Design and implementation of combined effect of modified DES and Hamming (224,128) Code data security techniques on the transmission of 128-bit digital data from one base station to another base station written in VHDL

1Paresh Kumar Pasayat, Asst.Professor, IGIT Government Engineering College, Odish, India
2Rajashree Nath, UG student, IGIT Government Engineering College, Odish, India
3 Bipasha Pradhan, UG student, IGIT Government Engineering College, Odish, India
4 Anil Kumar Dakua, UG student, IGIT Government Engineering College, Odish, India
5Swadhyaya Mohanty, UG student, IGIT Government Engineering College, Odish, India
6Abhinav Das, UG student, IGIT Government Engineering College, Odish, India

Abstract - The proposed paper mainly deals with the data security algorithm used to provide security to the 128-bit digital data before transmission into the space. The desired 128-bit data is encrypted using modified DES producing 128-bit middle data and Hamming (224,128) code technique to produce 224-bit encrypted data. As the proposed design is having the combined effect of both modified DES and Hamming(224,128) code data security techniques, the security level is very high as compared to the design having individual data security technique. Due to the increment of key size from 56-bits to 112-bits in modified DES, the design is more resistive to the Brute-Force Attack. This can be used in the field of Automated teller machine transactions(ATM), Banking sector, Military sector and Protecting confidential company information. The proposed work is done by using VHDL language. The code is tested and simulated using Xilinx ISE9.2i software.

Key Words: ALU (Arithmetic Logic Unit), Encryption, Decryption, VHDL (Very High speed Integrated Circuit Hardware Description Language).

1. INTRODUCTION

If the digital data is transmitted directly without using encryption technique, then there is more probability of hacking and corruption of data by the attacker. Due to which, the various data security techniques have been designed by the designer to provide security to the data. The transformation of original data into a data which is not in the readable form is known as encryption and the process of reversing it back to a readable form is known as decryption. The proposed design shows how the 128-bit data is transmitted into space after doing encryption using modified DES and Hamming (224,128) code techniques.

1.1 Project Model

1.2 LOGIC USED IN THE PROPOSED DESIGN

The flow chart of the proposed design is given as follows:
The logic used in the modified DES encryption is given as follows:

**ROUND KEY GENERATOR:**
- KEY\_OUT\_1 <= KEY\_IN(0) & KEY\_IN(95 DOWNTO 1);
- KEY\_OUT\_2 <= KEY\_IN(1) & KEY\_IN(0) & KEY\_IN(95 DOWNTO 2);
- KEY\_OUT\_3 <= KEY\_IN(2) & KEY\_IN(1) & KEY\_IN(0) & KEY\_IN(95 DOWNTO 3);
- KEY\_OUT\_4 <= KEY\_IN(3) & KEY\_IN(2) & KEY\_IN(1) & KEY\_IN(0) & KEY\_IN(95 DOWNTO 4);
- KEY\_OUT\_5 <= NOT KEY\_IN(95 DOWNTO 0);
- KEY\_OUT\_6 <= NOT KEY\_IN(95 DOWNTO 0);
- KEY\_OUT\_7 <= NOT KEY\_IN(95 DOWNTO 0);
- KEY\_OUT\_8 <= KEY\_IN(45) & KEY\_IN(95 DOWNTO 1);
- KEY\_OUT\_9 <= KEY\_IN(48) & KEY\_IN(95 DOWNTO 1);
- KEY\_OUT\_10 <= KEY\_IN(41) & KEY\_IN(95 DOWNTO 1);
- KEY\_OUT\_11 <= KEY\_IN(45) & KEY\_IN(94 DOWNTO 1) & KEY\_IN(90);
- KEY\_OUT\_12 <= KEY\_IN(91) & KEY\_IN(95 DOWNTO 1);
- KEY\_OUT\_13 <= KEY\_IN(45) & KEY\_IN(95 DOWNTO 1);
- KEY\_OUT\_14 <= KEY\_IN(46) & KEY\_IN(95 DOWNTO 1);
- KEY\_OUT\_15 <= KEY\_IN(40) & KEY\_IN(95 DOWNTO 1);
- KEY\_OUT\_16 <= KEY\_IN(1) & KEY\_IN(95 DOWNTO 1);

Here KEY\_IN is the 112-bit cipher key and KEY\_OUT is the 16 nos. of keys generated from the Round Key Generator.

**INITIAL PERMUTATION UNIT:**
- IPU\_DATA\_OUT\_0 <= IPU\_DATA\_IN(127);
- IPU\_DATA\_OUT\_1 <= IPU\_DATA\_IN(126);
- IPU\_DATA\_OUT\_2 <= IPU\_DATA\_IN(125);
- IPU\_DATA\_OUT\_3 <= IPU\_DATA\_IN(124);
- IPU\_DATA\_OUT\_4 <= IPU\_DATA\_IN(123 DOWNTO 4);
- IPU\_DATA\_OUT\_5 <= IPU\_DATA\_IN(122);
- IPU\_DATA\_OUT\_6 <= IPU\_DATA\_IN(121);
- IPU\_DATA\_OUT\_7 <= IPU\_DATA\_IN(120);
- IPU\_DATA\_OUT\_8 <= IPU\_DATA\_IN(119);
- IPU\_DATA\_OUT\_9 <= IPU\_DATA\_IN(118);
- IPU\_DATA\_OUT\_10 <= IPU\_DATA\_IN(117);
- IPU\_DATA\_OUT\_11 <= IPU\_DATA\_IN(116);
- IPU\_DATA\_OUT\_12 <= IPU\_DATA\_IN(115);
- IPU\_DATA\_OUT\_13 <= IPU\_DATA\_IN(114);
- IPU\_DATA\_OUT\_14 <= IPU\_DATA\_IN(113);
- IPU\_DATA\_OUT\_15 <= IPU\_DATA\_IN(112);
- IPU\_DATA\_OUT\_16 <= IPU\_DATA\_IN(111);

The different operations that are performed in the Fiestal Cipher Unit are XOR operation, swapping operation, bit append operation.

**DES FUNCTION[F(R,KEY)]:**

The different units of the DES function used in the Fiestal Cipher is given as follows:
XOR UNIT:

\[
XU_{\text{OUT} \_\text{DATA}} <= XU_{\text{IN} \_\text{DATA} \_\text{ONE}} \, \text{XOR} \, XU_{\text{IN} \_\text{DATA} \_\text{TWO}};
\]

SWAP UNIT:

\[
SU_{\text{OUT} \_\text{DATA} \_\text{ONE}} <= SU_{\text{IN} \_\text{DATA} \_\text{TWO}};
\]

\[
SU_{\text{OUT} \_\text{DATA} \_\text{TWO}} <= SU_{\text{IN} \_\text{DATA} \_\text{ONE}};
\]

BIT APPEND UNIT:

\[
BAU_{\text{OUT} \_\text{DATA}} <= BAU_{\text{IN} \_\text{DATA} \_\text{ONE}} \, \& \, BAU_{\text{IN} \_\text{DATA} \_\text{TWO}};
\]

There are 16 nos. of round in the modified DES and after the completion of the round 16, the final permutation operation is performed.

FINAL PERMUTATION:

\[
\begin{align*}
FPU_{\text{DATA} \_\text{OUT}(0)} &= FPU_{\text{DATA} \_\text{IN}(127)}; \\
FPU_{\text{DATA} \_\text{OUT}(1)} &= FPU_{\text{DATA} \_\text{IN}(126)}; \\
FPU_{\text{DATA} \_\text{OUT}(2)} &= FPU_{\text{DATA} \_\text{IN}(125)}; \\
FPU_{\text{DATA} \_\text{OUT}(3)} &= FPU_{\text{DATA} \_\text{IN}(124)}; \\
FPU_{\text{DATA} \_\text{OUT}(123\,\text{DOWNTO}\,4)} &= FPU_{\text{DATA} \_\text{IN}(123\,\text{DOWNTO}\,4)}; \\
FPU_{\text{DATA} \_\text{OUT}(124)} &= FPU_{\text{DATA} \_\text{IN}(3)}; \\
FPU_{\text{DATA} \_\text{OUT}(125)} &= FPU_{\text{DATA} \_\text{IN}(2)}; \\
FPU_{\text{DATA} \_\text{OUT}(126)} &= FPU_{\text{DATA} \_\text{IN}(1)}; \\
FPU_{\text{DATA} \_\text{OUT}(127)} &= FPU_{\text{DATA} \_\text{IN}(0)};
\end{align*}
\]

Algorithm For Hamming (224,128) code Encryption Unit

Step 1

First, 128-bit data is divided into 32 nos. of words each consisting of 4-bit data.

Step 2

The 7-bit Hamming (7,4) code encoding technique is applied to each word. For each word, the encoding unit generates 7-bit encoded data. The logic for implementing the Hamming code technique is given as follows:

Suppose, the 4-bit data (B) to be encoded is B3B2B1B0 and the 7-bit Hamming code (H) generated is H6H5H4H3H2H1H0.

Here, the value for each bit of H is given as follows:

\[
\begin{align*}
H6 &= B3 \, \text{XOR} \, B2 \, \text{XOR} \, B0; \\
H5 &= B3 \, \text{XOR} \, B1 \, \text{XOR} \, B0; \\
H4 &= B2 \, \text{XOR} \, B1 \, \text{XOR} \, B0; \\
H3 &= B3; \\
H2 &= B2; \\
H1 &= B1; \\
H0 &= B0;
\end{align*}
\]

Step 3

After that the Hamming codes corresponding to each word are appended to form the desired 224-bit encoded data.

The block diagram showing the encryption process using the above algorithm is shown as follows:

2. RESULTS AND DISCUSSION

The VHDL code of the proposed project is compiled, synthesized and simulated using Xilinx ISE 9.2i software and the desired results have been obtained. The simulation result of the 128-bit digital data given to the DES encryption block to produce 128-bit middle data is shown as follows:

![Simulation result of modified DES encryption block](image-url)

The simulation result of the 128-bit middle data given to the Hamming(224,128) code encryption block to produce 224-bit encrypted data is shown as follows:

![Simulation result of Hamming(224,128) code encryption](image-url)
The comparison study has been done based on the maximum combinational path delays of different data security algorithms obtained from the Xilinx software written VHDL code which is shown as follows:

**Table 1: Comparison study**

<table>
<thead>
<tr>
<th>Name of the data security algorithm</th>
<th>Maximum combinational path delay found from the latest work (in ns):T1</th>
<th>Maximum combinational path delay obtained from the proposed work (in ns):T2</th>
<th>Complexity in terms of threshold value of Maximum combinational path delay</th>
<th>Security Level on the basic of Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBSTITUTION CIPHER</td>
<td>1.5</td>
<td>1.89</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>TRANSPOSITION CIPHER</td>
<td>5.4</td>
<td>6.479</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>HAMMING CODE</td>
<td>7.4</td>
<td>8.468</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>PROPOSED ALGORITHM</td>
<td>21.589</td>
<td></td>
<td>Very High</td>
<td>Very High</td>
</tr>
</tbody>
</table>

**Fig-2: Simulation result of Hamming(224,128) code encryption block**

3. CONCLUSIONS

As the proposed design is having the combined effect of both modified DES and Hamming (224,128) code data security techniques, the security level is very high as compared to the design having individual data security technique. Due to the increment of key size from 56-bits to 112-bits in modified DES, the design is more resistive to the Brute-Force Attack.

REFERENCES


