A New Hash Function Based on SQH and NH[a]

Sun-Young LEE†, Kwang-Jo KIM‡ and Hideki IMAI†

† the University of Tokyo, Institute of Industrial Science, 7-22-1, Roppongi, Minato-ku, Tokyo, 106-8558, Japan. E-mail: lsy@imailab.iis.u-tokyo.ac.jp

‡ Information and Communications University, 139 Kajong-dong, Yusong-gu, Taejon 305-350, Korea. E-mail:kkj@icu.ac.kr

† the University of Tokyo, Institute of Industrial Science, 7-22-1, Roppongi, Minato-ku, Tokyo, 106-8558, Japan. E-mail: imai@iis.u-tokyo.ac.jp

Abstract

This paper proposes a new universal hash function that can serve as a tool in achieving authenticity and data integrity. We describe a construction of almost universal hash function suitable for fast software implementation and applicable to fast message authentication. Our hash function is constructed based on the square hash functions(SQH)[5] and the non-linear hash functions(NH)[2]. The security of our hash function is accomplished by the collision probability. And it shows that the proposed hash functions is more secure than SQH.

key words: universal hash functions, message authentication, collision probability, security.

[a] This work was performed as part of Research for the Future Program (RFTF) supported by Japan Society for the Promotion of Science (JSPS) under contract no. JSPS-RFTF 96P00604.

1 Introduction

Universal hash functions, which were first introduced by Carter and Wegman in [3], have a wide range of applications in many areas of computer science, including compilers, databases, search engines, parallel architectures, message authentication, complexity theory and many others. The use of universal hashing for authenticaion received much attention lately. In paricular, many recent works deal with efficient implementation if universal hashing as a tool for achieving fast and secure message authentication. This is also the motivation for our work; however, the construction presented here apply to the other uses of universal hashing as well.

As describe in [4], a universal hash function family can be used to build a MAC. The parties share a secret and randomly chosen hash function from the hash function family, and a secret encryotion key. A message is authenticated by hashing it with the shared hash function and then encrypting the resulting hash. Wegman and Carter showed that when the hash function family is strongly universal and the encryption is realized by a one-time pad, the adversary cannot forge with probability better than that obtained by choosing a random string for the MAC. As suggested many times before, the above approach is a promising one for building a highly-secure and ultra-fast MAC. Two fast universal hash function were proposed in [2] and [5], we call them NH and SQH, respectively. We construct a new hash function based on NH and SQH, and analyze the security.

In section 2 we review the basic definitions and properties of universal hash function families and their variants. In section 3 we review NH and SQH. In section 4 we give the basic construction of our hash, and prove some basic properties. Some concluding remarks are made in section 5.

2 Universal hashing

A universal hash function is a mapping from a finte set D with size d to a finite set R with size r. Usually we specify a family of functions H by specifying some finite set of strings, Key, and explaining how each string $K \in \mathsf{Key}$ names some function $\mathsf{H}_k \in \mathsf{H}$. We may then think of H not as a set of functions from D to R but as a single function $\mathsf{H} : \mathsf{Key} \times D \to R$, whose first argument we write as a subscript. A random element $h \in \mathsf{H}$ is determined by selecting uniformly at random a string $K \in \mathsf{Key}$ and setting $h \in \mathsf{H}_K$.

We are interested in hash-function families in which "collision"(when $h(M) = h(M')$ for distinct M, M') are infrequent.

Let H be a family of functions going from a domain D to a range R. Let ϵ be a constant such that $1/|R| \leq \epsilon \leq 1$. The probabilities denoted below are all taken over the choice of $h \in H$.

Definition 1. *H is a universal family hash functions if for all $x, y \in D$ with $x \neq y$, $Pr_{h \in H}[h(x) = h(y)] = 1/|R|$. H is an ϵ-almost-universal family of hash functions if $Pr_{h \in H}[h(x) = h(y)] \leq \epsilon$.*

Definition 2. *Let R be an Abelian Group and let '-' denote the substraction operation with respect to this group. Then H is a Δ-universal family hash functions if for all $x, y \in D$ with $x \neq y$, and all $a \in R$, $Pr_{h \in H}[h(x) - h(y) = a] \leq 1/|R|$. H is an ϵ-almost-Δ-universal if $Pr_{h \in H}[h(x) - h(y) = a] \leq \epsilon$.*

Definition 3. *H is a strongly universal family hash functions if for all $x, y \in D$ with $x \neq y$, and all $a, b \in R$, $Pr_{h \in H}[h(x) = a, h(y) = b] \leq 1/|R|^2$. H is an ϵ-almost-stronly-universal family of hash functions if $Pr_{h \in H}[h(x) = a, h(y) = b] \leq \epsilon/|R|$.*

3 MMH, SQH, and NH

In this section we overview some fast universal hash functions.

3.1 MMH : Multilinear Modular Hash Function

MMH was originally definde by Carter and Wegman. Halevi and Krawczyk discovered techniques to speed up the software implementation at negligible costs in the collision probabilities[6]. These hash functions are suitable for very fast software implementation. They apply to hashing variable sized data and to fast cryptographic message authentication. In this paper, we intorduce the basic construction of MMH.

Definition 4. *Let $k > 0$ be an integer. Let $x =< x_1, ..., x_k >$, and $m =< m_1, ..., m_k >$, $x_i, m_i \in Z_p$, The MMH family of functions from Z_p^k to Z_p is defined as follows : $MMH \equiv \{g_x : Z_p^k \longrightarrow Z_p | x \in Z_p^k\}$ where the functions g_x are defined as*

$$g_x = m \cdot x = \sum_{i=1}^{k} m_i x_i \mod p.$$

Halevi and Krawczyk proposed an optimized version of MMH, MMH_{32} which appeared in [6].

Theorem 1. *MMH_{32} is an ϵ-almost-Δ-universal family of hash functions with $\epsilon \leq 6 \cdot 2^{-32}$.*

3.2 SQH : Square Hash Function

Patel and Ramzan give an MMH-variant that can be more efficient than MMH in certain settings[5].

Definition 5. *Let $k > 0$ be an integer. Let $x =< x_1, ..., x_k >$, and $m =< m_1, ..., m_k >$, $x_i, m_i \in Z_p$, The SQH family of functions from Z_p^k to Z_p is defined as follows : $SQH \equiv \{g_x : Z_p^k \longrightarrow Z_p | x \in Z_p^k\}$ where the functions g_x are defined as*

$$g_x = \sum_{i=1}^{k} (m_i + x_i)^2 \mod p.$$

SQH should be faster than MMH because squaring can be implemented so it requires roughly half the number of basic wprd multiplications as multiplying two numbers. The optimization of MMH is applied to SQH as well.

Patel and Ramzan introduced additional techniques for speeding up their constructions. The techniques involve ignoring certain parts of the computation, while still retaining the necessary statistical properties for secure message authentication. They showed that we can ignore many of the carry bits in the computation of Square Hash and still get very strong performance for cryptographic applications.

An optimaized version of SQH, SQH_E is appeared in [5].

Theorem 2. *Let l be the word size on which you are computing and w is the total number of words needed to store x_i. Then SQH_E is an ϵ-almost-Δ-universal family of hash functions with $\epsilon \leq (6 \cdot \prod_{i=1}^{w} (4i + 1)^2)/2^{lw}$.*

3.3 NH

The novelty of this method is that all the arithmetic is "arithmetic that computers like to do" - no finite fields or non-trivial modular reductions come into the picture. A family of functions (with domain $A \subseteq \{0,1\}^*$ and range $B \subseteq \{0,1\}^*$) is a set of functions $H = \{h : A \to B\}$ endowed withe some distribution.

Fix an even $n \geq 2$(the "blocksize") and a number $w \geq 1$(the "wordsize"). The domain is $A = \{0,1\}^{2w} \cup \{0,1\}^{4w} \cup \cdots \cup \{0,1\}^{nw}$ and the range is $B = \{0,1\}^{2w}$. Each function in NH is named by an nw-bit string X; a random function in NH is given by a random nw-bit string X. We write the function indicated by X as $NH_X(\cdot)$.

Let U_w and U_{2w} represent the sets $\{0, ..., 2^w - 1\}$ and $\{0, ..., 2^{2w} - 1\}$, respectively. Arithmetic done modulo 2^w returns a result in U_w; arithmetic done 2^{2w} returns a result in U_{2w}. For integers x, y let $(x +_w y)$ denote $(x + y) \bmod 2^w$.

Let $M \in A$ and denote $M = M_1 \cdots M_l$, where $|M_1| = \cdots = |M_l| = w$. Similary, let $X \in \{0,1\}^{nw}$ and denote $X = X_1 \cdots X_n$, where $|X_1| = \cdots = |X_n| = w$. Then $NH_X(M)$ is defined as

$$NH_X(M) = \sum_{i=1}^{l/2} (x_{2i-1} +_w m_{2i-1}) \cdot \ (x_{2i} +_w m_{2i}) \ mod \ 2^{2w}$$

where $m_i \in U_w$ is the number that M_i represents, where $x_i \in U_w$ is the number that X_i represents, and the right-hand side of the above equation is understood to name the $2w$-bit string which represents the U_{2w}-valued integer result.

4 Proposed Hash Function

We propose a new hash function based on the construction of SQH and NH. We call it SNH. Fix an even $n \geq 2$(the "blocksize") and a number $w \geq 1$(the "wordsize").

Let U_{2w} and U_{4w} represent the sets $\{0, ..., 2^{2w} - 1\}$ and $\{0, ..., 2^{4w} - 1\}$, respectively. Arithmetic done modulo 2^{2w} returns a result in U_{2w}; arithmetic done 2^{4w} returns a result in U_{4w}.

Definition 6. *Let $M \in A$ and denote $M = M_1 \cdots M_l$, where $|M_1| = \cdots = |M_l| = w$. Similary, let $X \in \{0,1\}^{nw}$ and denote $X = X_1 \cdots X_n$, where $|X_1| = \cdots = |X_n| = w$. Then $SNH_X(M)$ is defined as*

$$SNH_X(M) \ = \sum_{i=1}^{l/2} ((x_{2i-1} + m_{2i-1})^2 \ mod \ 2^{2w}) \cdot ((x_{2i} + m_{2i})^2 \ mod \ 2^{2w}) \ mod \ 2^{4w}$$

where $m_i \in U_w$ is the number that M_i represents, where $x_i \in U_w$ is the number that X_i represents, and the right-hand side of the above equation is understood to name the $4w$-bit string which represents the U_{4w}-valued integer result. For integers x, y let $(x +_{2w} y)^2$ denote $(x + y)^2 \bmod 2^{2w}$.

4.1 Analysis

The following theorem bounds the collision probability of SNH.

Theorem 3. *For any even $n \geq 2$ and $w \geq 1$, SNH is 2^{-w}-almost-universal hash function on equal-length strings.*

Proof. Let M, M' be distinct members of the domain A with $|M| = |M'|$. We are required to show $Pr_X[SNH_X(M) = SNH_X(M')] \leq 2^{-w}$. Converting the message and key strings to n-vectors of w-bit words we invoke the definition of SNH to restate our goal as showing that

$$\begin{array}{ll} Pr[& \sum_{i=1}^{l/2}(x_{2i-1} +_{2w} m_{2i-1})^2(x_{2i} +_{2w} m_{2i})^2 \\ = & \sum_{i=1}^{l/2}(x_{2i-1} +_{2w} m'_{2i-1})^2(x_{2i} +_{2w} m'_{2i})^2] \end{array}$$

is no more than 2^{-w} where the probability is taken over uniform choices of $(x_1, \cdots, x_n)$ with each $x_i \in U_w$. Above all arithmetic is carried out in $Z/2^{4w}$.

Since M and M' are distinct, $m_i \neq m'_i$ for some $(1 \leq i \leq n)$. Since addition and multiplication in a ring are commutative, we lose no generality in assuming $m_2 \neq m'_2$. We now prove that for any choice of $x_2, , x_n$ we have $Pr_{x_1 \in U_w}[(m_1 +_{2w} x_1)^2(m_2 +_{2w} x_2)^2 + \sum_{i=2}^{l/2}(m_{2i-1} +_{2w} x_{2i-1})^2(m_{2i} +_{2w} x_{2i})^2 = (m'_1 +_{2w} x_1)^2(m'_2 +_{2w} x_2)^2 + \sum_{i=2}^{l/2}(m'_{2i-1} +_{2w} x_{2i-1})^2(m'_{2i} +_{2w} x_{2i})^2] \leq 2^{-w}$, which will imply the theorem. Collecting up the summations, let

$$\begin{array}{ll} y & = \sum_{i=2}^{l/2}(m_{2i-1} +_{2w} x_{2i-1})^2(m_{2i} +_{2w} x_{2i})^2 \\ & - \sum_{i=2}^{l/2}(m'_{2i-1} +_{2w} x_{2i-1})^2(m'_{2i} +_{2w} x_{2i})^2 \end{array}$$

and let $c = (m_2 +_{2w} x_2)^2$ and $c' = (m'_2 +_{2w} x_2)^2$. Note that c and c' are in U_{2w}, and since $m_2 \neq m'_2$, we know $c \neq c'$. We rewrite the above probability as

$$Pr_{x_1 \in U_w}[c(m_1 +_{2w} x_1)^2 - c'(m'_1 +_{2w} x_1)^2 + y = 0] \leq 2^{-w}.$$

In Lemma 1 below, we prove that there can be at most one x_1 in U_w satisfying $c(m_1 +_{2w} x_1)^2 - c'(m'_1 +_{2w} x_1)^2 + y = 0$, yielding the desired bounded. □

Lemma 1. *Let c and c' be distinct values from U_{2w}. Then for any $m, m' \in U_w$ and any $y \in U_{4w}$ there exists at most one $x \in U_w$ such that $c(m_1 +_{2w} x_1)^2 = c'(m'_1 +_{2w} x_1)^2 + y$ in $Z/2^{4w}$.*

Proof. $c(m_1 +_{2w} x_1)^2 = c'(m'_1 +_{2w} x_1)^2 + y$ in $Z/2^{4w}$ is satisfied for $m \geq m'$ and $m \leq m'$, then we note that it is sufficient to prove the case where $m = 0$. We proceed to therefore prove that for any $c, c' \in U_{2w}$, $m' \in U_w$ and any $y \in U_{4w}$ there is at most one $x \in U_w$ such that $cx^2 = c'(x + m')^2 + y$ in $Z/2^{4w}$. Since $x, m' < 2^w$, we know that $(x +_{2w} m')^2$ is either $(x + m')^2$ or $(x + m')^2 - 2^{2w}$, depending on whether $(x +_{2w} m')^2 < 2^{2w}$ or $(x +_{2w} m')^2 \geq 2^{2w}$ respectively. So now we have

$$\begin{array}{ll} x\{x(c - c') - 2m'c'\} & = c'm'^2 + y \\ and & (x + m')^2 < 2^{2w} \end{array} \quad (1)$$

$$\begin{array}{ll} x\{x(c - c') - 2m'c'\} & = c'm'^2 + y - c'2^{2w} \\ and & (x + m')^2 \geq 2^{2w} \end{array} \quad (2)$$

Lemma 2, presented next, shows that there is at most one solution to each of the equations above. The remainder of the proof is devoted to showing there cannot exist $x = x_1 \in U_w$ satisfying (1) and $x = x_2 \in U_w$ satisfying (2) in $Z/2^{4w}$. Suppose such a x_1 and x_2 did exist.

Then we have $(x_1+m')^2 < 2^{2w}$ with $x_1\{x_1(c-c') - 2m'c'\} = c'm'^2 + y$ and $(x_2+m')^2 \geq 2^{2w}$ with $x_2\{x_2(c-c') - 2m'c'\} = c'm'^2 + y - c'2^{2w}$. Subtracting the former from the latter yields $(x_2-x_1)\{(x_2+x_1)(c'-c) - 2m'c'\} = c'2^{2w}$. We show that this equation has no solution in $Z/2^{4w}$. There are cases:

Case 1 : $c' > c$. Since $x_2 - x_1 < 2^w$, $x_2 + x_1 < 2^w$, $c' - c < c'$, and $m' \neq 0$, and so necessaruly $(x_2-x_1)\{(x_2+x_1)(c'-c) - 2m'c'\} < c'2^{2w}$.

Case 2 : $c' < c$. $(x_2-x_1)\{(x_2+x_1)(c-c') - 2m'c'\} = -c'2^{2w}$ has no solution in $Z/2^{4w}$. As before, we convert to Z, to yield $(x_2-x_1)\{(x_2+x_1)(c-c') - 2m'c'\} = 2^{4w} - c'2^{2w}$. But again $x_2 - x_1 < 2^w$, $x_2 + x_1 < 2^w$, $c - c' < 2^{2w} - c'$, and $m' \neq 0$, so $(x_2-x_1)\{(x_2+x_1)(c-c') - 2m'c'\} < 2^{4w} - c'2^{2w}$.

Let $D_{2w} = \{-2^{2w} + 2^{w+1} - 1, \cdots, 2^{2w} - 2^{w+1} + 1\}$ be the values attainable from a difference of any two elements of U_{2w}. The following lemma completes the proof.

Lemma 2. *Let $x \in D_{2w}$ be nonzero. Then for any $y \in U_{4w}$, there exists at most one $a \in U_{2w}$ such that $ax = y$ in $Z/2^{4w}$.*

Proof. Suppose there were two distinct elements $a, a' \in U_{2w}$ such that $ax = y$ and $a'x = y$. Then $ax = a'x$ so $x(a-a') = 0$. Since x is nonzero and a and a' are distinct, the foregoing product $2^{4w}k$ for nonzero k. But x and $(a-a')$ are in D_{2w}, and therefore their products is in $\{-2^{4w} + 2^{3w+2} - 2^{2w+2} - 2^{2w+1} + 2^{w+2} - 1, \cdots, 2^{4w} - 2^{3w+2} + 2^{2w+2} + 2^{2w+1} - 2^{w+2} + 1\}$, which contains no multiples of 2^{4w} other than 0. □

5 Conclusions

We described a new family of universal hash functions based on construction of SQH and NH. And we show proposed hash function is almost universal hash functions whose collision probability is a third of SQH. It means that proposed hash function is more secure than SQH.

In the future, we want to implement the proposed hash function and to compare the speed of proposed hash function and those of SQH and NH.

References

[1] M.Atici, and D.R.Stinson, "Universal Hashing and Multiple Authentication," In proc. CRYPTO'96, Lecture Notes in Computer Science. Springer-Verlag, pp.16-29, 1996.

[2] Z.Black, S.Halevi, H.Krawczyk, T.Krovetz, and P.Rogaway, "UMAC: Fast and Secure Message Authentication,", In proc. CRYPTO'99, Lecture Notes in Computer Science. Springer-Verlag, pp.216-233, 1999.

[3] L.Carter and M.Wegman, "Universal Hash functions," Journal of Computer and System Sciences, 18: pp.143-154, 1979.

[4] L.Carter and M.Wegman, "New hash functions and their use in authentication and set equality," Journal of Computer and System Sciences, vol.22, pp.265-279, 1981.

[5] M.Etzel, S.Patel, and Z.Ramzan, "Square Hash: Fast Message Authentication cia Optimized Universal Hash Functions,", In proc. CRYPTO'99, Lecture Notes in Computer Science. Springer-Verlag, pp.234-251, 1999.

[6] S.Halevi, and H.Krawczyk, "MMH: Software message authentication in the Gbit/second rates,", In proc. of the 4th Workshop on Fast Software Encryption, Springer-Verlag, pp.172-189, 1997.

[7] T.Johansson, "Bucket hashing with small key size," In Proc. EUROCRYPT'97, Lecture Notes in Computer Sciences. Springer-Verlag, 1997.

[8] H.Krawczyk, "LFSR-based Hashing and Authentication," In Proc. CRYPTO'94, Lecture Notes in Computer Science. Springer-Verlag,pp.129-139, 1994.

[9] A.J.Menezes, P.C.van Oorschot, and S.A.Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.

[10] M.Naslund, "Universal Hash Functions & Hard Core Bits," In Proc. EUROCRYPT'95, Lecture Notes in Computer Science. Springer-Verlag, pp.356-366, 1995.

[11] W.Nevelsteen, and B.Preneel, "Software performance of universal hash functions," In proc. EUROCRYPT'99, Lecture Notes in Computer Science. Springer-Verlag, pp.24-41, 1999.

[12] P.Rogaway, "Bucket hashing and its application to fast message authentication," In Proc. CRYPTO'95, Lecture Notes in Computer Science, Springer-Verlag, pp.313-328, 1995.

[13] D.R.Stinson, "Universal hashing and authentication codes," In Proc. CRYPTO'91, Lecture Notes in Computer Sciences, Springer-Verlag, pp.1-10, 1991.

[14] V.Shoup, "On fast and provably secure message authentication based on universal hashing," In Proc. CRYPTO'96, Lecture Notes in Computer Sciences, Springer-Verlag, pp.313-327, 1996.