Note on zero-sum distinguishers of Keccak-f

In [1], Jean-Philippe Aumasson and Willi Meier introduced *zero-sum distinguishers*, a method to generate zero-sum structures for reduced-round Keccak-f[1600], the permutation underlying our SHA-3 submission Keccak. Their paper contained distinguishers for up to 16 rounds of Keccak-f[1600]. Recently, Christina Boura and Anne Canteaut extended this to 18 rounds in [6]. In this note we argue that the distinguishers are valid and are qualitatively different from generic methods, as they can partition the set of inputs into sets of zero-sum structures of specific sizes. We also put this in perspective, as generic methods allow generating zero-sum structures of small sizes, and the distinguishers covering more rounds have extremely high complexity (e.g., 2^{1369} for 18 rounds).

Nevertheless, after the publication of [1], we decided to increase the number of rounds of Keccak-f. The logic underlying this decision is our adoption of the *hermetic sponge strategy*, in which we tolerate no structural distinguisher for the permutation used in the sponge construction. The strength or applicability of the distinguisher in the context of the sponge construction plays no role in this aspect.

By increasing the number of rounds, we believe to have re-established the security margin of Keccak-f with respect to structural distinguishers.

1 The challenge

A zero sum structure for a function f is defined in [1] as a set $\mathcal{Z}$ of inputs z_i that sum to zero, and for which the corresponding outputs also sum to zero. The challenge is now to generate such a set in an efficient way. Hence the challenge is the following.

Challenge: given a function f from n to m bits and an integer N, construct a set $\mathcal{Z}$ of N inputs z_i (or the set of corresponding f-images) such that:

$$\sum_{0 \le i < N} z_i = 0 \text{ and } \sum_{0 \le i < N} f(z_i) = 0.$$

Given all inputs in $\mathcal{Z}$ except one and the f-outputs of all inputs of $\mathcal{Z}$ but one, the zero-sum structure allows the computation of the missing input and output by simply summing over the known elements and hence without calling f. If the size of $\mathcal{Z}$ is small, this may give an adversary an advantage in an attack. Clearly, the advantage diminishes as the size of $\mathcal{Z}$ grows.

2 A generic method

In this section we present a generic method for constructing a zero-sum structure inspired by Wagner's algorithm for the generalized birthday problem in [7] and by the attack against XHASH in [2], brought to our attention by Jean-Philippe Aumasson. For a method to become a structural distinguisher for a particular function, it shall have a lower complexity than this generic method.

Here is an outline of the method. We use the following notation: $X_i = [x_i | f(x_i)]^{\mathrm{T}}$, i.e., a column vector with components the bits of x_i followed by the bits of $f(x_i)$.

1. Take N random values x_i, compute $f(x_i)$ and form $X_i = [x_i | f(x_i)]^{\mathrm{T}}$.
2. Compute the bitwise sum of the vectors X_i and call the sum A:

$$\sum_{0 \le i < N} X_i = A\,. \tag{1}$$

3. Take $p = n + m + \epsilon$ random values y_i with $0 \leq i < p$ and ϵ a small integer, compute $f(y_i)$ and form $Y_i = [y_i | f(y_i)]^{\mathrm{T}}$.

4. Solve the following linear system of $n + m$ equations in the $n + m + \epsilon$ variables a_i over GF(2), with the bits of $(X_i \oplus Y_i)$ serving as (fixed) coefficients:

$$\sum_{0 \leq i < p} a_i (X_i \oplus Y_i) = A. \tag{2}$$

5. For a solution (a_i), form the set $\mathcal{Z}$ such that

$$z_i = \begin{cases} y_i & \text{if } i < p \text{ and } a_i = 1 \\ x_i & \text{otherwise.} \end{cases}$$

Clearly, the set $\mathcal{Z}$ is a zero-sum structure. Adding equations (1) and (2) gives:

$$\sum_{0 \leq i < N} X_i \oplus \sum_{0 \leq i < p} a_i (X_i \oplus Y_i) = \sum_{0 \leq i < p} (a_i Y_i \oplus \bar{a}_i X_i) \oplus \sum_{p \leq i < N} X_i = \sum_{0 \leq i < N} Z_i = 0.$$

This method requires that $p = n + m + \epsilon \leq N$ for some $\epsilon \geq 0$. The value of ϵ determines the a priori probability that the system of equations (2) has a solution: by increasing ϵ the probability that it has no solution decreases exponentially. If $N \gg n + m$, the probability of failure can be made arbitrarily small by increasing ϵ and the complexity can be approximated by N executions of f.

The computational effort is the sum of:

- $N + n + m + \epsilon$ evaluations of f,
- solving a system of $n + m$ linear equations in $n + m + \epsilon$ variables over GF(2), which can be done very efficiently, and
- taking the bitwise sum of N $(n + m)$-bit vectors.

3 The zero-sum distinguishers on KECCAK-f

The method for constructing zero-sum structures described in [1, 6] exploits the fact that adding a round in KECCAK-f only doubles the degree of the algebraic expression of the output bits in terms of the input bits, and only triples the degree of the algebraic expression of the input bits in terms of the output bits.

We discuss here the aspects that are relevant for the computational complexity of constructing the distinguishers and refer to [1, 6] for the details. We consider the complexity of the method constructing the set $\mathcal{Z}$ or the set of corresponding outputs.

Compared to the generic method, the method in [1, 6] has the following features.

- First, as opposed to the generic method, this method is deterministic.
- Second, in this method the size of $\mathcal{Z}$ cannot be freely chosen (above some minimum) but is limited to powers of two (above some minimum).
- Third, the non-maximal degree of the two parts of the (reduced-round) KECCAK-f permutation can be used to create partitions of inputs in many different zero-sum structures. The size of such partitions, using this method, is a multiple of the size of the individual zero-sum structures. Producing a single zero-sum structure still leads to the fastest distinguisher in this context.

Rounds	inv. + forw.	N	Rounds	inv. + forw.	N
6	$2+4$	2^{10}	12	$5+7$	2^{129}
7	$3+4$	2^{15}	13	$6+7$	2^{244}
8	$3+5$	2^{18}	14	$6+8$	2^{257}
9	$4+5$	2^{30}	15	$6+9$	2^{513}
10	$4+6$	2^{60}	16	$6+10$	2^{1025}
11	$5+6$	2^{60}	18	$7+11$	2^{1370}

Table 1: Size of zero-sum structures for reduced-round KECCAK-$f[1600]$ given in [1, 6]

For constructing $\mathcal{Z}$, one takes N states y_i (that forms a vector space of limited dimension) in some intermediate round and computes a number of rounds backwards to obtain the inputs z_i. Clearly, the complexity of this option (*the backward option*) is hence N times the computation of these inverse rounds. For constructing the outputs corresponding to $\mathcal{Z}$ (*the forward option*), one must compute a number of rounds forwards and the complexity is N times this forward computation. Table 1 lists the values of N for reduced-round versions of KECCAK-$f[1600]$ for given number of rounds. It also gives the number of inverse rounds and forward rounds.

As seen in Table 1, for all cases the number of inverse rounds is smaller than the number of forward rounds. Hence at first sight the backward option seems to be the most efficient one. However, mainly due to the complexity of the inverse of θ [5], the computation of the inverse round of KECCAK-$f[1600]$ has much higher complexity than the round itself. We think it is safe to assume that the inverse round takes twice as many computations as the forward round. In this light the forward option becomes the most efficient one. As the number of forward rounds is greater than half the number of rounds, the complexity of the method can be expressed as the computational equivalent of at least $N/2$ calls to the function under attack.

4 Implications for KECCAK-f

For the values of N given in Table 1 (and any larger power of two), the method for generating zero-sum structures of [1, 6] is more efficient than the generic method by a factor 2. Hence, the zero-sum distinguishers of [1, 6] are valid, albeit with a very small advantage. For instance, consider the case of KECCAK-$f[1600]$ reduced to 18 rounds. The method of [6] for the smallest value of N would have complexity 2^{1369} while for the generic method this is 2^{1370}. Note however that the generic method additionally allows generating zero-sum structures with any size $N > 3200$ at the cost of about $N + 3200$ calls to the function under attack.

We think it is very unlikely that the zero-sum distinguishers can result in the speedup of actual attacks against KECCAK calling reduced-round versions of KECCAK-f. Still, the distinguishers described in [1, 6] show non-ideal properties of the (reduced-round) KECCAK-f permutation and suggested us to increase the number of rounds in KECCAK-f [4].

The main reason behind this is our adoption of the hermetic sponge strategy [3]. This strategy imposes KECCAK-f to be free from structural distinguishers, without considering their strength or relevance for the KECCAK sponge function.

The existence of the distinguisher in [1] over 16 (out of 18) rounds of KECCAK-$f[1600]$ left only a security margin of 2 rounds. Moreover, we wanted to increase the security margin against other possible distinguishers that start from the middle and compute back- and forwards to get the corresponding in- and outputs. In this method adding two rounds to KECCAK-f only increases the algebraic degree to be considered in the attack by a factor 3. This is due to the fact that a KECCAK-f round has degree 2 and its

inverse only 3 [5]. We estimated that other types of distinguishers may be found that also exploit this fact or that the distinguishers may be further refined (e.g., as done in [6]). Therefore we decided to address this in round 2 of the SHA-3 competition by increasing the number of rounds (e.g., for KECCAK-$f[1600]$ from 18 to 24 rounds).

The KECCAK Team, January 2010
Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche

References

[1] J.-P. Aumasson and W. Meier, *Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi*, Available online, 2009, `http://131002.net/data/papers/AM09.pdf`.

[2] M. Bellare and D. Micciancio, *A new paradigm for collision-free hashing: Incrementality at reduced cost*, Eurocrypt, 1997, pp. 163–192.

[3] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, *Cryptographic sponges*, 2009, `http://sponge.noekeon.org/`.

[4] _____, KECCAK *specifications, version 2*, NIST SHA-3 Submission, September 2009, `http://keccak.noekeon.org/`.

[5] _____, KECCAK *sponge function family main document*, NIST SHA-3 Submission (updated), September 2009, `http://keccak.noekeon.org/`.

[6] C. Boura and A. Canteaut, *A zero-sum property for the Keccak-f permutation with 18 rounds*, Available online, 2010, `http://www-roc.inria.fr/secret/Anne.Canteaut/Publications/zero_sum.pdf`.

[7] D. Wagner, *A generalized birthday problem*, CRYPTO (M. Yung, ed.), Lecture Notes in Computer Science, vol. 2442, Springer, 2002, pp. 288–303.