Research Article

ACCENTS Transactions on Information Security, Vol 2(5)
ISSN (Online): 2455-7196
http://dx.doi.org/10.19101/TIS.2017.25002

Identical chaotic synchronization for hash generation

Medini H K[1*], Mahfooz Sheikh[2], Murthy DHR[2], Sathyanarayana SV[1] and GK Patra[2]
Jawaharlal Nehru National College of Engineering, Shivamogga, Karnataka[1]
CSIR Fourth Paradigm Institute, Bangalore, India[2]

Abstract
Chaotic synchronization since its advent in 1990 [1], possess immense potential in cryptography. The identical or complete synchronization is the coupling scheme used in the interacting systems that will emerge with time. Here the Lorenz systems of equations are utilized for synchronization of chaotic systems. Keys can be obtained after the two chaotic systems are completely synchronized, by linking two Lorenz systems that are chaotic in nature. These keys form the basis for generating hash values using the message digest. This paper emphasizes on the formulation of a new hash function based on chaotic systems using Lorenz equations in order to generate collision resistant hash values.

Keywords
Lorenz system, Chaos system, Cryptography, Coupling scheme.

1.Introduction
Although chaotic systems seemed to defy synchronization, past few years have seen research in synchronizing such systems by linking them with common signals. This motivates the use of chaotic systems in cryptanalysis that forms entirely a new domain [2]. A set of Ordinary Differential Equations(ODEs) called Lorenz equations, are used for the synchronization. The sensitivity of these equations to initial conditions with minor changes produce solutions with great disparities, a characteristic of chaos. However, the systems are said to be synchronized if the correlation coefficient and the standard deviation ratio gradually approaches to one [3], which is elaborated in section II. The rapid growth in computer networks and wireless de-vices, pose a greater need for the protection of information. In computer systems and networks upholding the integrity and authenticity of information is a prime requirement. Cryptography provides solution to these requirements to great extents. Hash functions are one way function that form integral part of cryptography to mainly ensure data integrity and authenticity. A secured hash function shall have enhanced security when it complies with certain standard performance parameters [4]. Since chaotic systems are very sensitive to initial conditions [5], these properties make them suitable for hash function design.

*Author for correspondence

However, secure hash algorithms have already been attempted using chaotic synchronization [5-7], the present work emphasizes on generating a key based on linking of two Lorenz systems that are chaotic in nature and generate hash values that show promising results for collision based attacks.

2.Chaotic synchronization using Lorenz equation
A phenomenon of Chaos Synchronization that may occur when two or more dissipative chaotic systems are coupled with each other. Chaotic synchronization can be achieved in varied methods depending on the nature of the interacting systems and the coupling scheme [8]. Lorenz system is a set of ODEs that are chaotic for certain parameter values and initial conditions. In this work, a type of synchronization called Identical Synchronization scheme or complete synchronization scheme is used for linking two identical chaotic systems. For a given set of initial conditions, the systems gradually develop identical to each other over time using identical synchronization scheme and the systems are said to be completely synchronized [9]. The chaos synchronization is achieved through Lorenz equations that are derived from Lorenz systems.

A. Lorenz systems
In this work, the synchronization of two systems is achieved using Lorenz system. The Lorenz system was first de-liberated by Edward Lorenz, a

remarkable achievement where chaotic solutions can be obtained for determined parameter values and initial conditions [10, 11]. The Lorenz system is a system of three ODEs, which are now noted as Lorenz equations, that are given by,

$$\begin{aligned} dx/dt &= \sigma (y - x) \\ dy/dt &= x (\rho - z) - y \\ dz/dt &= xy - \beta z \end{aligned} \quad (1)$$

Where x, y and z are system states, t is the time and σ, ρ, β are system parameters, where σ = 10, ρ= 28 and β = 8/3, for which *Figure 1* gives the structure of the Lorenz attractor.

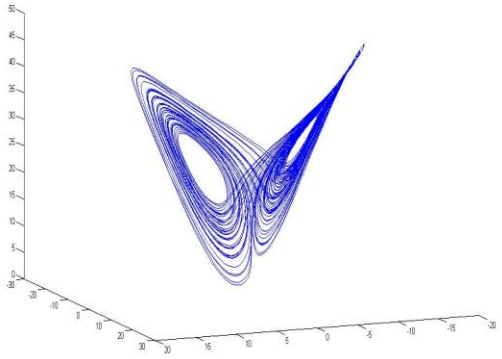

Figure 1 Lorenz attractor

The system is stable for smaller values of ρ and evolves to one or more fixed point attractors. But for larger values the fixed points become repulses and the trajectory is repelled by them in a very complex way. Now the following set of equations are considered at one instance,

$$\begin{aligned} dx_1/dt &= \sigma (y_1 - x_1) \\ dy_1/dt &= x_1 (\rho - z_1) - y_1 \\ dz_1/dt &= x_1 y_1 - \beta z_1 \end{aligned} \quad (2)$$

And the following set of equations at other instance,

$$dx_2/dt = dx_1/dt \quad dy_2/dt = x_2(\rho - z_2) - y_2 \quad (3)$$

$$dz_2/dt = x_2 y_2 - \beta z_2$$

With arbitrarily chosen initial conditions solution to eq(2) is obtained. Later the signal $x_1(t)$ is used to solve eq(3), with arbitrarily chosen initial conditions. The solution is obtained when $y_1(t)$ and $y_2(t)$, $z_1(t)$ and $z_2(t)$ are synchronized.

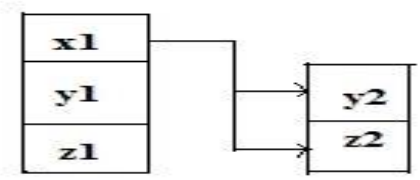

Figure 2 Master-slave schemes

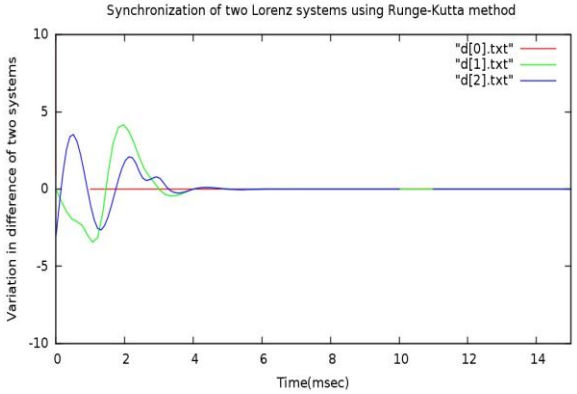

Figure 3 Variation of difference of two systems with time

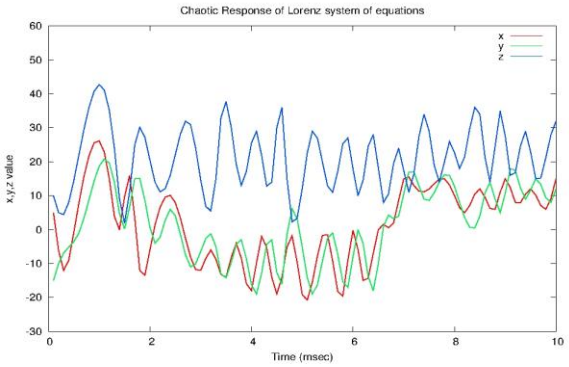

Figure 4 Variation of system states with time

Figure 2 illustrates the master-slave scheme, where the first system is called as master and the second one is the slave. These two systems in eq(2) and eq(3) are synchronized soon in a master slave kind of agreement and obtain solution when we start with eq(2), using Identical Synchronization scheme [12-13].

Figure 3 gives a plot of $d[0]=x_2-x_1$, $d[1]=y_2-y_1$ and $d[2]=z_2-z_1$ with time. It is observed that after some amount of time there is a complete synchronization of two chaotic systems. *Figure 4* gives the chaotic response of system states with time for the Lorenz system of equations.

3.Identical synchronization scheme

Synchronization of chaotic systems is challenging owing to their high sensitivity to initial conditions. As a result, it may not be possible that two chaotic systems staring at nearly the same initial conditions be synchronized. Nevertheless, the systems gradually evolve identical in time with a set of same initial conditions. The present work uses identical synchronization scheme to solve the Lorenz system of equations and obtain matching values for given two system of equations. These synchronized values form the input keys for hash functions.

4.Proposed hash function using chaotic synchronization

Many numerical methods are proposed for solving Lorenz equations. Runge-Kutta fourth order method has been used owing to its considerably good accuracy in obtaining numerical solution of ODEs. Synchronization of chaotic systems generates keys and the algorithm for it is as given below. A message digest is then generated with four different types of iterations. Each iteration considers intermediate hash value of previous iteration, part of the message and the synchronized keys in order to generate the final hash value.

A. Algorithm for chaotic synchronization

The steps to synchronize two chaotic systems is as follows,

1) The chaotic network parameters, σ, ρ, β are initialized.
2) With arbitrary chosen initial conditions, equation (2) is solved.
3) The initial conditions of equation (3) can be arbitrarily chosen.
4) Using the above initial conditions, Runge-Kutta fourth order method is used to solve the Lorenz equations.
5) The solutions are compared to be equal and the steps 2-4 are repeated when identical solutions could not be obtained.

B. Flowchart for chaotic synchronization

The flowchart for the synchronization of two chaotic systems is as shown in *Figure 5*. Here prior to solving eq(2) and eq(3), the network parameters need to be initialized. Later when synchronization is achieved the keys from both the systems are compared. If both the keys agree with each other, then the system is synchronized else entire process needs to be repeated till complete synchronization is attained.

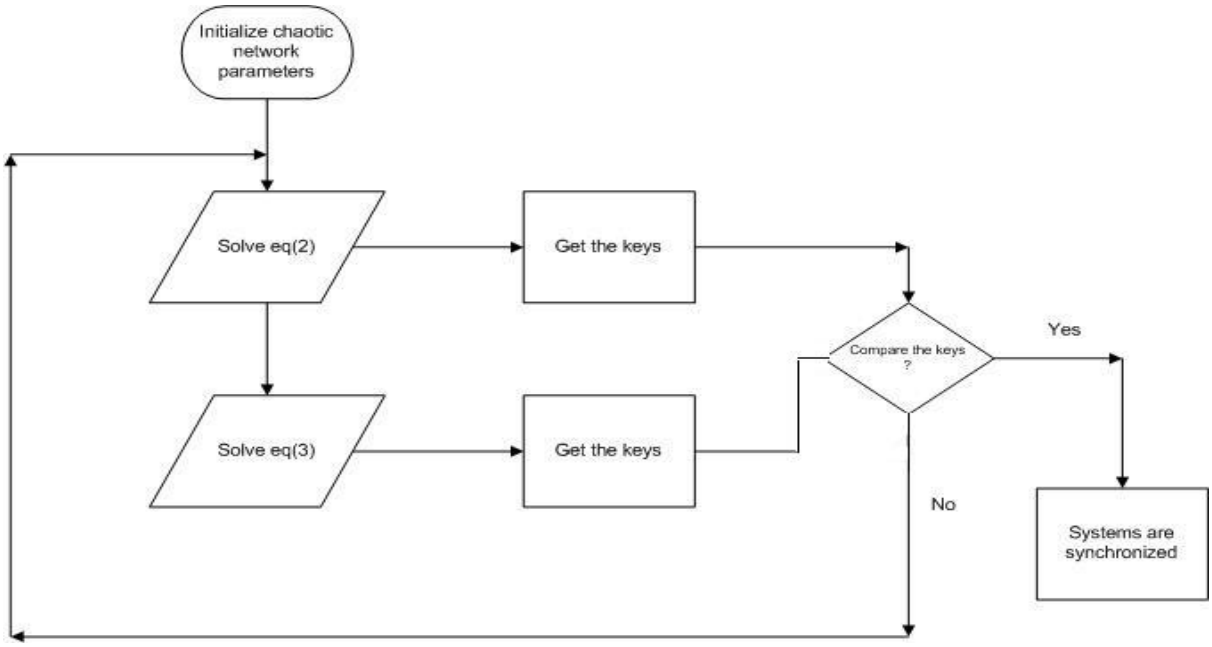

Figure 5 Flowchart for chaotic synchronization

C. Algorithm for hash function using chaotic systems

Algorithm to generate Hash Function using chaotic systems is as given below,

1) Enter a message: Enter some arbitrary length message and convert to binary.
2) Divide that message into four equal parts.
3) Perform padding up to 128 bits for each part.

4) Now four types of iterations should be accomplished,

a. First iteration
A1=first part of message
B1=secret key C1=first part of message
H1=(A1 XOR B1) AND C1 – First intermediate hash

b. Second iteration
A2=second part of message
B2=secret key XOR H1
C2=H1
H2= (A2 AND B2) OR C2-Second intermediate hash

c. Third iteration
A3=Third part of message
B3=secret key XOR H2
C3=H2
H3= (A3 AND B3) XOR C3-Third intermediate hash

d. Fourth iteration
A4=fourth part of message
B4=secret key XOR H3
C4=H3
H4= (A4 OR B4) AND C4-Fourth intermediate hash
5) Left rotate H1
6) Final digest= H1 XOR H4

The pictorial representation of this algorithm is as shown in *Figure 6*.

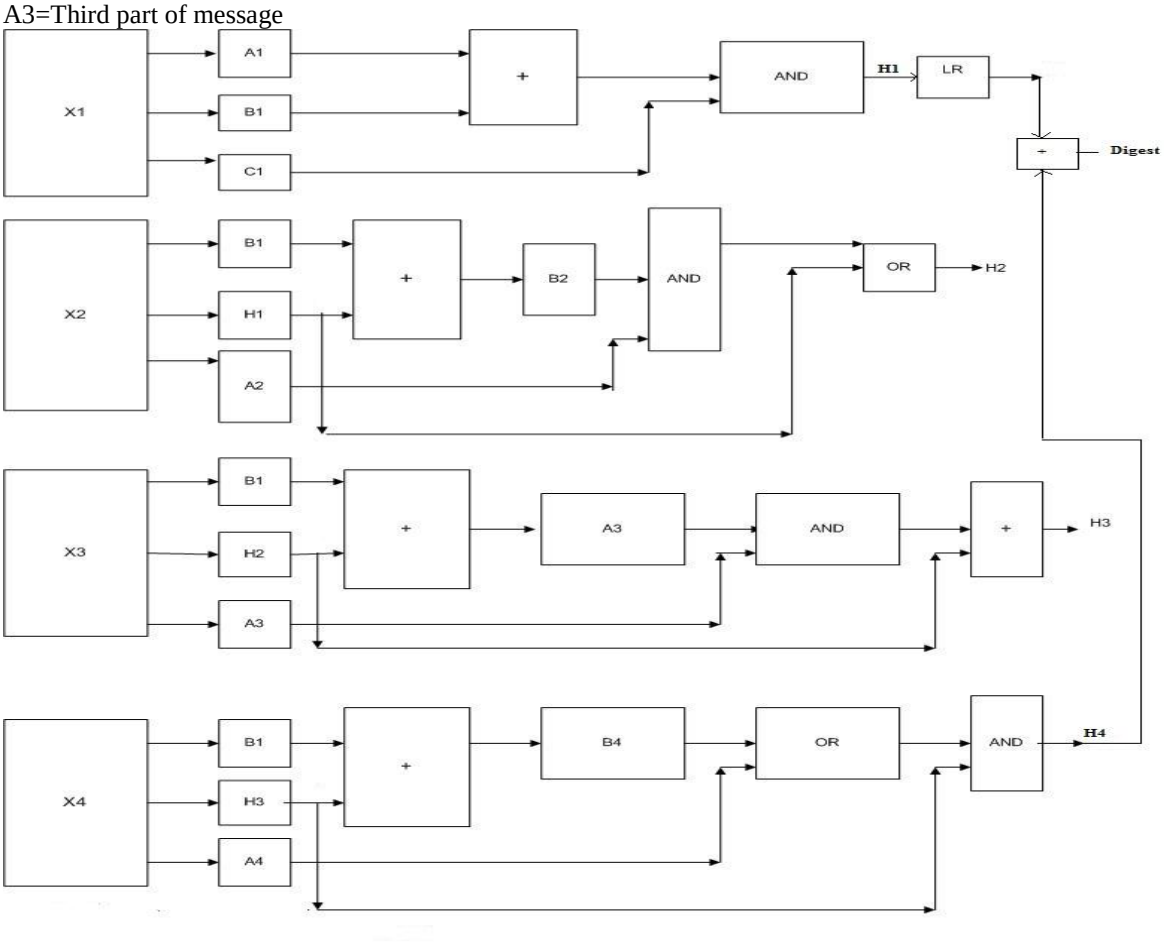

Figure 6 Proposed hash function diagram

5.Analysis of hash generator

A. Attacks on hash functions

This section describes the possible attacks on hash functions. The designed hash function needs to be resistant against the cryptographic attacks, in order to claim itself a secured hash algorithm. In the present analysis, we attempt to establish a collision resistance property for the proposed hash function.

a. Likeliness of hash collision (Collision attack)
In cryptography, a collision attack is the one, which strives to notice two different uneven length inputs generate a same hash value i.e., two divergent

messages m1 and m2 such that h(m1) = h(m2). Since hash functions are immune to their input message length and have predetermined output length. It is extremely unlikely that the two inputs produce the same hash result in the proposed algorithm. For example, Consider a 512 bit message and a 64 bit digest.

1) There are 2^{512} possible messages.
2) There are 2^{64} possible digests.
3) Therefore, there are $2^{512}=2^{128} = 2^{448}$ possible messages per digest.
4) In general, to find the collision, we need to try an average of 232 messages.

This algorithm was executed for nearly 400000 bits with 64 bit digest. The hash collision in this case was found to be 0.001%.

B. Sensitivity of the proposed chaotic hash function

The generated hash function is been assessed in terms of performance and security using standard techniques.

a. Sensitivity of hash value to the message alteration

With respect to the message the sensitivity of a hash value is evaluated. The message is set as follows, M1:The original message is (”Cryptography is all about conveying in the presence of rival. It circumscribes many issues such as, encryption, decryption, data authentication. The main objective is to preserve security across a in secured channel”).

M2:Change the second and third character of the message as e and h respectively of M1.
M3:Delete the fourth character of M1.
M4:Change the last character to s of M1.
M5:Delete the first character of M1.
The hash values are as listed in *Table 1*.

From *Table 1*, it is observed that the proposed hash function hash a good sensitivity to the alterations in message M1.

Table 1 Message and their hash values

Message	Hash values
M1	e0062719ce14b01521b0053cafafc09b
M2	1245c03f76949f64c0aed82197f78290
M3	6100539a0f8f13d2c9bc3b8f7413c282
M4	c12a29755575b5825817dac67b4e88f3
M5	25f8d4cc7d3c48f828a8c5ee9b80f1f2

b. Sensitivity of hash values to the initial conditions of chaotic systems

With respect to the varied initial conditions in chaotic systems, the sensitivity of a hash value is examined. The hash values are as listed in *Table 2*.
It is observed in *Table 2* that the sensitivity of hash values to the changes in initial conditions of chaotic systems is satisfactory.

Table 2 Initial conditions and their hash values

Initial conditions	Hash values
x_1=2.5,y_1=1.8,z_1=0.5	e0062719ce14b01521b0053cafafc09b
x_1=2.5,y_1=1.2,z_1=0.5	ca8f03d8fadc36d791141e38ba974d87
x_1=2.5,y_1=,1.8,z_1=0.8	9ec6bb4aa03f4204c86df79f3284c04c

c. Time complexity

In order to measure the time complexity of the proposed hashfunction, a comparison was made with the standard Secure Hash Algorithm(SHA1). SHA-1 is one of most popular hash functions and was issued by the National Institute of Standards and Technology in 1995 as a FIPS. FIPS stands for Federal information Processing Standard, which are US based and publicly announced standards for use in computer systems by non-military government agencies and government contractors. Different input message sizes were used with SHA-1 and the proposed hash function. The same set of data were used as the input message and the time taken by both hash functions was obtained. It is observed that both the hash functions are well within comparable limits. The evaluation of proposed hash function and the standard SHA1 in terms of execution time is given in *Figure 7*.

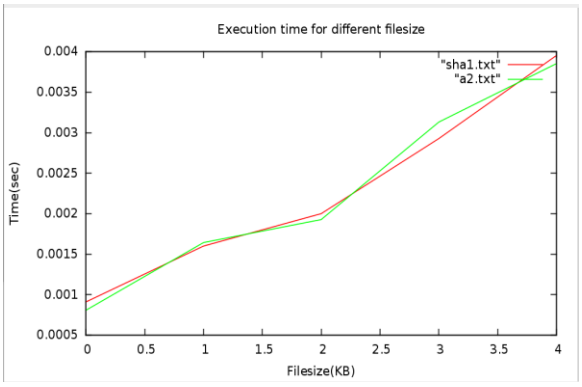

Figure 7 Execution time for different file sizes

6.Conclusion

A chaotic system based hash function is designed and analyzed in the present work. An attempt to utilize the chaotic Lorenz system of equations and their solution using identical synchronization scheme, in

order to generate a key for use in the hash function was accomplished. However, an indigenous approach to develop a hash function based on the generated key, shows promising results and also convincingly satisfies the standard collision resistance and sensitivity tests. Its parametric comparison for time complexity with SHA-1 algorithm has also emerged satisfactory and complying in great respects.

Acknowledgment

This work is supported by student programme for advancement in research knowledge (SPARK) and is developed as a part of the CySeRo component of the ARiEES project at CSIR-4PI, Bangalore. Authors are grateful to the head, CSIR-4PI for providing us the opportunity to carry out this work.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Pecora LM, Carroll TL. Synchronization in chaotic systems. Physical Review Letters. 1990; 64(8):821.

[2] Mahesh M, Patra GK. Synchronization of chaos in lorenz system and its application to cryptography.

[3] Yeh JP, Wu KL. A simple method to synchronize chaotic systems and its application to secure communications. Mathematical and Computer Modelling. 2008; 47(9):894-902.

[4] Meduri SR, Ogirala S, Reddy BT. Design of keyed secure 256 bit chaotic hash function. 2012; 3(3):4421-6.

[5] Mohammed MT, Rohiem AE, El-moghazy A, Ghalwash AZ. Chaotic based secure hash algorithm. 2013; 2(2): 127-33.

[6] Xiao D, Liao X, Deng S. Chaos based hash function. In chaos-based cryptography 2011 (pp. 137-203). Springer Berlin Heidelberg.

[7] Kwok HS, Tang WK. A chaos-based cryptographic hash function for message authentication. International Journal of Bifurcation and Chaos. 2005; 15(12):4043-50.

[8] Lü J, Zhou T, Zhang S. Chaos synchronization between linearly coupled chaotic systems. Chaos, Solitons & Fractals. 2002; 14(4):529-41.

[9] Patra GK, Ramamohan TR, Kumar VA, Thangavelu RP. Improve in security level of first generation chaotic communication system by mutual synchronization. In international conference on advanced computing and communications 2006 (pp. 195-8). IEEE.

[10] Beheshti S, Khaloozadeh H. Synchronization of chaotic systems with unknown time delay by sliding mode observer approach and unknown delay identification. In Iranian conference on electrical engineering 2013 (pp. 1-6). IEEE.

[11] Stork M. Digital chaotic systems examples and application for data transmission. Electrical and Electronics Engineering. 2009.

[12] Hamamci SE, Göğebakan V, Işık İ. A new chaotic system with chaos entanglement. In signal processing and communications applications conference 2015 (pp. 2597-600). IEEE.

[13] Q Wu. A chaos-based hash function. International conference on cyber-enabled distributed computing and knowledge discovery 2015 (pp. 1-4).

This paper is selected from proceedings of National Workshop on Cryptology-NWC 2016 organized at JNN College of Engineering Shimoga, Karnataka, India during 11-13, August 2016.