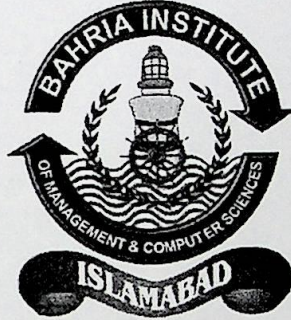


Absolute Security System



Developed by

Hafiz Misbahullah

Supervised by

Arshad Ali

A report is submitted to the department of Computer Sciences, Bahria Institute of management and Computer Sciences, Islamabad

In the partial fulfillment of the requirement for the degree of MCS

Department of Computer Sciences
Bahria University, Islamabad.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Department of Computer Sciences,
Bahria University, Islamabad.

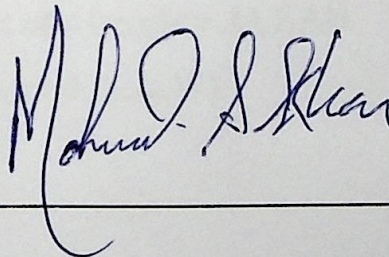
Final Approval

It is certified that we have read the project report submitted by **Hafiz Misbahullah** and it is judgment that this project id of sufficient standard warrant its acceptance by Bahria University, Islamabad for Master degree in Computer Science.

COMMITTEE

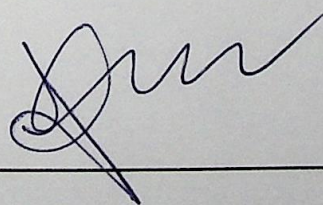
1. External Examiner:

Dr. M. A. Khan



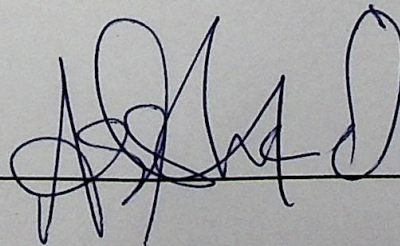
2. Internal Examiner:

Fazal-e-Wahab



3. Supervisor:

Arshad Ali



Dedicated

To

**The greatest and the Holiest man ever born
The last Prophet of Almighty ALLAH
Muhammad (S.A.W)**

Acknowledgement

All Praise to be Allah Almighty, the most Merciful, the most Gracious, without his help and blessing I was unable to complete the project.

This effort wouldn't have materialized into a reality had there not been the contributions of so many people. I would like to pay my thanks to my teachers for the valuable contributions, which enable me to be a part of Computer Science world.

Many words of thank for my supervisor **Mr. Arshad Ali**. He kept me encouraging all the time and provided guidance whenever I was in need of. He is the person who bailed me out of all the intricacies of research and I owe him the most heartfelt of my gratitude.

I want to pay my sincere thanks to my Project Partner Qazafi Mahmood for his valuable supervision, guidance, constant encouragement and personal involvement, which enable me to complete this challenging task.

Finally I would like to pay my deep gratitude to my loving parents my sisters and brothers for their good wishes and encouragement, without which I would have been unable to accomplish any thing worthwhile.

Hafiz Misbahullah

Project in Brief

Project Title: Absolute Security System

Organization: Department of Computer Sciences,
Bahria University, Islamabad

Undertaken By: Hafiz Misbahullah

Supervised By: Mr. Arshad Ali

Tool Used: Microsoft Visual C++ 6.0
ATL COM

Operating System: Windows NT, Windows 2000.

System Used: IBM Compatible

Date Started: 1st, Jan 2002.

Date Completed: 10, Sep 2002.

Abstract

Security systems today are built on increasingly strong cryptographic algorithms that foil pattern analysis attempts. However, the security of these systems is dependent on generating secret quantities for passwords, cryptographic keys, and similar quantities. The use of pseudo-random processes to generate secret quantities can result in pseudo-security. The sophisticated attacker of these security systems may find it easier to reproduce the environment that produced the secret quantities, searching the resulting small set of possibilities, than to locate the quantities in the whole of the number space.

Choosing random quantities to foil a resourceful and motivated adversary is surprisingly difficult. This underlying software, *Absolute Security*, implements the five pseudo random numbers generators and one pure random numbers generator and different methods for encryption. The system also provides a facility for storing and manipulating the keypad of random bits. It also provides the dial-up connectivity to the remote user for data transmission on the secure channel.

The data to be transmitted may include Audio, Video, and text. The user would only have to select different options among many options provided to identify the required service category. The system would then encrypt the selected data according to the algorithm selected. The connection would be established with the number selected by the user and the encrypted file could be sent easily to the destination.

TABLE OF CONTENTS

Acknowledgement	iii
Project in Brief	iv
Abstract	v
Table of Contents	vi
Table of Figures	vii
List of Tables.....	viii
1 Chapter 1: Introduction	1
1.1 Encryption and decryption.	3
1.2 History of Cryptography.	4
1.3 Information security and cryptography.	6
1.4 What is cryptography ?	8
1.4.1 Cryptographic goals.	9
1.5 How does cryptography work ?	10
1.6 Conventional cryptography.	11
1.7 Public key cryptography.....	11
1.8 Randomness Requirement.....	12
1.9 Traditional Pseudo-Random Sequences.....	14
1.10 Unpredictability.....	17
1.11 Problems with Clocks and Serial Numbers.....	18
1.12 Timing and Content of External Events.....	19
1.13 The Fallacy of Complex Manipulation.	20
1.14 The Fallacy of Selection from a Large Database.	21
2 Chapter 2: Windows Telephony API (TAPI)	22
2.1 TAPI is open, comprehensive, scalable, and integrated.....	23
2.2 The Microsoft Telephony Programming Model.	25
2.3 Unified Communications with TAPI 3.0.	27
2.4 TAPI Version 3.0.	28
2.5 Compatibility with previous versions of TAPI	30
2.6 TAPI Applications.....	33
2.7 Basics of Serial Communications.	35
2.8 Creating a TAPI Connection Using the MS Comm Control.....	36
2.8.1 The Communications Control	36
2.8.2 Possible Uses.....	36
2.8.3 Establishing the Serial Connection	36
2.8.3.1 Opening the Serial Port	38
2.8.4 Working with a Modem	39
2.8.5 Setting Receive and Transmit Buffer Properties.....	40
a. Buffer Memory Allocation.....	43
b. The RThreshold and SThreshold Properties.	43
c. The InputLen and EOFEnable Properties.	44
2.8.6 Managing the Receive and Transmit Buffers.....	44
2.8.7 Handshaking.....	47
2.8.8 The OnComm Event and the CommEvent Property	49

3	Chapter 3: System Analysis.....	52
3.1	Requirement specifications	53
3.1.1	Overview statement.....	53
3.1.2	Goals.....	53
3.1.3	Cryptographic Specifications	54
3.2	UML (Unified Modeling Language).....	54
3.2.1	Use Case.....	54
3.2.2	Actor.....	55
3.2.3	System Boundary	55
3.2.4	Associations	56
3.3	Use Case Diagram	57
3.3.1	Actor: Main Frame Window	58
3.3.2	Generate Keypad (Pure Random Bit Sequence)	58
3.3.3	Generate Keypad (Alternating Step/Shrinking Generator)	59
3.3.4	Encrypt Text.....	61
3.3.5	Decrypt Text.....	62
3.3.6	Encrypt/Record Audio.....	64
3.3.7	Decrypt/Play Audio.....	65
3.3.8	Establish Connection.....	67
3.3.9	Transmit Data File.....	68
3.4	Conceptual Model	70
4	Chapter 4: System Design	72
4.1	Software Architecture	73
4.2	Software Class Diagram.....	75
4.3	Sequence Diagrams	77
4.3.1	Generate Pure Random Numbers.....	78
4.3.2	Generate Pseudorandom Numbers (Alternating Step Generator)	79
4.3.3	Generate Pure Random Numbers (Shrinking Generator)	80
4.3.4	Test Random Numbers.....	81
4.3.5	Encrypt Text.....	82
4.3.6	Decrypt Text.....	83
4.3.7	Encrypt Audio	84
4.3.8	Decrypt Audio	85
4.3.9	Add Phone Numbers	86
4.3.10	Dial-up Settings.....	87
4.3.11	Transmit File	88
4.3.12	Receive File.....	89
4.4	Collaboration Diagrams	90
4.4.1	Generate Pure Random Numbers.....	91
4.4.2	Generate Pseudorandom Numbers (Alternating Step Generator)	92
4.4.3	Generate Pure Random Numbers (Shrinking Generator)	93
4.4.4	Test Random Numbers.....	94
4.4.5	Encrypt Text.....	95
4.4.6	Decrypt Text.....	96
4.4.7	Encrypt Audio	97
4.4.8	Decrypt Audio	98
4.4.9	Add Phone Numbers	99
4.4.10	Dial-up Settings.....	100

4.4.11	Transmit File	101
4.4.12	Receive File.....	102
5	Chapter 5: System Testing	103
5.1	Objective of Testing	104
5.2	Testing Strategies	104
5.2.1	Code Testing Strategy	105
5.2.2	Specification testing strategy	105
5.3	Level of Testing Performed.....	105
5.3.1	Unit Testing.....	105
5.3.2	Integration Testing	106
5.3.3	System Testing	106
5.4	Special Test Performed	106
5.4.1	Peak Load Testing.....	107
5.4.2	Performance Time Testing.....	107
5.5	Recovery Testing.....	107
	Bibliography and References	109

TABLE OF FIGURES

Figure 1.9-1: Linear Shift Register Pseudo-Random Number Generator.....	16
Figure 2.2-1: Microsoft Telephony Programming Model.....	26
Figure 2.5-1: Compatibility with previous versions of TAPI	31
Figure 2.5-2: Compatibility with previous versions of TAPI_2	32
Figure 2.6-1: Basic building blocks of TAPI applications.....	34
Figure 2.8.5-1: Setting modem properties.....	41
Figure 2.8.5-2: Setting Receive and Transmit buffer properties.....	41
Figure 2.8.5-3: Setting Hardware Properties.....	42
Figure 3.2.1-1: Use Case	55
Figure 3.2.2-1: Actor	55
Figure 3.2.3-1: System Boundary	55
Figure 3.2.4-1: Associations.....	56
Figure 3.3-1: System Use Case Diagram	57
Figure 3.4-1: Conceptual Model	71
Figure 4.1-1: Software Architecture.....	74
Figure 4.2-1: Software Class Diagram	76
Figure 4.3.1-1: Generate Pure Random Numbers	78
Figure 4.3.2-1: Generate Pseudorandom Numbers (ASG)	79
Figure 4.3.3-1: Generate Pseudorandom Numbers (SG)	80
Figure 4.3.4-1: Test Random Numbers.....	81
Figure 4.3.5-1: Encrypt Text.....	82
Figure 4.3.6-1: Decrypt Text.....	83
Figure 4.3.7-1: Encrypt Audio	84
Figure 4.3.8-1: Decrypt Audio	85
Figure 4.3.9-1: Add Phone Numbers	86
Figure 4.3.10-1: Dial-up Settings.....	87
Figure 4.3.11-1: Transmit File	88
Figure 4.3.12-1: Receive File.....	89
Figure 4.4.1-1: Generate Pure Random Numbers	91
Figure 4.4.2-1: Generate Pseudorandom Numbers (ASG)	92
Figure 4.4.3-1: Generate Pseudorandom Numbers (SG)	93
Figure 4.4.4-1: Test Random Numbers.....	94
Figure 4.4.5-1: Encrypt Text.....	95
Figure 4.4.6-1: Decrypt Text.....	96
Figure 4.4.7-1: Encrypt Audio	97
Figure 4.4.8-1: Decrypt Audio	98
Figure 4.4.9-1: Add Phone Numbers	99
Figure 4.4.10-1: Dial-up Settings.....	100
Figure 4.4.11-1: Transmit File	101
Figure 4.4.12-1: Receive File.....	102

LIST OF TABLES

Table 2.8.3-1	37
Table 2.8.7-1	48
Table 2.8.8-1	50
Table 2.8.8-2	51

CHAPTER 1 INTRODUCTION