International Research Journal of Pure Algebra -3(10), 2013, 321-324
Available online through www.rjpa.info ISSN 2248–9037

# HASHING WITH $SL_2$ USING NEW GENERATORS

**Joju K.T.[1*] & Lilly P.L. [2]**

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

*[2]Department of Mathematics, St. Joseph's College, Irinjalakuda, Kerala, India.*

*(Received on: 25-10-13; Revised & Accepted on: 04-11-13)*

***ABSTRACT***

*J.P Tillich and G. Zemor proposed a family of hash functions based on computations over a finite field of characteristic 2. In this paper we generate a family of hash functions by replacing the generators with new generators.*

## 1. INTRODUCTION

At CRYPTO 94 [13], Tillich and Zemor proposed a family of hash functions, based on computing a suitable matrix product in groups of the form $SL_2$ ($\mathbb{F}_2$n). In 2009, Grassl *et al* [11] found collisions for the construction. In 2010Christophe Petit *et al* [2] found the preimage and second preimage for the same. In 2012 we, Joju K.T and Lilly P.L [7] constructed a hash function using new generators for Tillich –Zemor hash function. We [7, 10] found collision and preimages for the same. Further we [6, 8, 9] constructed the keyed versions of the hash functions, they were still unbroken. Now we are going to construct a hash function with different generators for Tillich-Zemor hash function.

### I. I. Cryptographic Hash Functions and MACs

Hash functions [1, 3, 5] 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 [12] 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 ≠ x' such that h(x) = h(x')).

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

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 [12]. 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 new message $x' \neq x_i$

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

Definition: A MAC is a function h: $K \times M \rightarrow R$ where the key space $K = \{0, 1\}^k$, the message space $M = \{0,1\}^*$, and the range $R = \{0,1\}^n$ for $k,n \geq 1$

Early suggestions (SHA family) did not really use any mathematical ideas apart from Merkle-Damgard [5] 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 [13] 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 $R = \mathbb{F}_2[x]./(q(x))$ [4]. Where $\mathbb{F}_2$ is the field of two elements, $\mathbb{F}_2[x]$. is the ring of polynomials over $\mathbb{F}_2$. and (q(x)) is the ideal of $\mathbb{F}_2[x]$. 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].

Thus $R= \mathbb{F}_2[x]./(q(x))$ isomorphic to $\mathbb{F}_2n$ the field with $2^n$ elements .The matrices $A_0$ and $A_1$ are the following:

$A_0 = \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}$, $A_1 = \begin{pmatrix} x & x+1 \\ 1 & 1 \end{pmatrix}$.

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:

$$h'(b_1.....b_m) = A_{b_1}.....A_{b_m} \, mod q(x)$$

## 2. HASH FUNCTION

### 2.1 New hash function

Let $B_0$ and $B_1$ be the following matrices

$B_0 = (A_0^{-1})^T$ and $B_1 = (A_1^{-1})^T$ then $B_0 = \begin{pmatrix} 0 & 1 \\ 1 & x \end{pmatrix}$ and $B_1 = \begin{pmatrix} 1 & 1 \\ x+1 & x \end{pmatrix}$.

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

$$h(b_1....b_m) = B_{b_1}.....B_{b_m} \, mod q(x).$$

Since the matrices $A_0$ and $A_1$ generate $SL_2(R)$, by preliminary group theory $B_0$ and $B_1$ also generate the same.

As in the case of this new hash function this new construction is twofold: the hash function display a catenation property and one can associate to such a scheme a Cayley Graph, several parameters of which are relevant in security.

*Concatenation property*. If x and y are two texts, then their cancatenation xy has hashed value H(xy) = H(x)H(y). This clearly allows an easy parallelization of the scheme, and pre computations when parts of the message are known in advance.

 

*Parameters of the associated cayley Graph*. We can associate to this scheme the Cayley graph (G, S): its vertex set is G and there is a directed edge from $g_1$ to $g_2$ if and only if $g_1^{-1}g_2 \in S$. The following parameters are of fundamental importance when studying the security of the hash function.

**Definition:** Call the directed girth of a graph, the largest integer $\partial$ such that given any two vertices v and w, any pair of distinct directed paths joining v and w will be such that one of those paths has length(i.e. number of edges) $\partial$ or more.

This property of cayley graph gives the following property of hash function.

**Proposition 1:** If we replace k consecutive symbols of a text $x = x_1x_2....x_ix_{i+1}...x_{i+k}\ x_{i+k+1}.....x_t$ by a string of h consecutive symbols so that the resulting text $x' = x_1x_2....x_iy_{i+1}...y_{i+h}\ x_{i+k+1}.....x_t$ have the same hashed value, then sup (k, h) $\geq \partial$.

In other words, if we can obtain cayley graph with a large $\partial$, we protect against local modifications of the text. The following theorem establishes the girth for this Cayley graph.

**Theorem 2:** The girth of the Cayley graph associated with the group $SL_2(R)$ and generators $\{B_0,B_1\}$ is greater than n.

To prove this theorem, first consider the following lemmas.

**Lemma 3:** Let $S_1, S_2,...,S_k \in \{B_0,B_1\}$ and k < n. Then $S_1S_2...S_k$ has the form $M_{B_0}$ when $S_k = B_0$ or $M_{B_1}$ when $S_k = B_1$, where $M_{B_0} = \begin{pmatrix} a_{k-2}(x) & b_{k-1}(x) \\ c_{k-1}(x) & d_k(x) \end{pmatrix}$, $M_{B_1} = \begin{pmatrix} a_{k-1}(x) & b_{k-1}(x) \\ c_k(x) & d_k(x) \end{pmatrix}$ and $a_i$, $b_i$, $c_i$, $d_i$ are polynomials of degree i over $\mathbb{F}_2[x]$.

Proof: We prove the lemma by induction on k. Suppose that the lemma holds for strings of length $l < k$. Let $S_1, S_2,...,S_l \in \{B_0,B_1\}$. Suppose that $S_l = B_0$. By induction hypothesis the product $S_1S_2...S_l$. $B_0$ has the form

$$\begin{pmatrix} a_{l-2}(x) & b_{l-1}(x) \\ c_{l-1}(x) & d_l(x) \end{pmatrix} * B_0 = \begin{pmatrix} b_{l-1}(x) & a_{l-2}(x) + xb_{l-1}(x) \\ d_l(x) & c_{l-1}(x) + xd_l(x) \end{pmatrix}$$

Similarly, the product $S_1S_2...S_l.B_1$ has the form

$$\begin{pmatrix} a_{l-2}(x) & b_{l-1}(x) \\ c_{l-1}(x) & d_l(x) \end{pmatrix} * B_1 = \begin{pmatrix} a_{l-2}(x) + (x+1)b_{l-1}(x) & a_{l-2}(x) + xb_{l-1}(x) \\ c_{l-1}(x) + (x+1)d_l(x) & c_{l-1}(x) + xd_l(x) \end{pmatrix}.$$

As $l < k < n$, no reduction occurs modulo the irreducible polynomial of degree n. Thus in both the cases this follows the form stated in the lemma. In the case where $S_l = B_1$ the same process can be used to show that the product $S_1S_2...S_l$. $B_0$ has the form $M_{B_0}$ and $S_1S_2...S_l$. $B_1$ has the form of $M_{B_1}$.

**Lemma 4:** Let $S_1, S_2,...,S_k$ and $T_1,T_2,...,T_l$ be two different strings of $B_0$s and $B_1$s with $k, l < n$. Then the $S_1S_2...S_k \neq T_1T_2...T_l$.

**Proof:** By the previous lemma, these strings can only have the same form if $k = l$ and $S_k = T_l$.

By canceling $S_k$ from both sides and iterating this argument, we see that $S_i$ must equal $T_i$ for all $1 \leq i \leq k$.

Thus, to see that the girth of the Cayley graph associated with $SL_2(R)$ is at least n,

let $S_1,S_2,...,S_k$ and $T_1,T_2,...,T_l$ be from$\{B_0,B_1\}$ and $l, k < n$.

By above lemma, the products $S_1S_2...S_k$ and $T_1T_2...T_l$ must be different. Therefore the girth of the graph must be at least n.

*Expanding properties*. A desirable feature of any hash function is the equidistribution of the hashed values. This property can be guaranteed if the associated Cayley graph $\mathcal{C}$ (G, S) satisfies

**Proposition 5:** If $\mathcal{C}$(G,S) is a Cayley graph such that the gcd of its cycle lengths equal 1, then for the corresponding hash function, the distribution of hashed values of texts of length n tends to equidistribution when n tents to infinity.

Thus we got a hash function which meets all the security properties.

 

## 3. CONCLUSION

The advantages of this paper are 1.The new hash function is not vulnerable for collision, preimage and second preimage. 2. Its execution is easy. 3. It is based on the finite fields.4.Its execution is bitwise. This hash function can be used in Cloud Computing, Digital Signature and the other relative areas.

## 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, Analysis and Design of Cryptographic hash Functions, MAC algorithms and Block Ciphers, Doctoral Dissertation, KU Leuven2004D/2004/7515 ISBN 90-5682-527-5
2. Cristophe Petit, Jean-Jacques Quisquater, Preimages for the Tillich-Zemor hash function, Proceedings of the 17 th International Conference on Selected Areas in Cryptography pp 282-301, Springer-Verlag Berlin, Heidelberge 2011 ISBN:978-3-64[1]
3. Daugles R Stinson, *Cryptography theory and practice*, Second Edition, Chapman & Hall/CRC.
4. John R Durbin, *Modern Algebra*, John Wiley & Sons 2005.
5. Joju K.T and Sr. Lilly P.L Alternate form of Hashing with Polynomials. Proceedings of the International workshop in Cyber Security, St. Joseph's College, Irinjalakuda pp. 43-45, 20
6. Joju K.T and Lilly P. L, Keyed Tillich-Zemor Hash Function, Research Journal of Pure Algebra-vol. 3(1), 2013, www .rjpa .info ISSN 2248-9037, page: 24-32.
7. Joju K.T and Lilly P.L, Tillich-Zemor Hash Function with New Generators and Analysis. Research Journal of Pure Algebra-2(11), 2012, www.rjpa.info ISSN 2248-9037, page: 338-343.
8. Joju K.T and Lilly P.L] A Keyed Hash Function IOSR Journal of Mathematics (IOSR-JM)e-ISSN: 2278-5728. Volume 5, Issue 4 (Jan. - Feb. 2013), PP 47-55www.iosrjournals.org.
9. Joju K.T and Lilly P.L Alternate form of Tillich-Zemor Hash Functionwhich Resist Second Preimage, International J. of Math. Sci. & Engg. Appls. (IJMSEA) ISSN 0973-9424, Vol. 7 No. II (March, 2013), pp 79-98.
10. Joju K.T and Lilly P.L Preimage of Tillich–Zemor Hash Function with New Generators ,Journal of Applied Mathematical Sciences, Vol. 7, 2013, no. 85, 4237 – 4248 HIKARI Ltd, www.m-hikari.com http://dx.doi.org /10.12988/ams.2013.36329.
11. Markus Grassl, Ivana Ilic, Spyros Magliveras, and Rainer Steinwandt, Cryptanalysis of the Tillich-Zemor hash function, Journal of Cryptology, Volume 24 Number-1.pp 148-156.
12. B. Praneel: Analysis and Design of Cryptographic Hash Functions. Doctoral Dissertation K.U .Leuven Jan. 1993.
13. 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.

**Source of Support: University Grants Commission (Government of India), India.**
**Conflict of interest: None Declared**

 