

Telnet Encryption: DES3 64 bit Cipher Feedback

Status of this Memo

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.

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

Abstract

This document specifies how to use the Triple-DES (data encryption standard) encryption algorithm in cipher feedback mode with the telnet encryption option.

[1.](#) Command Names and Codes

Encryption Type

DES3_CFB64	3
------------	---

Suboption Commands

CFB64_IV	1
CFB64_IV_OK	2
CFB64_IV_BAD	3

[2.](#) Command Meanings

IAC SB ENCRYPT IS DES3_CFB64 CFB64_IV <initial vector> IAC SE

The sender of this command generates a random 8 byte initial vector, and sends it to the other side of the connection using the CFB64_IV command. The initial vector is sent in clear text. Only the side of the connection that is WILL ENCRYPT may send the CFB64_IV command.

[RFC 2947](#)

DES3 64 bit Cipher Feedback

September 2000

```
IAC SB ENCRYPT REPLY DES3_CFB64 CFB64_IV_OK IAC SE IAC SB ENCRYPT  
REPLY DES3_CFB64 CFB64_IV_BAD IAC SE
```

The sender of these commands either accepts or rejects the initial vector received in a CFB64_IV command. Only the side of the connection that is DO ENCRYPT may send the CFB64_IV_OK and CFB64_IV_BAD commands. The CFB64_IV_OK command MUST be sent for backwards compatibility with existing implementations; there really isn't any reason why a sender would need to send the CFB64_IV_BAD command except in the case of a protocol violation where the IV sent was not of the correct length (i.e., 8 bytes).

[3.](#) Implementation Rules

Once a CFB64_IV_OK command has been received, the WILL ENCRYPT side of the connection should do keyid negotiation using the ENC_KEYID command. Once the keyid negotiation has successfully identified a common keyid, then START and END commands may be sent by the side of the connection that is WILL ENCRYPT. Data will be encrypted using the DES3 64 bit Cipher Feedback algorithm.

If encryption (decryption) is turned off and back on again, and the same keyid is used when re-starting the encryption (decryption), the intervening clear text must not change the state of the encryption (decryption) machine.

If a START command is sent (received) with a different keyid, the encryption (decryption) machine must be re-initialized immediately following the end of the START command with the new key and the initial vector sent (received) in the last CFB64_IV command.

If a new CFB64_IV command is sent (received), and encryption (decryption) is enabled, the encryption (decryption) machine must be re-initialized immediately following the end of the CFB64_IV command with the new initial vector, and the keyid sent (received) in the last START command.

If encryption (decryption) is not enabled when a CFB64_IV command is sent (received), the encryption (decryption) machine must be re-initialized after the next START command, with the keyid sent (received) in that START command, and the initial vector sent (received) in this CFB64_IV command.

The following rules are to be followed for creating three DES encryption keys based upon the available encrypt key data:

`keys_to_use = bytes of key data / DES block size (8 bytes)`

where the keys are labeled "key1" through "key6" with "key1" being the first 8 bytes; "key2" the second 8 bytes; ... and "key6" being sixth 8 bytes (if available).

When two keys are available:

- . data sent from the server is encrypted with key1, decrypted with key2, and encrypted with key1;
- . data sent from the client is encrypted with key2, decrypted with key1, and encrypted with key2

When three keys are available:

- . data sent from the server is encrypted with key1, decrypted with key2, and encrypted with key3;
- . data sent from the client is encrypted with key2, decrypted with key3, and encrypted with key1

When four keys are available:

- . data sent from the server is encrypted with key1, decrypted with key2, and encrypted with key3;
- . data sent from the client is encrypted with key2, decrypted with key4, and encrypted with key1

When five keys are available:

- . data sent from the server is encrypted with key1, decrypted with key2, and encrypted with key3;

- . data sent from the client is encrypted with key2, decrypted with key4, and encrypted with key5

When six keys are available:

- . data sent from the server is encrypted with key1, decrypted with key2, and encrypted with key3;
- . data sent from the client is encrypted with key4, decrypted with key5, and encrypted with key6

In all cases, the keys used by DES3_CFB64 must have their parity corrected after they are determined using the above algorithm.

Note that the above algorithm assumes that it is safe to use a non-DES key (or part of a non-DES key) as a DES key. This is not necessarily true of all cipher systems, but we specify this behaviour as the default since it is true for most authentication systems in popular use today, and for compatibility with existing

implementations. New telnet AUTHENTICATION mechanisms may specify alternative methods for determining the keys to be used for this cipher suite in their specification, if the session key negotiated by that authentication mechanism is not a DES key and where this algorithm may not be safely used.

[6.](#) Security Considerations

Encryption using Cipher Feedback does not ensure data integrity; the active attacker has a limited ability to modify text, if he can predict the clear-text that was being transmitted. The limitations faced by the attacker (that only 8 bytes can be modified at a time, and the following 8-byte block of data will be corrupted, thus making detection likely) are significant, but it is possible that an active attacker still might be able to exploit this weakness.

The tradeoff here is that adding a message authentication code (MAC) will significantly increase the number of bytes needed to send a single character in the telnet protocol, which will impact performance on slow (i.e. dialup) links.

[7.](#) Acknowledgments

This document was based on the "Telnet Encryption: DES 64 bit Cipher Feedback" document originally written by Dave Borman of Cray Research with the assistance of the IETF Telnet Working Group.

Author's Address

Jeffrey Altman, Editor
Columbia University
612 West 115th Street Room 716
New York NY 10025 USA

Phone: +1 (212) 854-1344
EMail: jaltman@columbia.edu

Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be

followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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