A CHAOS-BASED KEYED HASH FUNCTION FOR MOBILE AD HOC WIRELESS APPLICATION

Winai Chankasame

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Engineering Technology
Graduate School
Thai-Nichi Institute of Technology
Academic Year 2015

Thesis Title	A Chaos-Based Key Hash Function For Mobile Ad Hoc Wireless Application
By	Winai Chankasame
Field of Study	Engineering Technology
Advisor	Asst. Prof. Dr. Wimol San-Um

The Graduate School of Thai-Nichi Institute of Technology has been approved and accepted as partial fulfillment of the requirements for the Master's Degree

…….…….. Dean of Graduate School
(Assoc. Prof. Dr. Pichit Sukchareonpong)
Month…..…...… Date…..…., Year……..

Thesis Committees

…….…….. Chairperson
(Dr. Surapong Pongyupinpanich)

…….…….. Committee
(Asst. Prof. Dr. Warakorn Srichavengsup)

…….…….. Committee
(Dr. Kanticha Kittipeerachon)

…….…….. Advisor
(Asst. Prof. Dr. Wimol San-um)

WINAI CHANKASAME: A CHAOS-BASED KEY HASH FUNCTION FOR MOBILE AD HOC WIRELESS APPLICATION. ADVISOR: ASST. PROF. DR. WIMOL SAN-UM, 60 PP.

This thesis presents the design of communication in the Mobile Ad hoc Networks (MANET) is challenging due to limited wireless transmission ranges of node mobility, limited power resources, and limited physical security. The advantages of MANET include simple and fast deployment, robustness, adaptive and self-organizing networks. Nonetheless, routing protocols are important operations for communication among wireless devices. Assuring secure routing protocols is challenging since MANET wireless networks are highly vulnerable to security attacks. Most traditional routing protocols and message authentication designs do not address security, and are mainly based on a mutual trust relationship among nodes. This paper therefore proposes a new chaos-based keyed hash function that can be used for communication protocols in MANET. The proposed chaotic map realizes an absolute-value nonlinearity, which offers robust chaos over wide parameter spaces, i.e. high degree of randomness through chaoticity measurements using Lyapunov exponent. The proposed keyed hash function structure is compact through the use of a single stage chaos-based topology. Hash function operations involve an initial stage when the chaotic map accepts input message and initial conditions, and a hashing stage where alterable-length hash values are generated iteratively. Hashing performances are evaluated in terms of original message condition changes, statistical analyses, and collision analyses. Results of hashing performances show that the mean changed probabilities are very close to 50%, and the mean changed bit number is also close to a half of hash value lengths. The proposed keyed hash function enhances the collision resistance, comparing to typical MD5 and SHA1, and is faster than other complicated chaos-based approaches.

Graduate School Student's Signature……..……..……

Field of Engineering Technology Advisor's signature……..…..……

Academic Year 2015

Acknowledgements

The author wishes to express his profound gratitude and respectfully dedicate this work to his parent and family members for their endless encouragements, love and sacrifices. The author is most grateful to his advisor, Asst. Prof. Dr. Wimol San-um, for his valuable supervision, support and encouragements throughout the study. In addition, grateful acknowledges are made to Asst. Dr. Surapong Srichavengsup, Asst. Prof. Dr. Warakorn Srichavengsup and Dr. Kanticha Kittipeerachon members of thesis committee, for their valuable suggestions and comments. The authors are grateful to Research and Development Division of Thai-Nichi Institute of Technology for research fund supports.

Winai Chankasame

Table of Contents

Table of Contents (Continued)

List of Table

List of Figures

Chapter 1
Introduction

1.1 Introduction

This chapter introduces the background of MANET systems. Motivation and statement of purposes are involved. This chapter also includes the objectives as well as the expected outcomes of the research. Some basic definitions of technical terms are also defined.

1.2 Background

A mobile Ad Hoc Network (MANET) has been emerged for a variety of applications such as in emergency search and rescue operations, in battlefields, or in setting up of instant communication among moving vehicles. Such a Mobile Ad Hoc Network normally is a type of multi-hop network formed temporally by a group of non infrastructure wireless mobile nodes that relays data packets from one to another in order to communicate with the remote nodes beyond the radio distance. In contrast to conventional architectures, the public media, the change of network topology, cooperation algorithms, the lack of centralized monitoring, and system management are vulnerable to mischievous attacks. Typically, three types of Ad Hoc network protocols are Proactive protocols (DSDV, WRP, CGSR, STAR, and OLSR), Reactive protocols (ABR, DSR, TORA, AODV, CBRP, RDMAR) and Hybrid protocol (ZRP). Nonetheless, two kinds of attacks in Ad Hoc Network are passive and active attacks. In passive attacks, the routing protocol is does not disturbed, but its only eavesdrops o routing traffic and extract the valuable information. However, malicious nodes in active network may disrupt and change the functions of a routing protocols I three possible approaches, i.e. routing information modification, false routing information fabrication, and node impersonation. Consequently, security issue has essentially becoming more challenging for secure protocols in Ad Hoc networks. Recently, a variety of secure protocols have been suggested, for instance, a *q*-composite model that increases the capability of anti-capture of protocols, a basic random key pre-distribution model by generating key pool and allotting its different parts to each node.

Other research approaches also studied key distributions through either random keys or hash table for authentications.

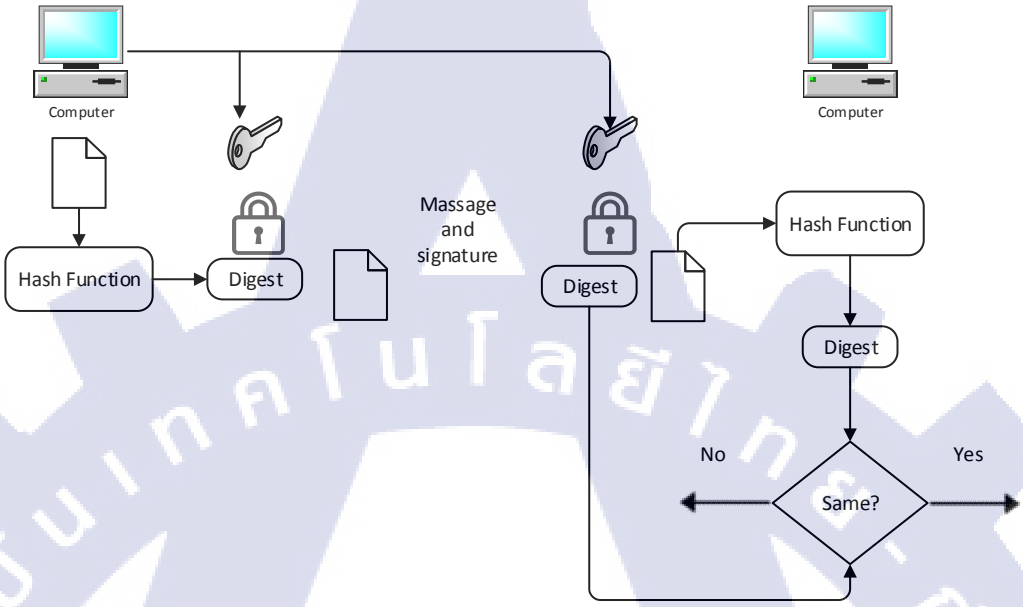


Figure 1.1 Block diagram of hash function in communication systems.

This paper alternatively focuses on an authentication of Ad Hoc wireless protocols and message through the validation using a hash function. Such a hash function has been a key technology in advanced cryptography, which encodes an arbitrary length input message into a hash value with fixed length. Typically, 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 general, preferable characteristics of hash functions include high possibility of collision resistance and high security against preimage and second-preimage attacks.Typical hash functions, for instance, Message Digest (MAD, MAD) and Secure Has Algorithm (SHA-1) have generally been exploited in software industries for purposes of integrity verification of electronically transmitted files as well as security in protocols. However, the typical hash functions have been designed based on logical operations or multi-round iterations, and therefore the hashing process efficiency, which depends upon inherent ciphers and complicated computation

processes, are necessarily required. Recently, it has been notified through the analysis of the collision frequencies that those of typical hash functions contain several undiscovered flaws. Therefore, multiple-block-length hash functions have been suggested. Nonetheless, designs and implementation of such multiple-block-length hash function are relatively complicated in terms of security and efficiency. As a pervasive feature in nature, chaos is a deterministic process generated by nonlinear dynamical systems that possess distinctive properties in pseudo-randomness, sensitivity to initial conditions and control parameters. Due to these properties, chaos-based hash algorithms have been of much interest as an alternative to that of typical hash functions. Several chaos-based hash function algorithms have therefore been proposed recently. 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.

1.3 Motivations

The motivation of this thesis is to find a new algorithm of hash function for security purposes in MANET system. Chaotic maps shall be used in this thesis as it is a new technique that receives much research attention.

1.4 Statement of Problem and Hypothesis

The problem found in previous research work is analysis of the collision frequencies that those of typical hash functions contain several undiscovered flaws. Although multiple-block-length hash functions have been suggested, designs and implementation of such multiple-block-length hash function are relatively complicated in terms of security and efficiency. Therefore, the hypothesis of this thesis is to design a new has function that can operate fast and offer high security levels.

1.5 Objectives

1.5.1 To design a keyed hash function that can be used for communication protocols in a mobile Ad Hoc Network

1.5.2 To apply a cheos-based keyed hash function for secure protocol and massege authentication in a mobile Ad Hoc Network

1.6 Research Scopes

1.6.1 Study a keyed hash function that be used for communication protocols in a mobile Ad Hoc Network

1.6.2 Apply to design a chaos-based keyed hash function for secure protocol and massege authentication in a mobile Ad Hoc Network

1.6.3 Measure the properties of a chaos-based keyed hash function using statistical measurements and various kinds of attacks.

1.7 Expected Outcomes

1.7.1 To achieve a keyed hash function that can be used for communication protocols in a mobile Ad Hoc Network

1.7.2 To achieve a new chos-based keyed hash function for secure protocol and massege authentication in a mobile Ad Hoc Network

1.8 Definitions

18.1 Polynomial is an expression consisting of variables and coefficients that involves only the operations of addition, subtraction, multiplication, and non-negative integer exponents. An example of a polynomial of a single indeterminate. Polynomials appear in a wide variety of areas of mathematics and science. For example, they are used to form polynomial equations, which encode a wide range of problems, from elementary word problems to complicated problems in the sciences; they are used to define polynomial functions, which appear in settings ranging from basic chemistry and physics to economics and social science; they are used in calculus and numerical analysis to approximate other functions. In advanced mathematics,

polynomials are used to construct polynomial rings and algebraic varieties, central concepts in algebra and algebraic geometry.

1.8.2 MATLAB is advanced computer program (High-level Language) for technical computing that includes numerical computation. Complex graphics And replication to visualize the image is simple and clear name MATLAB stands for Matrix Laboratory original MATLAB program is written to use in the calculation of matrix or a matrix software which MATLAB is a program developed unceasingly. The program is easy to understand. And complex programming When put to use, and can see the results quickly. For this reason it makes MATLAB program has been used extensively in various fields.

1.9 Conclusions

This chapter has introduced the background of MANET systems. Motivation and statement of purposes were stated. This chapter also summarized the objectives as well as the expected outcomes of the research. Some basic definitions of technical terms are also defined.

Chapter 2
Related Theories and Literature Reviews

2.1 Introduction

This chapter gives information for related theory including key concepts of chaos theory, and bifurcation. Two examples of chaotic amp including logistic and henon maps are described. The literature reviews for related works on chaos-based keyed hash function is also included.

2.2 Related Theory

2.2.1 Key Concepts of Chaos Theory

Chaos Theory is an alternative name for “nonlinear dynamical systems theory. The latter is an umbrella term for the study of phenomena such as attractors, bifurcations, chaos, fractals, catastrophes, and self-organization, all of which describe systems as they change over time. In this paper will examine the basic patterns of movement, and their applications to a wide range of psychological theories. Chaos itself is a particular nonlinear dynamic and is perhaps the centerpiece of this field of study. In chaotic phenomena, seemingly random events are actually predictable from simple deterministic equations. Thus a phenomenon that appears locally unpredictable may indeed be globally stable, exhibit clear boundaries, and display sensitivity to initial conditions. The latter property is also known as The Butterfly Effect. Chaos has a close relationship to other dynamics, however, such as attractors, bifurcations, fractals, and self-organization.

2.2.2 Bifurcation

Bifurcation in a term of dynamical system, a bifurcation is a period doubling, quadrupling, etc. that accompanies the onset of chaos. It represents the sudden appearance of a qualitatively different solution for a nonlinear system as some parameter is varied. The illustration above shows bifurcations of the logistic map as the parameter r is varied. Bifurcations come in four basic varieties: flip bifurcation, fold bifurcation, pitchfork bifurcation, and transcritical bifurcation. Bifurcations also

describe changes in the stability or existence of fixed points as a control parameter in the system changes. As a very simple explanation of a bifurcation in a dynamical system, consider an object balanced on top of a

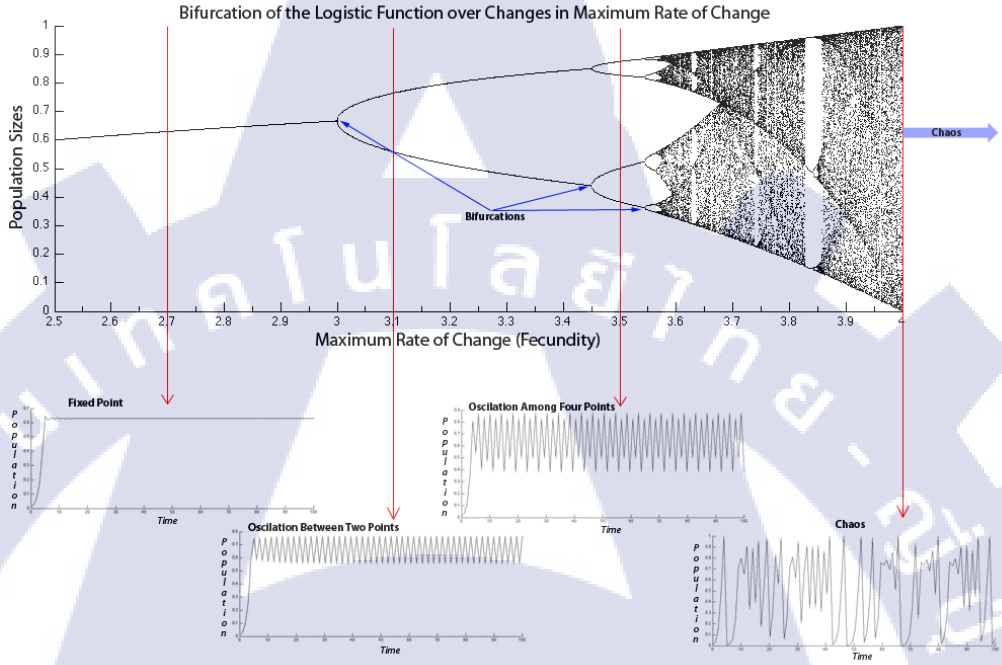


Figure 2.1 Demonstration of the bifurcation diagram and its time domain.

vertical graph. The mass of the object can be thought of as the control parameter. As the mass of the object increases, the graph's deflection from vertical, which is x, the dynamic variable, remains relatively stable. But when the mass reaches a certain point, the bifurcation point. The graph will suddenly buckle. Changes in the control parameter eventually changed the qualitative behavior of the system.

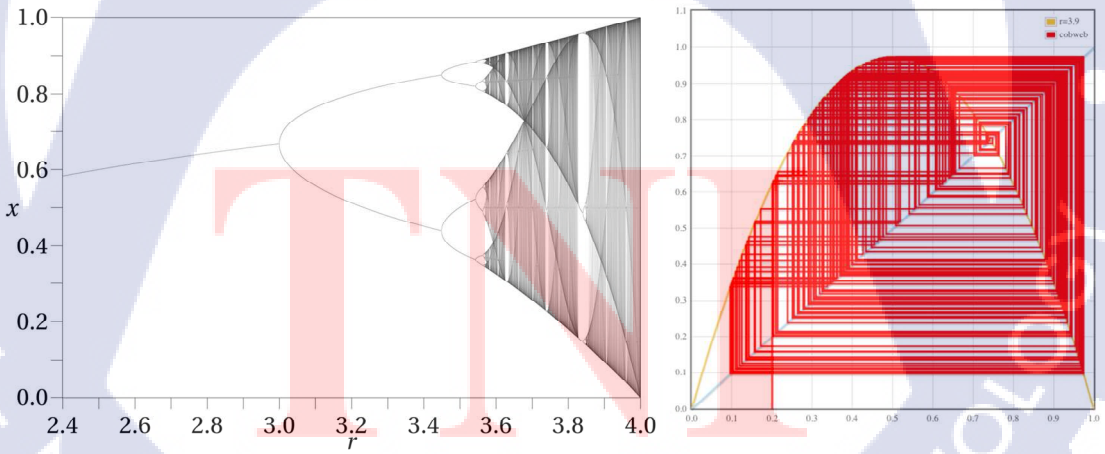
2.2.3 Logistic Map

The logistic map is a polynomial mapping of degree 2, often cited as an archetypal example of how complex, chaotic behavior can arise from very simple non-linear dynamical equations. The map was popularized in a seminal 1976 paper by the biologist Robert May, in part as a discrete-time demographic model analogous to

the logistic equation first created by Pierre François Verhulst. This nonlinear difference equation is intended to capture two effects; First, the reproduction where the population will increase at a rate proportional to the current population when the population size is small. Second, starvation where the growth rate will decrease at a rate proportional to the value obtained by taking the theoretical "carrying capacity" of the environment less the current population. However, as a demographic model the logistic map has the pathological problem that some initial conditions and parameter values lead to negative population sizes. Typically, the logistic map is expressed as

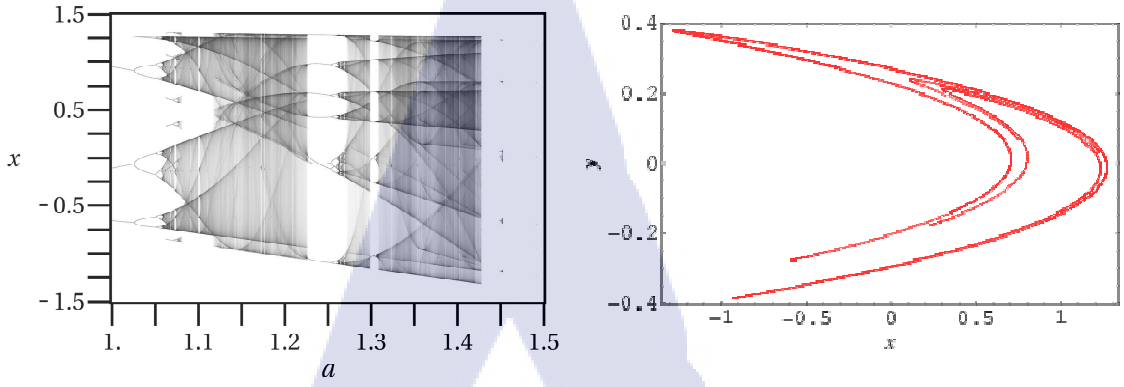
$$x_{n+1} = rx_n(1 - x_n). \tag{2.1}$$

The bifurcation parameter r is shown on the horizontal axis of the plot and the vertical axis shows the possible long-term population values of the logistic function. The bifurcation diagram nicely shows the forking of the possible periods of stable orbits from 1 to 2 to 4 to 8 etc. Each of these bifurcation points is a period-doubling bifurcation. The ratio of the lengths of successive intervals between values of r for which bifurcation occurs converges to the first Feigenbaum constant.



(a) (b)

Figure 2.2 (a) the bifurcation of logistic map, (b) cobweb plot of logistic maps



(a) (b)

Figure 2.3 (a) the bifurcation of Henson map, (b) Attractor of Henon maps

2.2.4 Henon Map

The Henon map is a discrete-time dynamical system. It is one of the most studied examples of dynamical systems that exhibit chaotic behavior. The Henon map takes a point (x_n, y_n) in the plane and maps it to a new point

$$x_{n+1} = 1 - ax_n^2 + y_n$$
$$y_{n+1} = bx_n . \quad (2.2)$$

The map depends on two parameters, *a* and *b*, which for the classical Henon map have values of a = 1.4 and b = 0.3. For the classical values the Henon map is chaotic. For other values of *a* and *b* the map may be chaotic, intermittent, or converge to a periodic orbit. An overview of the type of behavior of the map at different parameter values may be obtained from its orbit diagram.

2.2.5 Cryptographic Hash Function

A cryptographic hash function is a hash function which is considered practically impossible to invert, i.e. to recreate the input data from its hash value alone. These one-way hash functions have been called "the workhorses of modern cryptography". The input data is often called the message, and the hash value is often called the message digest or simply the digest. The ideal cryptographic hash function

has four main properties: First, it is easy to compute the hash value for any given message. Second, it is infeasible to generate a message that has a given hash. Third, it is infeasible to modify a message without changing the hash. Last, it is infeasible to find two different messages with the same hash.

Cryptographic hash functions have many information security applications, notably in digital signatures, message authentication codes (MACs), and other forms of authentication. They can also be used as ordinary hash functions, to index data in hash tables, for fingerprinting, to detect duplicate data or uniquely identify files, and as checksums to detect accidental data corruption. Indeed, in information security contexts, cryptographic hash values are sometimes called (digital) fingerprints, checksums, or just hash values, even though all these terms stand for more general functions with rather different properties and purposes.

Most cryptographic hash functions are designed to take a string of any length as input and produce a fixed-length hash value. A cryptographic hash function must be able to withstand all known types of cryptanalytic attack. At a minimum, it must have the following properties: (1) Pre-image resistance which is given a hash h it should be difficult to find any message m such that h = hash (m). This concept is related to that of one-way function. Functions that lack this property are vulnerable to preimage attacks, (2) Second pre-image resistance which is given an input m1 it should be difficult to find another input m2 such that m1 ≠ m2 and hash (m1) = hash (m2). Functions that lack this property are vulnerable to second-preimage attacks, (3) Collision resistance that should be difficult to find two different messages m1 and m2 such that hash (m1) = hash (m2). Such a pair is called a cryptographic hash collision. This property is sometimes referred to as strong collision resistance. It requires a hash value at least twice as long as that required for preimage-resistance; otherwise collisions may be found by a birthday attack.

These properties imply that a malicious adversary cannot replace or modify the input data without changing its digest. Thus, if two strings have the same digest, one can be very confident that they are identical. A function meeting these criteria may still have undesirable properties. Currently popular cryptographic hash functions are vulnerable to length-extension attacks: given hash (m) and len(m) but not m, by

choosing a suitable m' an attacker can calculate hash(m || m') where || denotes concatenation. This property can be used to break naive authentication schemes based on hash functions. The HMAC construction works around these problems.

Ideally, one may wish for even stronger conditions. It should be impossible for an adversary to find two messages with substantially similar digests; or to infer any useful information about the data, given only its digest. Therefore, a cryptographic hash function should behave as much as possible like a random function while still being deterministic and efficiently computable. Checksum algorithms, such as CRC32 and other cyclic redundancy checks, are designed to meet much weaker requirements, and are generally unsuitable as cryptographic hash functions. For example, a CRC was used for message integrity in the WEP encryption standard, but an attack was readily discovered which exploited the linearity of the checksum.

2.2.6Authentication

Authentication is the act of confirming the truth of an attribute of a single piece of data or entity. In contrast with identification which refers to the act of stating or otherwise indicating a claim purportedly attesting to a person or thing's identity, authentication is the process of actually confirming that identity. It might involve confirming the identity of a person by validating their identity documents, verifying the validity of a Website with a digital certificate, tracing the age of an artifact by carbon dating, or ensuring that a product is what its packaging and labeling claim to be. In other words, authentication often involves verifying the validity of at least one form of identification.

2.3 Literature reviews on Existing Chaos-Based Hash Function

2.3.1 Related Publications

Table 2.1 Summary of related publications with proposed scheme

Author	Year	Proposed Schemes
L. Feng et al. [1]	2009	Security bootstrap model of key pre-sharing by polynomial group in mobile Ad Hoc Network
S. Behnia et al. [2]	2009	Applications of tripled chaotic maps in cryptography
Huaqian Yang et al. [3]	2009	One-way hash function construction based on chaotic map network
R. Sparr and R Wernsdorf [4]	2008	Group theoretic properties of Rijndael-like ciphers
Q. Zhou et al. [5]	2008	Parallel image encryption algorithm based on discretized chaotic map
H. H. Nien et al. [6]	2007	Digital color image encoding and decoding using a novel chaotic random generator
K. W. Wong [7]	2003	A combined chaotic cryptographic and hashing scheme

Table 2.1 summarizes related publications with proposed scheme. As shown in Table 2.1, L. Feng et al. [1] presents the setting up a security module into a Mobile Ad Hoc Network (MANET) before its deployment is very crucial to its future security assurance. It can establish a solid network security shield gradually on groups of separated nodes lacking any guarantee by sharing common knowledge and communication protocols in a MANET. This paper proposed a security bootstrap model of key pre-sharing by a polynomial group based on (t, n) threshold schema of Lagrange Polynomial Group (LPG) and one-way hash function to establish a distributive security infrastructure. This model includes the following two processes: the first phase is to setup a key pre-sharing mechanism based on one-way hash

function and Lagrange Interpolation Polynomial Group (LIPG); and the second phase mainly deals with recovering key in a more secure way based on the digital signature of threshold schema. The one-way hash approach can prevent the spitted key pieces in a key pool from being exposed efficiently. In addition, the threshold digital signature can detect and block the DoS attack and other malicious fraudulent actions efficiently during the processes of key reconstruction and recovery. They performed the experiment under the OPNET simulation environment to validate the proposed approach from the points of view of computation complexity, the security of boot strap process, the performance of establishing secure links, the capability of resisting the capture and the network scale, etc. The experimental results show that this approach can ensure the security of MANET environment with better performance.

S. Behnia et al. [2] proposes security of information that has become a major issue during the last decades. New algorithms based on chaotic maps were suggested for protection of different types of multimedia data, especially digital images and videos in this period. However, many of them fundamentally were flawed by a lack of robustness and security. For getting higher security and higher complexity, in the current paper, they introduce a new kind of symmetric key block cipher algorithm that is based on tripled chaotic maps. In this algorithm, the utilization of two coupling parameters, as well as the increased complexity of the cryptosystem, makes a contribution to the development of cryptosystem with higher security. In order to increase the security of the proposed algorithm, the size of key space and the computational complexity of the coupling parameters should be increased as well. Both the theoretical and experimental results state that the proposed algorithm has many capabilities such as acceptable speed and complexity in the algorithm due to the existence of two coupling parameters and high security. Note that the cipher text has a flat distribution and has the same size as the plaintext. Therefore, it is suitable for practical use in secure communications.

H. Yang et al. [3] introduces a novel chaotic hash algorithm based on a network structure formed by 16 chaotic maps. The original message is first padded with zeros to make the length a multiple of four. Then it is divided into a number of blocks each contains 4 bytes. In the hashing process, the blocks are mixed together by the chaotic map network since the initial value and the control parameter of each tent

map are dynamically determined by the output of its neighbors. To enhance the confusion and diffusion effect, the cipher block chaining (CBC) mode is adopted in the algorithm. Theoretic analyses and numerical simulations both show that the proposed hash algorithm possesses good statistical properties, strong collision resistance and high flexibility, as required by practical keyed hash functions.

R. Sparr and R. Wernsdorf [4] provide conditions for which the round functions of an l-bit Rijndael-like block cipher generate the alternating group on the set $\{0,1\}^l$ .These conditions show that the class of Rijndael-like ciphers whose round functions generate the alternating group on their message space is large, and includes both the actual Rijndael and the block cipher used by the compression function of the WHIRLPOOL hash function. The result indicates that there is no trapdoor design for a Rijndael-like cipher based on the imprimitivity of the group action of its proper round functions which is difficult to detect.

Q. Zhou et al. [5] proposes a variety of chaos-based algorithms were proposed for image encryption. Nevertheless, none of them works efficiently in parallel computing environment. In this paper, we propose a framework for parallel image encryption. Based on this framework, a new algorithm is designed using the discretized Kolmogorov flow map. It fulfills all the requirements for a parallel image encryption algorithm. Moreover, it is secure and fast. These properties make it a good choice for image encryption on parallel computing platforms.

H. H. Nien et al. [6] proposes a novel chaotic system, in which variables are treated as encryption keys in order to achieve secure transmission of digital color images. Since the dynamic response of chaotic system is highly sensitive to the initial values of a system and to the variation of a parameter, and chaotic trajectory is so unpredictable, we use elements of variables as encryption keys and apply these to computer internet communication of digital color images. As a result, we obtain much higher communication security. We adopt one statistic method involving correlation coefficient c and FIPS PUB 140-1 to test on the distribution of distinguished elements of variables for continuous-time chaotic system, and accordingly select optimal encryption keys to use in secure communication of digital color images. At the transmitter end, we conduct RGB level decomposition on digital color images, and encrypt them with chaotic keys, and finally transmit them through computer internet.

The same encryption keys are used to decrypt and recover the original images at the receiver end. Even if the encrypted images are stolen in the public channel, an intruder is not able to decrypt and recover the original images because of the lack of adequate encryption keys. Empirical example shows that the chaotic system and encryption keys applied in the encryption, transmission, decryption, and recovery of digital color images can achieve higher communication security and best recovered images.

K. W. Wong [7] has proposed a fast chaotic cryptographic scheme based on iterating a logistic map with the look-up table updated dynamically. We found that after the whole encryption process, the final look-up table strongly depends on the message and so it can be considered as its hash value or message authentication code. In this Letter, we generalize the chaotic cryptographic scheme so that it can perform both encryption and hashing to produce the cipher text as well as the hash value for a given message. The collision resistance of the proposed hashing approach is also analyzed.

2.4 Conclusions

This chapter has provided information for related theory including key concepts of chaos theory, and bifurcation. Two examples of chaotic amp including logistic and henon maps were demonstrated. The literature reviews of seven papers on chaos-based keyed hash function were also included.

Chapter 3
Research Methodology

3.1 Introduction

This chapter presents research methodology, including research process, data collection, and research tool that uses in this thesis.

3.2 Research Processes

3.2.1 Study the operation of the keyed hash function in terms of its topology, distribution of has values, sensitivity of hash values, statistical analysis of confusion and diffusion, collision test, resistance to birthday attack.

3.2.2 Study the operation and properties of chaotic map, including Logistics maps, bifurcation, tree-dimensional LE, chaotic waveforms, cobweb plot, histogram,

3.2.3 Design chaotic map with power absolute value nonlinearity.

3.2.4 Run the statistical tests using NIST standard software.

3.2.5 Design the hash algorithms with minimum number of chaotic map, but high potential in all kinds of attacks.

3.2.6 Perform the analysis of hash function performances.

3.3 Data Collection

The main data is time domain waveforms of the chaotic maps. The, the data will perform the thresholding to get the random bit data. The random bit data will be used for NIST tests prior to statistical tests, including minimum changed bit number, maximum changed bit number, mean changed bit number, mean changed probability, standard deviation of change bit number, and standard deviation.

3.4 Research Tools

In this thesis, research tool is MATLAB version 2013a

3.5 Conclusion

This chapter has presented research methodology, including research process, data collection, and research tool that uses in this thesis.

Chapter 4
Experimental Results

4.1 Introduction

This chapter presents the experimental and experimental results of

4.2 Experimental

4.2.1 Experimental Apparatus Proposed Chaotic Map using Power Absolute-Value Nonlinearity

This paper proposed a simple chaotic map given by

$$x_{n+1} = \left|\alpha x_n - 1\right|^q \quad (1)$$

As for purposes of chaos measurements, preliminary investigations of the three chaotic maps are performed by bifurcation diagrams and Lyapunov exponent (LE) as for qualitative and quantitative measures of chaos, respectively. The bifurcation diagram indicates possible long-term values, involving fixed points or periodic orbits, of a system as a function of a bifurcation parameter. The stable solution is represented by a straight line while the unstable solutions are

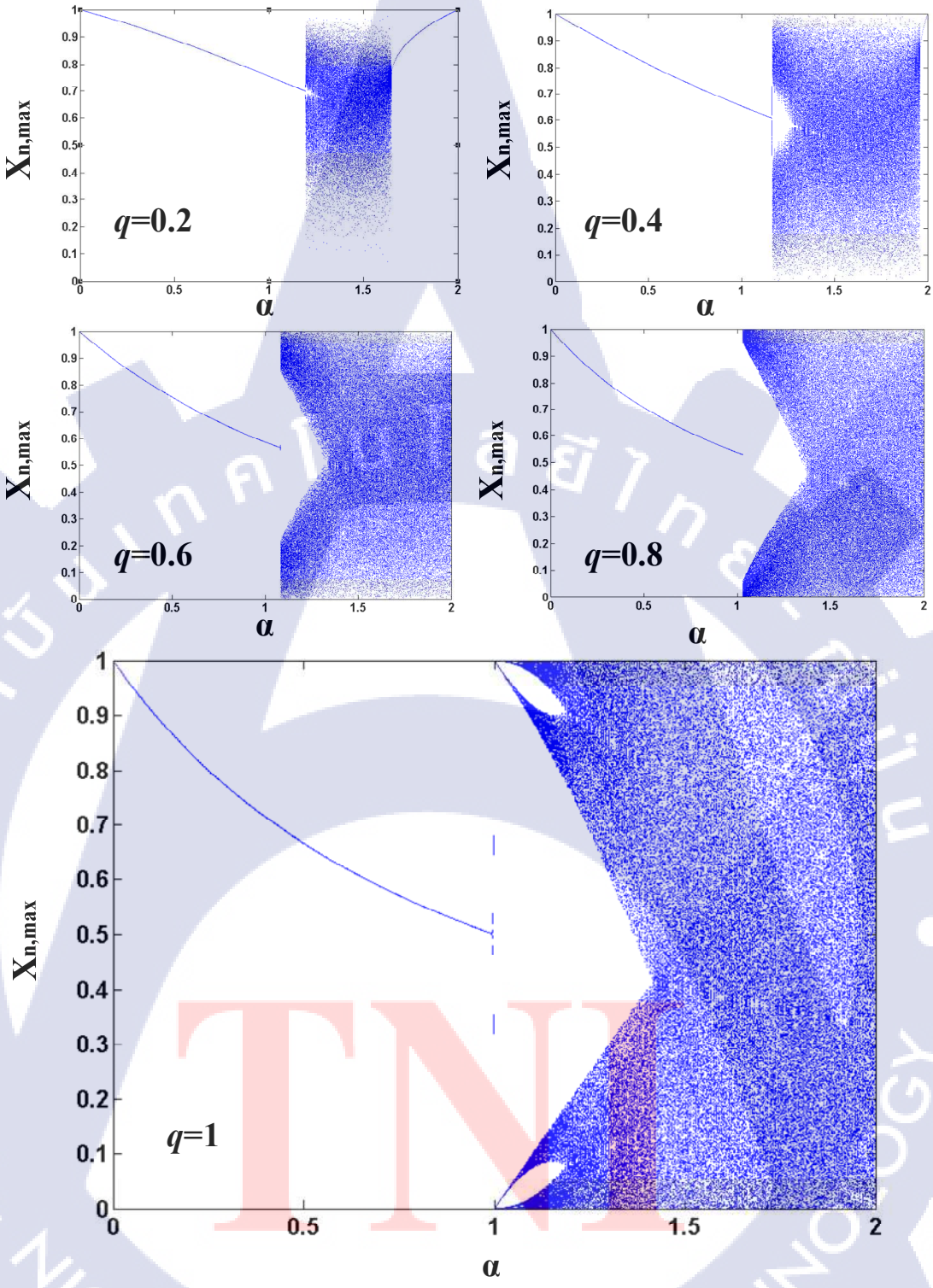


Figure 4.1 Developments of bifurcation ns of the proposed chaotic maps when the value of *q* is increased from 0.2 to 1 with a step size of 0.2.

generally represented by dotted lines, showing thick regions. On the other hand, the LE is defined as a quantity that characterizes the rate of separation of infinitesimally close trajectories and is expressed

$$\text{as } LE = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \log_2 \frac{d(x_{n+1})}{dx} \tag{2}$$

In order to analysis dynamics behaviors of system equilibria the Jacobian can be found through the absolute value of the first derivative as follows;

$$J(x_n) = \left| \frac{d(x_{n+1})}{dx} \right| = \left| \frac{d(|\alpha x_n - 1|^q)}{dx} \right| = \left| \frac{q\alpha |\alpha x_n - 1|^q}{\alpha x_n - 1} \right| \tag{3}$$

Typically, the chaotic map becomes unstable in the case where $J(x_n)>1$. From numerical simulation result of (2) maximum LE can be located at $q=1$ and α closed to 2. The system fixed points can be calculated by substituting x_n into x_{n+1} in (1) and omit the subscript n for simplicity. As a result, the equation becomes

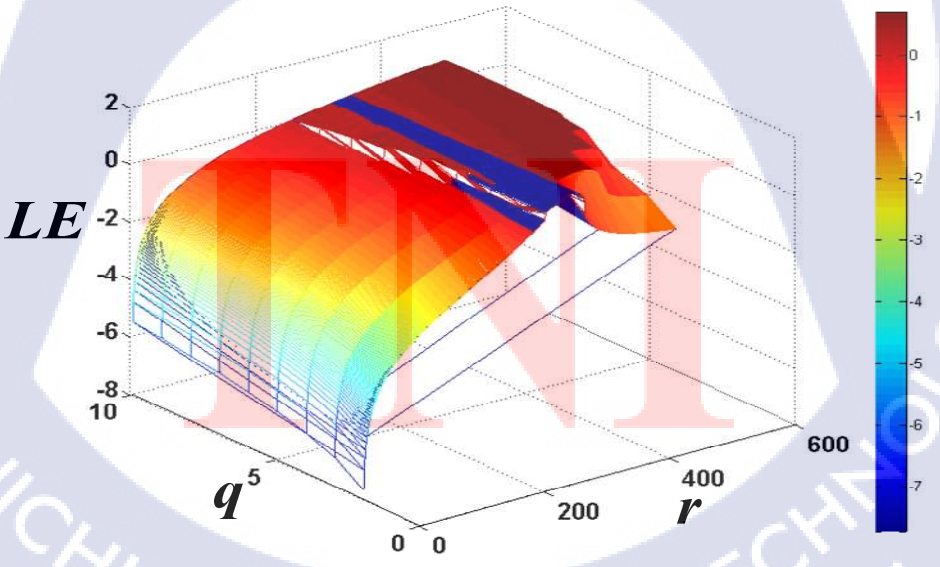


Figure 4.2 The three-dimensional LE spectrum where the maximum positive LE is largest when r approaches 600 (normalised to 2)

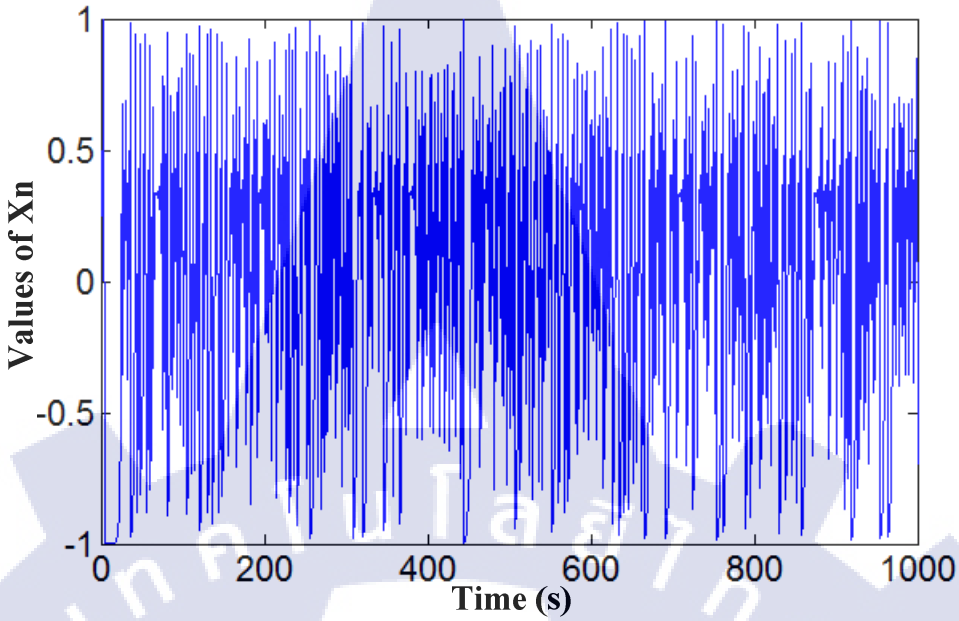


Figure 4.3 An apparently chaotic waveforms of the proposed chaotic map in time-domain over 0s to 1000s.

$$\left|\alpha x-1\right|^{q}=x \tag{4}$$

Solving Eq. (4) yields the fixed points. For case $q=1$, the fixed points (x_1^*, x_2^*) are given by

$$x_1^*=\frac{1}{\alpha-1} \quad x_2^*=\frac{1}{1+\alpha} \tag{5}$$

In order to find the value of α such that the proposed chaotic map operates under chaos regions, the maximum and minimum values of x_{n+1} is initially calculated by setting $J(x_n)=0$. In other words,

$$J(x_n)=\alpha sign(\alpha x_n-1)=0 \tag{6}$$

Therefore, the boundary of values of x_{n+1} depends upon the nonlinear term $sign(x_n)$ when $x_n > 0$ which falls in the region [0,1]. Consequently, the bifurcation parameter α

is limited to the region [0, 2]. Computer simulations of all bifurcation diagrams, the *LE* spectrum, Cobweb plots, and time-domain

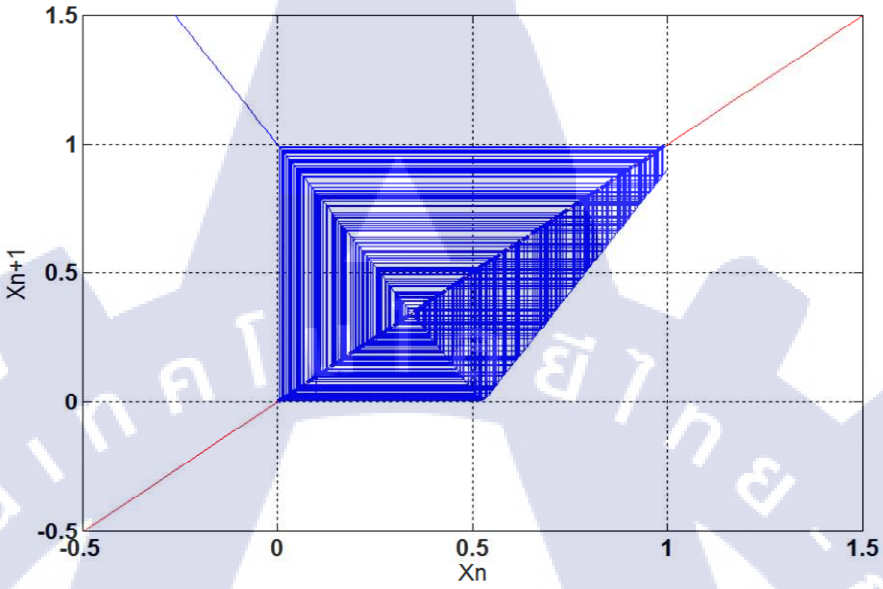


Figure 4.4 A Cobweb plot of the proposed chaotic map in Eq. (1).

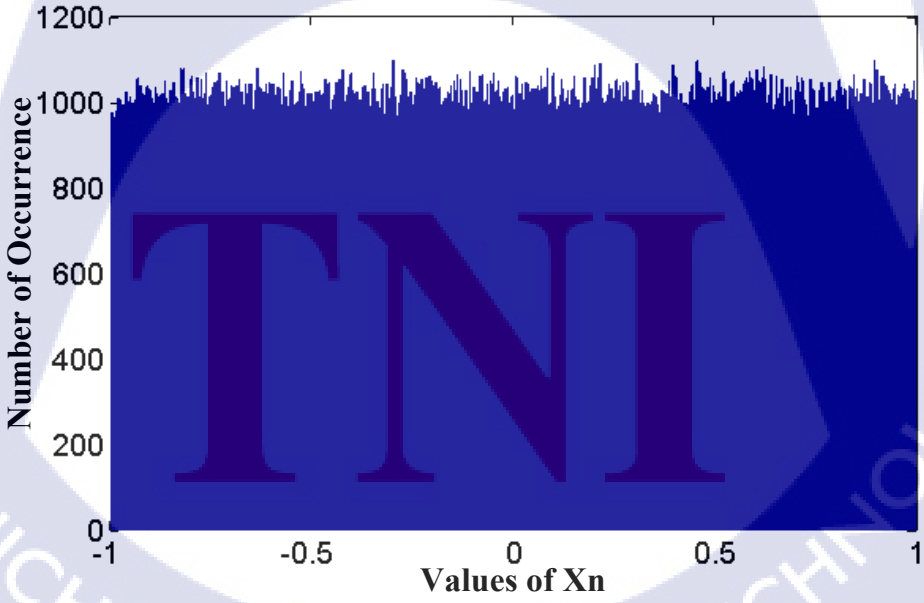


Figure 4.5 A histrogram of signal values over 1,000,000 itterations, showing an equally distributed values of *X*n.

waveform have been performed using MATLAB with an initial conditions of 0.1 and α = 1.99. Fig.4.1 shows developments of bifurcations of the proposed chaotic maps when the value of q is increased from 0.2 to 1 with a step size of 0.2. It is apparent that the smooth chaos appears at q=1, which is defined by the absence of periodic windows and coexisting attractors in some neighborhood of the parameter spaces. Fig.4.2 shows the three-dimensional LE spectrum where the maximum positive LE is largest when r approaches 600 (normalized to 2) and therefore the value of α = 1.99 was realized. Fig.4.3 demonstrates apparently chaotic waveforms of the proposed chaotic map in time-domain over 0s to 1000s. Finally, Fig.4.4 shows the Cobweb plot of the proposed chaotic map in Eq. (1).

4.2.2 Statistical Test

As for the illustrations, Fig.4.5 shows the apparently chaotic waveforms of the proposed chaotic map in time-domain over 0s to 1000s. In addition, Fig.4.3 shows the histogram of signal values over 1,000,000 iterations, showing equally distributed values of Xn. These characteristics show that the proposed chaotic map offer a relatively complex chaotic behaviors.

As for standard tests, the the National Institute of Standards and Technology (NIST) has provided a statistical tests suite in order to evaluate the randomness of binary sequences. This paper generates chaotic signals by the proposed two cases of the signum-based chaotic maps for 1,000,000 iterations and simply proceed a comparison with zero, i.e. bit “1” for any values that greater than zero and bit “0” for any values that smaller than zero. Subsequently, the NIST test suite from a special publication 800-22rev1a was realized using a typical 1,000,000 random bits. The test suite attempts to extract the presence of a pattern that indicates non-randomness of the sequences through probability methods described in terms of p-value. For each test methods, the p-value indicates the strength of evidence against perfect randomness hypothesis, i.e. a p-value greater than a typical confidence level of 0.01 implies that the sequence is considered to be random with a confidence level of 99%. Table 1 summaries NIST test results, indicating that the generated sequences from both cases of chaotic maps pass all standard 15 tests. As a result of the NIST tests, the randomness of proposed chaotic maps is sufficient for use as a mobility node model.

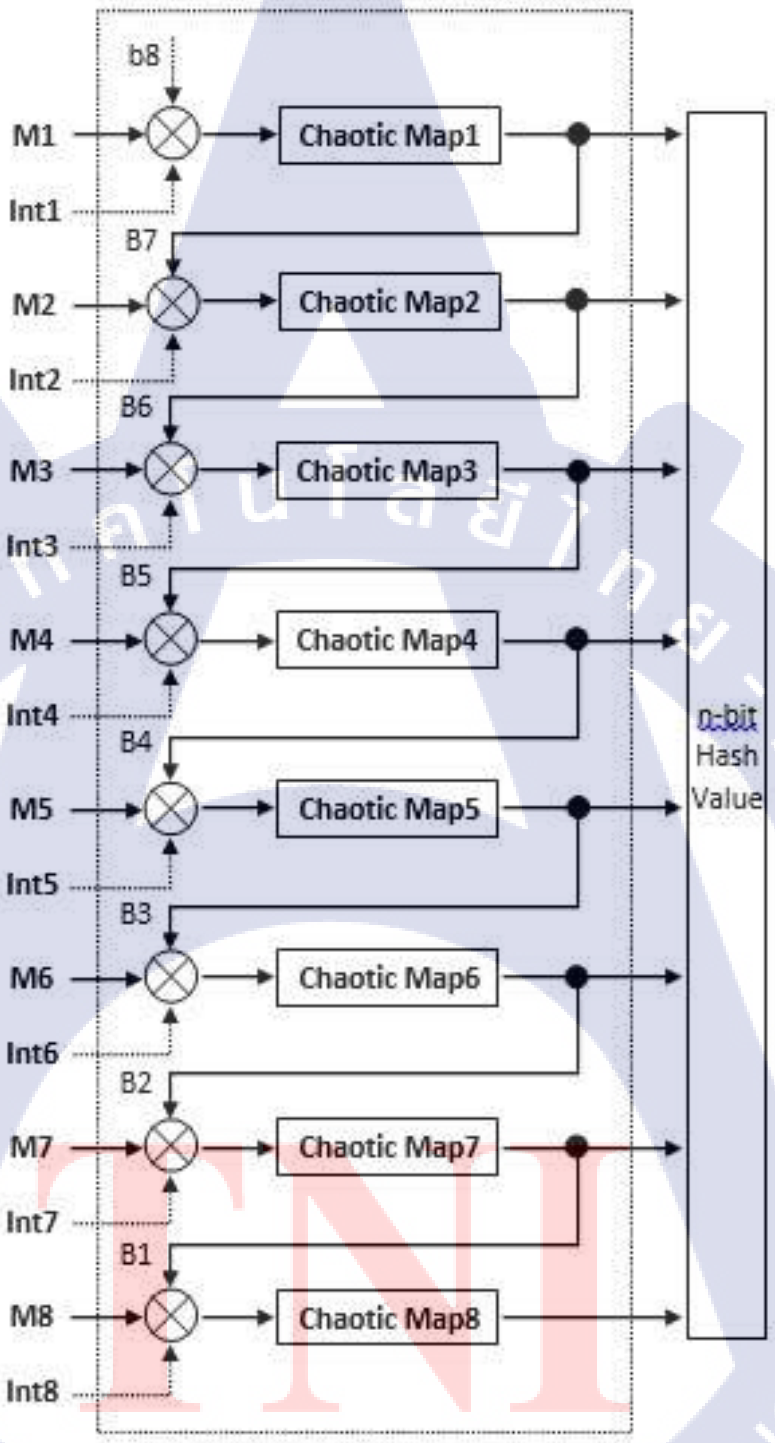


Figure 4.6 The proposed hash function comprises eight proposed chaotic maps connected in a circular network.

4.2.3 Proposed Hash Algorithm

The purpose of hash architecture design is to optimize for simple structure with few numbers of chaotic maps, but still 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.4.6 shows the proposed hash function comprises eight proposed 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.4.7 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 *n*-bit hash value. 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 proposed chaotic map.

4.2.4 Proposed Hash Algorithm

4.2.4.1 Distribution of hash value

One of the most important properties of hashing scheme is the uniform distribution of hash value, which is related to the security of hashing scheme. In Fig.4.8(a) the ASCII characters of the original message are localized within a small area from approximately 97 to 122. It can also be noticed that the “space” character (ASCII 32) is the most commonly used in the original message. In contrast, the hexadecimal hash values of the hashing scheme are uniformly distributed over the space of all possible hash values, as shown in Fig.4.8 (b). This indicates that no information of the original message is left after the diffusion and confusion processes.

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 (mod 8) = 0
end if
>> Divide the input message M into N sub-block of length 8 (Defined by S)
>> i, k, t = 1
>>Map M into integer with interval [0 , 1]
 # First Stage (Input Stage)
while (t < N)
while (i < 7)
>> $X(i,t) = |\alpha(M(k) \times X(i+1, t-1)) - 1|$
>> k = k+1
>> i = i+1
end
>> $X(8,t) = |\alpha(M(k) \times X(1, t-1)) - 1|$
>> k = k+1
>> t = t+1
end
>> i = 1
 # Second Stage
while (t < N+10)
while (i < 7)
>> $X(i,t) = |\alpha X(i+1, t-1) - 1|$
>> i = i+1
end
>> $X(8,t) = |\alpha X(1, t-1) - 1|$
>> t = t+1
end
>> H = Map X into integer with interval [0 , 2^n]
>> return H

Figure 4.7 Pseudo codes of the proposed hah function algorithm.

4.2.4.2 Sensitivity of hash value to the message and initial conditions

The aim of this subsection is to illustrate the high sensitivity of the proposed hashing scheme to tiny changes in the original message and the initial conditions. In order to investigate this issue, a series of experiments have been done under the following eight different conditions:

Case 1: The original message is: “Sensitivity of hash value to the message and initial conditions.”.

Case 2: Replace the first character of the original message by “A”. The two 128-bit hash values for C1 and C3 differ in 63 positions.

Case 3: Replace the character “i” in the word “initial” by “e” to become “enitial”. The two 128-bit hash values for

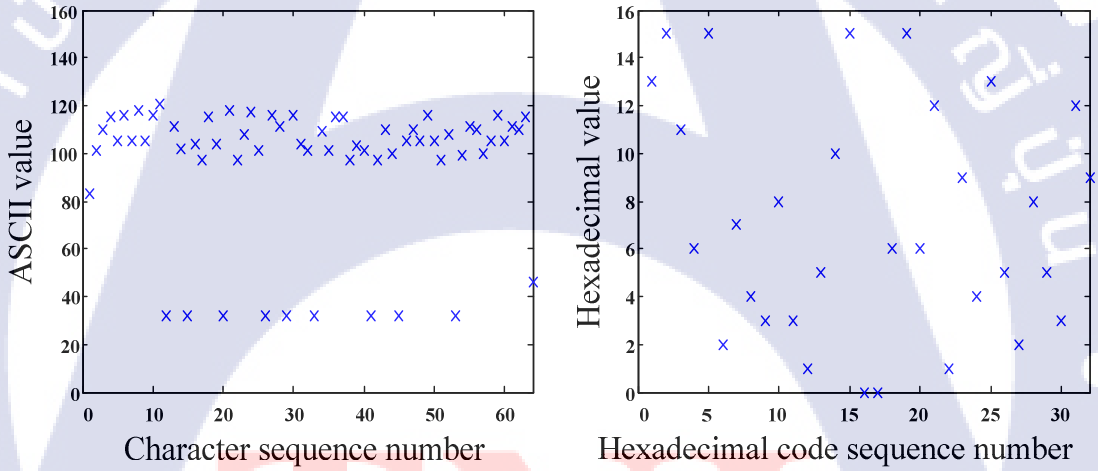


Figure 4.8 Spread of hash value: (a) distribution of the original message in ASCII: (b) distribution of the hash values in hexadecimal format.

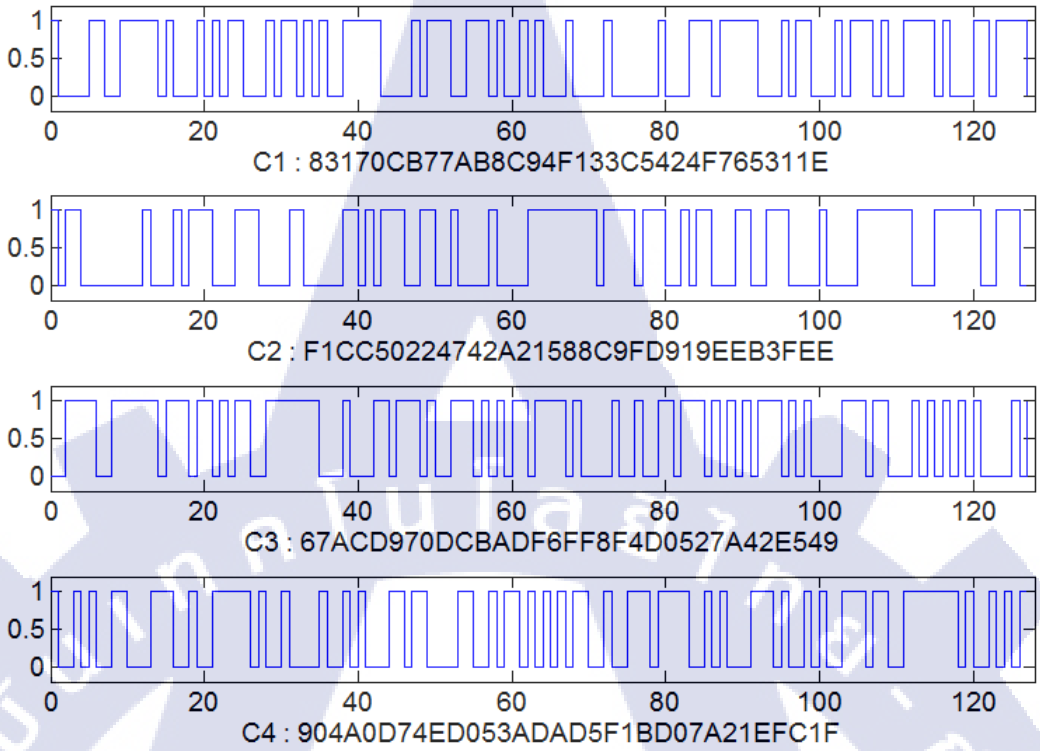


Figure 4.9 Binary sequences of hash values of eight conditions.

The corresponding graphical display of binary sequences is shown in Fig.4.9. The simulation result shows that any tiny change in the original message and initial condition leads to a 50% changing probability for each bit of hash value.

4.2.4.3 Statistical Analysis of Confusion and Diffusion

Confusion and diffusion are two essential design criteria for hashing scheme which are necessary to make it resistant to most attacks. Diffusion is intended to spread the original message statistics through the hash value in order to hide statistical properties of the original message. The aim of confusion is to use the transformation to make the relationship between the input bits and the output bits as complex as possible. The diffusion and confusion test has been performed as follows: a random message of length $L=50n$ is created and its $_n$-bit hash value is calculated. Then, a bit of the original message is randomly chosen and flipped, and the $_n$-bit hash value of the modified message is calculated. The two hash values are then compared

in order to quantify the number of changed bits. This experiment is performed N times for N = 256, 512, 1024, 2048 and 10000 for hash values of size n,

Table 4.1 Summary of Statistical measure methods.

Statistical measure methods		Formulas
(1)	Minimum changed bit number	$B_{\min} = \min\left(\{B_i\}_{i=1}^{N}\right)$
(2)	Maximum changed bit number	$B_{\max} = \max\left(\{B_i\}_{i=1}^{N}\right)$
(3)	Mean changed bit number	$\overline{B} = \frac{1}{N}\sum_{i=1}^{N} B_i$
(4)	Mean changed probability	$P = \frac{\overline{B}}{n} \times 100\%$
(5)	Standard deviation of the changed bit number	$\Delta B = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i - \overline{B})^2}$
(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 4.2 Statistical results for a 128-bit hash value

	N=256	**N=512**	**N=1024**	**N=2048**	**N=10000**
$B_{\min}$	46	45	45	44	42
$B_{\max}$	85	84	82	80	84
$\overline{B}$	63	63.04	63.23	62.65	62.9293
P (%)	49.22	49.25	49.4	48.94	49.16352
ΔB	6.19645569	6.627095	5.602838	5.479789	5.673732
ΔP (%)	4.840981007	5.177418	4.377217	4.281085	4.432603

Table 4.3 Statistical results for a 160-bit hash value

	N=256	**N=512**	**N=1024**	**N=2048**	**N=10000**
B_{min}	59	58	56	55	53
B_{max}	95	96	97	101	102
$\bar{B}$	78.35	78.89	78.66	78.86	78.9512
$_P$ (%)	48.97	49.31	49.16	49.28	49.3445
$_{\Delta B}$	6.117366	6.302809	6.261118	6.649782	6.561541
$_{\Delta P}$ (%)	3.823354	3.939256	3.913198	4.156114	4.100963

Table 4.4 Statistical results for a 256-bit hash value

	N=256	**N=512**	**N=1024**	**N=2048**	**N=10000**
B_{min}	103	101	102	102	99
B_{max}	146	154	154	152	158
$\bar{B}$	126.86	127.75	126.74	126.93	126.8779
$_P$ (%)	49.55	49.9	49.51	49.58	49.56168
$_{\Delta B}$	8.289252	8.555481	8.516115	7.904845	7.570246
$_{\Delta P}$ (%)	3.237989	3.341985	3.326608	3.08783	2.957127

where $_n$ = 128, 160 and 256. The six statistical measure methods, as listed in Table 4.2, are usually used for statistical analysis. B_i denotes the number of changed bits in the *i*-th test. Tables 4.3-4.5 list the results obtained in tests for $_n$ = 128, 160 and

256 and N = 256, 512, 1024, 2048 and 10000. Fig. 4.10 demonstrates the distribution of the number of bits changed for various values of N . The histogram distribution of the number of changed bits for $_n$ =256 and N =10000 is depicted in Fig.4.11. Based on the results in Tables 2-4, it is evident that the mean changed bit number $\bar{B}$ and the mean changed probability $_P$ are very close to the ideal values of $n/2$ and

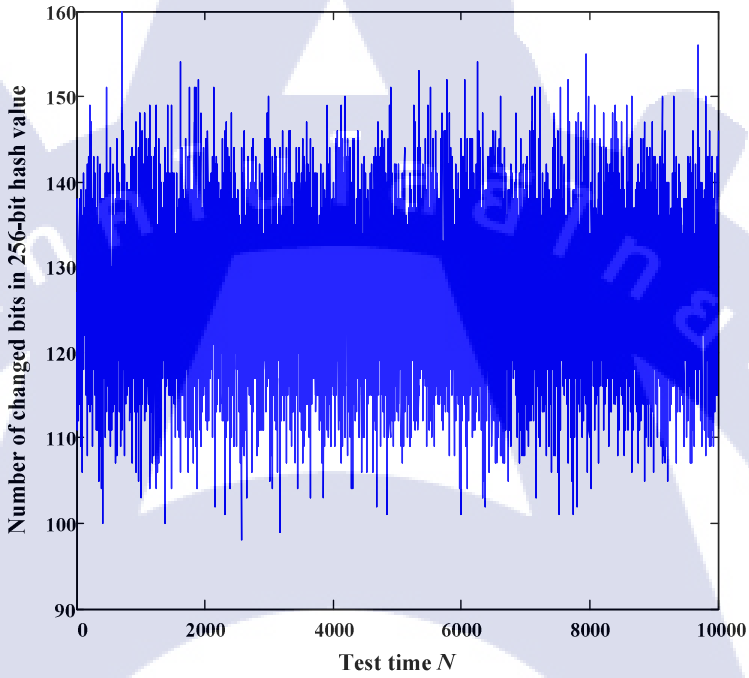


Figure 4.10 Binary sequences of hash values of eight conditions.

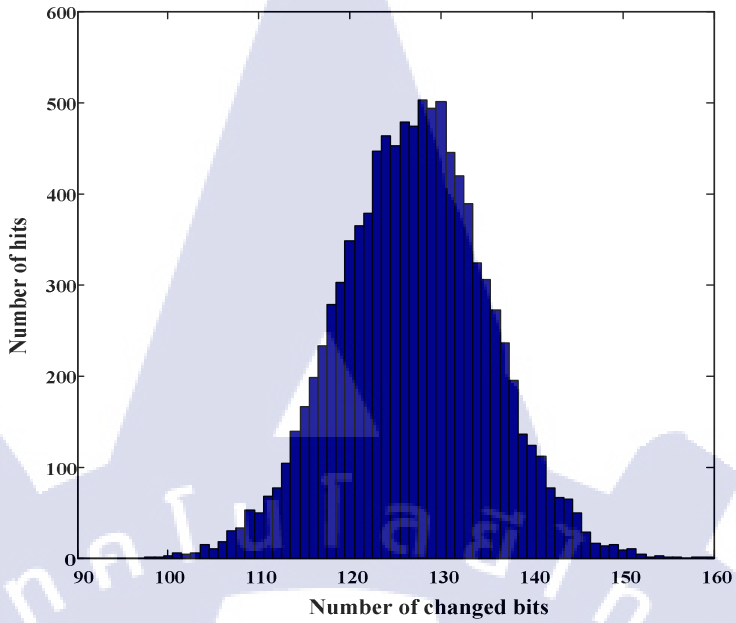


Figure 4.11 Histogram distribution of the number of changed bits for $_n$ =256 and N =10000.

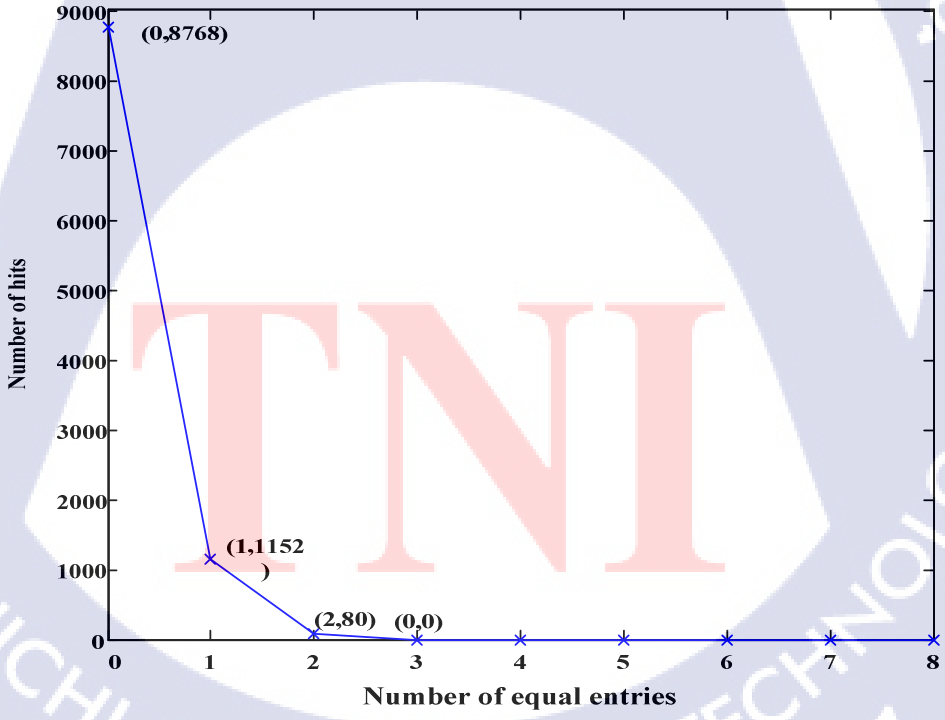


Figure 4.12 Distribution of the number of positions where the ASCII characters are the same for $_n$ =256 and N =10000

50% respectively. While both ΔB and ΔP are small for all tests, which indicates that confusion and diffusion capability of the proposed hashing scheme is stable. Collision Test Hash collision is a situation that occurs when two distinct input messages into a hash function produce the same hash values. In order to quantify the collision resistance of the proposed hashing scheme, the following collision test has been performed. The n-bit hash value of a random message of size $L=50n$ is created and stored in ASCII format. Then a bit in the random message is randomly chosen and toggled. The new hash value is created and stored in ASCII format. The two

Table 4.5 Statistical results for a 160-bit hash value

Absolute difference (*d*)	**Min.**	**Max.**	**Mean**
MD5(128bit)	590	2074	1304
SHA-1(160bit)	795	2730	1603
Zhang's scheme[17](128bit)	565	2022	1257
Kanso's scheme[18](128bit)	737	2320	1494
Wang's scheme [19] (128bit)	689	2295	1526
Ren's scheme [20] (128bit)	599	2455	1439
Wang's scheme [21] (128bit)	655	2064	1367
Our proposed scheme (128bit)	544	2400	1348
Our proposed scheme (160bit)	809	2782	1687
Our proposed scheme (256bit)	1402	3954	2716

hash values are compared with each other, and the number of ASCII characters with the same value at the same location, referred to as the number of hits, is counted. The absolute difference of the two hash values is computed as follows:

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

where m_i and m_i' denote the i-th ASCII character of the original and new hash value, respectively, and $dec(\ )$ maps m_i and m_i' to their equivalent decimal values.

This kind of test has been performed 10000 times. The minimum, maximum and mean values of d are presented in Table 4.7 and a plot of the distribution of the number of hits is illustrated in Fig.4.12. It can be noticed that the maximum number of equal character is only 2 and the collision probability is very low. It can be observed from Table 4.6 that for $_n$=128 and $_n$=160, the proposed hashing scheme provides better results than the existing algorithms such as MD5 and SHA-1, and comparable results to other chaos-based algorithms.

4.2.4.4 Resistance to birthday attach

A birthday attack is a kind of cryptographic attack that is based on mathematical behind the birthday problem in probability theory. It gets its name 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 example, if the size of the hash value is set to 256, the difficulty of the attack is 2^{128}. This keeps the system robust against this type of attack.

4.3 Conclusions

The design of communication protocols and message authentication in the Mobile Ad hoc Networks is challenging due to limited wireless transmission ranges of node mobility, limited power resources, and limited physical security. Assuring secure routing protocols is challenging since MANET wireless networks are highly vulnerable to security attacks. Most traditional routing protocols and message designs do not address security, and are mainly based on a mutual trust relationship among nodes. The new compact and robust chaos-based keyed hash function has been presented. The proposed chaotic map exploits absolute-value nonlinearity for

generating 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, and signal characteristics in time domains. The proposed hashing structure is relatively simple that enhances randomness quality for statistical performances. The designed hashing algorithms involve the initial stage when the chaotic 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 changed bit number that is also close to a half of hash value lengths. In addition, the collision tests proffer the average mean of 1359 and 1603 for the hash values of 128 bits and 160 bits, respectively. The proposed has function has superior performance over well-known algorithms such as MD5 and SHA1, and is comparable to other complex structures of chaos-based approaches. As a result, the proposed hash function has offered a potential alternative to protocol and message authentication methods in Ad Hoc Networks.

Chapter 5
Conclusion

5.1 Introduction

This chapter presents the conclusion of the A Chaos-Based Key Hash Function for Mobile Ad Hoc Wireless Application

5.2 Conclusion

The design of communication protocols and message authentication in the Mobile Ad hoc Networks is challenging due to limited wireless transmission ranges of node mobility, limited power resources, and limited physical security. Assuring secure routing protocols is challenging since MANET wireless networks are highly vulnerable to security attacks. Most traditional routing protocols and message designs do not address security, and are mainly based on a mutual trust relationship among nodes. The new compact and robust chaos-based keyed hash function has been presented. The proposed chaotic map exploits absolute-value nonlinearity for generating 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, and signal characteristics in time domains. The proposed hashing structure is relatively simple that enhances randomness quality for statistical performances. The designed hashing algorithms involve the initial stage when the chaotic 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 changed bit number that is also close to a half of hash value lengths. In addition, the collision tests proffer the average mean of 1359 and 1603 for the hash values of 128 bits and 160 bits, respectively. The proposed has function has superior performance over well-known algorithms such as MD5 and SHA1, and is comparable to other complex structures of chaos-based approaches. As a result, the proposed hash

function has offered a potential alternative to protocol and message authentication methods in Ad Hoc Networks.

References

References

[1] L. Feng et al., "Security bootstrap model of key pre-sharing by polynomial group in mobile Ad Hoc Network," *Journal of Network and Computer Applications*, vol. 32, no. 4, pp. 781–787, July 2009.

[2] S. Behnia et al., "Applications of tripled chaotic maps in cryptography," *Chaos Solitons & Fractals*, vol. 40, pp. 505–519, May 2009.

[3] H. Yang et al., "One-way hash function construction based on chaotic map network," *Chaos, Solitons and Fractals*, vol. 41, no. 5, pp. 2566–2574, September 2009.

[4] R. Sparr and R. Wernsdorf, "Group theoretic properties of Rijndael-like ciphers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3139-3149, September 2008.

[5] Q. Zhou et al., "Parallel image encryption algorithm based on discretized chaotic map," *Chaos Solitons &Fractals*, vol 38, no. 4, p. 1081–1092, November 2008.

[6] H. H. Nien et al., "Digital color image encoding and decoding using a novel chaotic random generator," *Chaos, Solitons & Fractals*, vol. 32, no. 3, pp. 1070–1080, May 2007.

[7] K. W. Wong, "A combined chaotic cryptographic and hashing scheme," *Physics Lettets A*, vol. 307, no. 5-6, pp. 292–298, February 2003.

[8] B. O. Brachtl et al., "Data *authentication using modification detection codes based on a public one way encryption function*," [Online]. Available: http://www.google.com/parents/US4908861. [Accessed: January 13, 2014].

[9] W. Hohl, et al., "Security of iterated hash functions based on block ciphers," *Lecture Notes in Computer Science*, vol. 773, pp. 379–390, 1994.

[10] L. R. Knudsen, and B. Preneel, "Fast and secure hashing based on codes," *Lecture Notes in Computer Science*, vol. 1294, pp. 485–498, 1997.

[11] J. Zhang et al., "Chaotic keyed hash function based on feedforward-feedback nonlinear digital filter," *Physics Lettets A*, vol. 362, no. 5-6, pp. 439–48, March 2007.

Appendix

A.1 Bifurcation of logistic map

```
scale = 10000; % determines the level of rounding
maxpoints = 100; % determines maximum values to plot
N = 5000; % number of "r" values to simulate
a = 0; % starting value of "r"
b = 4; % final value of "r"... anything higher diverges.
rs = linspace(a,b,N); % vector of "r" values
M = 1000; % number of iterations of logistics equation

% Loop through the "r" values
for j = 1:length(rs)
   r=rs(j); % get current "r"
   y=zeros(M,1); % allocate memory
   y(1) = 0.00005; % initial condition (can be anything from 0 to 1)
   y(2) = 0.1;
   x(1) = 0.05;

   for i = 2:M, % iterate

     y(i+1) = r-y(i)^2;         %simplest logistic map Map r=[0 2]
     %y(i+1) = r*(1-y(i)^2);     %simple logistic map Map r=[0 1.4]
     %y(i+1) = r*y(i)*(1-y(i));  %logistics Map r=[0 4]
     %y(i+1) = r*sin(y(i));      %sine map r=[0 3.1]
     %y(i+1) = r*cos(y(i));      %cos map r=[0 2.5]

   end
   out{j} = unique(round(scale*y(end-maxpoints:end)));
end

% Rearrange cell array into a large n-by-2 vector for plotting
data = [];
```

```
for k = 1:length(rs)
   n = length(out{k});
   data = [data;  rs(k)*ones(n,1),out{k}];
end

% Plot the data
figure(99);clf
h=plot(data(:,1),data(:,2)/scale,'b.');
%g=title('y(i+1) = cos(r*y(i)+X)');
set(h,'markersize',1)
%set(g,'Visible','on');
```

A.2 Bifurcation of Structure

```
% Henon Map : Bifurcation Structure
% IES LAB TNI, Bangkok, Thailand, 2013.

function Bifurcation_Structure
   clear all
   clc

   Parameter_Range_x    = [0, 4];
   Parameter_Range_y    = [0, 0.4];

   Number_Of_Sampling_x = 100;
   Number_Of_Sampling_y = 100;

   Step_x  = (Parameter_Range_x(2) - Parameter_Range_x(1)) /
Number_Of_Sampling_x;
   Step_y  = (Parameter_Range_y(2) - Parameter_Range_y(1)) /
Number_Of_Sampling_y;

   global r q
   for Count_x = 1: 1: Number_Of_Sampling_x + 1
      r = (Count_x - 1) * Step_x + Parameter_Range_x(1)
      for Count_y = 1: 1: Number_Of_Sampling_y + 1
         q = (Count_y - 1) * Step_y + Parameter_Range_y(1);
         [~, ~, DKY(Count_x, Count_y)] = Lyapunov_Exponents_2_Dimensional(50,
0.001, [0, 0]);
      end
   end

   % ----------------------- Map  Algorithm : RGB -------------------- %
   DKY_Max = real(max(max(DKY)))
```

```
   DKY_Min = real(min(min(DKY)))
   for Count_x = 1: 1: Number_Of_Sampling_x + 1
      for Count_y = 1: 1: Number_Of_Sampling_y + 1
         if DKY(Count_x, Count_y) > 0              % -    Choas   - %
            % # Gold      -> rgb(255, 215,   0) -> #ffd700          %
            % # Goldenrod -> rgb(218, 165,  32) -> #daa520          %
            Map(Count_x, Count_y, 1) = (255)*(1-(real(DKY(Count_x,
Count_y))/DKY_Max));
            Map(Count_x, Count_y, 2) = (255)*(1-(real(DKY(Count_x,
Count_y))/DKY_Max));
            Map(Count_x, Count_y, 3) = (255);
         elseif DKY(Count_x, Count_y) <= 0          % - Not  Choas - %
            Map(Count_x, Count_y, 1) = (255);
            Map(Count_x, Count_y, 2) = (255);
            Map(Count_x, Count_y, 3) = (255);
         else                                       % - Can't find - %
            Map(Count_x, Count_y, 1) = (255);
            Map(Count_x, Count_y, 2) = (255);
            Map(Count_x, Count_y, 3) = (255);
         end
      end

   end
   MapRGB(:, :, 1) = flipud(uint8(Map(:, :, 1)));
   MapRGB(:, :, 2) = flipud(uint8(Map(:, :, 2)));
   MapRGB(:, :, 3) = flipud(uint8(Map(:, :, 3)));
   % ------------------------------------------------------------------ %
   imtool(MapRGB)
   figure(1)
   surface((1: 1: Number_Of_Sampling_x + 1) * Step_x, (1: 1:
Number_Of_Sampling_y + 1) * Step_y, real(DKY))
end
```

A.3 Function Map

```
% Function Map : Henon Map
% IES LAB TNI, Bangkok, Thailand, 2013.

function Output = Function_Map(Input)
   global r q
   Output = zeros(size(Input));

   x_old = Input(1);
   x_now = Input(2);

   y_now = + x_old;
   x_new = - ((r.^2) * (x_now.^(1 + q))) + y_now + 1;

   Output(1) = x_now;
   Output(2) = x_new;
end
```

A.4 Logistics

```
x0 = 0.5; % Initial condition
N  = 1000; % Number of iterations
r=3.8;
% Matrices in matlab cannot have zero index
x = zeros(N,1);
x(1) = x0;
 % compute the orbit and print out results
for i=1:N

 % Logistics Map
 %x(i+1) = r*x(i)*(1-x(i));

 % sine map
 x(i+1) = r*sin(x(i));

end
 % graph the orbit
plot(x);
xlabel('t'); ylabel('Xn');
hold on
```

A.5 Lyapunov_Exponents_2_Dimensional

```
% Lyapunov Exponent & Kaplan-Yorke Dimension
% Map Method :
%   [1] T. S. Parker and L. O. Chua, Practical numerical algorithms for chaotic
systems, Springer–Verlag, New York, 1989.
%   [2] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, Determining
Lyapunov exponents from a time series, Physica D 16 (1985), 285–317.
% IES LAB TNI, Bangkok, Thailand, 2013.

function [LE1, LE2, DKY] = Lyapunov_Exponents_2_Dimensional
(Number_of_Iterates, Perturbation_Size, Initial_Conditions)

  n   = Number_of_Iterates; % Good at 10000
  eps = Perturbation_Size;  % Good at 0.001
  x   = Initial_Conditions;
  v1  = [1 ,0];
  v2  = [0 ,1];
  sum = [0, 0];

  for k = 1: 1: n
     v1(1) = v1(1) * eps;
     v1(2) = v1(2) * eps;
     v2(1) = v2(1) * eps;
     v2(2) = v2(2) * eps;

     v1(1) = v1(1) + x(1);
     v1(2) = v1(2) + x(2);
     v2(1) = v2(1) + x(1);
     v2(2) = v2(2) + x(2);

     v1 = Function_Map(v1);
```

```
        v2 = Function_Map(v2);
        x  = Function_Map( x);

        v1(1) = (v1(1) - x(1)) / eps;
        v1(2) = (v1(2) - x(2)) / eps;
        v2(1) = (v2(1) - x(1)) / eps;
        v2(2) = (v2(2) - x(2)) / eps;

        [norm, v1, v2] = GSR(v1, v2);

        sum(1) = sum(1) + log(norm(1));
        sum(2) = sum(2) + log(norm(2));
    end
    LE1 = sum(1) / k;
    LE2 = sum(2) / k;
    DKY = 1 + LE1 / abs(LE2);
end

function Output = Norm_Function(Input)
    Output = sqrt(Input(1).^2 + Input(2).^2);
end

function Output = Dot_Product(x, y)
    Output = x(1) * y(1) + x(2) * y(2);
end

function [norm, new_v1, new_v2] = GSR(v1, v2)
    norm(1) = Norm_Function(v1);
    v1(1)   = v1(1) / norm(1);
    v1(2)   = v2(2) / norm(1);
    Vector  = Dot_Product(v1, v2);
    x(1)    = v2(1) - Vector * v1(1);
```

```
    x(2)    = v2(2) - Vector * v1(2);
    norm(2) = Norm_Function(x);
    v2(1)   =  x(1) / norm(2);
    v2(2)   =  x(2) / norm(2);

    new_v1 = v1;
    new_v2 = v2;
end
```

A.6 Lyapunov_Exponents_2_Dimensional running

```
% Henon Map : Lyapunov Exponent & Kaplan-Yorke Dimension
% IES LAB TNI, Bangkok, Thailand, 2013.

function Run_Lyapunov_Exponents_2_Dimensional
   clear all
   clc

   Parameter_Range    = [0, 1];
   Number_Of_Sampling = 1000;
   Step  = (Parameter_Range(2) - Parameter_Range(1)) / Number_Of_Sampling;

   global r q
   q = 1;
   for Count = 1: 1: Number_Of_Sampling + 1
      r = (Count - 1) * Step + Parameter_Range(1)
      Parameter(Count) = r;
      [LE1(Count), LE2(Count), DKY(Count)] =
Lyapunov_Exponents_2_Dimensional(1000, 0.001, [0, 0]);
   end

   subplot(1, 2, 1)
   plot(Parameter, DKY);
   xlabel(                  'Parameter a', 'FontSize', 16, 'FontName', 'Cordia New',
'FontWeight', 'bold');
   ylabel('Henon Map : Kaplan-Yorke Dimension', 'FontSize', 16, 'FontName', 'Cordia
New', 'FontWeight', 'bold');
   grid on;

   subplot(1, 2, 2)
   plot(Parameter, LE1, Parameter, LE2);
```

```
    xlabel(                    'Parameter a', 'FontSize', 16, 'FontName', 'Cordia New',
'FontWeight', 'bold');
    ylabel(    'Henon Map : Lyapunov Exponents', 'FontSize', 16, 'FontName', 'Cordia
New', 'FontWeight', 'bold');
    grid on;
end
```

Science and Information Conference 2015
July 28-30, 2015 | London, UK

A Chaos-Based Keyed Hash Function for Secure Protocol and Messege Authentication in Mobile Ad Hoc Wireless Networks

Winai Chankasame and Wimol San-Um

Intelligent Electronic Systems (IES) Research Laboratory
Master Program of Engineering Technology
Faculty of Engineering, Thai-Nichi Institute of Technology (TNI)
Patthanakarn 37, Suanlaung, Bangkok, Thailand, 10250. Fax :(+662)-763-2700, Tel :(+662)-763-2600
Corresponding Author E-mail addresses: *wimol@tni.ac.th

Abstract— **The design of communication protocols in the Mobile Ad hoc Networks (MANET) is challenging due to limited wireless transmission ranges of node mobility, limited power resources, and limited physical security. The advantages of MANET include simple and fast deployment, robustness, adaptive and self-organizing networks. Nonetheless, routing protocols are important operations for communication among wireless devices. Assuring secure routing protocols is challenging since MANET wireless networks are highly vulnerable to security attacks. Most traditional routing protocols and message authentication designs do not address security, and are mainly based on a mutual trust relationship among nodes. This paper therefore proposes a new chaos-based keyed hash function that can be used for communication protocols in MANET. The proposed chaotic map realizes an absolute-value nonlinearity, which offers robust chaos over wide parameter spaces, i.e. high degree of randomness through chaoticity measurements using Lyapunov exponent. The proposed keyed hash function structure is compact through the use of a single stage chaos-based topology. Hash function operations involve an initial stage when the chaotic map accepts input message and initial conditions, and a hashing stage where alterable-length hash values are generated iteratively. Hashing performances are evaluated in terms of original message condition changes, statistical analyses, and collision analyses. Results of hashing performances show that the mean changed probabilities are very close to 50%, and the mean changed bit number is also close to a half of hash value lengths. The proposed keyed hash function enhances the collision resistance, comparing to typical MD5 and SHA1, and is faster than other complicated chaos-based approaches.**

Keywords-Mobility Management; Clustering; Topology control; Coverage, and connectivity; synchronization; Social networks.

I. INTRODUCTION

A mobile Ad Hoc NETwork (MANET) has been emerged for a variety of applications such as in emergency search and rescue operations, in battlefields, or in setting up of instant communication among moving vehicles [1]. Such a Mobile Ad Hoc Network normally is a type of multi-hop network formed temporally by a group of nonnfrastructure wireless mobile nodes that relays data packets from one to another in order to communicate with the remote nodes beyond the radio distance [2]. In contrast to conventional architectures, the public media, the change of network topology, cooperation algorithms, the lack of centralized monitoring, and system management are vulnerable to mischievous attacks. Typically, three types of Ad Hoc network protocols are (1) Proactive protocols (DSDV, WRP, CGSR, STAR, and OLSR), (2) Reactive protocols (ABR, DSR, TORA, AODV, CBRP, RDMAR) and (3) Hybrid protocol (ZRP). Nonetheless, two kinds of attacks in Ad Hoc Network are passive and active attacks. In passive attacks, the routing protocol is does not disturbed, but its only eavesdrops o routing traffic and extract the valuable information. However, malicious nodes in active network may disrupt and change the functions of a routing protocols I three possible approaches, i.e. routing information modification, false routing information fabrication, and node impersonation [3]. Consequently, security issue has essentially becoming more challenging for secure protocols in Ad Noc networks. Recently, a variety of secure protocols have been suggested, for instance, a q-composite model [4] that increases the capability of anti-capture of protocols, a basic random key pre-distribution model by generating key pool and allotting its different parts to each node [5-7]. Other research approaches also studied key distributions through either random keys or hash table for authentications.

This paper alternatively focuses on an authentication of Ad Hoc wireless protocols and message through the validation using a hash function. Such a hash function has been a key technology in advanced cryptography, which encodes an arbitrary length input message into a hash value with fixed length. Typically, 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 general, preferable characteristics of hash functions include high possibility of collision resistance and high security against preimage and second-preimage attacks.

Typical hash functions, for instance, Message Digest (MAD, MAD) and Secure Has Algorithm (SHA-1) have generally been exploited in software industries for purposes of integrity verification of electronically transmitted files as well as security in protocols. However, the typical hash functions have been designed based on logical operations or multi-round iterations, and therefore the hashing process efficiency, which depends upon inherent ciphers and complicated computation processes, are necessarily required. Recently, it has been notified through the analysis of the collision frequencies that those of typical hash functions contain several undiscovered flaws [2]. Therefore, multiple-block-length hash functions have been suggested [3-5]. Nonetheless, designs and implementation of such multiple-block-length hash function are relatively complicated in terms of security and efficiency.

As a pervasive feature in nature, chaos is a deterministic process generated by nonlinear dynamical systems that possess distinctive properties in pseudo-randomness, sensitivity to initial conditions and control parameters. Due to these properties, chaos-based hash algorithms have been of much interest as an alternative to that of typical hash functions. Several chaos-based hash function algorithms have therefore been proposed recently [6–910]. 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 [11]. 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.

This paper therefore proposed chaotic map that realizes a powered absolute-value nonlinearity, which offers robust chaos over wide parameter spaces, i.e. high degree of randomness through chaoticity measurements using Lyapunov exponent. The proposed keyed hash function structure is compact through the use of a single stage chaos-based topology for use as protocol verification. Hash function operations involve an initial stage when the chaotic map accepts input message and initial conditions, and a hashing stage where alterable-length hash values are generated iteratively. Hashing performances are evaluated in terms of original message condition changes, statistical analyses, and collision analyses.

II. Proposed Chaotic Map using Power Absolute-Value Nonlinearity

This paper proposed a simple chaotic map given by

$$x_{n+1} = |\alpha x_n - 1|^q \tag{1}$$

As for purposes of chaos measurements, preliminary investigations of the three chaotic maps are performed by bifurcation diagrams and Lyapunov exponent (LE) as for qualitative and quantitative measures of chaos, respectively. The bifurcation diagram indicates possible long-term values, involving fixed points or periodic orbits, of a system as a function of a bifurcation parameter. The stable solution is represented by a straight line while the unstable solutions are

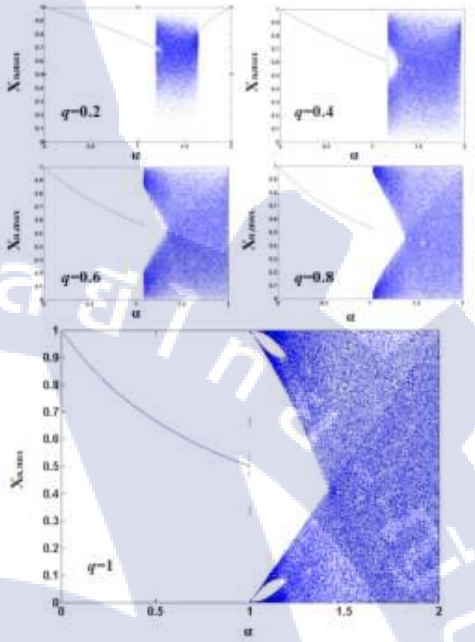


Figure 1. Developments of bifurcations of the proposed chaotic maps when the value of q is increased from 0.2 to 1 with a step size of 0.2.

generally represented by dotted lines, showing thick regions. On the other hand, the LE is defined as a quantity that characterizes the rate of separation of infinitesimally close trajectories and is expressed

$$\text{as } LE = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \log_2 \frac{d(x_{n+1})}{dx} \tag{2}$$

In order to analysis dynamics behaviors of system equilibria the Jacobian can be found through the absolute value of the first derivative as follows;

$$J(x_n) = \left| \frac{d(x_{n+1})}{dx} \right| = \left| \frac{d(|\alpha x_n - 1|^q)}{dx} \right| = \left| \frac{q\alpha |\alpha x_n - 1|^q}{\alpha x_n - 1} \right| \tag{3}$$

Typically, the chaotic map becomes unstable in the case where $J(x_n)>1$. From numerical simulation result of (2) maximum LE

can be located at $q = 1$ and α closed to 2. The system fixed points can be calculated by substituting x_n into x_{n+1} in (1) and omit the subscript n for simplicity. As a result, the equation becomes

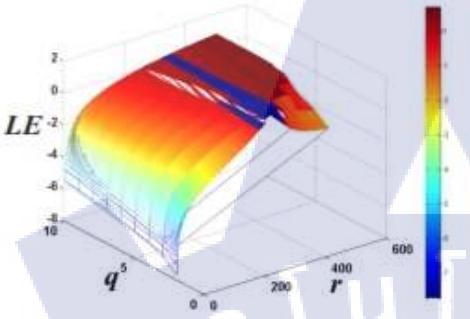


Figure 2. The three-dimensional LE spectrum where the maximum positive LE is largest when r approaches 600 (normalised to 2)

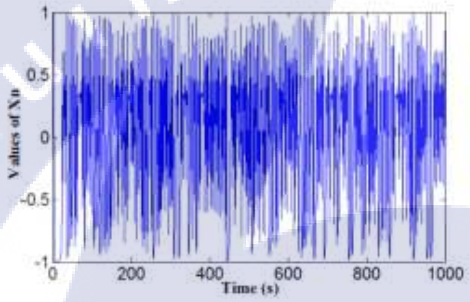


Figure 3. An apparently chaotic waveforms of the proposed chaotic map in time-domain over 0s to 1000s.

$$\left|\alpha x - 1\right|^{q} = x \tag{4}$$

Solving Eq. (4) yields the fixed points. For case $q = 1$, the fixed points (x_1^*, x_2^*) are given by

$$x_1^* = \frac{1}{\alpha - 1} \quad x_2^* = \frac{1}{1 + \alpha} \tag{5}$$

In order to find the value of α such that the proposed chaotic map operates under chaos regions, the maximum and minimum values of x_{n+1} is initially calculated by setting $J(x_n)=0$. In other words,

$$J(x_n) = \alpha sign(\alpha x_n - 1) = 0 \tag{6}$$

Therefore, the boundary of values of x_{n+1} depends upon the nonlinear term $sign(x_n)$ when $x_n > 0$ which falls in the region [0,1]. Consequently, the bifurcation parameter α is limited to the region [0, 2]. Computer simulations of all bifurcation diagrams, the *LE* spectrum, Cobweb plots, and time-domain

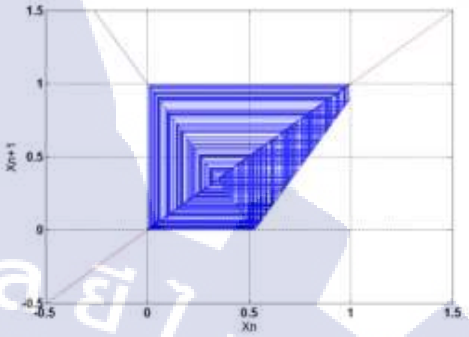


Figure 4. A Cobweb plot of the proposed chaotic map in Eq. (1).

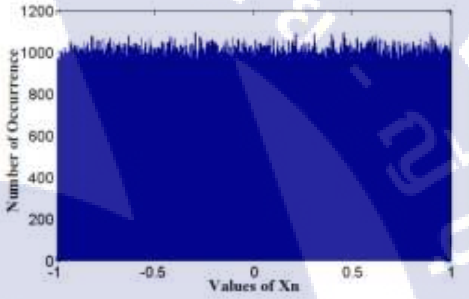


Figure 5. A histogram of signal values over 1,000,000 iterations, showing an equally distributed values of Xn.

waveform have been performed using MATLAB with an initial conditions of 0.1 and α = 1.99. Fig.1 shows developments of bifurcations of the proposed chaotic maps when the value of q is increased from 0.2 to 1 with a step size of 0.2. It is apparent that the smooth chaos appears at q=1, which is defined by the absence of periodic windows and coexisting attractors in some neighborhood of the parameter spaces. Fig.2 shows the three-dimensional LE spectrum where the maximum positive LE is largest when r approaches 600 (normalized to 2) and therefore the value of α = 1.99 was realized. Fig.3 demonstrates apparently chaotic waveforms of the proposed chaotic map in time-domain over 0s to 1000s. Finally, Fig.4 shows the Cobweb plot of the proposed chaotic map in Eq. (1).

III. STATISTICAL TESTS

As for the illustrations, Fig.5 shows the apparently chaotic waveforms of the proposed chaotic map in time-domain over 0s to 1000s. In addition, Fig. 3 shows the histogram of signal values over 1,000,000 iterations, showing equally distributed values of *Xn*. These characteristics show that the proposed chaotic map offer a relatively complex chaotic behaviors.

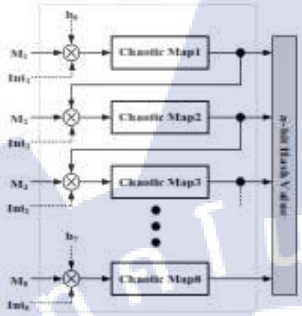


Figure 6. The proposed hash function comprises eight proposed chaotic maps connected in a circular network

As for standard tests, the the National Institute of Standards and Technology (NIST) has provided a statistical tests suite in order to evaluate the randomness of binary sequences. This paper generates chaotic signals by the proposed two cases of the signum-based chaotic maps for 1,000,000 iterations and simply proceed a comparison with zero, i.e. bit "1" for any values that greater than zero and bit "0" for any values that smaller than zero. Subsequently, the NIST test suite from a special publication 800-22rev1a was realized using a typical 1,000,000 random bits. The test suite attempts to extract the presence of a pattern that indicates non-randomness of the sequences through probability methods described in terms of p-value. For each test methods, the p-value indicates the strength of evidence against perfect randomness hypothesis, i.e. a p-value greater than a typical confidence level of 0.01 implies that the sequence is considered to be random with a confidence level of 99%. Table 1 summaries NIST test results, indicating that the generated sequences from both cases of chaotic maps pass all standard 15 tests. As a result of the NIST tests, the randomness of proposed chaotic maps is sufficient for use as a mobility node model.

IV. ROPOSED HASH ALGORITHM

The purpose of hash architecture design is to optimize for simple structure with few numbers of chaotic maps, but still 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.6 shows the proposed hash function comprises eight proposed 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 7 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 *n*-bit hash value. 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 proposed chaotic map.

V. PERFORMANCE ANALYSIS

A. Distribution of hash value

One of the most important properties of hashing scheme is the uniform distribution of hash value, which is related to the security of hashing scheme. In Fig. 8(a) the ASCII characters of the original message are localized within a small area from approximately 97 to 122. It can also be noticed that the "space" character (ASCII 32) is the most commonly used in the original message. In contrast, the hexadecimal hash values of the hashing scheme are uniformly distributed over the space of all possible hash values, as shown in Fig. 8(b). This indicates that no information of the original message is left after the diffusion and confusion processes.

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 (mod 8) = 0
end if
>> Divide the input message M into N sub-block of length 8 (Defined by S)
>> i, k, t = 1

>>Map M into integer with interval [0 , 1]

 # First Stage (Input Stage)
while (t < N)
  while (i < 7)
  >> $X(i,t)=\lfloor \alpha(M(k)\times X(i+1,t-1))-1 \rfloor$

```
        >> k = k+1
        >> i = i+1
        end
>> X(8,t) = | α(M(k) × X(1,t−1)) − 1 |
>> k = k+1
>> t = t+1
end
>> i = 1

# Second Stage
while (t < N+10)
        while (i < 7)
        >> X(i,t) = | αX(i+1,t−1) − 1 |
        >> i = i+1
        end
>> X(8,t) = | αX(1,t−1) − 1 |
>> t = t+1
end
 >> H = Map X into integer with interval [0 , 2^n]
>> return H
```

Figure 7. Pseudo codes of the proposed hash function algorithm.

B. Sensitivity of hash value to the message and initial conditions

The aim of this subsection is to illustrate the high sensitivity of the proposed hashing scheme to tiny changes in the original message and the initial conditions. In order to investigate this issue, a series of experiments have been done under the following eight different conditions:

Case 1: The original message is: "Sensitivity of hash value to the message and initial conditions.".

Case 2: Replace the first character of the original message by "A". The two 128-bit hash values for C1 and C3 differ in 63 positions.

Case 3: Replace the character "i" in the word "initial" by "e" to become "enitial". The two 128-bit hash values for C1 and C4 differ in 66 positions.

Case 4: Replace the last character of the original message "." by ",".The two 128-bit hash values for C1 and C5 differ in 62 positions.

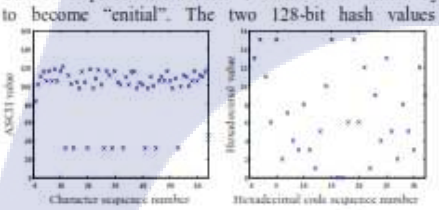


Figure 8. Spread of hash value: (a) distribution of the original message in ASCII; (b) distribution of the hash values in hexadecimal format.

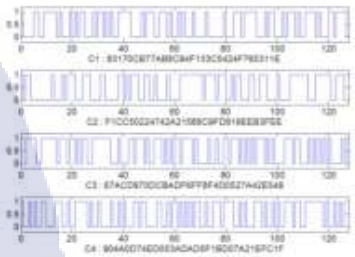


Figure 9. Binary sequences of hash values of eight conditions.

The corresponding graphical display of binary sequences is shown in Fig. 9. The simulation result shows that any tiny change in the original message and initial condition leads to a 50% changing probability for each bit of hash value.

C. Statistical Analysis of Confusion and Diffusion

Confusion and diffusion are two essential design criteria for hashing scheme which are necessary to make it resistant to most attacks. Diffusion is intended to spread the original message statistics through the hash value in order to hide statistical properties of the original message. The aim of confusion is to use the transformation to make the relationship between the input bits and the output bits as complex as possible. The diffusion and confusion test has been performed as follows: a random message of length $L = 50n$ is created and its n-bit hash value is calculated. Then, a bit of the original message is randomly chosen and flipped, and the n-bit hash value of the modified message is calculated. The two hash values are then compared in order to quantify the number of changed bits. This experiment is performed N times for $N = 256, 512, 1024, 2048$ and 10000 for hash values of size n,

TABLE I. SUMMARY OF STATISTICAL MEASURE METHODS.

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

(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 II. STATISTICAL RESULTS FOR A 128-BIT HASH VALUE

	N=256	N=512	N=1024	N=2048	N=10000
B_{min}	46	45	45	44	42
B_{max}	85	84	82	80	84
$\overline{B}$	63	63.04	63.23	62.65	62.9293
P (%)	49.22	49.25	49.4	48.94	49.16352
ΔB	6.19645569	6.627095	5.602838	5.479789	5.673732
ΔP (%)	4.840981007	5.177418	4.377217	4.281085	4.432603

TABLE III. STATISTICAL RESULTS FOR A 160-BIT HASH VALUE

	N=256	N=512	N=1024	N=2048	N=10000
B_{min}	59	58	56	55	53
B_{max}	95	96	97	101	102
$\overline{B}$	78.35	78.89	78.66	78.86	78.9512
P (%)	48.97	49.31	49.16	49.28	49.3445
ΔB	6.117366	6.302809	6.261118	6.649782	6.561541
ΔP (%)	3.823354	3.939256	3.913198	4.156114	4.100963

TABLE IV. STATISTICAL RESULTS FOR A 256-BIT HASH VALUE

	N=256	N=512	N=1024	N=2048	N=10000
B_{min}	103	101	102	102	99
B_{max}	146	154	154	152	158
$\overline{B}$	126.86	127.75	126.74	126.93	126.8779
P (%)	49.55	49.9	49.51	49.58	49.56168
ΔB	8.289252	8.555481	8.516115	7.904845	7.570246
ΔP (%)	3.237989	3.341985	3.326608	3.08783	2.957127

where n = 128, 160 and 256. The six statistical measure methods, as listed in Table 2, are usually used for statistical analysis. B_i denotes the number of changed bits in the i-th test. Tables 3-5 list the results obtained in tests for n = 128, 160 and 256 and N = 256, 512, 1024, 2048 and 10000. Fig. 10 demonstrates the distribution of the number of bits changed for various values of N. The histogram distribution of the number of changed bits for n =256 and N =10000 is depicted in Fig. 11. Based on the results in Tables 2-4, it is evident that the mean changed bit number $\overline{B}$ and the mean changed probability P are very close to the ideal values of $n/2$ and

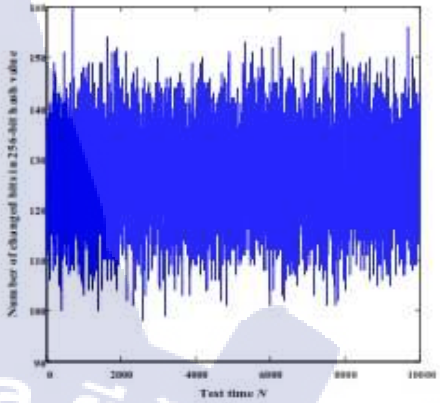


Figure 10. Distribution of the number of bits changed.

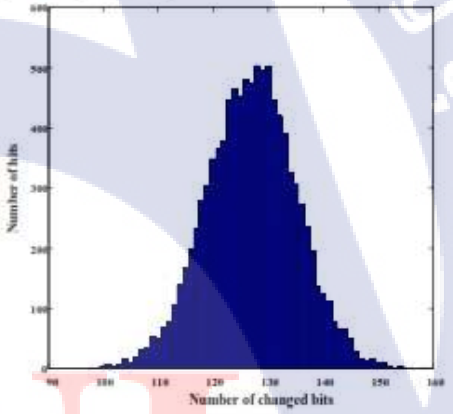


Figure 11. Histogram distribution of the number of changed bits for n =256 and N =10000.

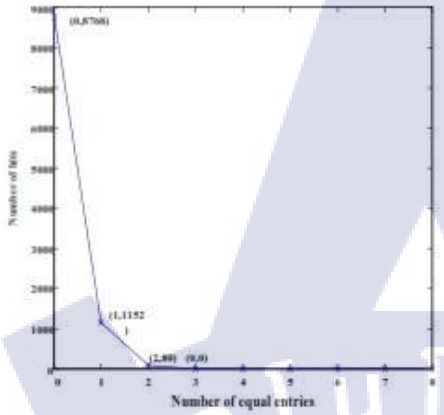


Figure 12. Distribution of the number of positions where the ASCII characters are the same for $n = 256$ and $N = 10000$.

50% respectively. While both ΔB and ΔP are small for all tests, which indicates that confusion and diffusion capability of the proposed hashing scheme is stable. Collision Test

Hash collision is a situation that occurs when two distinct input messages into a hash function produce the same hash values. In order to quantify the collision resistance of the proposed hashing scheme, the following collision test has been performed. The n-bit hash value of a random message of size $L = 50n$ is created and stored in ASCII format. Then a bit in the random message is randomly chosen and toggled. The new hash value is created and stored in ASCII format. The two

TABLE V. ABSOLUTE DIFFERENCE FOR 2 HASH VALUES, WHERE N = 10000.

Absolute difference (d)	Min.	Max.	Mean
MD5(128bit)	590	2074	1304
SHA-1(160bit)	795	2730	1603
Zhang's scheme[17](128bit)	565	2022	1257
Kanso's scheme[18](128bit)	737	2320	1494
Wang's scheme [19] (128bit)	689	2295	1526
Ren's scheme [20] (128bit)	599	2455	1439
Wang's scheme [21] (128bit)	655	2064	1367
Our proposed scheme (128bit)	544	2400	1348
Our proposed scheme (160bit)	809	2782	1687
Our proposed scheme (256bit)	1402	3954	2716

hash values are compared with each other, and the number of ASCII characters with the same value at the same location, referred to as the number of hits, is counted. The absolute difference of the two hash values is computed as follows:

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

where m_i and m_i' denote the i-th ASCII character of the original and new hash value, respectively, and $dec(\ )$ maps m_i and m_i' to their equivalent decimal values.

This kind of test has been performed 10000 times. The minimum, maximum and mean values of d are presented in Table 7 and a plot of the distribution of the number of hits is illustrated in Fig. 12. It can be noticed that the maximum number of equal character is only 2 and the collision probability is very low. It can be observed from Table 6 that for $n = 128$ and $n = 160$, the proposed hashing scheme provides better results than the existing algorithms such as MD5 and SHA-1, and comparable results to other chaos-based algorithms.

D. Resistance to birthday attack

A birthday attack is a kind of cryptographic attack that is based on mathematical behind the birthday problem in probability theory. It gets its name 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 example, if the size of the hash value is set to 256, the difficulty of the attack is 2^{128}. This keeps the system robust against this type of attack.

VI. CONCLUSIONS

The design of communication protocols and message authentication in the Mobile Ad hoc Networks is challenging due to limited wireless transmission ranges of node mobility, limited power resources, and limited physical security. Assuring secure routing protocols is challenging since MANET wireless networks are highly vulnerable to security attacks. Most traditional routing protocols and message designs do not address security, and are mainly based on a mutual trust relationship among nodes. The new compact and robust chaos-based keyed hash function has been presented. The proposed chaotic map exploits absolute-value nonlinearity for generating 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, and signal characteristics in time domains. The proposed hashing structure is relatively simple that enhances randomness quality for statistical performances. The designed hashing algorithms involve the

initial stage when the chaotic 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 changed bit number that is also close to a half of hash value lengths. In addition, the collision tests proffer the average mean of 1359 and 1703 for the hash values of 128 bits and 160 bits, respectively. The proposed has function has superior performance over well-known algorithms such as MD5 and SHA1, and is comparable to other complex structures of chaos-based approaches. As a result, the proposed hash function has offered a potential alternative to protocol and message authentication methods in Ad Hoc Networks.

ACKNOWLEDGEMENTS

The authors are grateful to Research and Development Division of Thai-Nichi Institute of Technology for research fund supports.

REFERENCES

[1] Li Fenga, Zili Lic, Yi Zhangb, "Security bootstrap model of key pre-sharing by polynomial group in mobile Ad Hoc Network", Journal of Network and Computer Applications, Vol 32, p. 781–787, 2009.
[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. Brachtl, D. Coppersmith, M.M. Hyden, S.M. Matyas, C.H. Meyer, J. Oseas, S. Pilpel, M. Schilling, "Data authentication using modification detectioncodes 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] 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.
[7] 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.
[8] 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.
[9] Behnia S, Akhshani A, Akhavan A, Mahmodi H. *"Applications of tripled chaotic maps in cryptography"*, Chaos Solitons & Fractals, 2009, Vol. 40, p. 505–519.
[10] Wong K. *"A combined chaotic cryptographic and hashing scheme"*, Phys Lett A, 2003, Vol. 307, p. 292–298.
[11] 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.