

A Topologically Simple Keyed Hash Function Based on Circular Chaotic Sinusoidal Map Network

N. Jiteurtragool, P. Ketthong, C. Wannaboon, and W. San-Um

Intelligent Electronics Systems Research Laboratory

Faculty of Engineering, Thai-Nichi Institute of Technology, Bangkok, Thailand, 10250

nattagit.j@gmail.com, patinya.ket@gmail.com, chatchai.wann@gmail.com, wimol@tni.ac.th

Abstract—this paper presents a new high-speed topologically-simple keyed hash function based on a sinusoidal map, which offers high-degree of chaos over most parameter space regions, and a circular network topology with eight sinusoidal maps. Hash function operations involve an initial stage when the sinusoidal map accepts input message and initial conditions, and a hashing stage when alterable-length hash values are generated iteratively. Simulations of nonlinear dynamics are described in terms of Cobweb map, Lyapunov exponent, and two-dimensional bifurcation structures. Performances are evaluated in terms of original messages and condition changes, statistical analyses, collision analyses, and speed analysis. The proposed hash function offers high-speed operation of less than 2.5 ms, the mean changed probabilities fall in the region of [49.09, 50.06] which is very close to 50%, and the mean changed bit number is also close to a half of hash value length. The collision tests reveal the average mean of 1387, 1647, and 2710 for the hash values of 128, 160, and 256 bits, respectively. The collision resistance is enhanced, comparing to MD5, SHA1, and some of other chaos-based approaches.

Keywords— Hash function, Chaotic Sinusoidal map.

I. INTRODUCTION

Advancements in communications have led to a great demand for reliable and robust information security, which can be achieved through cryptography. Significant features of cryptography involve data confidentiality, verification of data integrity, authentication and non-repudiation of origin [1]. A hash function is a key technology that plays an important role in advanced cryptography. Such a hash function typically encodes an arbitrary length input message into a hash value with fixed length. In general, two categories of hash functions are unkeyed and keyed hash functions whose specifications dictate only an input message and both input message and security keys, respectively. In addition, desirable performances of hash functions are high possibility of collision resistance and high security against preimage and second-preimage attacks.

Conventional hash functions such as MD4 and MD5, which are widely used in the software industry for verification of integrity of electronically transmitted files and SHA-1, which is widely used in security applications and protocols, have been realized based mainly on logical operations or multi-round iterations, and therefore hashing process efficiency depends upon inherent ciphers, necessarily requiring complicated computation processes. Recently, it has

been reported through the investigation on the collision frequencies that such conventional hash functions have many undiscovered flaws [2]. Consequently, multiple-block-length hash functions have alternatively been suggested [3-5]. Nonetheless, designs and implementation of such multiple-block-length hash function are relatively complicated in terms of security and efficiency.

Chaos-based hash algorithms have been of much interest as an alternative to that of conventional hash functions due to distinctive properties in pseudo-randomness and sensitivity to initial conditions and control parameters. Several chaos-based hash function algorithms have been proposed recently [6-9]. Although these algorithms have offered satisfied statistical performances in terms of well-accepted statistical performance and high collision resistance, small key space, flexibility problem, low performance, and weak security functions are some drawbacks that motivate researchers to participate in designing new fast, efficient and secure hash functions [1]. In addition, the structural topologies of such algorithms are somewhat complex as evident from multiple maps, multi-stage connections, or multiple feedback loops, leading to complicated signal processing and extensive iteration time.

This paper therefore presents a new hash function that exploits complex nonlinear dynamics of a sinusoidal map in order to increase the entropy in hash value generation processes. Unlike unimodal or piece-wise-linear chaotic maps in which chaoticity and parameter spaces are bounded, the sinusoidal map offer high-degree of chaos over most regions of parameter spaces. The proposed hash function realizes sufficient connection in a circular network type in which the output depends only on its previous state and another input from the successive maps. The proposed hash function comprises eight sinusoidal chaotic maps, accepts any arbitrary length of input message, and generates alterable length output bit stream. Initial conditions are employed as security keys. The rest of this paper is organized as follows. Section II demonstrates nonlinear dynamics of absolute sine map through Cobweb map, Lyapunov exponent, and two-dimensional bifurcation structures. Section III proposes the keyed hash function using sinusoidal map in a circular network. Section IV provides simulation results and discussions, involving original message and sensitivity, statistical analysis, collision resistance analysis. Conclusions are drawn in Section V.

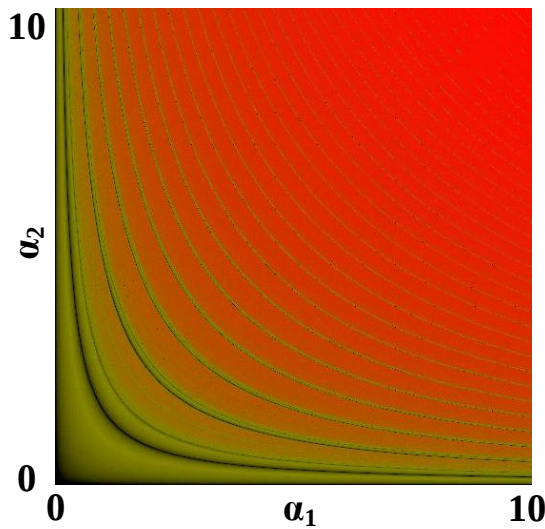


Figure 1. Bifurcation structure of parameters α_1 and α_2 over the parameter space $\alpha_1 = [0,10]$ and $\alpha_2 = [0,10]$.

II. NONLINEAR DYNAMICS ANALYSIS OF SINUSOIDAL MAP

This paper realizes the absolute sine map which is a variant of the original sine map as follows;

$$x_{n+1} = \alpha_1 \text{abs}(\sin(\alpha_2 x_n)) \quad (1)$$

where α_1 and α_2 can be considered as the parameters associated with amplitude and frequency, respectively. Since chaotic behaviors of chaotic maps determine global performance of hashing processes, Lyapunov exponent (LE) has been realized as a quantitative measure of chaoticity. The LE is defined as a quantity that characterizes the rate of separation of infinitesimally close trajectories and is given by

$$LE = \lim_{n \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \log_2 \frac{dX_{n+1}}{dX_n} \quad (2)$$

where N is the number of iterations. Typically, the positive LE indicates chaotic behaviors. Fig.1 shows the bifurcation structure of parameters α_1 and α_2 over the parameter space $\alpha_1 = [0,10]$ and $\alpha_2 = [0,10]$. It can be seen from Fig.1 that the increase in parameters α_1 and α_2 results in a greater value of LE. Moreover, the increases in either parameters α_1 or α_2 offer similar results. Therefore, this paper keeps α_2 at unity, and exploits α_1 as a bifurcation parameter.

Fig.2 (a) and (b) shows Lyapunov exponent versus the parameter α_1 and the bifurcation diagram exhibiting a period-doubling route to chaos of the peak of x_n versus parameter α_1 . Fig. 3 shows the Cobweb plot using $\alpha_1 = 10$. Fig.4 shows the time-domain waveforms at different values of initial conditions, i.e. 0.000100, 0.000110, 0.000101. It can be considered from Figs. 2 and 3 that the absolute sine map exhibits complex chaotic behaviors. In particular, Fig.4 emphasizes that chaotic map is sensitive to the initial condition since the small changes in initial conditions results in large different behaviors. As a result, this paper exploits such nonlinear behaviors for constructing a hash function.

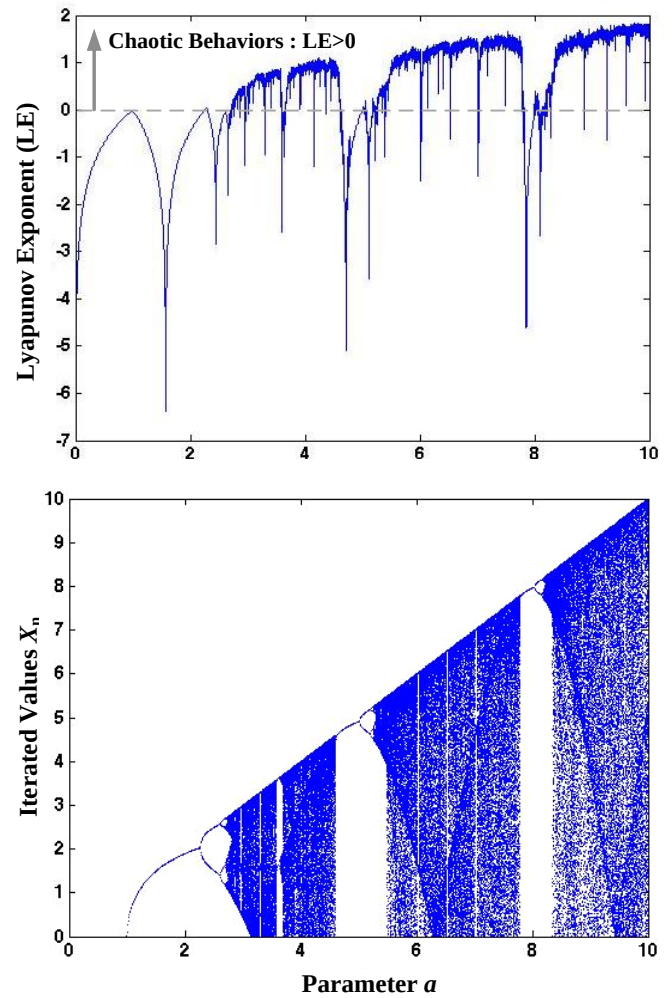


Figure 2. (a) Lyapunov exponent versus the parameter a . (b) The bifurcation diagram exhibiting a period-doubling route to chaos of the peak of x_n versus parameter a .

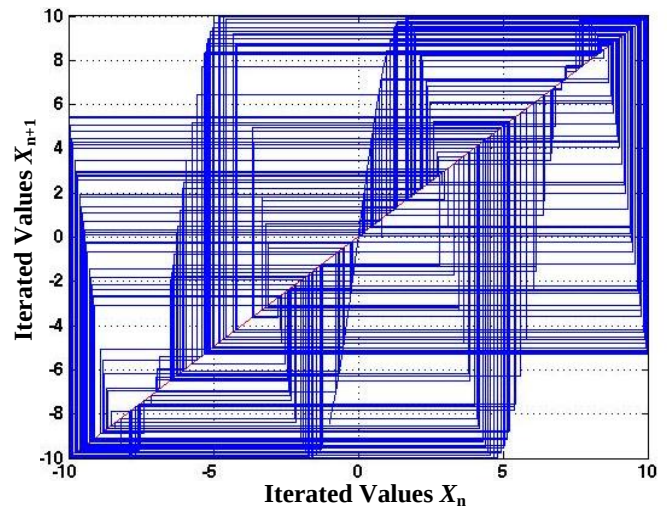


Figure 3. Cobweb plot where parameter $\alpha_1 = 10$.

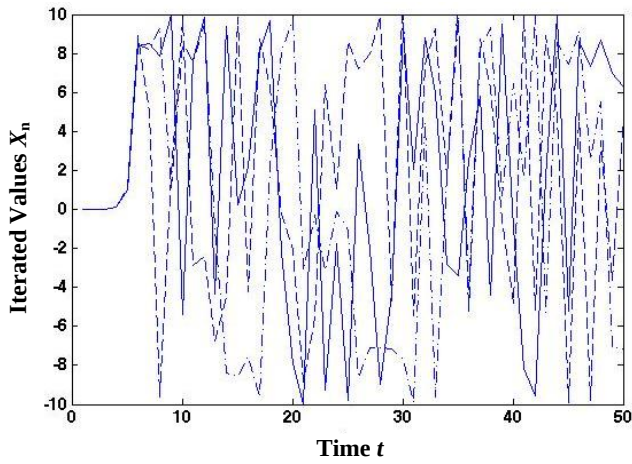


Figure 4. Time series plots where $x_0 = [0.000100, 0.000110, 0.000101]$.

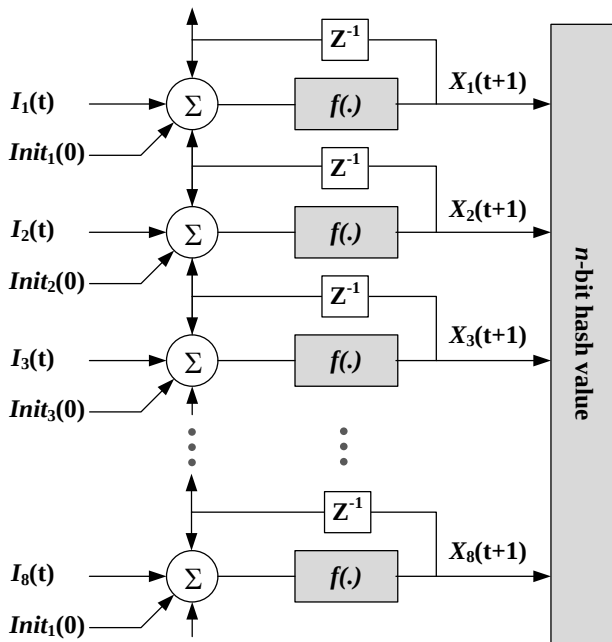


Figure 5. Structure of the proposed hash function.

III. PROPOSED KEYED HASH FUNCTION BASED ON CIRCULAR CHAOTIC SINUSOIDAL MAP NETWORK

The design purpose is to optimize for simple structure with few numbers of chaotic maps, but offer high complexity. The proposed hash function realizes sufficient connection in a circular network type in which the output depends only on its previous state and another input from the successive maps. Fig.5 shows the proposed hash function comprises eight sinusoidal chaotic maps connected in a circular network. Each chaotic map accepts any arbitrary length of input message, and generates alterable length output bit stream. Initial conditions are employed as security keys.

Fig 6 shows the pseudo codes of the proposed hash function algorithm. The input is a message of length L (M), a secret key ($X(i,0)$), and a hash value size (n) while the output is an

Input : M (a message of length L), $X(i,0)$ (a secret key), n (a hash value size)
Output : H (n -bits hash value)

Begin

if The message length, L , is not multiple of 8 **then**
 >> Append the tail of M with '0' for $L \pmod{8} = 0$
end if
 >> Divide the input message M into N sub-block of length 8 (Defined by S)
 >> $i, k, t = 1$

First Stage (Input Stage)

while ($t < N$)
 while ($i < 7$)
 >> $X(i, t) = 2^n \mid \sin(M(k) + X(i+1, t-1)) \mid$
 >> $k = k+1$
 >> $i = i+1$
 end
 >> $X(8, t) = 2^n \mid \sin(M(k) + X(1, t-1)) \mid$
 >> $k = k+1$
 >> $t = t+1$
end
 >> $i = 1$

Second Stage

for ($t < N+8$)
 while ($i < 7$)
 >> $X(i, t) = 2^n \mid \sin(X(i, t-1) + X(i+1, t-1)) \mid$
 >> $i = i+1$
 end
 >> $X(8, t) = 2^n \mid \sin(X(8, t-1) + X(1, t-1)) \mid$
 >> $t = t+1$
end
 >> $H = \text{Map } X \text{ into integer with interval } [0, 2^n]$
 >> **return** H

Figure 6. Pseudo codes of the proposed hash function algorithm.

n -bit hash value H . Two stages of hashing procedures include the initial stage and the hashing stage. For input stage, no output is available at the beginning available and consequently one iteration is required for the absolute sine map in order to employ the initial conditions as a feedback output. The input message is also applied to the chaotic maps in this input stage. Once the first stage is performed, the second stage generates the hash value iteratively through the delayed self feedback values and another value from the successive chaotic maps. It has been investigated that the minimum number of iterations equals to eight rounds, which is equal to the number of chaotic maps realizes, implying that the operation is completely circulated in one loop. It can be considered that the proposed hash function algorithm is relatively simple in terms of topology, but the complexity is mainly determined by the nonlinear dynamics of the absolute sine maps.

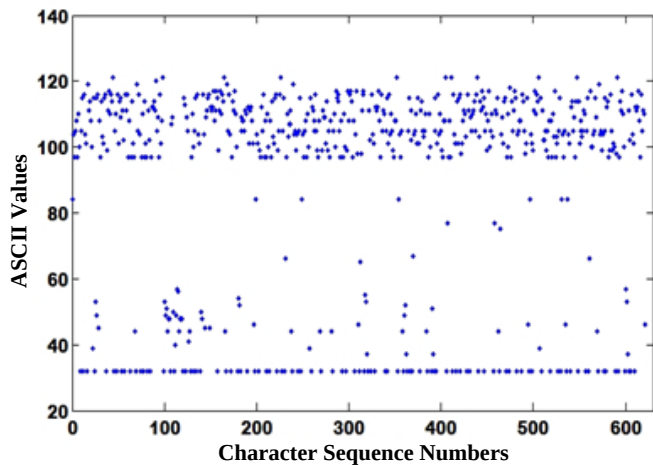


Figure 7. Distributions of the original message in ASCII code.

IV. SIMULATION RESULTS AND DISCUSSIONS

A. Original Messages and Condition Changes

Simulations of the proposed hash function algorithm has been performed in MATLAB using the initial conditions $x_1(0)$, $x_2(0)$, $x_3(0)$, $x_4(0)$, $x_5(0)$, $x_6(0)$, $x_7(0)$, and $x_8(0)$ of respective values 2110.3669, 3520.2235, 8080.4889, 7089.78, 2814.5919, 1670.9962, 4019.1227, and 2201.8885. Simulation of input message in ASCII code has been performed using following paragraph:

“Thailand is the world's 51st-largest country in terms of total area, with an area of approximately 513,000 km² (198,000 sq mi), and is the 20th-most-populous country, with around 64 million people. The capital and largest city is Bangkok, which is Thailand's political, commercial, industrial and cultural hub. About 75% of the population is ethnically Thai, 14% is of Chinese origin, and 3% is ethnically Malay the rest belong to minority groups including Mons, Khmers and various hill tribes. The country's official language is Thai. The primary religion is Buddhism, which is practiced by around 95% of the population.”

Fig. 7 shows the distributions of ASCII code of the original message, showing that the letters and symbols in ASCII Code are valued between 32 and 127 and the message are localized within a small area from approximately 90 to 127. It is also noticeable that the space also influences the ASCII code as shown in a bottom line at the value of 32. In addition, the input plain image has also been investigated using the standard grey-scale Lena image with 256×256 image size as depicted in Fig. 8. In order to evaluate the sensitivity of hash values to the original text message and image, simulations have been performed under seven particular conditions represented by C_1 to C_7 as follows;

- C1:** The original text message.
- C2:** Replace the first character of the original message by ‘t’.
- C3:** Add a space to the end of the original message.



Figure 8. the standard grey-scale Lena image with 256×256 image size



Figure 9. Binary sequences of hash values of the seven conditions.

- C4:** Delete the character ‘B’ in the word ‘Bangkok’.
- C5:** The Original image.
- C6:** Add 1 to the gray value of the pixel located at the upper left corner.
- C7:** Subtract 1 to the gray value of the pixel located at the upper right corner.

TABLE 1. SUMMARY OF STATISTICAL MEASURE METHODS.

Statistical measure methods		Formulas
1	Minimum number of changed bits	$B_{\min} = \min \left(\{B_i\}_{i=1}^N \right)$
2	Maximum changed bit number	$B_{\max} = \max \left(\{B_i\}_{i=1}^N \right)$
3	Mean changed bit number	$\bar{B} = \frac{1}{N} \sum_{i=1}^N B_i$
4	Mean changed probability	$P = \frac{\bar{B}}{n} \times 100\%$
5	Standard deviation of the changed bit number	$\Delta B = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (B_i - \bar{B})^2}$
6	Standard deviation	$\Delta P = \sqrt{\frac{1}{N-1} \sum_{i=1}^N \left(\frac{B_i}{n} - P \right)^2} \times 100\%$

Fig.9 shows the Hash values for those of seven conditions. corresponding graphical display of binary sequences. It can be considered that a very small difference in the original message causes large changes in the final hash value. Moreover, the output has values are completely difference. The proposed has function algorithm is very effective as reflected in Fig.7.

B. Statistical Analyses

Typically, hashing algorithm performances can be measured through diffusion and confusion [10]. On the one hand, diffusion is the spreading out of the influence of a single bit of original message over entire cipher bits in order to hide statistical structures of the original message. On the other hand, confusion is the use of transformations that complicate dependence of the statistics of cipher text on the statistics of plaintext. Hash function requires the message to diffuse its influence into the whole hash space. This indicates that the correlation between the message and the corresponding hash value should be as close as possible.

The statistical analysis procedures are as follows; First, a random message of length $L = n \times 50$ is generated and its n -bit hash value is computed. Second, a single bit of the original message is randomly selected and toggled, and a new n -bit hash value is computed. The two n -bit hash values are then compared with each other in order to obtain the number of changed bits. Simulations are repeated N times for $N = 256, 512, 1024, 2048$, and $10,000$ for hash values of length n , where $n = 128, 160$, and 256 . Table 1 summarizes six measures of statistical tests where N represents the total number of tests, n represents the hash value sizes, and B_i denotes the number of changed bits in the i^{th} test. The test results are summarized in Tables 2-4 for different hash value sizes and test numbers. Fig. 10 shows distribution of the number of the changes bits (B_i) for $n = 256$ and $N = 10000$. Fig.11 shows the histogram distribution of the number of changed bits (B_i) for $n = 256$ and $N = 10000$. For all value of N , the mean changed bit number $\bar{B}$ and the mean changed probability P are very close to ideal value $n/2$ bits and 50%, respectively. The small value of ΔB and ΔP indicates that confusion and diffusion capability of the proposed hash function algorithm is stable.

TABLE 2. STATISTICAL RESULTS FOR A 128-BIT HASH VALUE.

	N=256	N=512	N=1024	N=2048	N=10000
$B_{\min}$	48	44	45	43	48
$B_{\max}$	80	81	87	81	83
$\bar{B}$	64.07	63.08	62.87	62.84	62.89
P (%)	50.05	49.28	49.11	49.09	49.13
ΔB	5.70	6.04	5.98	5.63	5.70
ΔP (%)	4.45	4.72	4.68	4.40	4.46

TABLE 3. STATISTICAL RESULT FOR A 160-BIT HASH VALUE.

	N=256	N=512	N=1024	N=2048	N=10000
$B_{\min}$	64	62	59	61	57
$B_{\max}$	96	97	101	99	101
$\bar{B}$	79.43	79.04	78.87	79.09	78.89
P (%)	49.65	49.40	49.29	49.43	49.31
ΔB	6.02	6.39	6.25	6.37	6.47
ΔP (%)	3.76	3.99	3.91	3.98	4.05

TABLE 4. STATISTICAL RESULT FOR A 256-BIT HASH VALUE.

	N=256	N=512	N=1024	N=2048	N=10000
$B_{\min}$	103	98	105	101	96
$B_{\max}$	146	148	155	153	157
$\bar{B}$	126.73	126.37	127.33	126.75	127.04
P (%)	49.60	49.36	49.74	49.51	49.63
ΔB	7.97	8.10	8.18	7.98	8.14
ΔP (%)	3.11	3.16	3.19	3.12	3.18

C. Collision Analyses

Quantitative analysis on collision resistance is generally performed in two kinds of test methods. First, the hash value for a message randomly chosen is generated and stored in ASCII format. A bit in the message is selected randomly and toggled, and a new hash value is subsequently generated and also stored in ASCII format. Two hash values are compared, and the number of ASCII characters with the same value at the same location in the hash value is counted. The absolute difference between two hash values is computed based on

$$d = \sum_{i=1}^N |dec(m_i) - dec(m'_i)| \quad (1)$$

where m_i and m'_i are the i^{th} ASCII character of the original and the new hash value respectively, while the $dec(.)$ converts an ASCII character to its decimal value. Table 5 summarizes and compares the minimum, maximum and mean values of the absolute difference of original and new hash values. It can be considered from Table 5 that, for $n=128$ and $n=160$, the proposed hash function algorithm offers better results than that of existing algorithms, i.e. MD5 and SHA-1, and comparable results to those algorithms based on chaotic maps.

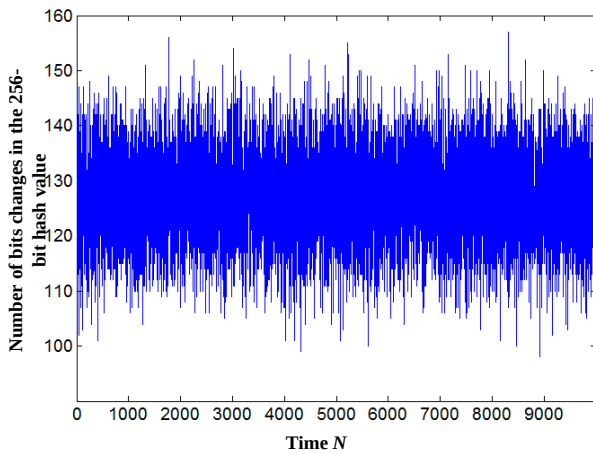


Figure 10. Plot distribution of the number of the changes bits (B_i) for $n = 256$ and $N = 10000$

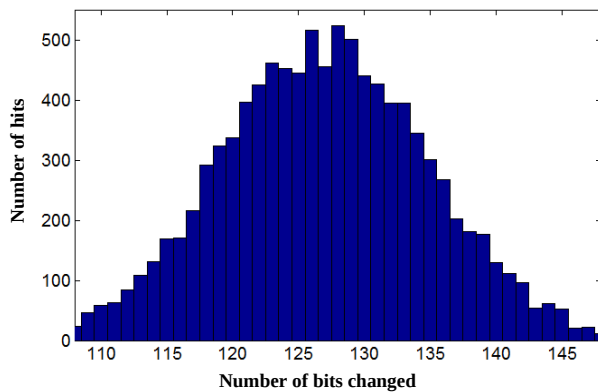


Figure 11. PHistogram distribution of the number of changed bits (B_i) for $n = 256$ and $N = 10000$.

TABLE 5. ABSOLUTE DIFFERENCE FOR HASH VALUES FOR $N = 10000$.

Absolute difference	Min.	Max.	Mean
MD5(128bit)	590	2074	1304
SHA-1(160bit)	795	2730	1603
Zhang's scheme[10](128bit)	565	2022	1257
Kanso's scheme[11](128bit)	737	2320	1494
Proposed scheme (128bit)	549	2590	1387
Proposed scheme (160bit)	602	2702	1647
Proposed scheme (256bit)	1509	4005	2710

D. Resistance to Birthday Attack

The proposed hash function is robust against birthday attack since the investigating length of hash value sequences are 128, 160, and 256. In fact, the algorithm is flexible so that the length of the hash value can be adjusted. For instance, if the hash value size is set to 256, the difficulty of the attack is 2^{128} . A greater size of the generated hash value makes this kind of attack almost impossible.

CONCLUSIONS

The new high-speed topologically-simple keyed hash function based on a sinusoidal map, which offers high-degree of chaos over most parameter space regions, and a circular network topology with eight sinusoidal maps have been

presented. Hash function operations have been designed for the initial stage when the sinusoidal map accepts input message and initial conditions, and the hashing stage when alterable-length hash values are generated iteratively. Simulations of nonlinear dynamics have been described in terms of Cobweb map, Lyapunov exponent, and two-dimensional bifurcation structures. Performances are evaluated in terms of original messages and condition changes, statistical analyses, collision analyses, and speed analysis. Statistical performances have indicated that the mean changed probability is very close to 50%, and the mean changed bit number is also very close to a half of hash value length. The collision tests have also indicated that the collision resistance of the proposed hash function algorithm is enhanced compared with MD5, SHA1, and some previous chaos-based hash function algorithms. This work has offered high-speed hashing operations and high resistance to collision with well-accepted statistical tests through the realization of circular chaotic sinusoidal map network.

REFERENCES

- [1] A. Kanso, M. Ghebleh, "A fast and efficient chaos-based keyed hash function", *Commun Nonlinear Sci Numer Simulat*, Vol 18, 2013, p. 109–123
- [2] Huaqian Yang, Kwok-Wo Wong, Xiaofeng Liao, Yong Wang, Degang Yang, "One-way hash function construction based on chaotic map network", *Chaos, Solitons and Fractals*, Vol. 41, 2009, p. 2566–2574
- [3] B.O. Brachtel, D. Coppersmith, M.M. Hyden, S.M. Matyas, C.H. Meyer, J. Oseas, S. Pilpel, M. Schilling, "Data authentication using modification detection codes based on a public one way encryption function", U.S. Patent Number 4,908,861, March 13, 1990.
- [4] Hohl W, Lai X, Meier T, Waldvogel C. "Security of iterated hash functions based on block ciphers", *Advances in cryptology-crypto93 LNCS*, vol.773, Springer-Verlag, 1994, p. 379–390.
- [5] Knudsen LR, Preneel B. "Fast and secure hashing based on codes", *Advances in cryptology-crypto'97 LNCS*, Vol. 1294, Springer-Verlag, 1997, p. 485–498.
- [6] Qing Zhou, Kwok-Wo Wong, Xiaofeng Liao, Tao Xiang, Yue Hu. "Parallel image encryption algorithm based on discretized chaotic map", *Chaos Solitons & Fractals* 2008, Vol 38(4), p. 1081–1092.
- [7] Nien HH, Huang CK, Changchien SK, Shieh HW, Chen CT, Tuan YY. "Digital color image encoding and decoding using a novel chaotic random generator", *Chaos, Solitons & Fractals*, 2007, Vol. 32, p. 1070–1080.
- [8] Behnia S, Akhshani A, Akhavan A, Mahmodi H. "Applications of tripled chaotic maps in cryptography", *Chaos Solitons & Fractals*, 2009, Vol. 40, p. 505–519.
- [9] Wong K. "A combined chaotic cryptographic and hashing scheme", *Phys Lett A*, 2003, Vol. 307, p. 292–298.
- [10] Zhang J, Wang X, Zhang W. "Chaotic keyed hash function based on feedforward-feedback nonlinear digital filter". *Phys Lett A*, 2007, Vol. 362, p. 439–48.
- [11] A. Kanso, H. Yahyaoui, M. Almulla, "Keyed hash function based on a chaotic map", *Information Sciences*, Vol. 186, 2012, p. 249–264.