Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi

Jean-Philippe Aumasson and Willi Meier

FHNW, Windisch, Switzerland

Abstract. We present a new type of distinguisher, called zero-sum distinguisher, and apply it to reduced versions of the Keccak-f permutation. We obtain practical and deterministic distinguishers on up to 9 rounds, and shortcut distinguishers on up to 16 rounds, out of 18 in total. These observations do not seem to affect the security of Keccak. We also briefly describe application of zero-sum distinguishers to the core permutations of Luffa and Hamsi.

The Keccak-f permutation operates on a 1600-bit state; it makes 18 rounds, and each round has algebraic degree 2, with respect to GF(2). Hence the n-round permutation has degree at most 2^n. The inverse permutation, however, has degree 3 (cf. [1, §5.9.3.1]).

Suppose one fixes $1600 - 513 = 1087$ bits of the initial state to some arbitrary value, and consider the 2^{513} states obtained by varying the 513 bits left. Our main observation is that applying the 9-round Keccak-f permutation to each of those states and xoring the 2^{513} 1600-bit final states obtained yields the zero state. This is because, for each of the 1600 Boolean components, the value obtained is the order-513 derivative of a degree-512 polynomial, which by definition is null. We call exhibition of such sets of values *zero-sum distinguishers*, and we call the sets of values *zero-sums*. More generally, a zero-sum distinguisher for a function is any method to find a set of values summing to zero such that their respective images also sum to zero.

Observing that the 5-round inverse permutation has degree at most $3^5 = 243$, one can do the following for the 14-round Keccak-f permutation:

1. Fix $1600 - 513 = 1087$ bits of the intermediate state after 5 rounds to some arbitrary value.
2. For each of the 2^{513} values of the bit left, compute 5 rounds backwards to obtain 2^{513} initial states.

One thus obtains 2^{513} distinct initial states that have the following remarkable properties:

- They sum (xor) to zero, since the sum is the order-513 derivative of a degree-243 mapping.
- Their images by the Keccak-f permutation sum to zero as well, because the mapping defined by 9 rounds of the Keccak-f permutation has degree at most $2^9 = 512$.

This method is *deterministic*, and costs the equivalent of less than 2^{512} evaluations of the Keccak-f permutations (i.e., 2^{513} times 5 inverse rounds) to compute values satisfying the zero-sum property.

One may do even better by exploiting the fact that the degree increases more slowly than expected. Indeed, Table 5.4 in [1] reports that the 5-round permutation has degree at most 17, and that the 3-round inverse permutation has degree 17 as well, with respect to variables in slice $z = 0$. Making the (worst-case) assumption that after 5 (resp., 3) rounds, the degree is multiplied by 2 (resp., 3) at every round, we have that:

- After 9 rounds, the permutation has degree ≤ 272, and after 10 rounds is has degree ≤ 544.
- After 5 rounds, the inverse has degree 153, and after 6 rounds it has degree ≤ 459.

But the bounds 272 and 153 are incorrect here; this is because bounds in Table 5.4 hold for bits in a same slice (a slice contains 25 bits), and when more than 25 degrees of freedom are needed, one needs to use bits in distinct slices, and so the bounds of Table 5.4 do not apply anymore (idem for bits in a same 64-bit lane from Table 5.5).

Therefore, the above trick to obtain lower upper bounds on the degree works only with at most 25 variables (resp., at most 64), when considering bounds from Table 5.4 (resp. 5.5). For degrees above those bounds, we shall make the (worst-case) assumption that the degree is 2^n (resp., 3^n) after n rounds forwards (resp., backwards).

Another trick to reduce the maximal degree (and thus the complexity of our distinguishers) is to ensure that the first round backwards has degree one, by avoiding having more than one variable in each row. There are $64 \times 5 = 320$ rows, hence this trick works when 320 or less variables are necessary. Similarly, when more than 320 but less than 640 variables are necessary, one can ensure that the first inverse round has degree one by setting at most two variable bits in each row, such that they are not adjacent.

Table 1 summarizes the complexity of our distinguishers for various parameter choices.

Table 1. Parameters of the best distinguisher for various total number of rounds. The columns "type of bounds" gives the type of bounds used, either with respect to bits in a same slice (Table 5.4, only if order ≤ 25), in one lane (Table 5.5, only if order ≤ 25), at most two per row ($3^{n-1}, 2^n$, only with order ≤ 640), or anywhere in the state ($3^n, 2^n$). For consistency, we give the normalized complexity in terms of evaluations of the permutation (assuming that computing a round has the same complexity as computing an inverse round), e.g., the complexity given is $2^{10} \times 2/6 = 2^{8.42}$ for the attack on the first line, since it requires 2^{10} evaluations of the two rounds of the inverse permutations, out of six rounds in total in the permutation considered.

type of	backwards		forwards		total	
bounds	#rounds	degree $\leq$	#rounds	degree $\leq$	#rounds	complexity
1 slice	2	9	4	9	6	$2^{8.41}$
1 lane	3	9	3	8	6	$2^{9.00}$
1 lane	3	9	4	15	7	$2^{14.77}$
1 slice	3	17	5	17	8	$2^{16.58}$
1 lane	4	27	5	30	9	$2^{29.83}$
1 lane	4	27	6	60	10	$2^{59.67}$
1-per-row	5	81	6	60	11	$2^{59.54}$
1-per-row	5	81	7	128	12	$2^{127.73}$
1-per-row	6	243	7	128	13	$2^{242.88}$
1-per-row	6	243	8	256	14	$2^{255.77}$
2-per-row	6	243	9	512	15	$2^{511.68}$
anywhere	6	729	10	1024	16	$2^{1023.88}$

We experimentally verified results in Table 1, for practical complexities, by finding zero-sums on 6, 7, and 8 rounds of the Keccak-f permutation. We also observed that the bounds on the degree used are not always tight. For example, the distinguisher on 7 rounds exploiting maximal degree 15 worked even when making certain order-10 derivatives (that is, making 2^{10} evaluations backwards instead of 2^{16}). For the distinguishers exploiting maximal degree 9, however, this degree was reached and order-10 derivatives were necessary.

The Keccak-f is the main component of the Keccak hash function submitted to the SHA-3 competition. Our observations do not seem to affect the security of Keccak. To the best of our knowledge, no deterministic and generic method is known to compute zero-sums. Our zero-sum distinguishers may be viewed as a measure on how many rounds are necessary for optimum security guarantees. This suggests that for Keccak-f 17 rounds would be necessary.

Table 2. Structure of the 10-round distinguisher.

4 rounds degree ≤ 27	6 rounds degree ≤ 60
$\longleftarrow$	$\longrightarrow$

Table 3. Structure of the 16-round distinguisher.

6 rounds degree ≤ 729	10 rounds degree ≤ 1024
$\longleftarrow$	$\longrightarrow$

Generally, zero-sum distinguishers are relevant for any n-bit permutation for which one can find some set of $m < n$ variables in some intermediate state such that in both directions all bits of the output are functions of degree strictly less than n in the variables chosen. Typical targets are algorithms with only bitwise logical operations, while those making modular additions (like so-called ARX algorithms) are unlikely to be vulnerable to zero-sum distinguishers.

For instance, one can apply zero-sum distinguishers to the permutation Q of Luffa [2, 3]: each of its eight rounds has degree three, both forwards and backwards, hence 4 rounds have degree at most $3^4 = 81$. This allows one to find zero-sums by computing order-82 derivatives with variables in the intermediate state after 4 rounds. This does not seem to affect the security of Luffa, and in fact its designers already noted that "8 step functions cannot be considered a perfect random permutation" [3, §3.1.2], based on the existence of a differential path with probability 2^{-224}.

The permutation P_f of Hamsi [4] is also vulnerable to zero-sum distinguishers: for the 256-bit version of Hamsi, P_f works on a 512-bit state, a round has degree 3 (both forward and backward), and it makes 6 rounds for the finalization. A zero-sum distinguisher on 2×3 thus makes about 2^{27} evaluations of the permutation. For the 512-bit version, P_f works on a 1024-bit state makes 12 rounds, which allows one to find zero-sums in 2^{729}. These distinguishers, however, are not relevant for the compression function of Hamsi, because of the redundancy in its input. A fortiori, the security of Hamsi seems unaffected.

A detailed exposition of our results will appear in an extended note.

Acknowledgments

We are grateful to the designers of Hamsi, Keccak, and Luffa for the verification of our results, and for their comments to improve this note.

References

1. Guido Bertoni, Joan Daemen, Michaeël Peeters, and Gilles Van Assche. Keccak sponge function family main document. Submission to NIST (updated), 2009. Version 1.2.
2. Christophe De Cannière, Hisayoshi Sato, and Dai Watanabe. Hash function Luffa: Specification. Submission to NIST, 2008.
3. Christophe De Cannière, Hisayoshi Sato, and Dai Watanabe. Hash function Luffa: Supporting document. Submission to NIST, 2008.
4. Özgül Kücük. The hash function Hamsi. Submission to NIST, 2008.