Two attacks on RadioGatún

Dmitry Khovratovich

University of Luxembourg

dmitry.khovratovich@uni.lu

Abstract. We investigate the security of the hash function design called RadioGatún in a recently proposed framework of sponge functions. We show that previously introduced symmetric trails can hardly be used to construct collisions and to find a second preimage efficiently. As a generalization of truncated differentials, trails with linear and non-linear restrictions on differences are proposed. We use these trails to find semi-free-start collisions and second preimages with the meet-in-the middle approach and the complexity in the gap between claimed security level and the birthday bound. We also provide some observations on lower bounds on the complexity of our methods with respect to the length of the trail used. This is the best attack on RadioGatún.

Keywords: hash functions, cryptanalysis, sponge.

RadioGatún [1], the subject of this paper, is a design of hash functions proposed by Bertoni et al. at the Second Cryptographic Hash Workshop in 2006. Though having been presented as a so called *iterative mangling function* it actually fits the *sponge* framework later proposed by the same authors [2, 3]. The hash functions Panama [6] and Grindahl [10] also have much common with RadioGatún and the sponge framework.

The sponge is an iterative construction, which is an alternative to the Merkle-Damgård design. The latter approach consists of the iterated application of the compression function, which gets a message block as the input and assumed to be collision-resistant. The sponge construction operates on smaller message blocks and a round function. After a message is fully processed the sponge generates output of infinite length by just consecutively applying the round function and taking a block in an internal state as a new output block.

Bertoni et al. proved [2] that such a construction is resistant against collision and (second-)preimage attacks of complexity lower than the birthday bound assuming that the round function is a randomly chosen permutation which properties are not exploited by an adversary. However, this assumption is not the case for concrete sponge-based hash functions so the designers claim a reduced security level (see the next section for careful explanation).

RadioGatún is actually a family of hash functions with the size l_w of the building block — word — as a parameter. Although the internal state of the hash function is rather big (58 words), the performance is quite impressive. For

example, RadioGatún with $l_w = 32$, which is claimed to be as secure as SHA-256, is twice faster [1]. This makes RadioGatún a very promising design in view of the NIST hash function competition [8].

This paper presents two attacks on RadioGatún: semi-free-start (or chosen IV) collision search and the second preimage search. The outline of the paper is as follows. First we describe the RadioGatún hash function and discuss on the claimed security level: why it is much lower than an intuitive bound with respect to the size of the internal state. In Section 2 we investigate the differential-based collision attacks on RadioGatún using the notion of differential trail. We show that previously introduced symmetric trails [1] do not provide attacks with reasonable complexity. We introduce trails with truncated differentials of linear form, which are extremely suitable for RadioGatún due to its slow diffusion.

Then we present collision and second-preimage attacks based on the trails discussed above. Both attacks use the invertibility of the round function and the abscence of the message scheduling in order to apply the meet-in-the middle approach. Collisions are found in the chosen IV framework though a bit slower second-preimage attack can be also converted to the collision search. The complexities of both attacks are in the gap between the claimed security level and the bound given by the birthday paradox. We also provide some theoretical observations on the lower bound on the complexities of the attacks which can be maintained with the truncated differential trails and the meet-in-the-middle approach.

1 RadioGatún

Description. RadioGatún operates on words of some integer length l_w. The parameter is fixed for a concrete hash function, and we denote by RadioGatún-l_w the corresponding hash function. An internal state of RadioGatún consists of two substates also called *belt* (B) and *mill* (A) of size 39 words and 19 words, respectively. The round function treats them differently. The belt is updated by a simple linear transformation and fed with 12 words of the mill and with 3 words of the message block in a linear way. The mill is fed with 3 words of the belt and 3 words of the message block in a linear way and afterwards is updated by a nonlinear function. The resulting round function is invertible (see Fig. 1).

There is no message scheduling in RadioGatún. First a message to be hashed is appropriately padded and then it is divided by 3-word blocks, which are used only once. The iteration starts with the state full of 0s. One step consists of the message injection and the application of the round function. After the message is fully processed RadioGatún iterates with 16 blank rounds (without any injections) and generates output of infinite length by just consecutively applying the round function and taking a 2-word block in the internal state[1] as a new output block.

We denote the injected block by M. Following this notation, the round function of RadioGatún transforms a state $S = (A, B)$ to a new state $S' = (A', B')$:

[1] More precisely, the second and the third words of the mill.

– $B \xleftarrow{\text{Message injection } (M) \text{ and shift}} B$;
– $A \xleftarrow{\text{Message injection } (M)} A$;
– $B' \xleftarrow{\text{Mill2Belt feedforward } (A)} B$;
– $A \xleftarrow{\text{Mill function}} A$;
– $A' \xleftarrow{\text{Belt2Mill feedforward } (B')} A$.

Only the Mill function is non-linear. Due to space limits a full description is skipped, so we refer to the original paper [1].

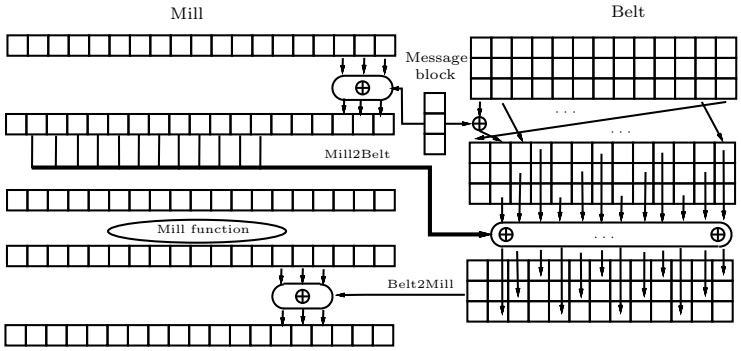

Fig. 1. One round of RadioGatún.

Security. The output of RadioGatún can be considered as a pseudorandom generator, which generates 2 words per step. The designers assume that each application will choose its own length of the hash digest. While for short l-bit outputs the complexity of collision search can be estimated as $2^{l/2}$ RadioGatún calls, this is evidently not for longer ones. As a result, a common security level should have been defined thus providing an upper bound on the complexity of a particular attack.

In the original paper the notion of *capacity* was introduced. The capacity of the ideal iterative mangling function is the size of internal state minus the size of the message block to be injected. However, since the RadioGatún round function is not ideal, the security level of RadioGatún was indicated by a smaller capacity of $19l_w$. This implicitly means that both collision and second-preimage attacks are slower than $2^{9.5l_w}$ though it was not clearly stated. The best non-trivial attack found by the designers requires 2^{46l_w} hash function calls and is substantially slower than the birthday attack, which requires about $2^{27.5l_w}$ hash queries[2]. Now the designers explicitly claim a security level of $2^{9.5l_w}$ operations

[2] The internal state contain 58 words, but a 3-word flexibility is provided by the injection of a message block not used before. See also Sec. 3.

for both the attacks [9] thus following so called *flat sponge claim* [2]. So we conclude that there is a big gap between the birthday bound with respect to the internal state and the security level. Any attack in this gap, though not breaking the security level, could be nevertheless interesting because it should point out weaknesses in the internal transformations.

2 Trails

In order to build a collision we consider so called *differential trails* [1, 5, 7], or simply *trails*. A trail is a pair of [hash function] iterations with restrictions on internal variables. Such restrictions may be imposed on the *differences* between variables or the *values* of particular variables.

This paper is partly inspired by the attack on Grindahl [12]. In this section some notions from that paper are used (control words, degrees of freedom) so we kindly ask the reader to familiarize with it.

2.1 Symmetric trails and trails with fixed differences

If a design operates on words of arbitrary length (like RadioGatún) then one may consider so-called *symmetric trails* that deal with word differences of form $000\ldots0$ and $111\ldots1$. In the original paper on RadioGatún [1] and in the attack on Panama [5] symmetric trails were discussed. Such trails are in some sense independent of the word length. However, their probabilities seem to drastically decrease as l_w grows.

Indeed, authors of [1] found a symmetric trail for the RadioGatún with 1-bit words such that a collision search following this trail would require about 2^{46} operations while the birthday bound is $2^{27.5}$. This observation made authors to claim that corresponding symmetric trail for RadioGatún-l_w (RadioGatún with l_w-bit words) would imply 2^{46l_w} as the complexity of the collision search.

However, the following observation make us to disagree with this generalization. Given a trail with fixed (non-truncated) values of differences (not only symmetric ones) an adversary actually knows the input and output differences $(\Delta_{in}, \Delta_{out})$ of the nonlinear function χ, the part of the Mill function. The pair $\langle \Delta_{in}, \Delta_{out} \rangle$ impose a set of conditions on input and output *values*. The number of conditions imposed on the input value is the Hamming weight of the input difference plus the number of 001-patterns[3] in the difference [5][4].

Let us estimate the average number of conditions. The average Hamming weight of the $19l_w$ bit word is $9.5l_w$, the average number of 001-patterns is $17/8l_w$. Thus we have about $11.5l_w$ conditions on the bits of the mill in each round. If there were no injection to the mill, only two rounds would give enough conditions to completely determine the value of the mill. However, $6l_w$ bits are

[3] The word $\underbrace{001}\,0000\,\underbrace{001}\,\underbrace{001}\,000$ contains three 001-patterns.

[4] Full description of function χ and its properties may be found in Daemen's PhD thesis [4, p. 126].

injected to the mill from the belt ($3l_w$) and the message ($3l_w$) thus compensating $6l_w$ conditions so we have about $5.5l_w$ bit conditions per round. As a result, one can define the values of the mill given only about 4 rounds of a trail. However, since a full collision trail covers at least 6 rounds, with high probability no message pair fits a given trail.

We can reformulate this result as an informal conjecture.

Conjecture 1. Either a differential trail with fixed values of differences has probability 0 or any 4 consecutive rounds of the trail completely define the message pair.

A counterargument may be that one might find a trail with low Hamming weight of the input difference. However, such a difference is likely to expand to an average one due to diffusion properties in the Mill function. So far there is no example of such low-weight trails.

We conclude that symmetric trails and trails with fixed differences seem to be insufficient to evaluate the security of RadioGatún.

2.2 Truncated differentials and linear space of differences

The key idea is to consider truncated differentials of linear form and exploit the linearity of transformations in the belt. We take a linear subspace $R \subseteq Z_2^{l_w}$ of dimension r. Let us also consider the first round such that the difference is injected by the message block. Let these injected differences belong to R.

If the Mill function provided an ideal diffusion then the probability that the difference in any word of the mill after applying the Mill function belongs to R would be about 2^{r-l_w}. However, words 0, 3, 6, 10, 11, 14 and 18 of the mill are not affected by the message injection, so there will be zero difference in them after the first round. Thus 8 of the 12 mill words that are feedforwarded to the belt are affected by the message injection. The difference in them is not randomly distributed but one can find R such that the R-difference appears with probability 2^{r-l_w}. The inverse of the Mill function provides the diffusion close to uniform.

An example of R for RadioGatún-8 might be the following space: $R = \{b_7 b_6 \dots b_1 b_0 \mid b_7 = b_6 = b_5 = 0\}$. Let A and A' be random mills such that $A[16] \oplus A'[16]$, $A[17] \oplus A'[17]$, and $A[18] \oplus A'[18]$ belong to R. Apply the Mill function to both mills and compute the difference $\Delta A = \{\Delta_0, \Delta_1, \dots, \Delta_{18}\}$. We made 1000 experiments and observed that the probability that $\Delta_i \in R$ is close to $1/8 = 125/1000$ (see Table 1).

i	1	2	4	5	7	8	9	12	13	15	16	17
$\#\{\Delta_i \in R\}$	145	134	116	141	115	122	109	134	132	138	129	106

Table 1. Distribution of differences in the output of the Mill function.

Thus we assume that 8 words enter the Mill2Belt feedforward with the difference from R with probability[5] $2^{8(r-l_w)}$. One of these differences is added to the difference imposed by the message injection. Since any linear space is closed under addition, all 10 non-zero differences (8 from the mill and 2 from the message injection) in the belt belong to R (see also Table 2, round 1).

Now we describe how this idea can be used in attacks.

3 Collision search

In this section we show how to find a state S and the two different messages m and m' that convert S to the same state. Following the notation from [2] there exist two paths $p \neq q$ from one state to another one. This is usually called *semi-free-start collision attack* [11]. In other words, we build a collision for messages with a chosen IV. Although the IV is fixed to 0 in RadioGatún, the IV that we get in the attack can be any intermediate internal state, which makes the attack interesting.

First we describe a simplified version of the attack, and then introduce several tricks, which lead to a full attack. We apply the meet-in-the-middle approach, because the initial state can be arbitrarily chosen, there is no message schedule, and the round function is invertible. As a result, we can start with a final state and step back.

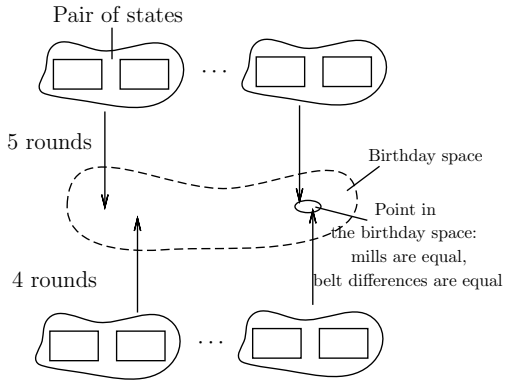

Fig. 2. Outline of the meet-in-the-middle collision search.

We start from a set of arbitrary chosen pairs of identical states. We vary injected message blocks during 5 rounds and difference in them and thus get set S_1 of pairs. 5 rounds are required to fill 38 of 39 belt words with differences. The

[5] We assume the independency of the separate events, which seems to be the case for non-trivial R and quite big (> 7) n.

sixth message injection fills the last belt word. We also start from another set of arbitrary chosen pairs of identical states and step *backwards* for 4 rounds varying message blocks and difference in them as well. As a result, we get set S_2 of pairs. If a pair belongs to both sets then we obtain a collision. The complexity of this approach is about 2^{58l_w} hash function queries. This process is briefly illustrated in Figure 2.

Round	Words with differences	
	Mill	Belt
1	1, 2, 4, 5, 7, 8, 9, 12, 13, 15, 16, 17	[1,0], [1,1], [1,2]
2	All	[1,0], [1,1], [1,2], [2,0], [2,1], [2,2], [4,2], [5,1], [7,2], [8,1], [9,0], [12,0]
3	All	[0,0], [1,0], [1,1], [1,2], [2,0], [2,1], [2,2], [3,0], [3,1], [3,2], [4,2], [5,1], [5,2], [6,0], [6,1], [7,2], [8,1], [8,2], [9,0], [9,1], [10,0], [10,2], [11,1], [12,0]
4	All	All except [0,1], [0,2], [6,0], [9,0]
5	All	All except [0,2]
6	All	All
7	All	All except [1,2], [2,1], [3,0], [4,2], [5,1], [6,0], [7,2], [8,1], [9,0], [10,2]
8	All	[0,0], [0,1] ,[0,2], [1,1], [2,0], [3,2], [4,1], [5,0], [6,2], [7,1], [8,0], [9,2], [10,1], [11,0], [12,0], [12,1], [12,2]
9	12–18	[0,0], [0,1], [0,2]

Table 2. Full trail. Words with differences after the round function is applied

Due to space limitations we can not provide here a full graphical representation of the resulting differential trail. However, it can be fully determined by the words with differences after each round. Providing that each message block has difference in all three words we derive the trail described in Table 2. The trail covers 9.5 rounds (after the 10th message injection all the words have zero difference).

In order to compare this approach with the birthday attack we introduce the notion of the *birthday space*. Assume that in order to get a collision we need to obtain two states that fit a particular relation (in the simplest case — two equal states). Then each class of equivalency is a point in the birthday space. The complexity of an attack that uses a birthday paradox is thus the square root of the size of the birthday space. If an adversary seeks an internal collision using the birthday paradox, the coincidence of all the words is not necessary. Since there is no message scheduling, and each message block can be chosen independently, it is enough to obtain two states colliding in all words not affected by the message injection and in three more words, which are the sums of words affected by the corresponding message words. Thus we have the birthday space of dimension

$55l_w$, and the complexity of the birthday attack is $2^{27.5l_w}$, which is smaller than our naive meet-in-the-middle attack, where the dimension of the birthday space is $2 * 19l_w + 2 * 39l_w = 116l_w$.

Next we show that pairs may not completely coincide. First we relax restrictions on the belt and show that the equality of the difference in the belt is enough.

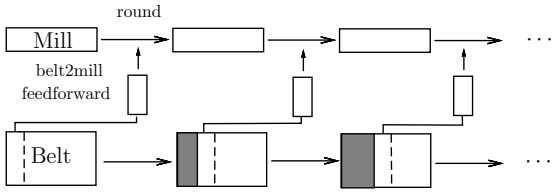

Fig. 3. Scheme of the belt recovery.

Proposition 1. *The belt-to-mill feedforward in $n \leq 13$ consecutive rounds can be considered as injection of n independently chosen 3-word blocks.*

Proof. Indeed, one can recover the belt from any $n \leq 13$ 3-word blocks without contradiction. This can be proved by the following observation. Let us derive the values of the belt words consecutively while iterating one round after another. At each step we derive 3 more known words and use the known message and mill values to carry out the known values to the next step. Due to slow rotation new values cover consecutive columns in the belt. This process is briefly illustrated in Figure 3.

Formally, denote the belt in the beginning of the trail by B and in the end by B'. It is easy to see that belt words are not mixed with each other, only message and mill words are added. Thus $B'[i,j] = B[i-n,j] + f(i,j)$ where n is the trail length and f is a function of the message and the mills. The belt words that are feedforwarded to the mill are derived from distinct B words. Thus giving any set of feedforward blocks, the mill values and message blocks one can recover the original B without contradiction.

Let us return to the 9-round trail. If a state from S_1 and a state from S_2 coincide in the mill and in the difference in the belt then the corresponding parts of the trail can be combined into one trail. At the same time, 27 words of the initial belt are recovered. The other 12 words can be assigned randomly. The dimension of the birthday space is $2 * 19l_w + 39l_w = 77l_w$ so the complexity of the second version of the attack is about $2^{38.5l_w}$.

Linear truncated differentials. In order to reduce the birthday space we impose restrictions on the differences that are fed to the belt. We choose integer $r \leq$

l_w (the exact value of r will be defined later) and a linear space $R \subset Z_2^{l_w}$ of dimension r that fits the assumptions of uniformity (see section 2.2). In order to obtain a desired difference we vary the injected messages. We choose the first message block in the pair randomly thus having $3l_w$ *degrees of freedom* (see also [12]). The second message should have the R-difference with the first one so we have $3r$ more degrees of freedom. Thus the probability that we find the words to be injected such that a given pair pass through the next round with R-restriction is $2^{8r-8l_w+3r+3l_w} = 2^{11r-5l_w}$ (if this value exceeds 1 then we just obtain more pairs).

The latter value can be also considered as a *multiplier* c such that if N pairs enter the round then $c \cdot N$ pairs with R-difference can be obtained from them after one iteration.

We need 5 rounds and one more message injection to fill the 39 words of the belt (we use the trail presented in Table 2) with R-differences. Let (A, B) denote the internal state as a pair of the mill and the belt in the beginning of sixth round. We also require that the 12 words of A that are feeded to the belt in the sixth round should also have R-difference (this is arranged by the message injection in rounds 4 and 5). To sum up, we need 56 R-difference words in the mills during five rounds while the freedom provided by injections is $5*(3r+3l_w) = 15r+15l_w$. Additionally, we randomly choose the words that are feedforwarded from the belt (see Proposition 1) in rounds 1-4 thus having $12l_w$ more degrees of freedom. As a result, if we start with 2^{n_1} pairs then $2^{n_1+15r+27l_w+56r-56l_w} = 2^{n_1+71r-29l_w}$ pairs pass through five rounds.

Now we consider the second part of the trail and proceed back from the aero-difference state. Only 3 message injections are needed to fill the belt with R-differences. However, the difference in the mill would coincide with the difference in the 12 words of the belt. We add one more round. Thus 48 words with R-difference should be obtained during the process. The multiplier is

$$2^{4*(12r-12l_w+3r+3l_w+3l_w)} = 2^{60r-24l_w}.$$

Finally, let us calculate the dimension of the birthday space. Recall that we need that pairs should coincide in the value of the mill, in the difference of the mill, and in the difference of the belt. The dimension of the resulting birthday space is $19l_w + (12r + 7l_w) + 39r = 51r + 26l_w$. However, we have not used the freedom that is provided by the message injection in round 6 and the belt-to-mill feedforward in round 5 yet. This freedom allows us to further relax the restriction on the coincidence of pairs: we do not care of values of the mill in 6 words and of 3 word differences. Finally, the resulting birthday space is of dimension $20l_w + 48r$.

Now we compute r such that the number of pairs throughout the attack is minimal. Let us denote by 2^{n_1} and 2^{n_2} the number of pairs that we start with from the first round and from the last round, respectively. Then the number of pairs and the complexity of the attack[6] is bounded by $\max(2^{n_1}, 2^{n_1+71r-29l_w},$

[6] We assume that the search for appropriate message blocks and belt words is of negligible cost and can be maintained with a lookup table.

2^{n_2}, $2^{n_2+60r-24l_w}$). The second requirement is that the number of pairs in the middle round should be enough to perform the birthday attack: $(n_1 + 71r - 29l_w) + (n_2 + 60r - 24l_w) = 20l_w + 48r$. The best solution is provided by the r equal to $0.4l_w$. This implies the equation $n_1 + n_2 = 39.8l_w$. The resulting complexity is $2^{19.9l_w}$.

Relaxation. Further we note that several words in the belt are updated by the mill words twice during the first 5 rounds. Since we need R-difference only in the middle state, arbitrary difference can be injected at the first time and later converted to the R-difference. As before, we expect the probability of getting an R-difference as 2^{r-l_w}. After this *relaxation* we have no restrictions on the difference in the message injection in the first round (the idea is illustrated in Figure 4). Furthermore, we have no restrictions on the mill difference in the first round. The only difference that should be maintained by the first injection is the difference in 3 mill words after round 2.

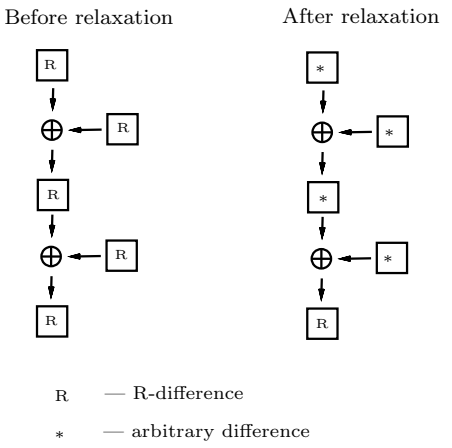

Fig. 4. Idea of the relaxation.

Following this approach we obtain probability 2^{48r-6l_w} for a random pair to come out of the first part of the trail. The probability for the second part of the trail (reverse process) is $2^{48r-12l_w}$. However, the number of pairs is no longer a monotonic function of the round number, so we adjust the value of r in order to keep the number of considered pairs minimal during the attack. The resulting complexity is about 2^{18l_w} hash function queries with $r = 4/13l_w$ and the birthday space of dimension $48r + 20l_w \approx 34l_w$. The number of pairs after every round is given in Table 3.

Strengthening. The fact that the number of pairs is not a monotonic function of the round number means that degrees of freedom are not properly used. Here we notice that after relaxation most words with R-differences are not added to each other so we can omit the restriction on linearity. One may consider a *group of differences* (instead of a linear space) of arbitrary size between 1 and 2^{l_w}.

In order to flatten the function of the number of pairs we consider particular words in the mill and *strengthen* the restriction on differences in them taking another space R for a particular word. As a result, we deal with several different R's, each with its own size.

The benefit is given as follows. Suppose we work with a two-round trail. The number of pairs is N before the first round, $2^l N$ ($l > 0$) before the second round,

Round	Degrees of freedom	Words to control	Number of pairs ($\log_2$)
0	-	-	$12.5l_w$
1	$6l_w$	3	$14.5l_w$
2	$9l_w$	10	$16.6l_w$
3	$9l_w$	11	$18l_w$
4	$9l_w$	13	$18l_w$
5	$9l_w$	13	$18l_w$

Round	Deg. of freedom	Words to control	N-r of pairs
10	-	-	$13.8l_w$
9	$9l_w$	10	$15.9l_w$
8	$9l_w$	10	$18l_w$
7	$9l_w$	13	$18l_w$
6	$9l_w$	15	$16l_w$

Table 3. The complexity of the collision search after the relaxation ($r = 4/13l_w$).

and N after the second round. Then the the overall complexity is bigger than both the initial and end values and is equal to $2^l N$. If we follow the idea of strengthening and add l more conditions on the difference after the first round (and in the end) then the number of pairs is reduced to N after the first round and to $\frac{N}{2^l}$ in the middle. The dimension of the middle space is also decreased by l. In order to maintain the birthday attack we must increase the initial number of pairs from N to $N2^{l/2}$. The complexity of the attack is thus reduced to $N2^{l/2}$.

Theoretical lower bound. One may ask the question what the smallest complexity is that we can achieve following the ideas of linear differences, relaxation and strengthening. Let us recall that the dimension of the middle space without restrictions on differences is $77l_w$. If we impose P linear restrictions on differences then the dimension will be $77l_w - P$. On the other hand, we have $51l_w$ degrees of freedom (provided by 6 message blocks and belt2mill feedforward blocks) to compensate the restrictions. Thus the multiplier of the first part of the trail is $2^{51l_w - P}$. The lowest complexity is achieved if the multiplier is equal to 1, so we obtain $P = 51l_w$ and the dimension of the middle space is $26l_w$. The number of pairs required by the birthday attack is 2^{13l_w} which is the lower bound.

4 Second preimage search

The idea of the second-preimage attack is similar to a simple collision one. While we looked for collisions with arbitrary pairs, in the second-preimage attack the first element of every pair is fixed and is equal to the original internal state. We pull a number of states through the iteration process from both ends and look for the coincidence in the middle round. We vary injected messages in order to obtain R-differences in the middle round. The trail is similar to that is given in Table 2, but the zero differences are now arbitrary differences.

Let us consider 10 rounds of the hash iteration to which we want to find a second preimage (called below the *original iteration*). Where these rounds should be located will be discussed later. Denote the internal states of the original iteration in the beginning of 11 consecutive rounds: $I_0, I_1, \ldots, I_{10}$. Suppose we also have N_1 states (the exact value will be also defined later) that are resulted from

iteration of the original zero state with some random message. Then we consider N_1 differences between these states and the state I_0 as the first difference in the 10-round trail, which is obtained from the trail in Table 2 by adding one more round in the beginning and replacing zero differences with arbitrary differences. Next for each of N_1 states we look for the 6-block messages that provide an R-difference state in the middle round (a state that has an R-difference in every word with the state I_6). As a result, we obtain a set S_1 of internal states. See also Figure 5 as an illustration.

Similarly, suppose we have N_2 states that are resulted from reverse iteration of the last internal state of the original iteration. We treat them in the similar way and look for the 4-block messages that provide an R-difference state in the middle round. Thus we obtain a set S_2 of internal states. Then we look for a state that is presented in both sets. Such a state implies a parallel iteration, which gives the same hash value.

Now let us estimate what are N_1, N_2 and the complexity of the attack. The injection in round 0 controls 4 mill words in the end of round 1 such that the resulting difference belong to R. The injections in rounds 1-4 control 12 words and the last one control 8 words. Thus the probability that a state can be pulled to the middle state with R-differences is $2^{(4+12*4+8)(r-l_w)+3*6r} = 2^{78r-60l_w}$. The same idea holds for reverse steps. We start with N_2 states and the proportion $2^{4*(12r-12l_w+3r)} = 2^{60r-48l_w}$ of them comes out of the iteration.

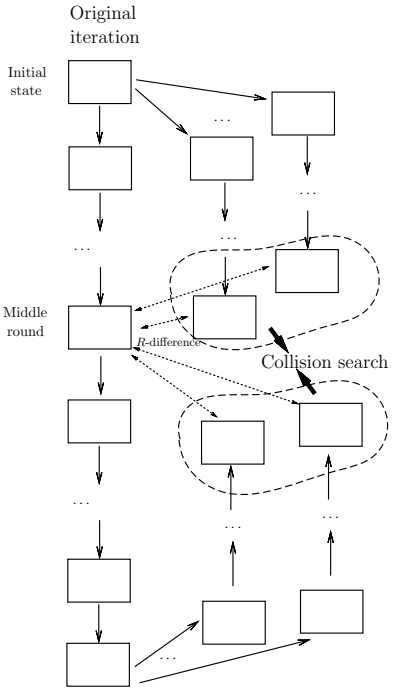

Fig. 5. Outline of the second preimage search.

The dimension of the birthday space in the beginning of 6-th round is $7l_w + 51r$ (12 words of the mill and all the words of the belt must have R-difference, and the other 7 mill words may have arbitrary difference). Given $3r$ more degrees of freedom from the message injection in round 6 we derive that $2^{3.5l_w+24r}$ internal states are required to perform the birthday attack.

The optimal complexity is given by the $r = 0.8l_w$ that converts the multiplier $2^{60r-48l_w}$ to one. Thus we derive

$$N_1 = 2^{3.5l_w+24r+60l_w-78r} = 2^{20.3l_w}; \quad N_2 = 2^{3.5l_w+24r+48l_w-60r} = 2^{22.7l_w}.$$

If we follow the method of relaxation and strengthening as described in Section 3 then the complexity about 2^{20l_w} could be achieved. This is actually the lower bound for these meet-in the middle attacks with 10-round trails, which can be checked following the method in Section 3.

5 Implementation of attacks

Though optimal r might be non-integer, we can take concrete values just to check whether our approach works in real life. Due to high complexity of the attack even with small number bits in a word we can not perform the attack as a whole but we tested the RadioGatún round function on different spaces R and encountered good distribution of differences in the output (see Table 1), especially in reverse steps. We also checked that values in the belt words and message blocks to be injected can be chosen such that the desired differences appear in the output of the non-linear function. Thus we substantiated the main assumptions made throughout the description of the attack.

One may also argue that RadioGatún with reduced number of *words* in the belt and in the mill may be considered as an easier object for the attack. However, the reduced round function and its inverse do not provide good differential characteristics (close to random) anymore. We checked this for the internal state that is reduced threefold. This (non-uniformity) is also the case with small l_w (the number of bits in a word), which makes our attack inefficient.

We also note that the attack becomes trivial for RadioGatún-1 since there are only two options for R, and both of them give high complexity.

6 Conclusions

We investigated the security of RadioGatún using differential trails with linear restrictions on differences. We applied the meet-in-the-middle approach and managed to reduce the complexity with help of new tricks such as relaxation and strengthening. We showed how to find semi-free-start collisions with complexity about 2^{18l_w} hash function calls and the second preimage with about $2^{22.7l_w}$ calls (with a possible improvement up to 2^{20l_w}). We also provided theoretical lower bounds on the complexity of the attack which follow the same approach.

The main weakness of the RadioGatún round function that we exploited is plenty of linear operations and slow diffusion in the belt. We suppose that a compromiss between adding more non-linearity in the primitive transformations and the speed might be found so the design could be seriously strengthened and the security level could be increased (say, up to 2^{16l_w}). As a result, a smaller versions (in terms of l_w) could be used as a 256/384/512-bit hash function.

Regarding RadioGatún itself, though our attacks do not break the claimed security level ($2^{9.5l_w}$), they are faster than the birthday attack and the attack that might be carried out from Grindahl [12]. Thus we conclude that RadioGatún is still resistant against differential-based collision search though this resistance is now provided only by a substantially low security level.

Acknowledgements

The author greatly thanks Alex Biryukov, Ivica Nikolic, Stefan Lucks, Joan Daemen and the RadioGatún team, and the anonymous reviewers for their valuable and helpful comments. The author is supported by PRP "Security & Trust" grant of the University of Luxembourg.

References

1. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Radiogatun, a belt-and-mill hash function. *NIST Cryptographic Hash Workshop*, 2006.
2. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Sponge functions. *ECRYPT Hash Workshop*, 2007, available at http://sponge.noekeon.org/.
3. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. On the indifferentiability of the sponge construction. In *EUROCRYPT*, LNCS, pages 181–197. Springer, 2008.
4. Joan Daemen. *Cipher and hash function design strategies based on linear and differential cryptanalysis.* PhD thesis, K.U.Leuven, March 1995.
5. Joan Daemen and Gilles Van Assche. Producing collisions for Panama, instantaneously. In *FSE'07*, volume 4593 of *LNCS*, pages 1–18. Springer, 2007.
6. Joan Daemen and Craig S. K. Clapp. Fast hashing and stream encryption with PANAMA. In *FSE'98*, volume 1372 of *LNCS*, pages 60–74. Springer, 1998.
7. Joan Daemen and Vincent Rijmen. *The Design of Rijndael. AES — the Advanced Encryption Standard.* Springer, 2002.
8. http://csrc.nist.gov/groups/ST/hash/index.html. *Cryptographic Hash Project.*
9. http://radiogatun.noekeon.org/.
10. Lars R. Knudsen, Christian Rechberger, and Søren S. Thomsen. The Grindahl hash functions. In *FSE'07*, volume 4593 of *LNCS*, pages 39–57. Springer, 2007.
11. Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.
12. Thomas Peyrin. Cryptanalysis of Grindahl. In *ASIACRYPT'07*, volume 4833 of *LNCS*, pages 551–567. Springer, 2007.

Attack	Type	Complexity	Origin
Collisions	Symmetric trails	2^{46l_w}	[1]
	Birthday	$2^{27.5l_w}$	-
	R-difference	$2^{19.9l_w}$	This paper
	After relaxation	2^{18l_w}	This paper
Second preimage search	Birthday	$2^{27.5l_w}$	-
	R-differences	$2^{22.7l_w}$	This paper
	After relaxation	$\sim 2^{20l_w}$	This paper*

* – hypothetical.

Table 4. Summary of attacks on RadioGatún.