Improved Form of Collision-Free Hashing From Lattice Problems

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

ABSTRACT
In 1995, Ajtai described a construction of one-way functions whose security is equivalent to the difficulty of some well-known approximation problems in lattice. We improve the hash function which meets all the three security properties.

KEYWORDS: *Collision, Hash function, Lattice, Generalized inverse.*

I. INTRODUCTION

At CRYPTO 94 [15], Tillich and Zemor proposed a family of hash functions, based on computing a suitable matrix product in groups of the form $SL_2(\mathbb{F}_2n)$. In 2009, Grassl et al [12] found collisions for the construction. In 2010Christophe Petit et al [3] found the preimage and second preimage for the same. In 2012 we, Joju K.T and Lilly P.L [7,10] constructed a hash function using new generators for Tillich –Zemor hash function. We [7,11] found collision and preimages for the same. Further we [5, 6, 8] constructed the keyed versions of the hash functions, they were still unbroken. Now we are going to construct a hash function whose security is depending on the SVP in lattice. In 1995, Ajtai [1] described a problem that is hard on the average if some well-known lattice problems are hard to approximate in the worst case, and demonstrated how this problem can be used to construct one way functions. We improve the hash function which meets all the three security properties.

1.1 Cryptographic Hash Functions and MACs

Hash functions [2, 9] 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 [14] 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 $x' \neq x$ 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 $h(x) = h(x')$).

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 [14]. 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 $k,n \geq 1$

1.2 Inverse of a Matrix

Let A be a matrix of order m x n. If m>n we define the left inverse of A as follows:

$$A_{left}^{-1} = (A^T A)^{-1} A^T.$$

If m<n we define the right inverse of A as follows:

$$A_{right}^{-1} = A^T (A A^T)^{-1}.$$

1.3 Lattices

A lattice is a discrete additive subgroup of $\mathbb{R}^n$, i.e., it is a subset $\Lambda \subseteq \mathbb{R}^n$ satisfying the following properties:

1. Λ is closed under addition and subtraction.
2. There is an $\epsilon > 0$ such that any two distinct lattice points $x \neq y \in T$ are at distance at least $\| x - y \| \geq \epsilon$.

It is also defined in the following manner [4,13]:

Let $B = [b_1,...,b_k] \in \mathbb{R}^{nxk}$ be linearly independent vectors in $\mathbb{R}^n$. The lattice generated by B is the set

$$\mathcal{L}(B) = \{ Bx : x \in \mathbb{Z}^k \} = \left\{ \sum_{i=1}^{k} x_i b_i : x_i \in \mathbb{Z} \right\}$$

of all the integer linear combinations of the columns of B. The matrix B is called a basis for the lattice $\mathcal{L}(B)$. The integers n and k are called the dimension and rank of the lattice. If n = k then $\mathcal{L}(B)$ is called a full rank lattice.

In 1995[1], Ajtai described a problem that is hard on the average if some well-known lattice problems are hard to approximate in the worst case, and demonstrated how this problem can be used to construct one way functions. His surprising discovery was that lattices, which up to that point were used only as tools in cryptanalysis, can actually be used to construct cryptographic primitives. His work sparked a great interest in understanding the complexity of lattice problems and their relation to cryptography.

Ajtai's discovery was surprising for another reason: the security of his cryptographic primitives is based on the worst case hardness of lattice problems. What this means is that if one succeeds in breaking the primitive, even with some small probability, then one can also solve any instance of a certain lattice problem. This remarkable property is what makes lattice based cryptographic constructions so attractive. In contrast, virtually all other cryptographic constructions are based on some average-case assumption. For example, in cryptographic constructions based on factoring, the assumption is that it is hard to factor numbers chosen from a certain distribution. But how should we choose this distribution? Obviously, we should not use numbers with small factors, but perhaps there

are other numbers that we should avoid? In cryptographic constructions based on worst-case hardness, such questions do not even arise.

II. HASH FUNCTION

Ajtai constructed the collision free hash function in the following manner:
For security [4,13] parameter n, we pick a random n x m matrix M with entries from $\mathbb{Z}_q$, where m and q are chosen so that

$$n\log q < m < \frac{q}{2n^4} \text{ and } q = O(n^c)$$

for some constant c>0 (e.g., $m = n^2$, $q = n^7$).
The hash function $h_M : \{0,1\}^m \rightarrow \mathbb{Z}_q^n$ is then defined for $s = s_1 s_2 \ldots s_m \in \{0,1\}$
$h_M(s) = Ms \bmod q = \sum_i s_i M_i \bmod q$, where M_i is the i^{th} column of M.
Notice that h_M's input is m-bit long, where as its output is nlogq bits long. Since we choose the parameters such that m > nlogq, there are collisions in h_M.
We improve the hash function in the following manner:
For security parameter n, we pick a random m x n (m>n) matrix M with entries from $\mathbb{Z}_q$, where m and q are chosen so that

$$n\log q < m < \frac{q}{2n^4} \text{ and } q = O(n^c)$$

for some constant c>0 and q is prime.
The hash function $h_{M_{left}^{-1}} : \{0,1\}^m \rightarrow \mathbb{Z}_q^n$ is then defined for
$s = s_1 s_2 \ldots s_m \in \{0,1\}^m$, as

$$h_{M_{left}^{-1}}(s) = M_{left}^{-1} s \bmod q = \sum_i s_i M_i \bmod q,$$

where M_i is the i^{th} column of M_{left}^{-1}. Where M_{left}^{-1} is the left inverse of M.
Notice that $h_{M_{left}^{-1}}$'s input is m-bit long, where as its output is nlogq bits long. Since we choose the parameters such that m > nlogq, there are collisions in $h_{M_{left}^{-1}}$. It is infeasible to find any of these collisions unless some well known lattice problem have good approximation in the worst case. It follows that, although it is easy to find solutions for the equations $M_{left}^{-1} s \equiv 0 (\bmod q)$, it seems hard to find binary solutions. It is also preimage resistant because we cannot find the left inverse of M_{left}^{-1}(as m>n). So we claim that the hash function, $h_{M_{left}^{-1}}$ obeys all the three security requirements for a hash function.

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]. M. Ajtai, Generating Hard Instances of Lattice Problems. In 28th ACM Symposium on Teory of Computing, 99-108, Philadelphia, 1996.
[2]. Bart Van Rompay, Analysis and Design of Cryptographic hash Functions, MAC algorithms and Block Ciphers, Doctoral Dissertation, KU Leuven 2004D/2004/7515 ISBN 90-5682-527-5.
[3]. 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.
[4]. O.Goldreich, S. Goldwasser, S. Halevi, Collision-free hashing from lattice problems, Eiectronic Colloquium, 1996, on computational Complexity, http://www.eccc.uni-trier.de/eccc/.
[5]. K. T Joju , P.L Lilly , A Keyed Hash Function, IOSR-Journal of Mathematics, (International), e-ISSN: 2278-5728, p-ISSN, : 2319–765X , *Volume 5, Issue 4 (Jan. - Feb. 2013), 47-55.*
[6]. K. T Joju , P.L Lilly , Keyed Tillich - Zemor Hash Function, Research Journal of Pure Algebra (RJPA) (International), ISSN 2248-9037, Vol 3, No.1, 2013, 24-32.
[7]. K. T Joju , P.L Lilly , Tillich-Zemor Hash Function with new Generators and Analysis, Research Journal of Pure Algebra (RJPA) ,(International),ISSN 2248-9037, Vol. 2, issue-11 (2012),338-343.
[8]. K. T Joju , P.L Lilly , Alternate form of Tillich-Zemor hash function which resist second preimage, International J. of Math. Sci. & Engg. Appls. (IJMSEA) (International), ISSN 0973-9424, Vol. 7, No. 2, 2013, 79-98.

[9]. K. T Joju , P.L Lilly , Alternate Form of Hashing with Polynomials, Armored Networks: Critical Cybernetic Defense Technologies ,(International), Excel India Publishers ISBN 93-82062-25-4, (2012), 43-45.

[10]. K. T Joju , P.L Lilly , Tillich-Zemor Hash with new Generators and Collision Analysis, Proceedings of NCMSC-2012 , (International), ISBN 978-93-82359-71-5, Mudranic Technologies Pvt. Ltd. Bangalore, 104-109.

[11]. K. T Joju , P.L Lilly , Preimage of Tillich–Zemor Hash Function with New Generators**,** International Journal of Applied Mathematical Sciences, ISSN 1312-885X (print), ISSN 1314-7552 (online), Vol.7, 2013, 4237-4248.

[12]. Markus Grassl, Ivana Ilic, Spyros Magliveras, and Rainer Steinwandt, Cryptanalysis of the Tillich-Zemor hash function, Journal of Cryptology, Vol. 24, No. 1, 148-156.

[13]. Oded Regev, Lattice-Based Cryptography**.** CRYPTO 2006: 131-141.

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

[15]. 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.