Performance Measurement of Biometric Image hash Using Neural Networks

1saheebpeer.pinitod , 2Dr.Vidya Kulkarni
Gogte Institute of Technology Udyambhag Belgaum

Abstract

Data authentication the Hash functions are used to generate Hash code by using different algorithms like MD4,MD5 and others in our paper we propose a new method of generating Hash Code for Biometric images using Neural network 50 sample images have been considered and performance of hash code is calculated by using two performance matrix like Collision resistance and Plaintext sensitivity. The Hash values are calculated using feed forward neural network and same network with feed back . Original images are subjected to one bit modification and hash values are compared the Plaintext sensitivity and collision resistance are nearly same as MD5 algorithm.

Keywords

Biometric Hash,Collision Resistance and Plaintext sensivity.

General Terms

Neural Network Hash Function, MD5 algorithm.

1. INTRODUCTION
A hash function H is a transformation that accepts variable-length input message M and returns a fixed-length hash string H, which is called the hash value. A Hash function f(M) generates a unique hash value H for a particular Biometric image and thus can be used for checking data integrity and authentication purposes. When Biometric image messages are considered preimage resistance and hit collision are parameters for evaluating the performance of hash functions. Using MD5 and RIPEMD160 collision resistance and plaintext sensitivity execution time is calculated . The applications of neural networks in areas of cryptography in general are discussed in [1-8]. The authors of [9-10] have used both chaos and neural networks in data encryption because of their cipher-suitable properties, such as parameter-sensitivity, time-varying, random-similarity, etc. Based on chaotic neural networks, a hash function is constructed, which makes use of neural networks' diffusion property and chaos' confusion property. This function encodes the plaintext of arbitrary length into the hash value of fixed length (typically, 128-bit, 256-bit or 512-bit).

Another demonstration of hash function implementation based on conservative chaotic system is proposed by authors of [11]. In their implementation the plaintext is divided into a group of message blocks by a fixed length and each message block is iterated some times through standard map. Both the iterations results of every round and the plaintext block determine the two initial values and the steps of iterations in next round. Some items of the result in the final round are chosen to be transformed into hash value of 128 bits. In the paper [12] a hash function construction method based on cellular neural network (CNN) with hyper-chaos characteristics is proposed. The chaos sequence generated by iterating CNN with Runge-Kutta algorithm, then the sequence iterates with every bit of the plaintext continually. Then hash code is obtained through. The corresponding transform of the latter chaos sequence from iteration. Hash code with different lengths could be generated from the former hash result. In [13] The MLP network structure developed for one way hashing consists of a hidden layer and an output layer. The hidden layer contains 64 neurons with 61 input including the bias. The weights of the hidden neurons are truncated to 3 decimal places, which α set to 1000. The MLP network thus structured is shown to be pre
image resistance, 2nd pre image resistance and collision resistance features.

In section 2 of our paper the neural network structures used in the proposed implementation are illustrated and explained. Section 3 provides the proposed algorithm details. Result calculation and sample data are elucidated in section 4. Conclusions are discussed in section 5.

2. NEURAL NETWORKS

Artificial neural network is an interconnected group of artificial neurons which use a mathematical model or a computational model[5] for information processing based on connectionist approach to computation. It is a network of simple processing elements which can exhibit complex behavior determined by the connections between processing elements and element parameters. Artificial neural network is an adaptive system that changes its structure based on external or internal information that flows through the network.

The method of setting the values for the weights enables the process of learning or training. The process of modifying the weights of the connections between network layers with the expected output is called training a network. The internal process that takes place when a network is trained is called learning. Figure 1 and Fig.2 show the neural network structure implemented in this paper without feedback and with feedback respectively.

The structure of feed-forward network as in Fig. 1 is made up of layers of neurons. For the purpose of the one-way hashing function, three layers of neurons are employed. The first layer is called ‘input’ layer, the next is the ‘hidden’ layer and the last layer is the ‘output’ layer. In our implementation, the input and the hidden layers consist of 128 neurons and the output layer consists of 64 neurons.

A sequence of binary bits are obtained from the image as mentioned in section 3. These bits are elements of out(1,n) from the input layer and are processed further by the hidden layer. The output of hidden layer becomes the input for the output layer. Due to the use of tan-sigmoidal function, non-linearity is introduced in each neuron of the output layer. In our structure, we have used a constant weight of 0.5 for each neuron and the bias is set to 0 for each neuron. Diffusion property is also satisfied due to the use of neural structure.

The structure of feedback network is again a structure of 3 layers of neurons. The interconnect between input and hidden layer output are as in feed forward structure.

Proposed Algorithm

For neural network structure without feedback
1. Read the input image
2. Convert the image into two dimensional pixel values
3. Determine the number of rows and columns of image
4. Create a row matrix of content values
5. Resize the row matrix into a matrix of size(r * 128). (r is the number of rows and it depends on the image which is being read)
6. Initialise each neuron in the input layer, hidden layer and output layer with a constant weight of 0.5 and a bias of 0.
7. Initialise the final layer output \([y_1(1), y_1(2), \ldots, y_1(64)]\) to zero
8. For \(k = 1\) to \(r\)
9. If(k=1)
10. Let the final layer output be y1(1), y1(2) . . . .y1(64)
11. End if
12. If (k>1)
13. \[hk-1 = [hk-2(1) EXOR yk(1)],[hk-2(2) EXOR yk(2)] . . . .
    [hk-2(64) EXOR yk(64)]\]
14. End if
15. End for k
16. The hash value of image without noise is hk-1
17. Modify the image by introducing one of the following
    four noises: One-bit change / Gaussian filtering / +5 degrees
    rotation / -5 degrees rotation.
18. Repeat Steps 7 to 15 for the modified image.
19. Let the hash of the modified image be Hnoise k-1.
20. Calculate the difference between hk-1 and Hnoise k-1.

For neural network structure with feedback
Repeat Steps 1 to 7 of without feedback
1. For k=1 to r
2. If(k>1)
3. Initialise t to 1
4. for j = 1 to 128 in steps of 2
5. out1(k,j) = EXOR[y(k-1,t),out1(k,j)];
6. out1(k,j+1) = EXOR[y(k-1,t),out1(k,j+1)];
7. t=t+1;
8. if(t == 65)
9. End for j
10. End if
11. End for k
12. Repeat Steps 16 to 20 of Without Feedback.

The input image is converted into two dimensional pixel values. 128 values of this matrix are given at a time to the input layer of the neural network. These values are passed through the hidden layer and 64 values of 38 bits each are obtained from each neuron of the output layer.

For the without feedback neural network structure, the output obtained from the consecutive iterations are XOR’ed to get the final hash value for the particular image. For the with feedback neural network structure, the values generated by each output neuron for the previous iteration are Xor’ed with the input values to two consecutive neurons in the current iteration. Let the hash value generated for the original image for both without feedback and with feedback structure be h1. Noise is introduced into the image and the procedure is repeated to obtain the new hash for the modified image. Let this hash be represented as h2. Sensitivity gives the number of bits change in the original image after addition of noise and is calculated as:

\[
\%\text{Sensitivity}= \frac{\text{Number of bits changed}}{\text{Total number of bits}} \times 100
\]

Hit collision is the number of digits (hex values) remaining same in the hash after addition of noise in the original image and is calculated as:

\[
\%\text{Hit collision} = \frac{\text{No. of hex values remaining same}}{\text{Total No. of hex values}} \times 100
\]

<table>
<thead>
<tr>
<th>Method</th>
<th>Plain Text Sensitivity (sec)</th>
<th>Collision Resistance (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without FB</td>
<td>106.015</td>
<td>5.312</td>
</tr>
<tr>
<td>With FB</td>
<td>73.453</td>
<td>6.75</td>
</tr>
<tr>
<td>RIPEMD1</td>
<td>76.219</td>
<td>17.016</td>
</tr>
<tr>
<td>MD5</td>
<td>76.969</td>
<td>6.938</td>
</tr>
</tbody>
</table>

3. Biometric images :

![Fig 3a:Biometric image](#)

![Fig 3b:Biometric image](#)
Hash code of fig 3a:
7B9E8D7B89A98A42AD657DBB725CA5BE7A6985407649
AE4A775A64B861424DA0B6B890698BA753504C65B3849
5C43A94F628FA766B9B28CB0B864B753A88BEE41619F4
D5196B5B56187B14DA2A04264584B4249B4577887B4
99A585BAD87E55B3936892AAA24445669F69B9E86B7706
5459DA0774A48686E54DBB

Hash code of fig 3b:
24047935D87A2312085E72B5DBD3D0466A550A25A8532
3BE832933557DC43DF8517BEA669D4AAD4E2FDAD3B
2C8579EF33FCAF348630211162B42C814C15B510EDBFD
5C23A8A54CE42FE700BF5B534F2899D91440A59ACD614
78F6B3958E6503160ABF0D45216DB82485515D2D0A
6A1907E91FA336546685DA

Execution time Analysis:

Collission Resistance

Number of Equal Entries

Without Feedback Plain text Sensitivity

With Feedback Plain text Sensitivity
4:CONCLUSION:

In this paper a unique hash value for a given Biometric image using neural networks is obtained. The code is written in MATLAB and tested for big range of biometric images and a unique hash obtained for each image. From the results it can be observed that the unique hash value obtained is sensitive to modifications made to the input biometric image. The sensitivity obtained is in the range of 45%-55%. For without feedback neural network structure sensitivity has been calculated using MD5 and RIPEMD160 Algorithm. These sensitivity values are compared with the values obtained from the proposed scheme using neural networks. The hit collisions obtained for the modified biometric image is found to be less than 10%. Lesser the hit collision better is the efficiency of the algorithm and thereby making cryptanalysis difficult. In this way it is concluded that with feedback network collision resistance and plaintext sensitivity is better.

5:REFERENCES:


