Internal collision attack on Maraca

Anne Canteaut and María Naya-Plasencia

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. The cryptanalysis mainly exploits two features of Maraca: the fact that the message block inserted at each round has the same size as the internal state, and some particular differential properties of the inner permutation.

1 Brief description of Maraca

Maraca is a new keyed hash function which has been submitted to the SHA-3 competition [1]. 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 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 $i+46$.

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 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.

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.

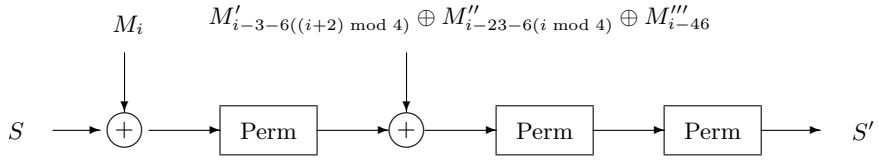

Figure1. Round i in Maraca

Structure of the permutation Perm. The permutation Perm is formed by 128 parallel applications of a unique 8×8 permutation P whose first three output bits are linear:

$$P_1(x_0, \ldots, x_7) = (x_0 \oplus x_4 \oplus x_5 \oplus x_7)$$
$$P_2(x_0, \ldots, x_7) = (x_1 \oplus x_2 \oplus x_3 \oplus x_5)$$
$$P_3(x_0, \ldots, x_7) = (x_1 \oplus x_3 \oplus x_4 \oplus x_5)$$

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 $\{1, \ldots, 1024\}$.

2 An important differential property of Perm

We 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 β the set of input differences which can lead to β, *i.e.*,

$$D_P(\beta) = \{\alpha \in \{0,1\}^8,\ \exists x \in \{0,1\}^8,\ P(x \oplus \alpha) \oplus P(x) = \beta\}.$$

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

$$(P_1(\alpha), P_2(\alpha), P_3(\alpha)) = (\beta_1, \beta_2, \beta_3). \quad (1)$$

Then, $D_P(\beta)$ is the intersection of the 5-dimensional vector space defined by (1) and the set

$$D_{P'}(\beta') = \{\alpha \in \{0,1\}^8,\ \exists x \in \{0,1\}^8,\ P'(x \oplus \alpha) \oplus P'(x) = \beta'\}$$

where P' is the mapping from $\{0,1\}^8$ to $\{0,1\}^5$ which corresponds to the five last coordinates of P only, and β' is the vector corresponding to the five last coordinates of β.

Now, we search for the output difference β which can be obtained from the highest number of input differences α, *i.e.*, for which the size $|D_P(\beta)|$ is maximal. The highest value which can be obtained for this size is 21, and it can be reached for 20 output differences β. An example of a such an output difference is $\beta = \texttt{0x3}$.

3 Internal collision attack on Maraca

Our attack consists in finding two padded messages of the same length with lead to the same 1024-bit internal state. It requires 2^{237} calls to the round function, which corresponds to a lower complexity than the generic collision attack for any hash length $h \geq 512$. The attack exploits the following two features of Maraca: the fact that the message block inserted at each round has the same size as the whole internal state and the particular structure of the Perm function. The first property may enable the attacker to control the whole internal state, and the second one enables the use of a sieving technique for finding a collision on the internal state.

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

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

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

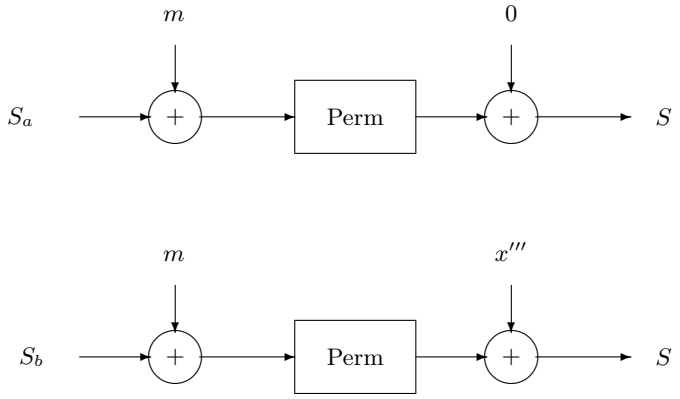

Figure2. Beginning of Round 49 for $\mathcal{M}_a$ (top) and $\mathcal{M}_b$ (bottom)

Now, we choose the message block x such as its rotated version x''' equals the 128-byte word $\sigma(\beta, \ldots, \beta)$ where β is an output difference for P which can be obtained from 21 input differences, *e.g.* $\beta = \texttt{0x3}$. It follows that any input difference in

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

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

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

The probability that an input difference belongs to D is $\frac{D}{2^{1024}}$. Then, randomly choosing $N_a = N_b = 2^{512}|D|^{-1/2}$ messages in $\mathcal{A}$ and in $\mathcal{B}$ enables to find a pair of internal states (S_a, S_b) at the beginning of Round 49 with $S_a \oplus S_b \in D$. Here, $|D| = (21)^{128}$, implying that we need $N_a = N_b = 2^{230.5}$.

However, if the set of input differences D does not have any particular structure and if its size is smaller than 2^{n-h} where n is the size of the internal state and h the length of the message digest, then determining whether two internal states are such that $S_a \oplus S_b \in D$ requires to examine all pairs (S_a, S_b); this requires more than $2^{h/2}$ operations.

Here, we use that the particular structure of P enables a divide-and-conquer search for such pairs. Actually, the set of input differences is included in an affine subspace V of codimension 384. For a given value of S_b, checking whether there exists S_a such that $S_a \oplus S_b$ belongs to D does not require to examine all possible values of S_a since only the values S_a such that $S_a \oplus S_b \in V$ must be considered. Based on this principle, the following attack can be mounted: for each of the N_a messages $\mathcal{M}_a$, we store in a table the value S_a and the 384 coordinates of $\mathrm{Perm}(S_a)$ which have degree 1. We then sort the table according to the 384-bit value of $\mathrm{Perm}(S_a)$. And, for each of the N_b messages $\mathcal{M}_b$, we compute the 384 coordinates of $\mathrm{Perm}(S_b) \oplus x'''$ which have degree 1 and we check whether there exists a state S_a such $\mathrm{Perm}(S_a) = \mathrm{Perm}(S_b) \oplus x'''$ for the 384 coordinates of degree 1 by a simple table look-up. All such pairs can then be determined by sorting the table of size $2^{230.5}$ and then by $2^{230.5}$ table lookups, while a generic collision attack for a 512-bit hash function requires exactly the same operations but for a table of size 2^{256}.

The average number of pairs (S_a, S_b) corresponding to a match on the 384 bits of degree 1 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 the remaining 5×128 coordinates of $\mathrm{Perm}(S_a) \oplus \mathrm{Perm}(S_b)$ equal the corresponding bits of x'''. Equivalently, we have to check whether $(S_a \oplus S_b)$ belongs to D. Since all the considered values of $(S_a \oplus S_b)$ belong to a subspace of codimension 384, the probability that $S_a \oplus S_b \in D$ is equal to

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

Once such a match 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 $(\mu_1, \ldots, \mu_{128})$ defined by

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

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., 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 while computation of the hash value. The attack then requires fewer than $2^{231.5} \times 49 = 2^{237}$ calls to the round function: 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, and only the beginning of Round 49 must be computed. The memory complexity is $2^{230.5}$ bits. The overall complexity is clearly less than for the generic collision attack when the length of the message digest is greater than or equal to 512.

References

1. Robert J. Jenkins Jr. Maraca - algorithm specification. Submission to NIST, 2008.