Decorrelated Fast Cipher: an AES Candidate

Extended Abstract

Henri Gilbert[2], Marc Girault[2], Philippe Hoogvorst[1], Fabrice Noilhan[1], Thomas Pornin[1], Guillaume Poupard[1], Jacques Stern[1], and Serge Vaudenay[1]

[1] Ecole Normale Supérieure – CNRS
[2] France Telecom

Contact e-mail: `Serge.Vaudenay@ens.fr`

Version: 1998, July 12th

Abstract. This report presents a response to the call for candidates issued by the National Institute for Standards and Technologies (the Advanced Encryption Standard project). The proposed candidate — called DFC as for "Decorrelated Fast Cipher" — is based on the recent decorrelation technique. This provides provable security against several classes of attacks which include Differential Cryptanalysis and Linear Cryptanalysis.

Digital criminality is nowadays a big threat for the electronic marketplace. For this reason, cryptography provides various algorithms based on a heart cryptographic primitive: encryption. The Digital Encryption Standard (DES) has been developed by IBM™ for the US Department of Commerce in the seventies for this purpose, but its secret-key length (56 bits) provides no sufficient security at this time, so this standard is now over.

So far, real-life encryption algorithms used to have an empirical-based security: they were designed from an intricate substitution-permutation network and believed to be secure until someone published an attack on them. In parallel, research yielded several general attacks strategies, namely Biham and Shamir's "differential cryptanalysis" and Matsui's "linear cryptanalysis" (both are particular cases of the more general "iterated attacks of order 2"), which provided a better understanding on how to manage with security arguments.

The laboratory of computer sciences of the *Ecole Normale Supérieure*, associated with the *Centre National pour la Recherche Scientifique* (CNRS), has recently developed a technique for making new encryption algorithms with a **provable security** against any iterated attacks of a fixed order (*e.g.* of order 2). Several properties of this technique — known as **decorrelation** — have been presented at international research conferences. In this extended abstract, we present a candidate for the "Advanced Encryption Standard" process of the US Department of Commerce, and which is based on decorrelation. We call it DFC as for "Decorrelated Fast Cipher".

DFC enables to encrypt any digital information with a key of length up to 256 bits. It has been implemented on various computer platforms with the following benchmarks.

microprocessor	cycles-per-bit	clock-frequency	bits-per-second
AXP™	4.36	600MHz	137.6Mbps
Pentium™	5.89	200MHz	34.0Mbps
SPARC™	6.27	170MHz	27.1Mbps

In addition, it has been implemented on a cheap smart card based on the Motorola™ 6805 microprocessor for which one block encryption requires 9.80ms. All these experiments yield a speed rate greater than all commercial implementations of DES, and with a much higher security.

Provable security is an important added value for cryptographic algorithms and is currently a hot topic in international conferences. The decorrelation technique is a part of this program.

1 Notations

All objects are bit strings or integers. Bit strings are represented in hexadecimal notations. For instance, $\texttt{d43}_{\mathrm{x}}$ denotes the bit string `110101000011`. Integers are represented in standard decimal notations. The notations used to manipulate them are as follows.

$\bar{s}$	convert bit string into an integer.
$\lvert x\rvert_\ell$	convert integer x into a bit string of length ℓ.
$s\lvert s'$	concatenation of two strings.
$\mathrm{trunc}_n(s)$	truncate a bit string to its n leftmost bits.

$s \oplus s'$	bitwise XOR
$s \wedge s'$	bitwise *and*
$\neg s$	bitwise negation.
$+, \times, \bmod$	natural arithmetic operations over the integers.

For instance, $\overline{\mathtt{d43}_\mathrm{x}} = 3395$ and $|3395|_{12} = \mathtt{d43}_\mathrm{x}$.

2 High Level Overview

The encryption function DFC_K operates on 128-bit message blocks by means of a secret key K of arbitrary length, up to 256 bits. The corresponding decryption function is DFC_K^{-1} and operates on 128-bit message blocks. Encryption of arbitrary-length messages is performed through standard modes of operation.

The secret key K is first turned into a 1024-bit "Expanded Key" EK through an "Expanding Function" EF, *i.e.* $\mathrm{EK} = \mathrm{EF}(K)$. As explained in Section 5, the EF function performs a 4-round Feistel scheme (see Feistel [3]). The encryption itself performs a similar 8-round Feistel scheme. Each round uses the "Round Function" RF. This function maps a 64-bit string onto a 64-bit string by using one 128-bit string parameter. It is defined in Section 3.

Given a 128-bit plaintext block PT, we split it into two 64-bit halves R_0 and R_1 so that $\mathrm{PT} = R_0|R_1$. Given the 1024-bit expanded key EK, we split it into eight 128-bit strings

$$\mathrm{EK} = \mathrm{RK}_1|\mathrm{RK}_2|\ldots|\mathrm{RK}_8 \tag{1}$$

where RK_i is the ith "Round Key".

We build a sequence $R_0, \ldots, R_9$ by the Equation

$$R_{i+1} = \mathrm{RF}_{\mathrm{RK}_i}(R_i) \oplus R_{i-1}. \quad (i = 1, \ldots, 8) \tag{2}$$

We then set $\mathrm{CT} = \mathrm{DFC}_K(\mathrm{PT}) = R_9|R_8$ (see Fig. 1).

More generally, given a bitstring s of length multiple of 128, say $128r$, we can split it into r 128-bit strings

$$s = p_1|p_2|\ldots|p_r.$$

From s we define a permutation Enc_s on the set of 128-bit strings which comes from an r-round Feistel scheme. For any 128-bit string

m which is split into two 64-bit halves x_0 and x_1 so that $m = x_0|x_1$. We build a sequence $x_0, \ldots, x_{r+1}$ by the Equation

$$x_{i+1} = \mathrm{RF}_{p_i}(x_i) \oplus x_{i-1} \quad (i = 1, \ldots, r) \tag{3}$$

and we define $\mathrm{Enc}_s(m) = x_{r+1}|x_r$. The DFC_K encryption function is thus obtained as

$$\mathrm{DFC}_K = \mathrm{Enc}_{\mathrm{EF}(K)} \tag{4}$$

(hence an 8-round Feistel Cipher). The EF function uses a 4-round version defined with Enc.

Obviously, we have $\mathrm{DFC}_K^{-1} = \mathrm{Enc}_{\mathrm{revEK}}$ where

$$\mathrm{revEK} = \mathrm{RK}_8|\mathrm{RK}_7|\ldots|\mathrm{RK}_1. \tag{5}$$

3 The RF Function

The RF function (as for "Round Function") is fed with one 128-bit parameter, or equivalently two 64-bit parameters: an "a-parameter" and a "b-parameter". It processes a 64-bit input x and outputs a 64-bit string. We define

$$\mathrm{RF}_{a|b}(x) = \mathrm{CP}\left(\left|\left(\left(\bar{a} \times \bar{x} + \bar{b}\right) \bmod (2^{64} + 13)\right) \bmod 2^{64}\right|_{64}\right) \tag{6}$$

where CP is a permutation over the set of all 64-bit strings (which appears in Section 4). This construction is the "pairwise decorrelation module".

4 The CP Permutation

The CP permutation (as for "Confusion Permutation") uses a look-up table RT (as for "Round Table") which takes a 6-bit integer as input and provides a 32-bit string output.

Let $y = y_l|y_r$ be the input of CP where y_l and y_r are two 32-bit strings. We define

$$\mathrm{CP}(y) = \left|\overline{(y_r \oplus \mathrm{RT}(\overline{\mathrm{trunc}_6(y_l)}))|(y_l \oplus \mathrm{KC})} + \overline{\mathrm{KD}} \bmod 2^{64}\right|_{64} \tag{7}$$

where KC is a 32-bit constant string, and KD is a 64-bit constant string. Permutation CP is depicted on Fig. 2.

The constants $\mathrm{RT}(0), \ldots, \mathrm{RT}(63)$, KC and KD will be set in Section 6.

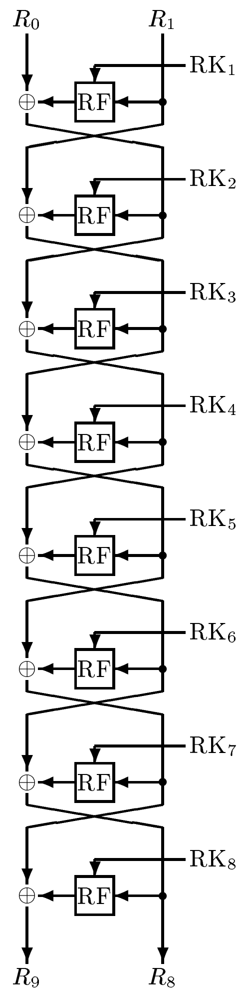

Fig. 1. An 8-Round Feistel Cipher.

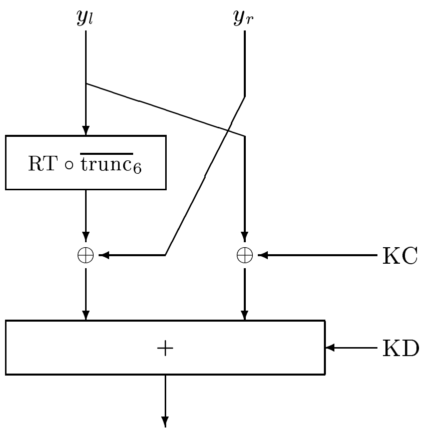

Fig. 2. The CP Permutation.

5 Key Scheduling Algorithm

In order to generate a sequence $\mathrm{RK}_1, \mathrm{RK}_2, \ldots, \mathrm{RK}_8$ from a given key K represented as a bit string of length at most 256, we use the following algorithm. We first pad K with a constant pattern KS in order to make a 256-bit "Padded Key" string by

$$\mathrm{PK} = \mathrm{trunc}_{256}(K|\mathrm{KS}). \tag{8}$$

If K is of length 128, we can observe that only the first 128 bits of KS are used. We define KS of length 256 in order to allow any key size from 0 to 256.

Then we cut PK into eight 32-bit strings $\mathrm{PK}_1, \ldots, \mathrm{PK}_8$ such that $\mathrm{PK} = \mathrm{PK}_1| \ldots |\mathrm{PK}_8$. We define

$$\mathrm{OAP}_1 = \mathrm{PK}_1|\mathrm{PK}_8 \tag{9}$$
$$\mathrm{OBP}_1 = \mathrm{PK}_5|\mathrm{PK}_4 \tag{10}$$
$$\mathrm{EAP}_1 = \mathrm{PK}_2|\mathrm{PK}_7 \tag{11}$$
$$\mathrm{EBP}_1 = \mathrm{PK}_6|\mathrm{PK}_3. \tag{12}$$

We also define

$$\mathrm{OAP}_i = \mathrm{OAP}_1 \oplus \mathrm{KA}_i \tag{13}$$
$$\mathrm{OBP}_i = \mathrm{OBP}_1 \oplus \mathrm{KB}_i \tag{14}$$

$$\mathrm{EAP}_i = \mathrm{EAP}_1 \oplus \mathrm{KA}_i \tag{15}$$

$$\mathrm{EBP}_i = \mathrm{EBP}_1 \oplus \mathrm{KB}_i \tag{16}$$

for $i = 2, 3, 4$ (where KA_i and KB_i are fixed constants defined in Section 6). The names of the variables come from "Odd a-Parameter", "Odd b-Parameter", "Even a-Parameter", and "Even b-Parameter" respectively, which will become clearer below.

We define

$$\mathrm{EF}_1(K) = \mathrm{OAP}_1|\mathrm{OBP}_1|\ldots|\mathrm{OAP}_4|\mathrm{OBP}_4. \tag{17}$$

It defines a four-round permutation which is $\mathrm{Enc}_{\mathrm{EF}_1(K)}$. Similarly,

$$\mathrm{EF}_2(K) = \mathrm{EAP}_1|\mathrm{EBP}_1|\ldots|\mathrm{EAP}_4|\mathrm{EBP}_4 \tag{18}$$

defines a four-round encryption function $\mathrm{Enc}_{\mathrm{EF}_2(K)}$.

The $\mathrm{Enc}_{\mathrm{EF}_1(K)}$ and $\mathrm{Enc}_{\mathrm{EF}_2(K)}$ enables to define the RK sequence. Namely, we let $\mathrm{RK}_0 = |0|_{128}$ and

$$\mathrm{RK}_i = \begin{cases} \mathrm{Enc}_{\mathrm{EF}_1(K)}(\mathrm{RK}_{i-1}) \text{ if } i \text{ is odd} \\ \mathrm{Enc}_{\mathrm{EF}_2(K)}(\mathrm{RK}_{i-1}) \text{ if } i \text{ is even.} \end{cases} \tag{19}$$

Finally we have

$$\mathrm{EF}(K) = \mathrm{RK}_1|\mathrm{RK}_2|\ldots|\mathrm{RK}_8. \tag{20}$$

6 On Defining the Constants

The previously defined algorithm depends on several constants:

- 64 constants $\mathrm{RT}(|0|_6), \ldots, \mathrm{RT}(|63|_6)$ of 32 bits,
- one 64-bit constant KD,
- one 32-bit constant KC,
- three 64-bit constants $\mathrm{KA}_2, \mathrm{KA}_3, \mathrm{KA}_4$,
- three 64-bit constants $\mathrm{KB}_2, \mathrm{KB}_3, \mathrm{KB}_4$,
- one 256-bit constant KS.

In order to convince that this design hides no trap-door, we choose the constants from the hexadecimal expansion of the mathematical e constant

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 2.\mathtt{b7e151628aed2a6abf7158}_{\mathrm{x}} \ldots \tag{21}$$

If EES is the "e Expansion String" of the first 2144 bits of this expansion (after the decimal point), we define

$$\mathrm{EES} = \mathrm{RT}(0)|\mathrm{RT}(1)|\ldots|\mathrm{RT}(63)|\mathrm{KC}|\mathrm{KD}. \tag{22}$$

In addition we define

$$\mathrm{trunc}_{640}(\mathrm{EES}) = \mathrm{KA}_2|\mathrm{KA}_3|\mathrm{KA}_4|\mathrm{KB}_2|\mathrm{KB}_3|\mathrm{KB}_4|\mathrm{KS}. \tag{23}$$

Here is the EES string.

$\texttt{b7e15162 8aed2a6a bf715880 9cf4f3c7 62e7160f 38b4da56}_\mathrm{x}$
$\texttt{a784d904 5190cfef 324e7738 926cfbe5 f4bf8d8d 8c31d763}_\mathrm{x}$
$\texttt{da06c80a bb1185eb 4f7c7b57 57f59584 90cfd47d 7c19bb42}_\mathrm{x}$
$\texttt{158d9554 f7b46bce d55c4d79 fd5f24d6 613c31c3 839a2ddf}_\mathrm{x}$
$\texttt{8a9a276b cfbfa1c8 77c56284 dab79cd4 c2b3293d 20e9e5ea}_\mathrm{x}$
$\texttt{f02ac60a cc93ed87 4422a52e cb238fee e5ab6add 835fd1a0}_\mathrm{x}$
$\texttt{753d0a8f 78e537d2 b95bb79d 8dcaec64 2c1e9f23 b829b5c2}_\mathrm{x}$
$\texttt{780bf387 37df8bb3 00d01334 a0d0bd86 45cbfa73 a6160ffe}_\mathrm{x}$
$\texttt{393c48cb bbca060f 0ff8ec6d 31beb5cc eed7f2f0 bb088017}_\mathrm{x}$
$\texttt{163bc60d f45a0ecb 1bcd289b 06cbbfea 21ad08e1 847f3f73}_\mathrm{x}$
$\texttt{78d56ced 94640d6e f0d3d37b e67008e1 eb64749a 86d1bf27}_\mathrm{x}$
$\texttt{5b9b241d}_\mathrm{x}$

7 Security Results

The design construction is based on decorrelation techniques (see [5–7]). From the decorrelation theory we know that a six-round Feistel cipher which uses RF with independent subkeys has a pairwise decorrelation distance less than $0{,}821.2^{-113}$ to the Perfect Cipher. We can thus give lower bounds on the complexity of differential cryptanalysis, linear cryptanalysis and general iterated attacks of order 1 which achieve an advantage at least 10%.

attack	differential	linear	iterated
complexity lower bound	2^{110}	2^{92}	2^{48}

These are attacks against a six-round encryption function when assuming that EK has a uniform distribution. It is applicable to DFC with the following assumption.

> “We cannot distinguish $\mathrm{Enc}_{\mathrm{EF}_1(K)}$ from the a truly random permutation within an advantage greater than 1%, with only 4 chosen plaintexts and a limited budget of US$1,000,000,000.”

(Limiting the budget gives an upper bound on the computation cost.)

These results suggest that the key should not be used more than 2^{48} times *i.e.* that we should not encrypt 4096TB with the same key. We believe that this restricts no practical application.

We also (pessimisticly) investigated the complexity of exhaustive search by extrapolating the technology improvements. We obtained the following rationales.

key length	80	128	192	256
computation lower bound (in years)	21.7	93.7	126.4	190.4

In our full report we also outlined that the DFC algorithm is weak when reduced to four rounds. We believe the decorrelation technique makes enough avalanche effect so that eight rounds provide a sufficient security.

8 Conclusion

We have proposed a dedicated block cipher algorithm which is faster than DES and hopefully more secure than triple-DES. In addition we provided proofs of security against some classes of general simple attacks which includes differential and linear cryptanalysis. This result is based on the decorrelation theory. We believe that this cipher is also “naturally” secure against more complicated attacks since our design introduced no special algebraic property. We believe that the best attack is still exhaustive search which is limited by the implementation speed (decreased by a factor of 5 due to the key scheduling algorithm). We (very pessimisticly) forecast that one need at least several decades to search a 80-bit key, which makes it safe until the Advanced Encryption Standard expires.

Our algorithm accepts 128-bit message blocks and any key size from 0 to 256. It can be adapted into a 64-bit variant (with a key size up to 128) as shown in the full report. We believe that it can be adapted to any other cryptographic primitive such as stream cipher, hash function, MAC algorithm.

Our algorithm can be implemented on traditional personal computers, as well as on cheap smart cards. We believe that it can be implemented in any other digital environment.

In conclusion we recommend this encryption algorithm as a candidate to the Advanced Encryption Standard process.

References

1. Data Encryption Standard. *Federal Information Processing Standard Publication 46*, U. S. National Bureau of Standards, 1977.
2. E. Biham, A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
3. H. Feistel. Cryptography and Computer Privacy. *Scientific American*, vol. 228, pp. 15–23, 1973.
4. M. Matsui. The First Experimental Cryptanalysis of the Data Encryption Standard. In *Advances in Cryptology CRYPTO'94*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 839, pp. 1–11, Springer-Verlag, 1994.
5. S. Vaudenay. Provable Security for Block Ciphers by Decorrelation. In *STACS 98*, Paris, France, Lectures Notes in Computer Science 1373, pp. 249–275, Springer-Verlag, 1998.
6. S. Vaudenay. Provable Security for Block Ciphers by Decorrelation. (Journal Version.) Submitted.
7. S. Vaudenay. The Decorrelation Technique Home-Page. URL:`http://www.dmi.ens.fr/~vaudenay/decorrelation.html`