On the Need for Multipermutations: Cryptanalysis of MD4 and SAFER

Serge VAUDENAY

Laboratoire d'Informatique, URA 1327 du CNRS
Département de Mathématiques et d'Informatique
Ecole Normale Supérieure

LIENS - 94 - 23

November 1994

On the need for multipermutations: Cryptanalysis of MD4 and SAFER

Serge Vaudenay
LIENS*
45, rue d'Ulm
75230 Paris Cedex 05
FRANCE

November 16, 1994

Abstract

Cryptographic primitives are usually based on a network with some gates. In [SV94], it is claimed that all gates should be multipermutations. In this paper, we investigate a few combinatorial properties of multipermutations. We argue that gates which fail to be multipermutations can open the way to unsuspected attacks. We illustrate this statement with two examples.

Firstly, we show how to construct collisions to MD4 restricted to its first two rounds. This allows to forge digests close to each other using the full compression function of MD4. Secondly, we show that some generalizations of SAFER are subject to attack faster than exhaustive search in 6.1% cases. This attack can be implemented if we decrease the number of rounds from 6 to 4.

In [SV94], *multipermutations* are introduced as formalization of perfect diffusion. The aim of this paper is to show that the concept of multipermutation is a basic tool in the design of dedicated cryptographic functions, as functions that do not realize perfect diffusion may be subject to some clever cryptanalysis in which the flow of information is controlled throughout the computation network. We give two cases of such an analysis.

Firstly, we show how to build collisions for MD4 restricted to its first two rounds[1]. MD4 is a three rounds hash function proposed in [Riv90]. An attack on

*The *Laboratoire d'Informatique de l'Ecole Normale Supérieure* is a research group affiliated with the CNRS

[1]This part of research has been supported by the CELAR.

MD4 restricted to its last two rounds is detailed in [BB91]. An other unpublished attack on the first two rounds has been found by Merkle (see the introduction of [BB91]). This attack does not produce a real collision, but different digests very close to each other (according to the Hamming distance). Here, we present a new attack which is based on the fact that an inert function is not a multipermutation. This attack requires less than one tenth of a second on a SUN workstation. Moreover, the same attack applied to the full MD4 compression function produces two different digests close to each other.

Secondly, we show how to develop a known plaintext attack to a variant of SAFER K-64, in which we replace the permutation $\exp_{45}$ by a (weaker) one. SAFER is a six rounds encryption function introduced in [Mas93]. It uses a byte-permutation (namely, $\exp_{45}$ in the group of nonzero integers modulo 257) for confusion. If we replace $\exp_{45}$ by a random permutation P (and $\log_{45}$ by P^{-1}), we show that in 6.1% of the cases, there exists a known plaintext attack faster than exhaustive search. Furthermore, this attack can be implemented for the function restricted to 4 rounds. This attack is based on the linear cryptanalysis introduced in [Mat93] and recently gave way to the first experimental attack of the full DES function in [Mat94].

1 Multipermutations

In [SV94], multipermutations with 2 inputs and 2 outputs are introduced. Here, we propose to generalize to any number of inputs and outputs.

Definition *A (r,n)-multipermutation over an alphabet Z is a function f from Z^r to Z^n such that two different $(r+n)$-tuples of the form $(x, f(x))$ cannot collide in any r positions.*

Thus, a $(1,n)$-multipermutation is nothing but a vector of n permutations over Z. A $(2,1)$-multipermutation is equivalent to a *latin square*[2]. A $(2,n)$-multipermutation is equivalent to a set of n two-wise *orthogonal* latin squares[3]. Latin squares are widely studied in [DK74].

An equivalent definition says that the set of all $(r+n)$-tuples of the form $(x, f(x))$ is an error correcting code with minimal distance $n+1$, which is the maximal possible. In the case of a linear function f, this is the definition of MDS codes: codes which reach Singleton's bound. For more details about MDS codes, see [MS77]. More generally, a (r,n)-multipermutation is equivalent to a $((\#Z)^r, r+n, \#Z, r)$-orthogonal array[4].

[2] a latin square over a finite set of k elements is a $k \times k$ matrix with entries from this set such that all elements are represented in each column and each row.

[3] two latin squares A and B are orthogonal if the mapping $(i,j) \mapsto (A_{i,j}, B_{i,j})$ gets all possible couples.

[4] a $(M, r+n, q, r)$-orthogonal array is a $M \times (r+n)$ matrix with entries from a set of q elements such that any set of r columns contains all q^r possible rows exactly $\frac{M}{q^r}$ times.

A multipermutation performs a *perfect diffusion* in the sense that changing t of the inputs changes at least $n-t+1$ of the outputs. If a function is not a multipermutation, one can find several values such that both few inputs and few outputs are changed. Those values can be used in cryptanalysis as is shown in two examples below. This motivates the use of multipermutations in cryptographic functions.

The design of multipermutations over a large alphabet is a very difficult problem, as the design of two-wise orthogonal latin squares in a well-known difficult one. The only powerful method seems to use an MDS code combined with several permutations at each coordinate.

In the particular case of 2 inputs, it is attractive to choose latin squares based on a group law: if we have a group structure over Z, we can seek permutations α, β, γ, δ, ϵ and ζ such that

$$(x, y) \mapsto (\alpha[\beta(x).\gamma(y)], \delta[\epsilon(x).\zeta(y)])$$

is a permutation, as it will be sufficient to get a multipermutation. Unfortunately, it is possible to prove that such permutations exist only when the 2-Sylow subgroup of Z is not cyclic[5], using a theorem from [HP55]. More precisely, they do not exist when the 2-Sylow subgroup is cyclic. They are known to exist in all solvable groups in which the 2-Sylow subgroup is not cyclic, but the existence in the general case is still a conjecture. Hence, Z should not have a cyclic group structure. For instance, we can use the $GF(2)^n$ group structure for $n > 1$. Such multipermutations are proposed in [SV94].

In MD4, the group structure of $GF(2)^{32}$ is used, but some functions are not multipermutations. On the other hand, in SAFER, the group structure of $\mathbb{Z}_{256}$, which is cyclic, is used, so without multipermutations.

2 Cryptanalysis of MD4

2.1 Description of MD4

MD4 is a hash function dedicated to 32 bits microprocessors. It hashes any bit string into a 128 bits digest. The input is padded following the Merkle-Damgård scheme [Dam89, Mer89] and cut into 512 bits long blocks. Then, each block is processed iteratively using the Davies-Meyer scheme [DP85, MMO85] and an encryption function C: if $B_1, \ldots, B_n$ is the sequence of blocks (the padded message), the hash value is

$$h_{B_n}(\ldots h_{B_1}(v_i) \ldots)$$

[5] we agree the trivial group is not cyclic. Actually, $x \mapsto x^2$ is an orthomorphism in all groups with odd order, in which the 2-Sylow subgroup is trivial.

where v_i is an Initial Value, and $h_x(v)$ is $C_x(v) + v$ (x is the key and v is the message to encrypt).

Here we intend to build a single block collision to $h(v_i)$, that is to say two blocks x and x' such that $C_x(v_i) = C_{x'}(v_i)$. It is obvious that this can be used to build collisions to the hash function. So, we only have to recall the definition of the function $C_x(v)$.

The value v is represented as 4 integers a, b, c and d (coded with 32 bits), and the key x is represented as 16 integers $x_1, \ldots, x_{16}$. The initial definition of C uses three rounds $i = 1, 2, 3$. The figure 1 shows the computational graph of a single round i. It uses a permutation σ_i and some boxes B_i^j. B_i^j is fed with a main input, a block integer $x_{\sigma_i(j)}$ and three side inputs. If p is the main input and q, r and s are the side inputs (from top to bottom), the output is

$$R^{\alpha_{i,j}}(p + f_i(q, r, s) + x_{\sigma_i(j)} + k_i)$$

where R is the right circular rotation, $\alpha_{i,j}$ and k_i are constants and f_i is a particular function. In the following, we just have to know that f_2 is the bit-wise majority function, σ_1 is the identical permutation, and

$$\sigma_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 1 & 5 & 9 & 13 & 2 & 6 & 10 & 14 & 3 & 7 & 11 & 15 & 4 & 8 & 12 & 16 \end{pmatrix}$$

2.2 Attack on the first two rounds

If we ignore the third round of C, it is very easy to build collisions. We notice that no B_2^j are multipermutations: if $p = 0$, $x = -k_2$ and two of the three integers q, r and s are set to zero, then $B_2^j(x, p, q, r, s)$ remains zero (the same remark holds with -1 instead of 0). So, we can imagine an attack where two blocks differ only in x_{16}, the other integers are almost all set to $-k_2$ and such that almost all the outputs of the first round are zero. This performs a kind of corridor where the modified values are controlled until the final collision.

More precisely, let $x_1, \ldots, x_{11}$ equal $-k_2$, x_{12} be an arbitrary integer (your phone number for instance) and x_{13}, x_{14} and x_{15} be such that the outputs a, c and d of the first round are zero. The computation of x_{13}, x_{14} and x_{15} is very easy from the computational graph. Thanks to the previous remark, we can show that the outputs a, c and d of the second round do not depend on x_{16} as the modified information in x_{16} is constrained in the register b. Thus, modifying x_{16} does not modify a, c and d.

Letting the b output be a function of x_{16}, we just have to find a collision to a 32 bits to 32 bits function. This can be done very efficiently using the birthday paradox or the ρ method. An implementation on a Sparc Station uses one tenth of second.

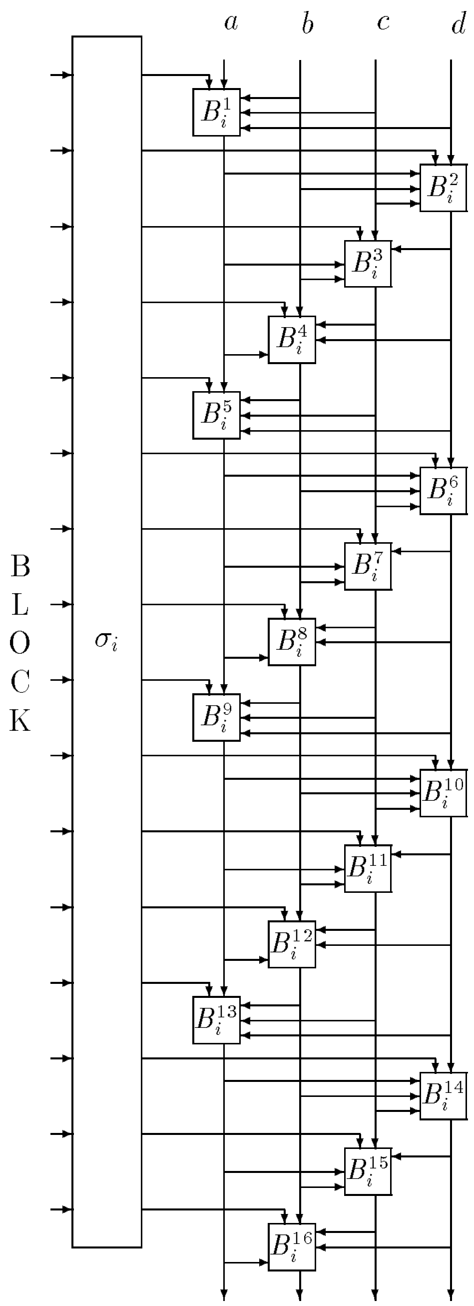

Figure 1: One round of C

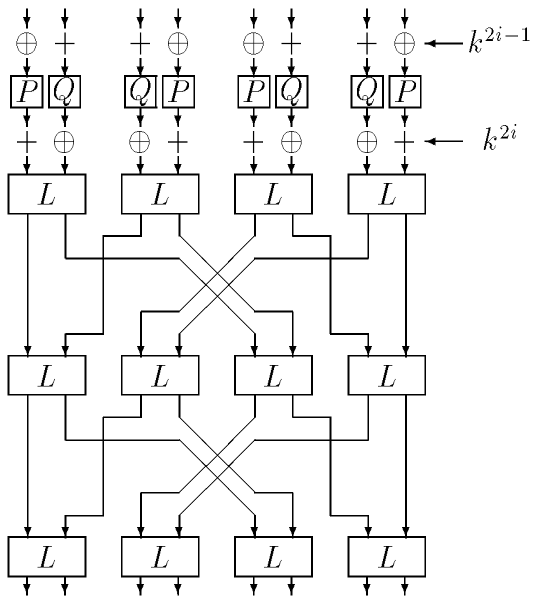

Figure 2: The ith round of SAFER

If we use the same attack on the full-MD4 function, since $\sigma_3^{-1}(16) = 16$, the only modified x occurs in the very last computation in the third round. So, if this round is fed with a collision, it produces a collision on the a, c and d output. The digests differ only in the second integer b. Hence, the average Hamming distance between both digests is 16.

3 Cryptanalysis of SAFER

3.1 Description of SAFER

SAFER is an encryption function dedicated to 8 bits microprocessors. It encrypts a 64 bits message using a 64 bits key. The key is represented as 8 integers $k_1, \ldots, k_8$. A key scheduling algorithm produces several subkeys $k_1^i, \ldots, k_8^i$. In the following, we just have to know that k_j^i is a simple function of k_j (and $k_j^1 = k_j$).

The encryption algorithm takes 6 rounds and a half. The ith round is summarized in figure 2. It uses the subkeys k_j^{2i-1} and k_j^{2i}. After the 6th round, the half round simply consists in xoring/adding the subkeys k_j^{13} as we would do in a 7th round.

$\oplus$ represents the xor operation on 8 bits integers. $+$ is the addition modulo 256. P is a permutation over the set of all integers defined in the SAFER design. Q is the inverse permutation of P. L is a linear permutation over the algebraic structure of the ring $\mathbb{Z}_{256}$, as

$$L(x,y) = (2.x + y, x + y) \pmod{256}.$$

In the original design, P is the exponentiation in base 45 modulo 257: all integers from 1 to 256 can be coded with 8 bits (256 is coded as zero) and represent the group of all invertible integers modulo 257. 45 is a generator of this group.

In practical implementations, we have to store the table of the permutation P. So, there is no reason to study SAFER with this particular permutation. Here, we will show that this choice is a very good one, as for 6.1% of all possible permutations, there exists a known plaintext attack faster than exhaustive search.

3.2 Linear cryptanalysis of SAFER

The permutation L is not a multipermutation, as we have

$$L_1(x+128, y) = L_1(x,y)$$

for all x and y (where L_1 denotes the first output of L). So, we have pairs of 4-tuples $(x, y, L(x,y))$ at Hamming distance 2. Actually, there are no $(2,2)$-multipermutations which are linear over $\mathbb{Z}_{256}$ as its 2-Sylow subgroup is cyclic (it is itself here). We can use this property of L_1 by a dual point of view noticing that some information about $L_1(x,y)$ only depends on y. Namely, we have

$$L_1(x,y) \cdot 1 = y \cdot 1$$

where $\cdot$ is the inner product over $(\mathbb{Z}_2)^8$, so, $y \cdot 1$ is the least significant bit of y. Similarly, we have

$$(L_1(x,y)\cdot 1) \oplus (L_2(x,y)\cdot 1) = x \cdot 1$$

Let us denote F the function defined by the three bottom layers on figure 2 (layers which uses L in a round). If $x_1, \ldots, x_8$ are the inputs of a round, the outputs are $F(y_1, \ldots, y_8)$ where $y_1 = P(x_1 \oplus k_1^1) + k_1^2$, ... We notice that if $F(y_1, \ldots, y_8) = (z_1, \ldots, z_8)$, we have a 2-2 *linear characteristic*

$$(z_3 \cdot 1) \oplus (z_4 \cdot 1) = (y_3 \cdot 1) \oplus (y_4 \cdot 1)$$

(this means there is a linear dependance using 2 inputs and 2 outputs of F). There are 5 other 2-2 linear characteristics:

$$\begin{aligned}
(z_2 \cdot 1) \oplus (z_6 \cdot 1) &= (y_2 \cdot 1) \oplus (y_6 \cdot 1)\\
(z_5 \cdot 1) \oplus (z_7 \cdot 1) &= (y_5 \cdot 1) \oplus (y_7 \cdot 1)\\
(z_3 \cdot 1) \oplus (z_7 \cdot 1) &= (y_5 \cdot 1) \oplus (y_6 \cdot 1)\\
(z_5 \cdot 1) \oplus (z_6 \cdot 1) &= (y_2 \cdot 1) \oplus (y_4 \cdot 1)\\
(z_2 \cdot 1) \oplus (z_4 \cdot 1) &= (y_3 \cdot 1) \oplus (y_7 \cdot 1).
\end{aligned}$$

If L were a multipermutation, the smallest characteristics would be a-b ones such that $a+b=6$. This means more information would be required in a cryptanalysis.

Let q denote $\text{Prob}_x[x \cdot 1 = P(x) \cdot 1] - \frac{1}{2}$, the bias which measures the dependence between the least lignificant bits of $P(x)$ and x. We get the same bias with Q in place of P. If $(x_1, \ldots, x_8)$ is a plaintext, if $y_1 = P(x_1 \oplus k_1)$, $y_2 = Q(x_2 + k_2)$, ..., $y_8 = P(x_8 \oplus k_8)$, and if $(z_1, \ldots, z_8)$ is the ciphertext, let us write

$$b(x,z) = (y_3 \cdot 1) \oplus (y_4 \cdot 1) \oplus (z_3 \cdot 1) \oplus (z_4 \cdot 1).$$

Lemma 1 in appendix A states that $b(x,z) = \phi(k)$ with probability $\frac{1}{2}(1+(2q)^{10})$ where $\phi(k)$ denotes the exclusive or of all k_3^i and k_4^i for $i = 2, \ldots, 13$. For a given (x,z), to compute $b(x,z)$, we only have to know k_3 and k_4. Lemma 1 states that it occurs with probability roughly equal to $\frac{1}{2}$ (the difference with $\frac{1}{2}$ is negligible against $(2q)^{10}$) when wrong k_3 and k_4 are used in the computation of $b(x,z)$. Thus, trying all the possible (k_3, k_4), it is possible to distinguish the good one from the other candidates by statistical measure.

Let us recall the following theorem (see [Fel57] for instance):

Theorem (Central limit theorem) *If B is the statistical average of N independent random variables with the same probability distribution of average μ and standard deviation σ, we have*

$$Prob\left[(B-\mu)\frac{\sqrt{N}}{\sigma} \in [a,b]\right] \to \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{t^2}{2}} dt.$$

Let $B(k_3, k_4)$ be the average of $b(x,z)$ over all the N available couples (x,z). Lemma 1 proves that the standard deviation of $b(x,z)$ is close to $\frac{1}{2}$. Let

$$\lambda = \frac{\sqrt{N}(2q)^{10}}{2}.$$

The central limit theorem states that if (k_3, k_4) is wrong,

$$\text{Prob}\left[\left|B(k_3,k_4) - \frac{1}{2}\right| < \frac{\lambda}{2\sqrt{N}}\right] \to \frac{1}{\sqrt{2\pi}} \int_{-\lambda}^{\lambda} e^{-\frac{t^2}{2}} dt;$$

and if (k_3, k_4) is good,

$$\text{Prob}\left[\left|B(k_3,k_4) - \frac{1}{2}\right| < \frac{\lambda}{2\sqrt{N}}\right] \to \frac{1}{\sqrt{2\pi}} \int_{\lambda}^{3\lambda} e^{-\frac{t^2}{2}} dt.$$

To get a probability greater than 95%, we have to reach $\lambda = 2:$ the good (k_3, k_4) is accepted with probability 95% and the bad ones are rejected with probability

98%. So, the number of plaintexts/ciphertexts required to distinguish the good (k_3, k_4) is

$$N \sim \frac{16}{(2q)^{20}}.$$

If $|q|$ is greater than 2^{-4}, this is faster than exhaustive search.

For only 4 rounds in SAFER, we have $N \sim \frac{16}{(2q)^{12}}$. So, for all permutations P which are biased ($q \neq 0$), this attack is faster than exhaustive search. For $|q| \geq 2^{-4}$, the attack can be implemented.

The analysis of the distribution of q shows that we have $|q| \geq 2^{-4}$ for 6.1% of the possible permutations P (see appendix B). We have $q = 0$ for only 9.9% of the permutations. Unfortunately (or fortunately), for the P chosen by Massey, we have $q = 0$, so, the weakness of the diffusion phase is balanced by the strength of the confusion phase. Actually, $q = 0$ is a property of all exponentiations which are permutations (see appendix B).

Further analysis can improve this attack. It is possible to use tighter computations. We can look for a better tradeoff between the workload and the probability of success. It is also possible to use several characteristics to decrease N (for more details, see [KR94]). Actually, it is possible to decrease N by a factor of 64.

Conclusion

In MD4, we have shown that the fact that f_2 is not a multipermutation allows to mount an attack. Similarly, in SAFER, the diffusion function is not a multipermutation. This allows to imagine another attack. This shows that we do need multipermutations in the design of cryptographic primitives. Research in this area should be motivated by this general statement.

Acknowledgments

I would like to thank Antoon Bosselaers for helpful informations. I thank the CELAR for having motivated the research on MD4. I also thank Jacques Stern and Hervé Brönnimann for their help.

References

[BB91] B. den Boer, A. Bosselaers. An attack on the last two rounds of MD4. In *Advances in Cryptology CRYPTO'91*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 576, pp. 194–203, Springer-Verlag, 1992.

[DK74] J. Dénes, A. D. Keedwell. *Latin squares and their applications*, Akadémiai Kiadó, Budapest, 1974.

[Dam89] I. B. Damgård. A design principle for hash functions. In *Advances in Cryptology CRYPTO'89*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 435, pp. 416–427, Springer-Verlag, 1990.

[DP85] R. W. Davies, W. L. Price. Digital signature – an update. In *Proceedings of the International Conference on Computer Communications*, Sydney, pp. 843-847, North-Holland, 1985.

[Fel57] W. Feller. *An Introduction to Probability Theory and its Applications*, vol. 1, Wiley, 1957.

[HP55] M. Hall, L. J. Paige. Complete mappings of finite groups. In *Pacific Journal of Mathematics*, vol. 5, pp. 541–549, 1955.

[KR94] B. R. Kaliski Jr., M. J. B. Robshaw. Linear cryptanalysis using multiple approximations. In *Advances in Cryptology CRYPTO'94*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 839, pp. 26–39, Springer-Verlag, 1994.

[Mas93] J. Massey. SAFER K-64: a byte-oriented block-ciphering algorithm. In *Fast Software Encryption – Proceedings of the Cambridge Security Workshop*, Cambridge, U. K., Lectures Notes in Computer Science 809, pp. 1–17, Springer-Verlag, 1994.

[Mat93] M. Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology EUROCRYPT'93*, Lofthus, Norway, Lectures Notes in Computer Science 765, pp. 386–397, Springer-Verlag, 1994.

[Mat94] 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.

[Mer89] R. C. Merkle. One way hash functions and DES. In *Advances in Cryptology CRYPTO'89*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 435, pp. 416–427, Springer-Verlag, 1990.

[MMO85] S. M. Matyas, C. H. Meyer, J. Oseas. Generating strong one-way functions with cryptographic algorithm. *IBM Technical Disclosure Bulletin*, vol. 27, pp. 5658–5659, 1985.

[MS77] F. J. McWilliams, N. J. A. Sloane. *The theory of error-correcting codes*, North-Holland, 1977.

[Riv90] R. Rivest. The MD4 Message Digest algorithm. In *Advances in Cryptology CRYPTO'90*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 537, pp. 303–311, Springer-Verlag, 1991.

[SV94] C.-P. Schnorr, S. Vaudenay. Black box cryptanalysis of hash networks based on multipermutations. To appear in *Advances in Cryptology EUROCRYPT'94*.

Appendix A

Lemma 1 *If $\phi(k)$ denotes the least lignificant bit of the sum of all k_3^i and k_4^i for $i = 2, \ldots, 13$, let us denote $y_3 = Q(x_3 + k_3)$, $y_4 = P(x_4 \oplus k_4)$ and*

$$b(x,z) = (y_3 \cdot 1) \oplus (y_4 \cdot 1) \oplus (z_3 \cdot 1) \oplus (z_4 \cdot 1)$$

where z is the encrypted message of x using an unknown key. $b(x,z) = \phi(k)$ holds with probability

$$\frac{1}{2}(1 \pm (2q)^{10+e})$$

where e is the number of wrong integers in (k_3, k_4) ($e = 0$ if both are good and $e = 2$ in most of cases). The standard deviation of $b(x,z)$ is

$$\frac{1}{2}\sqrt{1-(2q)^{20+2e}}.$$

Proof: Thanks to the property of the linear characteristic, if we denote by t_j^i the xor of the least significant bit of the input and the output of the P/Q box in position j in round $\#i$, it is easy to se that

$$b(x,z) = (y_3 \cdot 1) + (y_4 \cdot 1) + (y_3' \cdot 1) + (y_4' \cdot 1) + \sum_{i=2}^{6} \sum_{i=3}^{4} t_j^i + \phi(k') \pmod 2$$

where $\phi(k')$ denotes the real $\phi(k)$ and y_3' (resp. y_4') denotes the real y_3 (resp. y_4). Under the assumption that all inputs to P/Q boxes are uniformly distributed and independent, it is easy to prove by induction that

$$\text{Prob}\left[\sum_{i=2}^{6} \sum_{i=3}^{4} t_j^i = 0\right] = \frac{1}{2}(1 + (2q)^{10})$$

This finishes the case where k_3 and k_4 are good.

If k_3 or k_4 are wrong, let us denote $e = 2$ if both are bad, and $e = 1$ if only one is bad. Assume k_3 is bad without loss of generality. We have

$$\text{Prob}\left[(y_3 \cdot 1) \oplus (y_3' \cdot 1) = 0\right] = \frac{1}{2}(1 + 2q).$$

The $\pm$ comes from whether $\phi(k) = \phi(k')$ or not. This finishes the computation of the probability.

The standard deviation comes from the following formula which holds for all 0/1 random variables :

$$\sigma(b) = \sqrt{E(b)(1 - E(b))}.$$

□

Appendix B

Lemma 2 *If $q = Prob[x \cdot 1 = P(x) \cdot 1] - \frac{1}{2}$ where P is a permutation over $\{0, \dots, n-1\}$ (we assume that n is a multiple of 4), nq is always an even integer and for all integer k*

$$Prob\left[q = \frac{2k}{n}\right] = \frac{\left((\frac{n}{2})!\right)^4}{n!\left(\left(\frac{n}{4} - k\right)!\right)^2\left(\left(\frac{n}{4} + k\right)!\right)^2}$$

for a permutation P uniformly distributed.

Proof: If $k + \frac{n}{4}$ denotes the number of even integers x such that $P(x)$ is even, we have $q = \frac{2k}{n}$. So, we just have to enumerate the number of permutations for a given $k + \frac{n}{4}$.

We have to choose 4 sets with $k + \frac{n}{4}$ elements in sets with $\frac{n}{2}$ elements: the set of even integers which are mapped on even integers, the set of their images, the set of odd integers which are mapped on odd integers and the set of their images. We also have to choose 2 permutations over a set of $k + \frac{n}{4}$ integers (how to connect even to even integers and odd to odd integers) and 2 permutations over a set of $-k + \frac{n}{4}$ integers (how to connect even to odd integers and odd to even integers). So, the number of permutations is

$$\binom{\frac{n}{2}}{\frac{n}{4} + k}^4 \times \left(\left(\frac{n}{4} + k\right)!\right)^2 \times \left(\left(\frac{n}{4} - k\right)!\right)^2.$$

□

This allows to compute

$$\text{Prob}\left[|q| \geq 2^{-4}\right] \simeq 6.1\%$$

for $n = 256$ and

$$\text{Prob}\left[q = 0\right] \simeq 9.9\%.$$

Lemma 3 *For any generator g of $\mathbb{Z}_{257}^*$, the permutation $x \mapsto g^x$ is unbiased (i.e. $q = 0$).*

Proof: We have $(g^{128})^2 \equiv g^{256} \equiv 1 \pmod{257}$ so g^{128} is 1 or -1. As the exponentiation in base g is a permutation and $g^0 = 1$, we have $g^{128} \equiv -1 \pmod{257}$.

We have $g^{x+128} \equiv -g^x \equiv 257 - g^x \pmod{257}$, so, we can partition all the integers into pairs $\{x, x + 128\}$ of integers with the same least significant bit. The image of this pair by the exponentiation has two different least significant bits, so the bias q is 0. □