THE DEVELOPMENT OF ENCRYPTION AND DECRYPTION ALGORITHM FOR SECURED DATA TRANSFER BY USING VHDL

NURUL AIN BINTI ABDULLAH

This report is submitted in partial fulfillment of the requirements for the award of Bachelor of Electronic Engineering (Computer Engineering) With Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

May 2008
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI KEJURUTERAAN ELEKTRONIK DAN KEJURUTERAAN KOMPUTER

BORANG PENGESEHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : THE DEVELOPMENT OF ENCRYPTION & DECRYPTION
ALGORITHM FOR SECURED DATA TRANSFER BY USING VHDL
Sesi Pengajian : 2007/2008

Saya .........................NURUL AIN BINTI ABDULLAH .........................
(HURUF BESAR)
mengaku membenarkan Laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-
syarat kegunaan seperti berikut:
1. Laporan adalah hak milik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi
   pengajian tinggi.
4. Sila tandakan (√):
   - SULIT*
   - TERHAD*
   - TIDAK TERHAD

(Mengandungi maklumat yang berhubung kecemasan atau
kepentingan Malaysia seperti yang termaktub di dalam AKTA
RAHISIA RASMI 1972)
(Mengandungi maklumat terhad yang telah ditentukan oleh
organisasi/badan di mana penyelidikan dijalankan)

(TANDATANGAN PENULIS)
Alamat Tempat: JB7233 JALAN TEMBUSU
JASIN HEIGHTS
77000 JASIN
MELAKA
Tarikh: 9/5/2008

(COP DAN TANDATANGAN PENYELIA)

ABDUL MAJID B DARSONO
Pensyarah
Fakulti Kaj Eklokonik dan Kaj Kempudar (FKEKK),
Universiti Teknikal Malaysia Melaka (UTeM),
Karung Berkuno 1290,
Ayer Keroh, 75450 Malaka
Tarikh: 9/5/2008
"I hereby declare that this report is the result of my own work except for quotes as cited in the references."

Signature:  
Author: NURUL AIN BINTI ABDULLAH  
Date: 9/5/2008
“I hereby declare that I have read this report and in my opinion this report is sufficient in terms of the scope and quality for the award of Bachelor of Electronic Engineering (Computer Engineering) With Honours.”

Signature: ........................................
Supervisor's Name: ABDUL MAJID BIN DARSONO
Date: 9/5/2008
Dedicated for the hard times, for my supportive family and the ones who believe in me.
ACKNOWLEDGEMENT

Praises to Allah, for with His grace and blessings, this report of the Final Year Project has been completed successfully. I’d like to thank my supervisor, Mr Abdul Majid Darsono, for helping me throughout the entire project period. Not forgetting my family, for being the most supportive. Love you guys, to death. Also, gratitude to those who have been there through the times, the good and the bad, for believing in me and encouraged me to be stronger than I thought.
ABSTRACT

This project intends to develop a description for encryption and decryption algorithm by implementation of VHDL. Encryption and decryption is a vital part of cryptography. Encryption is a process of transforming information into something unreadable except those possessing the special key, while decryption is the reverse process of encryption. Encryption is now used to protect civilian's information. Certain information is deemed confidential and private therefore it has become a challenge to protect these kinds of information from hackers and the likes. Most of the time, these sensitive information is prone to attacks so security measures in the form of masking the information to make it unreadable to anyone except the intended recipient. This project's purpose is to describe the encryption and decryption algorithm, written in VHDL. Then, by using appropriate FPGA tools, the code description is simulated to predict the output. The VHDL description is later downloaded onto the FPGA board for software to hardware integration. The input is taken from a PS-2 keyboard, while the output is set to be the LCD display. By then, the input is tested with the description to ensure the correct output is displayed on the LCD.
ABSTRAK

# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER ITEM</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJECT TITLE</td>
<td>i</td>
</tr>
<tr>
<td>VERIFICATION FORM</td>
<td>ii</td>
</tr>
<tr>
<td>DECLARATION</td>
<td>iii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>v</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>vi</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>vii</td>
</tr>
<tr>
<td>ABSTRAK</td>
<td>viii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xiii</td>
</tr>
<tr>
<td>TABLE OF APPENDICES</td>
<td>xiv</td>
</tr>
</tbody>
</table>

## I

### INTRODUCTION

1.1 Introduction 1
1.2 Objectives 2
1.3 Problem Statement 3
1.4 Project Scope 4
1.5 Project Methodology 4
II \hspace{1cm} LITERATURE REVIEW

2.1 Introduction \hspace{1cm} 6

2.2 Types of Encryption Algorithm \hspace{1cm} 7
  2.2.1 Data Encryption Standard \hspace{1cm} 7
    2.2.1.1 History \hspace{1cm} 8
    2.2.1.2 Description \hspace{1cm} 8
    2.2.1.3 Overall Structure \hspace{1cm} 9
    2.2.1.4 The Feistel Function \hspace{1cm} 10
    2.2.1.5 Keyschedule \hspace{1cm} 12
    2.2.1.6 Security \hspace{1cm} 13

  2.2.2 Blowfish \hspace{1cm} 13
    2.2.2.1 The Algorithm \hspace{1cm} 14
    2.2.2.2 Security \hspace{1cm} 16

  2.2.3 Advanced Encryption Standard \hspace{1cm} 17
    2.2.3.1 Introduction \hspace{1cm} 17
    2.2.3.2 Description \hspace{1cm} 17
    2.2.3.3 High level cipher algorithm \hspace{1cm} 18
    2.2.3.4 Security \hspace{1cm} 21

III \hspace{1cm} PROJECT METHODOLOGY

3.1 Introduction \hspace{1cm} 23

3.2 Chosen Algorithm \hspace{1cm} 24

3.3 VHDL Implementation \hspace{1cm} 26

3.4 FPGA Board \hspace{1cm} 27

3.5 Flowchart \hspace{1cm} 28
# IV RESULT AND DISCUSSION

4.1 Discussion 32
4.2 The program 36
4.2.1 Devicelevel.vhd 36
4.2.2 Aescore.vhd 37
   4.2.2.1 Input capture 37
   4.2.2.2 Encryption Key Storage 38
   4.2.2.3 KeyExpansion Routine 39
   4.2.2.4 SubBytes 40
   4.2.2.5 ShiftRows 41
   4.2.2.6 MixColumns 41
   4.2.2.7 AddRoundKey 42
4.2.3 Prgrmctrl.vhd 43
4.2.4 Ps2_keyboard_interface.vhd 44
4.2.5 Lcd_top.vhd 44
4.3 The outcomes 44

# V CONCLUSION 47

REFERENCES 50

APPENDICES 51-57
# List of Tables

<table>
<thead>
<tr>
<th>NO</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Modes of operation</td>
<td>32</td>
</tr>
<tr>
<td>4.2</td>
<td>Input and output</td>
<td>33</td>
</tr>
<tr>
<td>4.3</td>
<td>Table of opcodes</td>
<td>34</td>
</tr>
<tr>
<td>4.4</td>
<td>Table of status word</td>
<td>34</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>NO</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The overall Feistel structure of DES</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>(a) The Feistel function (F-function) of DES</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>(b) The enlarged part of the substitution stage</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>The key-schedule of DES</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>(a) The Feistel structure of Blowfish</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>(b) The round function (Feistel function) of Blowfish</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>The SubBytes process</td>
<td>18</td>
</tr>
<tr>
<td>2.6</td>
<td>The ShiftRows process</td>
<td>19</td>
</tr>
<tr>
<td>2.7</td>
<td>MixColumns process</td>
<td>20</td>
</tr>
<tr>
<td>2.8</td>
<td>The AddRoundkey process</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Project Flowchart</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>Spartan-3 FPGA board</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>General Flowchart of Process</td>
<td>29</td>
</tr>
<tr>
<td>3.4</td>
<td>The flow of the program</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Input and output waveform</td>
<td>45</td>
</tr>
<tr>
<td>4.2</td>
<td>Design loading error</td>
<td>46</td>
</tr>
</tbody>
</table>
TABLE OF APPENDICES

<table>
<thead>
<tr>
<th>NO</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Softcode aescore.vhd</td>
<td>51</td>
</tr>
<tr>
<td>B</td>
<td>Spartan-3E FPGA Board datasheet</td>
<td>56</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

1.1 Introduction

Encryption is the process of transforming information to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The process will produce encrypted information. The word encryption also refers to the reverse process, decryption to make the encrypted information readable again (i.e. to make it unencrypted).

Militaries and governments have long facilitated the usage of secret communication. In modern times, encryption is now used in protecting information within many kinds of civilian systems. Encryption has long assist digital rights management to restrict the use of copyrighted material and in software copy protection to protect against reverse engineering and software piracy.

While then, cryptography is the system and procedure of masking information. These days, cryptography is considered to be a part of both mathematics and computer science, and is associated closely with information theory, computer security, and engineering. Cryptography is used in applications present in technologically advanced societies especially in the security of ATM cards, computer passwords, and electronic commerce, which all depend on cryptography.
By encrypting information, it is hidden under unintelligible gibberish which doesn’t make sense to human eyes but clearly defined in machine language. This step helps to protect a lot of these days consumer’s trade over the Internet.

This project aims to develop the existing algorithm for encryption and decryption. By using Very high speed integrated circuit Hardware Description Language (VHDL) to program selected algorithm standard, which reflects the encryption process that the algorithm is supposed to do.

There’s a lot of encryption algorithm available therefore, thorough understanding of each and every one of the algorithm is needed. Every algorithm has different scopes and ways, though they were created to serve the main purpose of encrypting information.

1.2 Objectives

This project aims are:

- To study and learn the concepts of cryptography
- To choose the suitable encryption and decryption algorithm standard
- To program using VHDL for the chosen algorithm standard
- To download softcode onto an FPGA board
- To test and troubleshoot the outcomes and problems in real-time

The first objective of the project is to study the concepts of cryptography. By learning the idea of and why encrypting information is important, the essential part before completing this project can be described successfully. Through identifying the vital information behind cryptography, then only the foundation can be implemented to the higher aspect of this project.
After grasping the basics, as there is a lot of encryption algorithm standard available nowadays, the most suitable algorithm will be chosen which signifies the purpose of the project. The chosen algorithm will be used to complete this project; whereas serving the rationale of encrypting and decrypting.

This project will also employ VHDL to program the chosen algorithm standard. Through the objective, the programming by using VHDL will mirror the process which exemplifies the algorithm itself. In the course of this stage, the information will be turned into gibberish and from gibberish into information again vice versa.

The program will then be downloaded onto a Field Programmable Gate Array (FPGA) board. This will allow the program to be tested on hardware using the correct input and output device. In this case, a keyboard as the input and Liquid Crystal Display (LCD) for the output. This stage also will be implemented with the last objective which is to test and troubleshoot the outcomes in real time. Integration to hardware sometimes will not be successful, therefore troubleshooting need to be done whether the program softcode or the hardware that caused the failure.

These objectives previously described are the vital aims for this project to be conducted successfully throughout the period of project commencement.

1.3 Problem Statement

Some information is deemed private and, thus the challenge to protect this information during transmission from hackers and the likes. At all times, the sensitive information is prone to attacks that will reveal the information, thus security measures have to be taken into consideration since some information may be the life and death of someone or another’s lifetime savings data.
These kinds of data when transferred over normal medium, can't be guaranteed would reach only the intended recipient, but possibly other party (ies) as well. Though these threats rarely happen, security can't be risked if in this case, the sensitive information are involved.

Cryptography provides a solution to this problem by ‘masking’ the information into something unrecognizable through a series of algorithm and providing key(s) to ‘unmask’ the information using the same procedure. Therefore only the person with the correct key can decrypt the information.

1.4 Project Scope

The scope of this project is to find the suitable algorithm to be used throughout the project period. Therefore, the full understanding of cryptography concepts are essential to be able to proceed with the project. This also includes a thorough study of available algorithm developing techniques and their advantages and disadvantages.

The algorithm developing method chosen will be developed by using VHDL, a machine description language. The soft code will be uploaded onto an FPGA board. The algorithm shall be able to encrypt the input entered by user through keyboard and decrypt them to be the exact input by user, displayed on the LCD.

1.5 Project Methodology

The project starts with studying the important concepts and terms for better understanding of the terminology to complete the project successfully. The flow of the softcode that is going to be written is charted for better viewing of the process that should take place. Then, the chosen algorithm standard is developed by using VHDL, a type of machine language as the project deals with encrypting and decrypting over
secured medium. The code written is then simulated on the simulation software to test whether the code written resulted in the expected outcome or not. By downloading the code, it should execute successfully with the input data entered by user shall be the output to determine the code developed is doing the correct processing. The correct display of output marks the end of the project as successful.
CHAPTER II

LITERATURE REVIEW

2.1 Introduction

Cryptography referred exclusively to encryption, where ordinary information (plaintext) is converted into unintelligible gibberish (called ciphertext). Decryption is the reverse, from unintelligible ciphertext decoded back to plaintext. A cipher is a pair of algorithms which perform this encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and, in each instance, by a key. [3]

This is a secret parameter (ideally, known only to the communicants) for a specific message exchange context. Keys are important, as ciphers without variable keys are trivially breakable and therefore less than useful for most purposes. [4] Historically, ciphers were often used directly for encryption or decryption, without additional procedures such as authentication or integrity checks.

In colloquial use, the term "code" is often used to mean any method of encryption or concealment of meaning. However, in cryptography, code has a more specific meaning; it means the replacement of a unit of plaintext (i.e., a meaningful word
or phrase) with a code word.[2] Codes are no longer used in serious cryptography—except incidentally for such things as unit designations—since properly chosen ciphers are both more practical and more secure than even the best codes, and better adapted to computers as well.

Some use the terms cryptography and cryptology interchangeably in English, while others use cryptography to refer to the use and practice of cryptographic techniques, and cryptology to refer to the subject as a field of study[5].

2.2 Types of Encryption Algorithm

There are various types of encryption available at the moment. The aim of each of the algorithm is the same: to encrypt information, though the level of encryption may vary according to the structure of the algorithm itself. This section will cover three types of the most popular encryption algorithm and comparing them. The algorithm are Data Encryption Standard (DES), Blowfish and Advanced Encryption Standard (AES).

2.2.1 Data Encryption Standard (DES)

The Data Encryption Standard (DES) is a method for encrypting information better known as cipher, is selected as an official Federal Information Processing Standard (FIPS) for the United States in 1976, and has been widely used internationally. The algorithm was with confidential design elements, and a fairly short key length. DES consequently was academically scrutinized, and modern understanding of block ciphers and their cryptanalysis were studied.[7]

DES is now regarded to be vulnerable for many applications. This is mainly due to the 56-bit key size being too small; DES keys have been broken in less than 24 hours.
There are also some results which reveal hypothetical flaws in the cipher, although they are impossible to mount in practice[5]. The algorithm is believed to be rationally secure in the form of Triple DES, although there are theoretical attacks.

2.2.1.1 History

In 1972, after concluding a study on the US government’s computer security requirements, the US standards body, NIST (National Institute of Standards and Technology) — acknowledged a need for a government-wide standard for encrypting uncategorized, sensitive information. On 15 May 1973, NBS solicited proposals for a cipher that would meet demanding design criteria after referring to the NSA.

DES was approved as a federal standard in November 1976, and published on 15 January 1977 as FIPS PUB 46, authorized for use on all unclassified data, despite the critiques. Later it was confirmed as the standard in 1983, revised as FIPS-46-1 in 1988, while in 1993 as FIPS-46-2, and lastly in 1998 as FIPS-46-3, prescribing Triple DES.[7]

The introduction of DES is considered to have been a catalyst for the academic study of cryptography, particularly of methods to crack block ciphers. According to a NIST retrospective about DES[7],

"The DES can be said to have "jump started" the nonmilitary study and development of encryption algorithms."

2.2.1.2 Description

DES is the conventional block cipher which is an algorithm that functions by taking a certain string with fixed-length of plaintext bits and after that putting the plaintext into a series of complicated operations where the plaintext will be transformed
into another ciphertext bitstring of the same length. In the case of DES, the block size is 64 bits.\[^{[4]}\] DES implements the usage of a key to customize the transformation, therefore the process of decryption can only be performed by those who know the particular key used to encrypt. The key apparently consists of 64 bits; however, only 56 of these are actually used by the algorithm since the rest of the eight bits are used solely for parity checking, and are thereafter discarded. Hence the effective key length is only 56 bits.

2.2.1.3 Overall structure

Figure 2.1 shows the overall structure of the algorithm. The representation of 16 rounds means the 16 identical stages of processing. The initial and final permutations which are inverses, where one will cancel the other’s actions, vice versa. Though these inverses have no cryptic implications, they were included in order to facilitate loading blocks in and out of mid-1970s hardware, apart from making DES run slower in software.\[^{[3]}\]

Before the main rounds, the block is split into two 32-bit halves and processed alternately; this criss-cross process is known as the Feistel scheme where this kind of structure ensures that decryption and encryption are very similar processes. The only difference would only be the reverse application of the subkeys during the decryption stage. The rest of the algorithm is identical which simplifies implementation, especially in hardware, as there is no need for separate encryption and decryption algorithms.\[^{[5]}\]

The F-block is used to jumble half a block together with some of the key. The output from the F-function is then combined with the other half of the block, and the halves are swapped before the next round. After the final round, the halves are not swapped; this is a feature of the Feistel structure which makes encryption and decryption similar processes.
2.2.1.4 The Feistel (F) function

The F-function, depicted in Figure 2.2, operates on half a block (32 bits) at a time and consists of four stages:

Expansion (E): By using the expansion permutation, the 32-bit half-block is expanded to 48 bits, by duplicating some of the bits.