

Internet Draft
[draft-hoffman-des40-01.txt](#)

Paul Hoffman
Internet Mail Consortium
Russ Housley
SPYRUS
Expires six months later

April 20, 1998

Creating 40-Bit Keys for DES

Status of this memo

This document is an Internet-Draft. 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."

To view the entire list of current Internet-Drafts, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nis.garr.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

1. Introduction

This document describes an method for shortening DES keys from 56 bits to 40 bits. The shortened keys are generally known as "DES-40". The motivation for this weakening is that some localities (such as the United States) give special preference to applications that use 40-bit keys. The weakened keys are then used with the DES encryption algorithm in the same manner as full-strength keys.

There are many possible methods for reducing a 56-bit key to a 40-bit key. The method in this draft was chosen because one method is needed for interoperability. Further, this method has been known to occasionally have been approved for export from the United States.

2. Creating 40-Bit Keys for DES

DES [[DES](#)] uses a 56-bit key. The key consists of eight 8-bit bytes; however the last (eighth) bit of each byte is used for parity, leaving [56](#) bits of key.

To weaken the 8-byte, 56-bit key into a 40-bit key, you set to zero the first four bits of every other byte in the key, starting with the first byte. Stated a different way, you take the bitwise logical AND

of the key and the binary value:

```
0000111111111111000011111111111100001111111111110000111111111111
```

Another way to picture this is:

Bit positions:

```
000000000011111111112222222222333333333333444444444455555555556666
```

```
0123456789012345678901234567890123456789012345678901234567890123
```

Use:

```
zzzzKKKpKKKKKKKpzzzzKKKpKKKKKKKpzzzzKKKpKKKKKKKpzzzzKKKpKKKKKKKp
```

Legend:

z = zero bit

K = key bit

p = parity bit

Some implementations of DES require the parity bit of each byte to be set correctly in order for the key to be accepted. DES requires that the last bit of each byte be a parity bit. DES uses odd parity, meaning that the number of 1 bits in each byte should be odd. Therefore, to complete the transformation to a 40-bit key, the software SHOULD cause the parity in each byte to be odd, changing the last bit if necessary.

[3. Security Considerations](#)

Current computer technology makes a brute-force attack on ciphertext that is encrypted with a 40-bit key fairly quick. This is true for any encryption algorithms, not just DES. Thus, 40-bit keys result in only weak security against decryption. As computers get faster, this weak security will become even weaker. Thus, 40-bit keys should never be used with data that has a high value if it is decrypted by an adversary. However, encrypting data with 40-bit keys prevents passive snoopers from immediately reading a message without using some significant but not onerous decryption effort.

The shortening method described in this draft causes a discernable pattern of zero bits in the resulting key. There is no known literature at this time that describes whether cyphertext encrypted with a key that has this pattern of zeros is easier to decrypt than cyphertext that has no pattern. However, because 40-bit keys are already inherently weak, a decrease in security from the pattern is not considered to be very important relative to the inherent weakness due to the short key length.

There are other methods for converting longer keys to shorter ones. For example, IBM has created a patented (and significantly more complex) method called "Commercial Data Masking Facility", or CDMF [[CDMF](#)]; other methods probably exist. These methods might result in keys that produce cyphertext that is harder (or easier) to determine through brute-force. A quick comparison of CDMF and DES-40 shows that

the brute-force attack against CDMF require one additional DES operation. Saving one DES operation does not seem to warrant the additional complexity.

[A](#). References

[CDMF] "Design of the Commercial Data Masking Facility Data Privacy Algorithm", 1st ACM Conference on Computer and Communications Security, ACM Press, 1993.

[DES] ANSI X3.106, "American National Standard for Information Systems-Data Link Encryption," American National Standards Institute, 1983.

[B](#). Authors' Addresses

Paul Hoffman
Internet Mail Consortium
[127](#) Segre Place
Santa Cruz, CA 95060
(408) 426-9827
phoffman@imc.org

Russ Housley
SPYRUS
[381](#) Elden Street, Suite 1120
Herndon, VA 20170
housley@spyrus.com