Internal collision attack on Maraca

Anne Canteaut[1], Maria Naya-Plasencia[2]

INRIA project-team SECRET
B.P. 105
78153 Le Chesnay Cedex, France
Anne.Canteaut@inria.fr, Maria.Naya_Plasencia@inria.fr

Abstract. We present an internal collision attack against the new hash function Maraca which has been submitted to the SHA-3 competition. This attack requires 2^{237} calls to the round function and its complexity is lower than the complexity of the generic collision attack when the length of the message digest is greater than or equal to 512. It is shown that this cryptanalysis mainly exploits some particular differential properties of the inner permutation, which are in some sense in contradiction with the usual security criterion which guarantees the resistance to differential attacks.

Keywords. hash function, collision attack, differential cryptanalysis, Boolean function.

1 Introduction

Maraca is a new keyed hash function which has been submitted to the SHA-3 competition [1]. It consists in applying a round permutation to the 1024-bit internal state, but one of its main features is that each message block is inserted four times, separated by 46 rounds. Then, a usual differential attack requires the study of the difference propagation on at least 46 rounds of the function. Here, we present a new type of collision attack, which leads to colliding internal states for Maraca. Our attack requires 2^{237} calls to the round function, *i.e.* at least 2^{24} times less than the generic collision attack for 512-bit message digest. The time complexity is also lower than the complexity of the generic collision attack. Breaking Maraca-512 does not have an important impact on the SHA-3 competition since Maraca has not advanced to the first round on the competition. However, our attack exhibits a new differential property of the inner permutation which may introduce some unexpected weaknesses. Most notably, we here point out that the resistance to our attack is in contradiction with the resistance to the usual differential attacks, and that finding a good inner permutation for Maraca raises some interesting open issues related to the construction of vectorial Boolean functions with good cryptographic properties.

After a brief description of Maraca, our attack is presented in Section 3 in a general setting, *i.e.* independently from the choice of the inner permutation. The attack has several variants: the basic general attack and a refinement based a a

Dagstuhl Seminar Proceedings 09031
Symmetric Cryptography
http://drops.dagstuhl.de/opus/volltexte/2009/1953

sieving procedure, which has a lower time complexity but only applies when the inner permutation has a particular algebraic structure. Section 4 then shows that this variant with sieving applies in the case of the inner permutation Perm used in Maraca. Finally, Section 5 investigates the properties of the inner permutation which guarantee that the hash function resists our attack. The interesting point is that this new security criterion is related to the differential properties of the permutation, and that there is a trade-off between this new criterion and the security criterion for classical differential attacks. For instance, we point out that some natural choices for the inner permutation, like a function based on the AES Sbox, increase the vulnerability to our attack.

2 Brief description of Maraca

As a keyed hash algorithm, Maraca takes as inputs a message and a key, and it produces a hash value of size h. The original message is padded as follows: the 1024-bit key is first appended to the message as a prefix, and the resulting message is then padded with a value depending on the key and on the message length, in order to get a padded message whose length is a multiple of 1024 bits. Note that our collision attack is considering messages of the same length and with the same key.

The internal state in Maraca has 1024 bits and the message blocks which are inserted at each round are of the same size as the internal state. Each message block M_i is inserted four times, at rounds i, $(i+21-6(i \bmod 4))$, $(i+41-6((i+2) \bmod 4))$ and $(i+46)$. More precisely, the original value of M_i is inserted at Round i, while rotated versions of M_i are inserted at the other three rounds, with rotations of 128 bits, 3×128 bits and 6×128 bits respectively. From now on, these rotated versions of M_i are denoted by M_i', M_i'' and M_i'''. It is worth noticing that the last round which uses the message block M_i is Round $i+46$.

The round function at Round i can be decomposed as follows:

- the new message block M_i is inserted for the first time by xoring it with the current internal state;
- a 1024-bit inner permutation Perm is applied to the internal state;
- $(M'_{i-3-6((i+2) \bmod 4)} \oplus M''_{i-23-6(i \bmod 4)} \oplus M'''_{i-46})$ is xored to the internal state;
- two iterations of Perm are applied to the internal state.

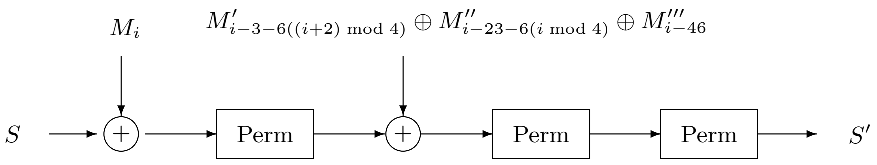
M_i
$M'_{i-3-6((i+2) \bmod 4)} \oplus M''_{i-23-6(i \bmod 4)} \oplus M'''_{i-46}$
S
Perm
Perm
Perm
S'

Fig. 1. Round i in Maraca

Then, we are ready to start the next round and to introduce the message block M_{i+1}, if any. If no message block has to be inserted anymore, the all-zero block is used. The message insertion phase ends up when all message blocks have been used four times, implying that, for an ℓ-block message, the message insertion phase consists of $(\ell + 46)$ rounds. The h-bit hash value is finally extracted from the internal state after applying 30 more iterations of Perm.

Since the internal state in Maraca has $n = 1024$ bits, the generic attack for finding an internal collision requires to hash around $2^{\frac{n}{2}}$ messages, *i.e.* at least 46×2^{512} calls to the round permutation. Actually, because of the padding and of the fact that each message block is inserted at four different rounds, we cannot search for colliding internal states which correspond to different rounds.

The generic collision attack for h-bit message digests requires to hash around $2^{\frac{h}{2}}$ messages, and requires at least $46 \times 2^{\frac{h}{2}}$ calls to the round permutation. Its time complexity basically corresponds to the cost of $2^{\frac{h}{2}}$ hashing.

3 General principle of the internal collision attack

Our attack against Maraca consists in finding two padded messages of the same length which lead to the same 1024-bit internal state. The attack exploits the fact that the message block inserted at each round has the same size as the whole internal state. This property may enable the attacker to control the whole internal state. This section first describes the general principle of the attack and exhibits the underlying property of the inner permutation. However, we will show in Section 3.3 that the time or memory complexity of the attack might be higher than for the generic collision attack in some cases. This might be overcome by exploiting some algebraic structure of the inner permutation. The first case, described in Section 3.4, is when the set of input differences D which is considered contains a large linear or affine subspace. The second case, presented in Section 3.5 and which will be used for Maraca, is when there exists a large (affine) subspace whose almost all elements belong to D. This second case actually enables the attacker to use a sieving phase which decreases the time complexity of the general attack.

3.1 Constructing two sets of messages leading to an internal collision

We consider two sets of padded messages using a given 1024-bit key K. Since all considered messages before padding are composed of 49 blocks of 1024 bits, all of them are post-padded with the same value, pad, which only depends on K and on the message length. This value does not play any role in the attack since it is the same for all messages and it is involved in the computation after the internal states collide. Both sets of padded messages are defined as follows:

$$\mathcal{A} = \{\mathcal{M}_a = (K, a, 0^{47}, m, \text{pad}),\ a \in \{0,1\}^{1024}\}$$

and

$$\mathcal{B} = \{\mathcal{M}_b = (K, b, 0, x, 0^{45}, m, \text{pad}),\ b \in \{0,1\}^{1024}\}$$

where x and m are two fixed 1024-bit blocks that will be defined later and where 0^i denotes the sequence formed by i occurrences of the all-zero 1024-bit block. In the following, the message blocks are denoted by M_i where i starts from 0, *i.e.*, $M_0 = K$ for all the messages we consider.

Let S_a (resp. S_b) denote the internal state obtained at the beginning of Round 49 when $\mathcal{M}_a$ (resp. $\mathcal{M}_b$) is hashed. We aim at finding a collision on the internal state at Round 49, before the second application of Perm, as depicted on Figure 2. Round 49 for $\mathcal{M}_a$ (resp. $\mathcal{M}_b$) actually consists of the following operations:

- xor m to the current internal state;
- apply Perm to the internal state;
- xor 0 (resp. x''') to the internal state;
- apply two additional iterations of Perm.

This comes from the fact that all message blocks M_i, $3 \leq i \leq 48$, in $\mathcal{M}_a$ vanish, implying that there is no message insertion after the first application of Perm at Round 49. All message blocks M_i, $3 \leq i \leq 48$, in $\mathcal{M}_b$ vanish except $M_3 = x$, implying that x''', corresponding to x rotated by 6×128, is xored to the internal state after the first application of Perm at Round 49. Then, all message blocks which are inserted after Round 49 are equal for both message sets. Thus, an internal collision occurs as soon as we are able to find three message blocks a, b and m which satisfy

$$\text{Perm}(S_a \oplus m) = \text{Perm}(S_b \oplus m) \oplus x'''. \qquad (1)$$

It is worth noticing that both S_a and S_b are independent of m.

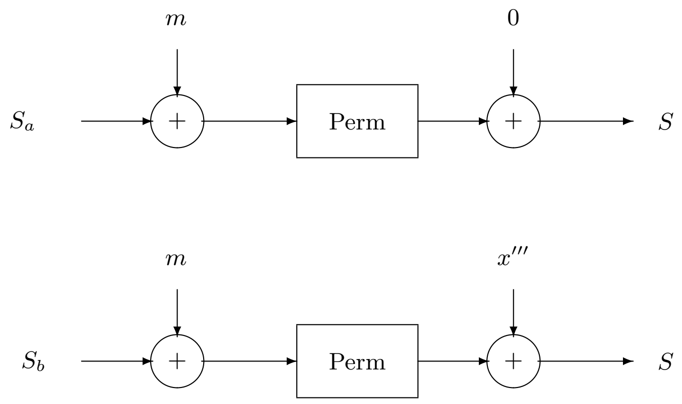

Fig. 2. Beginning of Round 49 for $\mathcal{M}_a$ (top) and $\mathcal{M}_b$ (bottom)

3.2 Underlying property of the inner permutation Perm

We now investigate the underlying property of Perm which makes Maraca vulnerable to the previously described attack. In the following, we express it in a more general setting which will be useful when we will consider other choices for Perm. Then, we denote by n the size of the internal state and by h the length of the message digest.

Equation (1) with $v = m \oplus S_a$ shows that finding an internal collision for both previously described message sets is equivalent to finding a pair (S_a, S_b) of internal states in $\mathbf{F}_2^n$ such that

$$\exists v \in \mathbf{F}_2^n, \;\; \mathrm{Perm}(v \oplus S_a \oplus S_b) \oplus \mathrm{Perm}(v) = \delta, \tag{2}$$

for a fixed value of δ chosen by the attacker. Let $D(\delta)$ denote the set of all input differences such that (2) holds, *i.e.*,

$$D(\delta) = \{\alpha \in \mathbf{F}_2^n, \;\; \exists v \in \mathbf{F}_2^n, \;\; \mathrm{Perm}(v \oplus \alpha) \oplus \mathrm{Perm}(v) = \delta\}.$$

In the case of any ambiguity on the function we consider, this set will be indexed by the function, *e.g.* $D_{\mathrm{Perm}}(\delta)$.

Then, the attack consists in finding a pair (S_a, S_b) of internal states such that $(S_a \oplus S_b) \in D(\delta)$. As a comparison, the generic birthday attack for finding an internal collision consists in finding a pair (S_a, S_b) of internal states in $\mathbf{F}_2^n$ such that $S_a \oplus S_b = 0$.

Then, randomly choosing $N_a = N_b = 2^{\frac{n}{2}} |D(\delta)|^{-1/2}$ messages in $\mathcal{A}$ and in $\mathcal{B}$ enables us to find a pair of internal states (S_a, S_b) at the beginning of Round 49 with $S_a \oplus S_b \in D(\delta)$. The data complexity of our attack, *i.e.* the number of calls to the hash function, is therefore smaller than the data complexity of the generic internal collision attack as soon as there exists an output difference δ such that $|D(\delta)| > 1$. In the case where the size of the internal state, n, is larger that the length h of the message digest, as in Maraca, our attack leads to a collision attack with data complexity smaller than the generic collision attack if there exists a difference δ such that $|D(\delta)| > 2^{n-h}$. Note that, in our attack, each call to the hash function actually corresponds to 49 calls to the round function since the first 49 blocks in each message $\mathcal{M}_a$ and $\mathcal{M}_b$ have to be proceeded but message block 0 is constant and has to be evaluated only once. As a comparison, the generic collision attack requires at least 46 calls to the round functions (and 30 additional calls to Perm) for each message which is hashed.

3.3 Time complexity of the general attack

However, if the set of input differences $D(\delta)$ does not have any particular structure, determining whether two internal states are such that $S_a \oplus S_b \in D(\delta)$ might be very time-consuming. The only general strategy which may have time complexity lower than $2^{\frac{n}{2}}$ consists in storing all N_a values of S_a and all N_b values of S_b in two tables. Then, all N_aN_b differences must be computed and compared to the elements in $D(\delta)$. This procedure has time complexity

$$N_aN_b \log(|D(\delta)|) = 2^n \frac{\log(|D(\delta)|)}{|D(\delta)|}.$$

The attack is then faster than the generic internal collision attack only if $|D(\delta)| > 2^{\frac{n}{2}}$, and it is faster than the generic collision attack only if $|D(\delta)| > 2^{n-\frac{h}{2}}$. But, in general, comparing all differences $S_a \oplus S_b$ with the elements of $D(\delta)$ requires the storage of $D(\delta)$, which needs an amount of memory higher than for the generic attack. However, this memory complexity can be much lower in some cases, for instance, if Perm corresponds to the concatenation of several copies of a smaller $\ell \times \ell$ Sbox (eventually followed by an affine permutation), then the attacker has to store the elements in

$$D_P(\delta') = \{\alpha \in \mathbf{F}_2^\ell,\ \ \exists m \in \mathbf{F}_2^\ell,\ \ P(m \oplus \alpha) \oplus P(m) = \delta'\}$$

only, for some $\delta' \in \mathbf{F}_2^\ell$.

3.4 Exploiting the algebraic structure of $D(\delta)$

Determining whether $S_a \oplus S_b \in D(\delta)$ for all (S_a, S_b) is much easier when $D(\delta)$ has a simple algebraic structure. The simplest case is when $D(\delta)$ is a linear or an affine subspace of dimension d. Then, it can be expressed as $D(\delta) = c + \langle e_1, \dots e_d\rangle$ where $(e_1, \dots, e_d)$ is a basis of the corresponding linear subspace and c is a constant vector in $\mathbf{F}_2^n$. Let $(e_{d+1}, \dots, e_n)$ be $(n-d)$ vectors in $\mathbf{F}_2^n$ such that $e_1, \dots e_n$ form a basis of $\mathbf{F}_2^n$. Then, an element $x \in \mathbf{F}_2^n$ belongs to $D(\delta)$ if and only if, for all i, $d+1 \leq i \leq n$,

$$x \cdot e_i = c \cdot e_i,$$

where $x \cdot y$ denotes the usual scalar product. Therefore, all pairs (S_a, S_b) with $S_a \oplus S_b \in D(\delta)$ can be found by storing in a table the $(n-d)$-bit words s_a composed of the $(n-d)$ coordinates $(S_a \cdot e_i)$, $d+1 \leq i \leq n$. Then, for each S_b, the attacker computes the $(n-d)$-bit word $s_b = (S_b \cdot e_i)_{d+1 \leq i \leq n}$ and checks whether $s_b \oplus c$ belongs to the table where c is the constant defining the affine subspace.

Then, when $D(\delta)$ is an affine subspace of dimension d, its size is 2^d, implying that the time complexity of the attack is $2(n-d)N_a = 2(n-d)2^{\frac{n-d}{2}}$. It requires a table of $(n-d)2^{\frac{n-d}{2}}$ bits. The time complexity of this attack is then always lower than the generic internal collision attack, and it improves the generic collision attack if $d > n-h$.

It is worth noticing that the attack only exploits the fact that any element in the considered subspace belongs to $D(\delta)$. Therefore, the same attack can be mounted if $D(\delta)$ contains an affine subspace V of dimension d, but in this case, we have $N_a = N_b = 2^{\frac{n-d}{2}}$ instead of $N_a = N_b = 2^{\frac{n}{2}}|D(\delta)|^{-\frac{1}{2}}$.

3.5 Using a sieving phase

In the case where the largest (affine) subspace included in $D(\delta)$ has dimension $d \leq n-h$, then the time complexity of our attack exceeds the time complexity of the generic collision attack. Then, the existence of a larger (affine) subspace V of dimension d which contains many elements of $D(\delta)$ can be used as a sieve for selecting the pairs (S_a, S_b) whose differences belong to $D(\delta)$. The attack then aims at finding a pair (S_a, S_b) such that $(S_a \oplus S_b) \in (D(\delta) \cap V)$. The data complexity has now increased to

$$N_a = N_b = 2^{\frac{n}{2}} |D(\delta) \cap V|^{-1/2}$$

which improves the generic collision attack if

$$|D(\delta) \cap V| > 2^{n-h}.$$

But, the time complexity is much lower. Actually, once the much smaller list of pairs with difference in V has been obtained, all differences $(S_a \oplus S_b)$ from this list can be exhaustively computed until a difference in $D(\delta) \cap V$ is found. The sieving phase selects

$$\frac{N_a N_b}{2^{n-d}} = 2^d \frac{1}{|D(\delta) \cap V|}$$

pairs (S_a, S_b) among the $2^n \frac{1}{|D(\delta) \cap V|}$ possible pairs. The overall time complexity is then

$$2(n-d) 2^{\frac{n}{2}} (|D(\delta) \cap V|)^{-\frac{1}{2}} + 2^d \log_2(|D(\delta) \cap V|)(|D(\delta) \cap V|)^{-1},$$

where the last term is the cost for checking whether a difference in the previous list belongs to $D(\delta) \cap V$. The attack is then faster than the generic collision attack as soon as the proportion of elements in V which belong to $D(\delta)$, *i.e.* $2^{-d}|D(\delta) \cap V| = |V|^{-1}|D(\delta) \cap V|$ exceeds $2^{-\frac{h}{2}}$.

4 Application to the inner permutation used in Maraca

4.1 Structure of the inner permutation Perm.

The inner permutation Perm used in Maraca is formed by 128 parallel applications of a unique 8 × 8 permutation P whose first three output bits are linear:

$$\begin{aligned} P_1(x_0, \ldots, x_7) &= (x_0 \oplus x_4 \oplus x_5 \oplus x_7) \\ P_2(x_0, \ldots, x_7) &= (x_1 \oplus x_2 \oplus x_3 \oplus x_5) \\ P_3(x_0, \ldots, x_7) &= (x_1 \oplus x_3 \oplus x_4 \oplus x_5) \end{aligned}$$

and the other five output bits have a higher degree. A constant is then added to all 1024 bits and a bit permutation is applied to the resulting 1024-bit output. Perm can then be seen as a function which applies to a 128-byte word $(b_1, \ldots, b_{128})$, and which outputs

$$\sigma(P(b_1), \ldots, P(b_{128}))$$

where σ is a permutation of the 1024 bits composing the internal state, *i.e.*,

$$\sigma(x_1, \ldots, x_{1024}) = (x_{\pi(1)}, \ldots, x_{\pi(1024)})$$

with π a permutation of $\{1, \ldots, 1024\}$.

4.2 Differential properties of Perm

We now focus on the difference table of the 8×8 Sbox P used in Perm. This difference table enables us to determine for each nonzero output difference δ the set of input differences which can lead to δ, *i.e.*,

$$D_P(\delta) = \{\alpha \in \mathbf{F}_2^8,\ \exists x \in \mathbf{F}_2^8,\ P(x \oplus \alpha) \oplus P(x) = \delta\}.$$

Since the first three coordinates of P, P_i, $1 \leq i \leq 3$, are linear, we have that any $\alpha \in D_P(\delta)$ must satisfy

$$(P_1(\alpha), P_2(\alpha), P_3(\alpha)) = (\delta_1, \delta_2, \delta_3). \tag{3}$$

Then, $D_P(\delta)$ is included in the 5-dimensional affine subspace defined by

$$(\delta_1, \delta_2, \delta_3, 0, 0, 0, 0, 0) \oplus \langle e_4, \ldots, e_8 \rangle$$

where $e_1, \ldots, e_8$ is the canonical basis of $\mathbf{F}_2^8$.

Now, we search for the output difference δ for which the size $|D_P(\delta)|$ is maximal. The highest value which can be obtained for this size is 21, and it can be reached for 20 output differences δ. An example of a such an output difference is $\delta =$ `0x3`.

4.3 Attack on Maraca-512

We now describe the concrete attack on Maraca. Using the notation defined in Sections 2 and 3, we choose the message block x such as its rotated version x''' equals the 128-byte word $\sigma(\delta, \ldots, \delta)$ where δ is an output difference for P which can be obtained from 21 input differences, *e.g.* $\delta =$ `0x3`. It follows that any input difference in

$$D = \{(\alpha, \ldots, \alpha),\ \alpha \in D_P(\delta)\}$$

can lead to the output difference x'''. In other words, for each pair of internal states (S_a, S_b) such that $S_a \oplus S_b$ belongs to D, there exists a message block m such that

$$\mathrm{Perm}(m \oplus S_a) = \mathrm{Perm}(m \oplus S_b) \oplus x'''.$$

Here, $|D| = (21)^{128}$, implying that we need $N_a = N_b = 2^{230.5}$.

We then use the particular structure of P for sieving the pairs (S_a, S_b): the set of input differences is included in an affine subspace V of dimension 640 (note that this is a particular case of the attack described in Section 3.5 where it was allowed that some elements of $D(\delta)$ do not belong to V). Using this subspace, we are able to find all pairs (S_a, S_b) whose differences belong to V. The average number of such pairs (S_a, S_b) is

$$\frac{N_a N_b}{2^{384}} = 2^{77}.$$

Now, for those 2^{77} favorable pairs of internal states, we have to check whether $(S_a \oplus S_b)$ belongs to D. This occurs with probability

$$\frac{|D|}{2^{5\times 128}} = 2^{-77}.$$

Once such a pair has been found, we can pick up a value of m which makes possible to obtain the desired output difference from the input difference $S_a \oplus S_b$. Such an m can be constructed as a 128-byte word $(\mu_1, \dots, \mu_{128})$ defined by

$$P(\mu_i \oplus (S_a)_i) \oplus P(\mu_i \oplus (S_b)_i) = \delta$$

where $(S_a)_i$ (resp. $(S_b)_i$) is the i-th byte in S_a (resp. S_b).

This procedure then leads to a pair of messages $\mathcal{M}_a \in \mathcal{A}$ and $\mathcal{M}_b \in \mathcal{B}$ such that

$$\mathrm{Perm}(S_a \oplus m) = \mathrm{Perm}(S_b \oplus m) \oplus x''',$$

i.e., to an internal collision after Round 49. Since all the blocks which must be inserted in the following rounds are the same for both messages, we clearly obtain an internal collision after the computation of the hash value. The attack then requires fewer than $2^{231.5} \times 49 = 2^{237}$ calls to the round function. The memory complexity is $2^{230.5}$ bits. From the analysis in Section 3.5, we deduce that the overall time complexity is 2^{240} operations, which is clearly less than for the generic collision attack when the length of the message digest is greater than or equal to 512. Note that the complexity of the last step of the attack, *i.e.* after sieving is negligible.

5 Resistance of other inner permutations to the attack

Since our attack does not exploit any classical weakness of Perm, we may wonder whether it comes from a unlucky choice for Perm and whether more appropriate choices for Perm could be easily found.

A permutation is vulnerable to our attack (in the sense that our attack improves the generic collision attack) if there exists an output difference δ such that one of the following conditions holds:

1. $|D(\delta)| > 2^{n-\frac{h}{2}}$;
2. there exists an (affine) subspace V such that $|D(\delta) \cap V| > 2^{n-h}$ and the proportion of elements in V which belong to $D(\delta)$ exceeds $2^{-\frac{h}{2}}$.

Condition 1 corresponds to the attack without any sieving, as described in Section 3.3. Condition 2 corresponds to the attack when the differences in the d-dimensional subspace V are first selected. From the analysis of Section 3.5, the data complexity of the attack is then $2^{\frac{n}{2}}|D(\delta) \cap V|^{-\frac{1}{2}}$, which must be less than $2^{\frac{h}{2}}$. The cost for finding the differences in $D(\delta)$ after sieving is then proportional to $|V|/|D(\delta) \cap V|$, implying that the time complexity is less than $2^{\frac{h}{2}}$ if the condition on the proportion of elements of $D(\delta)$ in V is satisfied. It is also worth noticing that Condition 2 includes the case where $D(\delta)$ contains an affine subspace of dimension at least $n - h$; this corresponds to the case where the proportion of elements of $D(\delta)$ in V is equal to 1. We now investigate these conditions. Note that both of them provide a lower bound on the size of $D(\delta)$ since Condition 2 implies that $|D(\delta)| > 2^{n-h}$.

5.1 Size of $D(\delta)$ and link with differential cryptanalysis

Because a large $|D(\delta)|$ is a necessary condition for resisting our attack, we first focus on its maximal value for a permutation F over $\mathbf{F}_2^n$. We denote by D_F the following parameter

$$D_F = \max_{\delta \in \mathbf{F}_2^n} |D_F(\delta)|.$$

A suitable permutation F for Maraca must have a small D_F. However, we can show that there is a trade-off between a small D_F and a good resistance to differential cryptanalysis.

It is well-known that differential cryptanalysis [2] exploits the fact that the nonlinear functions used in a primitive are such that the difference between the images of two inputs with a given difference takes the same value with a high probability. Therefore, the resistance of a function $F : \mathbf{F}_2^n \rightarrow \mathbf{F}_2^n$ to this attack is quantified by the following parameter [3,4],

$$\varDelta_F = \max_{\alpha,\beta \in \mathbf{F}_2^n,\ \alpha \neq 0} \varDelta(\alpha,\beta) \text{ with } \varDelta(\alpha,\beta) = |\{x \in \mathbf{F}_2^n, F(x \oplus \alpha) \oplus F(x) = \beta\}|.$$

A function with $\varDelta_F = \varDelta$ is said to be differentially $\varDelta$-uniform. This parameter $\varDelta$ must be as small as possible. Since any equation

$$F(x \oplus \alpha) \oplus F(x) = \beta$$

has an even number of solutions x, the minimal value for $\varDelta$ is 2 and the functions for which $\varDelta = 2$ are named almost perfect nonlinear (APN). However, since the existence of APN permutations of an even number of variables is an open problem, most permutations used in symmetric ciphers are differentially 4-uniform; the most famous example is the inverse function over the field $\mathbf{F}_{2^n}$ used in the AES.

Now, the relationship between both quantities D_F and $\varDelta_F$ comes from the following simple observation.

Proposition 1. *Let F be a permutation over $\mathbf{F}_2^n$. For any $\delta \in \mathbf{F}_2^n$ we have:*

$$\begin{aligned} D(\delta) &= \{\alpha \in \mathbf{F}_2^n, \exists x \in \mathbf{F}_2^n \text{ with } F(x \oplus \alpha) \oplus F(x) = \delta\} \\ &= \{F^{-1}(x \oplus \delta) \oplus F^{-1}(x),\ \ x \in \mathbf{F}_2^n\}. \end{aligned}$$

Proof. Let $x \in \mathbf{F}_2^n$ be a solution of

$$F(x \oplus \alpha) \oplus F(x) = \delta.$$

With $y = F(x)$, this equation can equivalently be written as

$$y \oplus \delta = F(x \oplus \alpha)$$

that means

$$F^{-1}(y \oplus \delta) = F^{-1}(y) \oplus \alpha.$$

We then deduce that the set $D(\delta)$ consists of all values $(F^{-1}(y \oplus \delta) \oplus F^{-1}(y))$ when y varies in $\mathbf{F}_2^n$.

From this simpler expression of $D(\delta)$, we deduce that any permutation F with a small Δ_F has a high D_F.

Theorem 1. *Let F be a permutation over $\mathbf{F}_2^n$. If F is differentially Δ-uniform, then, for any $\delta \in \mathbf{F}_2^n$, we have*

$$|D(\delta)| \geq \frac{2^n}{\Delta}$$

and equality holds for $\delta \neq 0$ if and only if for all $\alpha \in \mathbf{F}_2^n$, the equations

$$F^{-1}(x \oplus \delta) \oplus F(x) = \alpha,$$

have either 0 or Δ solutions.

Proof. Let $\Delta(\delta, \alpha)$ denote the number of solutions $x \in \mathbf{F}_2^n$ of

$$F^{-1}(x \oplus \delta) \oplus F^{-1}(x) = \alpha.$$

Then, we have

$$\sum_{\alpha \in \mathbf{F}_2^n} \Delta(\delta, \alpha) = 2^n$$

and

$$\sum_{\alpha \in \mathbf{F}_2^n} \Delta(\delta, \alpha) \leq \max_{\alpha} \Delta(\delta, \alpha) |D(\delta)| \leq \Delta_{F^{-1}} |D(\delta)|,$$

with equality if and only if

$$\forall \alpha \in \mathbf{F}_2^n,\ \Delta(\delta, \alpha) \in \{0, \Delta_{F^{-1}}\}.$$

Using that

$$\Delta_{F^{-1}} = \max_{\delta \neq 0, \alpha} \Delta(\delta, \alpha) = \Delta_F,$$

since both F and F^{-1} have the same parameter Δ [5], we deduce that, for any $\delta \neq 0$,

$$\Delta_F |D(\delta)| \geq 2^n.$$

Moreover, we obviously have

$$|D(0)| = 1$$

for any permutation F, completing the proof.

We then deduce the following direct corollary.

Corollary 1. *Let F be a permutation over $\mathbf{F}_2^n$. Then*

$$D_F = \max_{\delta \in \mathbf{F}_2^n} |D_F(\delta)| = 1$$

if and only if F has degree 1.

Proof. Any function obviously satisfies $\varDelta_F \leq 2^n$ with equality if and only if F has degree 1.

This corollary notably implies that, for any choice of Perm (except the trivial case where the hash function is linear), our attack requires fewer calls to the round function than the generic internal collision attack (without any consideration of time and memory complexity).

Let us now investigate some a priori natural choices for Perm and their impact on the complexity of our attack. For obvious implementation reasons, we assume that Perm consists of the concatenation of several copies of the same smaller Sbox P, eventually followed by a linear permutation as in the original function.

Example 1. Since no APN permutation of an even number of variables is known, we can slightly modify the size of the internal state, $n = mk$ with m odd and choose for P an APN permutation over $\mathbf{F}_2^m$. For instance, $m = 9$ and $k = 128$ could be an appropriate choice. From Theorem 1, we deduce that, for any nonzero $\delta \in \mathbf{F}_2^m$, $|D_P(\delta)| = 2^{m-1}$ since P^{-1} is APN, *i.e.* all equations $P^{-1}(x \oplus \delta) \oplus P^{-1}(x)$ have either 0 or 2 solutions. It follows that, for any $\delta = (\delta_1, \ldots, \delta_k) \in (\mathbf{F}_2^m)^k$ with all $\delta_i \neq 0$,

$$|D_{\mathrm{Perm}}(\delta)| = 2^{k(m-1)}.$$

Our attack (without sieving) then requires to hash

$$N_a = N_b = 2^{\frac{k}{2}}$$

messages in $\mathcal{M}_a$ and $\mathcal{M}_b$, where k is the number of copies of P. All the 2^k differences $(S_a \oplus S_b)$ can then be computed and compared to the elements of $D_{\mathrm{Perm}}(\delta)$. Due to the concatenated structure of Perm, checking whether each $(S_a \oplus S_b)$ belongs to $D_{\mathrm{Perm}}(\delta)$ costs at most

$$\sum_{i=1}^{k} \log_2(|D_P(\delta_i)|) = k(m-1) \text{ operations,}$$

leading to an overall time complexity less than or equal to $k(m-1)2^k$. This improves the generic collision attack as soon as the length of the message digest exceeds $2(k + \log(n-k))$ where n is the size of the internal state and k is the number of copies of P in Perm. For $k = 128$ and $n = 9 \times 128$, this corresponds to $h \geq 276$. The memory complexity corresponds to the storage of all internal states S_a and S_b and of all elements of $D_P(\delta_i)$ (which requires $m2^{m-1}$ bits since the attacker can choose the same value for all δ_i).

Example 2. If we want to keep the original parameters, *i.e.* $k = 128$ and $m = 8$, a natural choice for P is the inverse function over $\mathbf{F}_{2^8}$ as in the AES, or any linearly equivalent permutation. It is well-known that the inverse function over $\mathbf{F}_{2^m}$, m even, is differentially 4-uniform and that the equation

$$(x+\delta)^{-1} + x^{-1} = \gamma,\ \delta \neq 0$$

has 4 solutions x if and only if $\gamma = \delta^{-1}$ [6,5]. Thus, when x varies in $\mathbf{F}_{2^m}$ and differs from these 4 solutions, $((x+\delta)^{-1} + x^{-1})$ takes exactly $(2^{m-1} - 2)$ distinct values since each value is obtained for exactly 2 elements x. Using Proposition 1, we deduce that, when P corresponds to the inverse function over $\mathbf{F}_{2^m}$, for any nonzero $\delta \in \mathbf{F}_{2^m}$, $|D_P(\delta)| = (2^{m-1} - 2) + 1 = 2^{m-1} - 1$. Then, with our parameters, $|D_{\mathrm{Perm}}(\delta)| = (2^7 - 1)^{128} = 2^{894.5}$. Our attack (without sieving) then requires to hash

$$N_a = N_b = 2^{64.7}$$

messages in $\mathcal{M}_a$ and $\mathcal{M}_b$. Even without any sieving, it is faster than the generic collision attack since examining all differences $(S_a \oplus S_b)$ requires

$$128 \times 895 \times 2^{129.4} = 2^{146} \text{ operations}$$

and the memory cost is roughly 2^{76} bits. Therefore, if P is replaced by the inverse function in Maraca, our attack becomes much more efficient and its complexity is lower than the complexity of the generic attack when the length of the message digest exceeds 292.

5.2 Algebraic structure of $D(\delta)$

In the case where Perm is such that D_{Perm} exceeds $2^{n-\frac{h}{2}}$, *i.e.* $D_P \leq 2^{8-\frac{h}{2}}$ when Perm is the concatenation of 128 copies of a permutation P over $\mathbf{F}_2^8$, an efficient attack requires that $D(\delta)$ has a particular structure, as explained in Sections 3.4 and 3.5. For instance, Proposition 1 implies that a particular case where $D_P(\delta)$ is an affine subspace is the case where P^{-1} is quadratic.

Proposition 2. *Let F be a permutation over $\mathbf{F}_2^n$ such that F^{-1} has degree 2. Then, for any $\delta \in \mathbf{F}_2^n$, $\delta \neq 0$, $D(\delta)$ is an affine subspace.*

Note that this does not apply to the permutation P used in Maraca since we have $\deg(P) = 2$ and not $\deg(P^{-1}) = 2$.

More generally, the structure of $D(\delta)$ is an open problem which has been raised in [7,**?**] in the case of subspaces of codimension 1: the permutations F^{-1} such that all sets $D(\delta)$, $\delta \neq 0$ are affine hyperplanes are called *crooked functions*, and they correspond to almost bent functions [8, Lemma 5], which are a particular case of APN functions. However, the only examples of crooked functions known at present have degree 2[7]; they correspond to the case studied in Proposition 2. Here, the search for a good permutation for Maraca raises the more general open issue, related to the converse of Proposition 2.

Open problem 1 *Does there exist any permutation F over $\mathbf{F}_2^n$ with* $\deg(F^{-1}) > 2$ *such that $D(\delta)$ is an (affine) subspace for all $\delta \in \mathbf{F}_2^n$, $\delta \neq 0$.*

But, since our attack requires that $D(\delta)$ has a particular algebraic structure for one difference δ only, and not for all of them, it is related to the following more general problems.

Open problem 2 *Characterize the permutations F over $\mathbf{F}_2^n$ such that, there exists an input difference $\delta \neq 0$ for which*

$$D(\delta) = \{F(x \oplus \delta) \oplus F(x),\ x \in \mathbf{F}_2^n\}$$

is an (affine) subspace.

Finding an inner permutation for Maraca which resists our attack is related to the following issue.

Open problem 3 *For a permutation F over $\mathbf{F}_2^n$, find the smallest integer $h > 0$ such that there exists an input difference $\delta \neq 0$ and an (affine) subspace V which satisfy*

$$|D(\delta) \cap V| > 2^{n-h}$$

and

$$\frac{|V|}{|D(\delta) \cap V|} < 2^{\frac{h}{2}}.$$

6 Conclusions

We have presented an internal collision attack against Maraca, with complexity lower than the complexity of the generic attack. Besides this concrete cryptanalysis, the main interest of our attack is that the underlying weakness corresponds to some differential properties of the inner permutation which, to our best knowledge, have not been exploited before. Moreover, these differential properties are, in some sense, in contradiction with the security criterion corresponding to differential cryptanalysis. For instance, it appears that replacing the original permutation of Maraca by a commonly used Sbox increases the vulnerability of the hash function. Finding an inner permutation which resists our attack for 512-bit message digests is not an easy task, which is related to some interesting theoretical problems on Boolean functions.

References

1. Jr., R.J.J.: Maraca - algorithm specification. Submission to NIST (2008)
2. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology **4** (1991) 3–72
3. Nyberg, K., Knudsen, L.: Provable security against differential cryptanalysis. In: Advances in Cryptology - CRYPTO'92. Volume 740 of Lecture Notes in Computer Science., Springer-Verlag (1993) 566–574

4. Nyberg, K., Knudsen, L.: Provable security against a differential attack. Journal of Cryptology **8** (1995) 27–37
5. Nyberg, K.: Differentially uniform mappings for cryptography. In: Advances in Cryptology - EUROCRYPT'93. Volume 765 of Lecture Notes in Computer Science., Springer-Verlag (1993) 55–64
6. Carlitz, L., Uchiyama, S.: Bounds for exponential sums. Duke Math J. (1957) 37–41
7. Bending, T., der Flass, D.F.: Crooked functions, bent functions, and distance regular graphs. Electron. J. Combin. **5** (1998) R34.
8. Canteaut, A., Charpin, P.: Decomposing bent functions. IEEE Transactions on Information Theory **49** (2003) 2004–19