating Security Payload (ESP) [RFC-1827x] provides confidentiality for IP datagrams by encrypting the payload data to be protected. This specification describes the ESP use of the Cipher Block Chaining (CBC) mode of the US Data Encryption Standard (DES) algorithm [FIPS-46, FIPS-46-1, FIPS-74, FIPS-81].

The level of privacy provided by use of ESP DES-CBC in the Internet environment is far greater than sending the datagram as cleartext. However, in view of the current analysis of DES, it is suggested that DES is not a good encryption algorithm for the protection of even moderate value information for any length of time.

For an explanation of the use of CBC mode with this cipher, see [RFC-wwww].

For more explanation and implementation information for DES, see [Schneier95].

This document assumes that the reader is familiar with the related document "Security Architecture for the Internet Protocol" [RFC-1825x], that defines the overall security plan for IP, and provides important background for this specification.

In this document, the key words "MAY", "MUST", "recommended", "required", and "SHOULD", are to be interpreted as described in [RFC-2119].

1.1. Availability

There were a number of US patents (see [Schneier95] for listing). All patents have expired. Several freely available implementations have been published world-wide.

1.2. Performance

Phil Karn has tuned DES-CBC software to achieve 10.45 Mbps with a 90 MHz Pentium, scaling to 15.9 Mbps with a 133 MHz Pentium. Other DES speed estimates may be found at [Schneier95, page 279]. Your milage may vary.

2. Description

2.1. Block Size

The US Data Encryption Standard (DES) algorithm operates on blocks of 64-bits (8 bytes). This often requires padding before encrypting, and subsequent removal of padding after decrypting.

The output is the same number of bytes that are input. This facilitates in-place encryption and decryption.

2.2. Interaction with Authentication

There is no known interaction of DES with any currently specified Authenticator algorithm. Never-the-less, any Authenticator MUST use a separate and independently generated key.

3. Initialization Vector

DES-CBC requires an Initialization Vector (IV) that is 64-bits (8 bytes) in length [RFC-wwww].

By default, the 64-bit IV is generated from the 32-bit ESP Sequence Number field followed by (concatenated with) the bit-wise complement of the same 32-bit value:

SN || -SN

Alternative IV generation techniques MAY be specified when dynamically configured via a key management protocol.

Security Notes:

Using the Sequence Number provides an easy method for preventing IV repetition, and is sufficiently robust for practical use with the DES algorithm. But, when used alone, cryptanalysis might be aided by the rare serendipitous occurrence when the Sequence Number increments in exactly the same fashion as a corresponding bit position in the first block.

No commonly used IP (Next Header) Protocols exhibit this property. Never-the-less, inclusion of the bit-wise complement ensures that Sequence Number bit changes are reflected twice in the IV.

4. Keys

DES-CBC is a symmetric secret key algorithm. The secret DES key shared between the communicating parties is 56-bits in length. The 56-bit key is stored as a 64-bit (8 byte) quantity, with the least significant bit of each byte used as a parity bit.

4.1. Weak Keys

DES has 64 known weak keys, including so-called semi-weak keys and possibly-weak keys [Schneier95, pp 280-282] (shown in hex with parity bits):

```
0101 0101  0101 0101
1f1f 1f1f  0e0e 0e0e
e0e0 e0e0  f1f1 f1f1
fefe fefe  fefe fefe
```

semi-weak key pairs:

```
01fe 01fe  01fe 01fe    fe01 fe01  fe01 fe01
1fe0 1fe0  0ef1 0ef1    e0f1 e0f1  f10e f10e
01e0 01e0  01f1 01f1    e001 e001  f101 f101
1ffe 1ffe  0efe 0efe    fe1f fe1f  fe0e fe0e
011f 011f  010e 010e    1f01 1f01  0e01 0e01
e0fe e0fe  f1fe f1fe    fee0 fee0  fef1 fef1
```

possibly-weak keys:

```
1f1f 0101  0e0e 0101    e001 01e0  f101 01f1
011f 1f01  010e 0e01    fe1f 01e0  fe0e 01f1
1f01 011f  0e01 010e    fe01 1fe0  fe01 0ef1
0101 1f1f  0101 0e0e    e01f 1fe0  f10e 0ef1
--------------------
e0e0 0101  f1f1 0101    fe01 01fe  fe01 01fe
fefe 0101  fefe 0101    e01f 01fe  f10e 01fe
fee0 1f01  fef1 0e01    e001 1ffe  f101 0efe
e0fe 1f01  f1fe 0e01    fe1f 1ffe  fe0e 0efe
                        --------------------
fee0 011f  fef1 010e    1ffe 01e0  0efe 01f1
e0fe 011f  f1fe 010e    01fe 1fe0  01fe 0ef1
e0e0 1f1f  f1f1 0e0e    1fe0 01fe  0ef1 01fe
fefe 1f1f  fefe 0e0e    01e0 1ffe  01f1 0efe
```

```
fe1f e001  fe0e f101    0101 e0e0  0101 f1f1
e01f fe01  f10e fe01    1f1f e0e0  0e0e f1f1
fe01 e01f  fe01 f1e0    1f01 fee0  0e01 fef1
e001 fe1f  f101 fe0e    011f fee0  010e fef1
--------------------
01e0 e001  01f1 f101    1f01 e0fe  0e01 f1fe
1ffe e001  0efe f101    011f e0fe  010e f1fe
1fe0 fe01  0ef1 fe01    0101 fefe  0101 fefe
01fe fe01  01fe fe01    1f1f fefe  0e0e fefe
                        --------------------
1fe0 e01f  0ef1 f10e    fefe e0e0  fefe f1f1
01fe e01f  01fe f10e    e0fe fee0  f1fe fef1
01e0 fe1f  01f1 fe0e    fee0 e0fe  fef1 f1fe
1ffe fe1f  0efe fe0e    e0e0 fefe  f1f1 fefe
```

Implementations SHOULD take care not to select weak keys [CN94], although the likelihood of picking one at random is negligible.

4.2. Manual Key Management

When configured manually, 64-bits (8 bytes) are configured.

Keys with incorrect parity SHOULD be rejected by the configuration utility, ensuring that the keys have been correctly configured.

The 64 known weak keys SHOULD be rejected.

4.3. Automated Key Management

When configured via a Security Association management protocol, 64-bits (8 bytes) are returned for the key.

The key manager MAY be required to generate the correct parity. Alternatively, the least significant bit of each key byte is ignored, or locally set to parity by the DES implementation.

The 64 known weak keys MUST be rejected.

4.4. Refresh Rate

To prevent differential and linear cryptanalysis of collisions [RFC-wwww], no more than 2**32 plaintext blocks SHOULD be encrypted with the same key. Depending on the average size of the datagrams, the key SHOULD be changed at least as frequently as 2**30 datagrams.

5. ESP Alterations

5.1. ESP Sequence Number

The Sequence Number is a 32-bit (4 byte) unsigned counter. This field protects against replay attacks, and may also be used for synchronization by stream or block-chaining ciphers.

When configured manually, the first value sent SHOULD be a random number. The limited anti-replay security of the sequence of datagrams depends upon the unpredictability of the values.

When configured via an automated Security Association management protocol, the first value sent is 1, unless otherwise negotiated.

Thereafter, the value is monotonically increased for each datagram sent. A replacement SPI SHOULD be established before the value repeats. That is, no more than 2**32 datagrams SHOULD be sent with any single key.

5.2. ESP Padding

The Padding field may be zero or more bytes in length.

Prior to encryption, this field is filled with a series of integer values to align the Pad Length and Payload Type fields at the end of a 64-bit (8 byte) block boundary (measured from the beginning of the Transform Data).

By default, each byte contains the index of the byte. For example, three pad bytes would contain the values 1, 2, 3.

After decryption, this field MAY be examined for a valid series of integer values. Verification of the sequence of values is at the discretion of the receiver.

Operational Considerations

The specification provides only a few manually configurable parameters:

SPI

Manually configured SPIs are limited in range to aid operations. Automated SPIs are pseudo-randomly distributed throughout the remaining 2**32 values.

Default: 0 (none). Range: 256 to 65,535.

SPI LifeTime (SPILT)

Manually configured LifeTimes are generally measured in days. Automated LifeTimes are specified in seconds.

Default: 32 days (2,764,800 seconds). Maximum: 182 days (15,724,800 seconds).

Replay Window

Long term replay prevention requires automated configuration. Also, some earlier implementations used pseudo-random values. This check must only be used with those peers that have implemented this feature.

Default: 0 (checking off). Range: 32 to 256.

Pad Values

New implementations use verifiable values. However, some earlier implementations used pseudo-random values. This check must only be used with those peers that have implemented this feature.

Also, some operations desire additional padding to inhibit traffic analysis.

Default: 0 (checking off). Range: 7 to 255.

Key

The 56-bit key is configured as a 64-bit quantity, with parity included as appropriate.

Each party configures a list of known SPIs and symmetric secret-keys.

In addition, each party configures local policy that determines what access (if any) is granted to the holder of a particular SPI. For example, a party might allow FTP, but prohibit Telnet. Such considerations are outside the scope of this document.

Security Considerations

Users need to understand that the quality of the security provided by this specification depends completely on the strength of the DES algorithm, the correctness of that algorithm's implementation, the security of the Security Association management mechanism and its implementation, the strength of the key [CN94], and upon the correctness of the implementations in all of the participating nodes.

The padding bytes have a predictable value. They provide a small measure of tamper detection on their own block and the previous block in CBC mode. This makes it somewhat harder to perform splicing attacks, and avoids a possible covert channel. This small amount of known plaintext does not create any problems for modern ciphers.

At the time of writing of this document, [BS93] demonstrated a differential cryptanalysis based chosen-plaintext attack requiring 2**47 plaintext-ciphertext block pairs, and [Matsui94] demonstrated a linear cryptanalysis based known-plaintext attack requiring only 2**43 plaintext-ciphertext block pairs. Although these attacks are not considered practical, they must be taken into account.

More disturbingly, [Weiner94] has shown the design of a DES cracking machine costing $1 Million that can crack one key every 3.5 hours. This is an extremely practical attack.

One or two blocks of known plaintext suffice to recover a DES key. Because IP datagrams typically begin with a block of known and/or guessable header text, frequent key changes will not protect against this attack.

Changes from RFC-1829:

This specification results in the same default bits-on-the-wire as the 32-bit IV calculation method of RFC-1829. The 32-bit field is semantically identical to a Sequence Number when implemented as a counter (the recommended method).

The 64-bit explicit IV option is deprecated, as no hardware manufacturers were found that required it. It does not meet 64-bit field alignment expectations of IPv6, it is a cryptographically weaker construct than a calculated IV [Bellovin96], and it conflicts with the use of a Sequence Number immediately following the SPI.

Padding is a known series of integers, that may be checked upon receipt.

Many implementation details by Karn were found to be common to all ESP Ciphers, and are awaiting consolidation in the ESP specification.

Added an operational section.

Updated acknowledgements, references, and contacts.

Reorganized according to the new "road map" document.

Acknowledgements

The basic field naming and layout is based on "swIPe" [IBK93, IB93].

Participants in the IP Security Working Group modified this to a variable number of variable length fields. After a digression spanning 4 years, actual implementors mandated a return to these fewer well-known fields.

Some of the text of this specification was derived from work by Randall Atkinson for the SIP, SIPP, and IPv6 Working Groups.

Perry Metzger provided the original Security Considerations text, some of which is distributed throughout the document.

William Allen Simpson was responsible for the name and semantics of the SPI, the IV calculation technique(s), editing and formatting.

The use of known padding values was suggested in various forms by Robert Baldwin, Phil Karn, and David Wagner. This specification uses Self-Describing-Padding [RFC-1570].

Robert Baldwin, Steve Bellovin, Steve Deering, Karl Fox, Charles Lynn, Cheryl Madson, Craig Metz, Dave Mihelcic, Jeffrey Schiller, Norman Shulman and David Wagner provided useful critiques of earlier versions of this document.

References

[Bellovin96]
Bellovin, S., "Problem Areas for the IP Security Protocols", Proceedings of the Sixth Usenix Security Symposium, July 1996.

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

[CN94] Carroll, J.M., and Nudiati, S., "On Weak Keys and Weak Data: Foiling the Two Nemeses", Cryptologia, Vol. 18 No. 23 pp. 253-280, July 1994.

[FIPS-46]
US National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standard (FIPS) Publication 46, January 1977.

[FIPS-46-1]
US National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standard (FIPS) Publication 46-1, January 1988.

[FIPS-74]
US National Bureau of Standards, "Guidelines for Implementing and Using the Data Encryption Standard", Federal Information Processing Standard (FIPS) Publication 74, April 1981.

[FIPS-81]
US National Bureau of Standards, "DES Modes of Operation" Federal Information Processing Standard (FIPS) Publication 81, December 1980.

[IB93] Ioannidis, J., and Blaze, M., "The Architecture and Implementation of Network-Layer Security Under Unix", Proceedings of the Fourth Usenix Security Symposium, Santa Clara California, October 1993.

[IBK93] Ioannidis, J., Blaze, M., and Karn, P., "swIPe: Network-Layer Security for IP", Presentation at the 26th Internet Engineering Task Force, Columbus Ohio, March 1993.

[Matsui94]
Matsui, M., "Linear Cryptanalysis method for DES Cipher," Advances in Cryptology -- Eurocrypt '93 Proceedings, Berlin:

Springer-Verlag, 1994.

[RFC-1570]
Simpson, W., "PPP LCP Extensions", DayDreamer, January 1994.

[RFC-1825x]
Atkinson, R., "Security Architecture for the Internet Protocol", Naval Research Laboratory, July 1995.

[RFC-1827x]
Simpson, W., "IP Encapsulating Security Protocol (ESP) for implementors", work in progress.

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

[RFC-wwww]
Simpson, W.A, "ESP with Cipher Block Chaining (CBC)", work in progress.

[Schneier95]
Schneier, B., "Applied Cryptography Second Edition", John Wiley & Sons, New York, NY, 1995. ISBN 0-471-12845-7.

[Weiner94]
Wiener, M.J., "Efficient DES Key Search", School of Computer Science, Carleton University, Ottawa, Canada, TR-244, May 1994. Presented at the Rump Session of Crypto '93.

Contacts

Comments about this document should be discussed on the ipsec@tis.com mailing list.

Questions about this document can also be directed to:

Perry Metzger
Piermont Information Systems Inc.
160 Cabrini Blvd., Suite #2
New York, NY 10033

perry@piermont.com

William Allen Simpson
DayDreamer
Computer Systems Consulting Services
1384 Fontaine
Madison Heights, Michigan 48071

wsimpson@UMich.edu
wsimpson@GreenDragon.com (preferred)
bsimpson@MorningStar.com