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

The ESP CAST5-128-CBC Transform
draft-ietf-ipsec-ciph-cast-div-00.txt

Status of this Memo

Abstract

This document describes the CAST5-128-CBC block cipher transform interface used with the IP Encapsulating Security Payload (ESP). It provides a full-sized 128-bit key, with a more secure derived initialization variable, and a more efficient smaller datagram size.

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 with CAST5-128.

The CAST Design Procedure was originally developed by Carlisle Adams and Stafford Travares at Queen's University, Kingston, Ontario, Canada. Subsequent enhancements have been made over the years by Carlisle Adams and Michael Wiener of Entrust Technologies. CAST5-128 is the result of applying the CAST Design Procedure as outlined in [RFC-2144].

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

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 are a number of patents. Unfortunately, the CAST authors have not listed them in their drafts, as required as by the IETF. Watch this space.

1.2. Performance

It is speculated that CAST5-128 runs approximately the same speed as a highly optimized DES implementation. This is based on a non-optimized C++ implementation. It is hoped that this can be tuned to give even higher performance.

The following performance tests were run on a Pentium 90 MHz running the Windows NT operating system using 20 Kbyte buffers, and do not include file I/O. The DES-CBC implementation was not optimized for a 32-bit environment.

```
CAST5-64 bit key 12 round CBC encryption ..... 21,120,000 bits/sec
DES-CBC encryption ........................... 4,032,000 bits/sec
```

There is no data available on a full-sized 128-bit key with 16 rounds. Watch this space.

For comparison, 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. Your milage may vary.

2. Description

2.1. Block Size

The CAST5-128 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. Rounds

The algorithm MUST use the full 16 rounds.

2.3. Interaction with Authentication

There is no known interaction of CAST5-128 with any currently specified Authenticator algorithm.

3. Initialization Vector

CAST5-128-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 Security Parameters Index (SPI) field followed by (concatenated with) the 32-bit Sequence Number (SN) field. Then, the bit-wise complement of the 32-bit Sequence Number (SN) value is XOR'd with the first 32-bits (SPI):

```
   (SPI ^ -SN) || SN
```

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

Security Notes:

Incorporating the ESP Security Parameters Index (SPI) and the anti-replay ESP Sequence Number (SN) together can provide greater uniqueness and mutual protection between the first block and the ESP header. Modification of the SPI to alter the decryption key(s) will prevent correct decryption of the first block.

Using the Sequence Number (SN) provides an easy method for preventing IV repetition, and is sufficiently robust for practical use with the CAST5-128 algorithm. Inclusion of the bit-wise complement of SN ensures that bit changes are reflected twice in the IV.

4. Keys

CAST5-128 is a symmetric secret key algorithm. The secret CAST5-128 key shared between the communicating parties is 128-bits in length.

Although CAST5-128 can be used with shorter keys, these other keys sizes are not conformant with this specification.

4.1. Weak Keys

CAST5-128 has no known weak keys.

4.2. Refresh Rate

CAST5-128 is theorized to be immune to differential and linear cryptanalysis.

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.

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, less than 2**32 datagrams SHOULD be sent with any single key.

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

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

Pad Values

Default: 7 (checking on). Range: 7 to 255.

Key

The 128-bit key is configured as required.

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 CAST algorithm, the correctness of that algorithm's implementation, the security of the Security Association management mechanism and its implementation, the strength of the key, and upon the correctness of the implementations in all of the participating nodes.

Acknowledgements

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

Some of the text of this specification was derived from work by Roy Pereira and Greg Carter.

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

References

[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.

[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-2144]
Adams, C., "The CAST-128 Encryption Algorithm", May 1997.

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

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