ICT-2007-216676

ECRYPT II

European Network of Excellence in Cryptology II

Network of Excellence

Information and Communication Technologies

D.SYM.11
Final Hash Functions Status Report

Due date of deliverable: 31. January 2013
Actual submission date: 29. January 2013

Start date of project: 1 August 2008 Duration: 4 years

Lead contractor: Katholieke Universiteit Leuven (KUL)

Revision 1.0

Project co-funded by the European Commission within the 7th Framework Programme		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission services)	
RE	Restricted to a group specified by the consortium (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	

Final Hash Functions Status Report

Editor
Christian Rechberger (DTU)

Contributors
Christina Boura (DTU)
Bart Mennink (KUL)
Maria Naya-Plasencia (INRIA)

29. January 2013
Revision 1.0

The work described in this report has in part been supported by the Commission of the European Communities through the ICT program under contract ICT-2007-216676. The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Contents

Executive Summary

Breakthrough cryptanalytic results on popular hash functions like MD5 and SHA-1 in 2004 and 2005 by Wang et. al motivated standardization organizations, industry, and academic research groups alike to look into alternative hash function constructions. Now, some 8 years later, and after intense work on the design, cryptanalysis, proofs, benchmarking of hash functions, the NIST SHA-3 competition selected the hash function proposal Keccak as a new standard.

In this report we provide a state of the art survey of recent developments around Keccak in the areas of cryptanalysis and security proofs. This is our final report and it is a continuation of the previous deliverables D.SYM.4 and D.SYM.7.

Chapter 1

Introduction

Keccak, designed by Guido Bertoni and Joan Daemen and Michaël Peeters and Gilles Van Assche, was by the end of 2012 selected as the winner of the SHA-3 competition, which was formally started in 2008 with a call for submissions [NIS]. This helped form a vast body of new literature on many aspects of hash function design and analysis. The aim of this report is to provide a summary of all those developments that are relevant to Keccak.

After describing the design in Section 1.1, in chapter 2 we discuss security arguments that help to get confidence in the design. This includes reductionist security arguments, i.e. to what extent the security of the hash function can be reduced to properties of underlying building blocks. This also includes bounds on classical attack vectors like differential attacks.

In chapter 3 we will discuss various attempts to cryptanalyze Keccak and reduced versions of it, hence greatly extending the scope of the security analysis of the earlier chapter. This includes zero-sum, algebraic, rebound, and rotational attack vectors. Finally, in chapter 4, we conclude with open problems for research on Keccak.

1.1 Specifications summary

This section offers a summary of the Keccak specifications using pseudocode, sufficient to understand its structure and building blocks[1]. In no way should this introductory text be considered as a formal and reference description of Keccak. For the formal definition of Keccak, we refer to [BDPV12]. Any instance of the Keccak sponge function family makes use of one of the seven Keccak-f permutations, denoted Keccak-$f[b]$, where $b \in \{25, 50, 100, 200, 400, 800, 1600\}$ is the width of the permutation. These Keccak-f permutations are iterated constructions consisting of a sequence of almost identical rounds. The number of rounds ” depends on the permutation width, and is given by $n_\mathrm{r} = 12 + 2\ell$, where $2^\ell = b/25$. This gives 24 rounds for Keccak-$f[1600]$.

```
Keccak-$f[b](A)$
  for $i$ in $0 \ldots n_\mathrm{r} - 1$
    $A = \mathrm{Round}[b](A, \mathrm{RC}[i])$
  return $A$
```

[1] Thanks to Guido Bertoni and Joan Daemen and Michaël Peeters and Gilles Van Assche for providing this text

A Keccak-f round consists of a sequence of invertible steps each operating on the state, organized as an array of 5×5 *lanes*, each of length $w \in \{1, 2, 4, 8, 16, 32, 64\}$ $(b = 25w)$. When implemented on a 64-bit processor, a lane of Keccak-$f[1600]$ can be represented as a 64-bit CPU word.

```
Round$[b](A, \mathrm{RC})$
  $\theta$ STEP
    $C[x] = A[x,0] \oplus A[x,1] \oplus A[x,2] \oplus A[x,3] \oplus A[x,4]$,                $\forall x$ in $0 \ldots 4$
    $D[x] = C[x-1] \oplus \mathrm{ROT}(C[x+1], 1)$,                                          $\forall x$ in $0 \ldots 4$
    $A[x,y] = A[x,y] \oplus D[x]$,                                                           $\forall (x,y)$ in $(0 \ldots 4, 0 \ldots 4)$

  $\rho$ AND $\pi$ STEPS
    $B[y, 2x+3y] = \mathrm{ROT}(A[x,y], r[x,y])$,                                            $\forall (x,y)$ in $(0 \ldots 4, 0 \ldots 4)$

  $\chi$ STEP
    $A[x,y] = B[x,y] \oplus ((\mathrm{NOT}\, B[x+1,y])\, \mathrm{AND}\, B[x+2,y])$,          $\forall (x,y)$ in $(0 \ldots 4, 0 \ldots 4)$

  $\iota$ STEP
    $A[0,0] = A[0,0] \oplus \mathrm{RC}$

  return $A$
```

Here the following conventions are in use. All the operations on the indices are done modulo 5. A denotes the complete permutation state array and $A[x, y]$ denotes a particular lane in that state. $B[x, y]$, $C[x]$ and $D[x]$ are intermediate variables. The symbol $\oplus$ denotes the bitwise exclusive OR, NOT the bitwise complement and AND the bitwise AND operation. Finally, $\mathrm{ROT}(W, r)$ denotes the bitwise cyclic shift operation, moving bit at position i into position $i + r$ (modulo the lane size).

The constants $r[x, y]$ are the cyclic shift offsets and are specified in the following table.

	$x = 3$	$x = 4$	$x = 0$	$x = 1$	$x = 2$
$y = 2$	25	39	3	10	43
$y = 1$	55	20	36	44	6
$y = 0$	28	27	0	1	62
$y = 4$	56	14	18	2	61
$y = 3$	21	8	41	45	15

The constants $\mathrm{RC}[i]$ are the round constants. The following table specifies their values in hexadecimal notation for lane size 64. For smaller sizes they must be truncated.

RC[0]	0x0000000000000001	RC[12]	0x000000008000808B
RC[1]	0x0000000000008082	RC[13]	0x800000000000008B
RC[2]	0x800000000000808A	RC[14]	0x8000000000008089
RC[3]	0x8000000080008000	RC[15]	0x8000000000008003
RC[4]	0x000000000000808B	RC[16]	0x8000000000008002
RC[5]	0x0000000080000001	RC[17]	0x8000000000000080
RC[6]	0x8000000080008081	RC[18]	0x000000000000800A
RC[7]	0x8000000000008009	RC[19]	0x800000008000000A
RC[8]	0x000000000000008A	RC[20]	0x8000000080008081
RC[9]	0x0000000000000088	RC[21]	0x8000000000008080
RC[10]	0x0000000080008009	RC[22]	0x0000000080000001
RC[11]	0x000000008000000A	RC[23]	0x8000000080008008

We obtain the Keccak$[r, c]$ sponge function, with parameters capacity c and bitrate r, if we apply the sponge construction to Keccak-$f[r + c]$ and perform specific padding on the message input. The following pseudocode is restricted to the case of messages that span a whole number of bytes and where the bitrate r is a multiple of the lane size.

```
Keccak$[r,c](M)$
  Padding
  $P = M||\texttt{0x01}||\texttt{0x00}||\ldots||\texttt{0x00}$
  $P = P \oplus \texttt{0x00}||\ldots||\texttt{0x00}||\texttt{0x80}$

  Initialization
  $S[x,y] = 0,$                                   $\forall(x,y)$ in $(0\ldots4, 0\ldots4)$

  Absorbing phase
  for every block $P_i$ in $P$
    $S[x,y] = S[x,y] \oplus P_i[x+5y],$           $\forall(x,y)$ such that $x+5y < r/w$
    $S =$ Keccak-$f[r+c](S)$

  Squeezing phase
  $Z =$ empty string
  while output is requested
    $Z = Z||S[x,y],$                              $\forall(x,y)$ such that $x+5y < r/w$
    $S =$ Keccak-$f[r+c](S)$

  return $Z$
```

Here S denotes the state as an array of lanes. The padded message P is organised as an array of blocks P_i, themselves organized as arrays of lanes. The $||$ operator denotes byte string concatenation.

Chapter 2

Proofs for Keccak

We devide the proofs for Keccak into two parts. Reductionist security arguments, and arguments against classes of cryptanalytic attacks, basic differential attacks in particular. Whereas for Keccak there is actually no reductionist security proof because it uses a fixed permutation, Keccak is based on the Sponge construction which is proven secure in various ways assuming the underlying permutation to be ideal.

The fact that KECCAK comes with this kind of proofs should not be interpreted as it being invulnerable against any type of cryptanalysis, However, both security reductions and provable resistance against basic differential cryptanalysis guarantee that the hash function has no severe structural weaknesses, and in particular that the design does not suffer weaknesses that can be trivially exploited by cryptanalysts.

2.1 Reductionist security arguments

2.1.1 Preliminaries

We denote by $\mathrm{Func}(m,n)$ the set of all functions $f : \mathbb{Z}_2^m \to \mathbb{Z}_2^n$. A random oracle [BR93] is a function which provides a random output for each new query. A random m-to-n-bit function is a function sampled uniformly at random from $\mathrm{Func}(m,n)$. A random primitive will also be called "ideal". The set of functions Func may be restricted, for instance to contain block ciphers or permutations only.

2.1.2 Collision, Preimage, and Second Preimage Security

In the ideal model, a *compressing* function F (either on fixed or arbitrary input lengths) that uses one or more underlying building blocks is viewed insecure if there exists a successful information-theoretic adversary that has only query access to the idealized underlying primitives of F. The complexity of the attack is measured by the number of queries q to the primitive made by the adversary. In this work it is clear from the context which of the underlying primitives is assumed to be ideal. We consider preimage, second preimage and collision resistance. For each of these three notions, with $\mathbf{Adv}_F^{\mathrm{atk}}$, where $\mathrm{atk} \in \{\mathrm{pre}, \mathrm{sec}, \mathrm{col}\}$, we denote the maximum advantage of an adversary to break the function F under the security notion atk. The advantage is the probability function taken over all random choices of the underlying primitives, and the maximum is taken over all adversaries that make at most q queries to their oracles.

If a compressing function F outputs a bit string of length n, one expects to find collisions with high probability after approximately $2^{n/2}$ queries (due to the birthday attack). Similarly, (second) preimages can be found with high probability after approximately 2^n queries[1]. Moreover, finding second preimages is provably harder than finding collisions, and similar for preimages (depending on the specification of F) [RS04].

2.1.3 Indifferentiability

The indifferentiability framework introduced by Maurer et al. [MRH04] is an extension of the classical notion of indistinguishability; it ensures that a hash function has no structural defects. We denote the indifferentiability security of a hash function $\mathcal{H}$ by $\mathbf{Adv}_{\mathcal{H}}^{\text{pro}}$, maximized over all distinguishers making at most q queries of maximal length $K \geq 0$ message blocks to their oracles. We refer to Coron et al. [CDMP05] for a formal definition. An indifferentiability bound guarantees security of the hash function against specific attacks. Although recent results by Ristenpart et al. [RSS11] show that indifferentiability does not capture all properties of a random oracle, indifferentiability still remains the best way to rule out structural attacks for a large class of hash function applications.

It has been demonstrated in [AMP10a, AMP10b] that

$$\mathbf{Adv}_{\mathcal{H}}^{\text{atk}} \leq \mathbf{Pr}_{RO}^{\text{atk}} + \mathbf{Adv}_{\mathcal{H}}^{\text{pro}} \tag{2.1}$$

for any security notion atk, where $\mathbf{Pr}_{RO}^{\text{atk}}$ denotes the success probability of a generic attack against $\mathcal{H}$ under atk and RO is an ideal function with the same domain and range space as $\mathcal{H}$.

```
Keccak:
$(n, l, m) \in \{(256, 1600, 1088), (512, 1600, 576)\}$
$P : \mathbb{Z}_2^l \to \mathbb{Z}_2^l$ permutation
$f(h, M) = P(h \oplus (M \| 0^{l-m}))$

Keccak$(M) = h$, where:
  $(M_1, \ldots, M_k) \leftarrow M \| 10^{-|M|-2 \bmod m} 1$;  $h_0 \leftarrow iv$
  $h_i \leftarrow f(h_{i-1}, M_i)$ for $i = 1, \ldots, k$
  $h \leftarrow \mathsf{chop}_n(h_k)$
```

Figure 2.1: iv denotes an initialization vector, h denotes state values, M denotes message blocks.

2.1.4 Application and results for Keccak

The **Keccak** hash function [BDPA11] is a sponge function, but can also be considered as a parazoa function [AMP12] or a chop-Merkle-Damgård construction. The compression function f is based on a permutation $\mathbb{Z}_2^l \to \mathbb{Z}_2^l$. The hash function output is obtained by chopping off $l - n$ bits of the state[2]. Notice that the parameters of Keccak satisfy $l = 2n + m$. The Keccak hash function design is given in Fig. 2.1.

[1] Kelsey and Schneier [KS05] describe a second preimage attack on the Merkle-Damgård hash function that requires at most approximately 2^{n-L} queries, where the first preimage is of length at most 2^L blocks.

[2] We notice that sponge function designs are more general [BDPV07], but for Keccak this description suffices.

The compression function of Keccak is based on one permutation, and collisions and preimages for the compression function can be found in one query to the permutation [BCS05]. The Keccak hash function is proven indifferentiable from a random oracle up to bound $\Theta((Kq)^2/2^{l-m})$ if the underlying permutation is assumed to be ideal [BDPV08]. Using (2.1), this indifferentiability bound renders an optimal collision resistance bound for Keccak, $\mathbf{Adv}_{\mathcal{H}}^{\mathrm{col}} = \Theta(q^2/2^n)$, as well as optimal preimage resistance $\mathbf{Adv}_{\mathcal{H}}^{\mathrm{epre}} = \Theta(q/2^n)$ and second preimage resistance $\mathbf{Adv}_{=}^{\mathrm{esec}}\Theta(q/2^n)$.

2.2 Arguments against classes of cryptanalytic attacks

The security notions and design approaches mentioned so far substantially assume the underlying primitives of hash functions (such as compression functions, permutations or block ciphers) to behave in an idealized way, where the primitive is randomly drawn from the corresponding class of primitives. However, any practical setting requires the primitives to be efficiently implementable, their representation being compact. As a matter of fact, this does not comply to the random procedure of choice assumed, since it is extremely improbable to select a compactly implementable primitive at random. Thus, once the rule of domain extension has been proven sound assuming the idealness of the underlying primitive, the problem of evaluating the concrete primitive with respect to real-world attacks arises.

In here, echoing [ABM+12], we choose to employ the toolbox of differential cryptanalysis [BS91] to address the latter problem, particularly because this analysis approach is also responsible for the attacks on MD5 and SHA-1 [WYY05,WY05], that are the main motivation for the SHA-3 competition.

The security of hash functions with respect to such central requirements as (second) preimage and collision resistance can be reformulated in terms of the input and output differences of the underlying primitives.

Differentials, DP, EDP. Strictly speaking, for some primitive ϕ mapping to n bits, we do not want the differential probability (DP) of any non-trivial differential (Δ, ∇) over ϕ to significantly deviate from 2^{-n} (see [DR07] for a comprehensive statistical study of this parameter for idealized permutations and functions). The differential (Δ, ∇) for primitive ϕ consists of input difference Δ and output difference ∇. Once the parameters of ϕ are fixed (keys, salts, initial vectors, etc.), one speaks about the differential probability DP as the probability for (Δ, ∇) to hold averaged over all inputs. The expected DP (EDP) is the DP averaged over all sets of parameters for ϕ.

Differential trails, DTP, EDTP. For most practical constructions, however, also for Keccak, it is often impossible to derive any tight (and, thus, informative) upper bounds on DP or even EDP. That is why, to simplify the analysis, one frequently has to revert to differential trails and their probabilities for the evaluation of designs with respect to differential cryptanalysis [BS91, DR02]. A differential can be seen as the set of all difference propagation paths from Δ to ∇ through intermediate differences corresponding to the iterations of an iterative construction. Each of these paths is called a differential trail. The probability that a differential trail holds is referred to as differential trail probability (DTP). Similarly to differentials, the expected DTP averaged over all parameters will be denoted as EDTP. We refer to the upper bounds on EDP and EDTP (attained or not attained) as MEDP and MEDTP, respectively.

Table 2.1: Maximum EDTP (MEDTP) for Keccak.

	hash size n	rounds R	$\log_2$MEDTP for R	r for MEDTP $\approx 2^{-n}$	r/R for r with $\log_2$MEDTP $\approx -n$	$\log_2$MEDTP for $R/4$	$\log_2$ MEDTP for $R/3$	$\log_2$MEDTP for $R/2$
Keccak	224	24	-296	24	1	-74	-74	-148
	256	24	-296	24	1	-74	-74	-148
	384	24	-296	> 24	> 1	-74	-74	-148
	512	24	-296	> 24	> 1	-74	-74	-148

Bounds on DTP for Keccak. The Keccak permutation consists of 24 rounds. In [DA12] for the upper bound on the EDTP a value of 2^{-32} is proven for 3 rounds, 2^{-74} for 6 rounds, and 2^{-296} for the full 24 rounds.

A note on some limitations of the obtained results. Strictly speaking, given a bound on the expected differential trail probability it is hard to say what it exactly implies for the expected differential probability, since there might be strong differential effects. More, an upper bound on the expected differential trail probability (averaged over some parameters of the primitive) can make only a limited statement about the differential probability taken for the fixed parameters or the maximum differential probability taken over all inputs. At the same time, it is exactly the latter properties which are related to the real-world attack complexities on hash functions. However, even the latter properties only allow statements against basic differential attacks, and do not take into account more advanced techniques, such as message modification [WYY05, WY05], condition propagation [DR06, DMR07], or rebound attacks [MRST09, KNR10, DGPW12].

Chapter 3

Cryptanalysis

3.1 On the algebraic degree and zero-sum structures for Keccak-f

Some of the properties that were first analyzed for Keccak were properties related to its algebraic degree. More precisely, the existence of some structures, named zero-sums structures, was intensively studied for the inner permutation of Keccak in different research papers [AM09, BC10a, BC10b, BCC11, DL11, BC13b]. In the next section, we will present the notion of zero-sums and zero-sum partitions and we will show how these notions were applied to Keccak.

3.1.1 Zero-sums and zero-sum partitions

We start by introducing the notion of a zero-sum for a vectorial function F.

Zero-sums

Definition 1 *Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. A* zero-sum *for F of size K is a subset $\{x_1, \dots, x_K\} \subset \mathbf{F}_2^n$ of elements which sum to zero and for which the corresponding images by F also sum to zero, i.e.,*

$$\sum_{i=1}^{K} x_i = \sum_{i=1}^{K} F(x_i) = 0 \ .$$

In [AM09], Aumasson and Meier searched for the existence of zero-sums for some round-reduced versions of the permutation Keccak-f. They were able to construct zero-sums for up to 16 rounds of the permutation with complexity 2^{1025}. The method that they used is quite simple and elegant and is based on the fact that hash functions are constructions using no key and thus it is possible to perform computations starting from the middle. This idea was used for the first time by Knudsen and Rijmen [KR07] in order to construct distinguishers for block ciphers in the known-key model. For constructing zero-sums for 16 rounds of Keccak-f Aumasson and Meier computed that after 10 rounds of the permutation, the algebraic degree could not exceed $2^{10} = 1024$ as the algebraic degree of the round permutation is 2. In the same way, 6 rounds of the inverse permutation were estimated to be of degree at most $3^6 = 729$ as the degree of the inverse round permutation is 3. Thus by choosing a subspace $V \subset \mathbf{F}_2^{1600}$ of dimension $d > \max(1024, 729)$ in an intermediate state after 6 rounds and computing forwards and backwards the authors get easily a zero-sum for 16-rounds of KECCAK-f. This method will be described in details in the sequel.

For a given function $F : \mathbf{F}_2^n \to \mathbf{F}_2^m$, a generic method for finding zero-sums is given by Wagner's generalized birthday algorithm. However, for zero-sums of size $k > n+m$, Wagner's algorithm can be improved by the attack XHASH due to Bellare and Micciancio [BM97], as this was pointed out in [AKK+10, BDPV10]. A general algorithm inspired by the XHASH attack was described by Bertoni et al. [BDPV10] (Algorithm 2). The complexity of this generic algorithm is equal to approximately $k + n + m$ evaluations of F and the resolution of a linear system that can be done in $\mathcal{O}((n+m)^3)$.

In [AM09], the authors seemed to suggest that the existence of a single zero-sum for a permutation could imply a distinguishing property for it. In this direction, Boura and Canteaut investigated in [BC10a] the following question: Do zero-sums exist for every permutation and if yes, what is the minimal size of a zero-sum? They found thus that each permutation P possesses at least one zero-sum of size 5. Furthermore this lower bound is attended for a special class of permutations, the so-called APN permutations.

As a consequence, the existence of a zero-sum for a given function cannot be considered as a distinguishing property. However, if the function is a permutation then an interesting property holds: A coset of a zero-sum is still a zero-sum. This leads to a much stronger property, named *zero-sum partition.*

Zero-sum partitions

Definition 2 *Let P be a permutation from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. A* zero-sum partition *for P of size $K = 2^k$ is a collection of 2^{n-k} disjoint zero-sums $X_i = \{x_{i,1}, \dots, x_{i,2^k}\} \subset \mathbf{F}_2^n$, i.e.,*

$$\bigcup_{i=1}^{2^{n-k}} X_i = \mathbf{F}_2^n \text{ and } \sum_{j=1}^{2^k} x_{i,j} = \sum_{j=1}^{2^k} P(x_{i,j}) = 0, \ \forall 1 \leq i \leq 2^{n-k} .$$

The best known generic attack for finding zero-sum partitions of size k for a permutation of $\mathbf{F}_2^n$ consists in recursively applying the XHASH attack. The total complexity of this attack is approximated by

$$2^n - 2^k + (2n)^3(2^{n-k} - 1).$$

A particular point of all the generic attacks for finding zero-sum partitions is that the permutation has to be evaluated at almost all points of the subspace, since the research method is not deterministic. For this reason, it is very interesting to study zero-sum partitions coming from some structural property of the permutation.

3.1.2 Exploiting structural properties of a permutation

Most of the symmetric primitives have an iterative structure. We will present in this section some methods for searching for zero-sum partitions for iterative permutations of the form

$$P = R_r \circ \dots \circ R_1,$$

where R_i's are simple parametrized permutations, usually called *round permutations.* As presented in [BC10b] there are two general methods for searching for zero-sum partitions, one by exploiting the algebraic degree of the permutation and of its inverse and the second one based on properties of the diffusion part. The first method is the one introduced by Aumasson and Meier in [AM09] and used for constructing zero-sums for the inner permutations of the hash functions Keccak, Luffa and Hamsi.

Constructing zero-sum partitions from higher-order differentials

If F is a permutation, then every subspace $V \subset \mathbf{F}_2^n$ of dimension $(\deg F + 1)$ leads to a zero-sum partition. This results comes from the properties of higher-order differentials as

$$D_V F(x) = \sum_{v \in V} F(x+v) = 0, \text{ for every } x \in \mathbf{F}_2^n.$$

The only information that ones need to know in order to use this first approach is an upper bound for the degree of the round transformation and of its inverse.

Let $P = R_r \circ \cdots \circ R_1$ and t an integer $1 \leq t \leq r$. We define the functions F_{r-t} et G_t, implied in the decomposition of P :

- F_{r-t}: Function that consists of the $(r-t)$ last round transformations, that is $F_{r-t} = R_r \circ \cdots \circ R_{t+1}$
- G_t: Inverse function of the first t round transformations, that is $G_t = R_1^{-1} \circ \cdots \circ R_t^{-1}$.

The method introduced in [AM09] is described in Proposition 1 and can be visualized in Figure 3.1.

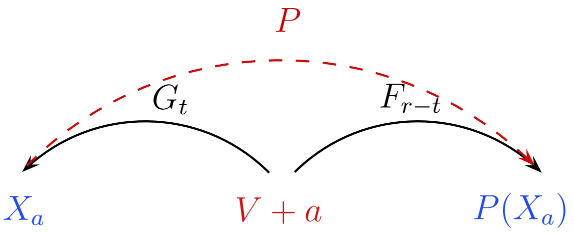

Figure 3.1: Method for constructing a zero-sum partition for an r-round permutation P.

Proposition 1 *Let d_1 and d_2 be such that $\deg(F_{r-t}) \leq d_1$ and $\deg(G_t) \leq d_2$. Let V be any subspace of $\mathbf{F}_2^n$ of dimension $d+1$ where $d = \max(d_1, d_2)$, and let W denote the complement of V, i.e., $V \oplus W = \mathbf{F}_2^n$. Then, the sets*

$$X_a = \{G_t(a+z),\ z \in V\}, \quad a \in W$$

form a zero-sum partition of $\mathbf{F}_2^n$ of size 2^{d+1} for the r-round permutation P.

It is obvious from this proposition that in order to construct zero-sums partitions for a given permutation, one has to estimate the degree of the iterated permutation while also the degree of the inverse permutation after several rounds. We will see later, how this applies to Keccak and we will present a detailed study on the evolution of the degree of Keccak-f. Before this, we present a second method, introduced in [BC10b] for constructing zero-sum partitions, exploiting this time the diffusion properties. This method, that also applies to Keccak is less efficient that the first one, however it permits in some cases to add one or two rounds to some already constructed zero-sum partitions. Theorem 1 presents the general method for adding two rounds to some already-known zero-sum partitions. This method can be easily generalized for adding more than two rounds, however this generalization is skipped in this document.

Exploiting the diffusion part

Before presenting the main theorem, we will introduce some notations given in [BC10a]. More precisely, we will denote by B_i the parts of the state on which apply the Sboxes of the round function. In the case of Keccak, the subspaces B_i correspond to the rows of the state, $i = 0, \ldots, 320$.

Theorem 1 *Let d_1 and d_2 be such that $\deg(F_{r-t-2}) \leq d_1$ and $\deg(G_t) \leq d_2$. Let L denote the linear part of the affine permutation A. Let V be a k-dimensional subspace $\mathbf{F}_2^n$, satisfying both following conditions:*

(i) *there exists a set $\mathcal{I} \subset \{0, \ldots, n_r - 1\}$ such that*

$$B_b := \bigoplus_{i \in \mathcal{I}} B_i \subset V \;\; et \; |\mathcal{I}| \geq \left\lceil \frac{d_2 + 1}{n_0} \right\rceil .$$

(ii) *there exists a set $\mathcal{J} \subset \{0, \ldots, n_r - 1\}$ such that*

$$B_f := \bigoplus_{j \in \mathcal{J}} B_j \subset L(V) \;\; et \; |\mathcal{J}| \geq \left\lceil \frac{d_1 + 1}{n_0} \right\rceil .$$

Let W denote the complement of V. Then, the sets

$$X_a = \{G_t \circ A_1^{-1} \circ \chi^{-1}(a + z), z \in V\}, a \in W,$$

form zero-sum partitions of $\mathbf{F}_2^n$ of size 2^k for the permutation r-round permutation P.

Application to Keccak

Both methods presented before were used in order to find zero-sum partitions for the inner permutation of Keccak. Aumasson and Meier [AM09] used the first method to find zero-sums structures for 16 rounds of Keccak-f. These results were then improved by Boura and Canteaut by combining both methods [BC10a, BC10b]. In these papers, the authors used more sophisticated bounds for the degree and took advantage of the diffusion part in order to extend the results of [AM09] to 17, 18, 19 and 20 rounds of the permutation.

Extension to 17 rounds It was shown in [BC10a], that 7 rounds of the inverse permutation of Keccak-f is at most 1369. This upper bound was a consequence of the direct application of a result of Canteaut and Videau [CV02] that can improve the trivial bound for the degree in the case when the values in the Walsh spectrum of the function are divisible by a high power of 2.

Theorem 2 *[CV02] Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ such that all values in its Walsh spectrum are divisible by 2^ℓ, for some integer ℓ. Then, for any $G : \mathbf{F}_2^n \to \mathbf{F}_2^n$, we have*

$$\deg(G \circ F) \leq n - \ell + \deg(G).$$

It is then mentioned in [BC10a] that the Walsh spectrum of the Sbox that is used in Keccak-f is divisible by 2^3. This is the same for the inverse Sbox. As there are 320 parallel applications of this Sbox, it is deduced that the non-linear function χ while also its inverse χ^{-1} are divisible by $2^{3 \cdot 320} = 2^{960}$. If we denote by R the round function of Keccak, we get from Theorem 2 the following bounds on the degree of 7 rounds of the inverse permutation:

$$\deg(R^{-7}) = \deg(R^{-6} \circ R^{-1}) \leq 1600 - 960 + 729 = 1369.$$

This permitted to extend the previous partitions obtained in [AM09] to one more round, by adding a round to the backwards direction. The size of these partitions was 2^{1370}.

Extension to 18 rounds The above presented zero-sum partitions can be easily extended to one more round, as shown in [BC10a], by choosing the subspace V to be the direct sum of some subspaces generated by the rows of the state and indexed by the set $\mathcal{I}$, i.e. $V = \bigoplus_{i \in \mathcal{I}} B_i$. Then as the nonlinear function χ applies independently to the rows of the state, the variables of different rows will not be mixed up together after the application of χ. This means that there exists a b such that $\chi(a + V) = b + V$. If now we set $A_1 = \pi \circ \rho \circ \theta$ and $A_2 = \iota$, the method of adding one more round to Keccak is illustrated in Figure 3.2.

$$\begin{tikzcd} {} & V+a \arrow[l, bend right, "G_t \circ A_1^{-1}"'] \arrow[r, bend left, "\chi"] & V+b \arrow[r, bend left, "F_{r-t-1} \circ A_2"] & {} \end{tikzcd}$$

Figure 3.2: Method for extending a zero-sum partition to one more round

In [BC10b] the authors set $V = \bigoplus_{i \in \mathcal{I}} B_i$ with $|\mathcal{I}| = 274$ and get zero-sum partitions for 18 rounds of the permutation.

Extension to 19 and 20 rounds By using Theorem 1 the authors in [BC10b] found zero-sum partitions for 19 and 20 rounds of Keccak-f of size 2^{1458} and 2^{1595}. The details of the method are skipped.

3.1.3 Improving the bounds on the algebraic degree for several rounds of Keccak-f

The results on Keccak were further improved in 2011 due to a better estimation of the evolution of the degree after several rounds of the internal permutation. More precisely, Boura, Canteaut and De Cannière established a new bound on the evolution of the algebraic degree of iterated permutations based on the SPN construction [BCC11]. Before presenting the main result, we introduce the following definitions and notations of [BCC11].

Let $F = (f_1, \ldots, f_m)$ be a function of $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. If $I \subset \{1, \ldots, m\}$, we denote by F^I the Boolean function of n variables corresponding to the product of the coordinates of F indexed by I :

$$F^I = \prod_{i \in I} f_i.$$

Definition 3 *Let $F = (f_1, \ldots, f_m)$ be a function of $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. Denote by $\delta_k(F)$, $1 \leq k \leq m$ the maximum degree of the product of k distinct coordinates of F.*

$$\delta_k(F) = \max_{I \subset \{0,1,\ldots,m\}, |I|=k} \deg(F^I).$$

By using the above notation, the authors proved the following bound that improves in most of the cases the trivial bound when the number of rounds gets high.

Theorem 3 *Let F be a function of $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ corresponding to the concatenation of m smaller balanced Sboxes, $S_1, \ldots, S_m$, defined over $\mathbf{F}_2^{n_0}$. Let $\delta_i(S) = \max_{1 \leq j \leq m} \delta_i(S_j)$. Then, for any function G of $\mathbf{F}_2^n$ into $\mathbf{F}_2^\ell$, we have that*

$$\deg(G \circ F) \leq n - \frac{n - \deg(G)}{\gamma}, \tag{3.1}$$

where

$$\gamma = \max_{1 \leq i \leq n_0 - 1} \frac{n_0 - i}{n_0 - \delta_i(S)}.$$

In particular, we deduce that

$$\deg(G \circ F) \leq n - \frac{n - \deg(G)}{n_0 - 1}.$$

Moreover, if $n_0 \geq 3$ and all Sboxes are of degree at most $n_0 - 2$, we have

$$\deg(G \circ F) \leq n - \frac{n - \deg(G)}{n_0 - 2}. \tag{3.2}$$

Let now apply Theorem 3 to the round permutation of Keccak-f that we denote by R. As $n_0 = 5 \geq 3$ and the degree of the non-linear permutation χ is 2 we have according to the bound (3.2),

$$\deg(G \circ R) = \deg(G \circ \chi) \leq 1600 - \frac{1600 - \deg(G)}{3}, \tag{3.3}$$

for every function G. In the same way, as $\deg(\chi^{-1}) = 3$, we have for every function G,

$$\deg(G \circ R^{-1}) = \deg((G \circ L^{-1}) \circ \chi^{-1}) \leq 1600 - \frac{1600 - \deg(G)}{3}, \tag{3.4}$$

where $L = \pi \circ \rho \circ \theta$.

In a paper that appeared later in 2011 [DL11], Duan and Lai observed that if we multiply two by two all the coordinated of χ^{-1}, the degree of the product is at most 3. By following the above notation this means that $\delta_2(R^{-1}) = \delta_2(\chi^{-1}) = 3$. By using this remark we compute the quantity γ of the Equation (3.1) for the function R^{-1}. Let

$$\gamma_i = \frac{n_0 - i}{n_0 - \delta_i(\chi^{-1})}.$$

We have that

$$\gamma_1(\chi^{-1}) = 2, \quad \gamma_2(\chi^{-1}) = 1.5, \quad \gamma_3(\chi^{-1}) = 2, \quad \text{et} \quad \gamma_4(\chi^{-1}) = 1.$$

Thus,

$$\gamma = \max_{1 \le i \le 4} \gamma_i = 2$$

and the Equation (3.4) gives now

$$\deg(G \circ R^{-1}) \le 1600 - \frac{1600 - \deg(G)}{2}. \tag{3.5}$$

Using the bounds (3.3) and (3.5), Table 3.1, containing the bounds on the degree of Keccak-f and its inverse, can be established. For the first rounds the results are obtained from the trivial bound while the results in bold are due to the new bound. For the inverse permutation, we present in the second column the results of the bound (3.4) and in the third column those of the bound (3.5).

forwards		backwards		
# rounds	bound deg(R^r)	# rounds	bound deg(R^{-r})	bound deg(R^{-r}) [DL11]
1	2	1	3	3
2	4	2	9	9
3	8	3	27	27
4	16	4	81	81
5	32	5	243	243
6	64	6	729	729
7	128	7	**1309**	**1164**
8	256	8	**1503**	**1382**
9	512	9	**1567**	**1491**
10	1024	10	**1589**	**1545**
11	**1408**	11	**1596**	**1572**
12	**1536**	12	**1598**	**1586**
13	**1578**	13	**1599**	**1593**
14	**1592**	14	**1599**	**1596**
15	**1597**	15	**1599**	**1598**
16	**1599**	16	**1599**	**1599**

Table 3.1: Upper bounds for the degree of several rounds of Keccak-f and its inverse

These estimations for the degree of Keccak-f permitted to find zero-sum partitions for the entire permutation Keccak-f. The size of these partitions was equal to 2^{1575}.

3.1.4 Explication of the observation of Duan and Lai in [DL11]

In [BC13b] Boura and Canteaut showed that the observation of Duan and Lai was not due to a random behavior of the function, but was caused by the fact that the function χ used in Keccak is of low algebraic degree. More precisely, the following result was proved.

Theorem 4 *Let F be a permutation of $\mathbf{F}_2^n$. Then, for any k and ℓ we have that*

$$\delta_\ell(F^{-1}) < n - k \text{ if and only if } \delta_k(F) < n - \ell. \tag{3.6}$$

This theorem explains the comportment observed on Keccak. Indeed, as $\delta_1(\chi) = \deg(\chi) = 2$, we get from Theorem 4 that $\delta_2(\chi^{-1}) < 4$.

3.1.5 Conclusion

The existence of zero-sum partitions does not threaten the security of the hash function. It is still an open question if these structures can be tranformed in some way to some form of attack, as a collision or a preimage attack. One of the reasons for which these structures have been so extensively studied for Keccak was the announcement by the Keccak team of the so-called "hermetic sponge strategy" that stated that no distinguisher should exist for the inner permutation. Probably the most interesting conclusion that we can keep from the research of such structures for Keccak is all the new results related on its algebraic degree. Keccak-f was an excellent example for studying the evolution of the algebraic degree of iterated permutations and provided the community with many new results in this direction.

3.2 On the Column Parity Kernel of Keccak-f

The linear transformation θ adds to each internal bit, the parity of two columns. If the parity of all the columns is even, θ acts as the indentity. As the authors of Keccak define in [BDPV12], the set of the states with all their columns summing to 0 is called the column parity kernel (CP kernel). This property applies to the values as well as to the differences.

The diffusion produced by θ is in consequence controlled while the attacker is able to stay in states that belong to the CP kernel. This property can for instance help the construction of low weight differential paths. Several analysis have exploited this property.

3.2.1 First (practical) results on reduced rounds

In [NPRM11] a distinguisher is presented on the recommended hash functions $\lfloor \text{Keccak}[1088,512] \rfloor_{256}$ and $\lfloor \text{Keccak}[1152,448] \rfloor_{224}$ when reduced to 4 rounds. Also a second preimage on two rounds, a collision on 2 rounds and a near collision on 3 rounds are described. These are the first practical results of cryptanalysis of the Keccak hash function setting where all the parameters but the number of rounds remain unchanged.

The analysis methods are based on different techniques, and propose a deep study of reduced-round Keccak and its resistance to attacks on the hash function scenario, which are stronger results than compression function ones. Though the number of rounds might seem quite reduced, the analysis are already quite technical.

Building double kernels. The paper first describes an efficient way of searching low weight differential paths. This method was used to find the differential paths that are applied for the distinguisher on 4 rounds, the collision on 2 rounds and the near collision on 3 rounds. For the distinguisher the concept of free bits as was defined in [KMNP10] is used in addition.

As previously stated, a state-difference is a *kernel* if it is invariant to one of the functions used in its permutation, θ, e.g. in each column we have a difference in zero or in an even number of bits. If we have a column where we have a difference in an odd number of bits, θ will spread this difference to 10 bits. Thus, for a low weight differential path we would like the state-differences to stay a kernel as long as possible. The designers of Keccak show in [BDPV12] that it is not possible to construct low weight differentials that are a kernel for three states in a row, however two states in a row is possible, though they are not given in the documentations. We will denote the two kernels in a row a *double kernel.*

For the search of the path, the following special property of the non-linear function χ is used: every 1-bit difference in a row constructed before χ will produce the same 1-bit difference after χ with probability 2^{-2}. Thus such a 1-bit difference will be invariant to the only non-linear part with probability 2^{-2}. If in addition we have a kernel, *i.e.* the difference is invariant to θ, we can concentrate on the remaining functions of the permutation ρ and π to find a double kernel.

For finding a double kernel the following procedure, represented in figure 3.2.1 is used. At first, the number of slices that will contain the difference in the first state is determined. Next, a bit is chosen in slice $z = 0$. Then, the following algorithm will be repeated for all bits in slice $z = 0$: From the chosen bit, the position after one application of ρ and π is computed. For this new bit position all bits in the same column are checked and their position is computed backwards by applying π^{-1} and ρ^{-1}. Next all possible bits in the same column are checked and their position after applying ρ and π are computed. This procedure continues until the wanted number of slices is affected. A double kernel is found if after the last step we are again at the original slice at the right column.

Collisions and distinguishers. This basic method allows to find all double kernels which have k active slices in each of the two kernels with a complexity of $25 * 4^{2k-1}$. Every solution will be found $2 * k$ times, since every point of the first kernel can be a starting point.

By this method we can find very fast all possible differential paths that are a kernel for two states in a row and have low hamming weight. This method was used to find suitable differential paths for the analysis. Once the best path was found, the conditional differential as described in [KMNP10] was applied for being able of detecting a bias on the output after 4 rounds, and collisions and near-collisions on 2 and 3 rounds could be built with very low complexity.

Preimage. In this paper also the first practical preimage attack is provided. Its complexity is about 2^{33} in time and 2^{29} in memory for two rounds. A meet-in-the-middle attack is performed. The issue is to find the coherent values in the middle, that are associated by the operations χ and θ. For this, the authors propose to start by finding the bits that verify the relations for a few slices. The idea is, for example, to consider first groups of three slices where we guess all the involved bits, and next we can do a sieving by just keeping the guessed ones that produce by χ and θ the values of the $5 \times 2 = 10$ known bits from the backwards computation of the group of three slices. This is possible as for computing the output of θ in

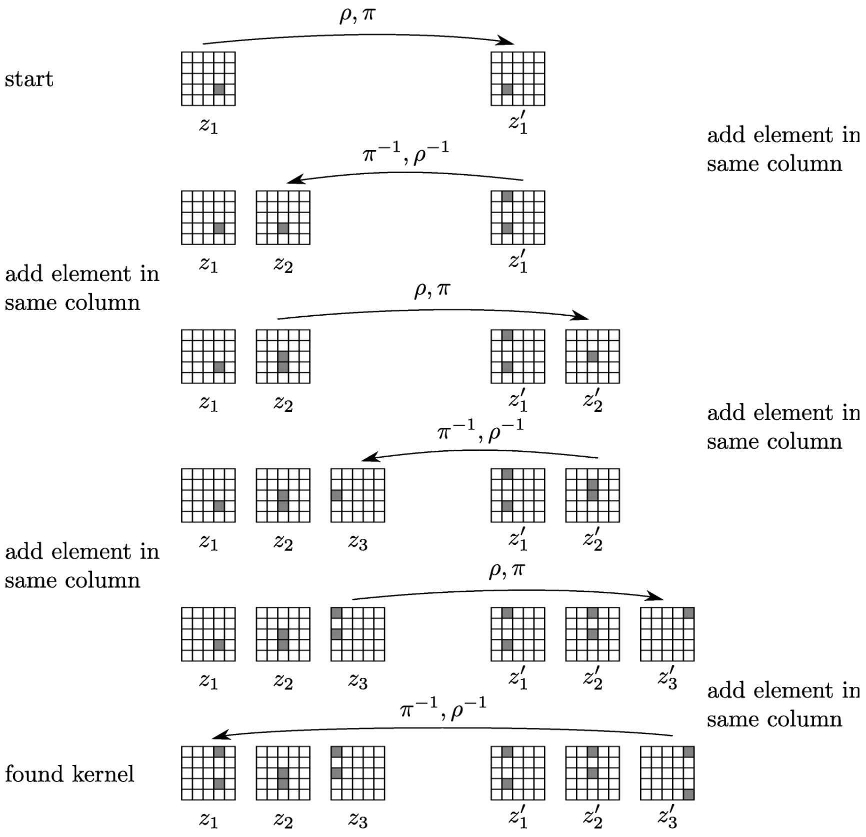

Figure 3.3: Double kernel on three slices

a specific slice, we need to know this same slice and the previous one in the input state. The key of this method is to find gradually partial solutions in parallel for a number of slices that gets bigger until reaching the solutions for the 64 slices by having merged the partial previous solutions.

Two methods can help in an efficient implementation of the attack. Let us assume we want to merge the block from slice i to j with the block from slice $j + 1$ to k. We first precompute a list containing all solutions for merging slice j and slice $j + 1$. We have 10 bits in each of the two slices, 1 repeated bit and 5 conditions from the output, thus we have in total 2^{14} solutions that we sort by the 2^{10} values in slice j. The cost of building this list is negligible in comparison to the remaining time complexities. Next, for each solution in the first block (i to j) we compute the values of the bits that will repeat in the second block. We will sort the solution in this first block by the value of the slice in j and the values of the repeated bits. We do the same thing for the second block ($j + 1$ to k) and sort it by the value of slice $j + 1$ and the values of the repeated bits. Now we can easily merge the two lists using the precomputed list of matches from slices j to $j + 1$. More details can be found in [NPRM11]

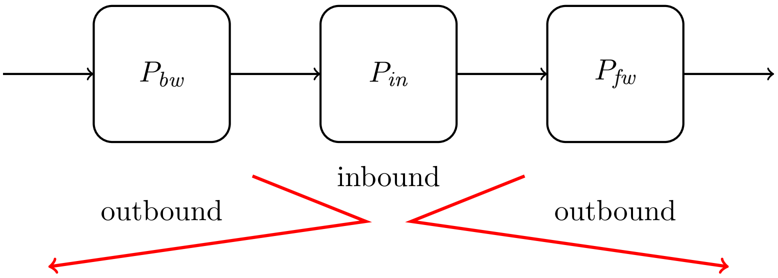

Figure 3.4: A schematic view of the rebound attack. The attack consists of an inbound and two outbound phases.

3.3 Rebound, algebraic and symmetry properties of the permutation and hash function

3.3.1 Differential approaches

In here, we briefly summarize the current known results with differential-style attacks vectors. Note that the terms characteristic, path, or trail are often used synonymously in various publications.

The rebound attack consists of two phases, called inbound and outbound phase, as shown in Figure 3.4. According to these phases, the compression function, internal block cipher or permutation of a hash function is split into three sub-parts. Let P be a permutation, such as KECCAK-f, then we get $W = W_{fw} \circ W_{in} \circ W_{bw}$. Hence, the part of the inbound phase is placed in the middle of the cipher and the two parts of the outbound phase are placed next to the inbound part. In the outbound phase, two high-probability (truncated) differential trails are constructed, which are then connected in the inbound phase. Similar to message modification, the freedom in the message, key-inputs or (internal) state variables is used to efficiently fulfill many conditions of a differential trail.

In case of the results on KECCAK by Duc et al. [DGPW12], the idea is to find differential trials for the outbound parts that allow to formulate a differential distinguisher for which the computational complexity of a generic approach is higher. As the diffusion in the backwards direction is much stronger in KECCAK, the trails are asymmetric in the sense that the forward trail is much longer than the backward trail. The end result is a differential distinguisher with complexity $2^{491.47}$ units, whereas a generic approach to produce the same distinguishing property is shown to need more than an equivalent of $2^{1057.6}$ units.

In [DDS12a] Dinur et al. present an combined differential/algebraic approach to practically produce collisions for 4 rounds of the Keccak-224 and Keccak-256 hash function. Also, this is extended to practical 5-round near-collisions. Its main ingredients are a “target difference algorithm” that bypasses the first round for a sufficiently large set of messages, and a high-probability differential characteristic for 2 or 3 rounds.

3.3.2 Symmetry and rotational properties

Two recent papers independently describe different approaches using a very related property of KECCAK. On of the properties of KECCAK is that all operations except constant addition are invariant with respect to rotation. This is exploited for a distinguisher for up to 5 rounds, and a preimage attack on up to 4 rounds by Morawiecki et al. [MPS12]. A high probability rotational trail is found for 3 rounds. This allows the authors to describe a preimage attack that is claimed to be up to 64 times faster than brute force, as 64 preimage candidates (64 is the maximal length of a lane in Keccak) can be testing in a single operation.

Dinur et al. [DDS12b] exploit a similar yet more generic property in a different way and obtain improved collision attacks, which also borrow some techniques like the target difference algorithm from [DDS12a]. New results include practical 3-round collisions for Keccak-384 and Keccak-512, as well as first theoretical collision attacks on 4-round Keccak-384 with an complexity of 2^{147}. Also, the first 5-round collision attack on Keccak-256 is given, with an estimated time complexity between 2^{108} and 2^{115} and a memory complexity of about 2^{92}.

Chapter 4

Conclusions

We believe Keccak to be an excellent alternative to the SHA-2 family of hash functions. Instead of more concluding remarks we give a list of open research problems.

We start with problems related to security proofs and arguments for Keccak.

- Better (tighter) bounds on the upper bound on the expected differential trail probability (EDTP). Only the results for 3 rounds are tight.
- Informative statements on the expected differential probability (EDP) rather than the EDTP, as these are more meaningful statements against real-world differential attacks.
- Bounds on the effectiveness of other attacks than differential attacks, e.g. those exploiting symmetry properties [DDS12c].
- Verification of proofs related to Keccak, especially on some recently proposed modes of use. In particular we point out the keyed spoinge construction [BDPV11].

Finally we give a list of open research problems related to the cryptanalysis of Keccak that might be interesting for reaching either better attack complexities for reduced-round version already attacked, or convincing results on a higher number of rounds as before.

- Study properties and behavior of the inverse of θ (a non intuitive and complex transformation).
- For the meet-in-the-middle and biclique [BKR11, KRS12] attack vector, overcome the challenge of matching at the large internal state.
- Find high probability differential characteristics for a higher number of rounds
- Investigate the extent the combination of condition propagation techniques and search strategies from SAT solvers that have been pioneered for the case of SHA-1 [DR06] and recently been applied to Hamsi (see [BC13a] and [Ku12, Section 4.4], can improve collision and preimage attacks for Keccak.

Work on various aspects of Keccak will very likely continue. This report will not be updated, instead we refer to the online resource “SHA-3 Zoo” [ECR13] that is likely to stay up to date with the ongoing developments around the security of Keccak.

Bibliography

[ABM$^+$12] Elena Andreeva, Andrey Bogdanov, Bart Mennink, Bart Preneel, and Christian Rechberger. On security arguments of the second round SHA-3 candidates. *Int. J. Inf. Sec.*, 11(2):103–120, 2012.

[AKK$^+$10] Jean-Philippe Aumasson, Emilia Käsper, Lars R. Knudsen, Krystian Matusiewicz, Rune Steinsmo Odegård, Thomas Peyrin, and Martin Schläffer. Distinguishers for the compression function and output transformation of hamsi-256. In Ron Steinfeld and Philip Hawkes, editors, *ACISP*, volume 6168 of *Lecture Notes in Computer Science*, pages 87–103. Springer, 2010.

[AM09] Jean-Phillipe Aumasson and Willi Meier. Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi. Presented at the rump session of Cryptographic Hardware and Embedded Systems - CHES 2009, 2009.

[AMP10a] Elena Andreeva, Bart Mennink, and Bart Preneel. Security reductions of the second round SHA-3 candidates. In *ISC 2010*, volume 6531 of *LNCS*, pages 39–53, Berlin, 2010. Springer-Verlag.

[AMP10b] Elena Andreeva, Bart Mennink, and Bart Preneel. Security reductions of the SHA-3 candidates. NIST's 2nd SHA-3 Candidate Conference 2010, 2010.

[AMP12] Elena Andreeva, Bart Mennink, and Bart Preneel. The parazoa family: Generalizing the sponge hash functions. *Int. J. Inf. Sec.*, 2012. To appear.

[BC10a] Christina Boura and Anne Canteaut. A zero-sum property for the KECCAK-f permutation with 18 rounds. In *IEEE International Symposium on Information Theory, ISIT'10*, pages 2488–2492. IEEE, 2010.

[BC10b] Christina Boura and Anne Canteaut. Zero-Sum Distinguishers for Iterated Permutations and Application to Keccak-f and Hamsi-256. In *Selected Areas in Cryptography-SAC'10*, volume 6544 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2010.

[BC13a] Christina Boura and Anne Canteaut. A new criterion for avoiding the propagation of linear relations through an Sbox. In *FSE'13*, Lecture Notes in Computer Science. Springer, 2013.

[BC13b] Christina Boura and Anne Canteaut. On the Influence of the Algebraic Degree of F^{-1} on the Algebraic Degree of $G \circ F$. *IEEE Transactions on Information Theory*, 59(1):691–702, 2013.

[BCC11] Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-Order Differential Properties of Keccak and *Luffa*. In *Fast Software Encryption-FSE'11*, volume 6733 of *Lecture Notes in Computer Science*, pages 252–269. Springer, 2011.

[BCS05] John Black, Martin Cochran, and Thomas Shrimpton. On the impossibility of highly-efficient blockcipher-based hash functions. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 526–541, Berlin, 2005. Springer-Verlag.

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The KECCAK sponge function family, 2011. Submission to NIST's SHA-3 competition.

[BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions, ECRYPT Hash Workshop 2007.

[BDPV08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 181–197, Berlin, 2008. Springer-Verlag.

[BDPV10] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Note on zero-sum distinguishers of KECCAK-f. Public comment on the NIST Hash competition, available at `http://keccak.noekeon.org/NoteZeroSum.pdf`, 2010.

[BDPV11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the security of the keyed sponge construction. SKEW 2011, available at `http://skew2011.mat.dtu.dk/proceedings/On%20the%20security%20of%20the%20keyed%20sponge%20construction.pdf`, 2011.

[BDPV12] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak reference, 2012.

[BKR11] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. In *ASIACRYPT*, pages 344–371, 2011.

[BM97] Mihir Bellare and Danielle Micciancio. A New Paradigm for Collision-Free Hashing: Incrementality at Reduced Cost. In *EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 163–192. Springer, 1997.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, New York, 1993. ACM.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In *CRYPTO '90*, volume 537 of *LNCS*, pages 2–21, Berlin, 1991. Springer-Verlag.

[Can12] Anne Canteaut, editor. *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*. Springer, 2012.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In *CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448, Berlin, 2005. Springer-Verlag.

[CV02] Anne Canteaut and Marion Videau. Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis. In *EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 518–533. Springer-Verlag, 2002.

[DA12] Joan Daemen and Gilles Van Assche. Differential Propagation Analysis of Keccak. In Canteaut [Can12], pages 422–441.

[DDS12a] Itai Dinur, Orr Dunkelman, and Adi Shamir. New Attacks on Keccak-224 and Keccak-256. In Canteaut [Can12], pages 442–461.

[DDS12b] Itai Dinur, Orr Dunkelman, and Adi Shamir. Self-Differential Cryptanalysis of Up to 5 Rounds of SHA-3. Cryptology ePrint Archive, Report 2012/672, to appear in FSE 2013, 2012. `http://eprint.iacr.org/`.

[DDS12c] Itai Dinur, Orr Dunkelman, and Adi Shamir. Self-Differential Cryptanalysis of Up to 5 Rounds of SHA-3. *IACR Cryptology ePrint Archive*, 2012:672, 2012.

[DGPW12] Alexandre Duc, Jian Guo, Thomas Peyrin, and Lei Wei. Unaligned Rebound Attack: Application to Keccak. In Canteaut [Can12], pages 402–421.

[DL11] Ming Duan and Xuajia Lai. Improved zero-sum distinguisher for full round Keccak-*f* permutation. IACR ePrint Report 2011/023, January 2011. `http://eprint.iacr.org/2011/023`.

[DMR07] Christophe De Cannière, Florian Mendel, and Christian Rechberger. Collisions for 70-Step SHA-1: On the Full Cost of Collision Search. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *Selected Areas in Cryptography*, volume 4876 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2007.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag, 2002.

[DR06] Christophe De Cannière and Christian Rechberger. Finding SHA-1 characteristics: General results and applications. In *ASIACRYPT'06*, volume 4284 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2006.

[DR07] Joan Daemen and Vincent Rijmen. Probability Distributions of Correlation and Differentials in Block Ciphers. *Journal of Mathematical Cryptology*, 1(3):221–242, 2007.

[ECR13] ECRYPT Symlab WG 1. The SHA-3 Zoo. available at `ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`, 2013.

[KMNP10] Simon Knellwolf, Willi Meier, and Maria Naya-Plasencia. Conditional differential cryptanalysis of NLFSR based cryptosystems. In *ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 130–145. Springer, 2010.

[KNR10] Dmitry Khovratovich, Ivica Nikolic, and Christian Rechberger. Rotational rebound attacks on reduced skein. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2010.

[KR07] Lars R. Knudsen and Vincent Rijmen. Known-Key Distinguishers for Some Block Ciphers. In *Advances in cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 315–324. Springer, 2007.

[KRS12] Dmitry Khovratovich, Christian Rechberger, and Alexandra Savelieva. Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family. In Canteaut [Can12], pages 244–263.

[KS05] John Kelsey and Bruce Schneier. Second preimages on n-bit hash functions for much less than 2^n work. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 474–490, Berlin, 2005. Springer-Verlag.

[Ku12] Özgül Küçük. Design and Analysis of Cryptographic Hash Functions. Phd Thesis. Available at `http://www.cosic.esat.kuleuven.be/publications/thesis-200.pdf`, 2012.

[MPS12] Pawel Morawiecki, Josef Pieprzyk, and Marian Srebrny. Rotational cryptanalysis of round-reduced Keccak. Cryptology ePrint Archive, Report 2012/546, to appear in FSE 2013, 2012. `http://eprint.iacr.org/`.

[MRH04] Ueli Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *TCC 2004*, volume 2951 of *LNCS*, pages 21–39, Berlin, 2004. Springer-Verlag.

[MRST09] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. The rebound attack: Cryptanalysis of reduced Whirlpool and Grøstl. In *FSE'09*, volume 5665 of *Lecture Notes in Computer Science*, pages 260–276. Springer, 2009.

[NIS] NIST. Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA3) Family. Federal Register / Vol. 72, No. 212 / Friday, November 2, 2007 / Notices.

[NPRM11] Maria Naya-Plasencia, Andrea Röck, and Willi Meier. Practical analysis of reduced-round Keccak. In *Indocrypt 2011*, volume 7107 of *Lecture Notes in Computer Science*, pages 236–254. Springer, 2011.

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *FSE 2004*, volume 3017 of *LNCS*, pages 371–388, Berlin, 2004. Springer-Verlag.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 487–506, Berlin, 2011. Springer-Verlag.

[WY05] Xiaoyun Wang and Hongbo Yu. How to Break MD5 and Other Hash Functions. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.

[WYY05] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.