JOURNAL OF THEORETICAL PHYSICS & CRYPTOGRAPHY

www.IJTPC.org/Publications

Improved form of Tillich-Zemor Hash Function

[1]Joju K.T., [2]Sr.Lilly P.L

[1]Department of Mathematics,Prajyoti Niketan College,Pudukad,Kerala, India.

[2]Department of Mathematics,St.Joseph'sCollege,Irinjalakuda,Kerala,India.

ABSTRACT

At CRYPTO ,94 Tillich and Zemor proposed a family of hash function based on computing a suitable matrix product in groups of the form $SL_2(F_2n)$. But Markus Grassl ,Ivana Illich ,Spyros Magliveras and Rainer Steinwadt found collision for the same between palindrome bit strings of length 2n+2. Christophe Petit, Jean-Jaques Quisquater found the second preimage and preimage for the same. We improve the hash function by using different generators and find collision, second preimage and preimage for the construction. In order to overcome this vulnerability we reinforce the hash function with key. Hence we get a secured hash function

Keywords

. Collision, Group, Hash function, Irreducible polynomial, Preimage, Second preimage.

2010 Mathematics subject classification: 94AXX ; 94A60; 11771; 14G50; 68P25; 81P94

1. INTRODUCTION

1.1 Cryptographic Hash Functions and MACs

Hash functions [1] are functions that compress an input of arbitrary length into fixed number of output bits, the hash result. If such a function satisfies additional requirements it can be used for cryptographic applications, for example to protect the authenticity of messages sent over an insecure channel. The basic idea is that the hash result provides a unique imprint of a message, and that the protection of a short imprint is easier than the protection of message itself. Related to hash functions are message authentication codes (MACs). These are also functions that compress an input of arbitrary length into a fixed number of output bits, but the computation depends on a secondary input of fixed length, the key. Therefore MACs are also referred to as keyed hash functions. In practical applications the key on which the computation of a MAC depends is kept secret between two communicating parties.

For an (unkeyed) hash function, the requirement that the hash result serves as a unique imprint of a message input implies that it should be infeasible to find colliding pairs of messages. In some applications however it may be sufficient that for any given hash result it is infeasible to find another message hashing to same result. Depending on these requirements Praneel [2] provides the following informal definitions for two different types of hash functions.

A one-way hash function is a function h that satisfies the following conditions:

1 .The input x can be of arbitrary length and the result h(x) has a fixed length of n bits.

2. Given h and an input x, the computation of h(x) must be easy.

3. The function must be one-way in the sense that given a y in the image of h , it is hard to find a message x such that h(x) = y (preimage-resistance), and given x and h(x) it is hard to find a message x' ≠ x such that h(x') = h(x)(second preimage-resistance).

A collision-resistant hash function is a function h that satisfies the following conditions:

1. The input x can be of arbitrary length and the result h(x) has a fixed length of n bits.

2. Given h and an input x, the computation of h(x) must be easy.

3. The function must be collision-resistant: this means that it is hard to find two distinct messages that hash to the same result(i.e., find x and x' with $x \neq x'$ such that $h(x) = h(x')$).

Message Authentication Codes

For a message authentication code, the computation depends on a secondary input, the secret key. The main idea is that an adversary without knowledge of this key should be unable to forge the MAC result for any new message, even when many previous messages and their corresponding MAC results are known. The following informal definition was given by Praneel[2]. A message authentication code or MAC is a function h satisfies the following conditions:

1.The input x can be of arbitrary length and the result h(K,x) has a fixed length of n bits. The function has a secondary input the key K, with a fixed length of k bits.

2.Given h, K and an input x, the computation of h (K, x) must be easy.

3.Given a message x (with unknown K), it must be hard to determine h(K,x).

4.Even when a large set of pairs $\{x_i, h(K, x_i)\}$ is known, it is hard to determine the key K or to compute h(K, x') for any mew message $x' \neq x_i$

Definition 1.1.1 A hash function h: D → R where the domain D = {0,1}*, and the range $R = \{0,1\}^n$ for some n ≥ 1.

Definition 1.1..2 A MAC is a function h: K X M→ R where the key space $K = \{0,1\}^k$, the message space $M = \{0,1\}^*$, and the range $R = \{0,1\}^n$ for k,n ≥ 1

Since its introduction at CRYPTO'94 the Tillich-Zemor hash function[3] has kept on appealing cryptographers by its originality, its elegance, its simplicity and its security[4]. The function computation can be parallelized and even the serial version is quite efficient as it only requires XOR, SHIFT and TEST operations. Uniform distribution of the outputs follows from a graph theoretical interpretation of the hash computation.

In 2009, Grassl et al [5] found collisions for the construction. In 2010Christophe Petit and Jean-Jacques Quisquater [6] found the preimage and second preimage for the same. In 2012 we, Joju K.T and P.L.Lilly [7,8,9] constructed a hash function using new generators for Tillich – Zemor hash function and found collision and preimages for the same. Further we [10,11,12] constructed the keyed versions of the hash functions, they were still unbroken.

In this paper we change the generators and prove that the hash function is vulnerable. In order to get a secured hash function we reinforce the hash function with key. We claim that it will resist palindrome collision as a consequence it will be second preimage resistant and preimage resistant.

This paper is organized as follows:

The Tillich-Zemor hash function and its palindrome collision is recalled in section 2. In section.3 we present the new hash function and verify that the keyed hash function resists the palindrome collision, second preimage resistant and preimage resistant.

2.1 Tillich-Zemor Hash function

Let n be a positive integer and let p(x) be an irreducible polynomial of degree n over the field $F_2[3]$. Let A_0 and A_1 be the following two matrices :

$A_0 = \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}$, $A_1 = \begin{pmatrix} x & x+1 \\ 1 & 0 \end{pmatrix}$,that have determinant 1. We call these matrices the generators of the Tillich-Zemor hash function. Let $v = b_1 \ldots . b_m \in \{0,1\}^*$, be the bitstring representation of a message . The Tillich-Zemor hash value of v is defined as:

$H(b_1 \ldots . b_m) = A_{b_1} \ldots\ldots A_{b_m} \bmod p(x)$

Let $K = F_2[x]/(p(x)) \approx F_2^n$. The image of the Tillich-Zemor hash function are the matrices of the group SL_2 (K),that is the group of matrices with elements in K and determinant 1.

Let h $(b_1 \ldots . b_m) = A_{b_1} \ldots\ldots A_{b_m}$ be the Tillich-Zemor hash function without modular reduction. i.e h: $\{0,1\}^* \rightarrow SL_2(F_2[x])$.

2.2.Palindrome Collision

If $v = b_1 \ldots . b_m \in M$ is a bitstring of length m; we denote $v^r = b_m \ldots\ldots b_1$, the reversal of v, ie the reflection of v which interchanges b_1 with b_m , b_2 with b_{m-1},etc. Bitstring $v \in M$ satisfying $v = v^r$ are known as palindrome. In order to have the palindrome collision we will make the following change in the generators.

Let $B_0 = A_0^{-1} A_0 A_0 = A_0 = \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}$ and $B_1 = A_0^{-1} A_1 A_0 = \begin{pmatrix} x+1 & 1 \\ 1 & 0 \end{pmatrix}$. Define the hash functions H' and h' with new generators as follows:

$H'(b_1 \ldots . b_m) = B_{b_1} \ldots\ldots B_{b_m} \bmod p(x)$ and $h'(b_1 \ldots . b_m) = B_{b_1} \ldots\ldots B_{b_m}$.Then we have the following proposition[5].

Proposition 1. Let v , v' ∈ M. Then H(v) = H(v') iff H'(v) = H'(v').

That is , collision for H and H' are equivalent.

In [5] Markus Grassl ,Ivana Illich ,Spyros Magliveras and Rainer Steinwadt observed the following property of palindrome messages.

Proposition 2. Let v be a palindrome of even length say v = $b_m \ldots b_1 b_1 \ldots b_m$. Let $a_0 \ldots a_m$ be the following polynomials

$$a_i = \begin{cases} 1, & if\ i = 0 \\ x + b_1 + 1, & if\ i = 1 \\ (x + b_i)a_{i-1} + a_{i-2} & if\ 1 < i \leq m \end{cases}$$

Then h'(v) = $\begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix}$ for a = a_m, d = a_{m-1} and for some b ∈ $F_2[x]$

Moreover, h'(0v0)+ h'(1v1) = $\begin{pmatrix} a^2 & a^2 \\ a^2 & 0 \end{pmatrix}$.

From proposition 2 we see that the square root of the upper left entries of h'(b_1b_1);h'($b_2b_1b_1b_2$);h'($b_3b_2b_1b_1b_2b_3$); etc , satisfying a Euclidean algorithm sequence (in reverse order) where each quotient is either x or x+1. These sequences are often called maximal length sequences for the Euclidean algorithm or maximal length Euclidean sequences. Mesirov and sweet[13] showed that, when a ∈ $F_2[x]$ is an irreducible, there exists exactly two polynomials d such that a,d are the first terms of a maximal length Euclidean sequences. In their collision algorithm[5] they apply Mesirov and Sweet's algorithm to the irreducible polynomial a = p(x) .

Proposition 3.(Mesirov and Sweet)[13] Given any irreducible polynomial p of degree n over F_2, there is a sequence of polynomials p_n, p_{n-1}, with $p_n = p$, and ,$p_0 = 1$ and additionally the degree of p_i is equal to i and $p_i \equiv p_{i-2} \bmod p_{i-1}$.

Note that once we know a polynomial q = p_{n-1} as mentioned in proposition 3 which matches our given polynomial $p_n = p$, the Euclidean algorithm will uniquely compute the sequence $p_n, p_{n-1}, \ldots, p_1, p_0 = 1$.

The quotients $x+\beta_i (i = 1,\ldots.n)$ occurring in Euclid's algorithm allow us to derive the bits b_i of the palindrome in proposition 2 .We have $p_1 = x+b_1+1$ and therefore $b_1 = \beta_1+1$, while $b_i = \beta_i$ for some i >1.That is the bit β_1 has to be inverted. Thus the desired collision will be

$H'(0\beta_n \ldots \beta_1^{-1}\beta_1^{-1} \ldots \beta_n 0) = H'(1\beta_n \ldots \beta_1^{-1}\beta_1^{-1} \ldots \beta_n 1)$ where

β_1^{-1}indicates the inversion of β_1.

2.3.To find the maximal length Euclidean sequence:

1.Construct a matrix A∈ $F_2^{(n+1)xn}$ from the n+1 polynomials

$g_0 \equiv x^0 \bmod p(x)$,

$g_i \equiv x^{i-1} + x^{2i-1} + x^{2i} \bmod p(x)$ for i= 1,2,.....,n

Placing in the i^{th} row of A the coefficients

$a_{i,0}, a_{i,1}, \ldots\ldots a_{i,n-1}$ of the polynomial

$g_i = a_{i,0} + a_{i,1} x + \ldots\ldots a_{i,n-1} x^{n-1}$.

2. Solve the linear system Au^t = (10.....01) where u = ($u_1 \ldots\ldots u_n$).

3.Compute q(x) by multiplying p(x) by $\sum_{i=1}^{n} u_i x^{-i}$ and taking only the

non negative powers of x.

2.4.To find Collision for specified parameters

For each choice of F_{2^n} = $F_2[x]/(p(x))$ we obtain two bitstrings v_1 , $v_2 \in \{0,1\}^* = M$ with $H'(0v_i v_i^r 0) = H'(1v_i v_i^r 1)$ for i= 1,2. ie , we obtain two collisions of bitstrings of length 2n+2. v_2 can be obtained by reversing v_1 followed by inverting the first and last bit. By proposition .1 we have $H(0v_i v_i^r 0) = H(1v_i v_i^r 1)$.From[5] wehave

Collision for $SL_2(F_2[X]/(x^{127}+x+1))$

By collision algorithm we have $H(0v_1v_1^r0)$ =

Column 1

x

1

Column 2

$1+x^2+x^3+x^{64}+x^{65}+x^{96}+x^{97}+x^{112}+x^{113}+x^{120}+x^{121}+x^{124}+x^{125}+x^{126}$

$x+x^2+x^{63}+x^{64}+x^{95}+x^{96}+x^{111}+x^{112}+x^{119}+x^{120}+x^{123}+x^{124}+x^{125}$

= $H(1v_1v_1^r1)$ and $H(0v_2v_2^r0)$ =

Column 1

x

1

Column 2

$1+x+x^2+x^{64}+x^{65}+x^{96}+x^{97}+x^{112}+x^{113}+x^{120}+x^{121}+x^{124}+x^{125}+x^{126}$

$1+x+x^{63}+x^{64}+x^{95}+x^{96}+x^{111}+x^{112}+x^{119}+x^{120}+x^{123}+x^{124}+x^{125}$

$=H(1v_2v_2^r1)$ where

v_1 = 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 000

v_2 = 100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 .

Calculations are done by using 'SCILAB' (An open source software for numerical computations) or[14] .

3.New Hash Function

Let $C_0 = A_0^T = \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}$ and $C_1 = A_1^T = \begin{pmatrix} x & 1 \\ x+1 & 1 \end{pmatrix}$ and define the new hash function as follows:

$H_1(b_1 \ldots . b_m) = C_{b_1} \ldots\ldots C_{b_m}$ modp(x). Now we try to find the collision, second preimage and preimage. For that we change the generators as follows:

Let $D_0 = C_0$ and $D_1 = C_0C_1C_0^{-1} = \begin{pmatrix} x+1 & 1 \\ 1 & 0 \end{pmatrix}$ which are the generators B_0 and B_1.

Suppose the hash function H_2 is defined as follows:

$H_2(b_1 \ldots . b_m) = D_{b_1} \ldots\ldots D_{b_m}$ modp(x). Hence the hash function defined by the generators D_0 and D_1 is the same as H'. We found collision for the hash function H. Therefore we can find collision for the hash function H_2. By the following proposition we can prove that H_1 is not collision resistant.

Proposition 3.1 . Let v , v' ∈ M. Then $H_1(v) = H_1(v')$ if and only if $H_2(v) = H_2(v')$.

Proof: Let $v = b_1 \ldots b_m$ and v = $b_1 \ldots b_r$.

H1(v) = $H_1(v')$ iff $C_{b_1} \ldots\ldots C_{b_m}$ modp(x) = $C_{b_1'} \ldots C_{b_r'}$ modp(x) iff

$C_0 C_{b_1} C_0^{-1} \ldots C_0 C_{b_m} C_0^{-1}$ modp(x) = $C_0 C_{b_1'} C_0^{-1} \ldots C_0 C_{b_r'} C_0^{-1}$ modp(x) iif

$D_{b_1} \ldots . D_{b_m}$ modp(x) = $D_{b_1'} \ldots . D_{b_r'}$ modp(x) iff $H_2(v) = H_2(v')$.

Hence collision in H_1 and H_2 are equivalent. We have the following collision

$H_1(0v_1v_1^r0)$ =

column 1

x

$1+x^2+x^3+x^{64}+x^{65}+x^{96}+x^{97}+x^{112}+x^{113}+x^{120}+x^{121}+x^{124}+x^{125}+x^{126}$

column 2

1

$x+x^2+x^{63}+x^{64}+x^{95}+x^{96}+x^{111}+x^{112}+x^{119}+x^{120}+x^{123}+x^{124}+x^{125}$

$=H_1(1v_1v_1^r1)$

Similarly we can verify that

$H_1(0v_2v_2^r0) = H_1(1v_2v_2^r1)$.

In order to get a secured hash function we reinforce this hash function with key and verify that it is collision resistant.

Let H_3 be the keyed hash function and it is defined as follows:

Let K = $\{(k_1 \ldots . k_n)$: $k_i = 0$ or $1\}$be the key space. Then K contains 2^n elements. We define the new hash function as follows:

$H_3(m_1 \ldots . m_n m_{n+1} \ldots .) = C_{m_1}^{k_1} \ldots\ldots C_{m_n}^{k_n} C_{m_{n+1}}^{k_1} \ldots \ldots$mod p(x)

Where $C_{m_i}^{k_i} = \begin{cases} I\,, if\ k_i = 0 \\ C_{m_i} if\ k_i = 1 \end{cases}$

If $k_i = 1\ \forall$ i then $H_3 = H_1$.

We claim that this keyed hash function resists the palindrome collision. We show that for a particular key $(k_1k_2 \ldots\ldots k_{256})$ =

(1000110001100011000110001100011000110001100011000
1100011000110001000110001100011000110001100011000
1100011000110001100011000110001100011000110001100
0011000110001100011000110001100011000110001100011
0001100011000110001100011000110001100011000110001
10001100011) from K, the key collection, we have

$H_3(0v_1v_1^r0)$ =

column 1

$x^3+x^4+x^5+x^8+x^{13}+x^{15}+x^{16}+x^{21}+x^{24}+x^{27}+x^{28}+x^{31}+x^{32}+x^{35}+x^{36}+x^{39}+x^{40}+x^{45}+x^{47}+x^{48}+x^{51}+x^{52}+x^{63}+$

$x^{64}+x^{69}+x^{71}+x^{79}+x^{80}+x^{85}+x^{88}+x^{91}+x^{92}+x^{95}+x^{96}+x^{101}+x^{103}$

$1+x^3+x^4+x^5+x^6+x^7+x^{10}+x^{12}+x^{15}+x^{16}+x^{18}+x^{22}+x^{23}+x^{24}+x^{26}+x^{27}+x^{28}+x^{29}+x^{31}+x^{34}+x^{35}+x^{36}+x^{37}+x^{39}+x^{42}+x^{44}+x^{47}+x^{50}+x^{51}+x^{52}+x^{53}+x^{63}+x^{64}+x^{66}+x^{70}+x^{79}+x^{80}+x^{82}+x^{86}+x^{87}+x^{88}+x^{90}+x^{91}+x^{92}+x^{93}+x^{95}+x^{96}+x^{98}+x^{102}$

column 2

$1+x^3+x^6+x^7+x^8+x^9+x^{10}+x^{12}+x^{15}+x^{17}+x^{18}+x^{22}+x^{23}+x^{25}+x^{26}+x^{27}+x^{31}+x^{32}+x^{33}+x^{34}+x^{35}+x^{39}+x^{40}+x^{41}+x^{42}+x^{44}+x^{47}+x^{48}+x^{49}+x^{50}+x^{51}+x^{63}+x^{65}+x^{66}+x^{70}+x^{79}+x^{81}+x^{82}+x^{86}+x^{87}+x^{89}+x^{90}+x^{91}+x^{95}+x^{97}+x^{98}+x^{102}$

$x+x^4+x^5+x^8+x^{11}+x^{16}+x^{17}+x^{19}+x^{21}+x^{24}+x^{25}+x^{27}+x^{28}+x^{32}+x^{35}+x^{36}+x^{40}+x^{43}+x^{48}+x^{51}+x^{52}+x^{64}+x^{65}+$

$x^{67}+x^{69}+x^{80}+x^{81}+x^{83}+x^{85}+x^{88}+x^{89}+x^{91}+x^{92}+x^{96}+x^{97}+x^{99}+x^{101}$

$H_3(1v_1v_1^r1)$ =

column 1

$x^2+x^3+x^5+x^6+x^{10}+x^{11}+x^{12}+x^{13}+x^{15}+x^{19}+x^{20}+x^{21}+x^{22}+x^{28}+x^{31}+x^{34}+x^{36}+x^{39}+x^{42}+x^{43}+x^{44}+x^{45}+x^{47}+$

$x^{50}+x^{52}+x^{63}+x^{67}+x^{68}+x^{69}+x^{70}+x^{71}+x^{79}+x^{83}+x^{84}+x^{85}+x^{86}+x^{92}+x^{95}+x^{99}+x^{100}+x^{101}+x^{102}+x^{103}$

$1+x+x^2+x^3+x^5+x^7+x^8+x^{10}+x^{13}+x^{17}+x^{19}+x^{20}+x^{23}+x^{25}+x^{27}+x^{28}+x^{29}+x^{32}+x^{34}+x^{36}+x^{37}+x^{40}+x^{42}+x^{45}+x^{48}+x^{50}+x^{52}+x^{53}+x^{65}+x^{67}+x^{68}+x^{71}+x^{81}+x^{83}+x^{84}+x^{87}+x^{89}+x^{91}+x^{92}+x^{93}+x^{97}+x^{99}+x^{100}+x^{103}$

column 2

$1+x+x^5+x^6+x^8+x^{11}+x^{12}+x^{17}+x^{19}+x^{21}+x^{22}+x^{25}+x^{27}+x^{32}+x^{35}+x^{40}+x^{43}+x^{44}+x^{48}+x^{51}+x^{65}+x^{67}+x^{69}+x^{70}+x^{81}+x^{83}+x^{85}+x^{86}+x^{89}+x^{91}+x^{97}+x^{99}+x^{101}+x^{102}$

$x^2+x^3+x^6+x^7+x^9+x^{10}+x^{12}+x^{15}+x^{17}+x^{20}+x^{22}+x^{23}+x^{25}+x^{27}+x^{28}+x^{31}+x^{33}+x^{34}+x^{35}+x^{36}+x^{39}+x^{41}+x^{42}+x^{44}+x^{47}+x^{49}+x^{50}+x^{51}+x^{52}+x^{63}+x^{65}+x^{68}+x^{70}+x^{79}+x^{81}+x^{84}+x^{86}+x^{87}+x^{89}+x^{91}+x^{92}+x^{95}+x^{97}+x^{100}+x^{102}$.

Clearly $H_3(0v_1v_1^r0) \neq H_3(1v_1v_1^r1)$. Hence H_3 resists palindrome collision. Hence we get a collision resistant hash function.

Next we show that H_3 is second preimage resistant. Since H' and H_2 are one at the same from[6] we have

Proposition:3.2 Let $\begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix}$ = $H_2(v)$ with a≠ 0 be the modified Tillich-Zemor hash value of some message v ∈ {0,1}*. Then

$H_1(0v0v) = H_1(1v1v) = H_1(v0v0) = H_1(v1v1) = I = H_1()$.

From[6] "A Can we also compute preimages? A toy example." Let $p(x) = x^{11}+x^2+1$ be the irreducible polynomial. The representation of the sentence "Grassl et al. have shown how to find collisions. But can we also compute preimages?" in binary it is V_1 = [0,1,0,0,0,1,1,1, 0,1,1,1,0,0,1,0, 0,1,1,0,0,0,0,1 ,0,1,1,1,0,0,1,1, 0,1,1,1,0,0,1,1, 0,1,1,0,1,1,0,0, 0,0,1,0,0,0,0,0, 0,1,1,0,0,1,0,1, 0,1,1,1,0,1,0,0, 0,0,1,0,0,0,0,0, 0,1,1,0,0,0,0,1, 0,1,1,0,1,1,0,0, 0,0,1,0,1,1,1,0 ,0,0,1,0,0,0,0,0, 0,1,1,0,1,0,0,0, 0,1,1,0,0,0,0,1, 0,1,1,1,0,1,1,0, 0,1,1,0,0,1,0,1 ,0,0,1,0,0,0,0,0,0,1,1,1,0,0,1,1, 0,1,1,0,1,0,0,0, 0,1,1,0,1,1,1,1, 0,1,1,1,0,1,1,1, 0,1,1,0,1,1,1,0, 0,0,1,0,0,0,0,0 ,0,1,1,0,1,0,0,0, 0,1,1,0,1,1,1,1, 0,1,1,1,0,1,1,1, 0,0,1,0,0,0,0,0, 0,1,1,1,0,1,0,0, 0,1,1,0,1,1,1,1, 0,0,1,0,0,0,0,0, 0,1,1,0,1,1,0,0, 0,1,1,0,1,1,1,0, 0,1,1,0,0,1,0,0, 0,0,1,0,0,0,0,0, 0,1,1,0,0,0,1,1, 0,1,1,0,1,1,1,0, 0,1,1,0,1,0,0,1, 0,1,1,1,0,0,1,1, 0,1,1,0,1,0,0,1, 0,1,1,0,1,1,1,1, 0,1,1,0,1,1,1,0, 0,1,1,1,0,0,1,1, 0,0,1,0,1,1,1,0, 0,0,1,0,0,0,0,0, 0,1,0,0,0,0,1,0, 0,1,1,1,0,1,0,1 ,0,1,1,1,0,1,0,0, 0,0,1,0,0,0,0,0, 0,1,1,0,0,0,1,1, 0,1,1,0,0,0,0,1, 0,1,1,0,1,1,1,0, 0,0,1,0,0,0,0,0, 0,1,1,1,0,1,1,1, 0,1,1,0,0,1,0,1, 0,0,1,0,0,0,0,0, 0,1,1,0,0,0,0,1, 0,1,1,0,1,1,0,0, 0,1,1,1,0,0,1,1, 0,1,1,0,1,1,1,1, 0,0,1,0,0,0,0,0, 0,1,1,0,0,0,1,1, 0,1,1,0,1,1,1,1, 0,1,1,0,1,1,0,1, 0,1,1,1,0,0,0,0, 0,1,1,1,0,1,0,1, 0,1,1,1,0,1,0,0, 0,1,1,0,0,1,0,1, 0,0,1,0,0,0,0,0, 0,1,1,1,0,0,0,0, 0,1,1,1,0,0,1,0, 0,1,1,0,0,1,0,1, 0,1,1,0,1,0,0,1, 0,1,1,0,1,1,0,1, 0,1,1,0,0,0,0,1, 0,1,1,0,0,1,1,1, 0,1,1,0,0,1,0,1, 0,1,1,1,0,0,1,1, 0,0,1,1,1,1,1,1]. The hash (H_1) value of this message for the polynomial p(x) is

Column 1

$1+x^3+x^4+x^8+x^9$

$1+x^3+x^7+x^8+x^9+x^{10}$

Column 2

$x^2+x^3+x^5+x^6+x^9$

$x+x^3+x^4+x^6+x^7+x^8$

The Mesirov and Sweet algorithm applied to p(x) gives

$q(x) = x^{10}+x^8+x^7+x^6+x^5+x^2+x+1$ and the palindrome message

v =01000011011111011000010 then by proposition 3.2 we have $H_1(0v0v) = H_1(1v1v) = H_1(v0v0) = H_1(v1v1) = I = H_1()$.
For example

$H_1(0,1,0,0,0,0,1,1,0,1,1,1,1,0,1,1,0,0,0,0,1,0,0,0,1,0,0,0,0,1,1,$
$0,1,1,1,1,0,1,1,0,0,0,0,1,0,0)= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

But

$H_3(0,1,0,0,0,0,1,1,0,1,1,1,1,0,1,1,0,0,0,0,1,0,0,0,1,0,0,0,0,1,1,$
$0,1,1,1,1,0,1,1,0,0,0,0,1,0,0) =$

Column 1

$1+x+x^2+x^4$

$1+x+x^3+x^6+x^7+x^9+x^{10}$

Column 2

$x^4+x^6+x^7+x^9$

$x^3+x^4+x^5+x^7+x^8+x^{10}$

$\neq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Similarly we can verify the other combinations. Hence H_3 is second preimage resistant.

Now we prove that the keyed Tillich-Zemor hash function H_3 is preimage resistant. In[6] Christophe Petit, Jean-Jaques Quisquater found preimage for the Tillich-Zemor hash function. They use two precomputing algorithms to find the preimages. In the toy example [6] by using the first precomputing algorithm they have following preimage for H_1: $V_{2=}$[1,0,1,1,1,1,1,0,1,0,0,1,0,1,1,1,0,1,1,0,1,1,0,1,1,0,1,1,1,0,1, 0,0,1,0,1,1,1,1,1,0,1,0,0,0,1,0,1,1,1,1,1,0,1,0,0,1,0,1,1,1,0,1,1, 0,1,1,0,1,1,0,1,1,1,0,1,0,0,1,0,1,1,1,1,1,1,0,1,0,1,1,1,1,1,0,1,1,0, 0,1,0,1,0,1,1,1,1,0,0,0,0,0,0,1,1,1,1,0,1,0,1,0,0,1,1,0,1,1,1,1,1, 0,0,0,0,0,1,1,0,1,1,0,1,0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1,1,1,0,1, 0,1,1,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,1,0,0,1,0,0,0,1,0,0,1,0,0,1,0, 1,1,0,1,0,0,1,0,0,1,0,0,0,1,0,0,1,0,1,1,0,0,0,1,0,1,1,1,1,1,1,0,1,0, 0,1,0,1,1,1,0,1,1,0,1,1,0,1,1,0,1,1,1,0,1,0,0,1,0,1,1,1,1,1,0,1,0, 0,1,0,0,1,0,1,1,1,1,1,0,1,0,1,1,1,1,1,0,1,1,0,1,1,1,1,1,1,0,1,0,1,1, 1,1,1,0,1,0,0,1,0,0,0,0,1,0,1,0,0,1,1,1,1,1,0,1,1,1,1,0,0,1,0,0,0,0,0, 0,1,0,0,1,1,1,0,1,1,1,1,1,0,0,1,0,1,0,0,0,0,0,1,0,1,1,1,0,1,0,1,1,1,1, 0,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,0,1,1,1,1,0,1,0,1,1,1,0,1,0,1,0,0,1,0,0, 0,0,1,0,1,0,0,1,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,1,0,0,1,0, 1,0,0,0,0,1,0,1,0,1,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,1,0,1,1,1,1,1,0,1,0, 0,0,0,0,1,1,1,1,1,1,0,0,0,0,1,0,1,1,0]. Since the hash value of this message is

Column 1

$1+x^3+x^4+x^8+x^9$

$1+x^3+x^7+x^8+x^9+x^{10}$

Column 2

$X^2+x^3+x^5+x^6+x^9$

$X+x^3+x^4+x^6+x^7+x^8$

$H_1(V_1) = H_1(V_2)$

Whereas $H_3(V_1)$ =

Column 1

$x^2+x^7+x^{10}$

$1+x^3+x^6+x^7+x^8$

Column 2

$X^2+x^5+x^6+x^7+x^8+x^9+x^{10}$

$1+x^2+x^3+x^5+x^9+x^{10}$

and $H_3(V_2)$ =

Column 1

$1+x^2+x^3+x^4+x^5+x^6+x^6+x^7+x^8$

$x+x^2+x^7+x^9+x^{10}$

Column 2

$1+x+x^4+x^5+x^6+x^8$

$1+x^3+x^4+x^5+x^7+x^{10}$

$H_3(V_1) \neq H_3(V_2)$. Thus H_3 is preimage resistant.

Conclusion

We know that Tillich-Zemor hash function is vulnerable. So improvement in it is essential. In[10,11,12] we found reinforced version of Tillich-Zemor hash function. Here we the found the secured form of the variant of the same. Which is the hash function H_3. Thus we get a secured hash function H_3.

Acknowledgement

I am so much grateful to University Grants Commission (Government of India) for giving me the opportunity to do the research under the faculty improvement program (FIP).

References

[1]. Bart Van Rompay., 2004. Analysis and Design of Cryptographichash Functions, MAC algorithms and Block Ciphers,Doctoral Dissertation,KU Leuven.

[2]. B. Praneel.,1993. Analysis and Design of Cryptographic Hash Functions, Doctoral dissertation K.U Leuven Jan.

[3]. J.P. Tillich ., G. Zemor.,1994. Hashing with SL_2, Advances in Cryptology Lecture Notes in Computer Science, 839 40-49

[4]. Cristophe Petit., Jean-Jacques Quisquater., Jean-Pierre Tillich., Gilles Zemor.,2009. Hard and easy Components of Collision Search in the Zemor-Tillich Hash Function: new Attacks and Reduced Variants with Equivalent Security, In M. Fischlin, editor, CT-RSA, Lecture Notes in Computer Science, 5473 182-194 .

[5]. Markus Grassl., Ivana Ilic., Spyros Magliveras., Rainer Steinwandt., 2009. Cryptanalysis of the Tillich-Zemor hash function, Journal of Cryptology, 24 (1) 148-156.

[6]. Cristophe Petit., Jean-Jacques Quisquater., 2011. Preimages for the Tillich-Zemor hash function, Proceedings of the 17 th International Conference on Selected Areas in Cryptography 282-301.

[7]. K.T .Joju ., P.L. Lilly.,2012. Tillich-Zemor Hash Function with New Generators and Analysis, Research Journal of Pure Algebra, 2(11) 338-343.

[8]. K. T Joju. , P.L Lilly. , 2013. Preimage of Tillich–Zemor Hash Function with New Generators, International Journal of Applied Mathematical Sciences. 7, 4237-4248.

[9]. K. T Joju. , P.L Lilly. ,2012. Tillich-Zemor Hash with new Generators and Collision Analysis, Proceedings of NCMSC-2012 . 104-109.

[10]. K. T Joju ., P.L Lilly ., 2013. A Keyed Hash Function, IOSR-Journal of Mathematics, *(4) 47-55.*

[11]. K. T Joju. , P.L Lilly ., 2013. Keyed Tillich - Zemor Hash Function, Research Journal of Pure Algebra ,3(1) 24-32.

[12]. K. T Joju. , P.L Lilly. ,2013. Alternate form of Tillich-Zemor hash function which resist second preimage, International J. of Math. Sci. & Engg. Appls. 7(2) 79-98.

[13]. Jill P. Mesirov., Melvin M. Sweet., 1987. Continued Fraction Expansions of Rational Expressions with Irreducible Denominators in Characteristic 2, Journal of Number Theory. 27 144-148.

[14]. Wieb Bosma., John Cannon., Catherine Playoust.,1997. The Magma Algebra System I:The User Language, Journal of Sympolic Computation, 24 235-265.