1

CS641: THE GREAT CAVES

TEAM DEDSEC

Guntas Singh Brar 180274

Divyanshu Bhardwaj 180253

Devanshu Gupta 180233

June 15, 2020

ASSIGNMENT 7
WECCAK (WEAK-KECCAK)

1. Inversions

1.1. Inverse of χ

If we consider the input as $a = a_0a_1a_2a_3a_4$ and output as $b = b_0b_1b_2b_3b_4$, then by the definition of χ we have:

$$b_0 = a_0 \oplus (a_1 \oplus 1).a_2 \quad (1)$$
$$b_1 = a_1 \oplus (a_2 \oplus 1).a_3 \quad (2)$$
$$b_2 = a_2 \oplus (a_3 \oplus 1).a_4 \quad (3)$$
$$b_3 = a_3 \oplus (a_4 \oplus 1).a_0 \quad (4)$$
$$b_4 = a_4 \oplus (a_0 \oplus 1).a_1 \quad (5)$$

From (2) we have(since $(a_2 \oplus 1).a_2 = 0$):

$$b_1.a_2 = a_1.a_2 \oplus (a_2 \oplus 1).a_3.a_2 = a_1.a_2$$

Now using this (1) becomes:

$$b_0 = a_0 \oplus (b_1 \oplus 1).a_2$$

Since the equations are symmetrical, by carrying out same operations on the five equations we obtain following set of equations:

$$b_0 = a_0 \oplus (b_1 \oplus 1).a_2 \quad (6)$$
$$b_1 = a_1 \oplus (b_2 \oplus 1).a_3 \quad (7)$$
$$b_2 = a_2 \oplus (b_3 \oplus 1).a_4 \quad (8)$$
$$b_3 = a_3 \oplus (b_4 \oplus 1).a_0 \quad (9)$$
$$b_4 = a_4 \oplus (b_0 \oplus 1).a_1 \quad (10)$$

From (6) we have:

$$a_0 = b_0 \oplus (b_1 \oplus 1).a_2$$

Plugging this value in (9) and manipulating we get:

$$a_3 = b_3 \oplus (b_4 \oplus 1).(b_0 \oplus (b_1 \oplus 1).a_2)$$

$$a_3 = b_3 \oplus b_0.(b_4 \oplus 1) \oplus (b_1 \oplus 1).(b_4 \oplus 1).a_2$$

*

Plugging this in value of a_3 in (2) we get:

$$a_1 = b_1 \oplus (b_2 \oplus 1).(b_3 \oplus b_0.(b_4 \oplus 1) \oplus (b_1 \oplus 1).(b_4 \oplus 1).a_2)$$
$$a_1 = b_1 \oplus (b_2 \oplus 1).b_3 \oplus b_0.(b_2 \oplus 1).(b_4 \oplus 1) \oplus (b_1 \oplus 1).(b_2 \oplus 1).(b_4 \oplus 1).a_2$$

Plugging this value of a_1 in (10) we get:

$$a_4 = b_4 \oplus (b_0 \oplus 1).(b_1 \oplus (b_2 \oplus 1).b_3 \oplus b_0.(b_2 \oplus 1).(b_4 \oplus 1) \oplus (b_1 \oplus 1).(b_2 \oplus 1).(b_4 \oplus 1).a_2)$$
$$a_4 = b_4 \oplus (b_0 \oplus 1).b_1 \oplus (b_0 \oplus 1).(b_2 \oplus 1).b_3 \oplus (b_0 \oplus 1).(b_1 \oplus 1).(b_2 \oplus 1).(b_4 \oplus 1).a_2$$

Plugging this value of a_4 in (8) we get:

$$b_2 = a_2 \oplus (b_3 \oplus 1).(b_4 \oplus (b_0 \oplus 1).b_1 \oplus (b_0 \oplus 1).(b_2 \oplus 1).b_3 \oplus (b_0 \oplus 1).(b_1 \oplus 1).(b_2 \oplus 1).(b_4 \oplus 1).a_2)$$

Using distributive property and the fact that $(x \oplus 1).x = 0$, we finally have:

$$a_2 = b_2 \oplus (b_3 \oplus 1).(b_4 \oplus (b_0 \oplus 1).b_1)$$

We can find a_2 using only output bits i.e. we have found the inverse of χ. Similarly we can write the following equations(which form the inverse of χ):

$$a_0 = b_0 \oplus (b_1 \oplus 1).(b_2 \oplus (b_3 \oplus 1).b_4) \tag{11}$$
$$a_1 = b_1 \oplus (b_2 \oplus 1).(b_3 \oplus (b_4 \oplus 1).b_0) \tag{12}$$
$$a_2 = b_2 \oplus (b_3 \oplus 1).(b_4 \oplus (b_0 \oplus 1).b_1) \tag{13}$$
$$a_3 = b_3 \oplus (b_4 \oplus 1).(b_0 \oplus (b_1 \oplus 1).b_2) \tag{14}$$
$$a_4 = b_4 \oplus (b_0 \oplus 1).(b_1 \oplus (b_2 \oplus 1).b_3) \tag{15}$$

1.2. Inverse of θ

To calculate the inverse of the theta function, we first switch to a polynomial notation for state. We can represent the state as a polynomial in x, y, z with binary coefficients where the coefficient of the term $x^i y^j z^k$ is equal to the value of $a[i][j][k]$.
In this setting, a translation by t_x, t_y and t_z can be represented as a multiplication by $x^{t_x} y^{t_y} z^{t_z}$ modulo $1 + x^5$, $1 + y^5$ and $1 + z^8$(for $w = 8$). Hence the state is an element of a polynomial quotient ring defined by the polynomial ring over $GF(2)[x, y, z]$ modulo the ideal generated by $\langle 1 + x^5, 1 + y^5, 1 + z^8 \rangle$.

When the state is represented by a polynomial, the mapping θ can be expressed as the multiplication (in the quotient ring defined above) by the following polynomial :

$$1 + \bar{y}(x + x^4 z) \text{ where } \bar{y} = \sum_{i=0}^{4} y^i = \frac{1 + y^5}{1 + y}$$

The inverse of θ will be multiplication by polynomial which is inverse of above mentioned polynomial. First we assume the inverse is of the form $1 + \bar{y}Q$ where Q is a polynomial in x and y:

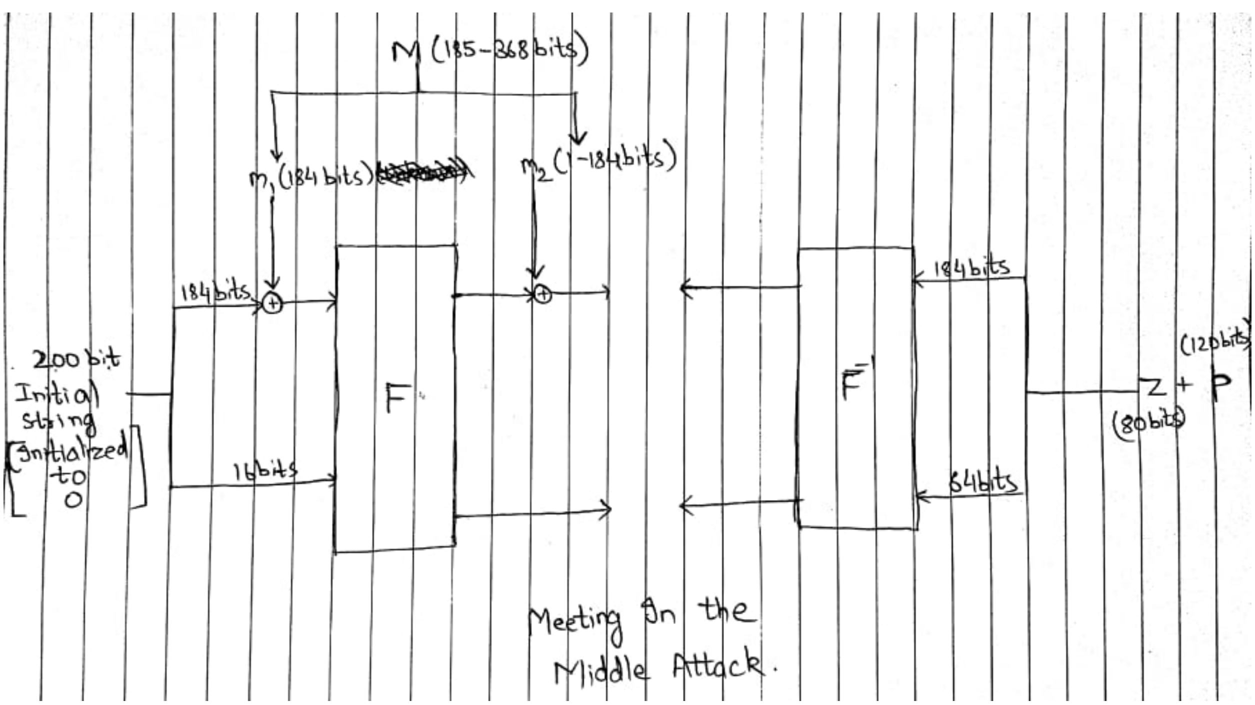
$$(1 + \bar{y}(x + x^4 z)) * (1 + \bar{y}Q) = 1 \bmod \langle 1 + x^5, 1 + y^5, 1 + z^8 \rangle$$

We can simply using SAGE to calculate Q and hence the inverse of theta function.

2. Attacks

2.1. Pre-Image Attack

For an output of 80 bits, we can append it with random bits and then invert it using the inverse of the function F to get an input(The function F can be inverted as all the four mappings θ, π, ρ and χ are invertible which results in R to be invertible). Now in case of the pre-image, the last 16 bits always have to be 0. So we cannot call this the pre-image as it may of may not have its last 16 bits as zero.
So, we formulate a Meet in the Middle Attack as a workaround.
Consider a message M having two blocks m_1 and m_2 with m_1 having length 184 bits and m_2 having length ranging from 1 bit to 184 bits. We will use the block m_1 to generate the last 16 bits of the 200 bit state and m_2 to modify the output $F(m_1)$ so that after the XOR, it matches with the input of the second round.
Now we generate random bits for m_1 and see the output $F(m_1)$. We create a set S_1 of the values of m_1 such that S_1 maps to a sufficiently large subset of all possible values of the last 16 bits of $F(m_1)$.
Now, from the other end, we append the known 80 bits(say Z) with random 120 bits(say p) and pass it through F^{-1} to get the input of the second round.
Observe that we can adjust m_2 to match the first 184 bits of $F(m_1)$ with the first 184 bits of the input to the second round, but we can't modify the last 16 bits.
So now we create a set S_2 by iterating the 120 bits (p) consisting of inputs to the second round that map to Z.
We now have two sets S_1 and S_2. If we are able to find an element in S_1, whose corresponding output in the first round has a 16 bit suffix that matches with the last 16 bits of an element in S_2, we have found a pre-image.
This collision can occur with a probability of 50% if the size of both the sets is around 2^8. Also we can find all the possible 2^{16} 16 bit suffixes with very high probability by iterating over $2^{16} \log 2^{16}$ number of random inputs(Assuming that if the input to F is random, then all the output bits are also random).



2.2. Collision Attack

We are given $F = R \circ R$. The collision attack we describe is independent of F. Let c_1 and c_2 be the output of m_1 and m_2 (each 184 bits) on operating F. If we can find m_1 and m_2 such that c_1 and c_2 (each 200 bits) have same last 16 bits then we can find M_1 and M_2 such that their hash is same.

Let u_1 and u_2 be the first 184 bits of c_1 and c_2. For any 184 bit string a, we can find r_1 and r_2 (each 184 bits) such that

$$r_1 = a \oplus u_1 \tag{16}$$
$$r_2 = a \oplus u_2 \tag{17}$$

Now consider $M_1 = m_1||r_1$ and $M_2 = m_2||r_2$ (|| means concatenation). M_1 and M_2 on input to F respectively give u_1 and u_2 first and then $u_1 \oplus r_1$ and $u_2 \oplus r_2$ are fed to F as input.
Since from (16) and (17) we have

$$u_1 \oplus r_1 = u_2 \oplus r_2 = a$$

Therefore the input becomes same and hence the output of F also becomes same. So we have found M_1 and M_2 with same hash and hence a collision attack.

We need to check at max $2^{16} + 1$ different random 184 bit strings to get two strings with same last 16 bits on feeding to F. These two strings will be our m_1 and m_2 in the attack. By the Birthday Paradox, we can get m_1 and m_2 in 2^8 different inputs with a very high probability.

2.3. Second Pre-image Attack

We have $F = R \circ R$ and let $H(m)$ denote the first 80 bits of $F(m)$ that form hash of m. We are given m and $H(m)$ and we have to find another m_2 such that $H(m_2) = H(m)$.
Let m' be a message such that last 16 bits of $F(m')$ are all zero. If we define a as follows:

$$a = F(m')[0:183] \oplus m$$

Now if we consider $m_2 = m'||a$, then we have:

$$F(m')[184:199] = 0000...0 \tag{18}$$
$$H(m_2) = F((F(m')[0:183] \oplus a)||F(m')[184:199])[0:79] \tag{19}$$
$$H(m_2) = F(m||000..0)[0:79] \tag{20}$$
$$H(m_2) = H(m) \tag{21}$$

Hence if found a m' that satisfies the above mentioned condition, we can find a second pre image for any m.

References

[1] SHA-3 Standard: Permutation-Based Hash and Extendable Output Functions
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

[2] KECCAK Sponge Function Family Main Document
https://keccak.team/obsolete/Keccak-main-1.1.pdf