Cryptography FAQ (05/10: Product Ciphers)

Message-ID: <cryptography-faq/part05_1040202968@rtfm.mit.edu>
Supersedes: <cryptography-faq/part05_1038303482@rtfm.mit.edu>
Expires: 22 Jan 2003 09:16:08 GMT
References: <cryptography-faq/part01_1040202968@rtfm.mit.edu>
X-Last-Updated: 1994/07/05
Newsgroups:
sci.crypt,talk.politics.crypto,sci.answers,news.answers,talk.answers
Subject: Cryptography FAQ (05/10: Product Ciphers)
Followup-To: poster
From: crypt-comments@math.ncsu.edu
Organization: The Crypt Cabal
Reply-To: crypt-comments@math.ncsu.edu
Date: 18 Dec 2002 09:16:16 GMT
X-Trace: 1040202976 senator-bedfellow.mit.edu 3948 18.181.0.29

Archive-name: cryptography-faq/part05
Last-modified: 94/06/07

This is the fifth of ten parts of the sci.crypt FAQ. The parts are mostly independent, but you should read the first part before the rest. We don't have the time to send out missing parts by mail, so don't ask. Notes such as ``[KAH67]'' refer to the reference list in the last part.

The sections of this FAQ are available via anonymous FTP to rtfm.mit.edu as /pub/usenet/news.answers/cryptography-faq/part[xx]. The Cryptography FAQ is posted to the newsgroups sci.crypt, talk.politics.crypto, sci.answers, and news.answers every 21 days.

Contents:

5.13. Can DES be used to protect classified information?
5.14. What are ECB, CBC, CFB, OFB, and PCBC encryption?

5.1. What is a product cipher?

A product cipher is a block cipher that iterates several weak operations such as substitution, transposition, modular addition/multiplication, and linear transformation. (A ``block cipher'' just means a cipher that encrypts a block of data---8 bytes, say---all at once, then goes on to the next block.) The notion of product ciphers is due to Shannon [SHA49]. Examples of modern product ciphers include LUCIFER [SOR84], DES [NBS77], SP-networks [KAM78], LOKI [BRO90], FEAL [SHI84], PES [LAI90], Khufu and Khafre [ME91a]. The so-called Feistel ciphers are a class of product ciphers which operate on one half of the ciphertext at each round, and then swap the ciphertext halves after each round. LUCIFER, DES, LOKI, and FEAL are examples of Feistel ciphers.

The following table compares the main parameters of several product ciphers:

cipher	block length	key bits	number of rounds
LUCIFER	128	128	16
DES	64	56	16
LOKI	64	64	16
FEAL	64	128	2^x, x >= 5
PES	64	128	8

5.2. What makes a product cipher secure?

Nobody knows how to prove mathematically that a product cipher is completely secure. So in practice one begins by demonstrating that the cipher ``looks highly random''. For example, the cipher must be nonlinear, and it must produce ciphertext which functionally depends on every bit of the plaintext and the key. Meyer [MEY78] has shown that at least 5 rounds of DES are required to guarantee such a dependence. In this sense a product cipher should act as a ``mixing'' function which combines the plaintext, key, and ciphertext in a complex nonlinear fashion.

The fixed per-round substitutions of the product cipher are referred to as S-boxes. For example, LUCIFER has 2 S-boxes, and DES has 8 S-boxes. The nonlinearity of a product cipher reduces to a careful design of these S-boxes. A list of partial design criteria for the S-boxes of DES, which apply to S-boxes in general, may be found in Brown [BRO89] and Brickell et al. [BRI86].

5.3. What are some group-theoretic properties of product ciphers?

Let E be a product cipher that maps N-bit blocks to N-bit blocks. Let E_K(X) be the encryption of X under key K. Then, for any fixed K, the map sending X to E_K(X) is a permutation of the set of N-bit blocks. Denote this permutation by P_K. The set of all N-bit permutations is called the symmetric group and is written S_{2^N}. The collection of all these permutations P_K, where K ranges over all possible keys, is denoted $E(S_{2^N})$. If E were a random mapping from plaintexts to ciphertexts then we would expect $E(S_{2^N})$ to generate a large subset of S_{2^N}.

Coppersmith and Grossman [COP74] have shown that a very simple product cipher can generate the alternating group A_{2^N} given a sufficient number of rounds. (The alternating group is half of the symmetric group: it consists of all ``even'' permutations, i.e., all permutations which can be written as an even number of swaps.) Even and Goldreich [EVE83] were able to extend these results to show that Feistel ciphers can generate A_{2^N}, given a sufficient number of rounds.

The security of multiple encipherment also depends on the group-theoretic properties of a cipher. Multiple encipherment is an extension over single encipherment if for keys K1, K2 there does not exist a third key K3 such that

$$E_{K2}(E_{K1}(X)) == E_{(K3)}(X) \qquad (**)$$

which indicates that encrypting twice with two independent keys K1, K2 is equal to a single encryption under the third key K3. If for every K1, K2 there exists a K3 such that eq. (**) is true then we say that E is a group.

This question of whether DES is a group under this definition was extensively studied by Sherman, Kaliski, and Rivest [SHE88]. In their paper they give strong evidence for the hypothesis that DES is not a group. In fact DES is not a group [CAM93].

5.4. What can be proven about the security of a product cipher?

Recall from above that P_K is a permutation produced by E under some key K. The goal of the designer of E is to ensure that P_K appears to be a random element of S_{2^N}, the symmetric group. Let R be an element of S_{2^N} selected randomly. We will say that P_K and R are indistinguishable if an observer given P_K and R in some order cannot distinguish between these two permutations in polynomial time. That is, with time bounded resources, the observer cannot determine which of the permutations is produced by E: the optimal decision is no better than simply guessing.

Luby and Rackoff [LUB88] have shown that a class of Feistel ciphers are secure in this sense when the round mapping is replaced by random boolean functions.

5.5. How are block ciphers used to encrypt data longer than the block size?

There are four standard ``modes of operation'' (and numerous non-standard ones as well). The standard modes of operation are defined in the U.S. Department of Commerce Federal Information Processing Standard (FIPS) 81, published in 1980. See the question about ECB below for more details.

Although they are defined for the DES block cipher, the ``modes of operation'' can be used with any block cipher.

5.6. Can symmetric block ciphers be used for message authentication?

You may use a symmetric cryptosystem block cipher to prove to yourself that you generated a message, and that the message wasn't altered after you created it. But you cannot prove these things to anyone else without revealing your key. Thereafter you cannot prove anything about messages authenticated with that key.

See ANSI X3.106-1983 and FIPS 113 (1985) for a standard method of message authentication using DES.

5.7. What exactly is DES?

DES is the U.S. Government's Data Encryption Standard, a product cipher that operates on 64-bit blocks of data, using a 56-bit key.

It is defined in FIPS 46-1 (1988) [which supersedes FIPS 46 (1977)]. FIPS are Federal Information Processing Standards published by NTIS. DES is identical to the ANSI standard Data Encryption Algorithm (DEA) defined in ANSI X3.92-1981.

5.8. What is triple DES?

Triple DES is a product cipher which, like DES, operates on 64-bit data blocks. There are several forms, each of which uses the DES cipher 3 times. Some forms use two 56-bit keys, some use three. The DES ``modes of operation'' may also be used with triple-DES.

Some people refer to E(K1,D(K2,E(K1,x))) as triple-DES.

This method is defined in chapter 7.2 of the ANSI standard X9.17-1985 ``Financial Institution Key Management'' and is intended for use in encrypting DES keys and IVs for ``Automated Key Distribution''. Its formal name is ``Encryption and Decryption of a Single Key by a Key Pair'', but it is referenced in other standards documents as EDE.

That standard says (section 7.2.1): ``Key encrypting keys may be a single DEA key or a DEA key pair. Key pairs shoud be used where additional security is needed (e.g., the data protected by the key(s) has a long security life). A key pair shall not be encrypted or decrypted using a single key.''

Others use the term ``triple-DES'' for E(K1,D(K2,E(K3,x))) or E(K1,E(K2,E(K3,x))).

All of these methods are defined only for ECB mode of operation. The security of various methods of achieving other modes of operation (such as CBC) is under study at the moment. For now, it should be assumed that other modes be defined as they are today, but with E(K1,D(K2,E(K1,x))) as the block cipher within the feedback mechanism creating the mode.

One of us (Ellison) has long advocated triple DES use in the form

```
E(K1, Tran( E(K2, Tran( E(K3, Compress( x )))))),
```

where each DES instance has its own key and IV (for CBC mode) and Tran is a large-block transposition program. Tran is available from [FTPTR]. This claims to gain security by diffusing single bit changes over a much larger block (Tran's block size). Other compositions of weak ciphers with DES are possible. For example, one could use:

```
E(K1, Prngxor(K4, Tran( E(K2, Tran( Prngxor(K5, E(K3, Compress( x )))))))),
```

where Prngxor() [FTPPX] is a simple stream cipher driven from a long-period pseudo-random number generator (PRNG), to make sure that all plaintext or ciphertext patterns are hidden while permitting the use of ECB mode for DES (since there are certain weaknesses in the use of inner CBC loops for multiple-DES, under some attacks, and we do not yet know if these show up under composition with Tran()).

5.9. What is differential cryptanalysis?

Differential cryptanalysis is a statistical attack that can be applied to any iterated mapping (i.e., any mapping which is based on a repeated round function). The method was recently popularized by Biham and Shamir [BIH91], but Coppersmith has remarked that the S-boxes of DES were optimized against this attack some 20 years ago. This method has proved effective against several product ciphers, notably FEAL [BI91a].

Differential cryptanalysis is based on observing a large number of ciphertexts Y, Y' whose corresponding plaintexts X, X' satisfy a known difference D = X+X', where + is componentwise XOR. In the basic Biham-Shamir attack, 2^{47} such plaintext pairs are required to determine the key for DES. Substantially fewer pairs are required if DES is truncated to 6 or 8 rounds. In these cases, the actual key can be recovered in a matter of minutes using a few thousand pairs. For full DES this attack is impractical because it requires so many known plaintexts.

The work of Biham and Shamir on DES revealed several startling observations on the algorithm. Most importantly, if the key schedule was removed from DES and a 16*48 = 768-bit key was used, the key could be recovered in less than 2^{64} steps. Thus independent subkeys do not add substantial security to DES. Further, the S-boxes of DES are extremely sensitive in that changing even single entries in these tables yields significant improvement in the differential attack.

Adi Shamir is quoted to say (NYTimes Oct 13 1991), ``I would say that, contrary to what some people believe, there is no evidence of tampering with the DES so that the basic design was weakened.''

5.10. How was NSA involved in the design of DES?

According to Kinnucan [KIN78], Tuchman, a member of the group that developed DES at IBM is quoted as saying, ``We developed the DES algorithm entirely within IBM using IBMers. The NSA did not dictate a single wire!'' Tuchman and Meyer (another developer of DES) spent a year breaking ciphers and finding weaknesses in Lucifer. They then spent two years strengthening Lucifer. ``Their basic approach was to look for strong substitution, permutation, and key scheduling functions ... IBM has classified the notes containing the selection criteria at the request of the NSA.... `The NSA told us we had inadvertently reinvented some of the deep secrets it uses to make its own algorithms,' explains Tuchman.''

On the other hand, a document called ``Involvement of the NSA in the development of DES: unclassified summary of the United States Select Committee on Intelligence'', printed in the IEEE Communications Magazine, p53-55, 1978, states: ``In the development of DES, NSA convinced IBM that a reduced keysize was sufficient; indirectly assisted in the development of the S-box structures; and certified that the final DES algorithm was, to the best of their knowledge, free from any statistical or mathematical weakness.''

Clearly the key size was reduced at the insistence of the NSA. The article further states that the NSA did not tamper with the algorithm itself, just the parameters, which in some sense resolves the apparent conflict in the remarks of Meyer and Tuchman presented above.

5.11. Is DES available in software?

Several people have made DES code available via ftp (see part 10 for pathnames): Stig Ostholm [FTPSO]; BSD [FTPBK]; Eric Young [FTPEY]; Dennis Furguson [FTPDF]; Mark Riordan [FTPMR]; Phil Karn [FTPPK]. A Pascal listing of DES is also given in Patterson [PAT87]. Antti Louko <alo@kampi.hut.fi> has written a version of DES with BigNum packages in [FTPAL].

FIPS 46-1 says ``The algorithm specified in this standard is to be implemented ... using hardware (not software) technology. ... Software implementations in general purpose computers are not in compliance with this standard.'' Despite this, software implementations abound, and are used by government agencies.

5.12. Is DES available in hardware?

The following paragraphs are quoted from messages sent to the editors. We don't vouch for the quality or even existence of the products.

Christian Franke, franke@informatik.rwth-aachen.de, says: ``1. Cryptech CRY12C102: 22.5Mbit/s according to Data Sheet, with 32 Bit interface. We use this one, because it was the only one available when we started the project. No problems ! 2. Pijnenburg PCC100: 20Mbit/s according to Data Sheet. Address: PIJNENBURG B.V., Boxtelswweg 26, NL-5261 NE Vught, The Netherlands. 3. INFOSYS DES Chip (Germany): S-Boxes must be loaded by software. So you can modify the Algorithm. Sorry, I don't have the data sheet handy. Please E-Mail me if you need further information.''

Marcus J Ranum, mjr@tis.com, says: ``SuperCrypt'' 100Mb/sec and faster DES and Proprietary Storage for 16 56-bit keys Key stream generator Integrated hardware DES3 procedure Extended mode with 112 bit keys; Computer Elektronik Infosys; 512-A Herndon Parkway,; Herndon, VA 22070; 800-322-3464.

Tim Hember, thember@gandalf.ca, says: Newbridge Microsystems sells an AM9568 compatible DES chip that operates at 25MHz, performs a round of encryption in 18 clocks, has a three-stage pipeline, supports ECB, CBC, CFB-8 and >>> CFB-1 <<<<. Further it is very reasonable priced as opposed to other high-end DES chips. Call Newbridge Microsystems, Ottawa, 613-592-0714. (... there are no import/export issues with Canada and the US). If you require custom DES or Public Key ICs then Timestep Engineering developed Newbridge's crypto chips and ICs for other commercial and educational establishments. They can be reached at 613-820-0024.

5.13. Can DES be used to protect classified information?

DES is not intended to protect classified data. FIPS 46-1 says: ``This standard will be used by Federal departments and agencies for the cryptographic protection of computer data when the following conditions apply: 1. ... cryptographic protection is required; and 2. the data is not classified according to the National Security Act of 1947, as amended, or the Atomic Energy Act of 1954, as amended.''

5.14. What are ECB, CBC, CFB, OFB, and PCBC encryption?

These are methods for using block ciphers, such as DES, to encrypt messages, files, and blocks of data, known as ``modes of operation.'' Four ``modes of operation'' are defined in FIPS 81 (1980 December 2), and also in ANSI X3.106-1983.

FIPS 81 specifies that when 7-bit ASCII data is sent in octets, the unused most-significant bit is to be set to 1.

FIPS 81 also specifies the padding for short blocks.

The four FIPS/ANSI standard DES modes of operation are:

Electronic Code Book (ECB),
Cipher Block Chaining (CBC),
K-bit Cipher FeedBack (CFB), and
K-bit Output FeedBack (OFB).

All four of the ANSI/FIPS modes have very little "error extension". For a single bit error in the cipherstream, none of them produce an error burst in the decrypted output stream of longer than 128 bits.

A fifth mode of operation, used in Kerberos and elsewhere but not defined in any standard, is error-Propagating Cipher Block Chaining (PCBC). Unlike the 4 standard modes, PCBC extends or propagates the effect of a single bit error in the cipherstream throughout remainder of the decrypted textstream after the point of error.

These 5 methods are explained below in a C-language-like notation.

Some symbols:

P[n] The n'th block of plaintext, input to encryption, output from decryption. Size of block determined by the mode.

C[n] The n'th block of ciphertext, output from encryption, input to decryption. Size of block determined by the mode.

E(m) The DES encryption function, performed on 64-bit block m, using the 16-key schedule derived from some 56-bit key.

D(m) The DES decryption function, performed on 64-bit block m, using the same key schedule as in E(m), except that the 16 keys in the schedule are used in the opposite order as in E(m).

IV A 64-bit ``initialization vector'', a secret value which, along with the key, is shared by both encryptor and decryptor.

I[n] The n'th value of a 64-bit variable, used in some modes.
R[n] The n'th value of a 64-bit variable, used in some modes.

LSB(m,k) The k least significant (right-most) bits of m.
e.g. m & ((1 << k) - 1)

MSB(m,k) The k most significant (left-most) bits of m.
e.g. (m >> (64-k)) & ((1 << k) - 1)

= ^ << >> & operators as defined in the c langage.

Electronic Code Book (ECB):

P[n] and C[n] are each 64-bits long.

```
Encryption:                     Decryption:
C[n] = E(P[n])                  P[n] = D(C[n])
```

Cipher Block Chaining (CBC):

```
        P[n] and C[n] are each 64-bits long.

        Encryption:                      Decryption:
        C[0] = E(P[0]^IV)                P[0] = D(C[0])^IV
(n>0)   C[n] = E(P[n]^C[n-1])            P[n] = D(C[n])^C[n-1]
```

Propagating Cipher Block Chaining (PCBC):

```
        P[n] and C[n] are each 64-bits long.

        Encryption:                      Decryption:
        C[0] = E(P[0]^IV)                P[0] = D(C[0])^IV
(n>0)   C[n] = E(P[n]^P[n-1]^C[n-1])    P[n] = D(C[n])^P[n-1]^C[n-1]
```

k-bit Cipher FeedBack (CFB):

```
        P[n] and C[n] are each k bits long, 1 <= k <= 64.

        Encryption:                      Decryption:
        I[0] = IV                        I[0] = IV
(n>0)   I[n] = I[n-1]<<k | C[n-1]        I[n] = I[n-1]<<k | C[n-1]
(all n) R[n] = MSB(E(I[n]),k)            R[n] = MSB(E(I[n]),k)
(all n) C[n] = P[n]^R[n]                 P[n] = C[n]^R[n]

        Note that for k==64, this reduces to:

        I[0] = IV                        I[0] = IV
(n>0)   I[n] = C[n-1]                    I[n] = C[n-1]
(all n) R[n] = E(I[n])                   R[n] = E(I[n])
(all n) C[n] = P[n]^R[n]                 P[n] = C[n]^R[n]
```

CFB notes: Since I[n] depends only on the plain or cipher text from the previous operation, the E() function can be performed in parallel with the reception of the text with which it is used.

k-bit Output FeedBack (OFB):

```
        P[n] and C[n] are each k bits long, 1 <= k <= 64.

        Encryption:                      Decryption:
        I[0] = IV                        I[0] = IV
(n>0)   I[n] = I[n-1]<<k | R[n-1]        I[n] = I[n-1]<<k | R[n-1]
(all n) R[n] = MSB(E(I[n]),k)            R[n] = MSB(E(I[n]),k)
(all n) C[n] = P[n]^R[n]                 P[n] = C[n]^R[n]

        Note that for k==64, this reduces to:

        I[0] = IV                        I[0] = IV
(n>0)   I[n] = R[n-1]                    I[n] = R[n-1]
(all n) R[n] = E(I[n])                   R[n] = E(I[n])
(all n) C[n] = P[n]^R[n]                 P[n] = C[n]^R[n]
```

```
  OFB notes: encryption and decryption are identical. Since I[n] is
  independent of P and C, the E() function can be performed in advance
of
  the receipt of the plain/cipher text with which it is to be used.


  Additional notes on DES ``modes of operation'':

  ECB and CBC use E() to encrypt and D() to decrypt, but the feedback
  modes use E() to both encrypt and decrypt. This disproves the
following
  erroneous claim: ``DES implementations which provide E() but not D()
  cannot be used for data confidentiality.''
```

Send corrections/additions to the FAQ Maintainer:
crypt-comments@math.ncsu.edu

Last Update January 01 2003 @ 00:33 AM