Research Journal of Pure Algebra -2(11), 2012, Page: 338-343
Available online through www.rjpa.info ISSN 2248–9037

TILLICH-ZEMOR HASH FUNCTION WITH NEW GENERATORS AND ANALYSIS

Joju K.T[1*] & Lilly P. L[2]
[1]*Department of Mathematics, Prajyoti Niketan College, Pudukad, Kerala, India, pin-680301*
[2]*Department of Mathematics, St. Joseph's College, Irinjalakuda, Kerala, India, pin-680121*

(Received on: 30-06-12; Revised & Accepted on: 20-11-12)

ABSTRACT

At CRYPTO '94, Tillich and Zemor proposed a family of hash functions, based on computing a suitable matrix product in groups of the form $SL_2(F_2n)$. Markus Grassl, Ivana Illich, Spyros Magliveras and Rainer Steinwandt constructed a collision between palindrome bit strings of length 2n+2 and Christophe Petit, Jean-Jacques Quisquater found the second preimage for Tillich and Zemor's construction. In this paper we construct a hash function by using different matrices for the image of the bits 0 and 1 and found the collision and second preimage for the new construction.

Keywords: *Collision, Euclidean algorithm, Groups, Hash function, Irreducible polynomial, Palindrome, Preimage.*

1. INTRODUCTION

A cryptographic hash function can provide assurance of data integrity. A hash function is used to construct a short "finger print" of some data; if the data is altered, then the finger print will no longer be valid. Even if the data is stored in an insecure place, its integrity can be checked from time to time by recomputing the finger print and verifying that the finger print has not changed [3]

A hash family is a four –tuple (X, Y, K, H) where the following conditions are satisfied:
X is a set of possible messages
Y is a finite set of possible message digests
K, the key space, is a finite set of possible keys

For each k ∈ K, there is a hash function H_k ∈ H. Each H_k: X → Y

An unkeyed hash function is a function H: X→ Y . An unkeyed hash function is a hash family in which there is only one possible key.

Security of Hash Functions: [10]
The following three properties are essential for a secured hash function.

Preimage resistance: it should be computationally infeasible to find an input which hashes to a specified output.

Second preimage resistance: it should be computationally infeasible to find a second input that hashes to the same output of a specified input

Collision resistance: it should be computationally infeasible to find two different inputs that hash to the same output. Early suggestions (SHA family) did not really use any mathematical ideas apart from Merkle-Damgard [8] construction for producing collision resistant hash functions from collision resistant compression functions, the main idea was just to "create a mess" by using complex iterations. We have to admit that a"mess" might be good for hiding purposes, but only to some extent.

At CRYPTO'94, Tillich and Zemor [9] proposed a family of hash functions, based on computing a suitable matrix product in groups of the form SL_2 (F_2n). Tillich-Zemor suggested a mathematical hash function, which hash bit by bit. That is"0"bit is hashed to a particular 2x2 matrix $A_0$and the "1" bit is hashed to another 2x2 matrix A_1. For example 11000100 is hashed to the matrix $A_1^2A_0^3A_1A_0^2$.It is possible only when this pair of elements A_0, A_1 should be from an Algebraic structure. Tillich and Zemor use matrices A_0, A_1 from the group $SL_2(R)$ where R = $F_2[x]/ (q(x))$ [4]. Where F_2 is the field of two elements, $F_2[x]$ is the ring of polynomials over F_2 and (q(x)) is the ideal of $F_2[x]$ generated by an irreducible polynomial q(x) of degree n where n is a prime. For example

$q(x)=x^{167}+x^7+x^6+x^5+x^4+x+1$ [5].

***Corresponding author:** Joju K.T[1*]*
[1]Department of Mathematics, Prajyoti Niketan College, Pudukad, Kerala, India pin.680301

Thus $R=F_2[x]/(q(x))$ isomorphic to F_2^n the field with 2^n elements .The matrices A_0 and A_1 are the following:
$A_0 = \begin{pmatrix}\alpha & 1\\ 1 & 0\end{pmatrix}$ $A_1 = \begin{pmatrix}\alpha & \alpha+1\\ 1 & 1\end{pmatrix}$, where $\propto$ is the root of the irreducible polynomial q(x).

For the bitstring $v = b_1.....b_m \in V = \{0,1\}^*$, where $\{0,1\}^*$is the collection of bit strings of arbitrary length. The Tillich – Zemor hash function h' is defined as follows:

$h'(b_1.....b_m) = A_{b_1}.....A_{b_m}$.

In[6] Markus Grassl, Ivana Illich, Spyros Magliveras and Rainer Steinwandt constructed a collision between palindrome bit strings of length 2n+2 and in[2] Christophe Petit , Jean-Jacques Quisquater found the second preimage for Tillich and Zemor hash function.

2. HASH FUNCTION

2.1 New hash function.

Let B_0 and B_1 be the following matrices
$B_0=A_0^{-1}$ and $B_1=A_1^{-1}$ then $B_0 = \begin{pmatrix}0 & 1\\ 1 & \alpha\end{pmatrix}$ and $B_1 = \begin{pmatrix}1 & \alpha+1\\ 1 & \alpha\end{pmatrix}$.

For the bitstring $v = b_1 \dots . b_m \in V$ we define the new hash function h as follows:

$h(b_1....b_m) = B_{b_1}.....B_{b_m}$.

2.2 Palindrome Collisions

Let $v \in V$ and |v| denote the length of the bitstring v.If $v=b_1....b_m \in V$ is of length m, we denote $v^r = b_m.....b_1$ the reversal of v . In our attack we will make use of palindromes, that is, bitstrings $v \in V$ satisfying $v=v^r$.

In order to find the palindrome collision we use the matrices $C_0 = B_0$ and $C_1 = B_O B_1 B_0^{-1}$. That is
$C_0=\begin{pmatrix}0 & 1\\ 1 & \alpha\end{pmatrix}$ and $C_1=\begin{pmatrix}0 & 1\\ 1 & \alpha+1\end{pmatrix}$

We define $H(b_1....b_m) = C_{b_1}.....C_{b_m}$

Proposition 1. Let $v, v' \in V$. Then h (v) = h(v') if and only if H(v) = H(v').

Proof: Suppose $v= b_1.....b_m$, $v'=b_1'.....b_r'$ are bitstrings in V.

Then h(v)= h(v') if and only if

$B_{b_1} \dots \dots B_{b_m} = B_{b_1'} \dots \dots . B_{b_m'} \Leftrightarrow B_0 (B_{b_1} \dots \dots . B_{b_1}) B_0^{-1} = B_0 (B_{b_1'} \dots \dots . B_{b_m'}) B_0^{-1}$

$\Leftrightarrow B_0 B_{b_1} B_0^{-1} B_0 B_{b_2} B_0^{-1} .....B_0 B_{b_m} B_0^{-1} = B_0 B_{b_1'} B_0^{-1} B_0 B_{b_2'} B_0^{-1} .....B_0 B_{b_r'} B_0^{-1}$

$\Leftrightarrow C_{b_1}.....C_{b_m} = C_{b_1'}.....C_{b_r'}$

$\Leftrightarrow H(b_1.....b_m)=H(b_1'.....b_r')$ that is,

$\Leftrightarrow H(v)=H(v')$.

The above proposition says that collision in h and H are equivalent.

Now we work inside the group $SL_2(F_2[x])$ of unimodular matrices over the polynomial ring $F_2[x]$ rather than F_2^n. Let $D_0, D_1 \in SL_2(F_2[x])$ with polynomial entries as follows:

$D_0 = \begin{pmatrix}0 & 1\\ 1 & x\end{pmatrix}$, $D_1 = \begin{pmatrix}0 & 1\\ 1 & x+1\end{pmatrix}$ and

we define $H': V \to SL_2(F_2[x])$ by

$H'(b_1.....b_m) = D_{b_1}.....D_{b_m} \in SL_2(F_2[x])$.

That is H' is defined as H, except that $H'(v) \in SL_2(F_2[x])$.

We apply H' to a particular subset of elements of V, namely, the set of all palindromes in V.

Lemma 1. Let $v \in V$ be a palindrome and write $H'(v) = \begin{pmatrix}a & b\\ c & d\end{pmatrix}$. Then b=c and d has degree, deg d= |v| and we have max. (deg a, deg b)≤|v|.

Proof: The proof is by induction on the length |v| of V. For |v|≤1, the statement holds, as H′ (v) is the identity matrix or D_0 or D_1, all the three of which satisfy the property. For a palindrome w of length m, H′ (w) is of the form $D_\beta \begin{pmatrix} a & b \\ c & d \end{pmatrix} D_\beta$

where $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$= H′ (v) for the palindrome v of length m-2 and where $\beta \in \{0,1\}$.

Now $D_\beta \begin{pmatrix} a & b \\ c & d \end{pmatrix} D_\beta = \begin{pmatrix} d & c+d(x+\beta) \\ b+d(x+\beta) & dx^2+(b+c)x+\beta(d+b+c) \end{pmatrix}$

By induction b=c and the first part follows. The degree statement is also true.

Define $\rho : V \to F_2[x]^{2x2}$ is defined by

ρ(v) =H′ (0v0) +H′ (1v1).

We are interested in evaluating ρ modulo a given irreducible polynomial, because $\rho(v) \equiv \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ modq(x) if and only if H(0v0) = H(1v1) is indeed a collision in $SL_2(F_2[x]/(q(x)))=G$.

Proposition 2. If v ∈ V is a palindrome of length |v|, then $\rho(v) = \begin{pmatrix} 0 & d \\ d & d \end{pmatrix}$ where d ∈ F_2[x] has degree |v|. Moreover, d is the lower right entry of H′ (v).

Proof:

$\rho(v) = D_0 H'(v) D_0 + D_1 H'(v) D_1$

$= \begin{pmatrix} 0 & d \\ d & d+b+c \end{pmatrix} = \begin{pmatrix} 0 & d \\ d & d \end{pmatrix}$,

Since b=c and the claim follows with the degree statement in Lemma 1.

Proposition 3. If v ∈ V is a palindrome of even length then H′ (v) $= \begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix}$ for some a, b, d ∈ F_2[x].

Proof: Let v = ww^r for some w ∈ V. The proof is by induction on |w|. If |w| =0 the hash H′(ww^r) is the identity matrix and the statement holds trivially.

Suppose now we extend a string w of given length by one bit, yielding a palindrome $\beta v \beta$= (βw)($w^r\beta$) with $\beta \in \{0,1\}$.By induction hypothesis,

we have H′(v) = H′(ww^r) = $\begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix}$ so that

H′ (0v0) = $\begin{pmatrix} 0 & 1 \\ 1 & x \end{pmatrix} \begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & x \end{pmatrix} = \begin{pmatrix} d^2 & b+d^2x \\ b+d^2x & a^2+d^2x^2 \end{pmatrix}$ and

H′ (1v1)=$\begin{pmatrix} 0 & 1 \\ 1 & x+1 \end{pmatrix} \begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & x+1 \end{pmatrix} = \begin{pmatrix} d^2 & b+d^2(x+1) \\ b+d^2(x+1) & a^2+d^2(x+1)^2 \end{pmatrix}$

That is in general

H′ (βvβ) = $\begin{pmatrix} d^2 & b+d^2(x+\beta) \\ b+d^2(x+\beta) & a^2+d^2(x+\beta)^2 \end{pmatrix}$

Consequently both the diagonal elements of H′($\beta v \beta$) are squars

since $a^2+d^2(x+\beta)^2 = [a+d(x+\beta)]^2$ and the result follows.

Corollary.1 Let v ∈ V be a palindrome of even length. Then ρ(v) = $\begin{pmatrix} 0 & d^2 \\ d^2 & d^2 \end{pmatrix}$ for some d ∈ F_2[x] with deg d = |v|/2. More specifically $d^2$is the lower right entry of H′ (v).

Proof. $\rho(v) = H'(0v0) + H'(1v1)$

$$= \begin{pmatrix} d^2 + d^2 & b + d^2x + b + d^2(x+1) \\ b + d^2x + b + d^2(x+1) & a^2 + d^2x^2 + a^2 + d^2(x+1)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & d^2 \\ d^2 & d^2 \end{pmatrix}.$$

Corollary 2. Let $b_n.....b_1b_1.....b_n \in V$ be a palindrome of length 2n.Then for $0 \leq I \leq n$, the square root p_i of the lower right entry of $H'(b_i.....b_1b_1.....b_i)$ is given by

$$p_i = \begin{cases} 1 & if\ i = 0 \\ x + b_i + 1 & if\ i = 1 \\ (x+i)p_{i-1} + p_{i-2} & if\ 1 < i \leq n \end{cases}$$

3. COLLISION AND EUCLIDEAN ALGORITHM

3.1. Construction of Palindrome

From corollaries 1 and 2 we see that the square roots of the lower right entries of $H'(b_1b_1)$, $H'(b_2b_1b_1b_2)$, $H'(b_3b_2b_1b_1b_2b_3)$, etc, satisfy Euclidean algorithm sequence(in reverse order) where each quotient is either x or x+1[2]. Those sequences are often called maximal length sequences for the Euclidean algorithm or maximal length Euclidean sequences and they have long been a topic of interest in number theory.

Mesirov and Swweet [7] showed that, when $q(x) \in F[x]$ is irreducible there exist exactly two polynomials p(x) such that q(x) and p(x) are the first terms of a maximal length Euclidean sequence. They also provide an algorithm to compute them, which will be given below.

Proposition 4. (Mesirov and sweet). Given any irreducible polynomial q of degree n over F_2, there is a sequence of polynomials

$p_n, p_{n-1}, ..., p_0$ with $p_n = q$, 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 $p = p_{n-1}$ as mentioned in proposition 4 which matches our given polynomial $p_n = q$, the Euclidean algorithm will uniquely compute the sequence $p_n, p_{n-1}, ...p_1, p_0 = 1$.

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

$H(0\beta_n.....\beta_1^{-1}\beta_1^{-1}.....\beta_n 0) = H(1\beta_n.....\beta_1^{-1}\beta_1^{-1}.....\beta_n 1)$

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

3.2. To find the maximal length Euclidean sequence:

1. Construct a matrix $A \in F_2^{(n+1)\times n}$ from the n+1 polynomials $g_o = x^0 \bmod q(x)$,
   $g_i = x^{i-1} + x^{2i-1} + x^{2i} \bmod q(x)$ for i= 1,2,.....,n

   Placing in the i^{th} row of A the coefficients
   $a_{i,0}, a_{i,1}, ...a_{i,n-1}$ of the polynomial
   $g_i = a_{i,0} + a_{i,1}x + ...a_{i,n-1}x^{n-1}$.
2. Solve the linear system $Au^t = (10.....01)$ where $u = (u_1.....u_n)$.
3. Compute p(x) by multiplying q(x) by $\sum_{i=1}^{n} u_i x^{-i}$ and taking only the non negative powers of x.

3.3 Collisions for specified Polynomials

Example 1. Let $q(x) = x^2 + x + 1$ be the irreducible polynomial.

Mesirov- Sweet algorithm:

1. $g_0 \equiv 1 \bmod q(x)$
$g_1 \equiv 1 + x + x^2 \bmod q(x)$
$g_2 \equiv x + x^3 + x^4 \bmod q(x)$

Hence $g_0 = 1+0x$
$g_1 = 0+0x$
$g_2 = 1+0x$

2. Solve $\begin{pmatrix} 10 \\ 00 \\ 10 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$.

The values of u are (1 1) and (1 0).

3 For the first value u, p(x) = x, the palindrome v = 1111 and H' (011110) + H'(111111) = $\begin{pmatrix} 0 & d^2 \\ d^2 & d^2 \end{pmatrix}$ where d = q(x) = x^2+x+1.

$$\rho(v) \equiv \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \bmod (x^2+x+1)$$

if and only if H(011110) = H(111111).

It is true, since H (011110) = $\begin{pmatrix} 0 & 1 \\ 1 & x+1 \end{pmatrix}$ = H (111111).

For the second value of u, p(x) = x+1, palindrome is v = 0000 and

We have the collision

H (000000) = $\begin{pmatrix} 0 & 1 \\ 1 & x \end{pmatrix}$ = H (100001).

Example 2. Let $q(x) = x^3+x+1$

Then the values of u are (110) and (101).

For u = (110), $p(x) = x^2+x+1$ and $\rho(v) = \begin{pmatrix} 0 & d^2 \\ d^2 & d^2 \end{pmatrix}$ where $d = q(x) = x^3+x+1$.

The collision is

H (01111110) = $\begin{pmatrix} 0 & 1 \\ 1 & 1+X \end{pmatrix}$ = H (11111111).

For u = (101), $p(x) = x^2$ and the collision is

H (00100100) = H (10100101).

By Proposition 1. Collision in h and H are equivalent. For higher degree irreducible polynomials q(x) we implement the attack in the computer algebra system Magma[1] on a standard PC .For each q(x) there will be two solutions for p(x) so we obtain two bit strings $v_1, v_2 \in \{0,1\}^n$ with $h(0v_iv_i^r0) = h(1v_iv_i^r1)$ for i=1,2.

That is, we obtain two collisions of bit strings of length 2n+2.The value v_2 can be obtain by reversing v_1 followed by inverting the first and last bit.

In example 1, $v_1 = 11$, $v_2 = 00$

In example 2, $v_1 = 111$, $v_2 = 010$.

3.4 Second Preimage

Here we found the second preimage for the new hash function. For that we have the following:

Proposition 5. Let $\begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix}$ = H (v) for some palindrome v of even length with d ≡0, then

h (ovov) = h(1v1v) = h(v0v0) = h(v1v1) = I = h().

Proof: We have $a^2d^2+b^2 = 1$ and $d \equiv 0$.

Hence $b \equiv 1$.By direct computation we have

$$H(0v) = \begin{pmatrix} 0 & 1 \\ 1 & \propto \end{pmatrix}\begin{pmatrix} a^2 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \propto +a^2 & 1 \end{pmatrix}$$

$$H(0v0v) = H(0v)\,H(0v) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$H(v0v0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$H(1v) = \begin{pmatrix} 0 & 1 \\ 1 & \propto +1 \end{pmatrix}\begin{pmatrix} a^2 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \propto +1+a^2 & 1 \end{pmatrix}$$

$$H(1v1v) = H(1v)\,H(1v) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$H(v1v1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Hence H (0V0V) = H(v0v0) = H(1v1v)
= H(v1v1) = I = H() .

If H(u) = *I* for some $u = b_1....b_m \in \{0,1\}^*$ then

$$h(u) = B_{b_1}.....B_{b_m} = B_0^{-1}C_{b_1}B_0....B_0^{-1}C_{b_m}B_0$$
$$= B_0^{-1}b_1 \ldots . C_{b_m}B_0 = B_0^{-1}\mathrm{H(u)}B_0$$
$$= B_0^{-1}IB_0 = I$$

Hence h (0v0v) = h(v0v0) = h(1v1v)
= h(v1v1) = *I* = h().

The message v in Proposition 5 can be obtained by applying Mesirov and Sweet's algorithm to d = q(x) as in the collision attack (see corollary 2).We therefore obtain a message u colliding with the void message for the hash function h .A second preimage algorithm is straightforwardly deduced, since for any $v \in \{0,1\}^*$ we have h(vu) = h(v) h(u) = h(v).

REFERENCES

1. Wieb Bosma, John Cannon, and Catherine Playoust, The Magma Algebra System I: The User Language. Journal of Sympolic Computation, 24 (1997), pp.235-265.
2. Christophe Petit and Jean-Jacques Quisquater, Preimage for the Tillich-Zemor hash function. Proceedings of SAC 2010, pp.282-301.
3. Daugles R Stinson, *Cryptography theory and practice*, Second Edition, Chapman & Hall/CRC.
4. John R Durbin, *Modern Algebra*, John Wiley & Sons.
5. Joju K.T and Lilly P.L Alternate form of Hashing with Polynomials. Proceedings of the International workshop in Cyber Security, St. Joseph's College, Irinjalakuda pp.2011, (IWCS2k11) 43-45, 2011
6. Markus Grassl, Ivana Ilic, Spyros Magliveras, and Rainer Steinwadt, *Cryptanalysis of the Tillich-Zemor Hash function*, Cryptology ePrint Archive, Report 2009/376, 2009, http://eprint.iacr.org/.
7. Jill P. Mesirov and Melvin M. Sweet. Continued Fraction Expansions of Rational Expressions with Irreducible Denominators in Characteristic 2. Journal of Number Theory, 27 (1987), pp.144-148.
8. Stefan Lucks, Design principles of Iterated Hash function, ePrint Archive: Report (2004), pp.1-22.
9. J.P.Tillich and G. Zemor, *Hashing with SL2*, Advances in Cryptology Lecture Notes in Computer Science, vol. 839(1994), Springer-Verlag, pp. 40-49.
10. Vladimir Shpilrain, *Hashing with polynomials*, Proceedings of ICISC 2006, Springer (2006), pp. 22-28.

Source of support: Nil, Conflict of interest: None Declared

