Network Working Group
Internet Draft
expires in six months

W A Simpson [DayDreamer]

July 1998

ESP with Cipher Block Chaining (CBC)
draft-simpson-cbc-01.txt

Status of this Memo

Copyright Notice

Abstract

This document describes the Cipher Block Chaining (CBC) mode, used by a number of IP Encapsulating Security Payload (ESP) transforms.

1. Introduction

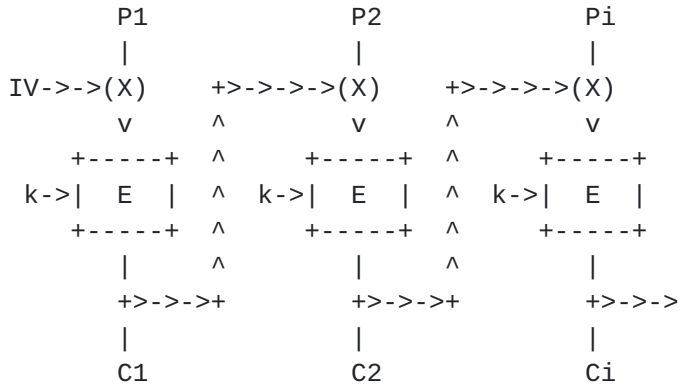
The Encapsulating Security Payload (ESP) [RFC-1827x] provides confidentiality for IP datagrams by encrypting the payload data to be protected. This specification describes the ESP use of the Cipher Block Chaining (CBC) mode.

CBC is used to mask patterns of identical blocks within the same datagram. Together with an Initialization Vector (IV) that is different for every datagram, identical plaintext payloads will each encrypt to different ciphertext payloads. As an added benefit, when the cipher output is effectively random in appearance (a characteristic of a good cipher), masking the plaintext with previous ciphertext will strengthen the entropy of the next input to the cipher.

CBC was first defined for DES in [FIPS-81], and generalized by [ISO-8732] and [ISO/IEC-10116]. For a technical exposition on CBC, see [MOV97]. For more explanation and implementation information for CBC, and a useful comparison with other modes of operation, see [Schneier95].

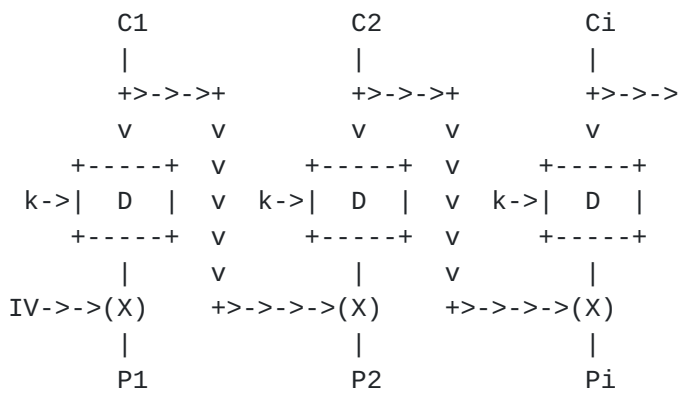
2. Description

2.1. Single Algorithm



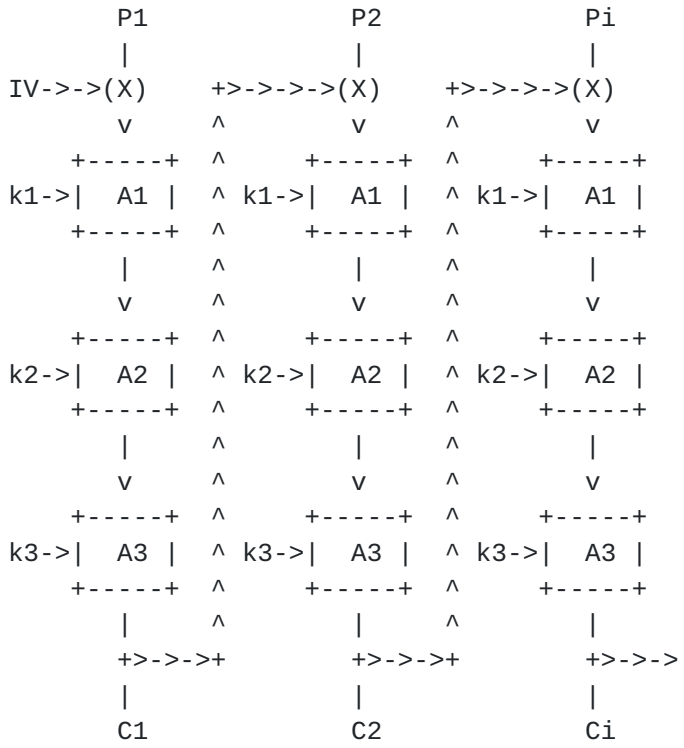
For each datagram, an Initialization Vector (IV) is XOR'd with the first plaintext block (P1). The keyed encryption function (Ek) generates the ciphertext (C1) for the block.

For successive blocks, the previous ciphertext block is XOR'd with the current plaintext (Pi). The keyed encryption function (Ek) generates the ciphertext (Ci) for that block.



To decrypt, the order of the manipulations is reversed (as shown).

2.2. Multiple Algorithms



When using multiple algorithms, the "outer" chaining technique is used.

For each datagram, an Initialization Vector (IV) is XOR'd with the first plaintext block (P1). The series of keyed algorithm functions (Ankn) generate the ciphertext (C1) for the block. Each algorithm uses an independant key.

For successive blocks, the previous ciphertext block is XOR'd with

the current plaintext (Pi). The series of keyed algorithm functions (Ankn) generate the ciphertext (Ci) for that block.

To decrypt, the order of the manipulations and keys is reversed (as shown earlier).

3. Initialization Vector

CBC requires an Initialization Vector (IV). The IV conceals initial blocks that repeat in multiple datagrams.

For ESP, each datagram generates its IV from material carried in the datagram. This ensures that decryption of the received datagram can be performed, even when some datagrams are lost, duplicated, or re-ordered in transit.

Security Notes:

Each IV is intended to be unique over the lifetime of the ESP cipher session-key(s). A counter is most commonly used to generate the IV, providing an easy method to prevent repetition.

However, cryptanalysis might be aided by the rare serendipitous occurrence when the counter repeatedly changes in exactly the same fashion as corresponding bit positions in the first block. Design of specific IV generation techniques must take this into account.

Ideally, the IV would be based on explicit fields carried in each datagram, but generated pseudo-randomly and protected from disclosure [VK83]. This completely protects the first block from undetectable modification. One such method could use the same cipher and key(s) in Electronic CodeBook (ECB) mode, enciphering the ESP Security Parameters Index (SPI) concatenated with the ESP Sequence Number (SN), to generate a keyed hash for an IV.

Incorporating the anti-replay ESP Sequence Number (SN) can provide both uniqueness and mutual protection between the first block and the ESP header. Modification of the SN to avoid anti-replay measures will also prevent correct decryption of the first block, which is most likely to contain datagram headers required for delivery. Attempts to modify the IV to deliberately redirect transport headers will also likely be detected by the transport checksums.

Alternatively, a pseudo-random number generator can be used to generate the IV. Care should be taken that the periodicity of the number generator is long enough to prevent repetition during the

lifetime of the session-key(s).

Historically, another pseudo-random number source has been the final ciphertext block of a previous datagram, extending CBC to an entire stream of data. This is a common link-level configuration, but does not meet the IP requirement to function reliably with lost, duplicated, and re-ordered datagrams. Also, this could be vulnerable to a datagram insertion attack similar to the splicing attack described later.

4. Integrity

CBC does not provide integrity for the datagram. A single ciphertext bit change will affect the current block, and a single corresponding bit of the following block. The remaining blocks will be unaffected, without any subsequent indication of the alteration.

Blocks can be easily appended to the datagram. When a different session-key was used to encrypt the appended blocks, the trailing blocks will be uninterpretable. When the same session-key was applied, even though that session-key is unknown, only the first two appended blocks will be garbage, and the remainder will decrypt correctly. Either case could be detrimental to the intended operations.

Therefore, depending upon the threat environment, when the ESP data is not otherwise verified (externally using AH or internally by the plaintext payload itself), it is recommended (but not required) that an Authenticator be provided.

Security Notes:

Historically, Cipher Block Chaining was designed for uni-directional streams of data. When a block is damaged in transmission, on decryption both it and the following block will be garbled, but all subsequent blocks will automatically be re-synchronized.

The cut and paste splicing attack described by [Bellovin95, Bellovin96] exploits the self-synchronization of CBC. If multiple users of a service have legitimate access to the same key, this feature can be used to insert or replay previously encrypted data of the other users, revealing their original plaintext. The usual (ICMP, TCP, UDP) transport checksum can detect this attack, but on its own is not considered cryptographically strong. In this situation, user or connection oriented integrity checking is needed.

5. Collisions

The "birthday paradox" probability of identical ciphertexts is squareroot(pi/2) * 2**(blocksize/2). Additional 2**(blocksize/2+n) ciphertexts yield 2**(2**n) collisions.

Each such collision reveals a linear relation between two (random) unknown plaintexts and two (random) known ciphertexts. So, an observer learns that Pi = Pj + K for some i, j, and a known constant [Maurer91, Knudsen94].

A datagram generally consists of several ciphertext blocks. The number of datagrams that can be safely exchanged under a single session-key is a function of the total size of the datagrams. Ciphers using CBC need to refresh keys more frequently than might otherwise be expected.

Security Notes:

For a 64-bit block cipher, the basic collision rate is on the order of 48 GigaBytes. While at first glance that might seem like a lot of data, a telephone conversation generates about 7,200 bytes per second, or 26 GigaBytes per hour, not including necessary transport headers. Thus, for this application, the key would require refreshment about once per hour to avoid linear cryptanalysis.

Security Considerations

Specific security limitations are described as notes in the relevant sections.

Acknowledgements

Most of the text of this specification was derived from earlier work by William Allen Simpson and Perry Metzger in multiple Request for Comments.

The mathematical explanation of the collision rate was provided by Bart Preneel, based on "folklore" from the late 1980s and analysis in the early 1990s.

The telephone analogy was provided by Bob Baldwin.

References

[Bellovin95] Bellovin, S., "An Issue With DES-CBC When Used Without Strong Integrity", Presentation at the 32nd Internet Engineering Task Force, Danvers Massachusetts, April 1995.

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

[ISO-8732] "Banking -- Key management (wholesale)", International Organization for Standardization, 1988.

[ISO/IEC-10116] "Information Processing -- Modes of Operation for an n-bit block cipher algorithm", International Organization for Standardization, 1991.

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

[Knudsen94] Knudsen, L., PhD thesis, 1994.

[Maurer91] Maurer, U., "Self-Synchronizing Stream Ciphers", EuroCrypt'91.

[MOV97] Menezes, A.J., van Oorschot, P., and Vanstone, S., "Handbook of Applied Cryptography", CRC Press, 1997.

[RFC-1827x] Atkinson, R., "IP Encapsulating Security Protocol (ESP)", Naval Research Laboratory, July 1995.

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

[VK83] Voydock, V.L., and Kent, S.T., "Security Mechanisms in High-level Networks", ACM Computing Surveys, Vol. 15, No. 2, June 1983.

Contacts

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

Questions about this document can also be directed to:

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

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

Full Copyright Statement