Development of an Encryption Algorithm for Transmitting a Plain Text

By
Mohamed Sirelkhatim Omer Hamid
B.Sc. (Honours) in Mathematical and Computer Science, University of Gezira, 2004.

A Dissertation
Submitted in Partial Fulfilment of the Requirements for the
Degree of Master of Science in Computer Science and Information

Department of Computer Science, Faculty of Mathematical and Computer Science,
University of Gezira

Supervisor: Dr. Ashraf Gasim Elsid
Co-supervisor: Dr. Elsadig Abdalla Aljack Fadalalla

January 2009
Development of an Encryption Algorithm for Transmitting a Plain Text

By
Mohamed Sirelkhatim Omer Hamid

Examination Committee

<table>
<thead>
<tr>
<th>Name</th>
<th>Position</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Ashraf Gasim Elsid</td>
<td>Chairman</td>
<td>.....................................................</td>
</tr>
<tr>
<td>Dr. Saife Eldin Fattoh Osman Yaseen</td>
<td>External Examiner</td>
<td>.....................................................</td>
</tr>
<tr>
<td>Dr. Ali Mohammed Abdelrahman Agoub</td>
<td>Internal Examiner</td>
<td>.....................................................</td>
</tr>
</tbody>
</table>

Date of examination: 6 / 1 /2009
Dedication

To My, Mother
To My, Father
To My, Friends
And to People I Love
ACKNOWLEDGEMENTS

First of all I truly thank my main supervisor Dr. Ashraf for his continuous advice and support during the course of this study. I appreciate the assistance offered to me by Mr. Mohammed Ibrahim. I never forget to thank my family for their patience and encouragement during my study. Great thank to all my friends for great help and support.
Development of an Encryption Algorithm for Transmitting a Plain Text

Mohamed Sirelkhatim Omer Hamid
M.Sc. in Computer Science and Information
(2009)
Department of Computer Science
Faculty of Mathematical and Computer Science

ABSTRACT

Transmitting data among private network is more secure than transmitting data among public networks such as internet. But unfortunately, private network is not available for majority of people or even majority of companies or organizations, as designing and implementation of private networks is too expensive. Virtual Private Networks solve the expressivity problem partially; but there are still a lot of security issues surrounding data transmitting through even Virtual Private Network. One of the security problems, intruders can snoop on the contents of sent messages. Moreover, they can prevent the message from reaching its intended destination. This dissertation endeavors to develop and implement a complex encryption algorithm for a plain text to be transmitted among public networks. Moreover, it embodies technique which guarantees that transmitted plain text will be receive only by the intended destination and nobody else.
تطوير خوارزمية تشفير لإرسال النص العادي

جامعة الجزيرة

محمد سرالختم عمر حامد

(2009)

الماجستير في علوم الحاسوب والمعلومات
قسم علوم الحاسوب
كلية العلوم الرياضية والحاسب

مستخلص

 عملية إرسال البيانات عبر الشبكات الخاصة تكون أكثر أماناً من إرسالها عبر الشبكات العامة كشبكة الإنترنت مثلًا. لكن الشبكات الخاصة لا تكون متاحة لغالبية الأفراد أو حتى الشركات أو المنظمات وذلك نسبة لارتفاع تكاليف تصميم وتنفيذ الشبكات الخاصة. الشبكات الإفتراضية الخاصة قللت جزئياً من تكاليف تنفيذ الشبكات الخاصة ولكن ما زالت هناك بعض المشاكل المتعلقة بسرية البيانات. من هذه المشاكل معرفة محتوى البيانات المرسلة لغير المرسل إليه وأيضاً إعاقة البيانات المرسلة من الوصول إلى المستقبل المحدد. يهدف هذا البحث إلى تطوير خوارزمية تشفير معقدة لإرسال نص عادي عبر شبكة عامة مع التأكد من أن المستلم هو المستلم الصحيح وعدم إمكانية إستلام البيانات بواسطة مستلم آخر.
Table of Contents

Dedication ........................................................................................................................................ i
Acknowledgement ......................................................................................................................... ii
Abstract ......................................................................................................................................... iii
Abstract in Arabic ......................................................................................................................... iv
List of Figures ................................................................................................................................... viii
List of Abbreviations .................................................................................................................. ix

Chapter 1: Introduction

1.1 Introduction ................................................................................................................................. 1
1.2 Problem Statement ..................................................................................................................... 1
1.3 Objectives of the Research ......................................................................................................... 2
1.4 Organization of the Dissertation .............................................................................................. 2

Chapter 2: Virtual Private Networks

2.1 Networks Concepts ................................................................................................................... 3
2.2 Virtual Private Networks ......................................................................................................... 3
2.3 Common Uses of VPNs .............................................................................................................. 5
2.3.1 Remote User Access Over the Internet ............................................................................... 5
2.3.2 Connecting Networks Over the Internet ............................................................................ 6
2.3.2.1 Using Dedicated Lines .................................................................................................. 6
2.3.2.2 Using Dial Up ................................................................................................................ 6
2.3.3 Connecting Computers over an Internet ............................................................................. 7
2.4 VPN Connections ....................................................................................................................... 8
2.4.1 Remote Access VPN Connection ....................................................................................... 8
# Chapter 2: VPN Connections

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.2</td>
<td>Router to Router VPN Connection</td>
<td>8</td>
</tr>
<tr>
<td>2.5</td>
<td>Basic VPN Requirements</td>
<td>8</td>
</tr>
<tr>
<td>2.5.1</td>
<td>User Authentication</td>
<td>9</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Address Management</td>
<td>9</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Data Encryption</td>
<td>9</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Key Management</td>
<td>9</td>
</tr>
<tr>
<td>2.5.5</td>
<td>Multiprotocol Support</td>
<td>9</td>
</tr>
<tr>
<td>2.6</td>
<td>Components of Virtual Private Networks</td>
<td>10</td>
</tr>
<tr>
<td>2.7</td>
<td>Properties of VPN Connections</td>
<td>11</td>
</tr>
<tr>
<td>2.7.1</td>
<td>Encapsulation</td>
<td>11</td>
</tr>
<tr>
<td>2.7.2</td>
<td>Authentication</td>
<td>11</td>
</tr>
<tr>
<td>2.7.3</td>
<td>Data encryption</td>
<td>12</td>
</tr>
</tbody>
</table>

# Chapter 3: Tunneling Technique

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Tunneling Basics</td>
<td>13</td>
</tr>
<tr>
<td>3.2</td>
<td>Tunneling Protocols</td>
<td>14</td>
</tr>
<tr>
<td>3.1.1</td>
<td>How Tunneling Works</td>
<td>14</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Tunneling Protocols and the Basic Tunneling Requirements</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Point to Point Protocol (PPP)</td>
<td>16</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Phase 1: PPP Link Establishment</td>
<td>16</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Phase 2: User Authentication</td>
<td>17</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Phase 3: PPP Callback Control</td>
<td>18</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Phase 4: Invoking Network Layer Protocol(s)</td>
<td>18</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Phase 5 Data-Transfer Phase</td>
<td>19</td>
</tr>
<tr>
<td>3.4</td>
<td>Tunnel Types</td>
<td>19</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Voluntary Tunneling</td>
<td>19</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Compulsory Tunneling</td>
<td>20</td>
</tr>
<tr>
<td>3.5</td>
<td>Advanced Security Features</td>
<td>21</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Symmetric Encryption vs. Asymmetric Encryption (Private Key vs. Public Key)</td>
<td>22</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Certificates</td>
<td>22</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Extensible Authentication Protocol (EAP)</td>
<td>23</td>
</tr>
</tbody>
</table>
3.5.4 Transaction-level Security (EAP-TLS) ................................................................. 23
3.5.5 IP Security (IPSec) ......................................................................................... 24
3.5.6 Negotiated Security Association ................................................................. 24
3.5.7 Authentication Header ................................................................................ 25
3.5.8 Encapsulation Security Header ................................................................. 25

Chapter 4: Design of Encryption Algorithm

4.1 The Algorithm Methodology ...................................................................... 26
4.2 Design and Implementation ..................................................................... 26
4.3 Algorithm Logic ........................................................................................ 27

Chapter 5: Algorithm Implementation

5.1 Development Software Tool ..................................................................... 36
5.1.1 Winsock Tool ......................................................................................... 36
5.1.2 Microsoft Visual Basic ........................................................................ 36
5.2 Presentation of Algorithm Functions ....................................................... 37
5.3 Algorithm Testing ..................................................................................... 43
5.4 Value of the Software Tool ..................................................................... 43

Chapter 6: Conclusion and Recommendations

6.1 Conclusion ............................................................................................... 44
6.2 Recommendations for Future Works ....................................................... 44

References .................................................................................................... 59
Appendix ........................................................................................................ 60
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page No</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The Logical Equivalent of a VPN Connection</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Using a VPN to Connect a Remote Client to a Private LAN</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Using a VPN to Connect Two Remote Sites</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>Using a VPN to Connect Two Computers On The Same LAN</td>
<td>9</td>
</tr>
<tr>
<td>2.5</td>
<td>component of virtual private network</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>Tunneling</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Compulsory Tunneling</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Macro Level – Development of Software Tool</td>
<td>39</td>
</tr>
<tr>
<td>4.2</td>
<td>Converting Key to Binary Number</td>
<td>40</td>
</tr>
<tr>
<td>4.3</td>
<td>Converting Key from Binary to Decimal</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Encrypting the Plain Text – First Level</td>
<td>42</td>
</tr>
<tr>
<td>4.5</td>
<td>Encrypting the Plain Text – Second Level</td>
<td>43</td>
</tr>
<tr>
<td>4.6</td>
<td>Reversing the Second Generated Text</td>
<td>44</td>
</tr>
<tr>
<td>4.7</td>
<td>Decrypting the Encrypted Text</td>
<td>45</td>
</tr>
<tr>
<td>4.8</td>
<td>Decrypting the Encrypted Text (Cont.)</td>
<td>46</td>
</tr>
<tr>
<td>5.1</td>
<td>Startup Window</td>
<td>49</td>
</tr>
<tr>
<td>5.2</td>
<td>Main Window</td>
<td>50</td>
</tr>
<tr>
<td>5.3</td>
<td>Remote Computer Determiner</td>
<td>51</td>
</tr>
<tr>
<td>5.4</td>
<td>Encryption and Transmission 1</td>
<td>52</td>
</tr>
<tr>
<td>5.5</td>
<td>Encryption and Transmission 2</td>
<td>53</td>
</tr>
<tr>
<td>5.6</td>
<td>Receiving and Decrypting 1</td>
<td>54</td>
</tr>
<tr>
<td>5.7</td>
<td>Receiving and Decrypting 2</td>
<td>55</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Networks</td>
<td></td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
<td></td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
<td></td>
</tr>
<tr>
<td>NAS</td>
<td>Network Access Server</td>
<td></td>
</tr>
<tr>
<td>SNA</td>
<td>System Network Architecture</td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
<td></td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
<td></td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
<td></td>
</tr>
<tr>
<td>PPTP</td>
<td>Point-to-Point Tunneling Protocol</td>
<td></td>
</tr>
<tr>
<td>MPPE</td>
<td>Microsoft Point-to-Point Encryption</td>
<td></td>
</tr>
<tr>
<td>L2TP</td>
<td>Layer Two Tunneling Protocol</td>
<td></td>
</tr>
<tr>
<td>IPSec</td>
<td>Internet Protocol security</td>
<td></td>
</tr>
<tr>
<td>ppp</td>
<td>Point-to-Point Protocol</td>
<td></td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
<td></td>
</tr>
<tr>
<td>IPX</td>
<td>Internet Packet Exchange</td>
<td></td>
</tr>
<tr>
<td>IKE</td>
<td>Internet Key Exchange</td>
<td></td>
</tr>
<tr>
<td>L2F</td>
<td>Layer 2 Forwarding</td>
<td></td>
</tr>
<tr>
<td>EAP</td>
<td>Extensible Authentication Protocol</td>
<td></td>
</tr>
<tr>
<td>NCP</td>
<td>Network Control Protocol</td>
<td></td>
</tr>
<tr>
<td>MPPC</td>
<td>Microsoft Point-to-Point Compression</td>
<td></td>
</tr>
<tr>
<td>LCP</td>
<td>Link Control Protocol</td>
<td></td>
</tr>
<tr>
<td>PAP</td>
<td>Password Authentication Protocol</td>
<td></td>
</tr>
<tr>
<td>CHAP</td>
<td>Challenge Handshake Authentication Protocol</td>
<td></td>
</tr>
<tr>
<td>MSCHAP</td>
<td>Microsoft Challenge Handshake Authentication Protocol</td>
<td></td>
</tr>
<tr>
<td>CBCP</td>
<td>Callback Control Protocol</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>FEP</td>
<td>Front End Processor</td>
<td></td>
</tr>
<tr>
<td>LAC</td>
<td>L2TP Access Concentrator</td>
<td></td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
<td></td>
</tr>
<tr>
<td>IDEA</td>
<td>International Data Encryption Algorithm</td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td>Certificate Authority</td>
<td></td>
</tr>
<tr>
<td>TLS</td>
<td>Transaction-level Security</td>
<td></td>
</tr>
<tr>
<td>AH</td>
<td>Authentication Header</td>
<td></td>
</tr>
<tr>
<td>ESP</td>
<td>Encapsulated Security Payload</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER ONE
INTRODUCTION

1.1 Introduction

Virtual private network (VPN) connects the components of one network over another network. VPNs accomplish this by allowing the user to tunnel through the internet or another public network in a manner that provides the same security and features formerly available only in private networks.

VPN technology is designed to address issues surrounding the current business trend toward increased telecommuting and widely distributed global operations, where employees must be able to connect to central resources and must be able to communicate each other.

To provide employees of a corporation with the ability to connect to the corporation computing resources regardless of their location, the corporation must deploy scalable remote access solution.

The remote access solution can be achieved by using Internet as it the cheapest connection technology. With an Internet solution, a few internet connections through independent service providers and VPN server computers can serve the remote networking need of hundreds or thousands of remote clients and branch offices.

1.2 Problem Statement

When a source sends a message to destination over a public network such as the Internet, the message passes through many computers, routers, switches and similar equipment before it arrives at the destination computer. An Intruder has many opportunities to intercept and read the message along the way and even to alter it, so that the message that the destination receives may be quite different from the original one. In fact, the intruder might even send the receiver a false message, disguised to appear as though it was sent by the sender.

The source and the destination want to ensure that no one can spy to their communication. Moreover, the source wants to be sure that only the intended destination can read the message he sends. This is a privacy issue.
The source and the destination also want to ensure that no one is tampering with their communication. The source wants to be sure that the message that intended destination will receive is exactly the same message that he sent, that is, the message was not tampered with in transit. This is an integrity issue. The source and the destination also want to ensure that no one is sending false messages. The destination wants to be sure that the message he received from the source really did come from the true source, and not from someone else. This is an authentication issue. All the above mentioned issues are not big deals when using private networks; however, it becomes vital problems when using public networks – the cheapest, available and common communication media.

1.3 Objectives of the Research

The aim of this dissertation is to develop and implement a complex encryption algorithm for plain text. This algorithm works as a tunneling technique used to transmit data among public networks. Moreover it adds or provides additional features to grantees the following characteristics:

- Intruders can not access the transmitted data.
- Snoopers can not get the real contents of transmitted data.
- Only the intended destination will receive the transmitted data.

1.4 Organization of the Dissertation

Chapter two describes major concepts of networking and slightly discusses virtual private networks. Chapter three discusses basics of tunneling and tunneling protocols and types of tunneling. It also describes the features of tunneling. Chapter four investigates the vision of designing the software tool. Value of the software tool is discussed and also its conceptual design. Chapter five covers the implementation of the software tool. Chapter six discusses and summarizes the work covered, its contribution to the field securing data, and also suggests some possible future directions.
CHAPTER TWO
LITERATURE REVIEW

2.1 Networks Concepts

Networks and networking have grown exponentially over the last 15 years understandably so. They’ve had to evolve at light speed just to keep up with huge increases in basic mission critical user needs such as sharing data and printers as well as more advanced demands such as video conferencing. Unless everyone who needs to share network resources is located in the same office area (an increasingly uncommon situation), the challenge is to connect the sometimes many relevant networks together so all users can share the networks’ wealth.

2.2 Virtual Private Networks

A virtual private network (VPN) is the extension of a private network that encompasses links across shared or public networks like the Internet. With a VPN, you can send data between two computers across a shared or public network in a manner that emulates a point-to-point private link. Virtual private networking is the act of creating and configuring a virtual private network. [5].

To emulate a point-to-point link, data is encapsulated, or wrapped, with a header that provides routing information, which allows the data to traverse the shared or public network to reach its endpoint. To emulate a private link, the data is encrypted for confidentiality. Packets that are intercepted on the shared or public network are indecipherable without the encryption keys. The link in which the private data is encapsulated and encrypted is a virtual private network (VPN) connection. The following illustration shows the logical equivalent of a VPN connection.
Users working at home or on the road can use VPN connections to establish a remote access connection to an organization server by using the infrastructure provided by a public network such as the Internet. From the user’s perspective, the VPN is a point-to-point connection between the computer (the VPN client) and an organization server (the VPN server). The exact infrastructure of the shared or public network is irrelevant because it appears logically as if the data is sent over a dedicated private link.

Organizations can also use VPN connections to establish routed connections with geographically separate offices or with other organizations over a public network such as the Internet while maintaining secure communications. A routed VPN connection across the Internet logically operates as a dedicated WAN link.

With both remote access and routed connections, an organization can use VPN connections to trade long-distance dial-up or leased lines for local dial-up or leased lines to an Internet service provider (ISP).

There are two types of Point-to-Point Protocol (PPP)-based VPN technology: [5].

1. Point-to-Point Tunneling Protocol (PPTP)

   PPTP uses user-level PPP authentication methods and Microsoft Point-to-Point Encryption (MPPE) for data encryption.

Fig. 2.1: The Logical Equivalent of a VPN Connection. [5].
2. Layer Two Tunneling Protocol (L2TP) with Internet Protocol security (IPSec)
L2TP uses user-level PPP authentication methods and computer-level certificates
with IPSec for data encryption, or IPsec in tunnel mode, in which IPsec itself
provides encapsulation (for IP traffic only).

2.3 Common Uses of VPNs
The next few subsections describe the more common VPN situations in more detail.

2.3.1 Remote User Access over the Internet
VPNs Provide remote access to corporate resources over the public internet, while
maintaining privacy of information. Figure 2 shows a VPN used to connect a remote user
to a corporate intranet.

Fig. 2.2: Using a VPN to Connect a Remote Client to a Private LAN [2]
Rather than making a long distance (or 1-800) call to a corporate or outsourced Network
Access Server (NAS), the user calls a local ISP. Using the connection to the local ISP,
the VPN software creates a virtual private network between the dialup user and the
corporate VPN Server across the Internet.
2.3.2 Connecting Networks over the Internet

There are two methods for using VPNs to connect local area network at remote sites; using dedicated lines to connect a branch office to a corporate LAN and using a dial-up line to connect a branch to a corporate LAN. [2].

2.3.2.1 Using Dedicated Lines

Rather than using an expensive long-haul dedicated circuit between the branch Office and the corporate hub, both the branch office and the corporate hub routers can use a local dedicated circuit and local ISP to connect to the internet. The VPN software uses the local ISP connections and the internet to create a virtual private network between the branch office router and corporate hub router.

2.3.2.2 Using a Dial-Up

Rather than having a router at the branch office make a long distance (or 1 - 800) call to a corporate or outsourced NAS, the router at the branch office can call the local ISP. The VPN software uses the connection to the local ISP to create VPN between the branch office router and the corporate hub router across the internet.

![Diagram](image)

**Fig 2.3: Using a VPN to Connect Two Remote Sites [2]**
In both cases, the facilities that connect the branch office and corporate offices to the internet are local. The corporate hub router that acts as a VPN server must be connected to the local ISP with a dedicated line. This VPN server must be listening 24 hours a day for incoming VPN traffic.

### 2.3.3 Connecting Computers over an Internet

In some corporate internetworks the departmental data is so sensitive that the department’s LAN is physically disconnected from the rest of the corporate internetwork. Although this protects the department’s confidential information, it creates information accessibility problems for those users not physically connected to the separate LAN.

![Diagram](image1)

**Figure 2.4: Using a VPN to Connect Two Computers On The Same LAN [2]**

VPNs allow the departments LAN to be physically connected to the corporate internetwork but separated by a VPN server. The VPN server is not acting as a router between the corporate internetwork and the department LAN. By using a VPN, the network administrator can ensure that only those user on the corporate internetwork who have appropriate credentials (based on a need- to-know policy within the company) can establish a VPN with the VPN server and gain access to the protected resources of the department.[2]. Additionally, all communication across the VPN can be encrypted for data confidentiality. [2]. Those users who do not have the proper credentials cannot view the department LAN.
2.4 VPN Connections

Creating a VPN connection is very similar to establishing a point-to-point connection by using dial-up connections and demand-dial routing. There are two types of VPN connections:[2]

- Remote access VPN connection
- Router-to-router VPN connection

2.4.1 Remote Access VPN Connection

A remote access client (a single user computer) makes a remote access VPN connection that connects to a private network. The VPN server provides access to the entire network to which the VPN server is attached. The packets sent from the remote client across the VPN connection originate at the remote access client computer. The remote access client (the VPN client) authenticates itself to the remote access server (the VPN server) and, for mutual authentication, the server authenticates itself to the client [2].

2.4.2 Router-to-router VPN connection

A router makes a router-to-router VPN connection that connects two portions of a private network. The VPN server provides a routed connection to the network to which the VPN server is attached. On a router-to-router VPN connection, the packets sent from either router across the VPN connection typically do not originate at the routers [2]. The calling router (the VPN client) authenticates itself to the answering router (the VPN server) and, for mutual authentication, the answering router authenticates itself to the calling router [2].

2.5 Basic VPN Requirements

Typically, when deploying a remote networking solution, an enterprise need to facilitate controlled access to corporate resources and information. The solution must allow roaming or remote clients to connect to LAN resources, and the solution must allow remote offices to connect to each other to share resources and information (LAN-to-LAN connections). In addition, the solution must ensure the privacy and integrity of data as it
traverses the internet.[6]. The same concerns apply in the case of sensitive data traversing a corporate internetwork. Therefore, a VPN solution should provide at least user authentication, address management, data encryption, key management, and multiprotocol support. [6].

2.5.1 User Authentication

The solution must verify the user’s identity and restrict VPN access to authorized users only. It must also provide audit and accounting record to show who accessed what information and when.

2.5.2 Address Management

The solution must assign a client’s address on the private net and ensure that private addresses are kept private.

2.5.3 Data Encryption

The data carried on the public network must be rendered unreadable to unauthorized clients on the network.

2.5.4 Key Management

The solution must generate and refresh encryption keys for the client and server.

2.5.5 Multiprotocol Support

The solution must handle common protocols used in the public network. These include IP, Internet Packet Exchange (IPX), and so on.

An internet VPN solution based on the Point to Point Tunneling Protocol (PPTP) or Layer 2 Tunneling Protocol (L2TP) meets all of these basic requirements and takes advantage of the broad availability of the internet. Another solution, including the new IP Security Protocol (IPSec), meet only some of these requirements, but remain useful for specific situations.
2.6 Components of virtual private networks

A VPN connection includes the following components:[5]

VPN server: A computer that accepts VPN connections from VPN clients.
VPN client: A computer that initiates a VPN connection to a VPN server. A VPN client can be an individual computer or a router.
Tunnel: The portion of the connection in which your data is encapsulated.
VPN connection: The portion of the connection in which your data is encrypted. For typical secure VPN connections, the data is encrypted and encapsulated along the same portion of the connection.
Tunneling protocols: Are protocols that are used to manage tunnels and encapsulate private data. Data that is tunneled must also be encrypted to be a VPN connection. The Windows Server 2003 family includes the PPTP and L2TP tunneling protocols.
Tunneled data: Data that is usually sent across a private point-to-point link.
Transit internetwork: Is the shared or public network crossed by the encapsulated data. The transit internetwork is always an IP internetwork. The transit internetwork can be the Internet or a private IP-based intranet.

The following illustration shows the components of a virtual private network.

![Diagram of a virtual private network]

2.5: Component of Virtual Private Network [5]
2.7 Properties of VPN Connections

VPN connections that use PPTP and L2TP/IPSec have the following properties:[2]

- Encapsulation
- Authentication
- Data encryption

2.7.1 Encapsulation

With VPN technology, private data is encapsulated with a header that provides routing information, which allows the data to traverse the transit internetwork.[2].

2.7.2 Authentication

Authentication for VPN connections takes three different forms:

1. User-level authentication by using PPP authentication

   To establish the VPN connection, the VPN server authenticates the VPN client that is attempting the connection by using a Point-to-Point Protocol (PPP) user-level authentication method and verifies that the VPN client has the appropriate authorization. If mutual authentication is used, the VPN client also authenticates the VPN server, which provides protection against computers that are masquerading as VPN servers.[2].

2. Computer-level Authentication By Using IKE

   To establish an IPSec security association, the VPN client and the VPN server use the Internet Key Exchange (IKE) protocol to exchange either computer certificates or a preshared key. In either case, the VPN client and server authenticate each other at the computer level. [2]. Computer certificate authentication is highly recommended as it is a much stronger authentication method. Computer-level authentication is only done for L2TP/IPSec connections.
3. Data Origin Authentication And Data Integrity

To verify that the data sent on the VPN connection originated at the other end of the connection and was not modified in transit, the data contain a cryptographic checksum based on an encryption key known only to the sender and the receiver. Data origin authentication and data integrity are only available for L2TP/IPSec connections.

2.7.3 Data encryption

To ensure confidentiality of the data as it traverses the shared or public transit internetwork, the data is encrypted by the sender and decrypted by the receiver. The encryption and decryption processes depend on both the sender and the receiver using a common encryption key. Intercepted packets sent along the VPN connection in the transit internetwork are unintelligible to anyone who does not have the common encryption key. The length of the encryption key is an important security parameter. You can use computational techniques to determine the encryption key. However, such techniques require more computing power and computational time as the encryption keys get larger. Therefore, it is important to use the largest possible key size to ensure data confidentiality. [2].
Tunneling is a method of using an internetwork infrastructure to transfer data for one network over another network. The data to be transferred (or payload) can be the frames by the originating node, the tunneling protocol encapsulates the frame in an additional header. The additional header provides routing information so that the encapsulated payload can traverse the intermediate internet work.

The encapsulated packets are then routed between tunnel endpoints over the internetwork. The logical path through which the encapsulated packets travel through the internetwork is called tunnel. Once the encapsulated frames reach their destination on the internetwork, the frame is unencapsulated and forwarded to its final destination. Tunneling includes this entire process (encapsulation, transmission, and unencapsulation of packets).

Figure 3.1: Tunnelling [5]
The transit internetwork can be any internetwork the internet is a public internetwork and is the most widely known real world example. There are many examples of tunnels that are carried over corporate internetworks.

The internet provides one of the most pervasive and cost effective internetworks.

### 3.2 Tunneling Protocols

For a tunnel to be established, both the tunnel client and the tunnel server must be using the same tunneling protocol. [2]. Tunneling technology can be based on either a Layer 2 or a Layer 3 tunneling protocol. [2]. These layers correspond to the Open Systems Interconnection (OSI) Reference Model. Layer 2 protocols correspond to the data-link layer and use frames as their unit of exchange. PPTP and L2TP and Layer 2 Forwarding (L2F) are Layer 2 tunneling protocols; both encapsulate the payload in a PPP frame to be sent across an internetwork. Layer 3 protocols correspond to the Network layer, and use packets. IP-over-IP and IP Security (IPSec) Tunnel Mode are examples of Layer 3 tunneling protocols. These protocols encapsulate IP packets in an additional IP header before sending them across an IP internetwork [2].

#### 3.2.1 How Tunneling Works

Layer 3 tunneling technologies For Layer 2 tunneling technologies, such as PPTP and L2TP, a tunnel is similar to a session; both of the tunnel endpoints must agree to the tunnel and must negotiate configuration variables, such as address assignment or encryption or compression parameters. [2]. In most cases, data transferred across the tunnel is sent using a datagram based protocol [2]. A tunnel maintenance protocol is used as the mechanism to manage the tunnel. We generally assume that all of the configuration issues have been handled out of band, often by manual processes. For these protocols, there may be no tunnel maintenance phase. For Layer 2 protocols (PPTP and L2TP), however, a tunnel must be created, maintained, and then terminated [2].

Once the tunnel is established, tunneled data can be sent. The tunnel client or server uses a tunnel data transfer protocol to prepare the data for transfer. For example, when the tunnel client sends a payload to the tunnel server, the tunnel client first appends a tunnel
data transfer protocol header to the payload. The client then sends the resulting encapsulated payload across the internetwork, which routes it to tunnel server. The tunnel server accepts the packets, removes the tunnel data transfer protocol header, and forward the payload to the target network. [2]. Information sent between the tunnel server and the tunnel client behaves similarly.

3.2.2 Tunneling Protocols and the Basic Tunneling Requirements

Because they are based on the well-defined PPP protocol, Layer 2 protocols (such as PPTP and L2TP) inherit a suite of useful features. These features and their Layer3 counterparts address the basic VPN requirements as outlined below.

User Authentication: Layer 2 tunneling protocols inherit the user authentication schemes of PPP, including the EAP methods. Many Layer 3 tunneling schemes assume that the endpoints were well known (and authenticated) before the tunnel was established. [2]. An exception to this is IPSec ISAKMP negotiation, which provides mutual authentication of the tunnel endpoints. (Most IPSec implementations support computer based certificates only, rather than user certificates. As a result, any user with access to one of the endpoint computers can use the tunnel. This potential security weakness can be eliminated when IPSec is paired with a Layer 2 protocol such as L2TP.) [2].

Token card support: Using the Extensible Authentication Protocol (EAP, Layer 2 tunneling protocols can support a wide variety of authentication methods, including one-time passwords, cryptographic calculators, and smart cards. Layer 3 tunneling protocols can use similar methods: for example, IPSec defines public key certificate authentication in its ISAKMP/Oakley negotiation. [2].

Dynamic address assignment: Layer 2 tunneling supports dynamic assignment of client addresses based on the Network Control Protocol (NCP) negotiation mechanism. Generally, Layer 3 tunneling schemes assume that an address has already been assigned prior to initiation of the tunnel. Schemes for assignment of addresses in IPSec tunnel mode are currently under development and are not yet available. [2].

Data compression: Layer 2 tunneling protocols support PPP-based compression schemes. For example, the Microsoft implementations of both PPTP and
L2TP use Microsoft Point-to-Point Compression (MPPC). The IETF is investigating similar mechanisms (such as IP Compression) for the Layer 3 tunneling protocols. [2].

Data encryption: Layer 2 tunneling protocols support PPP-based data encryption mechanisms. The Microsoft implementation of PPTP supports optional use of Microsoft Point-to-Point Encryption (MPPE), based on the RSAJRC4 algorithm. Layer 3 tunneling protocols can use similar methods; for example. IPSec defines several optional data encryption methods, which are negotiated during the ISAKMP/Oakley eXchange. The Microsoft implementation of the L2TP protocol uses IPSec encryption to protect the data stream from the client to the tunnel server. [2].

Key Management: MPPE, a Layer 2 protocol, relies on the initial key generated during user authentication, and then refreshes it periodically. IPSec explicitly negotiates a common key during the ISAKMP exchange, and also refreshes it periodically. [2].

Multiprotocol support: Layer 2 tunneling supports multiple payload protocols, which makes it easy for tunneling clients to access their corporate networks using IP, IPX, NetBEUI, and so on. In contrast, Layer 3 tunneling protocols, such as IPSec tunnel mode, typically support only target networks that use the IP protocol. [2].

### 3.3 Point-to-Point Protocol (PPP)

Because the Layer 2 protocols depend heavily on the features originally specified for PPP, it is worth examining this protocol more closely. PPP was designed to send data across dial-up or dedicated point-to-point connections. PPP encapsulates IP, IPX, and NetBEUI packets within PPP frames, and then transmits the PPP encapsulated packets across a point-to-point link. PPP is used between a dial-up client and an NAS. [4].

There are four distinct phases of negotiation in a PPP dial-up session. Each of these four phases must complete successfully before the PPP connection is ready to transfer user data. [4].

#### 3.3.1 Phase 1: PPP Link Establishment

PPP uses Link Control Protocol (LCP) to establish, maintain, and end the physical connection. During the initial LCP phase, basic communication options are selected. During the link establishment phase (Phase 1), authentication protocols are selected, but they are not actually implemented until the connection authentication phase (Phase 2).
Similarly, during LCP a decision is made as to whether the two peers will negotiate the use of compression and/or encryption. [4]. The actual choice of compression and encryption algorithms and other details occurs during Phase 4.

3.3.2 Phase 2: User Authentication
In the second phase, the client PC presents the user’s credentials to the remote access server. A secure authentication scheme provides protection against replay attacks and remote client impersonation. A replay attack occurs when a third party monitors a successful connection and uses captured packets to play back the remote client’s response so that it can gain an authenticated connection. [4]. Remote client impersonation occurs when a third party takes over an authenticated connection. [4]. The intruder waits until the connection has been authenticated, and then traps the conversation parameters, disconnects the authenticated user, and takes control of the authenticated connection.

Most implementations of PPP provide limited authentication methods, typically Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and Microsoft Challenge Handshake Authentication Protocol (MSCHAP). [4].

Password Authentication Protocol (PAP): PAP is a simple, clear-text authentication scheme. The NAS requests the user name and password, and PAP returns them in clear text (unencrypted). [4]. Obviously, this authentication scheme is not secure because a third party could capture the user’s name and password and use it to get subsequent access to the NAS and all of the resources provided by the NAS. PAP provides no protection against replay attacks or remote client impersonation once the user’s password is compromised. [4].

Challenge-Handshake Authentication Protocol (CHAP): CHAP is an encrypted authentication mechanism that avoids transmission of the actual password on the connection. [4]. The NAS sends a challenge, which consists of a session ID and an arbitrary challenge string, to the remote client. The remote client must use the MD5 one-way hashing algorithm to return the user name and an encryption of the challenge, session ID, and the clientTs password. The user name is sent unhashed. CHAP is an improvement over PAP because the clear-text password is not sent over the link. Instead, the password is used to create an encrypted hash from the original challenge. [4]. The server knows the client’s clear-text password and can, therefore,
replicate the operation and compare the result to the password sent in the client’s response. CHAP protects against replay attacks by using an arbitrary challenge string for each authentication attempt. CHAP protects against remote client impersonation by unpredictably sending repeated challenges to the remote client throughout the duration of the connection.

Microsoft Challenge-Handshake Authentication Protocol (MS-CHAP): MS-CHAP is an encrypted authentication mechanism very similar to CHAP.[4]. As in CHAP; the NAS sends a challenge, which consists of a session ID and an arbitrary challenge string, to the remote client. The remote client must return the user name and an MD4 hash of the challenge string, the session ID, and the MD4-hashed password. This design, which manipulates a hash of the MD4 hash of the password, provides an additional level of security because it allows the server to store hashed passwords instead of clear-text passwords. MSCHAP also provides additional error codes, including a password expired code, and additional encrypted client-server messages that permit users to change their passwords. In MS-CHAP, both the Client and the NAS independently generate an initial key for subsequent data encryption by MPPE. Therefore, MS-CHAP authentication is required to enable MPPE-based data encryption. [4]

During phase 2 of PPP link configuration, the NAS collects the authentication data, and then validates the data against its own user database or against a central authentication database server. [4]

3.3.3 Phase 3: PPP Callback Control

The Microsoft implementation of PPP includes an optional callback control phase. This phase uses the Callback Control Protocol (CBCP) immediately after the authentication phase. If configured for callback, both the remote client and NAS disconnect after authentication. The NAS then calls the remote client back at a specified phone number. This provides an additional level of security to dial-up networking. The NAS allows connections from remote clients physically residing at specific phone numbers only. [4]

3.3.4 Phase 4: Invoking Network Layer Protocol(s)

Once the previous phases have been completed, PPP invokes the various network control protocols (NCPs) that were selected during the link establishment phase (Phase 1) to
configure protocols used by the remote client. For example, during this phase the IP control protocol (IPCP) can assign a dynamic address to the dial-in user. In the Microsoft implementation of PPP, the compression control protocol is used to negotiate both data compression (using MPPC) and data encryption (using MPPE) for because both are implemented in the same routine. [4].

3.3.5 Phase 5: Data-Transfer Phase
Once the four phases of negotiation have been completed, PPP begins to forward data to and from the two peers. Each transmitted data packet is wrapped in a PPP header which is removed by the receiving system. If data compression was selected in phase 1 and negotiated in phase 4, data is compressed before transmission. If data encryption is selected and negotiated, data is encrypted before transmission. [4].

3.4 Tunnel Types
Tunnels can be created in various ways: [5]
Voluntary tunnels: A user or client computer can issue a VPN request to configure and create a voluntary tunnel. In this case, the user’s computer is a tunnel endpoint and acts as the tunnel client.
Compulsory tunnels: A VPN-capable dial-up access server configures and creates a compulsory tunnel. With a compulsory tunnel, the user’s computer is not a tunnel endpoint. Another device, the remote access server, between the user’s computer and the tunnel server is the tunnel endpoint and acts as the tunnel client. To date, voluntary tunnels are proving to be the more popular type of tunnel. The following sections describe each of these tunnel types in greater detail.

3.4.1 Voluntary Tunneling
Voluntary tunneling occurs when a workstation or routing server uses tunneling client software to create a virtual connection to the target tunnel server. [5]. To accomplish this, the appropriate tunneling protocol must be installed on the client computer. For the protocols discussed in this paper, voluntary tunnels require an IP connection (either LAN or dial-up).
In a dial-up situation, the client must establish a dial-up connection to the internetwork before the client can set up a tunnel. This is the most common case. The best example of this is the dial-up Internet user, who must dial an ISP and obtain an Internet connection
before a tunnel over the Internet can be created. For a LAN-attached computer, the client already has a connection to the internetwork that can provide routing of encapsulated payloads to the chosen LAN tunnel server. This would be the case for a client on a corporate LAN that initiates a tunnel to reach a private or hidden subnet on that LAN (such as the Human Resources network discussed previously). It is a common misconception that VPNs require a dial-up connection. They require only IP networking. Some clients (such as home computers) use dial-up connections to the Internet to establish IP transport. This is a preliminary step in preparation for creating a tunnel and is not part of the tunnel protocol itself. [5].

3.4.2 Compulsory Tunneling

A number of vendors that sell dial-up access servers have implemented the ability to create a tunnel on behalf of a dial-up client. The computer or network device providing the tunnel for the client computer is variously known as a Front End Processor (FEP) in PPTP an L2TP Access Concentrator (LAC) in L2TP or an IP Security Gateway in IPSec. [5]. To carry out its function the FEP must have the appropriate tunneling protocol installed and must be capable of establishing the tunnel when the client computer connects.

Figure 3.2: Compulsory Tunneling. [5]
In the Internet example, the client computer places a dial-up call to a tunneling enabled NAS at the ISP. For example, a corporation may have contracted with an ISP to deploy a nationwide set of FEPs. These FEPs can establish tunnels across the Internet to a tunnel server connected to the corporation’s private network, thus consolidating calls from geographically diverse locations into a single Internet connection at the corporate network.[5].

This configuration is known as compulsory tunneling because the client is compelled to use the tunnel created by the FEP once the initial connection is made, all network traffic to and from the client is automatically sent through the tunnel. With compulsory tunneling, the client computer makes a single PPP connection. When a client dials into the NAS, a tunnel is created and all traffic is automatically routed through the tunnel. An FEP can be configured to tunnel all dial-up clients to a specific tunnel server. The FEP could also tunnel individual clients, based on the user name or destination.

Unlike the separate tunnels created for each voluntary client, a tunnel between the FEP and the tunnel server can be shared by multiple dial-up clients. When a second client dials into the access server (FEP) to reach a destination for which a tunnel already exists, there is no need to create a new instance of the tunnel between the FEP and tunnel server. Instead, the data traffic for the new client is carried over the existing tunnel. Since there can be multiple clients in a single tunnel, the tunnel is not terminated until the last user of the tunnel disconnects. [5].

3.5 Advanced Security Features

Because the Internet facilitates the creation of VPNs from anywhere, networks need strong security features to prevent unwelcome access to private networks and to protect private data as it traverses the public network. User authentication and data encryption have already been discussed. This section provides a brief look ahead to the stronger authentication and encryption capabilities that are available with EAP and IPSec.
3.5.1 Symmetric Encryption vs. Asymmetric Encryption (Private Key vs. Public Key)

Symmetric, or private-key, encryption (also known as conventional encryption) is based on a secret key that is shared by both communicating parties. [1]. The sending party uses the secret key as part of the mathematical operation to encrypt (or encipher) plain text to cipher text. The receiving party uses the same secret key to decrypt (or decipher) the cipher text to plain text. Examples of symmetric encryption schemes are the RSA RC4 algorithm (which provides the basis for Microsoft Point-to-Point Encryption (MPPE), Data Encryption Standard (DES), the International Data Encryption Algorithm (IDEA), and the Skipjack encryption technology proposed by the United States government (and implemented in the Clipper chip). [1].

Asymmetric, or public-key, encryption uses two different keys for each user: one is a private key known only to this one user; the other is a corresponding public key, which is accessible to anyone. The private and public keys are mathematically related by the encryption algorithm. One key is used for encryption and the other for decryption, depending on the nature of the communication service being implemented. [1]

In addition, public key encryption technologies allow digital signatures to be placed on messages. A digital signature uses the sender’s private key to encrypt some portion of the message. When the message is received, the receiver uses the sender’s public key to decipher the digital signature to verify the sender’s identity.

3.5.2 Certificates

With symmetric encryption, both sender and receiver have a shared secret key. The distribution of the secret key must occur (with adequate protection) prior to any encrypted communication. However, with asymmetric encryption, the sender uses a private key to encrypt or digitally sign messages, while the receiver uses a public key to decipher these messages. [1].

The public key can be freely distributed to anyone who needs to receive the encrypted or digitally signed messages. The sender needs to carefully protect the private key only. To secure the integrity of the public key, the public key is published with a certificate. A certificate (or public key certificate) is a data structure that is digitally signed by a certificate authority (CA) an authority that users of the certificate can trust. The
certificate contains a series of values, such as the certificate name and usage. Information identifying the owner of the public key, the public key itself, an expiration date, and the name of the certificate authority. The CA uses its private key to sign the certificate. If the receiver knows the public key of the certificate authority, the receiver can verify that the certificate is indeed from the trusted CA and, therefore, contains reliable information and a valid public key. Certificates can be distributed electronically (through Web access or email), on smart cards, or on floppy disks.

In summary, public key certificates provide a convenient, reliable method for verifying the identity of a sender. IPSec can optionally use this method for end-to-end authentication. Remote access servers can use public key certificates for user authentication.

3.5.3 Extensible Authentication Protocol (EAP)
Most implementations of PPP provide very limited authentication methods. EAP is an IETF-proposed extension to PPP that allows for arbitrary authentication mechanisms for the validation of a PPP connection. [1]. EAP was designed to allow the dynamic addition of authentication plug-in modules at both the client and server ends of a connection. This allows vendors to supply a new authentication scheme at any time. EAP provides the highest flexibility in authentication uniqueness and variation. [1].

3.5.4 Transaction-level Security (EAP-TLS)
E.AP-TLS has been submitted to the IETF as a draft proposal for a strong authentication method based on public-key certificates. With EAP-TLS, a client presents a user certificate to the dial-in server, and the server presents a server certificate to the client. The first provides strong user authentication to the server; the second provides assurance that the user has reached the server that he or she expected. Both systems rely on a chain of trusted authorities to verify the validity of the offered certificate. [1].

The user’s certificate could be stored on the dial-up client computer or stored in an external smart card. In either case, the certificate cannot be accessed without some form of user identification (PIN number or name-and-password exchange) between the user and the client computer. This approach meets the something-you-know- plus-something-you-have criteria recommended by most security experts.
EAP-TLS is the specific EAP method implemented in Microsoft Windows 2000. Like MS-CHAP, EAP-TLS returns an encryption key to enable subsequent data encryption by MPPE. [1]

3.5.5 IP Security (IPSec)

IP Security (IPSec) was designed by the IETF as an end-to-end mechanism for ensuring data security in IP-based communications. IPSec has been defined in a series of RFCs, notably RFCs 1825, 1826, and 1827, which define the overall architecture, an authentication header to verify data integrity, and an encapsulation security payload for both data integrity and data encryption. IPSec defines two functions that ensure confidentiality: data encryption and data integrity. As defined by the IETF. [1]. IPSec uses an authentication header (AH) to provide source authentication and integrity without encryption, and the encapsulated security payload (ESP) to provide authentication and integrity along with encryption. With IPSec. only the sender and recipient know the security key. If the authentication data is valid, the recipient knows that the communication came from the sender and that it was not changed in transit. IPSec can be envisioned as a layer below the TCP/IP stack. This layer is controlled by a security policy on each computer and a negotiated security association between the sender and receiver. The policy consists of a set of filters and associated security behaviors. If a packet’s IP address, protocol, and port number match a filter, the packet is subject to the associated security behavior. [1]

3.5.6 Negotiated Security Association

The first such packet triggers a negotiation of a security association between the sender and receiver. ISAKMP/Oakley is the standard protocol for this negotiation. During an ISAKMP/Oakley exchange, the two computers agree on authentication and data-security methods, perform mutual authentication, and then generate a shared key for subsequent data encryption.

After the security association has been established, data transmission can proceed for each computer, applying data security treatment to the packets that it transmits to the remote receiver. The treatment can simply ensure the integrity of the transmitted data, or it can encrypt it as well.
3.5.7 Authentication Header
Data integrity and data authentication for IP payloads can be provided by an authentication header located between the IP header and the transport header. The authentication header includes authentication data and a sequence number, which together are used to verify the sender, ensure that the message has not been modified in transit, and prevent a replay attack.

The IPSec authentication header provides no data encryption; clear-text messages can be sent, and the authentication header ensures that they originated from a specific user and were not modified in transit.

3.5.8 Encapsulation Security Header
For both data confidentiality and protection from third-party capture, the encapsulation security payload (ESP) provides a mechanism to encrypt the IP L payload. ESP also provides data authentication and data integrity services; therefore, ESP headers are an alternative to AH headers in IPSec packets.
CHAPTER FOUR
DESIGN OF ENCRYPTION ALGORITHM

4.1 The Algorithm Methodology

Each style of program-user interaction necessitates a different approach to software development. Developing the software tool of this dissertation follows the structured approach as a methodology. Structured approach delivers relatively complete analysis models that capture all the system’s functions and data at a desirable level of abstraction, which does not depend on the software/hardware considerations. This analysis model is transformed later into a design model, followed by the implementation phase.

To describe software tool algorithms flowcharts are used. Flowcharts used to be a popular means for describing computer algorithms. They are still used for this purpose; modern techniques such as UML activity diagrams can be considered to be extensions of the flowchart. However, their popularity decreased when interactive computer terminals and third-generation programming languages became the common tools of the trade, since algorithms can be expressed much more concisely and readably as source code in such a language. [3]. Often, pseudo-code is used, which uses the common idioms of such languages without strictly adhering to the details of a particular one. Moreover, the simplicity of flowcharts encourages using them to depict the nature and flow of the steps in a process.

4.2 Design and Implantation

The primary objective of the algorithm is to guarantee secure transmission of data among public networks. Generally, tunneling technique is used to encapsulate transmitted data and to hides the destination. The software tool adds two new features to tunneling technique to ensure the security of transmission and data as well. Basically, it first checks the connection between the source and destination of data to be transmitted. Then, it encrypts the text.

The basic ideas of encryption of transmitted data in this algorithm are:
• The source and the destination of data, to be transmitted, agree on specific encryption/decryption key.
• Such key is converted to the relative binary number.
• The plain text is then encrypted according to the converted key; thus, a new text is generated.
• The new text is then encrypted again based also on the converted key. This generates another text.
• The last text is then reversed; thus, having a final text to transmit.

Moreover, the algorithm ensures that the connection between the source and the destination is established correctly, before sending the data, should be embodied. If the connection is not established correctly, the transmitted data will not receive its destination nor any other target.

When the data encrypted and the connection established correctly, the transmitted data is send to its indented destination. The destination receives encrypted data, and to get its real contents the data should be decrypted. Thus:

• The received text is reversed.
• The reversed text is decrypted depending on the binary key, thus a new text is generated.
• The generated text is decrypted again based on the binary key to get the plain text.

4.3 Algorithm Logic

A flowchart is a schematic representation of an algorithm or a stepwise process, showing the steps as boxes of various kinds, and their order by connecting these with arrows.

Flowcharts are used in designing or documenting a process or program.
Figure 4.1: Development of Software Tool
Figure 4.2: Converting Key to Binary Number
Figure 4.3: Converting Key from Binary to Decimal
Figure 4.4: Encrypting the Plain Text – First Level
Figure 4.5: Encrypting the Plain Text – Second Level
Figure 4.6: Reversing the Second Generated Text
Figure 4.7: Decrypting the Encrypted Text
I = 1

Char = Mid(NewStr2, i, 1)

KeyChar = Mid(key, Pos2, 1)

NewStr = NewStr + Chr(Asc(Char) Xor Asc(KeyChar))

Pos2 = Pos2 + 1

Pos2 = 0

I = I + 1

I <= Len(NewStr2)

Decrypt = NewStr

End

Figure 4.8: Decrypting the Encrypted Text (Cont.)
CHAPTER FIVE
ALGORITHM IMPLEMENTATION

5.1 Development Software Tool

Tools used to develop the software tool include Winsock, Microsoft Visual Basic.

5.1.1 Winsock Tool

Winsock is a tool that is used to link two programs together; either these two programs in same computer or in different computers; either these programs in the same network or in different networks.

The purpose of this tool is to transmit data from one side to another, such purpose provides benefits to both sides, but unfortunately this tool can be used to spy on others computers.

When using this tool, the IP address of the each computer should be determined. It is also necessary to specify the port number of the destination computer.

5.1.2 Microsoft Visual Basic

For the software tool, Microsoft Visual Basic version 6.0 is used to develop the interface. Microsoft released Visual Basic in 1987. It was the first visual development tool from Microsoft. From the start, Visual Basic wasn't a hit. It wasn't until release 2.0 in 1991 that people really discovered the potential of the language, and with release 3.0 it had become the fastest-growing programming language on the market.

Visual Basic's main selling point is the ease with which it allows the user to create nice looking, graphical programs with little coding by the programmer.

Visual Basic is not only a programming language, but also a complete graphical development environment. This environment allows users with little programming experience to quickly develop useful Microsoft Windows applications which have the ability to use OLE (Object Linking and Embedding) objects. Visual Basic also has the ability to develop programs that can be used as a front-end application to a database system, serving as the user interface which collects user input and displays formatted output in a more attractive and useful form than many SQL versions are capable of.

Moreover, Databases can be created from within Microsoft Visual Basic. That is, Microsoft Access version 7.0 is included as a component of both professional and enterprise editions of Microsoft Visual Basic version 6.0.
5.2 Presentation of Algorithm Functions

The software tool is divided into two sub-programs, the first is used by the source when they want to send a message. The second is used by the destination when they want to receive a message. The two programs can run on the same computer or on different computers connected through a network.

The following section exhibits the software tool screenshots and explains the purpose and usage of each screenshot.

Figure 5.1 shows the first window appears when the software tool is loaded; it reflects the title of the software tool – Secure Transmission of Data Through Public Networks - and its logo. This window appears on the monitor for three seconds as startup object.
Figure 5.2 exhibits the main window of the software tool which contains two menus: File and Help to allow the user of the system to access other windows and to obtain information about how to use this software tool. It contains also two command buttons: one to move the user to another window and the other to terminate the application.

![Main Window](image)

**Figure 5.2: Main Window**

Figure 5.3 exhibits the window which is designed to determine the IP address and port number of the remote computer – destination. If any of the IP addresses or port number is not specified correctly, an error message will appear. The reason behind checking the correctness of the IP address and port number is that, both of them are essential to build up the connection between the two computers when using Winsock tool. This window contains also two command buttons, the first is to check the validity of the IP address and the port number of the destination, and the other is to end the current window and move user to the main window.
Figure 5.4 shows the window which plays the vital role in the software tool. It allows the user to enter the key used for encryption which is converted automatically to a series of binary numbers.

It also allows the user to edit the plain-text and it shows also the text after being encrypted; i.e. it shows the text in the shape of which it will be transmitted.

This window also provides the connection status between the two computers – source and destination. The connection can take one of ten statuses: Connection Closed, Connection Open, Waiting, Suspended, Data Processing, Data Processing End, Connection Request, Connected, Connect Ended by Other Party and Error.

The data will be transmitted only when the connection status is "Connected" and the destination accepts to have transmitted data.
Figure 5.4: Encryption and Transmission 1

Figure 5.5 shows the same window in figure 5.4 "Encryption and Transmission 1", but here it shows the plain-text before and after encryption. It also exhibits the connection status which is "Connected". Thus the message is ready to be transmitted.
The above three figures show the windows of the first program that used by the source of the message.

Figure 5.6 exhibits the window that contains the key of decryption and its decimal form and the same key after being converted to its binary form. It also shows the received text before and after decryption.
Figure 5.6: Receiving and Decrypting 1

Figure 5.7 shows the same window in figure 5.6 "Receiving and Decrypting 1"; It exhibits the message when received and the key used to decrypt the message and the original message after being decrypted.

Figure 5.7: Receiving and Decrypting 2
5.3 Algorithm Testing:

As test for the encryption algorithm; a plain text has been typed and encrypted by the source. Then the encrypted text is transmitted. It is delivered to the indented destination when only he agrees to receive it. When received, it appears in the encrypted format. Then, the process of decryption performed by the destination generates exactly the plain text typed by the source.

5.4 Value of the Software Tool

Transmitting data among private networks is too expensive and not available for most of people. Thus, a cheaper media to send and receive data is needed. The solution comes through the using the public networks as virtual private networks. However, public networks are not secure to transmit data. The software tool tries to solve this problem by embodying the tunneling technology which encapsulates transmitted data, moreover, encrypts the transmitted data in such a way that even if intruders access the data they will not be able to know its real contents. This software tools provides the following feathers:

- It uses a very cheap, common and available communication media.
- It hides the real destination of the transmitted data.
- It uses the IP addresses and port numbers to grantee that data will be received by the intended destination.
- It encrypts the transmitted data using sophisticated encryption algorithms.
- Allows only the intended party to decrypt the transmitted data.
- Allows the intended destination to ensure that the entire message is received or not.
- No body rather than the intended destination can interrupt the transmitted data; i.e either the intended destination gets the transmitted data or delivery operation will be canceled.
CHAPTER FIVE
ALGORITHM IMPLEMENTATION

5.1 Development Software Tool
Tools used to develop the software tool include Winsock, Microsoft Visual Basic.

5.1.1 Winsock Tool
Winsock is a tool that is used to link two programs together; either these two programs in same computer or in different computers; either these programs in the same network or in different networks.

The purpose of this tool is to transmit data from one side to another, such purpose provides benefits to both sides, but unfortunately this tool can be used to spy on others computers.

When using this tool, the IP address of each computer should be determined. It is also necessary to specify the port number of the destination computer.

5.1.2 Microsoft Visual Basic
For the software tool, Microsoft Visual Basic version 6.0 is used to develop the interface. Microsoft released Visual Basic in 1987. It was the first visual development tool from Microsoft. From the start, Visual Basic wasn't a hit. It wasn't until release 2.0 in 1991 that people really discovered the potential of the language, and with release 3.0 it had become the fastest-growing programming language on the market.

Visual Basic's main selling point is the ease with which it allows the user to create nice looking, graphical programs with little coding by the programmer.

Visual Basic is not only a programming language, but also a complete graphical development environment. This environment allows users with little programming experience to quickly develop useful Microsoft Windows applications which have the ability to use OLE (Object Linking and Embedding) objects. Visual Basic also has the ability to develop programs that can be used as a front-end application to a database system, serving as the user interface which collects user input and displays formatted output in a more attractive and useful form than many SQL versions are capable of.

Moreover, Databases can be created from within Microsoft Visual Basic. That is, Microsoft Access version 7.0 is included as a component of both professional and enterprise editions of Microsoft Visual Basic version 6.0.
5.2 Presentation of Algorithm Functions

The software tool is divided to two sub-programs, the first is used by the source when want to send a message. The second is used by the destination when want to receive a message. The two programs can run in same computer on different computers connected through network.

The following section exhibits the software tool screenshots and explains the purpose and usage of each screenshot.

Figure 5.1 shows the first window appears when the software tool is loaded; it reflects the title of the software tool – Secure Transmission of Data Through Public Networks - and its logo. This window appears on the monitor for three seconds as startup object.

Figure 5.1: Startup Window
Figure 5.2 exhibits the main window of the software tool which contains two menus: File and Help to allow the user of the system to access other windows and to obtain information about how to use this software tool. It contains also two command buttons: one to move the user to another window and the other to terminate the application.

![Main Window](image)

**Figure 5.2: Main Window**

Figure 5.3 exhibits the window which is designed to determine the IP address and port number of the remote computer – destination. If any of the IP addresses or port number is not specified correctly, an error message will appear. The reason behind checking the correctness of the IP address and port number is that, both of them are essential to build up the connection between the two computers when using Winsock tool. This window contains also two command buttons, the first is to check the validity of the IP address and the port number of the destination, and the other is to end the current window and move user to the main window.
Figure 5.4 shows the window which plays the vital role in the software tool. It allows the user to enter the key used for encryption which is converted automatically to a series of binary numbers.

It also allows the user to edit the plain-text and it shows also the text after being encrypted; i.e it shows the text in the shape of which it will be transmitted.

This window also provides the connection status between the two computers – source and destination. The connection can take one of ten statuses:
- Connection Closed
- Connection Open
- Waiting
- Suspended
- Data Processing
- Data Processing End
- Connection Request
- Connected
- Connect Ended by Other Party
- Error

The data will be transmitted only when the connection status is "Connected" and the destination accepts to have transmitted data.
Figure 5.5 shows the same window in figure 5.4 "Encryption and Transmission 1", but here it shows the plain-text before and after encryption. It also exhibits the connection status which is "Connected". Thus the message is ready to be transmitted.
The above three figures show the windows of the first program that used by the source of the message.

Figure 5.6 exhibits the window that contains the key of decryption and its decimal form and the same key after being converted to its binary form. It also shows the received text before and after decryption.
Figure 5.6: Receiving and Decrypting 1

Figure 5.7 shows the same window in figure 5.6 "Receiving and Decrypting 1"; It exhibits the message when received and the key used to decrypt the message and the original message after being decrypted.

Figure 5.7: Receiving and Decrypting 2
5.3 Algorithm Testing:

As test for the encryption algorithm; a plain text has been typed and encrypted by the source. Then the encrypted text is transmitted. It is delivered to the indented destination when only he agrees to receive it. When received, it appears in the encrypted format. Then, the process of decryption performed by the destination generates exactly the plain text typed by the source.

5.4 Value of the Software Tool

Transmitting data among private networks is too expensive and not available for most of people. Thus, a cheaper media to send and receive data is needed. The solution comes through the using the public networks as virtual private networks. However, public networks are not secure to transmit data. The software tool tries to solve this problem by embodying the tunneling technology which encapsulates transmitted data, moreover, encrypts the transmitted data in such a way that even if intruders access the data they will not be able to know its real contents. This software tools provides the following feathers:

- It uses a very cheap, common and available communication media.
- It hides the real destination of the transmitted data.
- It uses the IP addresses and port numbers to grantee that data will be received by the intended destination.
- It encrypts the transmitted data using sophisticated encryption algorithms.
- Allows only the intended party to decrypt the transmitted data.
- Allows the intended destination to ensure that the entire message is received or not.
- No body rather than the intended destination can interrupt the transmitted data; i.e either the intended destination gets the transmitted data or delivery operation will be canceled.
CHAPTER SIX

Conclusion and Recommendations

6.1 Conclusion

Transmitting data among networks is a challenging task; that is; either to send data through a private secure network which is too expensive and not available for each one; or to transmit data through a public network which is not secure as any intruder can access and modify the transmitted data or even prevent the send from reaching its destination.

This dissertation aims to get the benefits of public networks as a cheap tool that allows transmitting data through different points in the network and then tries to secure the transmitted data.

To achieve its goal the dissertation embodies tunneling technique concepts; that are: it used Winsock tool to establish connection between source and destination and to ensure data will be transmitted only if the connection is established correctly and only the intended destination will receive the transmitted data.

It also adds a very strong feature that is not available in tunneling technique, which is; it uses very complex algorithms to encrypt transmitted data. The application ensures that if intruders access the transmitted data, they will not be able to get its real contents.

6.2 Recommendations for Future Works

The software tool does not provide a mean that guarantees transmitted data delivered completely. Also the software tool developed in this dissertation deals only with text data which is a real lack.

Future dissertations should solve the above problems by inventing and implementing an algorithm to find out that transmitted data is completely delivered; and by trying to find solutions to encrypt data types other than text data. The users of the software tool should not be allowed to enter an encryption key with length less than ten characters. It will be much better if the plain text to be encrypted is converted to Hexadecimal rather than to the Binary system.