HASHING WITH DISCRETE HEISENBERG GROUP USING NEW GENERATORS

Lilly P.L[1], Vibitha Kochamani .V[2]

[1,2]*Department of Mathematics, St. Joseph's College, Irinjalakuda, Kerala, India.*

[1]*sr.christy@gmail.com*,[2]*myvkumari@gmail.com*.

ABSTRACT--*We introduce a New Cryptographic Hash Functions, which will give a new hashed values corresponding to the new generators. We show why having a large girth, will satisfy the property that local modifications of a text will necessarily modify the hashed value.*

Keywords--*Cryptographic Hash function, Discrete Heisenberg group, New Generators, Girth*

INTRODUCTION[1]

Hash functions [3, 10, and 11] are simple and easy-to-compute, that takes a variable length input and converts it to a fixed-length output. 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. A cryptographic hash function can provide assurance of data integrity. Hash functions are widely used in numerous cryptographic protocols and a lot of work has already been put into devising adequate hashing schemes. Hash functions are used as compact representations or digital finger prints, of data and to provide message integrity. Some hash functions in current use have been shown to be vulnerable. Early suggestions (particularly SHA family) did not really use any mathematical ideas apart from Merkle-Damgard [3] 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. The basic idea initiated in the paper [13] is of looking for potentially good Hash functions among Cayley Graphs is that girth is a relevant parameter to hashing the group $G = SL_2(\mathbb{F}_p)$, the group of 2x2 matrices of determinant 1 over the integers modulo a prime p and $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and they analyzed the girth of the cayley graph of the group.

At CRYPTO 94 [11], Tillich and Zemor proposed a family of hash functions, based on computing a suitable matrix product in groups of the form SL2 ($\mathbb{F}_{2n}$). In [4], generate a family of hash functions by replacing the generators with new generators.

In [6], we introduce a Cryptographic Hash Functions that are in correspondence with directed cayley graph having large girth and small diameter, which satisfy the properties of the Hash function.

PRELIMINARIES [2]:

2.1 Definition: In [5] , the vertices 'x' of the graph G and one draws a directed edge from x to y, labeled by the group element g for any x, g Є G if and only if y=xg. The group consisting of all such vertices and edges will be denoted by Cay (G, G).

2.1.1. The Girth of a graph G, denoted g (G) is the length of the shortest cycle (if any) in G.

Depending on these requirements Praneel [1, 8] provides the following informal definitions for two different types of hash functions.

2.2. 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).

2.3. 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')).

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

The following definition is defined by [2, 7]

2.5.Definition: The Heisenberg group is the group of 3×3 upper triangular matrices of the form $\begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix}$ under the operation of matrix multiplication. Elements a, b and c can be taken from any commutative ring with identity, often taken to be the ring of real numbers (resulting in the "continuous Heisenberg group") or the ring of integers (resulting in the "discrete Heisenberg group").it is denoted by Heisgroup. From this definition, it is easily seen that the discrete Heisenberg group is generated by $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

2.6 HASH FUNCTION

Now, we devising the Hash Function as follows: to an arbitrary text of $\{0, 1\}^*$, associate the string of {A, B} obtained by substituting 0 for A and 1 for B, then assign to A and B values of adequately chosen matrices of Heis($\mathbb{F}_p$), then evaluate the product associated with the string of A's and B's in the groupHeis ($\mathbb{F}_p$), where $\mathbb{F}_p$is the field on p elements, p being chosen large prime number. The Hashed value is the computed product. A multiplication by 'A' or 'B' in Heis (F_p) requires essentially 9 additions, so hashing an n bit text requires 9n additions of logp bits, which is reasonably fast.

3.1 In [6], we change the generators, in order to get a new hashed function.

Let $A_1 = A^{-1}$ and $B_1 = B^{-1}$

i.e)Let $A_1 = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and $B_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$ be a new pair of generators of Heis ($\mathbb{F}_p$) and

Let $m = \mathrm{m}_0\mathrm{m}_1 \dots\dots\dots \mathrm{m}_n$ be a binary string. Then $\mathrm{H}(m) = \pi(\mathrm{m}_0)\pi(\mathrm{m}_1) \dots\dots\dots\dots \pi(\mathrm{m}_n)$, where

$\pi(\mathrm{m}_i) = \begin{Bmatrix} \mathrm{A}_1 for \mathrm{m}_i = 0 \\ \mathrm{B}_1 for \mathrm{m}_i = 1 \end{Bmatrix}$; 0≤ i ≤ n. This hash function is strongly related to the Cayley Graph associated with Heis ($\mathbb{F}_p$), generated by A_1, B_1 denoted byG_1.

From the table given below, we have shown that the difference between the Hash values using Discrete Heisenberg Group with generators A & B [6] and Hash Values using New Generators for different length of the strings.

Table 3.2

Bits	Hash Values of generators A and B	Hash Values of New generators A_1 and B_1
0110	$\begin{pmatrix} 1 & 2 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & -2 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix}$
1011010	$\begin{pmatrix} 1 & 3 & 4 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & -3 & 4 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{pmatrix}$
1011011001011	$\begin{pmatrix} 1 & 4 & 15 \\ 0 & 1 & 8 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & -5 & 20 \\ 0 & 1 & -8 \\ 0 & 0 & 1 \end{pmatrix}$

3.3 PROPERTIES OF HASH FUNCTION

Recall that in [6], the hash function construction presented above is directly associated with the cayley graph $G_1(G, S)$, where G is a group generated by the elements of the set S which also satisfies the Concatenation Property and Parameters of the associated Cayley Graph.

The following parameters are of fundamental importance, when studying the security of the hash function

Definition*:* The Girth of a graph, the largest integer g such that given any two vertices u and v , any pair of distinct paths joining u to v will be such that one of those paths has length g or more.

The definition of the girth, immediately leads to the following property of the Hash function, for the associated Cayley graph.

Proposition 1:

If we replace 'k' consecutive elements of the product,$x = x_1x_2 \dots\dots\dots x_ix_{i+1} \dots\dots\dots x_{i+k}x_{i+k+1} \dots\dots\dots x_t$ where

$x_j \in S, j = 1$ to twith 'l' string of consecutive elements $y_{i+1}, \dots\dots\dots y_{i+l} \in S$ such that

$x' = x_1x_2 \dots\dots\dots x_iy_{i+1} \dots\dots\dots y_{i+l}x_{i+k+1} \dots\dots\dots x_t$have the same hashed value, then max (k,l) ≥ g.

In other words, if we can obtain cayley graph with a large 'g', we protect against local modifications of the text.

We can determine the Girth of the Cayley Graph associated with generators A_1 and B_1.

Theorem 2:

The Girth of the Cayley Graph of Heis ($\mathbb{F}_p$) with generators A and B is atleast$\log_3(p)$.

Proof: Let $S_1, S_2, \dots\dots\dots S_K$ and $T_1, T_2, \dots\dots\dots\dots T_l$ be two different strings of A's and B's with $k, l < \log_3(p)$. The product of these strings can only have the same form if $k = l$ Then $S_K = T_l$, by cancelling S_K from both sides and iterating this argument, we see that S_i must be equal to $T_i, \forall\, 1 \le i \le k$.Thus the products of $S_1, S_2, \dots\dots\dots S_K$ and $T_1, T_2, \dots\dots\dots\dots T_l$ must be different. Therefore the girth of the graph is at least $\log_3(p)$

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] Luca Capogna, Donatella Danielli, Scott D. Pauls, Jeremy Tyson,*An Introduction to the Heisenberg Group and the Sub-Riemannian Isoperimetric*,Springer Science & Business Media, 08-Aug-2007 - Mathematics - 224 pages

[3] Daugles R Stinson, *Cryptography theory and practice*, Second Edition, Chapman & Hall/CRC.

[4] Joju K.T & Lilly P.L / Hashing with SL2 Using New Generators / International Research Journal of Pure Algebra-#(10) / Oct-2013,321-324

[5] Joseph.A.Gallian,*Contemporary Abstract Algebra*, 8th Edition University of Minnesota, Duluth,ISBN-10: 1133599702 | ISBN-13: 9781133599708

[6] VibithaKochamani.V, Lilly P.L and Joju K.T , *Hashing with Discrete Heisenberg Group and Graph with Large Grith* ,In Journal of Theoritical Physics and Cryptography,Vol.11, May 2016.

[7] A.de Mesmay (2009),*The Heisenberg Group and Pansu's Theorem*, Available from www.gipsa-lab.fr/~arnaud.demesmay/mesmay.pdf

[8] B. Praneel: *Analysis and Design of Cryptographic Hash Functions*. Doctoral Dissertation K.U .Leuven Jan. 1993.

[9] Jean-Pierre Tillich and Gilles Zémor, *Group theoretic hash functions*,Proceedings of the FirstFrench-Israeli Workshop on Algebraic Coding (London,UK), Springer-Verlag, 1993, pp. 90–110.

[10] V. Shpilrain. *Hashing with polynomials*, In ICISC 2006, pages 22–28. Springer, 2006,Lecture Notes in Computer Science No. 4296.

[11] 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.

[12] Gilles Zémor, *Hash functions and Cayley Graph. To appear in Designs*,Codes and Cryptography.

[13] Gilles Zémor, *Hash functions and graphs with large girths*, *EUROCRYPT* (Donald W. Davies, ed.), Lecture Notes in Computer Science, vol. 547, Springer, 1991,pp. 508–511.