A Robust Hash Function Using Cross-Coupled Chaotic Maps with Absolute-Valued Sinusoidal Nonlinearity

Wimol San-Um
Intelligent Electronic System Research Laboratory
Faculty of Engineering, Thai-Nichi Institute of Technology
1771/1 Pattanakarn Rd., Suanluang, Bangkok 10250, Thailand

Warakorn Srichavengsup
Computer Engineering, Robotics and Technology Laboratory
Faculty of Engineering, Thai-Nichi Institute of Technology
1771/1 Pattanakarn Rd., Suanluang, Bangkok 10250, Thailand

***Abstract*—This paper presents a compact and effective chaos-based keyed hash function implemented by a cross-coupled topology of chaotic maps, which employs absolute-value of sinusoidal nonlinearity, and offers robust chaotic regions over broad parameter spaces with high degree of randomness through chaoticity measurements using the Lyapunov exponent. Hash function operations involve an initial stage when the chaotic map accepts initial conditions and a hashing stage that accepts input messages and generates the alterable-length hash values. Hashing performances are evaluated in terms of original message condition changes, statistical analyses, and collision analyses. The results of hashing performances show that the mean changed probabilities are very close to 50%, and the mean number of bit changes is also close to a half of hash value lengths. The collision tests reveal the mean absolute difference of each character values for the hash values of 128, 160 and 256 bits are close to the ideal value of 85.43. The proposed keyed hash function enhances the collision resistance, comparing to MD5 and SHA1, and the other complicated chaos-based approaches. An implementation of hash function Android application is demonstrated.**

***Keywords*—Hash Function; Cross-Coupled Chaotic Map; Sinusoidal Nonlinearity; Information security; Authentication**

I. Introduction

The advancement in communication technologies have led to a great demand in reliable and robust information security, involving data confidentiality, verification of data integrity, authentication and non-repudiation of origin [1]. A hash function, which encodes an arbitrary length input message into a hash value with a fixed length, has played an important role in advanced information security, particularly including in cryptography and secure protocol methods. As for security purposes, the desirable performances of hash functions include high possibility of collision resistance and high security against preimage and second-preimage attacks.

The typical MD4, MD5, and SHA-1 hashing algorithms have extensively been realized in software industries for integrity verification of electronically transmitted files as well as security in protocols. Such typical hash functions are designed based on logical operations or multi-round iterations and therefore the hashing process efficiency depends upon inherent ciphers which necessarily require complicated computation processes. Moreover, it has been notified recently through the collision frequency analysis that those typical hash functions contain several undiscovered flaws [2]. In order to overcome such flaws, the multiple-block-length hash functions have been suggested [3-5]. Nonetheless, the implementation of such multiple-block-length hash functions is complicated in terms of security and computation processes.

As a ubiquitous aspect in nature, chaos is a deterministic nonlinear dynamical system that possesses distinctive properties, mainly involving pseudo-randomness and sensitivity to initial conditions and control parameters. With such properties, chaos-based hash algorithms have consequently been of much interest as an alternative to those of typical hash functions. Several chaos-based hash function algorithms have been proposed recently [6-9]. Despite the fact that these algorithms have offered satisfied statistical performances in terms of statistical performance and collision resistance, the difficulty in small key space, flexibility, low performance, and weak security functions are obstacles that elevate an attempt in designing efficient and secure hash functions. Furthermore, structural topologies of existing 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.

As for compact and effective chaos-based hash function implementations, this paper presents a new alternative in both chaotic map and hash function topology. The proposed chaotic map employs absolute-value of sinusoidal nonlinearity and offers robust chaotic regions over broad parameter spaces with high degree of randomness through chaoticity measurements using the Lyapunov exponent. The proposed hash function is implemented by a cross-coupled topology. Hash function operations involve an initial stage when the chaotic map accepts initial conditions and input messages in ASCII format, and a hashing stage that accepts input messages and generates the alterable-length hash values. Hashing performances are evaluated in terms of original message condition changes, statistical analyses, and collision analyses. The proposed keyed hash function enhances the collision resistance, comparing

to MD5 and SHA1, and is comparable to other complicated chaos-based approaches.

II. Proposed Chaotic Map using Absolute Sinusoidal Nonlinearity

A sinusoidal function typically contains an inherent infinite and complex nonlinearity described by an infinite Maclaurin series as follows;

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - ... \tag{1}$$

Such a sinusoidal function in (1) has therefore been utilized as a potential inherent nonlinearity in various chaotic maps. Unlike a single-modal chaotic map, i.e. a logistic map, or another family of multi-modal chaotic maps based on polynomial functions which comprises several mathematical terms, the sinusoidal function potentially provides complex chaotic time series with unique dynamical characteristics.

Table 1 summarizes related chaotic maps implemented based on sinusoidal functions. As for an attempt of using sinusoidal function in a particular application on a hash function design where the input is an ASCII code with values in the region of 32 to 126, the typical sine map in [10] limits the system parameter in the range of (0,4). Although the Iterative Chaotic Map with Infinite Collapses in [11] potentially offers an infinite parameter space, the output time series swing over the values (-1,1) and it consequently may not be suitable for value normalization in parameter space where the previous value is zero. Moreover, other chaotic maps in [12-16] require complicated mathematical models with more-than-one parameter spaces, resulting in the complex process of parameter optimization. This paper therefore presents a one-dimensional sinusoidal chaotic map in combination with absolute-value nonlinearity given by

$$x_{n+1} = \alpha \left|\sin\left(\omega x_n + \phi\right)\right| \tag{2}$$

where parameters ω and ϕ are frequency and phase shift, respectively. The absolute-value nonlinearity is suggested in order to limit the output values in the range of zero to one.

Fig. 1 shows the detailed block diagram of the proposed absolute sinusoidal chaotic map described in (1). It is seen from Fig.1 that the output is delayed and fed back through the absolute sinusoidal nonlinearity for each iteration for generating chaotic signals. In particular, the two parameters that significantly set dynamic behaviors are frequency and phase shift. In order to primarily analyze the complex dynamics of the proposed chaotic map, the bifurcation diagram is employed as a tool for a qualitative measure. The bifurcation diagram shows a period doubling that accompanies the onset of chaos, and also represents the sudden appearance of a qualitatively different solution for a nonlinear system as some parameter is varied. On the other hand, the positive Lyapunov exponent (λ) is realized as a tool for a quantitative measure. The Lyapunov exponent characterizes the rate of separation of infinitesimally close trajectories, and can be described as

$$\lambda = \lim_{t\to\infty} \lim_{\Delta x_o \to 0} \frac{1}{t} \ln \frac{|\Delta x(t)|}{|\Delta x_o|} \tag{3}$$

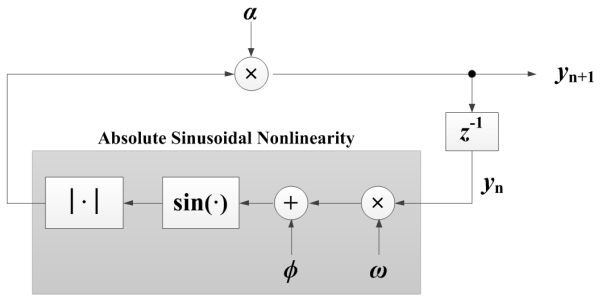


Fig. 1. A detailed block diagram of the proposed chaotic map using absolute sinusoidal nonlinearity described in (1).

where Δx_o is an initial separation of the two trajectories in phase space. Typically, the cases where $\lambda < 0$ and $\lambda = 0$ indicate that the orbit attracts to a stable fixed point or stable periodic orbit and a neutral fixed point, respectively. In the particular case where $\lambda > 0$, the orbit is unstable and the system exhibits chaotic orbits.

In order to investigate chaotic dynamics of the proposed chaotic map, simulations have been performed in MATLAB where initial conditions were set to 0.1 for all cases. Fig.1 shows bifurcating structures of the proposed chaotic map, which are obtained from the positive Lyapunov exponent, i.e. the dark-color region represents chaotic behaviors where $\lambda > 0$ while the white region represents non-chaotic behaviors where $\lambda = 0$ or $\lambda < 0$. Primary investigations on effects of values of parameters α and ω on chaotic dynamics were particularly performed in the region [0, 10] while the phase shift is in the region [0, π]. Noted that the chaotic behaviors for the phase shift in the region [π, 2π] completely resemble the dynamics in those of region [0, π]. In Figs.2 (a) and (b), the values of the gain α were respectively fixed at 1 and 10, and the parameters ω and ϕ were scanned. In Figs.2 (c) and (d), the values of phase shift ϕ were respectively fixed at 0 Radian and π Radian, and the parameters α and ω were scanned. Finally, the value of frequency ω was fixed at 10 rad/s while the parameters α and ϕ were scanned in Figs.2 (e) and (f). It can be concluded from Fig.1 that the proposed chaotic map has a unique pattern of bifurcation structure. The frequency and phase shift are two parameters that significantly set such bifurcation patterns while the gain potentially provide an increase in chaos region. In addition, the proposed chaotic map offers relatively robust chaos over most of the entire parameter spaces, and hence high-complexity operations of the hash function. The selection of parameter values of chaotic map should be in that of dark-color regions in order to guarantee chaotic outputs under most of iteration processes. It should be emphasized in Figs.2 (a) and (b) that the phase shift can be set at zero since chaotic dynamics are still apparent. Upon setting the fixed phase shift at zero Radian, Figs.3 (a) and (b) illustrates the bifurcating diagram and Lyapunov spectrum of the proposed chaotic map for the cases α=1 and α=10, respectively. It can be considered from Fig.3 (a) that the maximum values of the outputs fall within the range [0,1] while the maximum values of Fig.3 (b) increases correspondingly to an amplifying gain of α=10 in each iteration. Nonetheless, there has no significant difference in terms of chaoticity measured by the positive Lyapunov exponent. As a result, the frequency is a major parameter that

TABLE I. SUMMARY OF RELATED CHAOTIC MAPS IMPLEMENTED BASED ON SINUSOIDAL FUNCTIONS.

References	Types of Chaotic Maps	Mathematical Models
[10]	Sine Map	$x_{n+1} = \frac{a}{4}\sin(\pi x_n)$
[11]	Iterative Chaotic Map with Infinite Collapses	$x_{n+1} = \sin(\frac{a}{x_n})$
[12]	Circle Map	$x_{n+1} = x_n + \Omega - \frac{K}{2\pi}\sin(2\pi x_n)$
[13]	Climbing Sine Map	$x_{n+1} = M_a(x_n + a\sin(2\pi x_n))$
[14]	Sine Iterative Map	$x_{n+1} = \sin^2(a\arcsin(\sqrt{x_n}))$
[15]	Moir Grating Map	$x_{n+1} = 0.5 + 0.5\text{sign}(\sin(\frac{2\pi}{\lambda}x_n))$
[16]	Sine Square Map	$x_{n+1} = A\sin^2(x_n - B)$

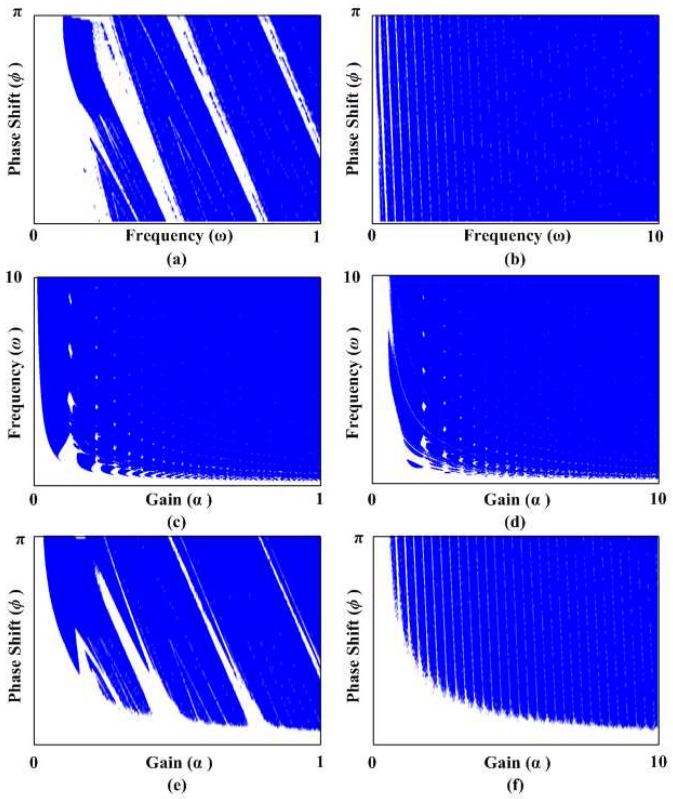


Fig. 2. Bifurcation structures of the proposed chaotic map using absolute sinusoidal nonlinearity.

determines chaotic dynamics and the gain can be arbitrarily set to any values based on required conditions with no changes in complexity.

As for a particular example, the proposed chaotic map is illustrated for it chaotic dynamics with parameter values α=1, ω=10, and ϕ=0, i.e. x_n+1=sin(10x_n). Fig.4 (a) shows an apparently chaotic time-domain waveform. Fig. 4(b) shows the Cobweb plots between x_n and x_{n+1}, indicating that the iteration are mapped over the absolute-value nonlinearity. Figs.4 (c) and (d) show the frequency spectrum using periodogram and autocorrelation plots, respectively. It can be seen that the values are distributed with flat spectrum feature and the values are random reflected by the low autocorrelation of less than approximately 0.01. It can be concluded that the map potentially offers robust and effective randomness for use in hashing algorithm.

III. PROPOSED KEYED HASH FUNCTION

The whole structure of the proposed hashing scheme is depicted in Fig. 5. Assuming that a 128-bit hash value is required, the procedures for generating hash values are described as follows:

(1) The secret keys of the algorithm include the selected initial conditions $y_{0,1}$ and $y_{0,2}$.

(2) The original message M is padded such that its length is a multiple of 2.

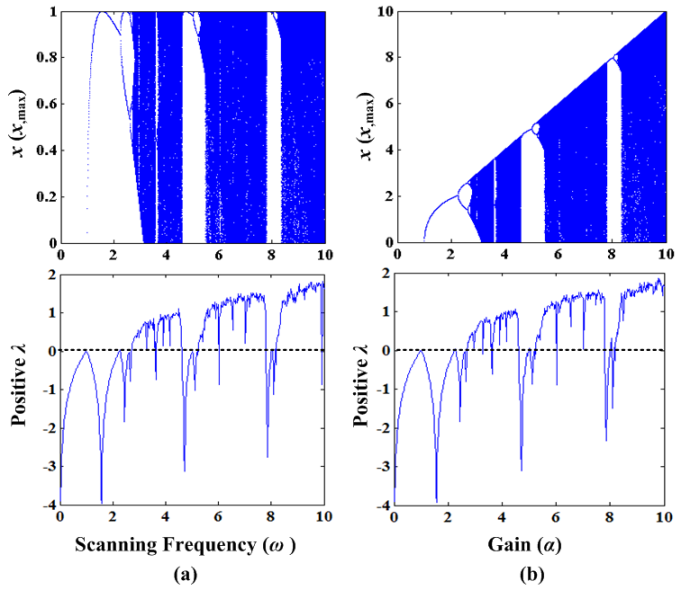


Fig. 3. Bifurcation diagram and Lyapunov spectrum of the proposed chaotic map using absolute sinusoidal nonlinearity.

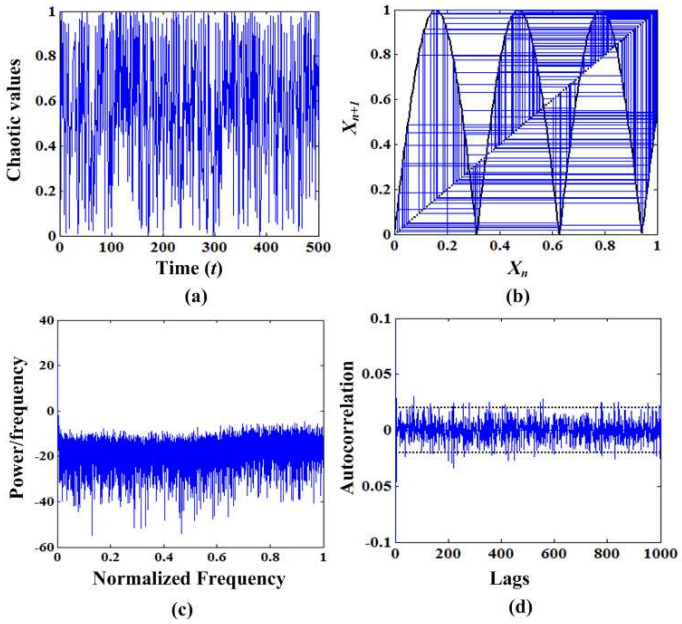


Fig. 4. Chaotic dynamics of the proposed chaotic map where α=1, ω=10, and ϕ=0 ; (a) Time-domain waveforms, (b) Cobweb plots, (c) Frequency spectrum using periodogram and (d) Autocorrelation plots.

(3) The padded message is divided into 2 sub-blocks of length S, $\omega_{i,1}$ and $\omega_{i,2}$, where $i = 1, ..., S$.

(4) For the input stage, S iterations are needed for the absolute sine map with the intention of generating the intermediate output. The first iterations after initial stage are $y_{i,1}(t) = \alpha_1 |\sin(\omega_{i,1} \cdot (y_{i-1,1} + y_{i-1,2}) + \phi_1)|$, and $y_{i,2}(t) = \alpha_2 |\sin(\omega_{i,2} \cdot (y_{i-1,1} + y_{i-1,2}) + \phi_2)|$, for $i = 1, ..., S$.

(5) The last two output values $y_{S,1}(t)$ and $y_{S,2}(t)$ are mapped into decimal integer values d_1 and d_2 with interval [0, 2^{64}]

(6) The decimal integer d_1 and d_2 are converted into 64-bit binary numbers b_1 and b_2.

(7) Finally, b_1 and b_2 are cascaded to form a 128 bit hash value H.

IV. Performance Analysis

A. Uniform Distribution of Hash Value

The uniform distribution of hexadecimal hash value is the crucial property of hashing scheme. In Fig. 6(a) the ASCII characters of the original message are localized in a small

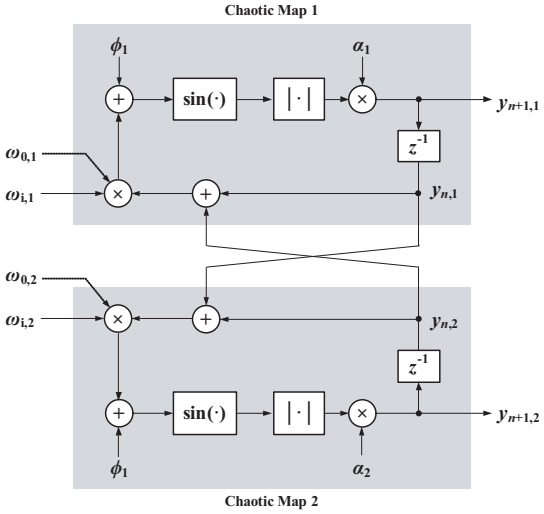


Fig. 5. Block diagram of the cross-coupled topology of the proposed keyed hash function.

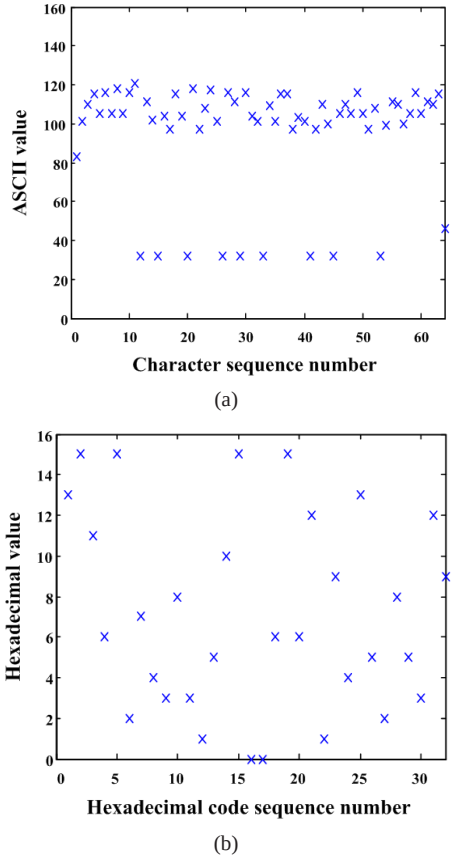


Fig. 6. The distribution of values : (a) original message in ASCII: (b) hash values in hexadecimal format.

interval between 97 and 122. On the contrary, the hash values of the hashing scheme are uniformly spread over the possible range of hash values as illustrated in Fig. 6(b). This signifies that no information of the original message remains after the hashing process.

Fig. 7. An original grayscale Lena image.

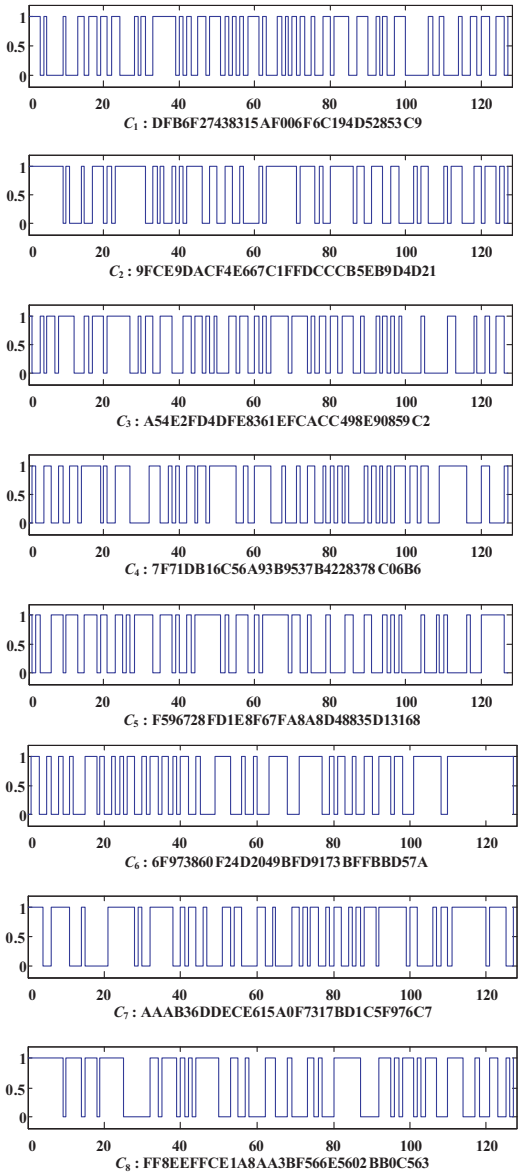


Fig. 8. Corresponding binary sequences of the cases C_1, C_2, C_3, C_4, C_5, C_6, C_7 and C_8.

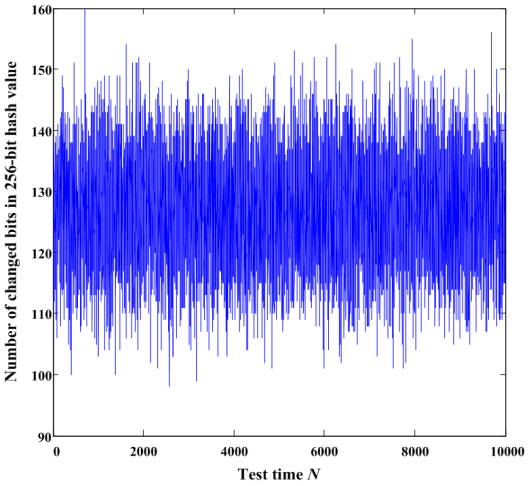


Fig. 9. Distribution of the number of bits changed.

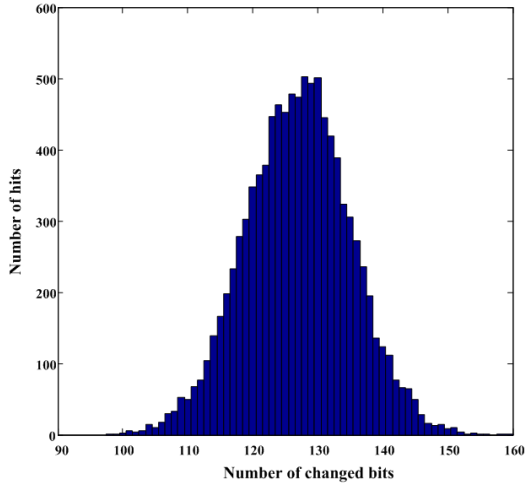


Fig. 10. Histogram of the distribution of number of bit changes for =256 and =10000.

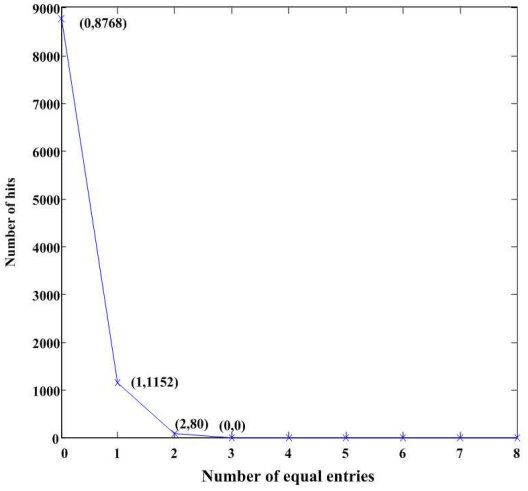


Fig. 11. Distribution of the number of locations where the ASCII characters are identical for =256 and =10000.

B. Sensitivity to Small Changes in Message and Initial Conditions

This subsection depicts the high sensitivity of the proposed hashing function to small changes in original messages and initial conditions. In order to explore this issue, the following tests have particularly been verified.

C_1: The original message is: “Sensitivity of hash value to the message and initial conditions.”.

C_2: Substitute the initial condition $y_{0,1}$ = 2233.4567 with $y_{0,1}$ = 2233.4568. The two 128-bit hash values for C_1 and C_2 differ in 65 positions.

C_3: Substitute the first character of the original message by “A”. The two 128-bit hash values for C_1 and C_3 differ in 63 positions.

C_4: Substitute the character “i” in the word “initial” by “e” to become “enitial”. The two 128-bit hash values for C_1 and C_4 differ in 66 positions.

C_5: Substitute the last character of the original message “.” by “,”. The two 128-bit hash values for C_1 and C_5 differ in 62 positions.

C_6: Lena original image (256 x 256 pixels) shown in Fig. 7.

C_7: Substitute the gray value of the pixel located at the upper left corner by “0”. The two 128-bit hash values for C_6 and C_7 differ in 63 positions.

C_8: Substitute the gray value of the pixel located at the lower right corner by “1”. The two 128-bit hash values for C_6 and C_8 differ in 63 positions.

The corresponding binary sequences of C_1, C_2, C_3, C_4, C_5, C_6, C_7 and C_8 are illustrated in Fig. 8. The result reveals that any small change in the original message and initial condition can result in a 50% chance of changing for each bit of hash value.

C. Confusion and Diffusion

Strong confusion and diffusion properties of the hashing scheme are essential to make it durable to most attacks. The purpose of diffusion is to disperse the hash values randomly over the possible range with the intention of hiding statistical properties of the original message. Confusion employs the transformation to make the relationship between the original message and hash value as complicated as possible.

The n-bit hash value of a random message of size $L = 50n$ is produced and displayed. Diffusion and confusion test procedure proceeds as follows: the n-bit hash value of a random message of size $L = 50n$ is computed. One bit of the random message is randomly selected and switched, and the n-bit hash value of the altered message is computed. Then two hash values are compared and the number of bit changes is quantified. This test is repeated N times for N = 256, 512, 1024, 2048 and 10000 for hash values of size n, where n = 128, 160 and 256. The statistical measures, as illustrated in Table 2, are employed for statistical computations where B_i is the number of bit changes in the i-th test. The results achieved

in tests for n = 128, 160 and 256 and N = 256, 512, 1024, 2048 and 10000 are demonstrated in Tables 3-5.

Distribution of the number of bits changed for various number of test times N is depicted in Fig. 9. Fig. 10 illustrates the histogram distribution of the number of bit changes for n=256 and N=10000. From the results in Tables 3-5, it can be observed that $\overline{B}$ and P are nearly the ideal values of $n/2$ and 50% respectively. All values of ΔB and ΔP are very small, which signifies that diffusion and confusion capability of the proposed hashing scheme is very strong and stable.

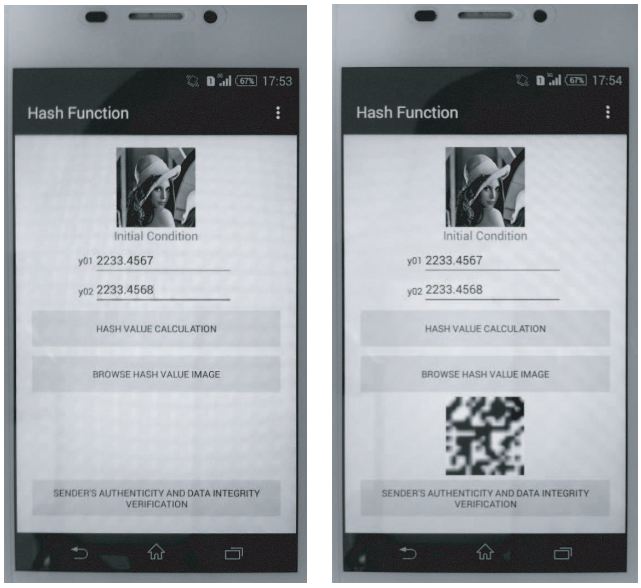
D. Collision Analysis

1) Collision Test: Hash collision occurs when two distinct input messages generate the identical hash values. The following collision test has been done with the purpose of measuring the collision resistance of the proposed hashing scheme. The n-bit hash value of a random message of size $L = 50n$ is produced and displayed in ASCII format. Then one bit in the generated message is randomly selected and flipped. The new hash value is also produced and displayed in ASCII format. The ASCII characters of these two hash values are then compared. The number of hits, which is the number of ASCII characters with the same value at the same position, is quantified. The absolute difference of these two hash values is expressed as

$$d = \sum_{i=1}^{n/8} |dec(m_i) - dec(m_i^{'})| \tag{4}$$

where m_i and $m_i^{'}$ are the i^{th} ASCII character of the original and modified hash values, respectively, and $dec()$ converts m_i and $m_i^{'}$ to the corresponding decimal numbers. This procedure is repeated 10000 times. Table 6 depicts the minimum, maximum and mean values of d. It can be seen that the average absolute difference of each character of the proposed hashing scheme is close to the ideal value of 85.43 [20]. This means that the proposed hashing scheme has the stronger collision resistance than the well-known approaches such as MD5 and SHA-1, and the other chaos-based approaches in [17,18,19,21,22]. The distribution of the number of hits is demonstrated in Fig. 11. It can be seen that the maximum number of equal character is only 2. The results signify that the probability of collision is extremely low.

2) Resistance to Birthday Attack: A birthday attack is a kind of cryptographic attack based on mathematical behind the birthday problem in probability theory. The name is obtained from the surprising result that in a room of 23 people, there is a probability of 50% that at least two people have the same birthday. The hashing scheme should be robust against birthday attack, which makes it difficult to find two distinct messages that have the same hash value. The difficulty of the birthday attack depends on the size of the hash value. For a secure hashing scheme with n-bit hash value, the difficulty of the attack is $2^{n/2}$. Therefore, the value of n is needed to be large enough to make a birthday attack computationally infeasible. For instance, if the size of the hash value is set to 256, the difficulty of the attack would be 2^{128}. This keeps the system robust against this type of attack.



(a) (b)

Fig. 12. Android application user interface : (a) image input for hash value calculation (b) displaying calculated hash value image in PNG format.

E. Speed Analysis and Hash Values Obtained from Different Environments

The running speed of the proposed hashing scheme on a 3.20 GHz Intel(R) Xenon(R) computer with 16 GB of RAM running Windows 7 (64 bit) is approximately 10.27 Megabits per second. The proposed hashing scheme is iterated using double precision floating-point arithmetic. The IEEE-754 floating-point standard was adopted in the early 1980s. If two computers with different platforms (operating systems and hardware) employ the IEEE-754 floating-point standard, two hash values produced by both computers must be identical [23]. In order to affirm this issue, the hash values of the origin message illustrated in section 4 are created in computers with different CPU, operation systems and amounts of memory. The results of the hash function at different environments are demonstrated in Table 7. As can be seen, it reveals that if the IEEE-754 floating-point standard is employed, the hash values obtained from different computers must be the same.

V. Implementation of Proposed Keyed Hash Function on Android Device

The implementations of the proposed keyed hash function on an Android device are presented in order to verify sender's authenticity and the integrity of transmitted data. Fig. 12 demonstrates Android application user interface for hash value calculation. The procedures for hash value calculation are described as follows:

(1) Input the initial conditions ($y_{0,1}$ and $y_{0,2}$).

(2) Choose and import the image file.

(3) Click on "HASH VALUE CALCULATION" button.

(4) The calculated hash value is displayed and saved as PNG file.

TABLE II. STATISTICAL ANALYSIS FORMULAS.

No.	Statistical measures	Formulas
(1)	Minimum number of bit changes	$B_{\min} = \min\left(\{B_i\}_{i=1}^{N}\right)$
(2)	Maximum number of bit changes	$B_{\max} = \max\left(\{B_i\}_{i=1}^{N}\right)$
(3)	Mean number of bit changes	$\overline{B} = \frac{1}{N}\sum_{i=1}^{N} B_i$
(4)	Mean changed probability	$P = \frac{\overline{B}}{n} \times 100\%$
(5)	Standard deviation of the number of bit changes	$\Delta B = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i - \overline{B})^2}$
(6)	Standard deviation	$\Delta P = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{B_i}{n} - P\right)^2} \times 100\%$

TABLE III. THE RESULTS OBTAINED THROUGH STATISTICAL MEASURES FOR A 128-BIT HASH VALUE.

Measures	N = 256	N = 512	N = 1024	N = 2048	N = 10000
$B_{\min}$	46	47	43	41	43
$B_{\max}$	82	80	80	81	83
$\overline{B}$	63.32	63.17	63.24	63.53	63.21
$P(\%)$	49.47	49.35	49.41	49.31	49.63
ΔB	5.59	5.78	5.55	5.64	5.65
$\Delta P(\%)$	4.36	4.51	4.34	4.41	4.41

TABLE IV. THE RESULTS OBTAINED THROUGH STATISTICAL MEASURES FOR A 160-BIT HASH VALUE.

Measures	N = 256	N = 512	N = 1024	N = 2048	N = 10000
$B_{\min}$	63	62	60	60	58
$B_{\max}$	98	100	102	99	101
$\overline{B}$	79.20	79.43	79.17	79.13	79.33
$P(\%)$	49.50	49.64	49.48	49.45	49.58
ΔB	6.11	5.99	6.18	6.08	6.02
$\Delta P(\%)$	3.82	3.74	3.86	3.80	3.76

TABLE V. THE RESULTS OBTAINED THROUGH STATISTICAL MEASURES FOR A 256-BIT HASH VALUE.

Measures	N = 256	N = 512	N = 1024	N = 2048	N = 10000
$B_{\min}$	104	103	106	103	94
$B_{\max}$	153	152	154	150	154
$\overline{B}$	127.87	127.20	127.49	127.20	127.46
$P(\%)$	49.95	49.68	49.80	49.68	49.79
ΔB	7.81	7.87	8.40	7.84	7.92
$\Delta P(\%)$	3.05	3.07	3.28	3.06	3.09

The initial conditions are shared between sender and receiver and the image file is transmitted to the receiver along with the calculated hash value image file. Android application user interface for sender's authenticity and data integrity verifications can be illustrated in Fig.13. The sender's authenticity and data integrity verification processes are described as follows:

(1) Input the initial conditions ($y_{0,1}$ and $y_{0,2}$).

(2) Choose and import received image and hash value image files.

(3) Click on "SENDER'S AUTHENTICITY AND DATA INTEGRITY VERIFICATION" button.

(4) The receiver-calculated hash value is compared with the received hash value.

(5) The results of sender's authenticity and data integrity verifications are displayed on the screen.

These results confirm that the proposed keyed hash function can be used to verify the sender's authenticity and the integrity of transmitted data on the Android device.

VI. CONCLUSION

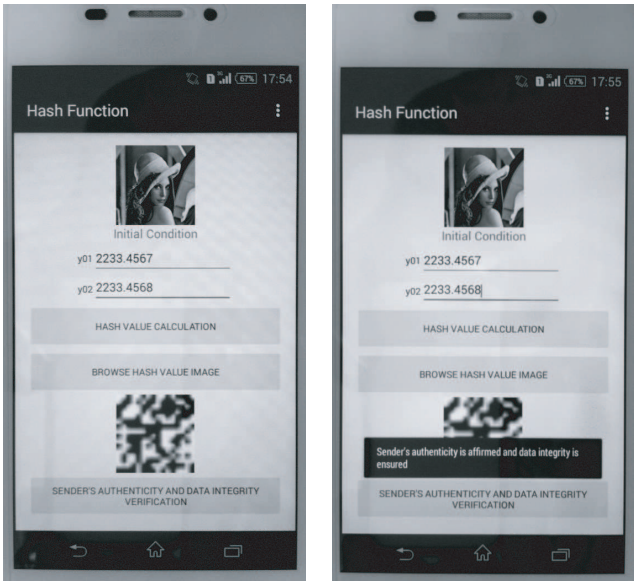
The new compact and robust chaos-based keyed hash function has been presented. The proposed chaotic map exploits absolute-value of sinusoidal nonlinearity for generating

TABLE VI. COMPARISON OF ABSOLUTE DIFFERENCE, WHERE = 10000.

Absolute difference (d)	Min.	Max.	Mean	Mean/Character
MD5(128bit)	590	2074	1304	81.5
SHA-1(160bit)	795	2730	1603	80.15
Kanso's scheme[17](128bit)	737	2320	1494	93.375
Wang's scheme [18] (128bit)	689	2295	1526	95.375
Ren's scheme [19] (128bit)	599	2455	1439	89.9375
Wang's scheme [20] (128bit)	655	2064	1367	85.4375
Xiao's scheme [21] (128bit)	658	2156	1431	89.44
Xiao's scheme [22] (128bit)	605	1952	1227	76.69
The proposed scheme (128bit)	544	2400	1348	84.25
The proposed scheme (160bit)	809	2782	1687	84.35
The proposed scheme (256bit)	1402	3954	2716	84.87

TABLE VII. HASH VALUES OBTAINED FROM DIFFERENT ENVIRONMENTS.

CPU	OS	Memory	Hash values
Intel Core 2 Duo E7400 2.80 GHz	Windows XP	2 GB	DFB6F27438315AF006F6C194D52853C9
Intel Core i3-2100 3.10 GHz	Windows 7	4 GB	DFB6F27438315AF006F6C194D52853C9
Intel Core i5 2.90 GHz	Mac OS	16 GB	DFB6F27438315AF006F6C194D52853C9



(a) (b)

Fig. 13. Android application user interface : (a) image input for verifying sender's authenticity and data integrity (b) displaying results of sender's authenticity data integrity verifications.

highly random iterated values in the diffusion process of ASCII input messages. Chaotic aspects have been investigated through bifurcation structures of Lyapunov exponent as well as Cobweb plots, autocorrelation, and signal characteristics in both time and frequency domains. The proposed hashing structure is relatively simple using only two chaotic maps in the cross-coupled topology that enhances randomness quality for statistical performances. The designed hashing algorithms involve the initial stage when the cross-coupled maps accept initial conditions utilized as secret keys, and the iterative hashing stage that accepts input messages and generates the alterable-length hash values. With such a compact hash function structure, simulation results have revealed several desirable features in terms of statistical performances, involving the mean changed probabilities that are very close to 50%, and the mean number of bit changes that is also close to a half of hash value lengths. In addition, the mean absolute difference of each character values for the hash values of 128, 160 and 256 bits are close to the ideal value of 85.43. This indicates that the proposed hash function has superior performance over well-known algorithms such as MD5 and SHA1, and the other complex structures of chaos-based approaches in [17,18,19,21,22]. A new implementation of hash function Android application has been demonstrated. As a result, the proposed hash function has offered a potential alternative to cryptography and secures protocol methods.

ACKNOWLEDGMENT

The authors are grateful to Research and Academic Services Division, Thai-Nichi Institute of Technology for financial supports. The authors would also like to thank Mr.Sivapong Nilwong for his useful suggestions.

REFERENCES

[1] A. Kanso and M. Ghebleh, "A fast and efficient chaos-based keyed hash function", Communications in nonlinear science and numerical simulation, Vol. 18, pp. 109-123, 2013.

[2] H. Yang, K. Wong, X. Liao, Y.Wang, and D.Yang, "One-way hash function construction based on chaotic map network", Chaos, Solitons and Fractals, Vol. 41, pp. 2566-2574, 2009.

[3] B. O. Brachtl, D. Coppersmith, M.M. Hyden, S.M. Matyas, C.H. Meyer, J. Oseas, S. Pilpel, and 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] W. Hohl, X. Lai, T. Meier, and C.Waldvogel, "Security of iterated hash functions based on block ciphers", In the proceeding of the 3th Annual International Cryptology Conference, Vol.773, pp. 379-390, 1994.

[5] L. R. Knudsen and B. Preneel, “Fast and secure hashing based on codes”, In the proceeding of the 7th Annual International Cryptology Conference, Vol. 1294, pp. 485-498, 1997.

[6] Q. Zhou, K.Wong, X. Liao, T. Xiang, and Y. Hu, “Parallel image encryption algorithm based on discretized chaotic map”, Chaos Solitons & Fractals, Vol 38(4), pp. 1081-1092, 2008.

[7] H. H. Nien, C.K. Huang, S.K. Changchien, H.W. Shieh, C.T. Chen, and Y.Y. Tuan, “Digital color image encoding and decoding using a novel chaotic random generator”, Chaos, Solitons & Fractals, Vol. 32, pp. 1070-1080, 2007.

[8] S. Behnia, A. Akhshani, A. Akhavan, and H. Mahmodi, “Applications of tripled chaotic maps in cryptography”, Chaos Solitons & Fractals, Vol. 40, pp. 505-519, 2009.

[9] K. Wong, “A combined chaotic cryptographic and hashing scheme”, Physics Letter A, Vol. 307, pp. 292-298, 2003.

[10] R. L. Devaney, “An Introduction to Chaotic Dynamical Systems”, Addison-Wesley, 1987.

[11] D. He, C. He, L. Jiang, H. Zhu, G. Hu, “Chaotic characteristic of a one-dimensional iterative map with infinite collapses”, IEEE Transactions on Circuits and Systems, Vol. 48, No.7, pp. 900- 906, 2001.

[12] N. Chatterjee and N. Gupte, “Synchronicity in coupled sine circle maps; some numerical results”, Physica A: Statistical Mechanics and its Applications, Vol.224, No.1-2, pp. 422-432, 1996.

[13] N. Korabel, and R. Klages, “Fractality of deterministic diffusion in the nonhyperbolic climbing sine map”, Physica D: Nonlinear Phenomena, Vol.187, No. 1-4, pp. 66-88, 2004.

[14] F. Y. Han and C.X. Zhu, “One kid based on double unidimenstional chaos system picture encryption algorithm”, Journal of Computer Engineering and Applications, Vol.43, No.20, pp. 50-51, 2007.

[15] V. Petrauskiene, R. Palivonaite, A. Aleksa, and M. Ragulskis, “Dynamic visual cryptography based on chaotic oscillations”, Communications in nonlinear science and numerical simulation, Vol. 19, pp. 112-120, 2014.

[16] Q. Wu, G. Wang, and L.Yuan, “E-mail Encryption Based on Dual Chaotic Map”, In the proceeding of International Workshop on Chaos-Fractals Theories and Applications, 2010 Date of Conference: 29-31 Oct. 2010 Page(s): 159 - 163.

[17] A. Kanso, H. Yahyaoui, and M. Almulla, “Keyed hash function based on a chaotic map”, Information Sciences, Vol. 186, pp. 249-264, 2012.

[18] Y. Wang, X. Liao, D. Xiao, and K. Wong. “One-way hash function construction based on 2D coupled map lattices”, Information Sciences, Vol. 178, pp.1391-1406, 2008.

[19] H. Ren, Y. Wang, Q. Xie, and H. Yang, “A novel method for one-way hash function construction based on spatiotemporal chaos”, Chaos, Solitons & Fractals, Vol. 42, pp. 2014-2022, 2009.

[20] Y. Wang, K. Wong, D. Xiao, “Parallel hash function construction based on coupled map lattices”, Communications in Nonlinear Science and Numerical Simulation, Vol. 16, No. 7, pp. 2810-2821, 2011.

[21] D. Xiao, X. Liao, S. Deng, “Parallel keyed hash function construction based on chaotic maps”, Phys Lett A, Vol. 372, 4682-4688, 2008.

[22] D. Xiao, X. Liao, Y. Wang, “Parallel keyed hash function construction based on chaotic neural network”, Neural computing, Vol. 72, pp. 2288-2296, 2009.

[23] X. Yi, “Hash function based on chaotic tent maps”, IEEE transactions on circuits and systems-II : Express briefs, Vol. 52, pp. 354-357, 2005.

Wimol San-Um was born in Nan Province, Thailand in 1981. He received B.Eng. Degree in Electrical Engineering and M.Sc. Degree in Telecommunications in 2003 and 2006, respectively, from Sirindhorn International Institute of Technology (SIIT), Thammasat University in Thailand. In 2007, he was a research student at University of Applied Science Ravensburg-Weingarten in Germany. He received Ph.D. in mixed-signal very large-scaled integrated circuit designs in 2010 from the Department of Electronic and Photonic System Engineering, Kochi University of Technology (KUT) in Japan. He is currently with Master of Engineering Technology program, Faculty of Engineering, Thai-Nichi Institute of Technology (TNI). He is also the head of Intelligent Electronic Systems (IES) Research Laboratory. His areas of research interests are chaos theory, artificial neural networks, control automations, digital image processing, secure communications, and nonlinear dynamics of chaotic circuits and systems.

Warakorn Srichavengsup obtained the B.Eng., M.Eng. and Ph.D. degree in Electrical Engineering from Chulalongkorn University, Bangkok, Thailand, in 1998, 2003 and 2009, respectively. He is currently a lecturer with the Department of Computer Engineering at Faculty of Engineering, Thai-Nichi Institute of Technology (TNI), Bangkok, Thailand. Prior to joining TNI, he was a visiting research student during 2008 with the Laboratory for Information and Decision Systems (LIDS) at the Massachusetts Institute of Technology (MIT). His main research interests are MAC protocol for high speed wireless local area networks, computer cryptography and information security.