Investigating Applications of The Target Difference Algorithm in Keccak Based Constructions

Midterm report of the thesis submitted to the

Indian Statistical Institute, Kolkata

For award of the degree

of

Masters of Technology in Cryptology and Security

by

Asim Manna

[Roll No: CrS1908]

Under the guidance of

Dr. Dhiman Saha

Department of Electrical Engineering and Computer Science
Indian Institute of Technology, Bhilai

Institute Supervisor

Dr. Goutam Paul

Cryptology & Security Research Unit (CSRU)
Indian Statistical Institute, Kolkata

Cryptology and Security Research Unit

INDIAN STATISTICAL INSTITUTE, KOLKATA

APRIL 2021

Contents

1 Introduction

Hash functions, particularly *Cryptographic Hash Functions (CHF)* are called the Swiss-army knife of crypto primitives. This is due to the multitude of applications that these functions contribute to. They are ubiquitous in today's digital world and are a part of almost all crypto constructions. The basic aim of a CHF is to ensure data integrity (which is why they have been referred to in coding theory literature as Modification Detection Codes) due to their ability to detect (un-)intentional modifications in data. However, they are widely deployed as the cores of Message Authentication Codes (MAC), Key Derivation Functions (KDF), Password storages, Data immutability applications like Blockchains and so on and so forth. Formally, CHF is a mathematical algorithm that maps data of arbitrary size ("message") to a bit array of a fixed size referred to as *hash* or *digest*. Recently, the notion of fixed size hash has been relaxed and the research community has witnessed constructions (SPONGE [BDPVA07]) that allow for variable (or arbitrary) length hash values. The most recent addition to the globally accepted CHF algorithms is KECCAK [BDPVA09] which won the SHA-3 competition [GJMG11] by NIST in 2012 after 5-years of intense worldwide public cryptanalysis.

This work aims to analyze KECCAK/SHA-3 and look at the differential properties of the construction. In particular this thesis will concentrate on the Target Difference Algorithm (TDA) [DDS12] introduced by Dinur *et al.* in FSE 2012. The algorithm is heuristic in nature and is used to generate a pair of input states of KECCAK/SHA-3 which after one round will produce a desired target difference. The strategy was later extended to two rounds by Qiao *et al.* in Eurocrypt 2017 to find new collision attacks on KECCAK. This algorithm combines basic algebraic techniques with differential cryptanalysis. The input of this algorithm is output difference and depending upon the output difference a system of linear equations is constructed based on the linear and non-linear layers of KECCAK which is then attempted to be solved. The solution may be consistent or inconsistent. If the system of equations is consistent then the corresponding input difference is retrieved. The next step is to get the conforming message pairs. The current work aims to understand and implement TDA on small variants of KECCAK and subsequently on the full version. Once the internal workings of the algorithm is clear, the goal is to apply the algorithm in devising a new distinguishers or collision on SHA-3 or other KECCAK-based constructions like ISAP [DEM+20] which is finalist of NIST LWC competition [NIS21] and instantiates KECCAK-[400] permutation.

Applications of Target Difference Algorithm

- Dinur *et al.*. [DDS12] used the TDA algorithm in the first part of their 4-round collinson attack on Keccak-224 and Keccak-256 in order to obtain a sufficiently large set of message pairs that satisfy the target difference after the first round of Keccak, where 4-rounds collisions were found by combining 3-round differential trails and 1-round connectors.
- Qiao *et al.*. [QSLG17] extended the above connectors one round further and hence achieve collision attacks for up to 5 rounds.
- In the paper "Practical Distinguishers against 6-Round Keccak-f Exploiting Self-Symmetry" [KSPC14], using the Simplified Target Internal Difference Algorithm they produced 2-round self-symmetric on Keccak permutation.

2 Preliminaries

In this Section, we will discuss the description on KECCAK and the Target Difference Algorithm. Subsection 2.1 illustrate the description of KECCAK and Target Difference Algorithm is explained in the Section 2.2.

2.1 Keccak Description

The KECCAK family of hash functions is based on the Sponge construction [BDPVA09]. The function f, in the sponge construction, is denoted by KECCAK-f [b], where b is the length of input string. KECCAK$-f[b]$ function is specialization of KECCAK $- p[b, n_r]$ family where $n_r = 12 + 2l$ and $l = log_2(b/25) = log_2(w)$ i.e.,

$$\text{KECCAK} - f[b] = \text{KECCAK} - p[b, 12 + 2l]$$

A state S, which is a b-bit string, in KECCAKis usually denoted by a three-dimensional grid of size $5 \times 5 \times w$. The value w depends on the parameter. We can see that from the Figure 1.1. For example, in the case of KECCAK$-f[1600]$, w is equal to 64. The value of b is 1600, so we have $l = 6$. Thus the f function in SHA-3 is KECCAK$-p[1600, 24]$. The hash function with output length d is denoted by

$$\textsc{Keccak} - d = \textsc{Keccak}[r := 1600 - 2d, c := 2d].$$

For example **Keccak**-384 means, the capacity $c = 768$ and $r = 832 = 13 \times 64$ i.e in the output state there are 13 lanes are active. The SHA-3 hash family supports minimum four different output length $d \in \{224, 256, 384, 512\}$.

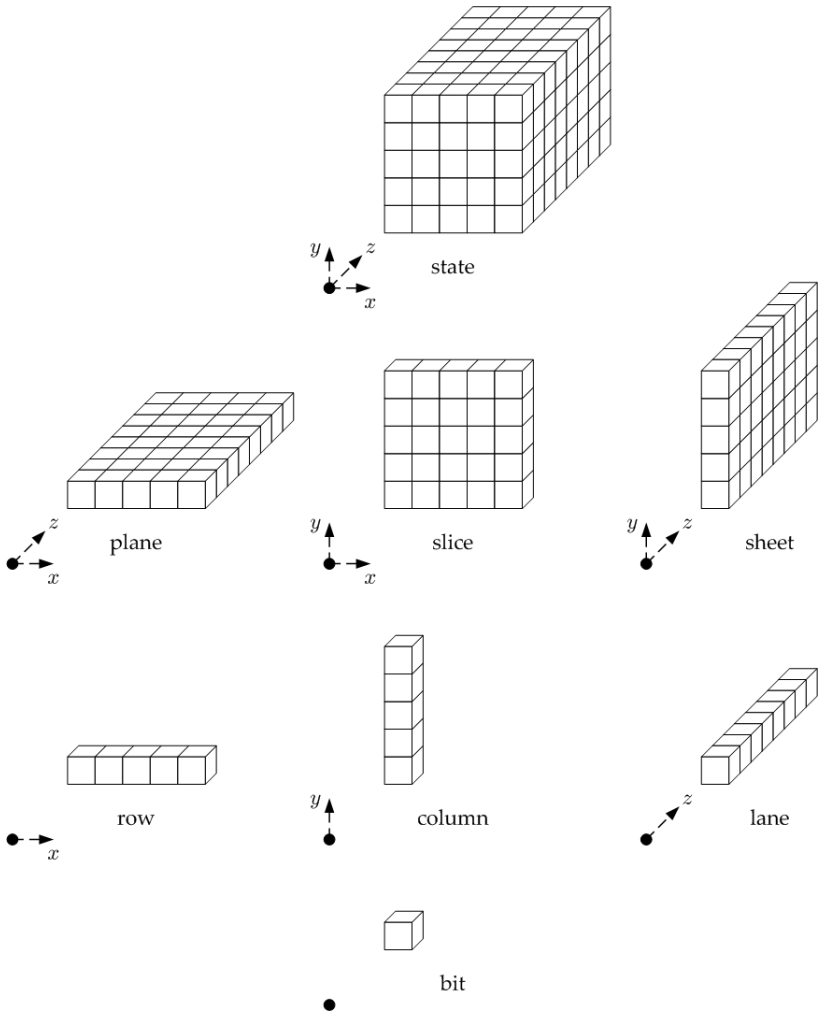

Figure 1.1: Parts of the State Array [Dwo15]

The round function p in **Keccak** consists of five steps, in each of which the state transformations specified by the step mapping. These step mappings are $\theta, \rho, \pi, \chi, \iota$. Let A and B respectively denote input and output states of a step mapping.

– θ **(Theta)** XOR to each bit the XOR of two columns. First column in same slice as the updated bit, second column in slice before updated bit. From the Figure 1.2, we can see the θ operation.
$B[x, y, z] = A[x, y, z] \oplus P[(x-1) \bmod 5, z] \oplus P[(x+1) \bmod 5, (z-1) \bmod w]$ where, $P[x, z] = \oplus_{y=0}^{4} A[x, y, z]$.

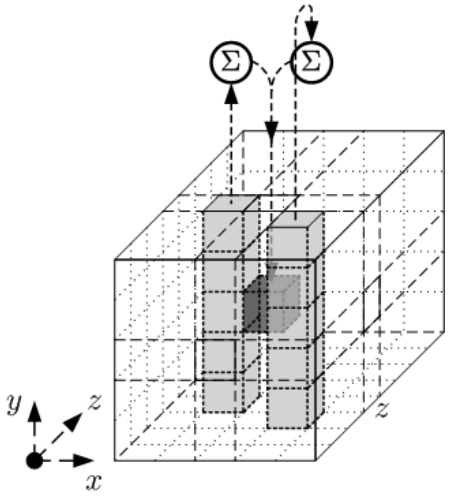

Figure 1.2: Illustration of θ applied to a single bit [Dwo15]

– ρ **(Rho)** Translate bits in z-direction. From the Figure 1.3, we can see the θ operation.

$$B[x, y, z] = A[x, y, z + \rho(x, y) \mod w]$$

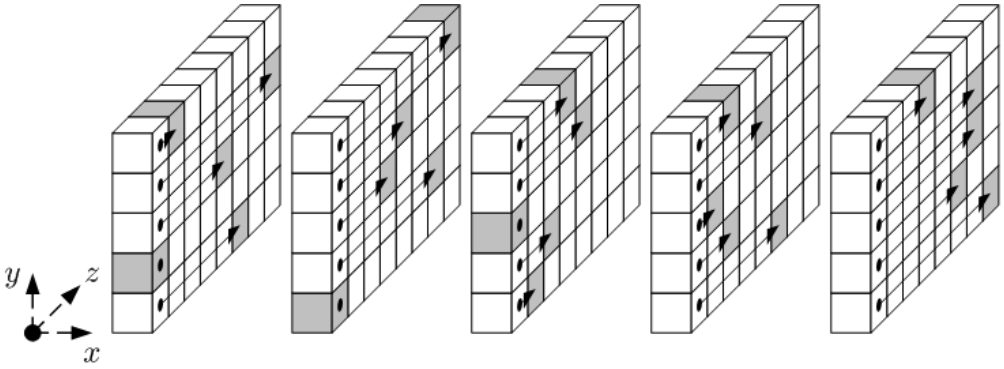

Figure 1.3: Illustration of ρ for b = 200 [Dwo15]

– π **(Pi)** Permute bits within a slice. From the Figure 1.4, we can see the π operation

$$B[y][2x + 3y][z] = A[x][y][z]$$

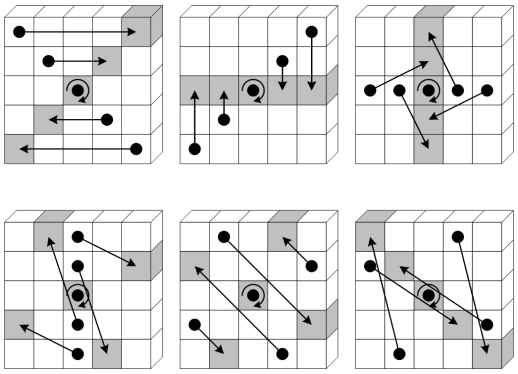

Figure 1.4: Illustration of π applied to a single slice [Dwo15]

– χ **(Chi)** This is a non-linear operation, where each bit in the original state is XOR-ed with a non-linear function of next two bits in the same row .From the Figure 1.5, we can see the χ operation

$$B[x, y, z] = A[x, y, z] \oplus ((A[(x+1) \bmod 5, y, z] \oplus 1) * (A[(x+2) \bmod 5, y, z]))$$

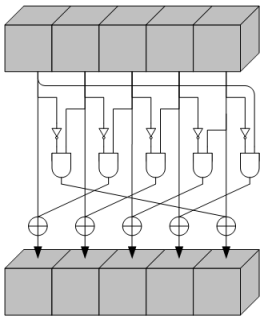

Figure 1.5: Illustration of χ applied to a single row [Dwo15]

– ι **(iota)** $B[0, 0, z] = A[0, 0, z] + RC[z]$

Thus a round in KECCAK is given by $Round(A) = \iota(\chi(\pi(\rho(\theta(A)))))$, where A is the initial state. The Chi operation of KECCAK is five bit in one S-box. Also, we know that the operation is non-linear of degree two.

2.2 Target Difference Algorithm [DDS12]

The main idea of TDA is for a given output difference Δ_T, we have to find a message pair (M^1, M^2) such that after one round KECCAK it provides the given output difference i.e $R(\bar{M}^1) + R(\bar{M}^2) = \Delta_T$, where $\bar{M}^1 = M^1||10000001||0^c$ and $\bar{M}^2 = M^2||10000001||0^c$. Now given output difference Δ_T, let Δ_I be the input difference i.e $\bar{M}^1 + \bar{M}^2 = \Delta_I$. The KECCAK consists of five step mappings θ, ρ, π, χ, ι. Here the mappings θ, ρ, π are linear and χ is non linear of degree 2. We consider L as a matrix, where $L = \rho \circ \pi \circ \theta$. The following observations are important.

- The last $c + 8$ bits of $\bar{M}^1$ are equal to $10000001||0^c$
- The last c bits of $(\bar{M}^1, \bar{M}^2)$ are same. So, The last c bits of Δ_I are zero because Δ_I is the difference (XOR) of two messages.

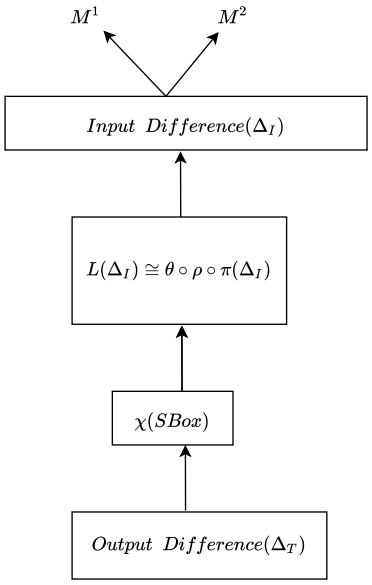

Figure 1.6: An Overview of the Target Difference Algorithm

First we split the algorithm into two phases: Difference Phase and Value Phase. In Difference Phase, we are trying to find message difference i.e Δ_I and in the Value Phase we find the value of messages. Figure 1.6 provides an overview of TDA.

Algorithm 1: Difference Phase

Result: Generate the system of equation E_Δ.

1 Initialize an empty linear equation system E_Δ, with variables for the unknown bits of $L(\Delta_I)$.
2 By inverting L, compute the coefficients of the $c + 8$ linear equations that equate the last $c + 8$ bits of Δ_I to zero, and add the equations to E_Δ.
3 For every non-active Sboxes, add to E_Δ the 5 equations which equate the corresponding bits to zero.
4 **if** E_Δ *is consistent* **then**
5 | Go to step 9.
6 **else**
7 | "Fail"
8 **end**
9 Iterate over the t active S-boxes according to the IDSD order, and for each one of them :

- Obtain the current 2-dimensional subset from the S-box IDSL (Refer Appendix B) according to the pointer, and obtain the $5 - 2 = 3$ affine equations [QSLG17] that define this subset.
- If E_Δ is consistent Go to next S-box.Otherwise, continue to the next subset in the S-box IDSL, by incrementing the pointer and going to above step. If the end of the IDSL reached, output "No Solution".

3 A Case Study on Difference Phase

In this Section, We will discuss the difference phase through an example i.e we will take one output difference of length b and then we are trying to find the corresponding input difference and message pairs. In KECCAK-$p[b, n_r]$, the width b must be 25, 50, 100, 200, 400, 800, or 1600 bits. But here we take an output difference of length 50 bits. We consider the capacity $c = 10$.

Input of this algorithm

For $b = 50$, there are 2 slices. The output difference $\Delta_T =$

1st slice

$$\begin{bmatrix}0 & 1 & 0 & 0 & 0\end{bmatrix}$$
$$\begin{bmatrix}0 & 0 & 1 & 1 & 0\end{bmatrix}$$
$$\begin{bmatrix}1 & 0 & 0 & 0 & 0\end{bmatrix}$$
$$\begin{bmatrix}0 & 0 & 0 & 0 & 0\end{bmatrix}$$
$$\begin{bmatrix}0 & 0 & 0 & 0 & 0\end{bmatrix}$$

2nd slice

$$\begin{bmatrix}0 & 1 & 0 & 0 & 1\end{bmatrix}$$
$$\begin{bmatrix}1 & 0 & 1 & 0 & 1\end{bmatrix}$$
$$\begin{bmatrix}1 & 0 & 0 & 1 & 0\end{bmatrix}$$
$$\begin{bmatrix}0 & 0 & 0 & 0 & 0\end{bmatrix}$$
$$\begin{bmatrix}0 & 0 & 0 & 0 & 0\end{bmatrix}$$

- Initialize an empty linear equation system E_Δ, with 50 variables for the unknown bits of $L(\Delta_I)$. Let $L(\Delta_I)$ is a set of 50 variables. Total there are 10 S-boxes and the number of slices is 2. The input of the S-boxes i.e $L(\Delta_I)$ is :

1st slice

$$\begin{bmatrix}x_{111} & x_{121} & x_{131} & x_{141} & x_{151}\end{bmatrix}$$
$$\begin{bmatrix}x_{211} & x_{221} & x_{231} & x_{241} & x_{251}\end{bmatrix}$$
$$\begin{bmatrix}x_{311} & x_{321} & x_{331} & x_{341} & x_{351}\end{bmatrix}$$
$$\begin{bmatrix}x_{411} & x_{421} & x_{431} & x_{441} & x_{451}\end{bmatrix}$$
$$\begin{bmatrix}x_{511} & x_{521} & x_{531} & x_{541} & x_{551}\end{bmatrix}$$

2nd slice:

$$\begin{bmatrix}x_{112} & x_{122} & x_{132} & x_{142} & x_{152}\end{bmatrix}$$
$$\begin{bmatrix}x_{212} & x_{222} & x_{232} & x_{242} & x_{252}\end{bmatrix}$$
$$\begin{bmatrix}x_{312} & x_{322} & x_{332} & x_{342} & x_{352}\end{bmatrix}$$
$$\begin{bmatrix}x_{412} & x_{422} & x_{432} & x_{442} & x_{452}\end{bmatrix}$$
$$\begin{bmatrix}x_{512} & x_{522} & x_{532} & x_{542} & x_{552}\end{bmatrix}$$

- We considered the 50×50 matrix L, where $L = \rho \circ \pi \circ \theta$. At first, we are finding L^{-1}. Now we are multiplying L^{-1} with $L(\Delta_I)$ and getting Δ_I as a set of 50 expressions. Now from the observation we get last $c = 10$ bits of Δ_I are zero. So we get 10 equations after equating the last 10 expressions of Δ_I with zero. We storing all these equations in E_Δ.

- Now, we have KECCAK S-boxes with input $L(\Delta_I)$ and output Δ_I. From the Δ_I we can see that there are total 3 (1st slice) + 3 (2nd slice)= 6 S-boxes are active and the remaining 4 S-boxes are non-active. For the non-active S-boxes, the input differences and output differences are zero. Since there are 4 non-active S-boxes. So we get 20 bits are zeroes in $L(\Delta_I)$ and store all these 20 equations in E_Δ. After adding these equation in E_Δ, we see that the system of equation E_Δ is consistent.

  $L(\Delta_I)$:

  1st slice

$$\begin{bmatrix} x_{111} & x_{121} & x_{131} & x_{141} & x_{151} \\ x_{211} & x_{221} & x_{231} & x_{241} & x_{251} \\ x_{311} & x_{321} & x_{331} & x_{341} & x_{351} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

  2nd slice:

$$\begin{bmatrix} x_{112} & x_{122} & x_{132} & x_{142} & x_{152} \\ x_{212} & x_{222} & x_{232} & x_{242} & x_{252} \\ x_{312} & x_{322} & x_{332} & x_{342} & x_{352} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Here, we have six active KECCAK S-boxes. For an active S-boxes consider the output difference (δ^{out}) and corresponding input difference (δ^{in}), where DDT(δ^{in}, δ^{out}) > 0. Now from the DDT table we can find the possible input difference set. For example, if the output

difference (δ^{out}) is 1 then the corresponding input difference set is {1, 3, 5, 7, 11, 15, 21, 23, 31}. After getting the input difference set, our next work is to find the possible 2-dimensional affine subspaces (Refer Section 4) from the input difference set. After obtaining the 2-dimensional affine subset we then narrow $5 - 2 = 3$ affine equations that defines this 5-bit input of current S-box to E_Δ. If E_Δ is consistent after adding these 3-affine equations in E_Δ, add the equations and continue to the next active S-box. Now if the E_Δ becomes inconsistent after adding these equations continue to the next affine subsets and so on. After the above procedure, the system of equation E_Δ :

$$x_{111} \oplus x_{112} \oplus x_{121} \oplus x_{122} \oplus x_{131} \oplus x_{142} \oplus x_{211} \oplus x_{231} \oplus x_{232} \oplus x_{241} \oplus x_{242} \oplus x_{251} \oplus x_{311} \oplus x_{312} \oplus x_{322} \oplus x_{332} \oplus x_{352} = 0$$

$$x_{121} \oplus x_{122} \oplus x_{131} \oplus x_{132} \oplus x_{141} \oplus x_{151} \oplus x_{212} \oplus x_{221} \oplus x_{241} \oplus x_{242} \oplus x_{251} \oplus x_{252} \oplus x_{311} \oplus x_{312} \oplus x_{321} \oplus x_{322} \oplus x_{331} \oplus x_{341} = 0$$

$$x_{111} \oplus x_{131} \oplus x_{132} \oplus x_{141} \oplus x_{142} \oplus x_{152} \oplus x_{211} \oplus x_{212} \oplus x_{222} \oplus x_{232} \oplus x_{251} \oplus x_{321} \oplus x_{322} \oplus x_{331} \oplus x_{332} \oplus x_{342} \oplus x_{351} = 0$$

$$x_{112} \oplus x_{121} \oplus x_{141} \oplus x_{142} \oplus x_{151} \oplus x_{152} \oplus x_{211} \oplus x_{212} \oplus x_{221} \oplus x_{222} \oplus x_{231} \oplus x_{242} \oplus x_{312} \oplus x_{331} \oplus x_{332} \oplus x_{341} \oplus x_{342} \oplus x_{352} = 0$$

$$x_{111} \oplus x_{112} \oplus x_{122} \oplus x_{132} \oplus x_{152} \oplus x_{221} \oplus x_{222} \oplus x_{231} \oplus x_{232} \oplus x_{241} \oplus x_{252} \oplus x_{311} \oplus x_{321} \oplus x_{341} \oplus x_{342} \oplus x_{351} \oplus x_{352} = 0$$

$$x_{111} \oplus x_{112} \oplus x_{121} \oplus x_{122} \oplus x_{132} \oplus x_{141} \oplus x_{212} \oplus x_{231} \oplus x_{232} \oplus x_{241} \oplus x_{242} \oplus x_{252} \oplus x_{311} \oplus x_{312} \oplus x_{321} \oplus x_{331} \oplus x_{351} = 0$$

$$x_{121} \oplus x_{122} \oplus x_{131} \oplus x_{132} \oplus x_{142} \oplus x_{152} \oplus x_{211} \oplus x_{222} \oplus x_{241} \oplus x_{242} \oplus x_{251} \oplus x_{252} \oplus x_{311} \oplus x_{312} \oplus x_{321} \oplus x_{322} \oplus x_{332} \oplus x_{342} = 0$$

$$x_{112} \oplus x_{131} \oplus x_{132} \oplus x_{141} \oplus x_{142} \oplus x_{151} \oplus x_{211} \oplus x_{212} \oplus x_{221} \oplus x_{231} \oplus x_{252} \oplus x_{321} \oplus x_{322} \oplus x_{331} \oplus x_{332} \oplus x_{341} \oplus x_{352} = 0$$

$$x_{111} \oplus x_{122} \oplus x_{141} \oplus x_{142} \oplus x_{151} \oplus x_{152} \oplus x_{211} \oplus x_{212} \oplus x_{221} \oplus x_{222} \oplus x_{232} \oplus x_{241} \oplus x_{311} \oplus x_{331} \oplus x_{332} \oplus x_{341} \oplus x_{342} \oplus x_{351} = 0$$

$$x_{111} \oplus x_{112} \oplus x_{121} \oplus x_{131} \oplus x_{151} \oplus x_{221} \oplus x_{222} \oplus x_{231} \oplus x_{232} \oplus x_{242} \oplus x_{251} \oplus x_{312} \oplus x_{322} \oplus x_{341} \oplus x_{342} \oplus x_{351} \oplus x_{352} = 0$$
$$x_{121} \oplus x_{131} = 1$$

$x_{121} \oplus x_{141} = 1$
$x_{121} \oplus x_{151} = 1$
$x_{251} = 1$
$x_{211} \oplus x_{221} = 0$
$x_{211} \oplus x_{231} = 0$
$x_{321} = 0$
$x_{311} \oplus x_{331} = 1$
$x_{311} \oplus x_{351} = 1$
$x_{112} \oplus x_{122} = 0$
$x_{112} \oplus x_{142} = 1$
$x_{112} \oplus x_{152} = 0$
$x_{252} = 0$
$x_{242} = 1$
$x_{212} \oplus x_{232} = 1$
$x_{312} \oplus x_{332} = 1$
$x_{322} \oplus x_{342} = 0$
$x_{312} \oplus x_{322} \oplus x_{352} = 1$

The system of equation E_Δ has 28 equations and 30 variables. So, the solution of this system of equation is not unique. Then one of the solution i.e $L(\Delta_I)$ is :

1st slice

$$\begin{matrix} [0 & 1 & 0 & 0 & 1] \\ [0 & 1 & 0 & 1 & 0] \\ [1 & 0 & 0 & 0 & 0] \\ [0 & 0 & 0 & 0 & 0] \\ [0 & 0 & 0 & 0 & 0] \end{matrix}$$

2nd slice

$$\begin{matrix} [0 & 1 & 0 & 0 & 0] \\ [1 & 1 & 1 & 1 & 1] \\ [0 & 1 & 1 & 1 & 0] \\ [0 & 0 & 0 & 0 & 0] \\ [0 & 0 & 0 & 0 & 0] \end{matrix}$$

After putting this solution on the 50 expression we get the inuput difference Δ_I.
Δ_I:

1st slice

$$\begin{bmatrix}0 & 0 & 1 & 0 & 1\\ 0 & 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 1 & 1\\ 0 & 1 & 1 & 0 & 1\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

2nd slice

$$\begin{bmatrix}0 & 1 & 0 & 1 & 0\\ 0 & 1 & 1 & 0 & 1\\ 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

4 Discussion on 2-dimensional Affine Subspaces of Keccak S-box

An affine subset of a vector space V to be a subset of the form

$$A = \{a + u \mid a \in V, u \in U\}$$

where U is a subspace of V.
Consider an input difference subset where every element with 5-bits. Now the cardinality of an 2-dimensional affine subspace should be 4. So, we get $(5 - 2) = 3$ equations from an 2-dimensional affine subspace.

For example, if we take an output difference 1 then the corresponding input difference set is {1, 3, 5, 7, 11, 15, 21, 23, 31}. If we consider the subset {1, 3, 5, 7} then this subset forms an 2-dimensional affine subspace.

		x_1	x_2	x_3	x_4	x_5
1	=	0	0	0	0	1
3	=	0	0	0	1	1
5	=	0	0	1	0	1
7	=	0	0	1	1	1

δ^{out}	# 2d affine subspace	δ^{out}	# 2d affine subspace
1	9	17	9
2	9	18	5
3	9	19	5
4	9	20	5
5	5	21	17
6	9	22	17
7	5	23	11
8	9	24	8
9	5	25	5
10	5	26	17
11	17	27	12
12	9	28	5
13	17	29	12
14	5	30	12
15	11	31	10
16	9		

Table 1.1: Number of 2D Affine Subspace corresponding to All Output Difference

Here we can see that $x_1 = 0;\ x_2 = 0;\ x_5 = 1$ and $x_3,\ x_4$ are variables. So, the subset $\{1,\ 3, 5,\ 7\}$ is an 2-dimensional affine subset. The affine equations are $x_1 = 0;\ x_2 = 0;\ x_5 = 1$. But, if we consider $\{1,\ 3,\ 11,\ 15\}$ then this subset does not forms an 2-dimensional affine subspace. We can see that for this set $x_1 = 0$, $x_5 = 1$ but we can not find any linear equation from these variables.

Property 1 [DDS12]. *For a non-zero 5-bit output difference δ^{out} to a* KECCAK *S-box, the set of possible input differences, $\{\delta^{in} | DDT(\delta^{in}, \delta^{out}) > 0\}$, contains at least 5 (and up to 17) 2-dimensional affine subspaces.*

The above statement gives the answer how many affine subspaces we can get from an input difference set corresponding to an output difference (δ^{out}). We have verified this statement practically. The number of 2-dimensional affine subspaces corresponding to all output difference is given in the Table 1.1 .

.

δ^{out}	**2D affine subspaces**	**Corresponding linear equations**
1	{1, 3, 5, 7}	$x_1 = 0,\ x_2 = 0,\ x_5 = 1$
	{1, 3, 21, 23}	$x_2 = 0,\ x_5 = 1,\ x_1 \oplus x_3 = 0$
	{1, 5, 11, 15}	$x_1 = 0,\ x_5 = 1,\ x_2 \oplus x_4 = 0$
	{1, 11, 21, 31}	$x_5 = 1,\ x_1 \oplus x_3 = 0\ ,\ x_2 \oplus x_4 = 0$
	{3, 11, 7, 15}	$x_1 = 0,\ x_4 = 1,\ x_5 = 1$
	{3, 11, 23, 31}	$x_4 = 1,\ x_5 = 1,\ x_1 \oplus x_3 = 0$
	{5, 21, 7, 23}	$x_2 = 0,\ x_3 = 1,\ x_5 = 1$
	{5, 21, 15, 31}	$x_3 = 1,\ x_5 = 1,\ x_2 \oplus x_4 = 0$
	{7, 15, 23, 31}	$x_3 = 1,\ x_4 = 1,\ x_5 = 1$
5	{1, 3, 28, 30}	$x_1 \oplus x_2 = 0,\ x_1 \oplus x_3 = 0\ ,\ x_1 \oplus x_5 = 1$
	{1, 11, 23, 29}	$x_5 = 1,\ x_1 \oplus x_3 = 0,\ x_1 \oplus x_2 \oplus x_4 = 0$
	{3, 11, 7, 15}	$x_1 = 0,\ x_4 = 1\ ,\ x_5 = 1$
	{12, 7, 23, 28}	$x_3 = 1,\ x_2 \oplus x_4 = 1\ ,\ x_2 \oplus x_5 = 1$
	{12, 15, 29, 30}	$x_2 = 1,\ x_3 = 1,\ x_1 \oplus x_4 \oplus x_5 = 0$
11	{3, 9, 13, 17}	$x_1 = 0,\ x_5 = 1,\ x_2 \oplus x_4 = 1$
	{3, 9, 23, 29}	$x_5 = 1,\ x_1 \oplus x_3 = 0,\ x_2 \oplus x_4 = 1$
	{3, 13, 23, 25}	$x_5 = 1,\ x_2 \oplus x_4 = 1,\ x_1 \oplus x_2 \oplus x_3 = 0$
	{3, 7, 25, 29}	$x_5 = 1,\ x_1 \oplus x_2 = 0,\ x_1 \oplus x_4 = 1$
	{3, 7, 27, 31}	$x_4 = 1,\ x_5 = 1,\ x_1 \oplus x_2 = 0$
	{9, 13, 25, 29}	$x_4 = 0,\ x_2 = 1,\ x_5 = 1$
	{9,13, 27, 31}	$x_2 = 1,\ x_5 = 1,\ x_1 \oplus x_4 = 0$
	{9, 24, 14, 31}	$x_2 = 1,\ x_3 \oplus x_4 = 0,\ x_1 \oplus x_3 \oplus x_5 = 1$
	{9, 7, 23, 25}	$x_5 = 1,\ x_2 \oplus x_3 = 1,\ x_2 \oplus x_4 = 1$
	{9, 14, 25, 30}	$x_2 = 1,\ x_3 \oplus x_4 = 0,\ x_3 \oplus x_5 = 1$
	{13, 24, 14, 27}	$x_2 = 1,\ x_1 \oplus x_3 = 1,\ x_1 \oplus x_4 \oplus x_5 = 1$
	{13, 7, 23, 29}	$x_3 = 1,\ x_5 = 1,\ x_2 \oplus x_4 = 1$
	{13, 14, 29, 30}	$x_2 = 1,\ x_3 = 1,\ x_4 \oplus x_5 = 1$
	{24, 25, 30, 31}	$x_1 = 1,\ x_2 = 1,\ x_3 \oplus x_4 = 0$
	{24, 27, 29, 30}	$x_1 = 1,\ x_2 = 1,\ x_3 \oplus x_4 \oplus x_5 = 0$
	{7, 14, 23, 30}	$x_3 = 1,\ x_4 = 1,\ x_2 \oplus x_5 = 1$
	{25, 27, 29, 31}	$x_1 = 1,\ x_2 = 1,\ x_5 = 1$

Table 1.2: List of all 2D Affine Equations for δ^{out}= 1, 5, 11.

δ^{out}	**2D affine subspaces**	**Corresponding linear equations**
30	{6, 10, 21, 25}	$x_1 \oplus x_4 = 1$, $x_1 \oplus x_5 = 0$, $x_2 \oplus x_3 = 1$
	{6, 21, 15, 28}	$x_3 = 1$, $x_1 \oplus x_4 = 1$, $x_1 \oplus x_2 \oplus x_5 = 0$
	{10, 11, 18, 19}	$x_3 = 0$, $x_4 = 1$, $x_1 \oplus x_2 = 1$
	{10, 11, 14, 15}	$x_1 = 0$, $x_2 = 1$, $x_4 = 1$
	{10, 18, 15, 23}	$x_4 = 1$, $x_1 \oplus x_2 = 1$, $x_3 \oplus x_5 = 0$
	{10, 14, 19, 23}	$x_4 = 1$, $x_1 \oplus x_2 = 1$, $x_1 \oplus x_5 = 0$
	{10, 15, 25, 28}	$x_2 = 1$, $x_1 \oplus x_4 = 1$, $x_1 \oplus x_3 \oplus x_5 = 0$
	{11, 18, 14, 23}	$x_4 = 1$, $x_1 \oplus x_2 = 1$, $x_1 \oplus x_3 \oplus x_5 = 1$
	{11, 14, 25, 28}	$x_2 = 1$, $x_1 \oplus x_4 = 1$, $x_3 \oplus x_5 = 1$
	{11, 15, 19, 23}	$x_4 = 1$, $x_5 = 1$, $x_1 \oplus x_2 = 1$
	{18, 14, 15, 19}	$x_4 = 1$, $x_1 \oplus x_2 = 1$, $x_1 \oplus x_3 = 1$
	{18, 23, 25, 28}	$x_1 = 1$, $x_2 \oplus x_4 = 1$, $x_2 \oplus x_3 \oplus x_5 = 0$
31	{5, 9, 19, 31}	$x_5 = 1$, $x_1 \oplus x_4 = 0$, $x_1 \oplus x_2 \oplus x_3 = 1$
	{5, 10, 19, 28}	$x_2 \oplus x_5 = 1$, $x_3 \oplus x_4 = 1$, $x_1 \oplus x_2 \oplus x_3 = 1$
	{5, 18, 24, 25}	$x_1 \oplus x_3 = 1$, $x_4 \oplus x_5 = 1$, $x_1 \oplus x_2 \oplus x_4 = 0$
	{5, 20, 14, 31}	$x_3 = 1$, $x_2 \oplus x_4 = 0$, $x_1 \oplus x_2 \oplus x_5 = 1$
	{9, 10, 28, 31}	$x_2 = 1$, $x_1 \oplus x_3 = 0$, $x_1 \oplus x_4 \oplus x_5 = 1$
	{9, 18, 7, 28}	$x_1 \oplus x_5 = 1$, $x_2 \oplus x_4 = 1$, $x_1 \oplus x_2 \oplus x_3 = 1$
	{9, 20, 14, 19}	$x_1 \oplus x_2 = 1$, $x_3 \oplus x_5 = 1$, $x_1 \oplus x_3 \oplus x_4 = 0$
	{10, 18, 7, 31}	$x_4 = 1$, $x_3 \oplus x_5 = 0$, $x_1 \oplus x_2 \oplus x_3 = 1$
	{10, 20, 7, 25}	$x_1 \oplus x_4 = 1$, $x_2 \oplus x_3 = 1$, $x_1 \oplus x_2 \oplus x_5 = 1$
	{18, 20, 25, 31}	$x_1 = 1$, $x_2 \oplus x_5 = 0$, $x_2 \oplus x_3 \oplus x_4 = 1$

Table 1.3: List of all 2D Affine Equations for δ^{out}= 30, 31.

Also, we need an order for choosing the affine subspaces. For that, we have to create another list named Input Difference Subset list (Refer Appendix B) for storing the affine subspaces with some order. The IDSLs are stored in the main Input Difference Subset Data structure (IDSD). The IDSD contains t entries (one entry per active S-box), sorted according to an IDSD order.

5 Conclusion & Future Work

In this part, we have understood and implemented the diference phase of TDA. We were taking an output difference (Δ_T) as an input and find one of the input difference (Δ_I). We have also determined the 2D affine subspaces and affine equations for all δ^{out}.

In the next part, we will understand and implement the remaining phase (value phase). Also, we will try to use TDA on some part of symmetric key cryptanalysis.

Bibliography

[BDPVA07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT hash workshop*, volume 2007. Citeseer, 2007. 1

[BDPVA09] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak specifications. *Submission to NIST (round 2)*, pages 320–337, 2009. 1, 2

[DDS12] Itai Dinur, Orr Dunkelman, and Adi Shamir. New attacks on keccak-224 and keccak-256. In 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*, pages 442–461. Springer, 2012. ii, 1, 2, 6, 13

[DEM+20] CE Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. Isap v2. 0. 2020. 1

[Dwo15] Morris Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions, 2015-08-04 2015. 3, 4, 5

[GJMG11] B Guido, D Joan, P Michaël, and VA Gilles. The Keccak SHA-3 Submission. 2011. 1

[KSPC14] Sukhendu Kuila, Dhiman Saha, Madhumangal Pal, and Dipanwita Roy Chowdhury. Practical distinguishers against 6-round keccak-f exploiting self-symmetry. In David Pointcheval and Damien Vergnaud, editors, *Progress in Cryptology - AFRICACRYPT 2014 - 7th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 28-30, 2014. Proceedings*, volume 8469 of *Lecture Notes in Computer Science*, pages 88–108. Springer, 2014. 2

[NIS21] NIST. Lightweight cryptography standardization. https://csrc.nist.gov/News/2021/lightweight-crypto-finalists-announced, March 2021. 1

[QSLG17] Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo. New collision attacks on round-reduced keccak. In *EUROCRYPT (3)*, pages 216–243. Springer, 2017. 2, 7

Appendices

A Differential Distribution Table (DDT) of Keccak S-box

The χ mapping of KECCAK takes a 5-bit input and gives 5-bit output. So the size of DDT is 32×32. Output difference left to right and input difference up to down :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	8	0	8	0	8	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	8	0	0	0	8	0	0	0	8	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	8	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	8	0	0	0
5	0	4	0	4	0	0	0	0	0	0	0	0	0	4	0	4	0	4	0	4	0	0	0	0	0	0	0	0	0	4	0	4
6	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0
7	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2
8	0	0	0	0	0	0	0	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	4	0	0	4	4	0	0	4	0	0	0	0	0	0	0	0	4	0	0	4	4	0	0	4
10	0	0	4	4	0	0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	4	4	
11	0	4	4	0	0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	4	4	0
12	0	0	0	0	4	4	0	0	0	0	0	0	4	4	0	0	0	0	0	0	4	4	0	0	0	0	0	0	4	4	0	0
13	0	0	0	0	4	0	0	4	4	0	0	4	0	0	0	0	0	0	0	0	4	0	0	4	4	0	0	4	0	0	0	0
14	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2
15	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	4	4	4	4	4	0	0	0	0	4	4
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
20	0	0	0	0	4	0	4	0	0	0	0	0	4	0	4	0	0	0	0	0	0	4	0	4	0	0	0	0	0	4	0	4
21	0	4	0	4	0	0	0	0	0	0	0	0	0	4	0	4	4	0	4	0	0	0	0	0	0	0	0	0	4	0	4	0
22	0	0	4	0	4	0	0	0	0	0	4	0	4	0	0	0	0	0	0	4	0	4	0	0	0	0	0	4	0	4	0	0
23	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0
24	0	0	0	0	0	0	0	0	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	4	0	0	0	0
25	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
26	0	0	0	0	0	0	0	0	4	4	0	0	0	0	4	4	4	4	0	0	0	0	4	4	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0
28	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2
29	0	0	0	0	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	0	0	0	0
30	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0	0
31	0	2	2	0	2	0	0	2	2	0	0	2	0	2	2	0	2	0	0	2	0	2	2	0	0	2	2	0	2	0	0	2

B Input Difference Subset List

The input differences subsets, for each of the active S-boxes, are stored in a sorted input difference subset list, or IDSL. In the difference phase, we give precedence to input difference subsets containing such input differences: Assume that Δ_T assigns a specific S-box an output difference δ^{out}, and we want to compare two input difference subsets $\{\delta_1^{in}, \delta_2^{in}, \delta_3^{in}, \delta_4^{in}\}$ and $\{\delta_5^{in}, \delta_6^{in}, \delta_7^{in}, \delta_8^{in}\}$ such that $\mathrm{DDT}(\delta_1^{in}, \delta^{out}) \geq \mathrm{DDT}(\delta_2^{in}, \delta^{out}) \geq \mathrm{DDT}(\delta_3^{in}, \delta^{out}) \geq \mathrm{DDT}(\delta_4^{in}, \delta^{out}) > 0$ and $\mathrm{DDT}(\delta_5^{in}, \delta^{out}) \geq \mathrm{DDT}(\delta_6^{in}, \delta^{out}) \geq \mathrm{DDT}(\delta_7^{in}, \delta^{out}) \geq \mathrm{DDT}(\delta_8^{in}, \delta^{out}) > 0$. We first compare the sizes of the largest subset for which the size is bigger. If the sizes are equal , we compare $\mathrm{DDT}(\delta_2^{in}, \delta^{out})$ and $\mathrm{DDT}(\delta_6^{in}, \delta^{out})$, and so on.