ISSN:0975-9646

Sambasiva Rao Meduri et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (3) , 2012, 4421 - 4426

Design of Keyed Secure 256Bit Chaotic Hash Function

Sambasiva Rao Meduri ,Srinivasareddy Ogirala, B.Tirapathi Reddy

Department of Computer science & Engineering, Lakireddy Bali Reddy College of Engineering (Autonomous), Mylavaram, Krishna(dist)-521230,India.

Abstract-The main contribution of the paper containing two steps, one is building stepby step a chaotic hash function starting from a weak but basic algorithmand second is analyzing its strengths and weaknesses in order to make it stronger.We start with a basic chaotic hash function with a 256-bit message digestbased on Baptista's encryption algorithm. In the next steps, a pseudorandomnumber generator using chaotic tent map is incorporated within the hash algorithm and perturbation and block chaining approaches areused to strengthen the hash function. We have carried out both the preimage and second-preimage resistance analysis and proved that the proposed hash function is strong against both these attacks. Further, the standard collisionanalysis is performed in the space of 1-bit neighborhood of a given message.The hash function is shown to exhibit diffusion effect with averagehamming distance between message digests obtained as 127 bits which isclose to the ideal of 50%. The collision performance compares favorably with that of chaotic hash functions proposed in the recent literature. Itis to be emphasized that the existing chaotic hash functions in the literatureuse a multitude of chaotic maps whereas we show in this paperthat using two chaotic maps judiciously achieves a secure hash function.

Keywords -Chaotic Hash function, Pseudo Random Number, Perturbation, Hamming distance, Message digest

1.INTRODUCTION

A secure hash function is a function that takes a variable-length input string andconverts it to a fixed-length (smaller) output string called hash value or messagedigest. h : $(0, 1)^* \rightarrow (0, 1)^n$ for some integer n is such that h satisfies the three security properties:collision resistance, preimage and second preimage resistance [1], and having certain properties to make it appropriate for use as a primitive in many cryptographic and Web applicationssuch as message integrity, digital signatures and authentication.

Recent investigations reveal that several well-known methods such as MD5,SHA1 and RIPEMD too are not immune to collisions [2, 3]. Chaotic maps provideanother potential avenue to look for secure encryptions [4–6]. The crucialproperty of sensitivity to initial conditions for a chaotic function proves to bevery useful in this context [7]. A function *f* is said to be sensitive at a point *x*,if the trajectories of the dynamical system defined by *f* change drastically forpoints *y* that are initially very close to *x*. Hence sensitivity seems to be a tailormadefeature that satisfies the collision resistance requirement while constructinga hash function.During the last three decades chaotic dynamics played a major role in thefield of nonlinear sciences. The Important characteristics like the randomness ofdynamical behavior, sensitivity to initial conditions and possession of positiveLyapunov exponents, that the chaotic maps possess make these prime candidatesfor many cryptographic applications.The novelty in this paper is that we propose the method in a systematic fashion. We consider a basic chaotic hash algorithm based on Baptista's encryption scheme and start strengthening it conducting analysis for security along the way. The hash function is proposed taking advantage of the strengths of Baptista's scheme and it is shown by computational analysis that the performance of the hash function is on par with recent chaotic hash function proposed in the literature without requiring a network of chaotic maps as in [8] thus improvingthe time complexity of the algorithm. The paper is organized as follows: Section 2presents the properties of keyed hash functions. Section 3 is devoted to the chaotic map used for incorporating randomness in hash algorithm and defining the suggested hash function. Section 4 presents the hashing algorithm. In Section 5, the performance of the suggested keyed hash function is analyzed. Finally, we end with some concluding remarks.

2. KEYED HASHFUNCTIONS

The seminal paper of Baptista [9] on chaotic cryptography inspires us to proposea one-way hash function based on chaotic maps. A thorough analysis of Baptista'sscheme was carried out by Alvarez et al [10] and they show that Baptista'salgorithm is vulnerable to all the four of cipher text only, known plain text, chosenplain text and chosen cipher text attacks. On the other hand, it was shown inthe literature that Baptista's scheme has a lot of potential and could be modifiedto build a hash function. K.W.Wong modified Baptista's algorithm by adoptinga dynamic look-up table to avoid collisions and preimage attack [11] and thencame up with a hashing scheme in 2003 [12]. X.Yi [13] in 2005 proposed a hashfunction based on chaotic tent maps which is claimed to be better than Wong'sscheme in its computational complexity. More recently H.Yang et al [14] havepublished another hash function based on a chaotic map network and Q.Yanget al [15] have published a hash function based on cell neural network.These approaches use a multitude of chaotic maps [16–19] and we show inthis paper that using two chaotic maps judiciously achieves a secure hash function.

A Secure Keyed One-Way HashFunction $\{h_k : k \in K\}$ satisfies the following properties:

1) the function $h_k$is keyed one-way. That is,

a) Given k and m, it is easy to compute $h_k(m)$.
b) Without knowledge of k, it is hard to find m when$h_k(m)$ is given and to find $h_k(m)$ when only mis given.
2) The function $h_k$is keyed collision free, that is, withoutthe knowledge of k it is difficult to find two distinctmessages m and m′ s.t.$h_k(m) = h_k(m')$.
3) Images of function $h_k$has to be uniformly distributedin order to counter statistical attacks.
4) Length l of produced image has to be larger than 128bits in order to counter birthday attacks.
5) Key space size has to be sufficiently large in order tocounter exhaustive key search.

3.CHAOTIC MAP

The tent map is considered among the good Chaotic Maps exhibiting chaotic behavior.It is used for the generation of random-like real numbers uniformly distributed in [0, 1].
The tent map is defined by the following equation:

$$x_{n+1} = \begin{cases} ux_n & x_n < \frac{1}{2} \\ u(1-x_n) & x_n \geq \frac{1}{2} \end{cases} \qquad \text{-----------------------------(1)}$$

Where the pair (x0, u) forms the initial condition and control parameter of Eq. (1). A sequence of reals in [0, 1], known as theorbit of Eq. (1), is generated for a given pair (x0,u). For its simplicity, the tent map has found applications in differentareas including cryptography and communications. Thetent map possesses the following properties:
(1) it is a noninvertibletransformation of unit interval onto itself,
(2) it is chaotic for u= 2 (0, 1),
(3) it is ergodic, and the invariant measure of Eq. (1) is uniform on [0, 1]. Furthermore, for a close to 0.5, Eq. (1) generates real numbers equally likely distributed in thecomplete range between 0 and 1. Moreover, the binary digits obtained from these real numbers according to a thresholdfunction are random-like.

In addition to the mentioned properties, some other properties possessed by the tent map, which can be consideredas advantages of Eq. (1) over existing maps, stand behind the motivation of considering the tent map in generation of randomlike binary sequence. These properties include:
1. It has a large key space.
2. It is simple to implement.
3. It can be executed faster than some other existing maps.
4. It has no periodic windows in the chaotic region.
5. It is capable of generating real numbers uniformly distributed in [0, 1].
6. Binary sequences, obtained from the transformation of the orbit of Eq. (1), pass all the statistical tests included in the NISTtest suite [20].For further readings on the dynamics of the tent map as defined in Eq. (1) the reader is referred to [21,22].Based on the excellent random statistic characteristics possessed by the orbits generated by Eq. (1) and their binary transformations.

3.1Tests for randomness of binary sequence generated by chaotic tent map

To test the randomness of sequence of bits generated by algorithm based on tent map,weconsidered theTest suite for pseudo random bits which is standard NIST[20](National Institute of Standards and Technology) testsuite.The results are represented via a table with prows and q columns.The number of rows,p,corresponds to the number of statistical tests applied.The number of columns, qare distributed as follows:The firstcolumn one is the corresponding statistical test,column 2 is the P-value that arises via the application of achi-square test,column 3 is the proportion of binary sequences thatpassed.From the results of test suite if the computed P-value is less than 0.01then that sequence isnon-random.Otherwise,concludethatthesequenceisrandom. We are considered there are five test cases by varying the input bit stream length starting from 100 bits to one million bits and analyzed theuniformityofp-valuesandtheproportionof passingsequences, by changing the input bit stream length there are five test cases are considered starting from bit stream length 100 as test case1,1000 as test case2,10000 as test cas3,100000 as test case4, 1000000 as test case5.

Statisticaltest	P-value	Pass rate
Frequency	0.009535	0.9500
Block Frequency	0.009535	0.9500
CumulativeSums(forward)	0.554420	1.0000
CumulativeSums(Reverse)	0.055361	1.0000
Runs	0.616305	1.0000
Longest Run	0.000000	*1.0000
overlappingtemplate	0.000000	*1.0000
Universal	0.000000	*1.0000
approximateentropy	0.000000	*1.0000

Table3.1:Resultsfor test case1

Statisticaltest	P-value	Pass rate
Frequency	0.090936	0.9900
Block Frequency	0.042808	0.9900
CumulativeSums(forward)	0.000233	0.9900
CumulativeSums(Reverse)	0.514124	0.9900
Runs	0.616305	1.0000
Longest Run	0.350485	0.9900
overlappingtemplate	0.000000	*1.0000
Universal	0.000000	*1.0000
approximateentropy	0.000000	*1.0000

Table3.2:Resultsfor test case2

Statisticaltest	P-value	Pass rate
Frequency	0.437274	1.0000
Block Frequency	0.455937	0.9800
CumulativeSums(forward)	0.946308	1.0000
CumulativeSums(Reverse)	0.514124	1.0000
Runs	0.595549	0.9700
Longest Run	0.474986	0.9900
overlappingtemplate	0.366918	0.9700
Universal	0.000000	*1.0000
approximateentropy	0.000000	*0.7800

Table3.3:Resultsfor test case3

Statisticaltest	P-value	Pass rate
Frequency	0.616305	0.9900
Block Frequency	0.102526	1.0000
CumulativeSums(forward)	0.474986	0.9900
CumulativeSums(Reverse)	0.851383	0.9900
Runs	0.935716	1.0000
Longest Run	0.514124	1.0000
overlappingtemplate	0.383827	1.0000
Universal	0.000000	*1.0000
approximateentropy	0.554420	0.9900

Table3.4:Resultsfor test case4

Statisticaltest	P-value	Pass rate
Frequency	0.739918	1.0000
Block Frequency	0.637119	0.9800
CumulativeSums(forward)	0.834308	1.0000
CumulativeSums(Reverse)	0.085587	1.0000
Runs	0.834308	0.9600
Longest Run	0.851383	0.9800
overlappingtemplate	0.366918	0.9900
Universal	0.657933	0.9900
approximateentropy	0.851383	1.0000

Table3.5:Resultsfor test case5

The minimum pass rate for each statistical test with the exception of the random excursion(variant)test is approximately=0.960150 for a sample size=100 binary sequences.It can be seen that all the five test cases pass 8tests as they achieve P value greaterthan0.01.The first three test cases do not pass the More's universal statistical test and approximate entropy test. One of the reasons could be the length of the input.It can be seen that the Test case 5 passes More's universal statistical test and its input bit length is greater than 10^5 and the cases 4and5 pass the Entropy test with their input bit length being greater than10^4.We proposed aPRBG based on the family often t maps and evaluate its strength using NIST test suite. One of the main aims of this work is to use the PRBG in designing a secure hashfunction.

4.ALGORITHM FOR CRYPTOGRAPHIC KEYED HASH FUNCTION:

In this section, we consider a keyed hash function based on Baptista encryption algorithm[9]. In the algorithm the encryption is performed based on the dynamics of the logistic map.The Equation for logistic map as follows.

$$X_{n+1}=rx_n(1-x_n) \text{----------------------} (2)$$

where the pair (x0,r) form the initial condition and control parameter of Eq. (2). Baptista divides the input domain into $\in$ -intervals I_a where I_a is associated with the character a. The encryption algorithm maps each character a of the message to the number of iterations n that the logistic map takes to reach I_a. We modify Baptista's encryption algorithm to build a hashing function which we call the basic algorithm in the paper. Experiments showed that the basic algorithm is not secure against collisions. Two variations of the basic algorithm are proposed to strengthen the security of the hash function. The block diagram of the above algorithm is depicted in the Figure 1. The secret key of the suggested keyed hash function consists of the initial condition and control parameter of logistic map.

We have used the pseudorandom number generator which is developed by using tent map in development of hash function. In this while taking out put iteration number from the of the logistic map as encrypted value of a input byte value will be tested by the PRBG generator if the generator value is grater then the threshold value then the iteration will be considered otherwise it will be discarded. So the the confusion will be created. and same time the initial seed value of logistic map will give a output function value which is between zero and one and this value will new input for next iteration of the map will create diffusion in the system.

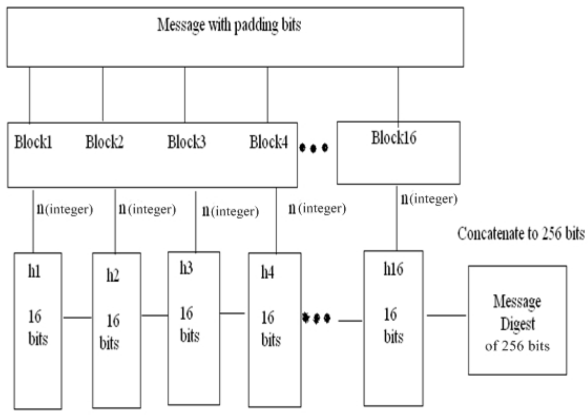

Figure1: Block diagram of hash function

The binary message, M, is processed in blocks of length n, where n is a multiple of 8, such as 128, 160, 256, 512, 1024 and 2048. The procedure for producing an n-bit hash value is given in the main message hashing algorithm (function hash) presented next.

Algorithm: Hash function

1. The bit sequence is divided into 16 blocks B_0,B_1,………....$B_{15}$and each block is an integer number of bytes.
2. Choose a secret value x_0, the control parameter of the logistic equation and set a threshold 0<n<1.
3. Repeat the steps 3-9 for each block B_i, i = 0, 1 ...15.
4. Repeat the steps 4-8 for each byte m=0, 1, 2...k of the message of a block B
5. Compute the ASCII value of m, say A (m), let d (m) =0.A (m)
6. Compute the initial value x' as follows:
   x' (m) = (d (m) +x' (m-1)) mod1 where x' (0) = x_0
7. Iterate the logistic map up to n times where f^n(x`) reaches the 0-interval associated with the character A(m).
8. Generate the random number t using the$PRBG_{tent}$. If t<η, repeat Step 6 to find the next n.
9. Reset x (m+1) = f^n(x (m)).

10. Define for the block B_i,h_i=n. Thus a block gets encrypted by 16 bits.
11. The final hash value of the plain text P is obtained by concatenatingh_i's. i.e. $h(P)=(h_0,h_1,h_2\ldots\ldots h_{15})$.

5.PERFORMANCE ANALYSIS

5.1 Hash sensitivity to original message

In this section, we conduct several tests to determine the performance of our proposed chaotic hash function. We also provide a comparison with some existing hash functions, especially those based on chaotic maps.It is argued earlier that the proposed hash function satisfies preimage and second preimage resistance properties. In this section the function is empirically tested for collision resistance following similar work proposed in the literature.

We have considered five different massages and their hash values are generated and by flipping one bit in the input bit sequence, the corresponding hash value is computed and their changed bit number is calculated.

input	hash	changed hash after bit toggling	Changed bit number
P1	5f0124f8c93e324f8c93e324f8c93e38e011601160116011601 1	df012400c90032018c01e300f8813e00e0016001600160016001	83
P2	24f8c93e324f8c93e324f8c93e324f8c93e324f8c93e324f8c9 3	2400c93e324f8c93e324f8c93e32400c90630078c03e004f8013	103
P3	2a811601160116011601160116018e01160116011601160116 01	6a011601160116011601160116018e011001100110011001 1001	100
P4	0a8124f8c93e324f8c93e324f8c93e324f8c93e324f8c93e324 f	ea0124f8c93e324f8c93e324f8c930324f8c90032018c01e300f	54
P5	56818e0116011601160116011601 8e0116011601160116011601	96818e01160116011601160116018801000100010001000 10001	100

Table 6. MD generated for the input messages P1, P2, P3, P4, P5 obtained could be quantified by looking at the minimum hamming distance and maximum hamming distance.

P1= "welcome to lbrce today we are going to submit our project submission this is about designing secure hash function"

P2="hash function is a special kind of one way function which can be classified into two categories known as keyed hash function"

P3=" Many CHAOTIC hash functions have been proposed in the literature One of the latest is by WONG which we take it as basis"

P4=" Keyed hash function specifies a single input parameter called MESSAGE and a keyed hash function which involves two INPUTS"

P5=" Earlier WONG had proposed a hash function based on the dynamic table algorithm which is a modification of BAPTISTA algorithm"

The ideal diffusion effect should be that any minor change in the plain text leads to a 50% changing probability in the bit sequence of the Message digest.

5.2 Statistical Analysis

A hash function is said to be resistant to collisions if it is computationally infeasible to find two inputs x, x_0 that will get hashed to the same output i.e., $h(x) = h(x_0)$. Collision analysis can be done in two ways.

5.2.1 Looking for a Collision in the Whole Space

For a Large Number of Different Randomly Generated Messages N input messages say P1, P2, P3, · · · PN of length 800 bits are chosen randomly to test for collisions. Hash values are computed for Pi, i = 1. . . 10,000 here and it is observed that no two hash values coincide and completely different hash values are obtained for all the 10,000 input messages. Hence the strong chaotic algorithm is found to have zero collisions in this experiment. Of course this experiment has to be done for 722/1 messages to really show that the hash function is not prone to birthday attack.

5.2.2 Looking for Collision in 1-Bit Neighborhood of Message- Statistical analysis

In order to measure confusion and diffusion six measures are proposed in the literature. If a plain text is changed randomly in one bit and this experiment is carried out for a large N number of times, the difference in the hash value then obtained could be quantified by looking at the minimum hamming distance, maximum hamming distance etc. The ideal diffusion effect should be that any minor change in the initial condition, control parameter or plain text leads to a 50% changing probability in the bit sequence of the Message digest. We tabulate the performance of the hash function in terms of these four Hamming measures.

1. Minimum changed bit number: B_{min} = min $(B_1,B_2,\ldots\ldots,B_n)$.
2. Maximum changed bit number: B_{max} = max $(B_1,B_2,\ldots\ldots,B_n)$.
3. Mean changed bit number: $\bar{B}=1/n\sum_{1=1}^{k}$ Bi .
4.Mean changed probability: P=$\left(\frac{\bar{B}}{256}\right)*100\%$.
5. Standard deviation of the changed bit number:$\Delta B=\sqrt{\frac{1}{N-1}\sum_{i=1}^{n}(Bi-\bar{B})^2}$.
6. Standard deviation: $\Delta P=\sqrt{\frac{1}{N-1}\sum_{i=1}^{n}(\frac{Bi}{256}-P)^2*100\%}$.

	N=256	N=512	N=1024	N=2048	N=5000
B_{min}	73	73	73	73	73
B_{max}	150	150	151	157	162
$\bar{B}$	122.011719	122.093750	124.110352	125.377441	129.608002
P(%)	47.660828	47.692871	48.480606	48.975563	50.628124
ΔB	10.514037	10.250820	10.264173	10.002853	9.595396
ΔP %	4.775420	4.773946	4.850426	4.898765	5.063192

Table 7. Hamming distance measures to evaluate collision resistance.

From the table it is shown that that the average hamming distance 129.6 approximately to 128 for 5000 messages shows that it is ideal(50%) diffusion effect. So the proposed algorithm strong against attacks.

The distribution of changed bit number against number of runs has shown in the following figure, initially till 100 to 1000 runs the changed bit number value is raising from 100 to 120 and then it stabilizes around 128 shows ideal diffusion effect.

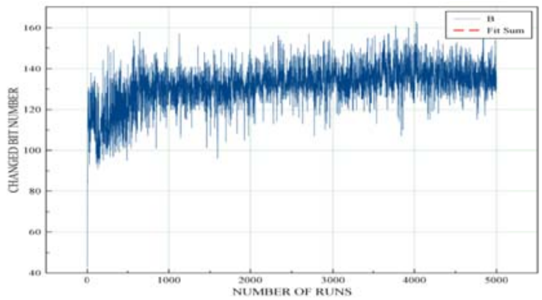

Figure 2: Distribution of the changed bit number

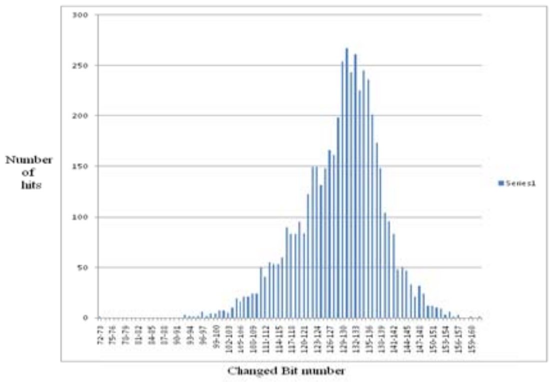

Figure 3: Distribution of changed bit number Bi

Data Set: Input message of size 720KB is taken for generating hash value sayMD which is a 256 bit stream. A bit i is randomly chosen and toggled in themessage. Let the hash value of the perturbed input be MDi. The performanceof the hash function is evaluated in terms of the four hamming measures defined above.

Results: The experiments are carried out for N = 5000 toggled messages forStrong algorithm (which use $PRNG_{tent}$). The messages obtained by changing one bit in the original message exhibit hash values that have, on average, nearly 47% of the bits different from the original Message Digest. Further note that in each experiment, the average number of bits changed in the MD gets doubled for the strong algorithm. These results are improved considerably by using the chaotic tent map to generate pseudo-random numbers of VotePRNG for strong chaotic hash function. The Figure 4 shows that the proposed strong chaotic algorithm which uses $PRNG_{Tent}$ exhibits desirable security with the number of changed bits due to a 1-bit toggle in plain text being 129 which is very close to the ideal value of 50% probability.

5.3 Collision analysis

As it is not easy to provide a mathematical proof on the collision resistance of the proposed chaotic hash function, two kinds of experiments are carried out. In the first, the hash value for a paragraph of arbitrarily- chosen message is generated and stored in ASCII format. Then a bit in the message is selected randomly and toggled. A new hash value is then generated. The two hash values are compared and the random number of ASCII characters with the same value at the same location is counted using the formula as suggested. In a second experiment, two thousand randomly chosen texts are generated and their hash values calculated and assessed for collisions. The results of both the experiments are given in the tables respectively.

$$W = \sum_{i=1}^{N} (\mathrm{f}(\mathrm{t}(\mathrm{ei}).\mathrm{t}(\mathrm{ei}')))$$

The absolute difference between the two hash values is calculated by the following formula:

$$d = \sum_{i=1}^{N} |(\mathrm{t}(\mathrm{ei}) - \mathrm{t}(\mathrm{ei}')|$$

Where F(x, y) =1 if x=y
=0 if x≠y

ei and ei' are the i[th] ASCII character of the original and the new hash value, respectively and the function t(.) converts the entries to their equivalent decimal values. This kind of collision test is performed 10,000 times. The maximum, mean, and minimum values of d are listed in the table. The experimental values of $W_n(2)=3$ and $W_n(w)=0$ for w=3,4,5,... to 16.The distribution of the number of ASCII characters with the same value at the same location in the hash value is shown.

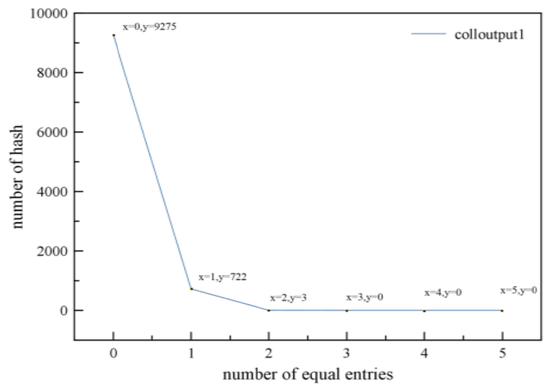

Figure 4: Distribution of the number of ASCII characters with the value at the same location in the hash value

Absolute difference (d)	Maximum	Minimum	Mean
Proposed scheme	2418	1178	1787.984619

Table 8: Collision resistance analysis for strong chaotic algorithm using $PRNG_{Tent}$

Resistance to birthday attack

Our suggested hash function is robust against birthday attack. In fact, the algorithm is flexible so that the length of the hash value can be tuned. For instance, if the hash value size is set to 256, the difficulty of the attack is 2^{128}. A greater size of the generated hash value, makes this kind of attack almost impossible.

In Table 8, we present the minimum, maximum and mean values of the absolute difference of original and new hash values. From this table one can easily observe that for n = 128 and n = 160 our hashing algorithm gives better results than existing algorithms such as MD5 and SHA-1 and those based on chaotic maps such as [23,24,25].

The space in which the tests are conducted is large enough to indicate thatthe values obtained by B_{avg}etc lie close to the true values of the distribution.It is important to note that the proposed strong chaotic algorithm makesuse of only iteration of two maps, the logistic map and the chaotic tent mapand achieves 127-bit diffusion where as the scheme proposed by Yang et al. whichuses a 16 chaotic map network which is only improved by 1.03 bit confusionand significantly increases the chaotic complexity.

The Table 7 shows the existing algorithm meets all the requirements for 256 bit hash function. and it is comparable to present all chaotic based hash functions and here we have used only two chaotic maps.

6 Conclusions

A new one-way chaotic hash function is developed based on Baptista's encryptionalgorithm that gives an output of 256 bit message digest. The algorithm can beadapted to evolve 512 bit length message digest.The secure hash functions that follow Merkel-Damgard construction schemes generally have a multitude of functions networked together to make the compressionfunction secure. In general, there are no logical arguments providedfor the proposed design. In this paper, starting with a nice encryption schemethat is proposed by Baptista which is used to design a hash function, it is arguedlogically why certain plug-ins are required and how these schemes help in making the function secure. It is not possible to prove collision resistance theoretically,hence following similar work in literature we present computationalresults to show that the proposed chaotic hash function exhibits 50% mixing ofbits in the output of message digest on a one-bit change in the input messageand hence is collision resistant. The function is further analyzed and shown topossess preimage and second preimage resistance.It is shown that the performance of the hash function is comparable withsome of the latest algorithms proposed in the literature. By using the proposed hash function will design digital signature in future.

References

1. Stinson, D.R.: Cryptography: Theory and Practice. CRC Press (1995).
2. Wang, S., Yu, H.: How to break MD5 and other hash functions. Proc. EUROCPYPT2005 Advances in Cryptology, LNCS 3494 (2005) 19–35.
3. Wang,X., Yin, Y., L., Yu,H : Finding collisions in the full SHA-1. Proc. CRYPTO2005 Advances in Cryptology, LNCS, 3621, (2005) 17–36.
4. Jakimoski, G., Kocarev, L. : Chaos and cryptography: Block encryption ciphersbased on chaotic maps. IEEE Trans. on Circuits and Systems I: FundamentalTheory and Applications. 48:2 (2001) 163–169.
5. Jakimoski, G., Kocarev, L. : Analysis of some recently proposed chaos-based encryption algorithms. Phys. Lett. A 291 (2001) 381–384.
6. Kocarev, L.: Chaos-based cryptography: a brief overview, IEEE Circuits and SystemsMagazine, 1:3 (2001) 6–21.
7. Devaney,R., L.: An introduction to chaotic dynamical systems. Addison-Wesley(1989).
8. H., Yang, K., W., Wong, X., Liao, Y., Wang, D., Yang.: One-way hash function construction based on chaotic map network, Chaos, Solitons& Fractals, 41:5,(2009) 2566–2574.
9. Baptista, M., S.: Chaos in cryptography. Phys. Lett. A 240 (1998) 50–54.
10. Alvarez, G., Montoya, F., Romera, M., Pastor, G.: Cryptanalysis of an ergodic chaotic cipher. Physics Letters A 311 (2003) 172–179.
11. K., W., Wong.: Modified Baptista type chaotic cryptosystem , Phys. Lett. A 298,(2002) 238–242.
12. K., Wong.: A combined chaotic cryptographic and hashing scheme. PhysLettA307, (2003) 292–298.
13. X., Yi.: Hash function based on chaotic tent maps, IEEE Transactions on Circuits and Systems-II: Express Briefs, 52:6 (2005) 354–357.
15. Q., Yang, T., Gao, L., Fan, Q., Gu.: Analysis of one-way alterable length hash function based on cell neural network, Journal of Information Assurance and Security5 (2010) 196–200.
16. Y.,Wang, X., Liao, D., Xiao, K.,Wong.: One-way hash function construction basedon 2D coupled map lattices. Inform Sci (2008) 1391–406.
17. K., W., Wong.: A fast chaotic cryptographic scheme with dynamic look-up table,Phys. Lett. A 298, (2002) 238–242.
18. D., Xiao , X., Liao, S., Deng.: One-way hash function construction based on thechaotic map with changeable-parameter. Chaos, Solitons& Fractals, 24, (2005)65–71.
19. Zhang, J, Wang, X, Zhang, W.: Chaotic keyed hash function based on feedforwardfeedback nonlinear digital filter. PhysLett A (2007) 439–48.
20. NIST Special Publication 800-22 rev1, A statistical test suite for the validation of random number generators and pseudo random number generatorsForcryptographic applications, online document, <http://www.csrc.nist. gov/groups/ST/toolkit/rng/documentation_software.html>, 2008.
21. M. Hasler, Y. Maistrenko, An introduction to the synchronization of chaotic systems: coupled skew tent maps, IEEE Transactions on Circuits andSystems-I: Fundamental Theory and Applications 44 (10) (1997) 856–866.
22. S. Li, Analyses and new designs of digital chaotic ciphers, Ph.D. Thesis, School of Electronics and Information Engineering, Xian Jiaotong University,Xian, China, <http://www.hooklee.com/pub.html>, 2003.
23. Y. Wang, X. Liao, D. Xiao, K. Wong, One-way hash function construction based on 2D coupled map lattices, Information Sciences 178 (2008) 1391–1406.
24. D. Xiao, X. Liao, S. Deng, One-way hash function construction based on the chaotic map with changeable-parameter, Chaos, Solitons and Fractals 24(2005) 65–71.
25. J. Zhang, X. Wang, W. Zhang, Chaotic keyed hash function based on feedforward–feedback nonlinear digital filter, Physics Letters A 362 (2007) 439–448.