Secured Encryption Algorithm for Two Factor Biometric Keys

P. Muthu Kannan, Anupriya Asthana
1 Associate Professor, Galgotias University, India, E-mail: pmkannan@gmail.com
2 PG Scholar, Galgotias University, India, E-mail: a234asthana@gmail.com

Abstract—Wireless networks are increasingly used in various applications. Implementing security aspects for those networks are very critical as the communication is done in free space. Biometric is an emerging technology which provides effective solution to the security issues of wireless communication. Encryption is combined with the biometrics to provide better security for the wireless networks. Two factor biometric key can also be used to provide enhanced security level. In this work, Cryptographic keys which are generated from face and fingerprint are combined using an effective algorithm. This algorithm can provide better security to wireless communication.

Keywords—Biometric, Encryption, Biometric encryption, Two factor Biometric key, Cryptography, Bit -shifting.

I. INTRODUCTION

Biometric is an emerging technology which provides an effective solution regarding security issues of wireless communication. Biometric identifies or verifies individuals accurately in real time based upon their unique physical characteristics such as faces, hands, irises, and fingerprints or behavioural characteristics such as typing rhythm, gait and voice. These types of data are distinct from personal information and cannot be reverse-engineered to recreate personal information. Hence, it cannot be stolen and used to access personal information. Because of these inherent attributes, biometric is an effective means to provide secure privacy and prevent identity theft.

Conventional cryptography uses encryption keys, usually 128-bits or more. The problem with these conventional cryptography techniques is that a person cannot memorize such a long random key and it can be guessed, found or stolen by an attacker with a brute force search [10].

On the other hand, Biometric Encryption is a type of technology which has enormous potential to enhance privacy and security. Biometric Encryption is a combination of biometric and cryptography. Hence, it can be used as a solution to this problem as it is difficult for an intruder to know the biometric key [4].

Two factor biometric key can be used to provide enhanced security level. It uses key which is generated from more than one biometric. This provides reliable biometric keys for encryption algorithms and can be used for better security [8].

II. PROBLEM

Biometric key authentication process suffers from attacks like presenting fake biometrics, tampering with the biometric feature presentation, attacking the channel between stored template and the matching unit, corrupting the matching unit.

To avoid these attacks, two factor authentication biometric key is used. In this biometric key is generated from two biometric features. Each biometric feature will generate its own key. These two keys are combined with certain algorithm to give biometric key. In this paper, an algorithm is proposed with which the biometric keys can be processed to generate cryptographic key for suitable encryption procedures.

III. PROPOSED ALGORITHM

This section covers the details of proposed algorithm used to generate two factor biometric key of 32-bits. Algorithm is shown in fig.1. In this algorithm, operations like Ex-ORing, left & right shifting, splitting and shuffling are used. Two look-up tables T1 & T2 (shown in Table I and II) for conversion of 16-bits to 32-bits are also used. This algorithm comprises of the following steps:

Step 1:
Two 16-bits matrices, ‘a’ & ‘b’, are considered as inputs. These matrices ‘a’ & ‘b’ are derived from biometric keys generated from finger print and face respectively [8]. The logic behind the selection of bits to generate these 16-bits matrices is that: row-wise entries of matrix ‘a’ are the bits selected from those columns of 144-bits key [8] which are multiples of 9 and for matrix ‘b’ 14-bit key [8] is taken as whole with two zeros padded in last, to make it 16-bits. These inputs are converted into single 16-bits matrix ‘c’ by ex-oring different rows and columns of matrix ‘a’ with matrix ‘b’.

Step 2:
Row elements are left- shifted and a key, K of 16-bits is generated.
Step 3: To convert key, K into 32-bits look-up table T1 is used.
Step 4: 32-bits key is split into two parts K1 & K2, each of 16-bits.
Step 5: 2-bits right and left shifting are done with K1 & K2 respectively. This generates K1’ & K2’.
Step 6: K1’ & K2’ are Ex-ORed to generate K’, (16-bits).
Step 7: This 16-bits key is converted into 32-bits using look-up table T2 and split into two equal parts, K1” & K2”.
Step 8: Shuffling of four 8-bits blocks is done (two blocks from each, K1” & K2”).

The look-up tables are created with the following logic:
In table I, starting element of first and second column is taken as 15 and 0 respectively. Then first column is filled by decrementing and second column by incrementing by 2. Column three and four are filled in similar fashion as column second and first respectively, but in reverse order.

In table II, first four rows are filled from right-to-left whereas last four rows from left-to-right with multiples of 2, 3, 4 & 5.

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Table T1</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Table T1</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

![Fig.1. Proposed Algorithm](image)

IV. IMPLEMENTATION OF THE ALGORITHM
In this section, the functioning of the algorithm is discussed. Two random matrices ‘a’ and ‘b’, each of 16-bits are considered such as:

a = [1 0 0 1; 1 1 0 1; 0 0 1 0; 1 0 0 1]
b = [1 1 1 0; 1 1 1 0; 1 0 1 0; 0 1 0 0]
For converting these two 16-bits matrices into single 16-bits matrix Ex-OR operations are done with various rows & columns of ‘a’ and ‘b’ as shown below:

\[
\begin{align*}
& c_{12} = c_{21} = \text{xor} (a_{21}, b_{12}) \\
& c_{13} = c_{31} = \text{xor} (a_{31}, b_{13}) \\
& c_{14} = c_{41} = \text{xor} (a_{41}, b_{14}) \\
& c_{23} = c_{32} = \text{xor} (a_{32}, b_{23}) \\
& c_{24} = c_{42} = \text{xor} (a_{42}, b_{24}) \\
& c_{34} = c_{43} = \text{xor} (a_{43}, b_{34})
\end{align*}
\]

Thus the 16-bit matrix ‘c’ will be:

\[
c = [1 \ 0 \ 1 \ 1; \ 0 \ 1 \ 1 \ 0; \ 1 \ 1 \ 1 \ 0; \ 1 \ 0 \ 0 \ 1]
\]

Now, on left shifting each row element key, K of 16-bit will be generated. On writing row-wise:

\[
K = 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1
\]

This K is then converted into 32-bit by using table I:

\[
\begin{array}{cccccc}
K1 & & & & & \\
\hline
1 & 15 & 0 & 0 & 1 & 14 \\
0 & 13 & 1 & 2 & 0 & 12 \\
1 & 11 & 1 & 4 & 0 & 10 \\
1 & 9 & 0 & 6 & 1 & 8 \\
0 & 7 & 1 & 8 & 0 & 6 \\
1 & 5 & 0 & 10 & 1 & 4 \\
1 & 3 & 0 & 12 & 1 & 2 \\
1 & 1 & 1 & 14 & 0 & 0 \\
\end{array}
\]

On splitting table I into two halves:

\[
K1 = 10 \ 01 \ 11 \ 10 \ 01 \ 10 \ 11 \\
K2 = 11 \ 01 \ 01 \ 10 \ 01 \ 11 \ 01 \ 00 \ 01 \ 01
\]

Right shifting (2-bits):

\[
K1' = 11 \ 10 \ 01 \ 11 \ 10 \ 01 \ 10 \ 10
\]

Left shifting (2-bits):

\[
K2' = 01 \ 01 \ 10 \ 01 \ 11 \ 10 \ 01 \ 11
\]

K1’ (Ex-OR) K2’:

\[
K' = 10 \ 11 \ 11 \ 10 \ 01 \ 11 \ 11 \ 01
\]

Again, converting into 32-bits with the help of table II:

\[
\begin{array}{cccccc}
K1'' & & & & & \\
\hline
0 & 8 & 1 & 6 & 1 & 4 \\
1 & 12 & 1 & 9 & 1 & 6 \\
1 & 0 & 1 & 12 & 0 & 8 \\
1 & 4 & 1 & 15 & 1 & 10 \\
\end{array}
\]

\[
\begin{array}{cccccc}
K2'' & & & & & \\
\hline
1 & 2 & 1 & 4 & 1 & 6 \\
1 & 3 & 1 & 6 & 1 & 9 \\
1 & 4 & 0 & 8 & 1 & 12 \\
\end{array}
\]

\[
\begin{array}{cccccc}
1 & 5 & 1 & 10 & 1 & 15 \\
1 & 3 & 1 & 11 & 1 & 14 \\
1 & 2 & 1 & 12 & 1 & 13 \\
\end{array}
\]

\[
\begin{array}{cccccc}
K1''' & & & & & \\
\hline
0111 \ 1111 \ 1101 \ 1111 \\
1 & 3 & 1 & 11 & 1 & 14 \\
\end{array}
\]

\[
\begin{array}{cccccc}
K2''' & & & & & \\
\hline
1110 \ 1111 \ 1101 \ 1111 \\
4 & 2 & 1 & 12 & 1 & 13 \\
\end{array}
\]

Thus 32-bits key, K’’’ is obtained.

V. RESULT

With the proposed algorithm for the generation two factor biometric key, a 32-bits key is successfully generated. The two different 16-bits biometric keys, 1001110100101001010110111001001001, from finger print and face respectively are converted into 32-bits biometric key which is as follows:

Columns 1 through 18:

0 1 1 1 1 1 1 1 1 0 1 1 1

Columns 19 through 32:

0 1 1 1 1 1 1 1 0 1 1 1

VI. CONCLUSION

In this paper, an algorithm is proposed and implemented for the generation of two factor biometric key in order to provide more secure accessing of wireless networks. This algorithm combines the two biometric features. The objective of privacy concern is addressed by biometric encryption technique.

With this technique it will be highly difficult for the intruder to attack because to gain the knowledge of both the biometrics and their corresponding keys is very difficult. Also the algorithm for two factor biometric key authentication is to be known. So, the wireless networks can be more secured if two factor biometric key is combined with such an algorithm.

REFERENCES


