



**NEAR EAST UNIVERSITY**

**GRADUATE SCHOOL OF APPLIED  
AND SOCIAL SCIENCES**

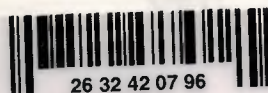
**ANALYSIS OF WAP SECURITY AND  
CRYPTOGRAPHY**

**Wisam Abu Rajab**

**Master Thesis**

**Department of Computer Engineering**

**Nicosia-2003**



26 32 42 07 96

**Wisam AbuRajab: Analysis of WAP Security and Cryptography**



**Approval of the Graduate School of Applied and  
Social Sciences**

**Prof. Dr. Fakhraddin Mamedov**  
**Director**

**We certify this thesis is satisfactory for the award of the  
Degree of Master of Science in Computer Engineering**

**Examining Committee in charge:**

**Assoc. Prof. Dr. Rahib Abiyev, Committee Chairman, Computer  
Engineering Department, NEU**

**Assist. Prof. Dr. İlham Huseynov, Committee Member, Computer  
Information Systems, NEU**

**Assist. Prof. Dr. Doğan Haktanir, Committee Member, Computer  
Engineering Department, NEU**

**Prof. Dr. Fahrettin Mamedov, Supervisor, Dean of Engineering  
Department and Vice president of  
NEU**

**DEPARTMENT OF COMPUTER ENGINEERING**  
**DEPARTMENTAL DECISION**

**Date: 30/06/2003**

**Subject:** Completion of M.Sc. Thesis

**Participants:** Assoc. Prof. Rahib Abiyev, Assoc. Prof. Dr. Dogan Haktanir, Assist. Prof. Dr. Ilham Huseynov, Bader Bader, Amjad Hammouda, Ali Abukhorj and Wisam Aburajab.

**DECISION**

We certify that the student whose number and name are given below, has fulfilled all the requirements for a M .S. degree in Computer Engineering.

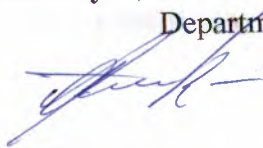
**CGPA**

971009

Wisam AbuRajab

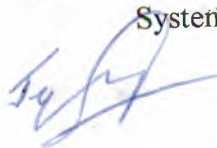
3.50

Assoc. Prof. Dr. Rahib Abiyev, Committee Chairman, Computer Engineering  
Department, NEU

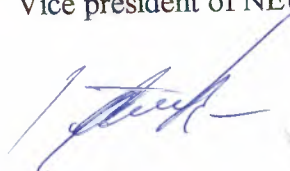


Assist. Prof. Dr. Doğan Haktanir, Committee Member, Computer Engineering  
Department, NEU

Assist. Prof. Dr. Ilham Huseynov, Committee Member, Computer Information  
Systems, NEU



Prof. Dr. Fahrettin Mamedov, Dean of Engineering Department and  
Vice president of NEU



Chairman of Department  
Assoc. Prof. Dr. Doğan İbrahim



NEU

**JURY REPORT****DEPARTMENT OF  
COMPUTER ENGINEERING**

Academic Year: 2002-2003

**STUDENT INFORMATION**

<b>Full Name</b>	Wisam AbuRajab		
<b>Undergraduate degree</b>	BSc.	<b>Date Received</b>	<b>Spring</b> 1999-2000
<b>Institution</b>	Near East University	<b>CGPA</b>	2.09

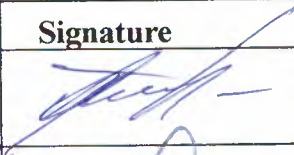
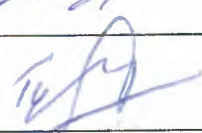
**THESIS**

<b>Title</b>	Analysis of WAP security and Cryptography		
<b>Description</b>	The aim of this thesis is to analyze WAP security and provide high security level in the WAP, which can be applied by encryption/ decryption the client/server connection.		
<b>Supervisor</b>	Prof.Dr.Fahrettin Mamedov	<b>Department</b>	Electrical & Electronic


**JURY'S DECISION**

<p>The jury has decided to accept / <del>reject</del> the student's thesis. The decision was taken <del>unanimously</del> / by majority.</p>
--

**JURY MEMBERS**

<b>Number Attending</b>	3	<b>Date</b>	30/06/2003
<b>Name</b>		<b>Signature</b>	
Assoc. Prof. Dr. Rahib Abiyev, Member Chairman of the jury			
Assist. Prof. Dr. Ilham Huseynov, Member			
Assist. Prof. Dr. Dogan Haktanir, Member			

**APPROVALS**

<b>Date</b> 30/06/2003	 <b>Chairman of Department</b> Assoc. Prof. Dr. Doğan İbrahim
---------------------------	--



## ACKNOWLEDGEMENTS

Ode to my Family and to all Martyrs in my Sweet homeland Palestine.

Especially to my Aunt and to my Mother (God rest you in Peace)

Thank you Father.

I would like to thank my supervisor Prof. Dr. Fahrettin Mamedov for his help.

Special thanks to Dr. Adnan Khashman, Dr. Dogan Haktanir and to Dr. Rahib Abiyev.

Thanks for you all dear teachers.

Thanks to my advisor Prof. Dr. Senol Bektas and to Mr. Tayseer Alshanableh.

Special thanks to the man who was always beside me thank you Bader Bader

Thanks Mohammad Bader

## List of Abbreviations

- API:** Application Programming Interface.
- BNF:** Backus-Naur Form.
- CA:** Certification Authority.
- CDMA:** Code Division Multiple Access
- CDPD:** Cellular Digital Packet Data
- CSD:** Circuit Switched Data
- CTLA:** Cellular Telecommunications Industry Association
- DECK:** A series of WML cards. A WML deck is also an XML document
- DEVICE:** Network entity capable of sending and receiving packets of information and has a unique device address.
- DNS:** Domain Name Server
- DTD:** Document Type Definitions
- DUT:** Device under Test
- ECMA:** European Computer Manufacturers Association
- ELEMENT:** An element specify the markup and structural information in a WML deck. Some elements contain a start and end tag such as the `<p>` and `</p>` tag, others are single elements such as the `<br/>` tag.
- ETSI:** European Telecommunication Standardization Institute
- GPRS:** General Packet Radio Service
- GSM:** Global System for Mobile Communication
- HDML:** Handheld Device Markup Language Invented by phone.com, predecessor to WML
- HDTP:** Handheld Device Transport Protocol
- HSCSD:** High Speed Circuit Switched Data
- HTML:** Hypertext Markup Language
- IANA:** Internet Assigned Number Authority
- ICS:** Implementation Conformance Statement
- IDEN:** Integrated Digital Enhanced Network
- IETF:** Internet Engineering Task Force
- IIS:** Internet Information Server
- IMC:** Internet Mail Consortium

**i-Mode:** Packet based information service for mobile phones from NTT DoCoMo (Japan). First to provide Web browsing from cell phones.

**ISO:** Internet Mail Consortium ISO International Standards Organization

**ISP:** Internet Service Provider

**ITTP:** Intelligent Terminal Transfer Protocol

**IWF:** Interworking Function

**LA:** License Agreement

**LSB:** Least Significant Bits

**MExE:** Mobile Station Execution Environment

**MMI:** Man Machine Interface

**MMM:** Mobil Media Mode

**MSB:** Most Significant Bits

**MSC:** Mobile Switch Center

**OEM:** Original Equipment Manufacturer

**OSI:** Open System Interconnection

**OASIS:** Organization for the Advancement of Structured Information Standards

**PDA:** Personal digital Assistant

**PDC:** Personal Digital Cellular

**PHS:** Pocket Handy Phone System

**PPP:** Point to Point Protocol

**PR:** Problem Report

**PSTN:** Public Switched Telephone Network

**RFC:** Request For Comments

**SCR:** Static Conformance Requirements

**SDML:** Signed Document Markup Language

**SGML:** Standardized Generalized Markup Language

**SMS:** Short Message Service

**SSL:** Secure Socket Layer

**TeleVAS:** Telephony Value Added Services

**TSD:** Test Suite Deficiency

**TSMA:** Test Suite Maintenance Authority

**UCS-4:** Universal Character Set 4 byte [ISO10646]

**UMTS:** Universal Mobile Telecommunications System



**URL:** Universal Resource Locator  
**USSD:** Unstructured Supplementary Services Data  
**UTF-8:** Transformation Format 8 [ISO10646]  
**W3C:** World Wide Web Consortium  
**WAE:** Wireless Application Environment  
**WAP:** Wireless Application Protocol  
**WBMP:** WAP Bitmap  
**WBXML:** WAP Binary Extensible Markup Language  
**WCMP:** Wireless Control Message Protocol  
**WCR:** WAP Certification Report  
**WML:** Wireless Markup Language  
**WMLScript:** A scripting language used to program the mobile device.  
**WSL:** Wireless Session Layer  
**WSP:** Wireless Session Protocol  
**WTA:** Wireless Telephony Applications  
**WTLS:** Wireless Transport Layer Security  
**WTP:** Wireless Transfer Protocol  
**WWW:** World Wide Web  
**XHTML:** Extended Hypertext Markup Language  
**XML:** Extensible Markup Language

## ABSTRACT

Wireless networks are taking place instead of the wired networks recently, because of many reasons among of which is the cost and reliability. Sometimes, it seems to be impossible to connect your work with a wired network. This spreading of the wireless networks does not mean that this type is totally safe so there are some problems related to the security of the data. The widespread reliance on networking in business and the meteoric growth of the Internet and online services are strong testimonies to the benefits of shared data and shared resources. With wireless LANs, users can access shared information without looking for a place to plug in, and network managers can set up or augment networks without installing or moving wires.

Hundreds of millions of Internet users around the world have become accustomed to an Internet beyond boundaries. One site flows to the next, a jungle of software, protocols, media and people connecting, signal, noise, mixing, evolving, together. Internet security risks aren't to be taken lightly, but they can all be managed and minimized just like other security risks in business.

Wireless Application Protocol (WAP) is a controversial subject, since it is good in some manners but also it has some problems related to security, same as in wireless networks.

The WAP WTLS (wireless transport layer security) was designed to provide authentication, data privacy and data integrity to the WAP. It is expected that the protocol will be fielded of million devices in the near future. Although the WTLS protocol was modeled after studied TLS well, I had been identified some security problems inside it, which means that the protocol should be revised seriously and some radical actions must be taken related to this problem.

An SSL based programs had been programmed on visual basic tested the security of the WAP trying to solve the threads of the WAP security. After compiling and testing the two programs, it was so clear that these two programs can be used for many aims and can manage some problems in WAP security.

## TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT</b>	i
<b>LIST OF ABBREVIATIONS</b>	ii
<b>ABSTRACT</b>	vi
<b>CONTENTS</b>	vii
<b>INTRODUCTION</b>	1
<b>1. WIRELESS APPLICATION PROTOCOL</b>	3
1.1. Overview	3
1.2 . History	5
1.2.1. Formation	5
1.2.2. WAP's goals	5
1.3 . Technology	7
1.4 . WAP development issues	12
1.4.1. <i>Push not supported</i>	13
1.4.2. Wireless telephony application delayed	13
1.4.3. Lack of cookies for session management	14
1.4.4. Premature Encryption endpoint	14
1.4.5. Small downloadable unit size	14
1.5 . WAP developer's toolkits	15
1.6 . WAP client and gateways	15
1.7. Applications	16
1.8 . WAP and the WEB	18
1.8.1. WAP and Web Heredity's	19
1.8.2. Specifications of how it works	20
1.8.3. Communications between client and server	21
1.8.4. The wireless markup language WML	22
1.8.5. Additional intelligence via WMLScript	23
1.8.6. The business case	24
1.9 . Summary	25
<b>2. INTERNET SECURITY AND WIRELESS LANS</b>	26
2.1 . Overview	26
2.2 . Internet Security Risks and Remedies	26



2.2.1. Hackers	27
2.2.2. Industrial Espionage	27
2.2.3. Hi-tech Criminals	27
2.2.4. Viruses	27
2.3. Business Communication over the Internet: Risks and Remedies	27
2.3.1. E-mail Security	28
2.3.2. The Risks	28
2.4. Security Concerns About Websites	31
2.4.1. Websites Used Only For Advertising	31
2.4.2. Websites Used To Make Sales and Get Paid	31
2.5. Special Case	32
2.5.1. Situation in Australia	32
2.5.2. Processing the Credit Cards	32
2.5.3. Internet Banking	33
2.5.4. SET (Secure Electronic Transactions)	33
2.6. Wireless LANS	33
2.6.1. Wireless	34
2.6.2. Narrowband Technology	34
2.6.3. Spread Spectrum Technology	35
2.6.4. Frequency-Hopping Spread Spectrum Technology	35
2.6.5. Direct-Sequence Spread Spectrum Technology	35
2.6.6. Infrared Technology	35
2.6.7. Wireless LANs Work	36
2.6.8. Wireless LAN Configurations	37
2.6.9. Security	38
2.6.10. Safety	38
2.7. Making a Secure Wireless Transaction	38
2.7. Summary	41
<b>3. WAP SECURITY</b>	42
3.1. Overview	42
3.2. Background	42
3.3. What security is about	43
3.3.1. The importance of security	43

3.3.2. The role of security	43
3.3.3. The basic issues	44
3.3.4. Concepts	45
3.3.5. Protocol Stacks	45
3.3.6. Encryption	48
3.3.7. Certificates	50
3.3.8. WTLS	51
3.4. Communication Models	53
3.4.1. Internet communication model	53
3.4.2. Wireless communication model	56
3.5. WAP security issues	57
3.5.1. The gateway	57
3.5.2. User versus device	63
3.6. Future	64
3.6.1. WTLS	64
3.6.2. End to end security	64
3.6.3. WIM	64
3.7. Summary	65
<b>4. SECURITY IN THE WTLS</b>	66
4.1. Overview	66
4.2. Introduction	66
4.3. Data Communication Security	67
4.3.1. Privacy	68
4.3.2. Authentication	68
4.3.3. Integrity	69
4.4. Wireless Transport Layer Security	69
4.4.1. Specification	70
4.4.2. WTLS Internal Architecture	71
4.4.3. Authentication	76
4.4.4. Key Exchange	78
4.4.5. Privacy	79
4.4.6. Integrity	80
4.4.7. Secure State	81

4.4.8. Evaluation of the WTLS	82
4.4.9. Reasons for Defects	86
4.4.10. Known Security Holes	87
4.4.11. The Accepted Level of Security	90
4.5. Summary	93
<b>5. CRYPTOGRAPHY AND ITS ALGORITHMS</b>	94
5.1. Overview	94
5.2. Cryptography	94
5.2.1. Cryptanalysis	94
5.2.2. Classical Encryption Techniques	96
5.2.3. Public-Key Cryptography	98
5.2.4. The RSA Algorithm	100
5.3. The Client/Server encryption/decryption program	104
5.3.1. The Aim of The Program	104
5.3.2 The Details of the Program	105
5.4. Summary	109
<b>CONCLUSION</b>	110
<b>REFERENCES</b>	112
<b>APPENDIX-A</b>	A-1
<b>APPENDIX-B</b>	B-1
<b>APPENDIX-C</b>	C-1



## INTRODUCTION

The huge growth of the wireless mobile services urges the demand for the end-to-end secure connections. The security layer in the WAP [1] is the WTLS [1] (wireless transport layer security). It is aim to provide authentication, data integrity, and data privacy for applications in cellular phones and other small wireless terminals. It is based on the TLS and SSL protocols [6], but with a number of changes that had been carried by the WAP Forum to meet the new needs. While designing the WTLS the requirements of the mobile networks have been taken into account; datagram connection, cryptography exporting restrictions, and low bandwidth, limited processing power and memory capacity, have all been considered. WTLS is expected to be fielded with millions of devices in few years [1].

The aim of this thesis is to investigate the wireless application protocol security, its advantages and disadvantages. In order to do so the security of the WTLS should been analyzed. Background information was given like the concept of data security. The common security terms like authentication, privacy, and integrity were explained. Also WTLS was presented which was the most important part of this research. The WTLS main problems were mentioned and discussed and impacts were evaluated.

WTLS was found to be a good security solution, but it needs to be revised.

Improvements must be done to the protocol as soon as possible. This means that major changes should be taken into action. To prove a sufficient security, the supported algorithms must be combined in an appropriate way. The anonymous authentication should not be allowed and the null ciphers should be denied. If all the defined security holes will be fixed, then the WTLS provides a sufficient security level, otherwise a radical decision must be taken into action towards the WTLS and its work.

Thesis consists of five chapters, introduction and conclusion.

In the first chapter, an overview of the WAP had been shown. The history, the formation, the services and the issues of the WAP also had been mentioned. The WAP

developers' toolkit, the WAP gateways, and some applications had been introduced. Then the relation between the WAP and WEB had been shown and discussed.

In the second chapter, the wireless networks security, the internet security and their relation had been discussed. The security holes in the wireless networks and their remedies had been also illustrated. The internet security concerns and their attackers had been mentioned analyzed, discussed and then their remedies were supposed.

In the third chapter, WAP security had been introduced and analyzed. The importance of security, the protocol stacks and the communications had been mentioned. The WAP security issues like the gateway and the user versus device had been analyzed. Then a look to the future had been mentioned. Introduction to end-to-end security, WIM and WTLS had also been given.

In the fourth chapter, the WTLS security had been analyzed. It started with an overview of the WTLS then the data communication security had been discussed. After all, the WTLS specifications, architecture, security level and security problems had been analyzed. Finally, is the estimation of the WTLS Security is discussed and analyzed to reach a point of view of whether it is applicable and acceptable or not.

In the fifth chapter, again the cryptographic logarithms had been investigated and discussed more precisely than before. Then the programs of server/client encryption/decryption and an RSA calculator had been introduced and implemented. The programs were successfully tested and captured the input and output of each one as figures. Finally, the source codes of these programs are attached in the appendices A, B, and C in the end of the research.



# 1. WIRELESS APPLICATION PROTOCOL

## 1.1 Overview

The Wireless Application Protocol (WAP) is an open, global standard that empowers mobile users with wireless devices to easily access and interact with information and services instantly. [1]

WAP is simply a set of standards that allows developers of applications and mobile devices to make compatible products. The WAP standards were developed by a mobile industry funded group called the WAP Forum and are based on common web standards like IP and XML to make sure it integrates well with current technology. It also makes the development of WAP based pages. [1]

The Wireless Application Protocol (WAP) is a hot topic that has been widely hyped in the mobile industry and outside of it. WAP is simply a protocol- a standardized way that a mobile phone talks to a server installed in the mobile phone network. It is amazing how in just few months, it has become imperative for all Information Technology companies in Nordic countries for example and beyond to have a WAP division. Many advertising agencies and "dot.coms" have announced WAP services. [1]

WAP provides a standardized way of linking the Internet to mobile phones; its founder members include the major wireless vendors of Nokia, Ericsson and Motorola, plus a newcomer Phone.com. By April 2000, the WAP Forum had over 350 member companies. [1] Mobile information services, a key application for WAP, have not been as successful as many network operators expected. WAP is seen as a way to rectify this situation. On the other hand WAP also has its detractors and controversies, because it is very difficult to configure WAP phones for new WAP services, with 20 or so different parameters needing to be entered to gain access to a WAP service. Compared with the installed base of Short Message Service (SMS) compliant phones, the relative number of handsets supporting WAP is tiny. WAP is a protocol that runs on top of an underlying bearer. None of the existing GSM bearers for WAP- the Short Message Service (SMS), Unstructured Supplementary Services Data (USSD) and Circuit Switched Data (CSD) are optimized for WAP. [1] The WAP standard is incomplete, with key elements such as Push (proactive sending of information to mobile devices)



and wireless telephony (updating address reports and the like) included in the WAP 1.2, standardized in late 1999 and implemented in the spring of 2000. [1] Other protocols such as SIM Application Toolkit and Mobile Station Application Execution Environment (MexE) are respectively already widely supported or designed to supercede WAP. WAP services are expected to be expensive to use since the tendency is to be on-line for a long Circuit Switched Data (CSD) call as features such as interactivity and selection of more information are used by the end user. Without specific tariff initiatives, there are likely to be some surprised WAP users when they see their mobile phone bill for the first time after starting using WAP. [1]

The definition of the WAP programming model, which is based on the WWW programming model, ensures existing tools like web servers etc. can be used. A markup language based on XML called the Wireless Markup Language (WML) and a compact version of JavaScript called WMLscript, which is basically JS without the support for mouse or keyboard input devices.

Specifications define how the 'microbrowser' should present WAP markup. The microbrowser is a scaled down version of a web browser and resides on the mobile. A framework for Wireless Telephony Applications (WTA) that allows access to telephony functionality like placing a call by clicking a link, Since the WAP standard was defined with the mobile device in mind it offers some nice advantages to simply clipping web content to make it fit for mobile devices. WAP is much optimized in size using a few tricks like translating the text headers in binary code and simplifying protocols to make sure it works well in the low bandwidth wireless environment. It defines a model for a microbrowser that has a very small footprint to make it work on low memory devices like mobile phones. It implements some new (voice based) functionality that isn't available in normal web standards. And the fact that the markup language is based on XML, which is a W3C standard, pretty much guarantees the continuing support of the web community. WML's XML roots also make it possible to do automatic content transformation, which allows content formatted in an XML markup language like XSL (eXtensible Style Language) to be automatically translated to a related language like HTML for web browsers or WML for microbrowsers. [1, 15]

HDML which is developed by Phone.com and is the predecessor of WML, it is not a very widespread markup language, although some microbrowsers still support HDML.

## **1.2. History**

### **1.2.1. Formation**

Motorola, Nokia, Ericsson and the US software company Phone.com (formerly Unwired Planet) were the initial partners that teamed up in mid 1997 to develop and deploy the Wireless Application Protocol (WAP). WAP is an attempt to define the standard for how content from the Internet is filtered for mobile communications. Content is now readily available on the Internet and WAP was designed as the (rather than one) way of making it easily available on mobile terminals. [1]

The WAP Forum was formed after a US network operator Omnipoint issued a tender for the supply of mobile information services in early 1997. It received several responses from different suppliers using proprietary techniques for delivering the information such as Smart Messaging from Nokia and HDML from Phone.com (then called Unwired Planet). Omnipoint informed the tender responders that it would not accept a proprietary approach and recommended that various vendors get together to explore defining a common standard. Finally, there was not a great deal of difference between the different approaches, which could be combined and extended to form a powerful standard. These events were the initial stimulus behind the development of the Wireless Application Protocol, with Ericsson and Motorola joining Nokia and Unwired Planet as the founder members of the WAP Forum. [1]

### **1.2.2. WAP's Goals**

WAP had been designed to meet the following:

- Independent of wireless network standard.
- Open to all.
- Proposed to the appropriate standards bodies.
- Scalable across transport options.
- Scalable across device types.
- Extensible over time to new networks and transports.

As part of the Forum's goals, WAP will also be accessible to (but not limited to) the following:

GSM-900, GSM-1800, GSM-1900

CDMA IS-95

TDMA IS-136

3G systems - IMT-2000, UMTS, W-CDMA, Wideband IS-95

WAP defines a communications protocol as well as an application environment. In essence, it is a standardized technology for cross-platform, distributed computing.

Sound similar to the World Wide Web, in that WAP is very similar to the combination of HTML and HTTP except that it adds in one very important feature: optimization for low-bandwidth, low-memory, and low-display capability environments. These types of environments include PDAs, wireless phones, pagers, and virtually any other communications device. [26]

Some critics and second-guessers have pondered the need for a technology such as WAP in the marketplace. With the widespread proliferation of HTML, is yet another markup language really required? As we've discussed here, in a word, YES! WAP's use of the deck of cards "pattern" and use of binary file distribution meshes well with the display size and bandwidth constraints of typical wireless devices. Scripting support gives us support for client-side user validation and interaction with the portable device again helping to eliminate round trips to remote servers. WAP is a young technology that is certain to mature as the wireless data industry as a whole matures; however, even as it exists today, it can be used as an extremely powerful tool in every software developer's toolbox. [1]

The Wireless Application Protocol takes a client server approach. It incorporates a relatively simple microbrowser into the mobile phone, requiring only limited resources on the mobile phone. This makes WAP suitable for thin clients and early smart phones. WAP puts the intelligence in the WAP Gateways whilst adding just a microbrowser to the mobile phones themselves. Microbrowser-based services and applications reside temporarily on servers, not permanently in phones. The philosophy behind Wireless Application Protocol's approach is to utilize as few resources as possible on the



handheld device and compensate for the constraints of the device by enriching the functionality of the network. [8]

The Wireless Application Protocol is designed for use with any mobile phone from those with a one line display to a smart phone and any existing or planned wireless service such as the Short Message Service, Circuit Switched Data, Unstructured Supplementary Services Data (USSD) and General Packet Radio Service (GPRS). Indeed, the importance of WAP can be found in the fact that it provides an evolutionary path for application developers and network operators to offer their services on different network types, bearers and terminal capabilities. [1]

The design of the WAP standard separates the application elements from the bearer being used. This helps in the migration of some applications from SMS or Circuit Switched Data to GPRS for example.

### 1.3. Technology

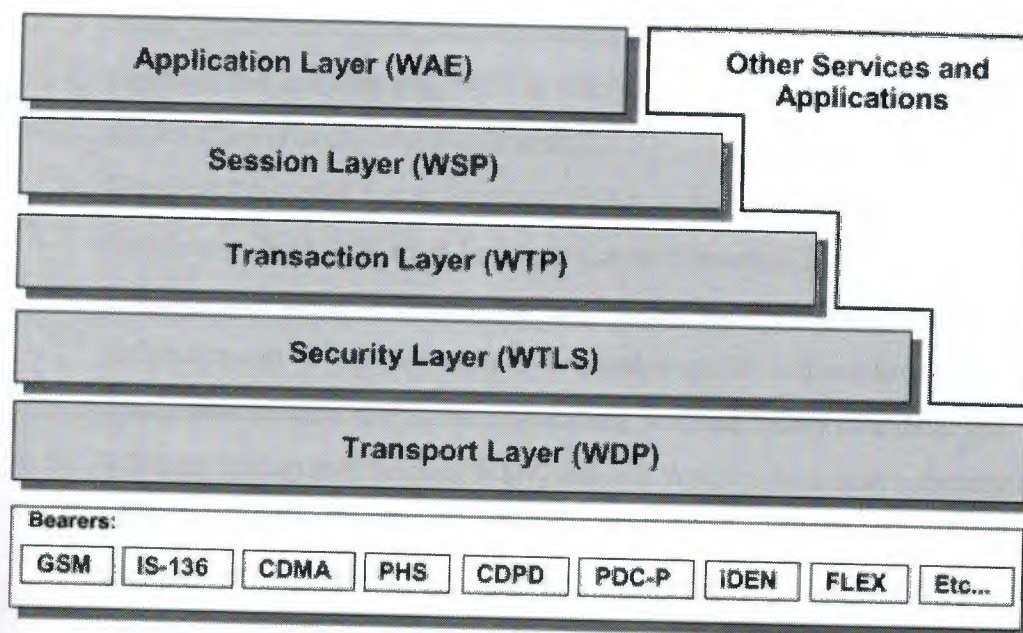
The Wireless Application Protocol embraces and extends the previously conceived and developed wireless data protocols. Phone.com created a version of the standard HTML (HyperText Markup Language) Internet protocols designed specifically for effective and cost-effective information transfer across mobile networks. Wireless terminals incorporated a HDML (Handheld Device Markup Language) microbrowser, and Phone.com's Handheld Device Transport Protocol (HDTP) then linked the terminal to the UP.Link Server Suite, which connected to the Internet, or intranet where the information being requested resides. The Internet site content was tagged with HDML. This technology was incorporated into WAP- and renamed using some of the many WAP-related acronyms such as S WMLS, WTP and WSP. [1]

Someone with a WAP- compliant phone uses the in-built microbrowser to make a request in WML (Wireless Markup Language), a language derived from HTML especially for wireless network characteristics. This request is passed to a WAP Gateway that then retrieves the information from an Internet server either in standard HTML format or preferably directly prepared for wireless terminals using WML. If the content being retrieved is in HTML format, a filter in the WAP Gateway may try to translate it into WML. A WML scripting language is available to format data such as calendar entries and electronic business cards for direct incorporation into the client



device. The requested information is then sent from the WAP Gateway to the WAP client, using whatever mobile network bearer service is available and most appropriate.

The WAP is a layered protocol stack that contains a session protocol, a transaction protocol, a security protocol, and a datagram protocol. This stack isolates the application from the bearer when used as a transport service. This stack can be seen on figure 1.1 below. [8]



**Figure 1.1 WAP Protocol Stack**

The WAP Stack Protocol consists of the following layers:

1. Wireless Application Environment WAE which defines the user interface on the phone. The aim of the WAE is to develop application environment to facilitate the development of services that support multiple bearers. To achieve this, the WAE contains the Wireless Markup Language (WML), WMLScript- a scripting micro-language similar to JavaScript- and the Wireless Telephony Application (WTA). [1]
2. Wireless Session Protocol WSP is a sandwich layer that links the WAE to two-session services, one connection oriented operating above the Wireless Transaction Protocol and a connectionless service operating above the Wireless Datagram Protocol.
3. Wireless Transaction Protocol WTP, runs on top of a datagram service such as User Datagram Protocol (UDP); part of the standard suite of TCP/IP protocols,

to provide a simplified protocol suitable for low bandwidth mobile stations. WTP offers three classes of transaction service: unreliable one way request, reliable one way request and reliable two way request respond. Interestingly, WTP supports Protocol Data Unit concatenation and delayed acknowledgement to help reduce the number of messages sent. This protocol therefore tries to optimize the user experience by providing the information that is needed when it is needed- it can be confusing to received confirmation of delivery messages when you are expecting the information itself. By stringing several messages together, the end user may well be able to get a better feel more quickly for what information is being communicated.[1]

4. Wireless Transport Layer Security WTLS, incorporates security features that are based upon the established Transport Layer Security (TLS) protocol standard. It includes data integrity checks, privacy on the WAP Gateway to client leg and authentication. Where SA is the source address, SP is the source port, DA is the destination address, DP is the destination port and UD is user data. [8]
5. Wireless Datagram Protocol WDP, Allows WAP to be bearer independent by adapting the transport layer of the underlying bearer. WDP presents a consistent data format to the higher layers of the WAP protocol stack thereby conferring the advantage of bearer independence to application developers. The September 1999 London meeting of the WAP Forum included a decision from the SMS Experts Group that the single common standardized interface between the SMS Center and the WAP Gateway would be a subset of SMPP (Short Message Peer to Peer Protocol) [1]. A PDU (Protocol Data Unit) set has been added to SMPP version 3.4 for this purpose. There will be no SMPP specific legacy- in other words; SMS Center manufacturers that do not support SMPP can evolve their SMS Center external interface to support the new SMPP commands for connecting to WAP Gateways. Basically, this is a victory for Logica, the creators of SMPP, who spun control of the protocol off in 1999 to an "independent" SMPP Forum [1].

#### Optimal WAP Bearer:

- a) Short Message Service; given its limited length of 160 characters per short message, SMS may not be an adequate bearer for WAP because of the weight protocol of the protocol. The overhead of the WAP protocol that would



be required to be transmitted in an SMS message would mean that even for the simplest of transactions several SMS messages might in fact have to be sent. This means that using SMS as a bearer can be a time consuming and expensive exercise. Only one network operator- SBC of the US- is known to be developing WAP services based on SMS. [1]

b) Circuit Switched Data CSD, most of the trial WAP based services use CSD as the underlying bearer. Since CSD has relatively few users currently, WAP could kickstart usage of and traffic generated by this bearer. However, CSD lacks immediacy- a dial up connection taking about 10 seconds is required to connect the WAP client to the WAP Gateway, and this is the best case scenario when there is a complete end to end digital call- in the case of the need for analog modem handshaking (because the WAP phone does not support V.110 the digital protocol, or the WAP Gateway does not have a digital direct connection such as ISDN into the mobile network), the connect time is increased to about 30 seconds. [1]

c) Unstructured Supplementary Services Data USSD is a means of transmitting information or instructions over a GSM network. USSD has some similarities with SMS since both use the GSM network's signaling path. Unlike SMS, USSD is not a store and forward service and is session-oriented such that when a user accesses a USSD service, a session is established and the radio connection stays open until the user, application, or time out releases it. This has more in common with Circuit Switched Data than SMS. USSD text messages can be up to 182 characters in length. USSD has some advantages as a tool for deploying services on mobile networks like the Turnaround response times for interactive applications, Users do not need to access any particular phone menu to access services, services based on USSD work just as well and in exactly the same way when users are roaming, Unstructured Supplementary Services Data (USSD) works on all existing GSM mobile phones, Both SIM Application Toolkit and the Wireless Application Protocol support USSD, and the incorporation of USSD Stage 2 into GSM. It also has some disadvantages in that: USSD was previously a one-way bearer useful for administrative purposes

such as service access; Stage 2 is more advanced and interactive. By sending in a USSD2 command, the user can receive an information services menu. As such, USSD Stage 2 provides WAP-like features on EXISTING phones. USSD strings are typically complicated for the user to remember, involving the use of the "\*" and "#" characters to denote the start and finish of the USSD string. However, USSD strings for regularly used services can be stored in the phonebook, reducing the need to remember and reenter them. USSD could be an ideal bearer for WAP on GSM networks. [1]

d) General Packet Radio Service GPRS is a new packet-based bearer that is being introduced on many GSM and TDMA mobile networks. It is an exciting new bearer because it is immediate (there is no dial up connection), relatively fast (up to 177.2 kbps in the very best theoretical extreme) and supports virtual connectivity, allowing relevant information to be sent from the network as and when it is generated.

There are two efficient means of delivering proactively sending ("pushing") content to a mobile phone: by the Short Message Service which is of course one of WAP bearers or by the user maintaining more or less a permanent GPRS (mobile originated) session with the content server. However, mobile terminated IP traffic might allow unsolicited information to reach the terminal. Internet sources originating such unsolicited content may not be chargeable. A possible worse case scenario would be that mobile users would have to pay for receiving unsolicited junk content. This is a potential reason for a mobile vendor NOT to support GPRS Mobile Terminate in their GPRS terminals. However, by originating the session themselves from their handset, users confirm their agreement to pay for the delivery of content from that service. Users could make their requests via a WAP session, which would not therefore need to be blocked. As such, a WAP session initiated from the WAP microbrowser could well be the only way that GPRS users can receive information onto their mobile terminals. [1] Since all but the early WAP enabled phones will also support the General Packet Radio Service, WAP and GPRS could well be synergistic and be used widely together. For the kinds of interactive, menu based information exchanges that WAP anticipates; Circuit Switched Data is not immediate enough because



of the need to set up a call. Early prototypes of WAP services based on Circuit Switched Data were therefore close to unusable. SMS on the other hand is immediate but is ALWAYS store and forward, such that even when a subscriber has just requested information from their microbrowser, the SMS Center resources are used in the information transfer. As such, GPRS and WAP are ideal bearers for each other. [1] Additionally, WAP incorporates two different connection modes- WSP connection mode or WSP connectionless protocol. This is very similar to the two GPRS Point to Point services- connection oriented and connection less. [1] The predominant bearer for WAP-based services will depend on delays in availability of WAP handsets and delays in the availability of GPRS terminals. If WAP terminals are delayed, most WAP terminals will support GPRS as well. If the first WAP terminals support SMS and Circuit Switched Data, but not GPRS, then SMS could become the predominant initial WAP bearer. [1] WAP certainly will be important for the development of GPRS-based applications. Because the bearer level is separated from the application layer in the WAP protocol stack, WAP provides the ideal and defined and standardized means to port the same application to different bearers. As such, many application developers will use WAP to facilitate the migration of their applications across bearers once GPRS based WAP protocols are supported.

#### **1.4. WAP Development Issues**

WAP version 1.2 may be the first version of the protocol that is actually workable in terms of delivering easy to use and innovative non-voice mobile services. WAP version 1.2 is finalized as a specification in late 1999 and first available in spring 2000 [1]. It will support Push services (proactive delivery of information from a WAP Gateway to a WAP terminal), User Profiles, WDP Tunneling, WMLscript, CryptoLibrary, Wireless Telephony Application, Wireless Application Environment enhancements and other features. There are several non-standardized or unresolved issues relating to WAP that application developers should be aware of: Push Not Supported, Wireless Telephony Application Delayed, and Lack of Cookies for Session Management, Premature Encryption Endpoint and Small Downloadable Unit Size.

#### **1.4.1. Push Not Supported**

The WAP WSP specification defines the WSP push operation and a WSP push PDU (Protocol Data Unit). A push operation is not specified for the HTTP protocol, used by the WAP Gateway server to communicate with content hosts.

To support pushes, the server has to provide an application interface to allow server based applications to generate a push to a mobile client. The support of pushes on the client side depends on the capabilities of the handsets to handle pushed content. For example, The Nokia OTA configuration proposal to the WAP Forum describes the use of a connectionless push over the SMS bearer, to transfer the configuration data to the handset. [1]

#### **1.4.2. Wireless Telephony Application Delayed**

The wireless telephony application WTA is a collection of telephony specific extensions for call and feature control mechanisms, merging data networks and service networks (WAP Forum 1998). [1] The WTA framework integrates advanced telephony services using a consistent user interface and allows network operators to increase accessibility for various special services in their network. Most of the WTA functionality is reserved for the network operators for security and stability reasons.

The so-called Wireless Telephony Application (WTA) was only defined by the WAP Forum in June 1999 [1]. The WTA gives WAP some of the features that SIM Application Toolkit incorporates such as access to phone report and call handling. WTA extends the basic WAE application model in three different ways:

- **Content Push:** A WTA origin server can push content like pushing WML Decks, WML Script to the client, in order to enable the client to handle new network events that were unknown before.
- **Handling of network events:** A device can have a table indicating how to react to certain events from the mobile network. Events could be an incoming call or text message. The device can look up how to react, e.g., look up in a private phonebook in order to map the incoming phone number onto a name.
- **Access to telephony functions:** Applications running on the client can access telephony functions from WML or WML Script in a very simple way. Many functions are available in libraries for setting up calls, making phonebook entries etc. We can define the following three kinds of libraries:

- 1) Common network services: This class contains libraries for services common to all mobile networks.
- 2) Network specific services: Libraries in this class depend on the capabilities of the mobile network. Also, this class might contain operator specific libraries.
- 3) Public services: This class contains libraries with publicly available functions for example 'make call' to set up a phone call. [1]

#### **1.4.3. Lack of Cookies for Session Management**

There are no "cookies" for session management, i.e. to hold the session together.

Cookies are used on the fixed Internet to identify the web browser and thereby assist in providing customized and streamlined services. Instead, some WAP applications use indexes in the URL as an alternative.

The cookie information is transmitted via HTTP headers. Because WAP WSP is based on HTTP headers, it should be possible to transmit cookie information to the clients.

The problem may be the clients itself, which may currently not support the handling of cookie HTTP header information or to save this information to a persistent storage in the mobile phone. [1]

#### **1.4.4. Premature Encryption Endpoint**

The Wireless Transport Layer Security defines encryption between the Mobile Station and the WAP Gateway. The "endpoint" of the encrypted WTLS data is the WAP Gateway proxy server. To have a secure connection to content host (e.g. banking server) the Gateway proxy server has to establish secure (https) connections to this host. In this case the proxy server has access to the decrypted data received via WTLS from the mobile station or from the content host via https. [1]

#### **1.4.5. Small Downloadable Unit Size**

WAP incorporates no compression techniques for the textual content, although the WML markup commands are compressed. Additionally, the "deck"- the smallest unit of downloadable information in Wireless Markup Language- is limited to a maximum of 1400 bytes. This means that applications need to be specifically designed to be very code efficient by using templates and variables and keeping information on the server and using the cache on the phone.



WML byte code converting defines a (maybe inefficient) compression technique by string tables. With this technique duplicate strings in the WMLC bytecode are avoided. This reduces the size of the data to transfer to the mobile client. The WSP SDU size of 1400 bytes is a default value. An increased size may be negotiated by a mobile client within the WSP capabilities. The WAP transport layer (WTP) is able to handle greater SDU sizes than 1400 too, by using SAR (Segmentation and Re-assembly). [1]

After presenting different aspects of WAP, this section deals once more with the scope of standardization efforts.

WAP tries to use existing technologies and philosophies as much as possible, mainly from the Internet. Thus, the simplest protocol stack, stack number3, does not require new protocols or implementations. If an application needs only unreliable datagram service without security, WAP offers a way to use UDP if the bearer network provides IP service like that in GPRS. Many complex stacks based on this very simple stack. The typical WAP application, i.e., a WAP user agent such as a WML or a WTA user agent, may require the full stack of protocols as shown in stack1. These user agents run in the WAE and rely on, e.g., the WSP push service for pushing WTA events from a WTA server to the client. [1]

### **1.5. WAP Developer's Toolkits**

There are at least four WAP toolkits available for software developers to use to assist in the speedy development of WAP-based services. These are supplied by Dynamical Systems Research (DSR), Ericsson, Nokia and Phone.com.

### **1.6. WAP Clients and Gateways**

WAP is a client server philosophy, requiring a microbrowser in the mobile phone and a WAP Gateway connected to the mobile network. By early 2000, WAP clients such as the Nokia 7110 were becoming available in quantity and other phone vendors such as Alcatel and Motorola have announced that they are introducing support for the Wireless Application Protocol across their entire product range. [1] However, since WAP requires a larger screen size and more memory to handle the WAP stack, it costs more to produce a WAP handset and will therefore mean more expensive mobile phone



prices. WAP phones will therefore be distinguishable from their non WAP counterparts to the informed observer- and will have the "WWW: MMM" branding anyway- which the WAP Forum founders have agreed on to depict WAP terminals. Support by mobile phones for WAP will be the simple largest determinant of when WAP is a success. [1]

SIM Application Toolkit is another wireless protocol that enables a similar functionality set to WAP. SIM Application Toolkit has been around for longer than WAP and is at a later stage of development and deployment than WAP but is a GSM only technology that has not been widely adopted by leading mobile phone vendors such as Nokia and Ericsson. SIM Application Toolkit is supported by perhaps a quarter of the installed base of GSM phones. It may be that application developers need to support BOTH WAP and SIM Application Toolkit AND standard SMS in their Gateways so that the applications and services can be offered to ALL mobile phone users, rather than just a subset. Widespread reach is of course essential in maximizing use of the services and helping build a wireless Internet portal that is popular with all mobile phone users. [1]

Despite today's lack of an installed base of WAP capable mobile phones, there are several vendors of WAP Gateways that network operators; content providers and application developers can work with to develop WAP-based services. WAP Gateways are installed into the mobile phone network to provide a gateway between the Internet and different mobile nonvoice services such as the Short Message Service, Circuit Switched Data and General Packet Radio Service. The WAP Gateway is essentially a piece of middleware, taking information from a web server, processing it, and sending it out over the mobile network to a WAP client. [1]

Each of the WAP Gateways has strengths and weaknesses. Selection will depend on intended use for the platform.

### **1.7. Applications**

WAP is being used to develop enhanced forms of existing applications and new versions of today's applications.

Existing mobile data software and hardware suppliers are adding WAP support to their offering, either by developing own WAP interface or more usually partnering with one of the WAP Gateway suppliers profiled above. WAP is also given a significant impetus

for new players to add mobile as a new distribution channel for their existing products and services- for example, CNN and Nokia teamed up to offer CNN Mobile and Reuters and Ericsson teamed up to provide Reuters Wireless Services.

The Wireless Application Protocol will allow customers to easily reply to incoming information on the phone by allowing new menus to access mobile services. This is part of the business case for network operators- by making the value-added services more easily to reply to and request (using menus instead of keywords; for example), WAP can help generate additional traffic on the network and therefore revenue. [1]

Application developers wrote proprietary software applications and had to port that application to different network types and bearers within the same platform.

By separating the bearer from the application, WAP facilitates easy migration of applications between networks and bearers. As such, WAP is similar to Java in that it simplifies application development. This reduces the cost of wireless application development and therefore encourages entry to the mobile industry by software developers. [1]

Corporate applications that are being enhanced and enabled with a WAP interface include:

- Job Dispatch
- Remote Point Of Sale
- Customer Service
- Remote Monitoring Such As Meter Reading
- Vehicle Positioning
- Corporate Email
- Remote LAN Access
- File Transfer
- Web Browsing
- Document Sharing/ Collaborative Working
- Audio
- Still Images
- Moving Images
- Home Automation

Consumer Applications that are being enhanced and enabled with a WAP interface include:

- Simple Person to Person Messaging
- Voice and Fax Mail Notifications
- Unified Messaging
- Internet Email
- Prepayment
- Ringtones
- Mobile Commerce
- Affinity Programs
- Mobile Banking
- Chat
- Information Services [1]

## **1.8 WAP and the Web**

From a certain viewpoint, the WAP approach to content distribution and the Web approach are virtually identical in concept. Both concentrate on distributing content to remote devices using inexpensive, standardized client software. Both rely on back-end servers to handle user authentication, database queries, and intensive processing. Both use markup languages derived from SGML for delivering content to the client. In fact, as WAP continues to grow in support and popularity, it is highly likely that WAP application developers will make use of their existing Web infrastructure (in the form of application servers) for data storage and retrieval. [1]

### **1.8.1 WAP and Web Heredity's**

WAP (and its parent technology, XML) will serve to highlight the Web's status as the premier n-tier application in existence today. WAP allows a further extension of this concept as existing "server" layers can be reused and extended to reach out to the vast array of wireless devices in business and personal use today. Note that XML, as opposed to HTML, contains no screen formatting instructions; instead, it concentrates on returning structured data that the client can use as it sees fits. [7]

As time went on, managers were eventually even able to make the business case for client/server access to mainframe databases from Windows applications. This opened



up existing databases to improved reporting, charting, and other user interface features. Managers and shop foremen can access parts inventories, repair schedules, shop budgets, and other useful information in order to plan work crew schedules and employee tasking. [7]

*It was just another small step from there for management to take advantage of the Web development skills by Web-enabling various mainframe applications (buzzword alert: we now call this Enterprise Application Integration or EAI). With this information on the Web, information can be shared with parts suppliers and contractors which has greatly reduced ordering times and costs involved. One problem remains, however: out of 10,000 employees and contractors, only about 500 actually interact with the databases. The remainder of the employees continually fills out paperwork, issue reports to their manager, or manually key in data when they return from working on a ship. If the other 9500 employees actively involved in welding, pipefitting, installing electrical cable, and testing electronics could all wirelessly retrieve and/or edit data when they actually need to; Small, inexpensive devices are given to each employee based on their tasking requirements. Some require handheld devices with built-in barcode scanners, others require keypads, and others require simple digital displays. WAP allows a suite of client applications to be built which reuse existing server applications and databases. In addition, these applications can be dynamically downloaded and run on any of these devices. If an electronics tester runs into a bad vacuum tube, he scans the barcode. If a cable installer realizes that 500 more feet of a specific type of cable are required, he selects the "Order Cable" menu option from his wireless phone. If someone installing HVAC ventilation wants to know which pipes or cables run through a specific section of the ship, he enters the query in on his PDA and retrieves either data or imagery information. [1]*

In any industry that involves employees stepping out of their office to complete a job, wireless applications will be abundant. WAP helps standardize the applications that will proliferate using wireless communication technologies. Imagine the Web without the combination of HTML and HTTP leaving us instead with "open" specifications from Sun Microsystems, Microsoft, and IBM. I will go out on a limb and say that there is no

chance the Web would be where it was today without freely available, vendor-neutral, open standards. [7]

### **1.8.2. Specifications of How It Works**

WAP uses some new technologies and terminologies, which may be foreign to the software developer; however the overall concepts should be very familiar. WAP client applications make requests very similar in concept to the URL concept in use on the Web. As a general example, consider the following explanation (exact details may vary on a vendor-to-vendor basis). [18]

A WAP request is routed through a WAP gateway which acts as an intermediary between the “bearer” used by the client (GSM, CDMA, TDMA, etc.) and the computing network that the WAP gateway resides on (TCP/IP in most cases). The gateway then processes the request, retrieves contents or calls CGI scripts, Java servlets, or some other dynamic mechanism, then formats data for return to the client. This data is formatted as WML (Wireless Markup Language), a markup language based directly on XML. Once the WML has been prepared (known as a deck), the gateway then sends the completed request back (in binary form due to bandwidth restrictions) to the client for display and/or processing. The client retrieves the first card off of the deck and displays it on the monitor. [7]

The deck of cards metaphor is designed specifically to take advantage of small display areas on handheld devices. Instead of continually requesting and retrieving cards (the WAP equivalent of HTML pages), each client request results in the retrieval of a deck of one or more cards. The client device can employ logic via embedded WMLScript (the WAP equivalent of client-side JavaScript) for intelligently processing these cards and the resultant user inputs.

To sum up, the client makes a request. This request is received by a WAP gateway that then processes the request and formulates a reply using WML. When ready, the WML is sent back to the client for display. As mentioned earlier, this is very similar in concept to the standard stateless HTTP transaction involving client Web browsers.

### **1.8.3. Communications between Client and Server**

The WAP Protocol Stack is implemented via a layered approach (similar to the OSI network model). These layers consist (from top to bottom) of:

Wireless Application Environment (WAE)

Wireless Session Protocol (WSP)

Wireless Transaction Protocol (WTP)

Wireless Transport Layer Security (WTLS)

Wireless Datagram Protocol (WDP)

Bearer (GSM, IS-136, CDMA, GPRS, CDPD, etc.) [1]

According to the WAP specification, WSP offers means to provide HTTP/1.1 functionality by means of extensible request-reply methods, composite objects, content type negotiation, exchange client and server session headers, interrupt transactions in process, push content from server to client in an unsynchronized manner and negotiate support for multiple, simultaneous asynchronous transactions. [7]

WTP provides the protocol that allows for interactive browsing (request/response) applications. It supports three transaction classes: unreliable with no result message, reliable with no result message, and reliable with one reliable result message. Essentially, WTP defines the transaction environment in which clients and servers will interact and exchange data. [1]

The WDP layer operates above the bearer layer used by your communications provider. Therefore, this additional layer allows applications to operate transparently over varying bearer services. While WDP uses IP as the routing protocol, unlike the Web, it does not use TCP. Instead, it uses UDP (User Datagram Protocol) which does not require messages to be split into multiple packets and sent out only to be reassembled on the client. Due to the nature of wireless communications, the mobile application must be talking directly to a WAP gateway (as opposed to being routed through myriad WAP access points across the wireless Web) which greatly reduces the overhead required by TCP. [23]



For secure communications, WTLS is available to provide security. It is based on SSL and TLS.

#### **1.8.4. The Wireless Markup Language (WML)**

WML is, in fact, an XML document type defined by a standard XML Document Type Definition, or DTD. However the following code gives an example of a simple WML file.

```
Hello World!
```

The first two lines are required. They give the XML version number and the public document identifier, respectively. From there, all WML decks (one WML file equals one deck) begin and end with the tags. Individuals' cards are arranged with the tags. Also, note that WML, like XML, is case-sensitive! Included in the WML specification are elements that fall into the following categories: Decks/Cards, Events, Tasks, Variables, User Input, Anchors/Images/Timers, and Text Formatting. See the WML tutorial for specific examples on using these elements to build applications.

WML is a markup language that is based on XML (eXtensible Markup Language). The official WML specification is developed and maintained by the WAP Forum, an industry-wide consortium founded by Nokia, Phone.com, Motorola, and Ericsson. This specification defines the syntax, variables, and elements used in a valid WML file.

A valid WML document must correspond to this DTD (Document Type Definition) or it cannot be processed. WML basics and an example will be present. This example will demonstrate events and navigation as well as data retrieval from server CGI scripts. Discussion of client-side scripting and state management will be presented in the WML Script tutorial.

Here we will explore and list the basics of both the WML and WMLscript languages, in the sense that both are part of the WAP specification as defined by the members of the WAP Forum. Since the currently available mobile devices are all version 1.1 compatible only, we will use this version although the latest version is 1.2. Although the general syntax of WML looks a lot like HTML there are a few notable differences. First of all, the document structure; while an HTML page is generally built up out of a

header and a body; WML pages have one header and one (optional) template but can have multiple "body's" called cards.

### **1.8.5. Additional Intelligence via WMLScript**

The purpose of WMLScript is to provide client-side procedural logic. It is based on ECMAScript (which is based on Netscape's JavaScript language), however it has been modified in places to support low bandwidth communications and thin clients. The inclusion of a scripting language into the base standard was an absolute must. While many Web developers regularly choose not to use client-side JavaScript due to browser incompatibilities (or clients running older browsers), this logic must still be replaced by additional server-side scripts. This involves extra roundtrips between clients and servers which is something all wireless developers want to avoid. WMLScript allows code to be built into files transferred to mobile client so that many of these round-trips can be eliminated. According to the WMLScript specification, some capabilities supported by WMLScript that are not supported by WML are:

- Check the validity of user input
- Access to facilities of the device. For example, on a phone, allow the programmer to make phone calls, send messages, add phone numbers to the address book, access the SIM card etc.
- Generate messages and dialogs locally thus reducing the need for expensive round-trip to show alerts, error messages, confirmations etc.
- Allow extensions to the device software and configuring a device after it has been deployed.

WMLScript is a case-sensitive language that supports standard variable declarations, functions, and other common constructs such as if-then statements, and for/while loops. Among the standard's more interesting features are the ability to use external compilation units (via the use URL pragma), access control (via the access pragma), and a set of standard libraries defined by the specification (including the Lang, Float, String, URL, WMLBrowser, and Dialogs libraries). The WMLScript standard also defines a bytecode interpreter since WMLScript code is actually compiled into binary form (by the WAP gateway) before being sent to the client.



WMLScript is based on JavaScript, but it has been adapted for use in the low bandwidth environment of mobile devices. For instance WMLScript can be compiled into bytecode to speed up interpretation by the device and it lacks some of the more advanced features of JavaScript.

Like JavaScript, WMLScript has precompiled libraries of functions you can call from your WAP page. But unlike JavaScript it lacks objects and their methods; therefore you have to rely on the six available Standard Libraries in WMLScripts which are: Lang, Float, String, URL, WMLBrowser, and Dialogs.

#### **1.8.6. The Business Case**

WAP's biggest business advantages are the prominent communications vendors who have lined up to support it. The ability to build a single application that can be used across a wide range of clients and bearers makes WAP pretty much the only option for mobile handset developers at the current time. Whether this advantage will carry into the future depends on how well vendors continue to cooperate and also on how well standards are followed. [1]

It is very, very early on in the ballgame and already vendor toolkits are offering proprietary tags that will only work with their microbrowser. Given the history of the computing industry and competition, in general, this was to be expected. However, further differentiation between vendor products and implementations may lead to a fragmented wireless Web. [1]

WAP also could be found lacking if compared to more powerful GUI platforms such as Java, for instance. For now, processor speeds, power requirements, and vendor support are all limiting factors to Java deployment but it's not hard to imagine a day in the near future where Java and WAP exist side-by-side just as Java and HTML do today. In that circumstance, Java would hold a clear advantage over WAP due to the fact that a single technology could be used to build applications for the complete range of operating devices. Of course, on the flip side, the world is not all Java and there will always be a place for markup languages in lieu of full-blown object-oriented platforms. [17]



### 1.9. Summary

In this chapter the Wireless Application Protocol's overview, historical background, technology, WAP development issues, WAP developer's toolkits and the WAP client and gateways we had seen. The Wireless Application Protocol (WAP) is an important development in the wireless industry because of its attempt to develop an open standard for wireless protocols, independent of vendor and airlink. The goals of the Wireless Application Protocol had also been discussed.

## **2. INTERNET SECURITY AND WIRELESS LANS**

### **2.1. Overview**

Hundreds of millions of Internet users around the world have become accustomed to an Internet beyond boundaries. One site flows to the next, a jungle of software, protocols, media and people connecting, signal, noise, mixing, evolving, together. It seems silly to ignore the security of the system \_as a whole\_, but we still do. A helpful analogy might be to consider the Internet more a living organism than a neighborhood. A security compromise is can behave more like a disease than a "breakin". It is often contagious, and can spread. Remotely exploitable security vulnerabilities are like the natural wounds of the skin. They are relatively rare, sometimes difficult to squirm through, but once inside, infection can begin. [23]

The Internet is the world's largest network of networks. When one access the resources offered by the Internet, in fact he does not connect to the Internet, but connect to a network that is eventually connected to the Internet backbone, a network of extremely fast network components. This is an important point: the Internet is a network of networks. [23]

### **2.2. Internet Security Risks and Remedies**

Internet security risks aren't to be taken lightly, but they can all be managed and minimized like other security risks in business. There are Internet security precautions that must be follows. The user needs to know what they are, how much protection they give, what they cost, how to get them installed and how to use them. Setting up tight security over the Internet is mainly a matter of knowledge. [23]

Suppose that a downtown bank may need a vault that costs millions. In contrast, 'bank vault' security on the Internet may cost little, if the people involved know enough. Two penniless but astute 16-year-olds could send each other Internet messages just as safely as two banks. An Internet bank needs more security precautions than an Internet CD shop. [23]

The most spread Internet risks are: Hackers, industrial espionage, hi-tech criminals and viruses.

#### **2.2.1. Hackers**

There are many hackers (it's hard to know exactly how many). Many of them have unimpressive skills, aren't creative, and simply borrow someone else's hacking software for their exploits. There are routine and simple security measures to protect Internet traffic against the junior-grade hackers. Some hacker masterminds can find new ways to break into computers. But such people are rare.

#### **2.2.2. Industrial Espionage**

In the past, fax interception has sometimes been used for industrial espionage (microwave and satellite links make interception easy). Industrial spies must have turned their attention to e-mail and other Internet traffic, if that's where the secrets are flowing, that's where they will look. It is easy to stop this sort of spying by (scrambling) encrypting messages in a way that even well-financed spies will find the messages practically impossible to read. [23]

#### **2.2.3. Hi-Tech Criminals**

This is similar to the risk from clever and greedy hackers. The criminal interception can't be ruled out if the electronic communications are valuable.

There is powerful protection available. The main cost is training the personnel, because the software needed isn't expensive.

#### **2.2.4. Viruses**

Some people think that if they connect to the Internet, their computer systems will immediately start picking up "Internet" viruses. It can happen. But it's also one of the easiest risks to manage. They need anti-virus software to scan incoming files. Internet viruses are no more menacing than viruses from CD ROMS or floppy disks. [23]

### **2.3. Business Communication over the Internet: Risks and Remedies**

the Internet has enormous appeal as a business communication system: it is cheap, it is wonderfully versatile, and it can be used for one-to-one communications or widened to



conference-like communications. It can even be used to set up a private computer network. One can use these systems to communicate with customers, suppliers, overseas offices, the bank etc.

### **2.3.1. E-mail Security**

A business now needs e-mail in the same way it needed a fax machine a few years ago. One can send a typed message, a spreadsheet, a word-processing document, a drawing, a photograph, anything that's an electronic file and it's fast and cheap.

### **2.3.2. The Risks**

E-mail is forwarded through a chain of computers. The linking computers are owned by individuals, universities, companies, or governments etc., and are in many countries. A company could set up a computer that forwarded Internet mail, for example. There's nothing technical to stop the people who administer those computers from scanning e-mail going through their machines. It can be done automatically, looking for certain key words. The people who send e-mail or receive it wouldn't know it was happening. Hackers are capable of similar snooping. They can do it by planting 'sniffer' software on one or more computers that forward e-mail. The 'sniffer' then copies messages that contain credit-card numbers, certain people's names, words like 'password', etc. The hacker then gets a copy. So that's one risk with e-mail: It can be intercepted undetectably, and copied. [23]

Another risk is e-mail forgery. It is elementary to fake a sender's name and e-mail address. If a businessperson receives a faked e-mail and is taken in by it, it could be expensive or embarrassing, while the solution is to encrypt e-mail before it's sent. It can be done so only one chosen person can decrypt (unscramble) it. A good encryption system also include a 'digital signature' proves who sent the e-mail. A final risk, not to be forgotten, is that e-mail tends to be saved by the people who receive it. Sometimes they forward it to others, who also save it. So a written e-mail casually in a few minutes, maybe thinking of it in the same way as making a relaxed phone call, may end up stored for years on many people's disk drives. This should be kept in mind when reviewing e-mail before press the 'send' button. [23]

Some people try to protect their e-mail messages by first saving them as word processing files, with a password. Then they send the word-processing files by e-mail, as attachments. The concept is broadly right. Unfortunately, word-processing password systems are too weak for the realities of the Internet. There are free password 'crackers' that can break most of those password systems. If someone knows how to intercept a file, it's a good bet password will soon be displayed on their computer screen. No protection at all. A company in the US sells sophisticated software that cracks all the common business-application password systems: Microsoft Word, Excel, and Money, WordPerfect, Lotus 123, Novell Netware, and others. It's meant for business people who forget their passwords and need to re-open their 'protected' files. But anyone can buy the cracking software. A sounder approach is to use a password with the compression software pkzip. It is used to compress electronic files before they're sent across the Internet. It also has a password protection system that's much stronger than the systems used in word-processors. Another advantage: everyone who has been using the Internet for a while has a copy of pkzip. They need it to decompress files they download from the web. So there's no software to 'set up'. There are several pkzip password crackers on the Internet. Most of them work by systematically trying everything until the password is found. Some of them start with a 'dictionary attack', trying the words in a large dictionary. Then they might work through every combination of the lowercase letters, working up through longer and longer random 'words', then start including uppercase letters and numbers and so on. [23]

Warning 1: pkzip is regarded as 'low level' security by cryptographers. There are ways to break into it, apart from just trying passwords until the right one turns up. The most advanced and efficient methods may be in the hands of national security agencies only. Or they may not be. It's noticeable that pkzip is not export-restricted from the US. [26]

Warning 2: There is free software that will extract a pkzip password if known (or can guessed) a small but exact part of the message. If someone can guess part of any message, they can read all the other messages. [23]

"Pretty Good Privacy" (PGP) is encryption software first put together in 1991 by an American cryptographer, Phil Zimmermann. Then it was distributed free on the Internet. For three years, he was under threat of prosecution by the US Government for



exporting a "munition". The encryption system was so strong it was classified under cold-war rules as a weapon of war. [15]

It is certainly a strong, well-tested and carefully reviewed system. Its workings are open for inspection and they always have been. It has been around for quite a long time. It passes all the practical tests. PGP is the best publicly available system for sending electronic messages securely.

There are two drawbacks, but not imposing ones.

1. It has been an elite system, so it's unlikely to be found among the software of the business person aimed to communicate with. They have to be encouraged to install it and learn how to use it.
2. The early versions were satisfactory for software enthusiasts, but off-putting for most business people. There were obscure software commands and the manual was steeply challenging. The latest version has fixed most of that. PGP is no longer hard to use, or hard to understand. [23]

PGP can be used to encrypt any electronic file. So it's useful for a range of electronic security, other than just e-mail messages. The situation today is that PGP software can be used in most developed countries without breaking any laws or infringing patents, including US ones. It is important to spell out this legal point, because PGP has carried a history of "trouble" and "spooks" with it.

There is something else that increases the fog in this subject, even an encryption system that passes every known mathematical test might still be programmed badly. It only takes a slight mistake or oversight, and suddenly there's a simple way to defeat the system. The following are two practical tests of encryption software:

1. There are independent cryptography experts who revel in exposing security flaws (indeed, it adds to their stature in the profession). So if a prominent encryption system is open for inspection.
2. The longer the system has been used without anyone finding a flaw in it, the more soundly conceived and soundly programmed.

FTP stands for "File Transfer Protocol". It's a way to copy a file from your hard drive to the hard drive of another computer connected to the Internet. In business, it is mainly



used to send long files, or to send large numbers of files. But it is risky.

A Virtual Private Network (predictably known as a VPN) is a way to enlarge a company's private intranet by using the public Internet. One can connect distant offices cheaply, for example, without having to lease dedicated lines. VPN software creates private 'tunnels' through the Internet. Anyone who intercepts the messages passing through these tunnels will only get encrypted and unreadable strings of bits. [25]

That's what is supposed to happen. But VPN is fairly new, its quality varies, and there have been some notable embarrassments about the encryption systems. The underlying system recommended by professional cryptographers is a standard known as IPSec. VPN systems using that standard are on offer by major vendors, but they are exported from the US and their encryption strength is severely weakened (they use a 40-bit key instead of 128 bits). [25]

## **2.4. Security Concerns About Websites**

### **2.4.1. Websites Used Only For Advertising**

If one uses a website simply to make information available to own prospects or customers, then the security risks are almost nonexistent. For one thing, there is no confidential information anyone can steal. A hacker could break into the computer where the website's files are stored. If someone malicious wanted to change website files, that person would have to get illegal access to the web-host computer. In other words, 'hack in' somehow. But that wouldn't give the hacker access to the computers in the company. The damage stops at the web-host computer. The main risk is temporary embarrassment, if pranks are played with website. All it normally takes to correct that is a public announcement.

### **2.4.2. Websites Used To Make Sales and Get Paid**

This is complicated. Technical people will set up a payment system, but as an executive or manager needs to be clear on the security risks.

Collecting the credit-card details

The least will need is a secure way to collect people's credit-card details. The web-host computer will respond by setting up a secure link to the customer's computer. At least it

tries to. Even a secure server may fail if the customer's browser isn't equipped to do it. A few customers may be using out-of-date browsers that can't handle it. In that case, the customer will get a message saying there's no compatible security system available, and do they want to entrust their credit-card details to an insecure link? (No, usually.)

Nothing can be done about old browsers, in most cases, the customer's browser and the secure server will successfully set up an encrypted link between them. The technical details of this process make a considerable subject. [23]

## **2.5. Special Case**

### **2.5.1. Situation in Australia**

In Australia, almost any browser will connect at the lowest level of security. That's because US export rules cripple the built-in encryption of browsers that can be exported from the US. Technically, the session keys are limited to just 40 bits instead of 128 bits, and the RSA keys limited to 512 bits instead of 1024. That makes all the difference between a powerful encryption system and one that's fairly easily broken provided someone has the interest. A 40-bit session key can be broken, but it still takes some time and a fair amount of computer power. It's unlikely anyone would go to that much trouble just to steal one credit-card number. Customers often feel quite secure, even with the lowest security. [3]

There's another complication we need to discuss, because it's important commercially. It is possible to restore a crippled browser to full-strength encryption. The full-strength browser will then connect to a US secure server with full 128-bit encryption. But it may only connect to an Australian secure sever with 40-bit encryption. That's because Australian secure servers themselves may use exported American encryption software, and that too is crippled. [2]

### **2.5.2. Processing the Credit Cards**

Having collected the credit-card details, need to process them and get the funds transferred into one's bank account. They will take a cut, and send you a monthly cheque.

1. People's credit-card details need to be stored securely. While individual credit cards aren't a very tempting target for hackers or criminals, it's different with hundreds or



thousands of credit-card numbers in one place. If the details are stored, they should be stored on a separate computer, not connected to the Internet.

2. If one's company processes the credit-card details, you need a secure way to get the credit-card details from the secure server. Web hosts offer all kinds of ways to send the credit-card details. They range from completely insecure to very secure. Some web hosts simply e-mail the card details to you (with no security). Some will e-mail them to you using PGP encryption. [2]

### **2.5.3. Internet Banking**

What to look for in a bank's Internet encryption system Key lengths. Check if their RSA keys are 1024 bits or more, and the session keys are 128 bits or more. There have been many restrictions on exporting strong US cryptography, and some experts are also wary about secret 'back doors' and similar worries. To get around that, one Australian bank for example, licensed an important part of its encryption system from Switzerland. They say it was specifically to avoid the USA regulations that "prevent the export of symmetric key encryption technology with keys longer than 40 bits."

Many security failures in electronic encryption have been traced to unwarranted pride, or foolish secrecy: the people who developed the system were unwilling to let outside experts review it. [2]

### **2.5.4. SET (Secure Electronic Transactions)**

SET is a system for making payments securely over the Internet. It was developed by credit-card issuers and some major software and computer companies in the US. It uses encryption to make the transactions secure, and digital signatures to identify both merchants and buyers. One will have to wait until enough of customers have been issued digital signatures and have installed 'wallet' software on their computers. That will take time. [2]

## **2.6. Wireless LANS**

A wireless local area network (LAN) is a flexible data communications system implemented as an extension to or as an alternative for, a wired LAN. Using radio frequency (RF) technology, wireless LANs transmit and receive data over the air, minimizing the need for wired connections. Thus, wireless LANs combine data connectivity with user mobility. [1]

Wireless LANs have gained strong popularity in a number of vertical markets, including the health-care, retail, manufacturing, warehousing, and academia. These industries



have profited from the productivity gains of using hand-held terminals and notebook computers to transmit real-time information to centralized hosts for processing. Today wireless LANs are becoming more widely recognized as a general-purpose connectivity alternative for a broad range of business customers.

### **2.6.1 Wireless**

The widespread reliance on networking in business and the meteoric growth of the Internet and online services are strong testimonies to the benefits of shared data and shared resources. With wireless LANs, users can access shared information without looking for a place to plug in, and network managers can set up or augment networks without installing or moving wires. Wireless LANs offer the following productivity, convenience, and cost advantages over traditional wired networks: mobility, installation Speed and Simplicity, Installation Flexibility, Reduced Cost-of-Ownership and Scalability. [4]

Wireless LANs frequently augment rather than replace wired LAN networks, often providing the final few meters of connectivity between a wired network and the mobile user. It has many applications in the real life ranges from the hospital up to school etc...

There is a range of technologies to choose from when designing a wireless LAN solution; each technology has its own set of advantages and limitations. These technologies are: Narrow Band Technology, Spread Spectrum Technology, Frequency-Hopping Spread Spectrum Technology, Direct-Sequence Spread Spectrum Technology, and Infrared Technology. [2]

### **2.6.2. Narrowband Technology**

A Narrowband radio system transmits and receives user information on a specific radio frequency and keeps the radio signal frequency as narrow as possible just to pass the information. Undesirable crosstalk between communications channels is avoided by carefully coordinating different users on different channel frequencies. A private telephone line is much like a radio frequency. When each home in a neighborhood has its own private telephone line, people in one home cannot listen to calls made to other homes. In a radio system, privacy and noninterference are accomplished by the use of

separate radio frequencies. The radio receiver filters out all radio signals except the ones on its designated frequency. [25]

### **2.6.3. Spread Spectrum Technology**

Most wireless LAN systems use spread-spectrum technology, a wideband radio frequency technique developed by the military for use in reliable, secure, mission-critical communications systems. Spread-spectrum is designed to trade off bandwidth efficiency for reliability, integrity, and security. In other words, more bandwidth is consumed than in the case of Narrowband transmission, but the tradeoff produces a signal that is, in effect, louder and thus easier to detect, provided that the receiver knows the parameters of the spread-spectrum signal being broadcast. If a receiver is not tuned to the right frequency, a spread-spectrum signal looks like background noise. There are two types of spread spectrum radio: frequency hopping and direct sequence. [25]

### **2.6.4. Frequency-Hopping Spread Spectrum Technology**

Frequency-hopping spread-spectrum (FHSS) uses a Narrowband carrier that changes frequency in a pattern known to both transmitter and receiver. Properly synchronized, the net effect is to maintain a single logical channel. To an unintended receiver, FHSS appears to be short-duration impulse noise. [25]

### **2.6.5. Direct-Sequence Spread Spectrum Technology**

Direct-sequence spread-spectrum (DSSS) generates a redundant bit pattern for each bit to be transmitted. This bit pattern is called a chip (or chipping code). The longer the chip; the greater the probability, that the original data can be recovered; and the more bandwidth required. Even if one or more bits in the chip are damaged during transmission, statistical techniques embedded in the radio can recover the original data without the need for retransmission. To an unintended receiver, DSSS appears as low-power wideband noise and is rejected (ignored) by most Narrowband receivers. [25]

### **2.6.6. Infrared Technology**

Infrared (IR) systems use very high frequencies, just below visible light in the electromagnetic spectrum, to carry data. Like light, IR cannot penetrate opaque objects; it is either directed (line-of-sight) or diffuse technology. Inexpensive directed systems provide very limited range (3 ft) and typically are used for personal area networks but occasionally are used in specific wireless LAN applications. High performance directed IR is impractical for mobile users and is therefore used only to implement fixed sub-



systems. Diffuse (or reflective) IR wireless LAN systems do not require line-of-sight, but are limited to individual rooms. [25]

### **2.2.7 Wireless LANs Work**

Wireless LANs use electromagnetic airwaves (radio or infrared) to communicate information from one point to another without relying on any physical connection. Radio waves are often referred to as radio carriers because they simply perform the function of delivering energy to a remote receiver. The data being transmitted is superimposed on the radio carrier so that it can be accurately extracted at the receiving end. This is generally referred to as modulation of the carrier by the information being transmitted. Once data is superimposed (modulated) onto the radio carrier, the radio signal occupies more than a single frequency, since the frequency or bit rate of the modulating information adds to the carrier. [25]

Multiple radio carriers can exist in the same space at the same time without interfering with each other if the radio waves are transmitted on different radio frequencies. To receive data, a radio receiver tunes in one radio frequency while rejecting all other frequencies. [25]

In a typical wireless LAN configuration, a transmitter/receiver (transceiver) device, called an access point, connects to the wired network from a fixed location using standard cabling. At a minimum, the access point receives, buffers, and transmits data between the wireless LAN and the wired network infrastructure. A single access point can support a small group of users and can function within a range of less than one hundred to several hundred feet. The access point (or the antenna attached to the access point) is usually mounted high but may be mounted essentially anywhere that is practical as long as the desired radio coverage is obtained. [25]

End users access the wireless LAN through wireless-LAN adapters, which are implemented as PC cards in notebook or palmtop computers, as cards in desktop computers, or integrated within hand-held computers. Wireless LAN adapters provide an interface between the client network operating system (NOS) and the airwaves via an antenna. The nature of the wireless connection is transparent to the NOS. [25]



### 2.6.8. Wireless LAN Configurations

Wireless LANs vary from being simple to complex. At its most basic, two PCs equipped with wireless adapter cards can set up an independent network whenever they are within range of one another. This is called a peer-to-peer network. On-demand networks such as in this example require no administration or pre-configuration. In this case each client would only have access to the resources of the other client and not to a central server. [25]

Installing an access point can extend the range of an ad hoc network, effectively doubling the range at which the devices can communicate. Since the access point is connected to the wired network each client would have access to server resources as well as to other clients. Each access point can accommodate many clients; the specific number depends on the number and nature of the transmissions involved. Many real-world applications exist where a single access point services from 15-50 client devices. Access points have a finite range, on the order of 500 feet indoor and 1000 feet outdoors. In a very large facility such as a warehouse, or on a college campus it will probably be necessary to install more than one access point. Access point positioning is accomplished by means of a site survey. The goal is to blanket the coverage area with overlapping coverage cells so that clients might range throughout the area without ever losing network contact. The ability of clients to move seamlessly among a cluster of access points is called roaming. Access points hand the client off from one to another in a way that is invisible to the client, ensuring unbroken connectivity. [25]

To solve particular problems of topology, the network designer might choose to use Extension Points to augment the network of access points. Extension Points look and function like access points, but they are not tethered to the wired network, as are APs. EPs function just as their name implies: they extend the range of the network by relaying signals from a client to an AP or another EP. EPs may be strung together in order to pass along messaging from an AP to far-flung clients, just as humans in a bucket brigade pass pails of water hand-to-hand from a water source to a fire. [25]

One last item of wireless LAN equipment to consider is the directional antenna. Let's suppose you had a wireless LAN in your building A and wanted to extend it to a leased building, B, one mile away. One solution might be to install a directional antenna on

each building, each antenna targeting the other. The antenna on A is connected to your wired network via an access point. The antenna on B is similarly connected to an access point in that building, which enables wireless LAN connectivity in that facility. [26]

While wireless LANs provide installation and configuration flexibility and the freedom inherent in network mobility, one should be aware of the following factors when considering wireless LAN systems: Range and coverage, Throughput, Integrity and Reliability, Compatibility with the Existing Network, Interoperability of Wireless Devices, Interference and Coexistence, the Licensing Issues, Simplicity/Ease of Use, the Cost and the Security and Safety. [26]

#### **2.6.9. Security**

Because wireless technology has roots in military applications, security has long been a design criterion for wireless devices. Security provisions are typically built into wireless LANs, making them more secure than most wired LANs. It is extremely difficult for unintended receivers (eavesdroppers) to listen in on wireless LAN traffic. Complex encryption techniques make it impossible for all but the most sophisticated to gain unauthorized access to network traffic. In general, individual nodes must be security-enabled before they are allowed to participate in network traffic. [25]

#### **2.6.10. Safety**

The output power of wireless LAN systems is very low, much less than that of a handheld cellular phone. Since radio waves fade rapidly over distance, very little exposure to RF energy is provided to those in the area of a wireless LAN system. Wireless LANs must meet stringent government and industry regulations for safety. No adverse health affects have ever been attributed to wireless LANs.

### **2.7. Making a Secure Wireless Transaction**

With the correct infrastructure in place a mobile phone could quickly gain acceptance as a personal trusted device, enabling the user to make secure payments in a multi-channel environment. Such a development would, in turn, see the traditional mobile operator evolve into a 'Trusted Operator'. A Trusted Operator is the key to making secure wireless transactions possible between end users (mobile users) and content/ service providers. Using digital signatures and end-to-end confidentiality, Trusted Operators are able to open up mobile e-services that require a high level of confidentiality. These

include banking, payment and voting applications as well as secure remote access to corporate networks. [26]

A central feature of this security is the use of the SIM card as a storage device for crypto keys. SmartTrust's Trusted Operator concept supports almost any SIM-enabled device, allowing operators to target such services to an entire subscriber base. The following are key considerations for an operator wishing to deliver secure transactional-based services today:

- Ability to deploy services to an entire subscriber base
- High-end security
- Ability to deploy true end-to-end confidentiality
- Ability to offer complete enterprise PC security services
- Need for high-end application interface

Making the most of an existing infrastructure and subscriber base is extremely important for operators. It is also vital to have SIM card support that allows applications to be developed and reached by any SIM-enabled device within the network.

The security offering from SmartTrust leverages traditional technologies used within today's GSM networks, PKI (Public Key Infrastructure) or a combination of both. The only basic requirement is the use of the SmartTrust specified WIB on the SIM and associated plug-ins. The WIB (wireless internet browser) is implemented by all major SIM suppliers. [26]

It is possible to tailor the level of security depending on the requirements of the content provider. At the top of the spectrum is the ability to securely transmit confidential information and leverage legally compliant digital signatures.

The mobile handset is not just a phone, but also the world's most commonplace smart card reader. The SmartTrust security solution allows any compliant PC application to utilize the SIM's signing keys. This makes it possible to use a handset as a locally connected device, for the signing of e-mails and strong client authentication during remote access among other security related tasks.



Many content providers believe they do not require PKI-level security. However, the lifetime cost for PKI is usually lower than that of most other security concepts due to PKI's easy manageability and scalability. Thus, when the inherent complexity of PKI is reduced it becomes a highly attractive alternative. Offering a high-level application interface makes PKI easy to use, and allows content providers to develop and deploy applications quickly. [26]

It's important to make sure that any infrastructure investment offers a high-degree of protection against obsolescence. That's why SmartTrust bases its solutions on open standards and the WIB, a globally approved USIM concept. The Trusted Operator concept can be broken down into two distinct categories, Infrastructure and Application. The infrastructure serves as the foundation on which secure transactions can flow. The procedures for setting up the security infrastructure depend on whether it is symmetric or asymmetric (PKI). PKI requires a party acting as a Certification Authority (CA) to be in control of certificate management; a trusted third party often handles this role. Issuing certificates can be achieved in a batch or an on-line environment. A combination of both methods may also be used. The traditional security model in a GSM environment is based on symmetric keys making the secure distribution of keys to contracted content providers an important feature. The symmetric mechanism can be based on a derived, or matching key concept; several distribution models are available. The derived key model is considerably easier to maintain and if this model meets the necessary security requirements it should be used.

The prerequisite for the concept is that the SmartTrust Delivery Platform, with its SDM (Service and Device Management) and WIG (Wireless Internet Gateway), is installed and that WIB (Wireless Internet Browser) enabled SIMs with associated plug-ins are available. [26]

## **2.8. Summary**

This chapter was all about the Internet and wireless networks securities that can not be separated away from the WAP security. The most important security risks had been discussed and supposed their remedies. The security concerns about websites.

SET (Secure Electronic Transaction) had also been under investigation. The wireless LANs and the wireless technology had also been discussed. And how to get a secure wireless network also had been discussed.

### **3. WAP SECURITY**

#### **3.1. Overview**

Security of applications and computer systems is an issue that is so important and always must be under revision. As corporations have utilized technologies, such as remote access, Java and component technologies, and infrastructural advances like the Internet, to facilitate new ways of working, new ways of doing business with clients, partners and suppliers, and even to create entirely new products, services and business models, the need for mechanisms to secure applications, networks and systems has become more and more important.

WAP is another technology that extends the reach of communication networks, provides new opportunities for innovative corporations, and adds to the complexity of the environment within which applications need to be designed, built and deployed. There is a set of concerns over how secure WAP is as a technology, and whether it is robust enough to implement mobile commerce applications, and other applications with stringent security requirements. [1]

#### **3.2. Background**

Before beginning investigation of WAP security, it is worth noting that there is no such thing as a secure system. The phrase 'secure system' means one that cannot be compromised or accessed without authorization. Considering that hackers who set out to compromise or penetrate systems are resourceful and always target unexpected aspects of the systems, it would be a brave fool who declared a system to be immune to attack. What can be said is that a particular system meets certain predefined security criteria in that it can withstand attacks of a known type, and is therefore considered secure enough for its intended purpose. [2, 8]

To make a decision whether WAP is 'secure' or not is disappointed. It is only feasible to make the assertion that WAP is or is not 'secure enough' for a particular application when you understand the security requirements of that application, the environment in which that application is to be deployed, the likelihood that the application will be subject to attempts to compromise its security, and the nature of the attempts that are likely to be made. Even then the statement is only valid until something changes in the



environment, or someone discovers a new security exposure in the network, the environment, the technologies used or the platform on which the application is deployed. The facilities and technologies that WAP has to offer for building and deploying secure applications, will be investigated in following sections of this research. [8]

### **3.3. What Security Is About**

Let us begin with the investigation of the topic of security with a discussion of what security is about and why it matters. In this section we will investigate:

1. The importance of security in mobile applications.
2. The role of security in protecting data and systems.
3. The basic issues which security solutions of all types need to address

#### **3.3.1. The Importance of Security**

Most people are aware of the need for securing information such as credit card numbers, but the need for security in both the wired and wireless environments is much broader than that.

At the moment, information often has a commercial value. Many dot-com organizations make money through the sale or re-sale of information. This is not a new thing, newspapers have been doing it for centuries, but the new channels for this kind of commercial activity have lowered the barriers to entry and increased the amount and the value of the information available.

Information can also be sensitive for many reasons like ranging from a justifiable desire for privacy to information that is sensitive on a national security level. Sometimes the sensitivity comes from the content of the information, at other times the timing of the information. For example, it is unacceptable to allow some stock market investors to become aware of an impending profits warning from a company before others, so the information is regarded as sensitive until it is published formally to all investors. [2]

The power associated with information must also not be underrated. Some organizations have legal obligations to safeguard certain items of information. In some cases divisions

within organizations are subject to similar constraints. There are many examples of information that are intrinsically powerful, for example, information about military weapons. Along with all of the sensitivity that naturally accompanies information, there is a growing need to communicate digitally, because of the speed and convenience of doing so. However, in certain ways these digital communications are more vulnerable to compromise. Two major weaknesses in digital communications arise from the fact that it is notoriously easy to intercept digital messages, and the fact that it is notoriously difficult to establish identity conclusively in an online environment.

All of this leads us to two inevitable conclusions that drive the need for robust security implementations: computer systems are critical to the operation of almost every society on earth; and computer systems are very vulnerable to abuse.

### **3.3.2. The Role of Security**

Security is both an enabling and disabling technology. Its purpose is to enable communications and transactions to take place in a secure environment without fear of compromise, while at the same time disabling non-legitimate activities and access to information and facilities. Non-legitimate activities include eavesdropping, pretending to be another party (also known as spoofing), or tampering with data during transmission. In general these activities are either unacceptable or illegal outside of the digital environment, so security simply helps to enforce the status quo in that sense. [2]

### **3.3.3. The Basic Issues**

There are a number of basic issues around security that have to be addressed. Almost all of these have parallels in the real world, and often the solutions are based on, or similar to, real-world solutions. These basic issues are:

- Authentication; being able to validate that the other party participating in a transaction is who the party claims to be, or a legitimate representative of that party.
- Confidentiality; being able to ensure that the content and meaning of communications between two parties do not become known to third parties.
- Integrity; being able to ensure that messages received are genuine and have not been tampered with or otherwise compromised.

- Authorization; being able to ascertain that a party wanting to perform some action is entitled to perform that action within the given context.
- Non-repudiation; being able to ensure that once a party has voluntarily committed to an action it is not possible to subsequently deny that the commitment was given by that party. [1, 10]

### 3.3.4. Concepts

Familiarity with some concepts relating to digital communications and to security are required in order to understand the points made later in this research, and the place within the communications process of the existing security solutions.

### 3.3.5. Protocol Stacks

There is an industry standard theoretical protocol stack that was developed by the Open Systems Initiative (OSI) many years ago, in part to facilitate a common understanding of the functionality provided by a protocol stack and to facilitate comparisons between different vendor's implementations. This stack is shown in diagram 3.1.

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

**Diagram 3.1: Protocol Stacks**



The 'bottom most' layer of the OSI stack, Layer 1 or the physical layer, defines the properties of the physical medium through which communications are transmitted and the characteristics of electrical transmission through that medium.

*data link*

Above that is Layer 2, the data link layer. The data link layer is responsible for the transmission of data over the physical medium and also for the addressing of devices on the network.

*network*

The third layer is the network layer, which is responsible for network addressing and for the routing of data between networks.

*transport*

The transport layer is the fourth layer and is responsible for preparing data for transmission across the data-link. This includes such functions as segmentation and reassemble of packets of information, and also sequencing of packets and retransmission of packets that get lost or corrupted.

*session*

Layer 5 is the session layer, which is responsible for establishing and maintaining sessions between two devices across a network. What exactly this entails depends on the protocols involved.

*presentation*

Above layer 5 is the presentation layer, which is responsible for translation and reformatting of data that is transmitted or received over the network. This helps to facilitate communication between computers that are based on different architectures and which utilize different information representation schemes.

*application*

The last layer, layer 7, is the application layer, which is responsible for identifying requests for remote resources and for the reformatting of those requests as remote requests. This allows applications to operate independently of the location of the services that they utilize. [1, 10]

*summary*

Although the details of the actual protocol stack will vary depending on a whole host of factors (such as the type of network, where the client or server device resides) in general

in the wired world certain participants in the stack are more common and more typical than others. The mapping of the wired protocol stack onto the OSI model is as follows:

Physical layer - UTP or co-axial cable

Data link Layer - Ethernet, Token-ring, FDDI or PPP

Network layer - IP

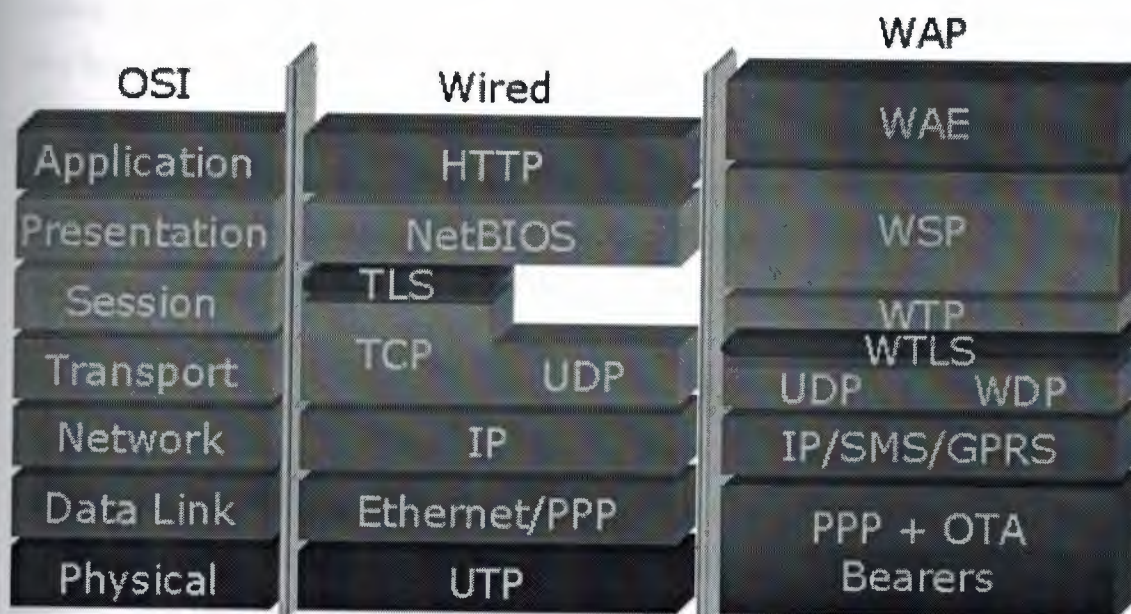
Transport layer - TCP or UDP

Session layer - TCP

Presentation - varies, but could be NetBIOS or XDR

Application layer - depends on the service being invoked; a typical example is HTTP

In the wireless world a similar kind of mapping exists, although with different protocols at each layer. The mapping of both the fixed-wire and WAP protocol stacks is shown in diagram3.2.



**Diagram3.2: The mapping of fixed-wire and WAP protocol stacks**

The WAP protocol stack contains the following elements:

Physical and Data link layers - depends in part on the type of wireless network, but with WAP it will be PPP over one or more over-the-air bearer protocols.

Network layer - IP is the network layer protocol of choice, although not all wireless networks are capable of transmitting IP, so SMS or some other non-packet network protocol may be used.



Transport layer - the transport layer protocol of choice is UDP, but it may not be feasible over non-IP networks. For this and other reasons, WAP defines an additional transport layer protocol, WDP, which can be used where UDP cannot.

Session layer - In the wireless world some of the functionality of the session layer is incorporated into WTP, while other aspects are included with WSP.

Presentation layer - this functionality is included in WSP.

Application layer - some aspects of application layer functionality are taken care of by WSP, whereas others are implemented in the Wireless Application Environment. [1, 8]

### **3.3.6. Encryption**

Cryptography is the study of encryption, or the science of encoding data into another format that cannot easily be decoded or understood, using some sort of mathematical algorithm. The mathematical algorithms are based on an intractable (difficult to solve) problem. There are two of these problems that are commonly used for encryption: one is finding the prime factors of a very large integer; the other is finding the logarithm of a very large number to a known base. [2]

Developing and proving the robustness of an encryption algorithm (called a cipher) is extremely difficult, so there are relatively few of these algorithms in existence. If everyone used the same few algorithms their effectiveness at concealing information would be severely limited, so the algorithms use keys, which are strings of bits, to 'customize' the behavior of the algorithm. What this means, in effect, is that the same algorithm can be used to encode the same original information twice using two different keys and produce two completely different encoded forms. This helps to make these algorithms useful for multiple people from the point of view that in order to decode the message both the algorithm and the key have to be known. [2]

In general, the strength of the algorithm (usually defined in terms of how much effort is required to decode an encoded message) depends on the length of the key.

Unfortunately the relationship is not actually that simple, because keys of equivalent lengths can provide different levels of protection when used with different algorithms. Therefore there is no general rule about how long a key should be, although some guidelines do exist for various algorithms. The problem with these guidelines is that as



computer power increases the ease with which algorithms can be cracked also increases, so it is necessary to be constantly aware of advances in this area. [2]

All cryptographic algorithms, because of their computationally intensive nature are computationally expensive, which is a nice way of saying that they are slow on most computers. This has implications in most applications, where processing power is not unlimited and where response times count. However, it is also true that not all algorithms are equally computationally expensive. [8]

In particular, there is a class of ciphers that are particularly expensive, but which provide some particularly useful features. These are called asymmetric ciphers. Their less computationally expensive counterparts are called symmetric ciphers. Symmetric ciphers make use of the same key to both encode and decode the data. [2]

The problem with these types of ciphers is that both the party encoding the message and the party decoding the message need to have a copy of the key, and finding a secure way to exchange the key is an intractable problem in its own right. Asymmetric ciphers make use of a complex mathematical property of the underlying algorithms that allows two different keys to be used - one for encryption, and one for decryption. The key that is used for encryption is known as the public key, and is derived from the private key, which is used for decryption. This arrangement means that there is no need to exchange keys, as the public key cannot be used for decryption, so it doesn't matter if it falls into the wrong hands. The private key has to be carefully guarded, but this is relatively easy to achieve, as there is no need for anyone other than the rightful owner to be given access to the key. [3]

One way that we can address some of the performance issues associated with encryption, yet still make use of the most robust encryption methods available, is to make use of symmetric ciphers for most encryption and asymmetric ciphers to facilitate the exchange of the symmetric keys. In fact it is a little bit more complex than this, because these mechanisms of key exchange are often not used to exchange the symmetric key itself, but are instead used to exchange a piece of information called the pre-master secret, which is exchanged in encrypted format using asymmetric

encryption. This pre-master secret can be used in conjunction with public and private keys to generate a secret key that is used for the symmetric encryption. The means by which this is achieved is quite clever, but it will not be explained here because there isn't enough space to go into all of the mathematics and the detail of how the ciphers work, which would be required to understand how it is done. [3]

### 3.3.7. Certificates

Certificates are a convenient place for storing and managing public keys. They also form the basis of authentication in digital communications, being the digital equivalent of a passport. Like a passport, they have to be issued by a recognized authority and contain certain things that allow the subject's identity to be confirmed and the certificate's validity to be ascertained. The former is achieved by including some identifying information on the subject, along with the subject's public key. The latter is achieved by certificates being issued by a recognized Certification Authority, and being digitally signed by that authority. The Certification Authority's signature is widely and publicly available for use in validating the certificate. [24]

Digital signatures are based on hash algorithms (also called message digests), which produce a 'digested' version, called the hash code, of the text that they take as input. The hash function is deterministic, which means that the hash value that it produces is dependent on the text that it takes as input in such a way that any alteration in the text produces a significant change in the hash code. A good hash function is also a one-way function, meaning that the function cannot be derived from the hash value and the input text, and it is also collision resistant, which means that no two input values should produce the same hash value. Digital signatures are based on a special type of hash function that takes a key as input, as well as the original text. This means that the hash value is dependent on both the input text and the key, and therefore if you and I both sign some text using our own keys, the hash value produced will be different. In this sense digital signatures are slightly unlike real-world signatures, in that they will vary depending on the content that is being signed, which also makes them almost impossible to forge. [24]





Certificates are fairly complex documents, and are usually presented and validated on behalf of the user without any human intervention. This has two ramifications: The certificates end up stored on computer, floppy disks, etc.

It is impossible to track down copies of certificates if it becomes necessary to change or replace one.

The first of these issues causes some problems in the wireless-world, which we will investigate later on. The second is addressed by means of Certificate Revocation Lists (CRLs). These are lists that are maintained by the Certification Authorities of certificates that have been issued, but that have become invalid for some reason or another. CRLs should be consulted before simply accepting a certificate as being valid.

Because of the large universal need for certificates, it is not feasible for a single organization to be responsible for the administration of all certificates, so there is a facility whereby certification authority can be delegated to other organizations. Any organization, theoretically, can act as a certification authority, and many organizations fulfill that capacity for certificates used internally, for example by employees. However, certificates that are valid in the public domain have to be certified by a recognized authority. Certificate chains make this feasible; by chaining certificates to the certificates that certify their authenticity a trail is built back to some authority that can be deemed to be acceptable.

### **3.3.8. WTLS**

WTLS is the Wireless Transport Layer Security protocol. As can be ascertained by the name, it operates at, or more correctly just above, the transport layer in the OSI protocol stack. It is based on transport layer security (TLS), which is the de facto security implementation on the Internet. It works by establishing a session between a client and a server (which in the case of WTLS is the WAP gateway), during which it negotiates security parameters to be used to protect the session. These include the encryption protocols to be used, signature algorithms, public keys, pre-master secrets, or the exchange of certificates, depending on the capabilities of both the client and the server and the required level of security. The process of establishing a session is called the handshake. Once a session has been established all communications between the mobile



device and the WAP gateway are encrypted, and therefore should be unintelligible if they are intercepted. [8]

WTLS includes support for both a full handshake, with negotiation of all security parameters, and for a 'lightweight' handshake in which the security parameters of another session are reused. Support is also provided for session suspend and resume, which is useful in a wireless environment where reception quality is not always that good and where connections can easily be lost. The sessions can continue to exist despite a terminated connection and can be resumed on reconnection. Using this facility, it is possible to have sessions that last for days at a time.

The advantages of sessions that can continue to exist for days at a time must be weighed against the security implications of this feature. The longer the session remains valid for, the longer the secret keys remain valid for, and, presumably, the greater the number of messages exchanged that are encrypted using this key. This all provides material to someone wanting to crack the security protecting the session and compromise the messages. To guard against this, WTLS allows keys to be renegotiated periodically during a session. *Renegotiating keys is not as computationally expensive as establishing the keys in the first place*, so this is still more efficient than tearing down and re-establishing the session.

Another advantage of WTLS over TLS is that it operates over UDP. TLS requires a reliable transport protocol, in particular TCP, so it cannot be used over UDP. WTLS addresses this shortcoming, and also functions over WDP in the absence of UDP.

Certificates, for all of their usefulness, were not really designed with mobile devices in mind. WAP defines a new format of certificate that is optimized for storage on mobile devices and for transmission over constrained networks. These certificates still provide all of the functionality and security of their more heavyweight counterparts, but rely on the server to perform more of the processing under some circumstances.

WTLS therefore provides a comprehensive, optimised solution for both client and server based authentication using certificates, secure exchange of symmetric keys, anonymous and authenticated encryption of data, and support for digital signing of data.

There are three classes of WTLS implementation defined in the WAP specification. They are:

Class 1: Anonymous key exchange with no authentication.

Class 2: Certificate based server authentication. Server key is anonymous or authenticated, client key is anonymous.

Class 3: Certificate based client and server authentication. Both client and server keys are anonymous or authenticated. [1]

### 3.4. Communication Models

The best way to achieve an understanding of the merits of the implementation of security in the wireless environment is to compare it to the implementation of security in the fixed-wire world, that is, the Internet.

#### 3.4.1. Internet Communication Model

A typical example of the Internet communication model is shown in diagram3.3.

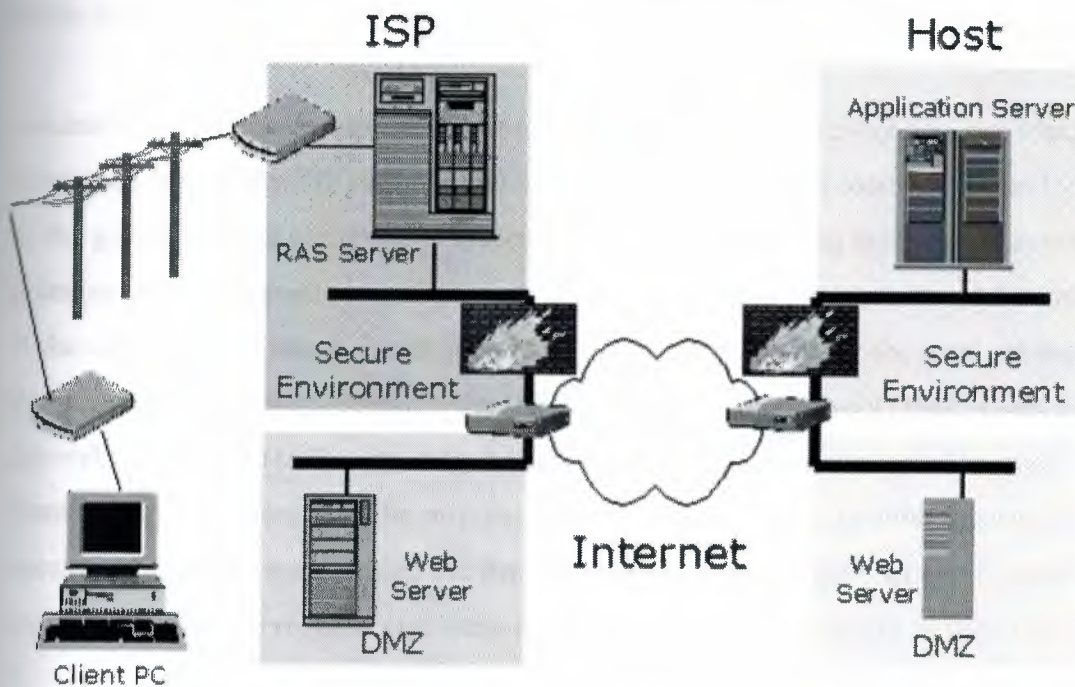


Diagram3.3: An Internet Communication Model.



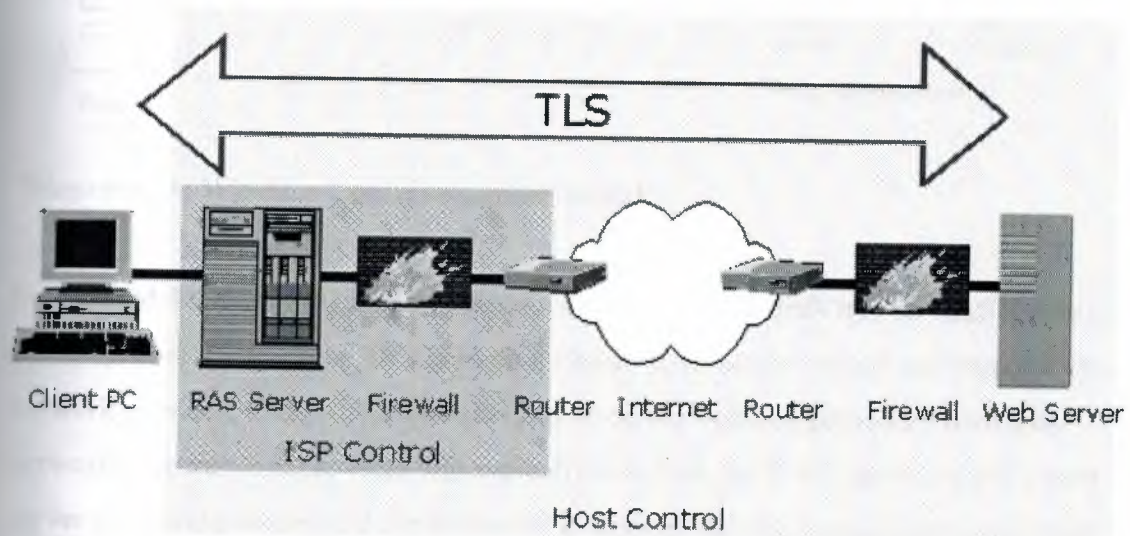
The Internet communication model assumes that a client PC connects to a server via an ISP dial-up connection. The client will be connected into the ISP systems over a PSTN or ISDN link, with PPP usually used as the bearer protocol. The connection point on the ISP network is to a RAS server, which will perform certain functions on behalf of the remote client. In particular, the RAS server effectively acts as a proxy for the remote client, collecting network packets and forwarding them across the dial-up link. The RAS server is responsible for validating the client that is dialing in, and there are various means at its disposal to do that. The RAS server is typically on a secure part of the ISP network and thus provides the illusion to all other devices that the remote client is in fact also on the local network. [15, 23]

The ISP secure network environment is usually isolated from the Internet by means of a firewall of some sort. This firewall will attempt to regulate traffic that enters the local network, and protect the devices on the local network from malicious attacks over the Internet. The ISP may also choose to run one or more web servers and/or other facilities in a way that is more easily accessible to the public, and by extension also more vulnerable to attack. This area of the network is referred to as the demilitarized zone (DMZ), and is usually on a separate network segment from the secure area. Note that this is only one possible configuration for a network. Any particular implementation is likely to be far more complex and to be different in any number of ways. [23]

Access to the Internet is typically facilitated by one or more gateway devices, which are connected both to the ISP network and to some other network, possibly one run by one of the global Internet backbone providers. Any message entering the network across the gateway will be forwarded from gateway to gateway across the Internet, until it arrives at the destination network. It will then cross the gateway and enter the local network of the target host. In a way similar to the ISP, the host may also have a DMZ which houses the web server, with traffic entering the secure network filtered through a firewall. The firewall may only permit traffic originating from the web server to enter the secure network. On the network behind the firewall will reside any additional applications required to fulfil the request, and these will be used by the web server as required.



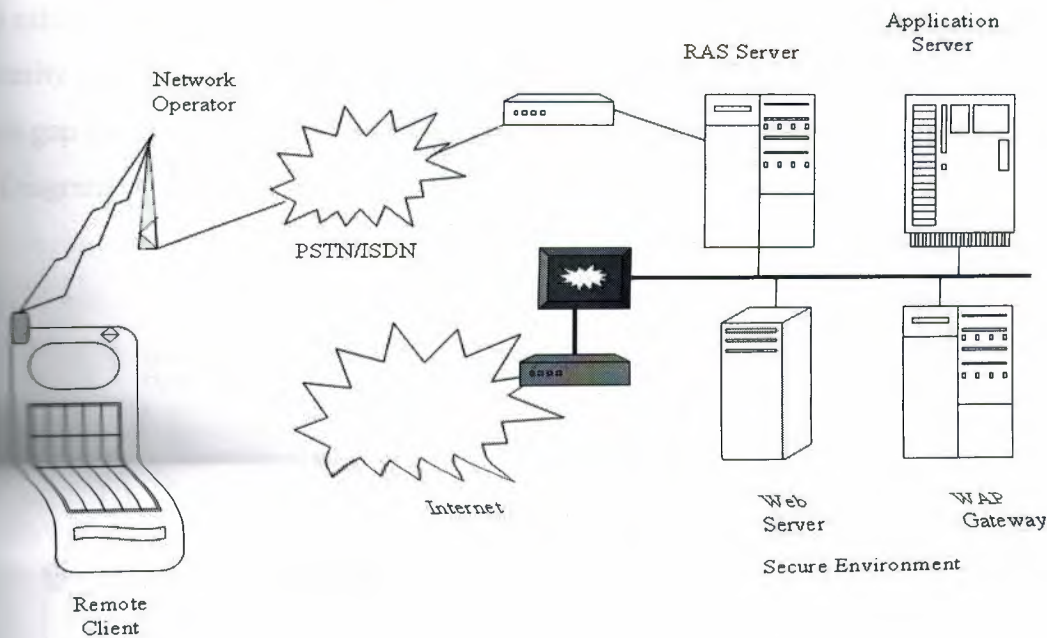
In examining the Internet model from the perspective of who controls or has the ability to influence the connection from a security point of view, it is apparent that the TLS connection exists between the client device and the web server. In effect this forms a tunnel between the client and server, and anyone penetrating this tunnel would not be able to decipher any messages intercepted. The ISP retains responsibility for the devices on its own network and for validating that the client is permitted to connect to the network in the first place, but has no ability to influence the TLS session. The extent of each parties influence is illustrated in Diagram3.4. [8, 23]



**Diagram3.4: The TLS connection**

### 3.4.2. Wireless Communication Model

The wireless communication model is more complex because there are more ways in which the connection could be achieved. The model that we will examine at this point in time is one which many, possibly the majority of, connections that take place between the person-in-the-street and some WAP enabled web site will take place over. That is not to say that this is necessarily the best model from any particular point of view, just that many connections will be effected in this manner. This model is illustrated in Diagram3.5.



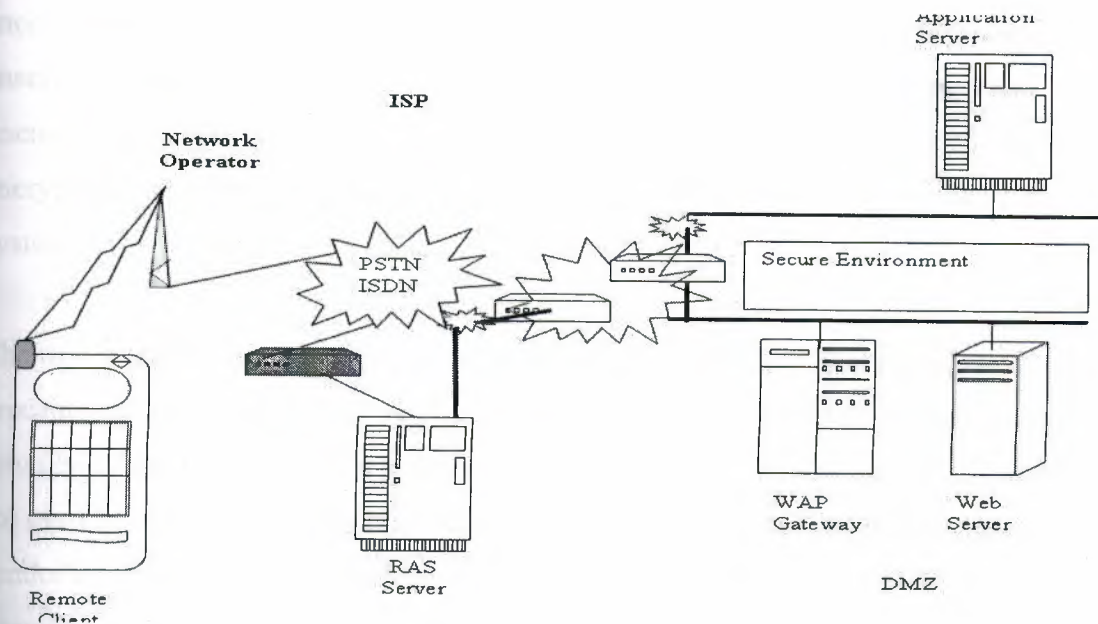
**Diagram3.5: Wireless Communication Model.**

In this model the remote client is a mobile device, but still dials into an RAS server on some network somewhere. This is likely to be an RAS server hosted and owned by the network provider, and is therefore likely to be on the network provider's own local network. The network provider will typically also host the WAP gateway, and a web server to provide access to the premium rate services that the network provider offers to their members. If access is required to services hosted on another server somewhere across the network, then the WAP gateway will act as a proxy for the client mobile device in establishing the required sessions with the remote host.

From the point of view of security, this scenario has various implications. WTLS is the security protocol that will be used to secure communications to and from the mobile device, but the mobile device's session is necessarily with the WAP gateway rather than the remote host's web server. At the gateway, the secure session terminates and all encrypted material is decrypted. Should there be a requirement for a secure session for communication with the web server, it will be established by the WAP gateway on behalf of the mobile device. The WAP gateway will use TLS to establish such a secure session. While TLS is obviously a robust security protocol, it remains a fact that the secure session is not between the mobile device and the web server. There are actually two secure sessions in play: one between the mobile device and the WAP gateway and

the other between the WAP gateway and the web server. This means that there is a security gap, in which the data is not encrypted, at the WAP gateway. [8]

This gap and the span of control of the host server and network operator are illustrated in Diagram3.6.



**Diagram3.6: The gap and the span of control of the host server and network operator.**

The host server's span of control is severely compromised in comparison to the Internet model. In fact, the host has absolutely no control over the security that exists between the mobile device and the WAP gateway. The host also has limited control over the TLS session between the WAP gateway and the web server, and will be limited to providing security that does not exceed a level determined by the network operator. This may or may not be adequate for the host. [8]

### 3.5. WAP Security Issues

There are two issues with regard to security in the WAP environment. There are ways of addressing both of these issues, but they both remain issues that need to be addressed.

#### 3.5.1. The Gateway

We have established that there is a security gap in the WAP model in the form of the WAP gateway. Because of the way that WAP works it is not feasible to do away with



the gateway, so we need to establish to what extent it actually is a risk and what the alternative ways of addressing the risk are. It can be argued that the WAP gateway is not actually a security risk because the gateway vendors are aware of the issue and therefore take steps to ensure that the process of decrypting from WTLS and re-encrypting into TLS cannot easily be compromised. Typical of the steps taken will be to ensure that the decryption and re-encryption takes place in memory, that keys and unencrypted data are never saved to disk, and that all memory used as part of the encryption and decryption process is cleared before being handed back to the operating system. [8]

The first problem with all of this is that there are no standards or guarantees about these precautions. You have no way of ascertaining how robust your vendor's implementation actually is, and in the case of a gateway that is hosted by a network operator you may not even be able to tell whose implementation it is. One can also question the vendor's promises: in a heavily loaded server, how exactly does the gateway prevent the operating system from swapping memory pages out to swap space?

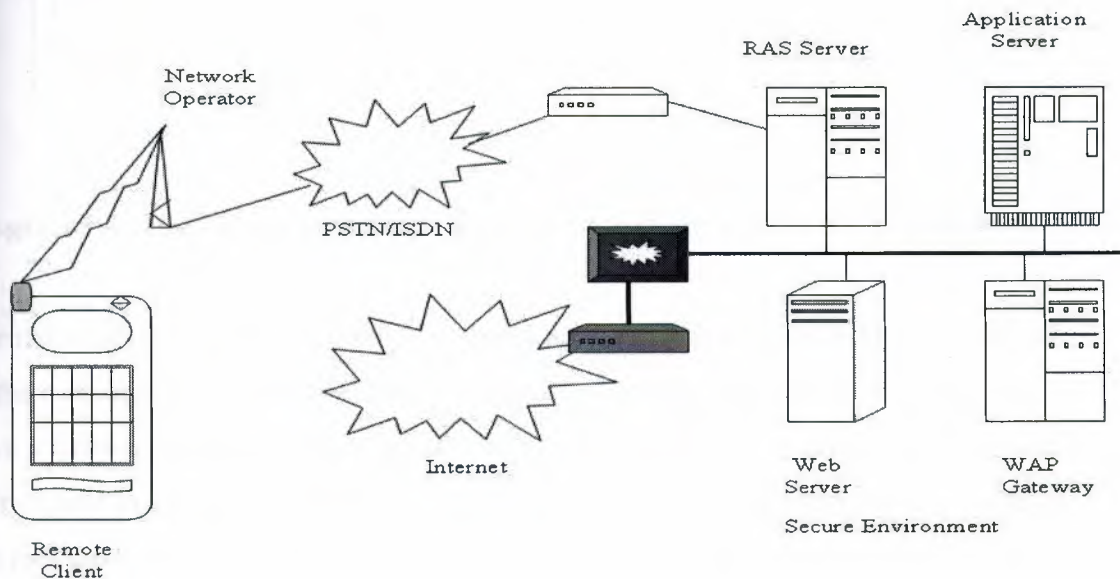
In a sense, saying that the vendors are aware of the issue and taking steps to address it is comforting. However, it must also be remembered that Microsoft are aware of the security exposures of the Internet Explorer browser and the Outlook mail client, but they continually take flack as a result of a seemingly unending stream of security loopholes and exposures that are constantly being discovered and exploited. I am sure that a vendor that has extremely competent programming staff and designers, and which implements their product only on a very secure operating system in a thoroughly secured environment under the control of extremely competent administrators, could provide a reasonably secure implementation. Still, I am equally sure that there is still an exposure around the gateway and that sooner or later it will become a target for hackers.

What you need to consider is how much of an exposure it is for the kinds of applications that you are developing. For some applications the risk-reward ratio, when compared to the cost of implementing a more secure solution, may be small enough that the vendor decides to take the risk. For others, where the risk by far outweighs any possible reward, there is no question that it is a complete show stopper.

If we accept that there is an exposure at the gateway, no matter how small or how hard the vendors work to protect the unencrypted data, the real question then becomes: who hosts the gateway? Whoever hosts the gateway has the responsibility for protecting it and the data that goes through it, and also has access (potentially, at least) to all of the data that goes through the gateway in unencrypted form.

The good news is that it is entirely possible for you to host your own gateway, although before doing so you should consider the implications, in terms of cost and otherwise, of doing so. There are also two different architectures that can be implemented to facilitate hosting your own gateway, and each has different characteristics in terms of security and cost overheads.

The first model, which is shown in diagram3.7, is probably only suitable if you want to provide access to a limited number of people who are not the general public, possibly employees:

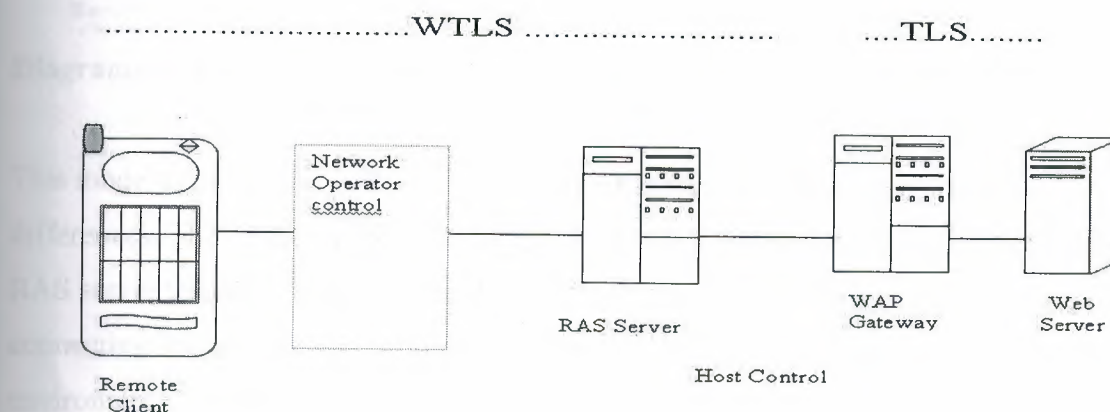


**Diagram3.7: A Sample WAP Model.**

Here, security is absolutely paramount. In this scenario you would choose to establish an environment similar to any other highly secure dial-up environment. You would establish a bank of dial-up modems connected to one or more RAS servers on your local network. You would be responsible for establishing, maintaining and administering the

environment, including details such as dial-up security. You would then be able to strictly control who has access to the gateway, when this access is possible, and via what telephone numbers. You could implement dial-back to a limited set of numbers, control the IP addresses available, issue and use your own certificates for authentication, and anything else that would contribute to your secure environment. All of the relevant servers would be a secure segment of your local network, and access to and from the Internet may or may not be available. If it is available it will almost certainly be protected by one or more firewalls. [25]

In this environment, as illustrated in Diagram3.8, the network operator's sphere of influence is almost non-existent:

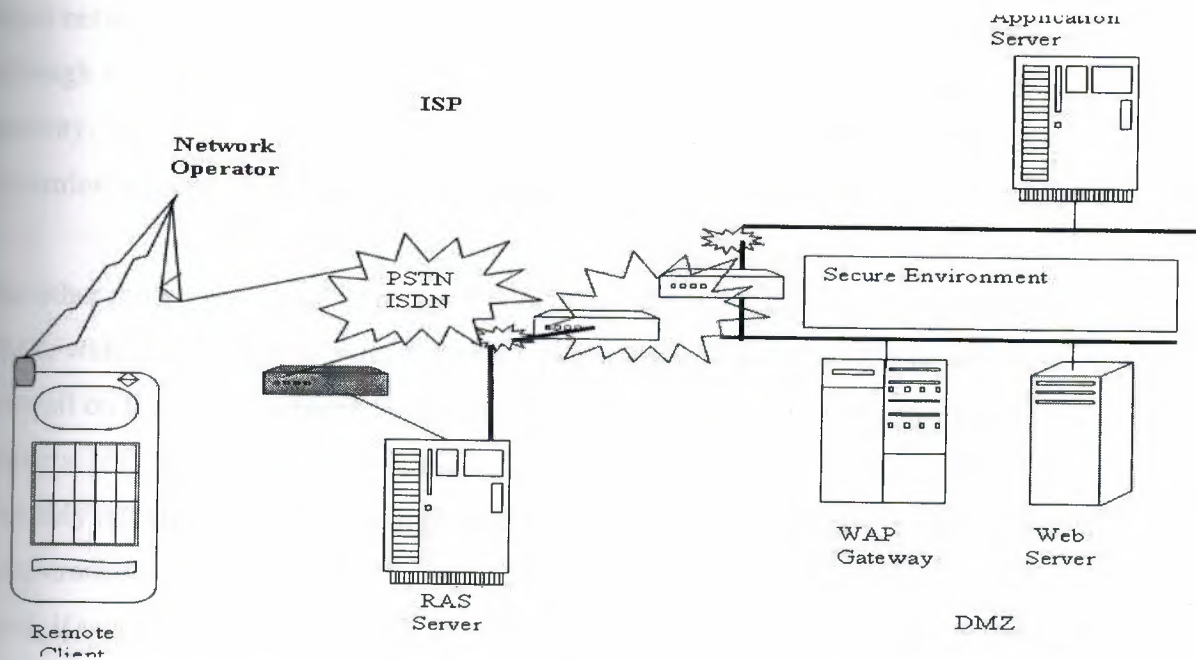


**Diagram3.8: The network operator's sphere of influence is almost non-existent.**

The network operator is restricted to connecting the call and has no influence over any of the communications between the client and server. The network operator does not have a gateway that participates in the communication process, and has no role to play with regard to security. The mobile device establishes a WTLS session that tunnels through the RAS server to the gateway, and a TLS session from the gateway to the web server, all on your own secure network.

The second model eliminates the need for the modems and RAS server by making use of the services provided by an ISP. The diagram3.9 shows this model:





**Diagram3.9: RAS server by making use of the services provided by an ISP.**

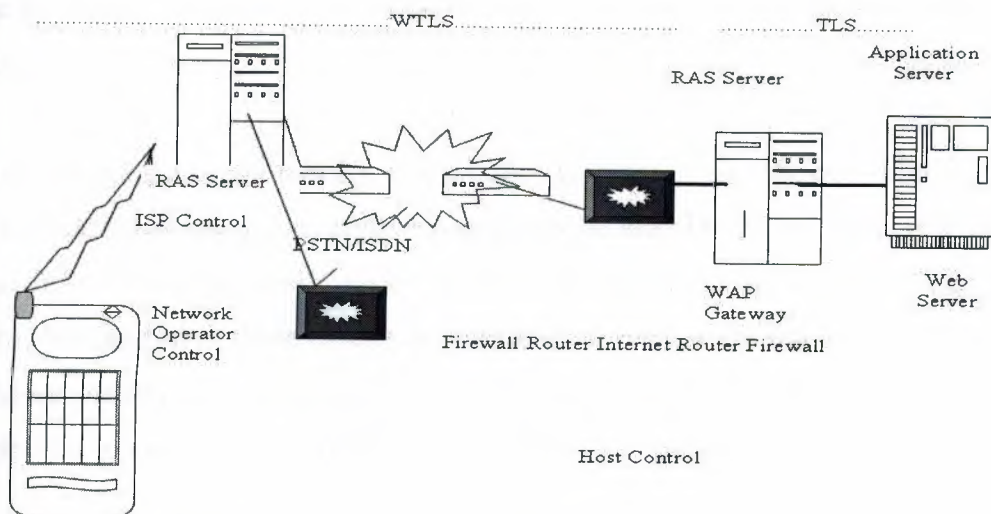
This model is in fact very similar to the Internet model, although there are some differences. The remote mobile device will establish a dial-up connection with the ISP's RAS server through a modem hosted by the ISP. The network operator is restricted to connecting the call and has no further influence on the session or the security environment. The RAS server at the ISP acts as a proxy for the mobile device on the ISP's network, and provides all the services that it would to a fixed-wire dial-up client. The ISP network is connected to the Internet via a gateway and is protected by a firewall. [25]

The host's environment would usually be similar to an environment for access by fixed-wire clients over the network. The major difference would be that the host would have a WAP gateway available on the network, typically in the DMZ. Any secure connection from the mobile device would establish a secure session that tunnels through the ISP's RAS server to the WAP gateway. The WAP gateway would then establish a secure TLS session through to the web server, which would make use of services on the application servers hosted on the secure network behind the firewall. [25]

In this scenario we are making use of WTLS in a similar way to a Virtual Private Network (VPN), in that the mobile device establishes a secure tunnel through to the

target network. In the case of a VPN, the tunnel is typically to the router on the network, although it doesn't have to be, whereas in this model the 'VPN' tunnels to the WAP gateway. You will need to examine the security requirements of your application to determine whether WTLS provides a secure enough 'VPN' for your application. [25]

The other thing to be aware of in this model is that the WAP gateway is typically on the DMZ, which means that it is not as heavily protected as it would be if it were behind the firewall on the secure network segment. This makes it more vulnerable to attack by hackers. If there is little chance of the WAP gateway being targeted then this is probably not an issue, but for a large retail bank, for example, where the gains to be had from cracking the gateway may be significant, it may present a temptation. On the other hand, if you have to provide public access to your WAP gateway then there is little in the way of feasible alternatives, unless you want to become a network provider in your own right. The span of control of the network operator, ISP and host are shown in the diagram3.10.



**Diagram3.10: The span of control of the network operator, ISP and host.**

For almost all applications that have security requirements that prohibit the use of a network provider's gateway, one of these two models will almost certainly be sufficient. The trick is to match the trade-offs, in terms of cost and overheads, against your security requirements and risk to achieve an optimal solution. [25]

### 3.5.2. User versus Device

The second issue that is worth considering with mobile devices, and which is not really a consideration for fixed-wire devices, is the issue of whom or what is being authenticated by the certificate. A certificate is a reasonably large and complex thing, certainly too complex to type in each time it is required, had been previously mentioned. The result is that the certificate usually ends up being held on your computer, often without you even being aware that it is there, and the system will take care of presenting and validating certificates as and when required. While this is very convenient, it does have some security implications, in that anyone who gains access to your computer can make use of your certificates. The prerequisite is for the person to gain access to your computer. In many cases this is not that easy to achieve, requiring breaking and entering or something similar.

Mobile devices are different in that they are mobile and are therefore carried around. Where access to data or services has to be strictly controlled it will not be an acceptable solution to store certificates on the phone if those certificates provide access to data and services.

The most immediate way of tackling this problem is to accept that the certificate is going to be stored on the phone, and the phone may be lost. The certificate is still made use of to validate that the mobile device is entitled to access the network, which at least serves to eliminate all of those mobile devices that do not have the required certificate. Once the mobile device is reported missing the certificate can be placed on a certificate revocation list to ensure that it does not provide access in the future.

To further validate that the current user of the authenticated device is the rightful user you can make use of a variety of systems, which vary in their complexity and robustness from a simple PIN number through to a SecureID token. While it is easy to dismiss a PIN as being inadequate, pause to remember that almost all of us make use of automated teller machines, and in doing so daily rely on simple PIN numbers to protect our financial assets.



### **3.6. Future**

It is always difficult and risky to gaze into the future and predict what is coming down the line, but it is also necessary to make educated assessments of the current technology and what is likely to be addressed in the near future.

#### **3.6.1. WTLS**

WTLS, being based on an established and stable standard, which is the minor subject of this research, is going to be investigated and tested later, in chapter 4. Consequently, we could get an idea about the WTLS future.

#### **3.6.2. End-To-End Security**

The WAP Forum has made it clear that they are aware of the issues around the security gap at the WAP gateway. They have also made it clear that they intend to plug the gap by providing an end-to-end security standard in a subsequent release. There have been hints that they would attempt to address this through changes to WTLS, but I think that this is marketing rather than technology speaking. The issue does not arise because of any weakness in WTLS and is caused solely by the position that the gateway fulfils in the WAP communications chain. In order to address the issue, either the gateway has to be eliminated or some other solution has to be implemented, probably at a higher level in the protocol stack. The WAP Forum has also indicated that the WMLScript Crypto library may be extended in the future to include cryptographic functions. At this point in time there is only a function that supports signing data. To my mind, it seems logical that the way to implement end-to-end security is by means of encryption functionality at the application level. A necessary prerequisite for this, however, will be the capability of mobile devices to deal with the processing loads associated with encryption functions. Part of the solution to this problem may actually lie in the WIM. [1]

#### **3.6.3. WIM**

The Wireless Identity Module specification is new in the WAP 1.2 specification. It provides a means to offload the storage of keys and of cryptographic functions onto what is described as a tamper proof device. This is basically a smart card, although it could also be a SIM. The specification covers only the low level capabilities of a WIM in the current specification, and doesn't present an API for making use of a WIM when

release. The introduction of the WIM could help to address the issues around authenticating the device as opposed to the user. [1]

### **3.7. Summary**

This chapter concerned of what was the aim of this thesis, it was about the WAP Security in major and the WTLS in minor. It started with an overview then a background then to what we have secure and why. Communication models of the internet and the WAP had been investigated. WAP security issues like the gateway and the user versus device also had been investigated. After all, it was a look at the future of the WTLS, END to END Security and the WIM.

## 4. SECURITY IN THE WTLS

### 4.1 Overview

The WAP (wireless transport layer security) WTLS protocol was designed to provide authentication, data privacy and data integrity for wireless terminals. It is based on the widely used TLS v1.0. *The requirements of the mobile networks* must be considered when designing WTLS. The low bandwidth, datagram connection, limited processing power and memory capacity, and cryptography exporting restrictions had been considered. These days the protocol is fielded and it is estimated that the protocol will be contained in millions of devices in the coming few years [1]. Consequently this poses the demand for the end-to-end secure connection.

Even though the WTLS is closely modeled after the well studied TLS protocol, it is not so hard to identify a number of serious problems in it. Most of these problems will be mentioned and discussed. The WTLS [1] (Wireless Transport Layer Security) protocol is the security layer of the WAP (Wireless Application Protocol). The authors of WTLS took TLS and tried to add datagram support, optimize the packet size, and select fast algorithms into the algorithm suite.

### 4.2 Introduction

The increasingly growing wireless market increases the demand on the value added mobile services, which the Wireless Application Protocol had been designed for. The WAP defines a set of protocols in transport, security, transaction, session, and application layers to enable a creation of advanced mobile services. It had been developed by the international organisation called WAP Forum; it also makes possible the mobile subscriber to take an advantage of the WAP services all around the world regardless of the local mobile network technology. The WAP is independent of the bearer services. The security layer in Wireless Application Protocol is the Wireless Transport Layer Security. The primary goal of the WTLS is to provide privacy, data integrity, and authentication for WAP applications. The security is needed in order to safely connect to the services, such as online banking and e-commerce. The client and the server must be authenticated and the connection has to be encrypted. the data should not be modified during the transfer if man-in-the-middle attacks occurred, which means that man-in-the-middle attacks must be prevented. The subscriber must be assured that the service he is using is the required one and it is the safest, that urges the service to



use a strong authentication with certificates in some cases. Although the traffic in the air is encrypted in several mobile networks, the complete end-to-end security is not provided by the mobile network. The WTLS provides the transport service interface for the upper level layer. This interface is similar to the transport service interface below the WTLS. The WTLS is based on the well known TLS v1.0 security layer used in Internet, but with a number of modifications and changes were needed because of the nature of wireless networks. The wireless networks require support for both datagram and connection oriented transport layer protocols. The mobile equipment sets requirements for the algorithms because of the limited processing power and memory. In addition, the low bandwidth must be dealt with and the restrictions on exporting and using cryptography must be considered.

The WTLS incorporates new features such as datagram support, optimized packet size and handshake, and dynamic key refreshing. It has been optimized for low-bandwidth bearer networks with relatively long latency. Fast algorithms are chosen into the algorithm suite. The mobile equipment like cellular phone can be constructed to support only a set of cipher suites.

The security of the WTLS version 1.1 will be analyzed. Although the wireless network sets a great deal of requirements, it is possible to provide the acceptable security level. For instance, a security protocol that allows an intruder to eavesdrop the data is not acceptable. On the other hand, the absolute security combined with a good usability is in theory impossible. The number of security problems has already been discovered in the WTLS as a consequence of the modifications. The development work of the WTLS is continued to block up these security holes, thus, giving us better security in the future. The comprehensive security also depends on other factors than the WTLS.

#### **4.3. Data Communication Security**

Data communication security is comprised of smaller security entities. In the following sections, the different entities are introduced and explained as criteria for the analysis of the WTLS. The following descriptions are based on the definitions introduced by Amoroso and Schneider. [4, 9]

#### 4.3.1 Privacy

The main tool for providing privacy is cryptography. A plaintext is simply encrypted and decrypted to implement privacy. If the plaintext is encrypted using a strong encryption, it is almost impossible for eavesdropper to decrypt and read the original content. The requirements of the strong encryption are met when the security is created by using a shared secret, not secret algorithm. The keyspace, from which the used shared secret is chosen, has to be large. Moreover, the used cryptographic method must produce an output which appears random to all statistical tests. Finally, the used method should be resistant to all known attacks. [4, 12]

Encrypted data will be useless if the recipient is not able to decrypt it. The sender and the recipient have to share a method to encrypt and decrypt the data. They both have to know the used cryptographic method and the shared secret. The shared secret is a piece of information known by both parties but no one else.

There is also other kind of privacy, which is not always the case that the information has a recipient. Sometimes there is data that is not supposed to be decrypted by anybody e.g. Unix-type passwords. This kind of encryption method is called a one way encryption, in other words, there is no formula for a reverse process back to the original information. Hash-values are the most common method for the one way encryption.

#### 4.3.2 Authentication

According to Schneider, authentication is defined as: It should be possible for the receiver of a message to ascertain its origin; an intruder should not be able to masquerade as someone else. [9]

Authentication is a technique to ensure that the stated identity of the user is correct. In the beginning, the other party introduces itself and claims to have some identity. This is not enough. The contacted party also needs to know for sure that the contacting party is the one it claims to be. The contacting party has to present some verification to prove its identity. It can be as simple as passwords, or more complicated digital signature or certificate. But then again, the contacting party also wants to be sure that the other end is valid. The contacted party has to present some identification about itself.

After the authentication, the service provider can be sure that the service is available to the user who has correct rights to use the service. On the other hand, the user can be confident about the service provider.

#### **4.3.3 Integrity**

Schneider defines integrity, "It should be possible for the receiver of a message to verify that it has not been modified in transit; an intruder should not be able to substitute a false message for legitimate one." [9]

Maintaining integrity means securing the reliability of the information. We have to figure out a way to prevent unauthorized changes or at least find the means to notice those modifications. Integrity is guaranteed by calculating checksums from the original information to be sent. Of course, just a plain checksum is not enough. We need some sender-related information mixed into calculations e.g. information is signed with the user's digital signature.

In most cases, maintaining integrity is more critical than guaranteeing privacy. It is more important that the information is received unaltered but seen by someone else than somebody has been able to modify it without making out the whole information. For example, bank transactions apply this category. It is embarrassing if someone finds out how much money you have but it is infuriating if somebody steals your money.

#### **4.4 Wireless Transport Layer Security**

The WTLS is designed to provide security in the Wireless Application Environment. The wireless mobile networks pose new challenges for implementing security architecture compared with the traditional connection oriented models like that used in Internet. Security in the WAP architecture should enable services to be extended over potentially mobile networks while also preserving the integrity of user data. The denial of service should also be prevented. The wireless mobile networks set many requirements to the security layer. The existing secure protocols cannot be used in mobile networks without adaptation. [1]



One of the most important requirements is to support for low data transfer rates. For instance, the SMS as a bearer can be as slow as 100 bit/s. The amount of overhead must be kept as small as possible because of the low bandwidth. Compared with the industry-standard Transport Layer Security (TLS), formerly known as Secure Socket Layer (SSL), datagram transport layer must also be supported because of the nature of the wireless mobile network. The protocol should handle lost, duplicated, and out-of-order datagrams without breaking the connection state. [1]

Other issues include slow interactions, limited processing power, and memory capacity. They also include the restrictions on exporting and using cryptography. The round-trip times can be long and the connection should not be closed because of that. For instance, the time between the request and the response using the SMS bearer can be as long as 10 seconds. Used cryptographical algorithms must be light enough so that the mobile terminals are able to execute them. The number of cryptographical algorithms has to be minimized and small-sized algorithms must be used. The amount of available RAM in the mobile terminals must be taken into account. Export laws in some countries do not let strong cryptography to be exported outside the country. For that reason, the best permitted security level, as defined by the legislation of each area should be achieved all the time. There are also differences between exporting a strong authentication and encryption. In many cases, strong authentication is allowed to be used but strong encryption is prohibited. [1]

Briefly, the objective of the WTLS is to be a lightweight and efficient protocol with respect to bandwidth, memory and processing power.

#### **4.4.1 Specification**

The WTLS layer operates above the transport protocol layer and it provides the upper level layer of the WAP with a secure transport service interface. The interface preserves the transport interface below it, and it also presents methods to manage secure connections. [1]

The WAP, by means of the WTLS, provides end-to-end security between the WAP protocol endpoints. Actually the end points are the mobile terminal and the WAP gateway. When the WAP gateway makes the request to the origin server, it will use the

SSL below HTTP to secure the request. This means that the data is decrypted and again encrypted at the WAP gateway. [1]

The complete secure connection between the client and the service can be achieved in two different ways. The safest way for the service provider is to place a WAP gateway in their own network. Then the whole connection between the client and the service can be trusted because the decryption will take a place not until the transmission has reached the service provider's own network, not in the mobile operator's network. When placing the WAP gateway outside the mobile operator's network, the dial-up connection lines are needed. The second way is to include the functionality of the WAP gateway to the origin server. This gives the highest security solution available. [1]

The service and content providers can also trust the mobile operator's gateway and use virtual private networks to connect their servers to the WAP gateway. But then they do not have possibility to manage and control the parameters used by the WTLS at the WAP Gateway. [1]

The negotiating parties are able to decide the security features they want to utilize during the connection. According to the security requirements, the applications enable and disable the WTLS features. For instance, privacy may be left out if the network already provides this service at the lower layer. The connection between two terminals can also be secured by the WTLS. [1]

#### **4.4.2 WTLS Internal Architecture**

The WTLS Record Protocol is a layered protocol which accepts raw data from the upper layers to be transmitted and applies the selected compression and encryption algorithms to the data. Moreover, the Record Protocol takes care of the data integrity and authentication. Received data is decrypted, verified and decompressed and then handed to the higher layers. [1] The Record Protocol is divided into four protocol clients. [1] The protocol stack is shown in Figure 4.1.

<b>Handshake Protocol</b>	<b>Alert Protocol</b>	<b>Application Protocol</b>	<b>Change Cipher Protocol</b>
<b>Record Protocol</b>			

**Fig 4.1 WTLS internal architecture [1]**

- **The Alert Protocol**

The Record Protocol also provides a content type of alert messages. There are three types of alert messages: warning, critical, and fatal. Alert messages are sent using the current secure state, i.e. compressed and encrypted, or under null cipher spec, i.e. without compression or encryption.

If the alert message, labeled as fatal, is sent, then both parties terminate the secure connection. Other connections using the secure session may continue but the session identifier must be invalidated so that the failed connection is not used to establish new secure connections.

A critical alert message results in termination of the current secure connection. Other connections using the secure session may continue and the secure identifier may also be used for establishing new secure connections.

The connection is closed using the alert messages. Either party may initiate the exchange of closing messages. If a closing message is received, then any data after this message is ignored. It is also required that the notified party verifies termination of the session by responding to the closing message.

Error handling in the WTLS is based on the alert messages. When an error is detected the detecting party sends an alert message containing the occurred error. Further procedures depend on the level of the error that occurred.

- **The Change Cipher Spec Protocol**

The Change Cipher Spec is sent to peer either by the client or the server. When the Change Cipher Spec message arrives, the sender of the message sets the current write state to the pending state and the receiver also sets the current read state to the pending state. The Change Cipher Spec message is sent during the handshake phase after the security parameters have been agreed on. [1]



## • The Handshake Protocol

All the security related parameters are agreed on during the handshake. These parameters include attributes such as used protocol versions, used cryptographic algorithms, information on the use of authentication and public key techniques to generate a shared secret. The flowchart of the handshake is shown in figure 4.2.

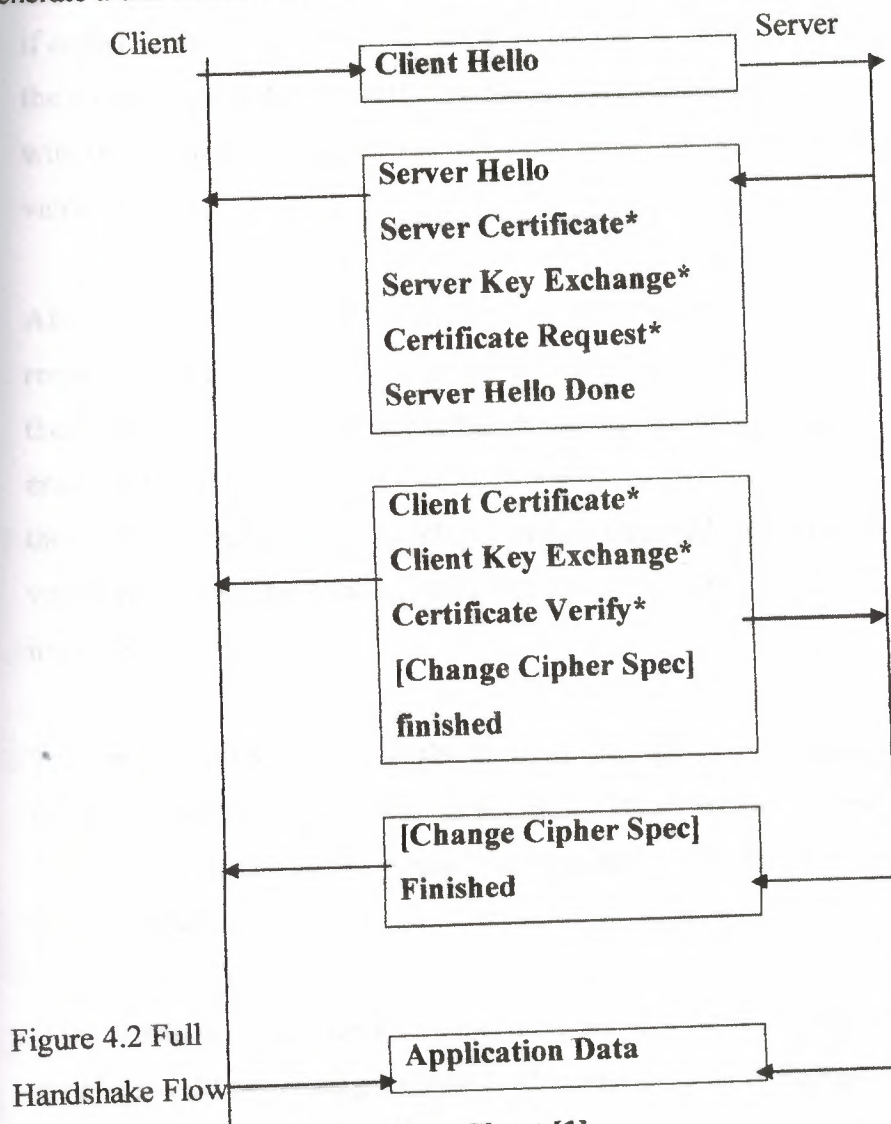


Figure 4.2 Full Handshake Flow

Figure 4.2 Full Handshake Flow Chart [1]

The handshake starts with a Hello message. The client sends a Client Hello message to the server. The server must respond to the message with a Server Hello message. In the two hello messages, communicating parties agree on the session capabilities. For example, the client announces the supported encryption algorithms and the trusted certificates known by the client. The server responds by determining the session

properties to be used during the session. If the client does not suggest some property server must decide one.

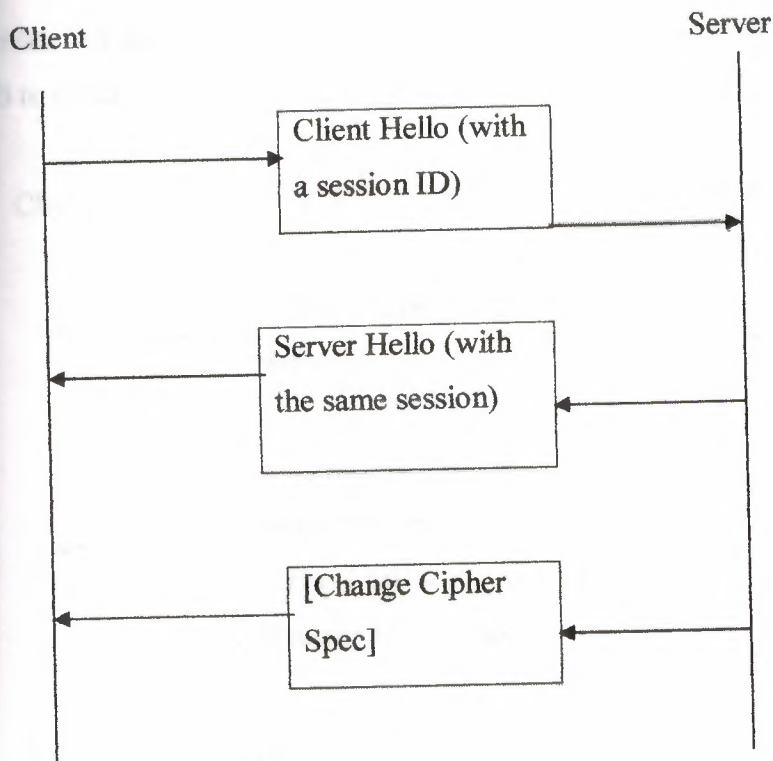
After the client has sent the Client Hello message it starts receiving messages until the Server Hello Done message is received. The server sends a Server Certificate message if authentication is required on behalf of the server. Moreover, the server may require the client to authenticate itself. The Server Key Exchange is used to provide the client with the public key which can be used to conduct or exchange the pre-master secret value.

After receiving the Server Hello Done the client continues its part of the handshake. At request, the client sends a Client Certificate message where it authenticates itself. Then the client sends a Client Key Exchange message containing either a pre-master secret encrypted with the server's public key or the information that both parties can complete the key exchange. Finally, the client sends a Finished message which contains verification of all the previous data including the calculated security related information.

The server must respond with the Finished message where it also verifies the exchanged and the calculated information. In addition, either party must send a change cipher spec message. By means of this message parties decide that they start using the negotiated session parameters.

If the client and server decide to resume a previously negotiated session the handshake may be started by sending a Client Hello message where the Session Identifier is initialized with the identifier of the previous session. If both parties share a common session identifier they may continue the secure session. The parties may start using the connection after they have confirmed the session and informed the other party with the change cipher spec message.

The WTLS also defines an abbreviated handshake where only the hello and the Finished messages are sent. In this case, both parties must have a shared secret which is used as a pre-master secret. [1]



**Figure 4.3 Resumed Connection Handshake Flow Chart [1]**

Another variation is the optimized full handshake where the server can retrieve client's certificate using the trusted third party, based on the information provided by the client in the Client Hello message. The information provided by the certificates both parties are able to complete the shared secret values using the Diffie-Hellman key exchange method. The server has to send the Server Hello, Certificate, and Finished messages to the client in order to complete the handshake on the server's behalf. The client responds with the Client Finished message.

Suppose Ali wants to authenticate John. John has a pair of keys, one public and one private. John discloses to Ali his public key. Ali then generates a random message and sends it to John.

A → B random-message

John uses his private key to encrypt the message and returns the encrypted version to Ali. B → A {random-message} bobs-private-key

Ali receives this message and decrypts it by using John's previously published public key. He compares the decrypted message with the one he originally send to John, if match, he knows he is talking to John. An imposter presumably wouldn't know John's



private key and would therefore be unable to properly encrypt the random message for Ali to check.

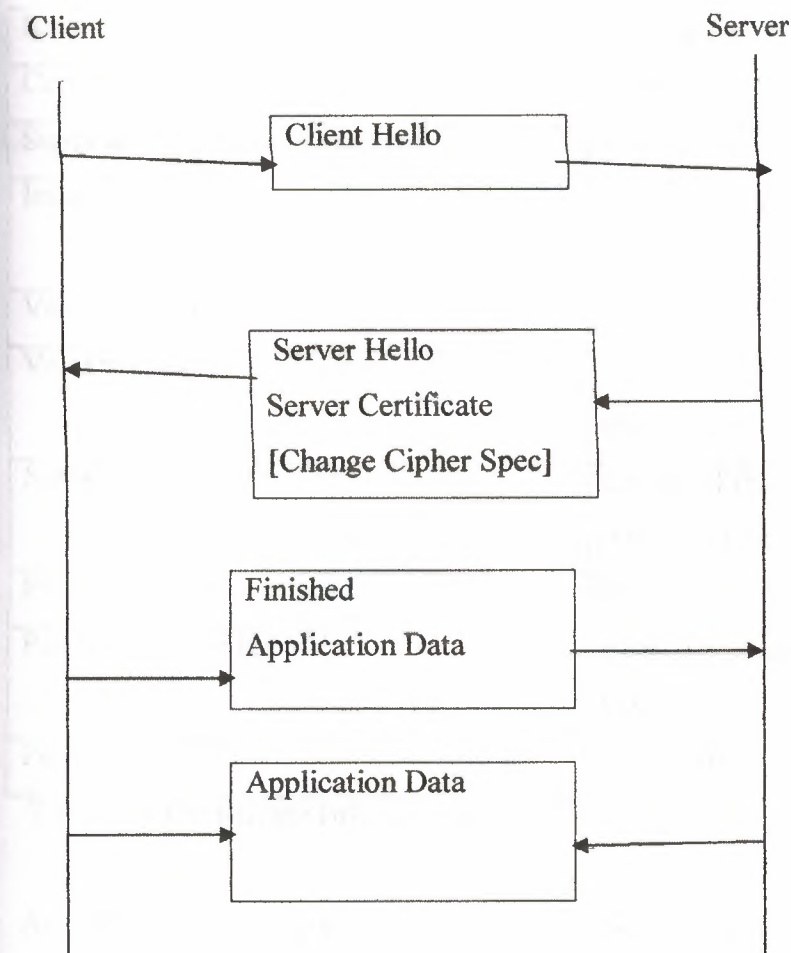


Figure 4.4 Optimized Full Handshake Flow Chart [1]

4.4.3 Authentication

Authentication in the WTLS is carried out with certificates. Authentication can occur between the client and the server or the client only authenticates the server. The latter procedure can happen only if the server allows it to occur. The server can require the client to authenticate itself to the server. However, the WTLS specification defines that authentication is an optional procedure.

Currently, X.509v3 [X509], X9.68 and WTLS certificates are supported. When the WTLS specification version 1.1 was released the X9.68 certificate was not defined yet. The WTLS certificate is optimized for size.

Authentication procedure immediately follows after the client and server hello messages. When the authentication is used, the server sends a Server Certificate message to the client. The certified information given by the server is listed in Table 4.1

Item	Description
Certificate version	Version of the certificate
Signature algorithm	Algorithm used to sign the certificate
Issuer	Defines the party who has signed the certificate, usually some CA
Valid not before	The beginning of validity period
Valid not after	The point of time after the certificate is no more valid
Subject	Owner of the key, associated with the public key being certified
Public key type	Type (algorithm) of the public key
Parameter specifier	Specifies parameter relevant for the public key
Public key	The Public key being certified

**Table 4.1 Certificate Information**

Actually, the receiving end gets a list of certificates. The list is a chain of certificates where the first one is the server's own certificate. Each of the following certificates certifies the one preceding it. According to the WTLS specification, to optimize the traffic and the client processing it is possible for the server to send only one certificate; the server certificate certified with CA's public key which is distributed independently.

The server may also send a Certificate Request message to the client in order to authenticate it. The message will immediately follow the Server Certificate message and the Server Key Exchange message (if sent). Needless to say, the Certificate Request message is optionally sent only if the Server Certificate message is sent. In the message the server lists all the accepted certificate authorities. If no authorities are listed, client may send any certificate.

At request, the client sends a Client Certificate message back to the server. The client end certificates follow the same structure as the server certificates. If the client does not have a suitable certificate the client must send an empty certificate message. Moreover, the client may send a fatal handshake failure alert message and close the secure connection. The Client Certificate message tends to contain multiple certificates. This is acceptable because the certificate list is processed by the server which is likely to possess more processing power than the client.

An explicit verification is carried out by the client, if the Client Certificate message is sent. The client concatenates all the messages received from the server or created by it and calculates a hash value to be signed. This message is sent to the server which can ensure that authentication is gone well so far.

#### **4.4.4 Key Exchange**

In order to ensure a secure communication channel encryption keys or initial values to calculate keys have to be exchanged in a secure manner. The certified exchange of public keys was described in the previous section. However, it is possible that the Server Certificate Message did not contain enough data to allow client to exchange the pre-master secret (pre-master secret is an initial value which is used to calculate the master secret). In this case a Server Key Exchange message is used to provide such data.

The key exchange mechanism of the WTLS also provides an anonymous way to exchange keys. In this procedure, the server sends a Server Key Exchange message which contains the public key of the server. The key exchange algorithm may be RSA [RSA], Diffie-Hellman [DH1], or the elliptic curve Diffie-Hellman [ECDH]. The message does not contain any certified information.

With both the RSA and the anonymous RSA the client encrypts pre-master secret with the server's public key and sends it back to the server in the Client Key Exchange message. With Diffie-Hellman based algorithms the client and the server calculate the pre-master secret based on one's private key and the counterpart's public key. This



message is omitted if some Diffie-Hellman -based algorithm was used and the client certificate was requested so that the client was able to respond it.

If the client has listed the cryptographic key exchange methods, which it supports, the server may choose whether it is going to use client's suggestions or define another method. If the client has not proposed any method the server has to indicate them.

#### **4.4.5 Privacy**

Privacy in the WTLS is implemented by means of encrypting the communication channel. The used encryption methods and all the necessary values for calculating the shared secret are exchanged during the handshake.

In the first messages, the Client Hello and the Server Hello messages, random values are exchanged. In latter phases the client and the server exchange the pre-master secret. This value is transferred over a secure connection as described in the previous section. These values are used to calculate the master secret. The master secret is a 20-byte sequence which is calculated with the following formula:  $\text{master\_secret} = \text{PRF}(\text{pre\_master\_secret}, \text{"master secret"}, \text{ClientHello.random} + \text{ServerHello.random})$

PRF stands for Pseudo-random Function which takes as input a secret, a seed, and an identifying label and produces an output of arbitrary length.

The used encryption algorithm is chosen in the Server Hello message. In this message the server informs the client that it has chosen a single cipher suite. The client provides the server with a list of cipher suites. The cipher suites comprise of a bulk encryption algorithm and a MAC algorithm. The first item in the list is the client's preference. If the server does not find an acceptable cipher suite the handshake fails and connection is closed.

Currently the most common bulk encryption algorithms are supported such as RC5 [RC5] with 40,56 and 128 bit keys, DES [DES] with 40 and 56 bit keys, 3DES [3DES], and IDEA [IDEA] with 40,56 and 128 bit keys. All the algorithms are block cipher algorithms, no streams cipher except NULLs are supported.

Encryption keys are conducted based on a key block. The key block is calculated from the initial values transferred during the handshake.

$\text{key\_block} = \text{PRF}(\text{master\_secret} + \text{expansion\_label} + \text{seq\_num} + \text{server\_random} + \text{client\_random}).$

The key block is dependent on a sequence number which makes the key block variable. The key block is recalculated in certain intervals based on the key fresh frequency. The key fresh frequency is negotiated in the Client hello and the Server hello messages. The expansion label is just a string expression for calculation. The client uses string "client expansion" and the server "server expansion". The encryption key, initial vector and MAC secret are conducted from the key block based on the key lengths required by the chosen algorithms. [1]

#### **4.4.6 Integrity**

Data integrity is ensured using the message authentication codes (MAC). The used MAC algorithm is decided at the same time as the encryption algorithm. The client sends a list of supported MAC algorithms where the preferred algorithm is the first in the list. The server returns the selected algorithm in the Server Hello message.

The WTLS supports common MAC algorithms, such as SHA [SHA] and MD5 [MD5]. There are several different versions of both algorithms e.g. SHA exists with 0, 40 and 80 bit MAC sizes. The keyed MACs are calculated using the SHA-1. The modified algorithms are based on the SHA-1 but only part of the output is used. Same kinds of versions exist of the MD5 algorithm.

A special MAC algorithm is the SHA\_XOR\_40 which is a 5-byte checksum. First the input data is divided into the 5-byte blocks. Then all blocks are XOR'ed one after another. It is required that the XOR MAC must be encrypted and is only used for CBC mode block ciphers. The algorithm is intended for devices with limited CPU resources. The MAC is generated over the compressed WTLS data. The following values are used to calculate the MAC:

$\text{HMAC\_Hash} = (\text{MAC\_Secret}, \text{seg\_num WTLS\_Compressed\_data.record\_type} + \text{WTLS\_Compressed\_data.data\_length} + \text{WTLS\_Compressed\_data.fragment});$   
HMAC\_Hash illustrates the used keyed MAC algorithm e.g. SHA-1 or MD5. The MAC\_Secret value is one of the key block values. After the HMAC\_Hash value is

generated the determined length of the MAC value is set to the WTLS cipher text-structure. [1]

#### 4.4.7 Secure State

We discussed how the secure session is negotiated. After the negotiations, both communicative parties have a uniform secure state which contains the security parameters described in Table 4.2.

Item	Description
Connection End	Indicates whether the entity is considered a client or a server
Bulk Encryption Algorithm	An algorithm used for bulk encryption.
MAC Algorithm	The algorithm to be used for guaranteeing the message integrity/authentication
Compression Algorithm	The algorithm used to compress data before encryption. All the information required to do compression.
Master Secret	A 20 byte secret between the two peers in the secure connection
Client Random	A 16 byte value provided by the client
Key Refresh	The time interval how after some connection state parameters are updated (encryption key, MAC secret and IV).
Sequence Number Mode	Which scheme is used to produce sequence numbers in the secure connection. Current options are implicit, explicit sequence numbering or Off.

**Table 4.2 the security parameters of the secure connection [1]**

The current state is made by means of the security parameters. The current state is continuously updated. Each connection state includes the elements such as the current encryption keys, MAC keys, IVs and sequence numbers. Both the server and the client have separate secret keys for encryption, MACs, etc. [1]



#### 4.4.8 Evaluation of the WTLS

Knowing the security concepts and the ways how the WTLS implements them, the evaluation can be done. Multiple security holes have already been found but the decision whether they are serious enough to affect the whole protocol design is under investigation.

*The SSL is developed by Netscape. It has been universally accepted in Internet for authenticated and encrypted communication between clients and servers. The new Internet Engineering Task Force (IETF) standard called Transport Layer Security (TLS) is based on the SSL. This was recently published as an IETF Internet-Draft, The TLS Protocol Version 1.0. [5]*

The SSL protocol provides privacy, authentication, and integrity. Data is encrypted with symmetric cryptography and authentication is done using asymmetric or public key cryptography. The integrity of messages is checked using a keyed MAC. Secure hash functions like SHA or MD5 are used for MAC computations. The goals of the SSL are cryptographic security, interoperability, extensibility, and relative efficiency. [5]

The SSL is a layered protocol. At each layer, messages may include fields for length, description, and content. The SSL takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result. Received data is decrypted, verified, decompressed, and reassembled and then delivered to higher level clients. [5]

The TLS version 1.0 and SSL version 3.0 are very similar. They differ from each other almost nothing. [5]

The WTLS is good at the surface. Most of the text in the WTLS specification has been adopted, word to word, from the TLS Specification. However, many of the changes that were made by WAP Forum have led to security problems. [8]

#### • Predictable Ivs Lead To Chosen-Plain Text Attacks Against Low Entropy Secrets

While the TLS protocol was designed to be used over a reliable transport (such as TCP/IP), the WTLS protocol should be able to operate over an unreliable datagram transport where datagrams may be lost, duplicated, or reordered. If CBC mode is being used, this requirement makes it necessary for the IV to be either contained in the packet itself or that the IV for that block is somehow derived from the data already available to the recipient. WTLS uses a linear IV computation, even for reliable transports.

When a block cipher is used in CBC mode, the IV for encrypting each packet is computed as follows:

$$IV_s = IV_0 + (s/s/s/s) \quad (1)$$

Where  $s$  is a 16-bit sequence number of the packet and  $IV_0$  is the original IV, derived during key generation. The plain text blocks  $Ps,0,ps,1,\dots$  in the packet  $s$  are encrypted as follows:

$$Cs,0 = Ek(IV_s + Ps,0) \quad (2)$$

$$Cs,i = Ek(cs,i-1 + ps,i), \text{ for } i > 0 \quad (3)$$

Consider a terminal application (such as telnet), where each keypress is sent as an individual packet. Ali enters his password into this application, and Eve captures these packets. Eve now has blocks of type

$$Cs,0 = Ek(ps,0 + IV_0 + (s/s/s/s)) \quad (4)$$

Where  $Ps,0$  contains an unknown letter Ali's password. Note that  $s$  is known to Eve. Now somehow Eve gets hold of Ali's channel, perhaps through an echo feature in some application. Eve guesses that the unknown letter in the password is  $L$ . Eve sends the following packet through Ali's channel:

$$Pr,0 = L + (s/s/s/s) + (r/r/r/r) \quad (5)$$

Where  $r$  is the sequence number of this packet. One can see that because  $(r/r/r/r)$  cancels out in the CBC computation, a right guess  $L = Ps,0$  leads to matching cipher texts  $Cr,0 = Cs,0$ . In other words, this is an oracle that tells whether the guessed password letter was correct. The entire password can be brute forced, letter by letter, with a few hundred tests using this oracle.

While the above description of the attack is highly simplified, one can see that a too easily predictable IV leads to situations where low-entropy secrets can be read.

This attack is similar to the attacks described by Bellare against the IPSec protocol [8].

### • The XOR MAC and Stream Ciphers

The WTLS protocols supports, among other MACs, a 40-bit XOR MAC. The XOR MAC works by padding the message with zeros, dividing it into 5-byte blocks and xoring these blocks together. Note that this construction differs from the one presented.

[8]

The specifications states the XOR MAC is only intended for some devices with very limited CPU resources. The specification also tells us that the XOR MAC may not



provide as strong message integrity protection as SHA when export able encryption is being used. In fact it is easy to see that the XOR MAC does not provide any message integrity protection if stream ciphers are being used, regardless of the key length.

If one inverts a bit position  $n$  in the cipher text, the MAC can be made to match by inverting the bit ( $n \bmod 40$ ) in the MAC. This can be repeated arbitrary number of times. Thus, when stream ciphers are used, the XOR MAC does not provide any integrity protection. [8]

- **35-Bit DES Encryption.**

The 40-bit DES encryption method is defined to use five bytes of keying material.

Because of the parity bits contained in each byte of a DES key, there are only  $5 * 7 = 35$  effective key bits in five bytes. This amounts to a reduction of the keyspace by a factor of 32. Note that the 56-bit DES has the correct amount of keying material (8 bytes). The protocol clearly does not meet its requirement of reaching the best possible security level in export-weaken encryption modes. [8]

- **The PKCS #1 Attack**

The RSA signatures and encryption are performed according to PKCS #1, version 1.5 [8]. Daniel Bleichenbacher and others have demonstrated that if the protocol includes an oracle that tells whether a given packet has a correct PKCS #1 v 1.5 padding, RSA messages can be decrypted with approximately  $2^n$  where  $n = 20$ , chosen cipher text queries [2,3]. In some implementations the WTLS error messages `bad_certificate` and `decode_error` may provide such an oracle to the attacker.

It is better if the 2.0 version of the PKCS # 1 to be used instead. [8]

- **Unauthenticated Alert Message**

Some of the alert messages used in the protocol are sent in cleartext and are not properly authenticated. Most of these messages are warnings and do not cause the session to be terminated.

Since an alert message can take up a sequence number "slot" in the protocol, an active attacker may replace an encrypted datagram with an unauthenticated plaintext alert message with the same sequence number without being detected. This leads to a truncation attack that allows arbitrary packets to be removed from the data stream.



It is recommended that all messages affecting the protocol state should be properly unauthenticated. [8]

- **Other Plaintext Leaks.**

Under exportable keys the initial IV of each packet can be determined by an eavesdropper from the Hello messages and the sequence number alone. We are not aware of export laws in any country that would mandate this.

The change of keys can be determined by an eavesdropper, because the *record\_type* field is sent unencrypted. This field determines the type of the message, one type being the Change Cipher Spec type.

Also, the existence of encrypted error messages can be determined from the *record\_type* field. The exact nature of the encrypted error messages can not be determined. [8]

- **Portable Plaintext Attacks**

In order to mount an exhaustive key search on a symmetric cipher, one needs to have enough known or probable plaintext, so that the correct key can be recognized with trial decryption of one or more blocks. Attacks of this type against the IPSec protocol have been considered. [8]

Observed that brute force attacks against the block ciphers in WTLS can be easily mounted, because the correct keys can be always recognized with a trial decryption of the last block of each packet. Let us assume that a 64-bit block cipher is used. The last block is padded to the next full 8-byte limit by filling it with the padding length. In other words, if the last byte of  $E_k^{-1}(C_i) + C_{i-1}$  is  $n$ , the preceding  $n$  bytes of the plaintext must also contain this number. If this is passed, then the key can be furthermore verified with the last block of arbitrary number of packets.

- **A note on Diffie-Hellman key agreement.**

The WTLS specification includes 512-and 768-bit primes'  $p_1, p_2$ , along with the generators that are to be used in Diffie-Hellman computations. The group order of the multiplicative subgroup generated by the generator is not given.

The absence of the group order makes it impossible to check that the given public value belongs to the correct multiplicative subgroup, as in DSA and KEA. [8]

It is known that if the group order is relatively smooth, the discrete algorithm problems become substantially easier to an attacker that knows the factorization of the group order. [8]

To verify that the group was not divisible with small factors of  $p-1$  (for either of these groups), we ran Pail Zimmerman's GMP-ECM elliptic curve factoring program for a total of 100 hours of CPU time on 333 MHz Sun Ultra 10s.

It has been found 4 factors (largest 52 bits) of the 512-bit number  $P1 - 1$ , leaving a 438-bit composite cofactor that verifies that the group order of group1 is not divisible with the factors found.

It also has been found that 7 factors (largest 70 bits) of the 768-bit number  $P2 - 1$ , leaving a 658-bit composite cofactor, which verifies that, the group order of group2 is not divisible with the factors found.

Then it has been suspected that the group order information was left out from the specification, not because of an attempt to mount a back door into WTLS, but because the authors did not see the relevance of the group order to the security of Diffie-Hellman key exchange. The group orders of the elliptic curve groups are given. All of these are prime.

#### **4.4.9 Reasons for Defects**

The protocol has been developed to support very wide range of mobile devices. The weakest devices cannot support heavy encryption because of the limitations of CPU, memory and bandwidth resources. There is no real security when allowing the client to choose null or weak encryption methods. The security cannot be provided for the devices that cannot execute heavy algorithms.

Allowing anonymous connection to be established can be very risky. Anonymous authentication prongs the connection to man-in-the-middle attacks. To prevent this problem to the client should define that during the handshake it will not support key exchange suites without authentication. It should always authenticate the server it is going to use. The client will remain anonymous if it does not send its own certificate to the server. The server can ask the client to send its certificate, but the client can send

just an empty response. Then it is up to server if it accepts the client without authentication. The Server Key Exchange Method message will only be sent when using anonymous method like ECDH\_anon, RSA\_anon and DH\_anon.

The support for the large amount of algorithms makes system vulnerable because some of the algorithms have been proven weak. Thus allowing the connections to be encrypted using weak algorithms raises security issues. To solve this problem the clients and servers should only accept to use only strong algorithms.

The restrictions legislated by governments do not allow the algorithms with too long keys to be exported. For instance, the governments of United States only allow 40-bit keys to be exported. This sets the limitations to the security requirements. As long as these kinds of laws are valid, there no way to provide decent security. However, it is a good feature that the WTLS takes into consideration the local legislation and restrictions so that only one version of the protocol is needed. The Wassenaar agreement applies that the 56-bit keys are allowed to be used in the financing applications. [1]

#### **4.4.10 Known Security Holes**

The number of potential security problems has been identified in the WTLS. The WTLS specification has been adopted from the TLS specification with some modifications and changes. These modifications and changes have at least partly led to some security problems including the chosen plaintext data recovery attack, the datagram truncation attack, the message forgery attack and the key-search shortcut for some exportable keys. [8]

Initial vectors called IVs are used by the CBC mode block ciphers to create entropy. Entropy is needed to protect symmetric key that is used in the CBC mode block cipher. Without IV, an original plain text would be encrypted with master key. This would open a possibility to use brute force method to find the shared secret. The usage of IV prevents this to happen because the first block in the packet is first XOR'ed with IV. Knowing the content of the original packet does not help in any way, because it is XOR'ed. [8]



Because the WTLS supports an unreliable datagram support where datagrams may be lost, duplicated, or reordered, the CBC mode needs a new IV for encrypting each packet. The used IV is computed XOR'ing the sequence number of the packet and the original IV, which is derived during the key generation. This is also called a linear IV computation. The first plaintext block in the packet is then XOR'ed with the computed IV. The original IV is computed based on values sent during the handshake. All these values containing client\_random, server\_random and sequence number are sent without encryption, so they can be eavesdropped. These predictable IVs lead to chosen-plaintext attacks against low-entropy secrets. This security problem affects privacy. [8]

The WTLS supports a 40-bit XOR MAC, which works by padding the message with zeros, dividing it into 5-byte blocks and XOR'ing these blocks together. The XOR MAC does not provide any message integrity protection if stream ciphers are being used, regardless of the key length. A bit can be inverted in the ciphertext if the inverting is also done to the MAC. Thus the integrity check will be successful even when the content has been modified. This security problem affects integrity. [8]

The DES key contains a parity bit in each byte. When using a 40-bit key, the effective key length of the DES encryption is actually only 35 bits ( $5 \times 7 = 35$ ). Nevertheless, a 56-bit DES-key has the correct amount of keying material, 56 bits ( $8 \times 7 = 56$ ). The best possible security level in export-weakened encryption modes has not been reached. Although the expanded key material in the 40-bit DES is eight bytes, actual key material is only five bytes. This security problem affects privacy. [8]

The PKCS #1 version 1.5 contains security problem when used with the protocol including an oracle that tells whether a given packets has a correct PKCS #1 version 1.5 padding. If the system in some way tells to the intruder whether the used key is correct, it said that the system has an oracle. Using this oracle the intruder can try to find the correct key trying all possibilities and checking the response of the system. [8]

The RSA signatures and encryption are performed according to the PKCS #1 version 1.5 in the WTLS, thus enabling the RSA messages decrypting with approximately  $2^{20}$  chosen ciphertext queries. In the WTLS bad\_certificate and decode\_error may

provide an oracle to be used for illegal decrypting. This security problem affects authentication. [8]

The unauthenticated alert messages, used in the WTLS, let the active attacker to replace an encrypted datagram with an unauthenticated plaintext alert message with the same sequence number without being detected. This security problem affects integrity. [8]

The `record_type` field is sent unencrypted. The eavesdropper can determine the change of keys reading the contents of this field. The existence of encrypted error messages can be determined from this field. This security problem affects privacy. [8]

The brute force attacks against the block ciphers can be mounted, because the correct keys can always be recognized with a trial decryption of the last block in each packet. The last block is padded to the next full 8-byte limit by filling it with the padding length. [8]

The WTLS specification includes pre-defined values for variables used in Diffie-Hellman computations, but the group order of the multiplicative subgroup is left out. The absence of the group order makes it impossible to check that the given public value belongs to the correct multiplicative subgroup. This can be considered only as a minor problem, but it may affect to authentication. [8]

The 40-bit encryption is easily broken, so implementations requiring strong security should not allow 40-bit keys. Similarly, anonymous Diffie-Hellman is strongly discouraged because it cannot prevent man-in-the-middle attacks. For example, certificate chains containing 512-bit RSA keys or signatures are not appropriate for high-security applications. [6]

Whenever the server is authenticated, the channel should be secure against man-in-the-middle attacks, but completely anonymous sessions are inherently vulnerable to such attacks. Anonymous servers cannot authenticate clients, since the client signature in the certificate verify message may require a server certificate to bind the signature to a particular server. Completely anonymous connections only provide protection against



passive eavesdropping. Unless an independent tamper-proof channel is used to verify that the finished messages were not replaced by an attacker, server authentication is required in environments where active man-in-the-middle attacks are a concern. [6]

Care must be taken in designing and seeding PRNGs. PRNGs based on secure hash operations, most notably MD5 and/or SHA, are acceptable, but cannot provide more security than the size of the random number generator state. [6]

512-bit RSA keys are not secure enough for high-value transactions or for applications requiring long-term security. When the public key in the certificate cannot be used for encryption, the server signs a temporary RSA key, which is then exchanged. In exportable applications, the temporary RSA key should be the maximum allowable length (i.e., 512 bits). Keys should be changed often. For typical electronic commerce applications, it is suggested that keys be changed daily or every 500 transactions, and more often if possible. Note that while it is acceptable to use the same temporary key for multiple transactions, it must be signed each time it is used. [6]

For the man-in-the-middle attack, an attacker must actively change one or more handshake messages. If this occurs, the client and the server will compute different values for the handshake message hashes. As a result, the parties will not accept each others' finished messages. Without the master\_secret, the attacker cannot repair the finished messages, so the attack will be discovered. [6]

#### **4.4.11 The Accepted Level of Security**

A sufficient security level is always a compromise between the usability and the strength of the used encryption method. It is impossible to define certain standards for a sufficient level of security because the needed level is dependent on the transmitted data. The transmitted information always has some value; hence, the owner of the information decides how much effort is put to preserve the confidentiality.

A bank's strategic plan is such information which requires a usage of a remarkably strong encryption methods. It may even be the case that encryption requirements are not met because of the export regulations. On the other hand, some information may not



require any encryption. Encryption is more than just selecting the algorithm. Different algorithms have a variable amount of requirements in order to work properly. Usually the stronger the selected algorithm is the more it requires computing resources. From the WTLS's point of view the provided security level is always a trade-off with the usage of limited resources. There is no point in using over 50 % of the limited computing resources for encryption and decryption or create excess traffic to the narrow bandwidth. However, the WTLS has to ensure certain security level in order to be used for commercial purposes. [26]

The worst flaw in the WTLS is that it allows users to choose extremely weak algorithms. An example of a weak algorithm is the SHA\_XOR\_40 which should provide integrity for data. The other defect is that it is the server which makes the ultimate decision on the used algorithms though the decision is made based on the client's suggestions. If the client does not suggest certain algorithms which the server is willing to use the connection cannot be established. [26]

For reference Table 4.3. gives a picture of the average time estimates for a hardware brute-force attacks. The figures are based on the estimates made in 1995.

Cost [\$]	40	56	64	80	112	128
100 K	2 s	35 h	1 a	70 000 a	10xE14	10xE19
1 M	0.2 s	3.5 h	37 d	700e0 a	10xE13	10xE18
10 M	0.02 s	21 min	4 d	700 a	10xE12	10xE17
100 M	2 ms	2 min	9 h	70 a	10xE11	10xE16
1 G	0.02 ms	13 s	1 h	7 a	10xE10	10xE15

**Table 4.3 Average time estimates for a hardware brute-force attack in 1995. [9]**

According to the Moore's Law, nowadays the time estimates can be divided by 10. In the WTLS, the most used encryption algorithms for ciphering the communication channel will be the RC5\_CBC with 40- and 56-bit keys and DES\_CBC with a 40-bit key. Needless to say, 40 or 56 bits are not enough. There are no technical restrictions for using longer keys, even the current CPU resources and the available bandwidth would be adequate for a stronger encryption. The export regulations dictate the current level of security. The WTLS also provides 3DES\_CBC\_EDE with a 168-bit key but it is not allowed to be used outside the USA. [26]

In addition to the short key length, some of the keying material is used as parity bits in DES-based algorithms. This causes the effective key length drop even more. Actually, the effective key lengths are 35 bits, DES\_40, and 56 bits, DES\_64. Generally, the DES is an old standard and there are several implementations for breaking DES.

The public keys are used in exchanging the pre-master; hence, it is vital that the exchange is performed in a secure manner. Table 4.4. Illustrates some recommendations for the public-key lengths.

Year	vs. Individual	vs. Corporation	vs. Government
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

**Table 4.4. Recommended public key lengths (in bits) [9]**

In the WTLS, for anonymous key exchange only 512- and 768-bit versions are available. For certified authentication no restrictions are set on behalf of the WTLS. According to the Wassenaar Arrangement, it does not have any restrictions for authentication or digital signatures. This applies on the condition that only directly authentication related information is transferred using these protection mechanisms.[26]

Assuming that certified authentication uses key lengths above 1024 bits, the WTLS provides a sufficient level of confidentiality for the key exchange and the authentication. But when it comes to anonymous key exchange, a 512-bit key is far too short in order to provide a secure key exchange. The 768-bit option is on the limits of acceptability.

The keyed MAC functions, SHA-1 and MD5, provided by the WTLS can be considered secure if the full key length is used. According to Schneider, there are no known cryptographic attacks against SHA or MD5. However, it is said that MD5 has a weakness in the compression function, but it has no practical impact on the security of the hash function. [9]

#### 4.5 Summary

The WTLS is the first attempt to provide a secure end-to-end connection for the Wireless Application Protocol. The most common protocols, such as TLS version 1.0 and SSL version 3.0, were adopted as a basis of the WTLS. However, it was not possible to apply the procedures, used in the traditional connection-oriented world, as such. The development work resulted in a protocol which resembles the TLS but it has some properties in order to adjust to the wireless world. [1]

The WTLS supports a coverable span of algorithms to meet the requirements of privacy, authentication, and integrity. Currently, privacy is implemented using the block ciphers, such as DES\_CBC, IDEA, and RC5\_CBC. RSA- and Diffie-Hellman-based key exchange suites are supported to authenticate the communicating parties. Finally, integrity is implemented with SHA-1 and MD5 MAC algorithms. [1]

In this chapter, a number of security flaws and shortcomings in the WAP WTLS protocol had been identified. Among of them are: a chosen plaintext data recovery attack, a datagram truncation attack, a message forgery attack, and a key-search shortcut for some exportable keys. WTLS is clearly weak and needs to be revised. It is obvious that the WTLS protocol appear to be more vulnerable than to attacks than TLS. Most of these attacks had been mentioned in the previous parts.



## **5. CRYPTOGRAPHY AND ITS ALGORITHMS**

### **5.1 Overview**

The security in the WAP depends on encrypting/decrypting the server/client applications. In order to encrypt a WAP server, a strong encryption technique must be applied carefully. In the following parts a server/client encrypting/decrypting program is going to be analyzed. The program specializes in developing, deploying and maintaining security systems for internet and extranet systems. The security services include client authentication systems, data encryption systems, logging and trace systems, VPN's and custom security systems.

### **5.2 Cryptography**

Cryptographic systems are generally classified along three independent dimensions:

1. The type of operations used for transforming plaintext to ciphertext, where all encryption algorithms are based on two major principles: substitution, in which each element in the plaintext like bit, letter, group of bits or letters is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost.
2. The number of used keys, if both the sender and the receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and the receiver use a different key, the system is referred to as asymmetric, two-key, or public-key encryption.
3. The way, in which the plaintext is processed, a block cipher processes the input one block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along. [2,3]

#### **5.2.1 Cryptanalysis**

The process of attempting to discover X and K or both is known as cryptanalysis. The strategy used by the cryptanalyst depends on the nature of the encryption scheme and the information available to the cryptanalyst. In some cases, not even the encryption algorithm is known, but in general we can assume that the opponent does not know the algorithm used for encryption. One possible attack under these circumstances is the brute-force approach of trying all the possible keys. If the key space is very large, this

Type of Attack	Known to Cryptanalyst
Ciphertext only	<ul style="list-style-type: none"> <li>• Encryption algorithm</li> <li>• Ciphertext to be decoded</li> </ul>
Known plaintext	<ul style="list-style-type: none"> <li>• Encryption algorithm</li> <li>• Ciphertext to be decoded</li> <li>• One or more plaintext-ciphertext pairs formed with the secret key</li> </ul>
Chosen plaintext	<ul style="list-style-type: none"> <li>• Encryption algorithm</li> <li>• Ciphertext to be decoded</li> <li>• Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key</li> </ul>
Chosen ciphertext	<ul style="list-style-type: none"> <li>• Encryption algorithm</li> <li>• Ciphertext to be decoded</li> <li>• Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key</li> </ul>
Chosen text	<ul style="list-style-type: none"> <li>• Encryption algorithm</li> <li>• Ciphertext to be decoded</li> <li>• Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key</li> <li>• Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key</li> </ul>

**Table 5.1 Types of Attacks on Encrypted Messages [2]**

becomes impractical. The ciphertext-only attack is the easiest to define against because the opponent has the least amount of information to work with. In table 5.1 there are types of attacks on Encrypted Messages.

### 5.2.2 Classical Encryption Techniques

The classical encryption techniques are the basis of today's used techniques, are classified to two techniques: the substitution and transposition techniques.

1. Substitution Techniques: a substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns. Caesar Cipher, Monoalphabetic ciphers, Playfair cipher, Hill Cipher, and Polyalphabetic ciphers are substitution techniques.

*Caesar Cipher*: is the earliest known use of a substitution cipher and the simplest that was created by Julius Caesar. It involves replacing each letter of the alphabet with the letter standing three places further down the alphabet. If we assign a numerical a numerical equivalent to each letter (a=1, b=2, etc), then the algorithm will be expressed as follows. For each plaintext letter P, substitute the ciphertext letter C:

$$C = E(P) = (P+3) \bmod (26)$$

A shift may be of any amount, so that the general Caesar algorithm is

$$C = E(P) = (P+K) \bmod (26)$$

Where K takes on a value in the range 1 to 25. The decryption algorithm is simply:

$$P = D(C) = (C - K) \bmod (26)$$

There are three important characteristics of this technique:

1. The encryption and decryption algorithms are known
2. There are only 25 keys to try
3. The language of the plaintext is known and easily recognizable.

*Monoalphabetic Ciphers*: With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution. If, instead, the "cipher" line can be any permutation of the 26 alphabetic characters, then there are 26! Or greater than 4 times 10 to the power 26 possible keys.



This is 10 orders of magnitude greater than the key space for DES and would seem to eliminate brute-force techniques for cryptanalysis. Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet. A countermeasure is to provide substitutes, known as homophones for single letter. If the number of symbols assigned to each proportional to the relative frequency of that letter, then single-letter frequency information is completely obliterated.

*Playfair Cipher:* The best known multiple-letter Encryption cipher is playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext digrams. The playfair algorithm is based on the use of a 5 X 5 matrix of letters constructed using a keyword. The playfair cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are  $26 \times 26 = 676$  digrams. [2]

*Hill Cipher:* Developed by the mathematician Lester Hill in 1929. The encryption algorithm takes  $m$  successive plaintext letters and substitutes for them  $m$  ciphertext letters. The substitution is determined by  $m$  linear equations in which each character is assigned a numerical value ( $a = 0, b = 1 \dots z = 25$ ). For  $m = 3$ , the system can be described as follows:

$$C_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$$

$$C_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$$

$$C_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$$

This can be expressed in form:

$$C = KP$$

Where  $C$  and  $P$  are column vectors of length 3, representing the plaintext and ciphertext, and  $K$  is a 3 X 3 matrix, representing the encryption key. Operations are performed mod 26.

Decryption requires using the inverse of the matrix  $K$ . The inverse  $K^{-1}$  of a matrix  $K$  can be defined by the equation:  $KK^{-1} = K^{-1}K = I$ , where  $I$  is the matrix that is all zero except for ones along the main diagonal from the upper left to lower right. The inverse of a matrix does not always exist, but when it does, it satisfies the preceding equation.

In general, the Hill system can be expressed as follows:

$$C = Ek(P) = KP$$

$$P = D_k(C) = K^{-1}C = K^{-1}KP = P$$

Although the Hill cipher is strong against a ciphertext-only attack, it is easily broken with a known plaintext attack.

*Polyalphabetic Ciphers:* Another way to improve on the simple monoalphabetic is to use different monoalphabetic substitutions as one proceeds through the plaintext message. The general name for this approach is polyalphabetic cipher. All these techniques have the following in common:

1. A set of related monoalphabetic substitution rules is used.
2. a key determines which particular rule is chosen for a given transformation.

To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating word.

Decryption is equally simple. The key letter again identifies the row. The position of the ciphertext letter in that row determines the column, and the plaintext letter is at the top of that column.

2. *Transposition Techniques:* a very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher. The simplest such cipher is the rail fence technique in which the plaintext is written as a sequence of diagonals and then read off as a sequence of rows. This sort of thing would be trivial to cryptanalyze. A more complex scheme is to write the message in rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm. [2,3]

### 5.2.3 Public-Key Cryptography

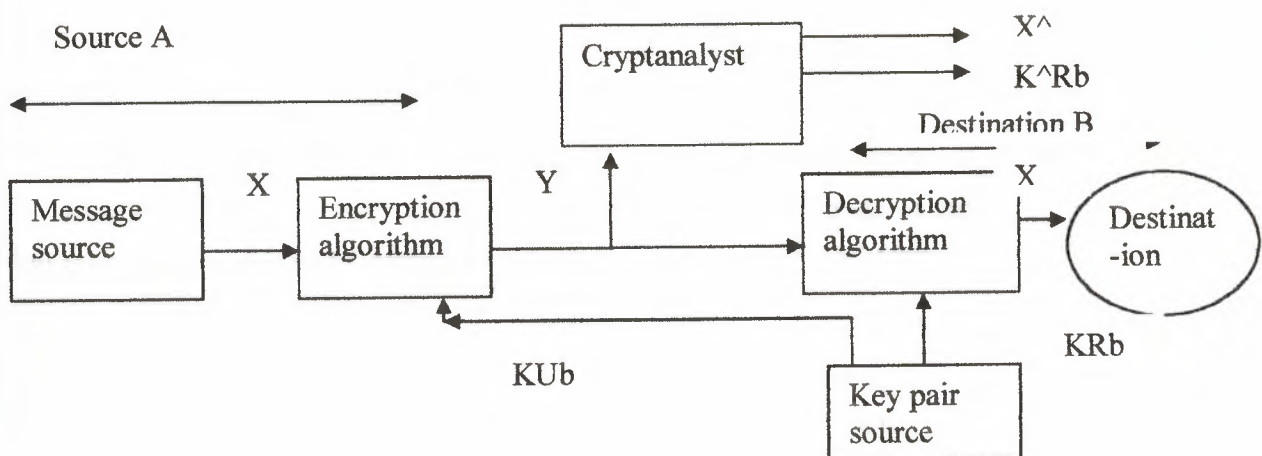
Public-key cryptography provides a radical departure from all that has gone before. For one thing, public-key algorithms are based on mathematical functions rather than on substitution and permutation. More important public-key cryptography is asymmetric involving the use of two separate keys, in contrast to symmetric conventional encryption, which uses only one key. The use of two keys has profound consequences in all areas of confidentiality, key distribution, and authentication.

Table 5.2 summarizes some important aspects of conventional and public-key encryption.

Conventional Encryption	Public-Key Encryption
<i>Needed to work</i>	<i>Needed to work</i>
1. The same algorithm with the same key is used for encryption and decryption.	1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption
2. The sender and receiver must share the algorithm and the key	2. The sender and The receiver must each have one of the matched pair of keys(not the same one)
<i>Needed for Security</i>	<i>Needed for Security</i>
1. The key must be kept secret	1. One of two keys must be kept secret
2. It must be impossible or at least impractical to decipher a message if no other information is available.	2. It must be impossible or at least impractical to decipher a message if no other information is available.
3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

**Table 5.2 Conventional and Public-Key Encryption**

In figure 5.1 we can distinguish the secrecy of the Public-Key Cryptosystem.



**Figure 5.1 Public-Key Cryptosystem: secrecy.**



There is some source A for a message, which produces a message in plaintext,

$$X = [X_1, X_2, \dots, X_M]$$

The M elements of X are letters in some finite alphabet. The message is intended for destination B. B generates related pair of keys: a public key,  $K_{Ub}$ , and a private key,  $K_{Rb}$ .  $K_{Rb}$  is known only to B, whereas  $K_{Ub}$  is publicly available and therefore accessible by A.

With the message X and the encryption key  $K_{Ub}$  as input, A forms the ciphertext

$$Y = [Y_1, Y_2, \dots, Y_N]$$

$$Y = E_{K_{Ub}}(X)$$

The intended receiver, in possession of the matching private key, is able to invert the transformation:

$$X = D_{K_{Rb}}(Y)$$

$$Y = E_{K_{Ra}}(X)$$

$$X = D_{K_{Ua}}(Y)$$

It is however, possible to provide both the authentication function and confidentiality by a double use of the public-key scheme:

$$Z = E_{K_{Ub}}[E_{K_{Ra}}(X)]$$

$$X = D_{K_{Ua}}[D_{K_{Rb}}(Z)]$$

#### 5.2.4 The RSA Algorithm

The (Rivest-Shamir-Adleman) RSA scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and  $n-1$  for some  $n$ . The scheme developed by Rivest, Shamir, and Adleman makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number  $n$ . That is, the block size must be less than or equal to  $\log_2(n)$ ; in practice, the block size is  $2^k$  bits, where  $2^k < n \leq 2^{k+1}$ . Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C:

$$C = M^e \bmod n$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

Both, the sender and the receiver must know the value of  $n$ . the sender knows the value of  $e$ , and only the receiver knows the value of  $d$ . Thus, this is a public-key of  $KU = \{e, n\}$  and a private key of  $KR = \{d, n\}$ .

There are three possible approaches to attacking the RSA algorithm are as follows:

- Brute force: This involves trying all possible private keys.
- Mathematical attacks: There are several approaches, all equivalent in effect to factoring the product of two primes.
- Timing attacks: These depend on the running time of the decryption algorithm.

The defense against the brute-force approach is the same for RSA as for the other cryptosystems- namely, use a large key space. Thus, the larger the number of bits in  $e$  and  $d$ , the better. However, because calculations involved, both in key generation and encryption/decryption, are complex, the larger the size of the key, the slower the system will run.

There are three approaches to attacking RSA mathematically:

- Factor  $n$  into its two prime factors. This enables calculation of  $\Phi(n) = (p-1) \times (q-1)$ , which, in turn, enables determination of  $d = e^{-1} \pmod{\Phi(n)}$ .
- Determine  $\Phi(n)$  directly, without first determination  $p$  and  $q$ . Again, this enables determination of  $d = e^{-1} \pmod{\Phi(n)}$ .
- Determine  $d$  directly, without first determination  $\Phi(n)$ .

The RSA encryption algorithm uses a one-way function, which is relatively easy to calculate in one direction, but extremely difficult to reverse the calculation. For example it is relatively simple for someone to calculate the square of a value using a pencil and paper, but it is difficult to find the square root of a value. Most of us could calculate the square of 63 as 3969, but what is the square root of 6889? The answer is 93, which is not a easy to determine (without the aid of a calculator).

Public-key encryption is the best way to secure data. With this method a user generates two electronic keys, typically with hundreds or thousands of bits. These keys are special number and relate to extremely large prime numbers (as it is difficult to factorize large prime numbers. For example, I have two prime numbers (small ones), and when I multiple them together I get the value of:

1,354,657

What was the original prime numbers? With public key encryption these numbers typically have thousands of bits, which gives values from 1 to 1,797,693,134, 862,

315,907,729,305,190,789,..... (in total, it has 309 digits). Imagine finding the factors for two numbers that are this long?

With RSA, initially the person picks two prime numbers. For example:

$$p=11 \text{ and } q=3$$

Next, the  $n$  value is calculated. Thus:

$$n = p \times q = 11 \times 3 = 33$$

Next PHI is calculated by:

$$PHI = (p-1)(q-1) = 20$$

The factors of  $PHI$  are 1, 2, 4, 5, 10 and 20. Next the public exponent  $e$  is generated so that the greatest common divisor of  $e$  and  $PHI$  is 1 ( $e$  is relatively prime with  $PHI$ ).

Thus, the smallest value for  $e$  is:

$$e = 3$$

The factors of  $e$  are 1 and 3, thus 1 is the highest common factor of them. Thus  $n$  (33) and the  $e$  (3) values are the public keys. The private key ( $d$ ) is the inverse of  $e$  modulo  $PHI$ .

$$d = e^{-1} \bmod [(p-1) \times (q-1)]$$

This can be calculated by using extended Euclidian algorithm, to give the private key,  $d$  of 7.

Thus  $n=33$ ,  $e=3$  and  $d=7$ .

The PARTY2 can be given the public keys of  $e$  and  $n$ , so that PARTY2 can encrypt the message with them. PARTY1, using  $d$  and  $n$  can then decrypt the encrypted message.

For example, if the message value to decrypt is 4, then:

$$c = m^e \bmod n = 4^3 \bmod 33 = 31$$

Therefore, the encrypted message ( $c$ ) is 31.

The encrypted message ( $c$ ) is then decrypted by PARTY1 with:

$$m = c^d \bmod n = 31^7 \bmod 33 = 4$$

Which is equal to the message value.

On RSA a simple RSA calculator had been programmed on the visual basic program to help the aim of this research on the security of the WAP. The program had been tested successfully and that can be seen in figure 5.2, figure 5.3 and figure 5.4. This calculator helps in understanding the RSA theory.





Figure 5.2 The greeting message

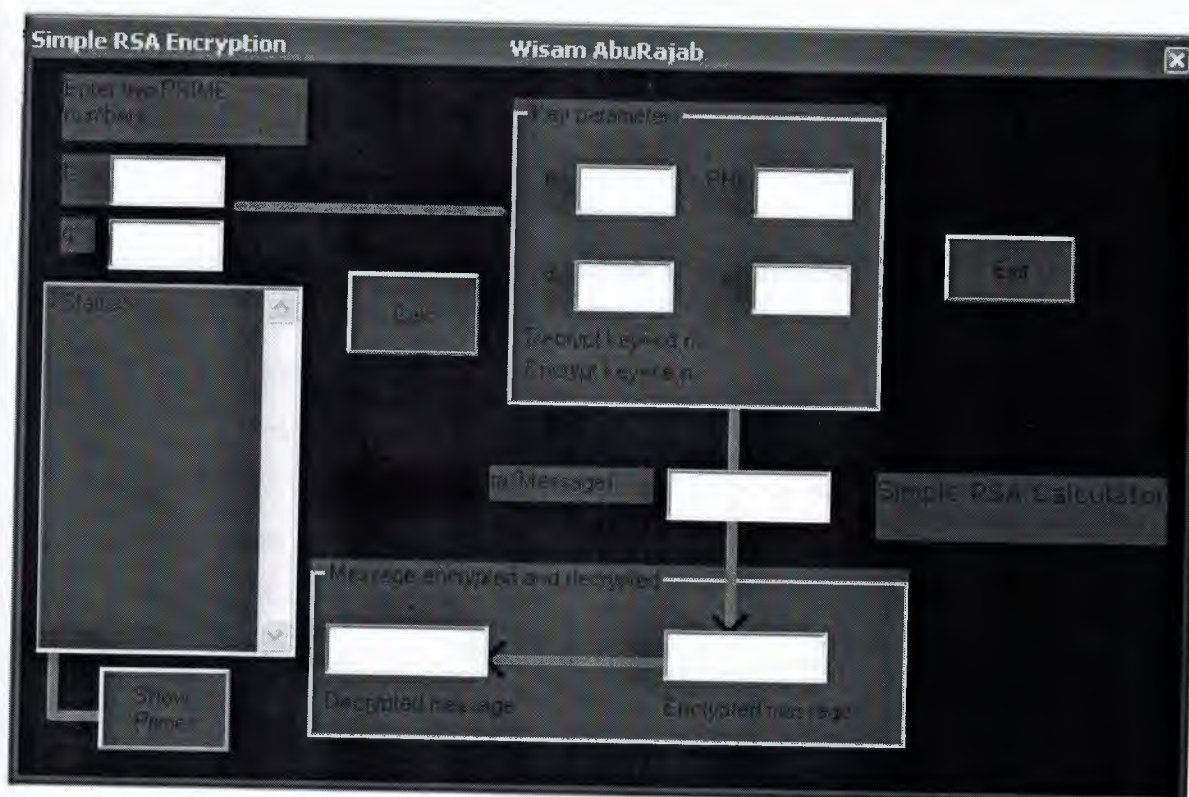


Figure 5.3 The calculator before test

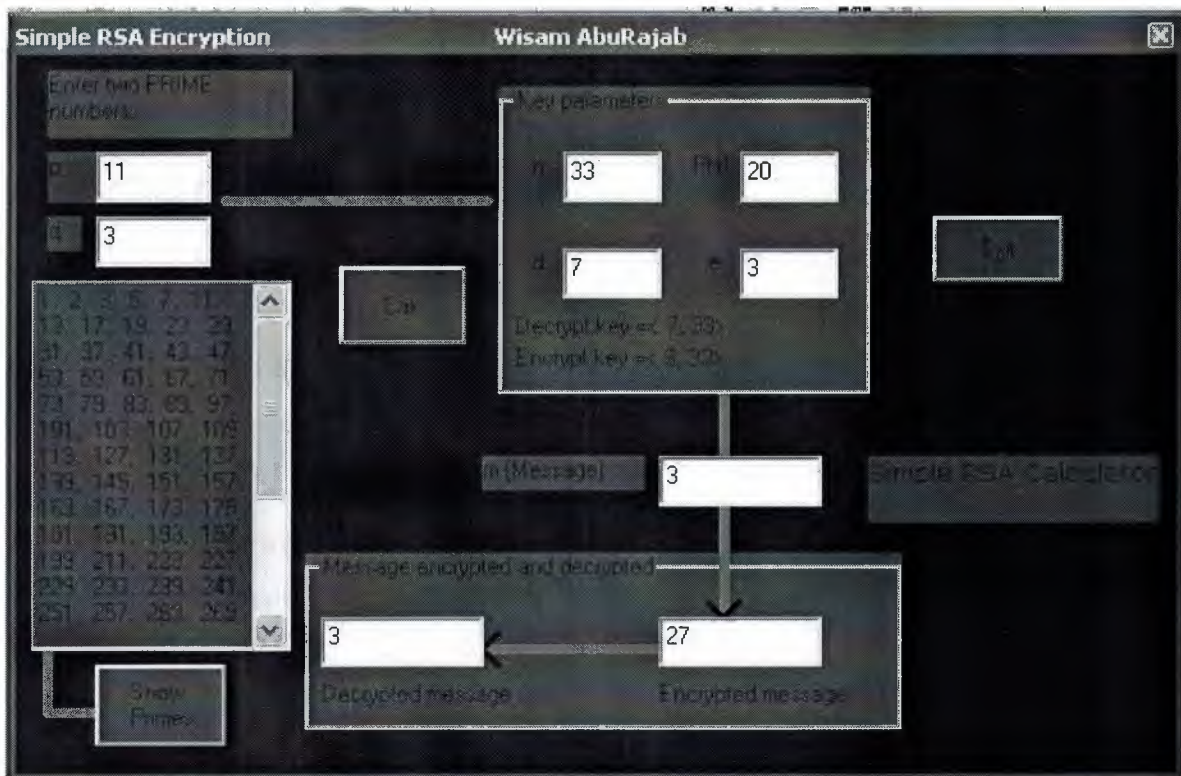


Figure 5.4 The Calculator after being tested.

### 5.3. The Client/Server encryption/decryption program

The WAP security relies on the encryption of the server and decrypting the client, in this case SSL encryption takes place. In order to receive payment information (credit cards) over the Internet, for example, encryption is required. Our advanced security features are provided using the open SSL protocol, which has been published on the Internet and adopted by major providers of Internet hardware and software products, financial institutions, and certification authorities. We offer Server Authentication, which allows any SSL-Compatible client to verify the identity of the server using a certificate and a digital signature; data encryption, which ensures the privacy of client-server communications by encrypting the data stream between the two entities (user's computer and web site); and data integrity, which verifies that the contents of a message arrive at its destination in the same form in which it was sent.

#### 5.3.1 The Aim of The Program

Our secured server provides integrated features designed to enable secure electronic commerce and communications. These features are required if you want to deploy an on-line payment system, prevent data from being intercepted on the internet/network, or



to restrict information to a certain group of individuals. User authorization controls access to individual files or directories using:

- User name and Password
- Client Certificates
- Client Certificates via VPN Servers
- Two-Factor Authentication Systems (Password and Crypto Card)
- PKI digital certificates
- Domain Name (xxx.com, xxx.org, xxx.net; etc. address)
- Host Name
- Client's Network IP address
- Named groups

### 5.3.2 The Details Of The Program

The program is divided to two main parts the Client and Server as shown in figures 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14 and 5.15

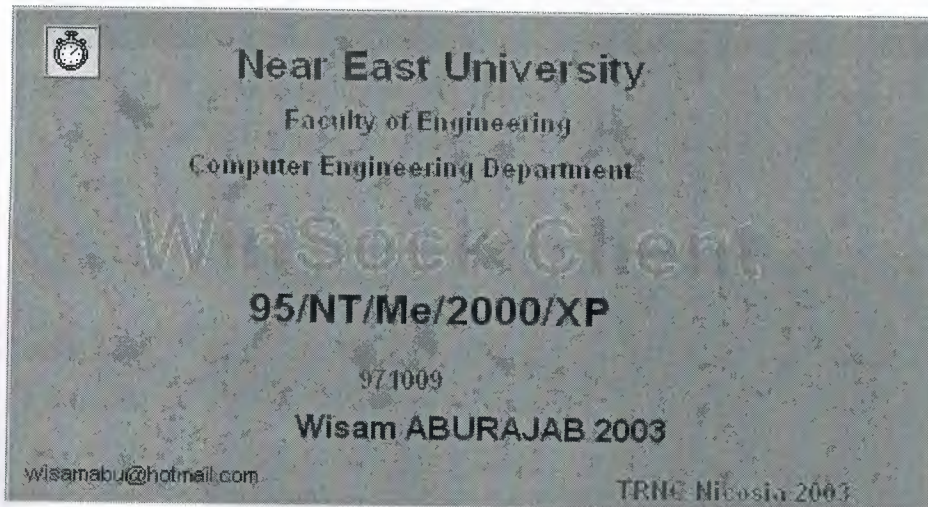


Figure 5.5 TCP/IP Client Welcome Message



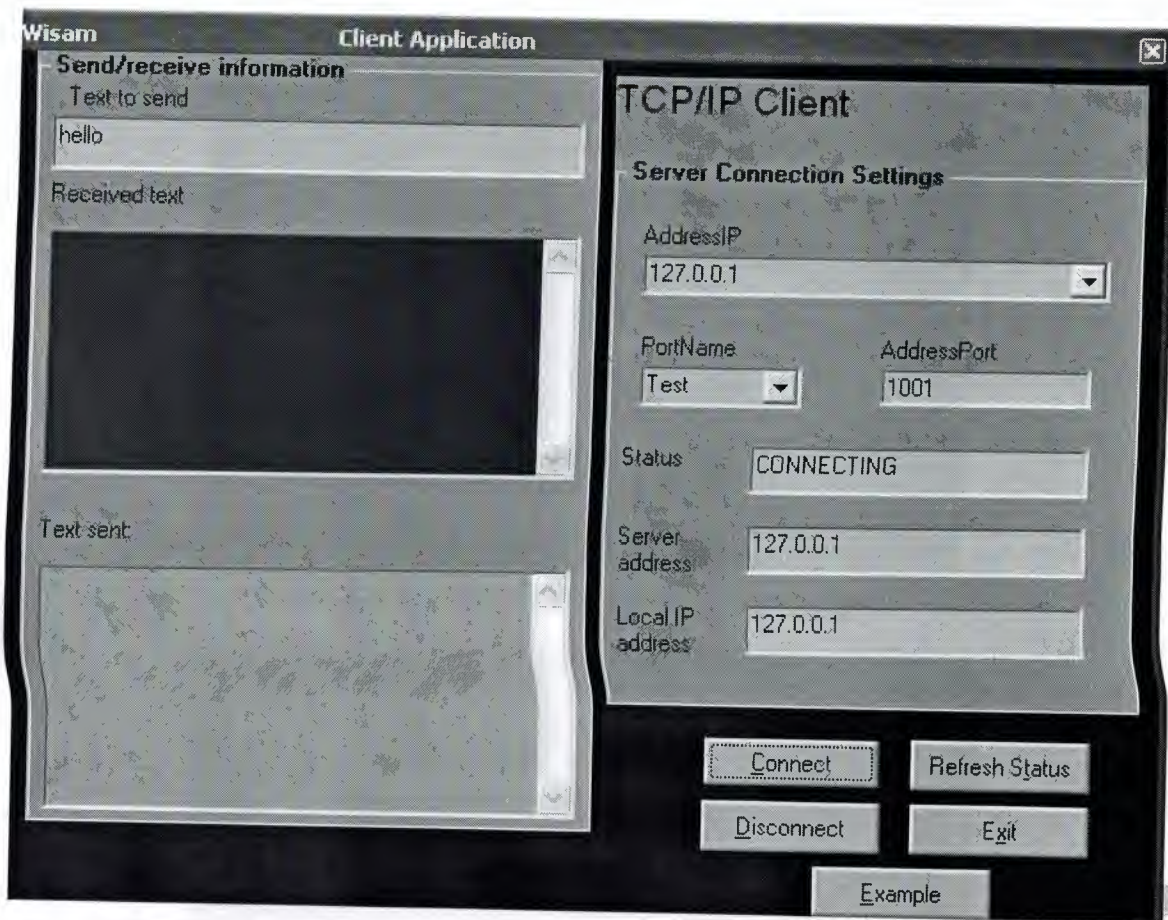


Figure 5.6 TCP/IP Client during running



Figure 5.7 HTTP Client welcome message





Figure 5.8 Simple HTTP Client at Work



Figure 5.9 FTP Client welcome message

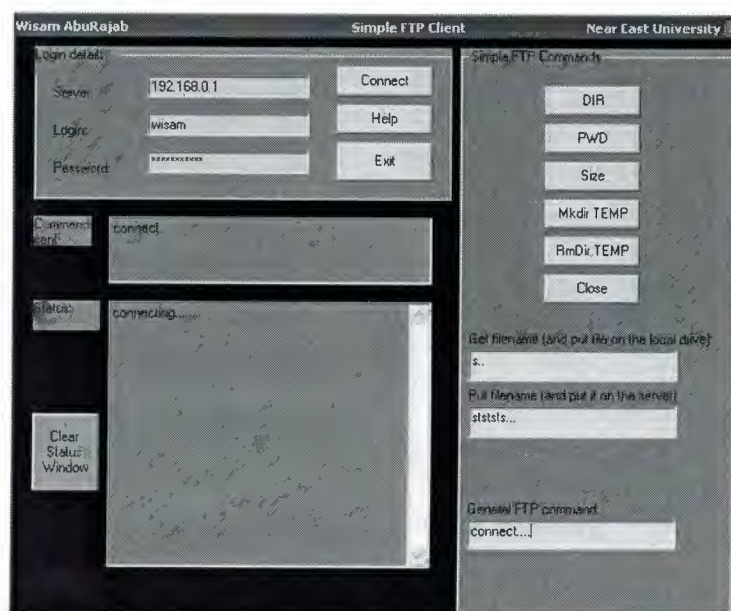


Figure 5.10 Simple FTP Client at work





Figure 5.11 Simple Server welcome message

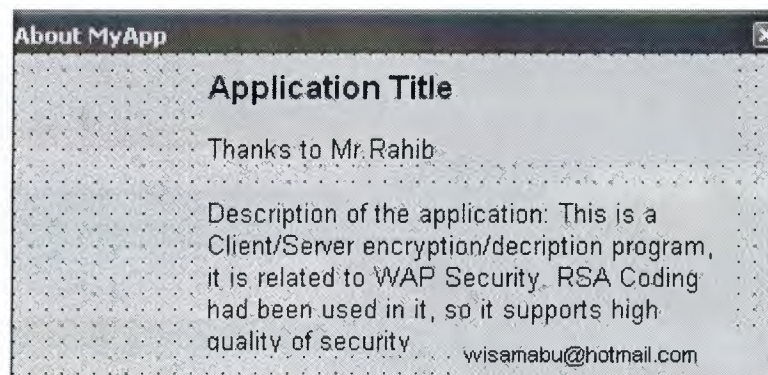


Figure 5.12 Application Title

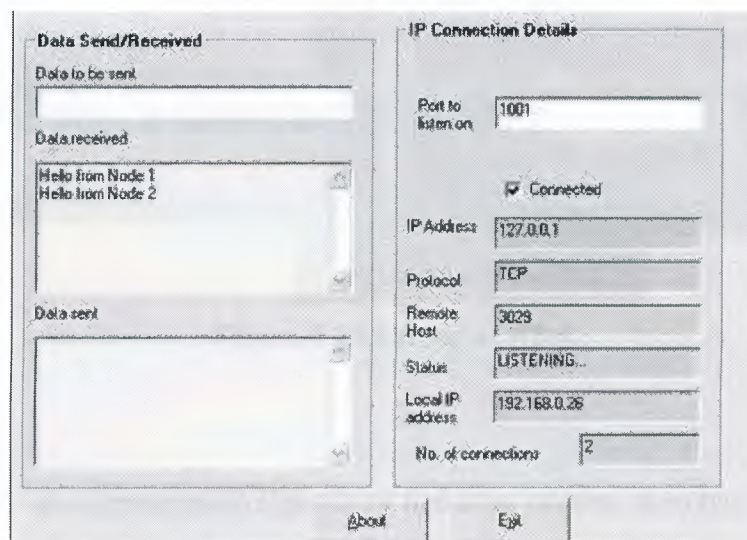


Figure 5.13 Server while running



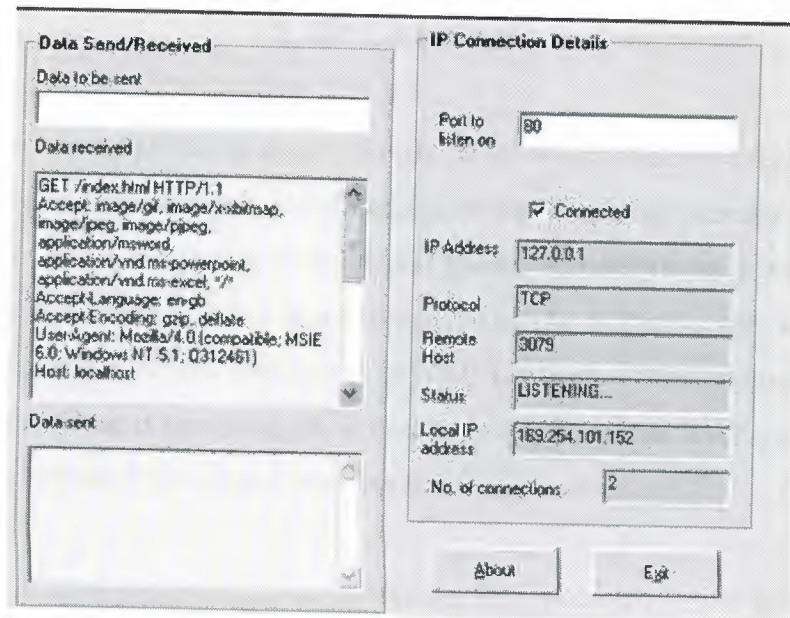


Figure 5.14 Server Connecting

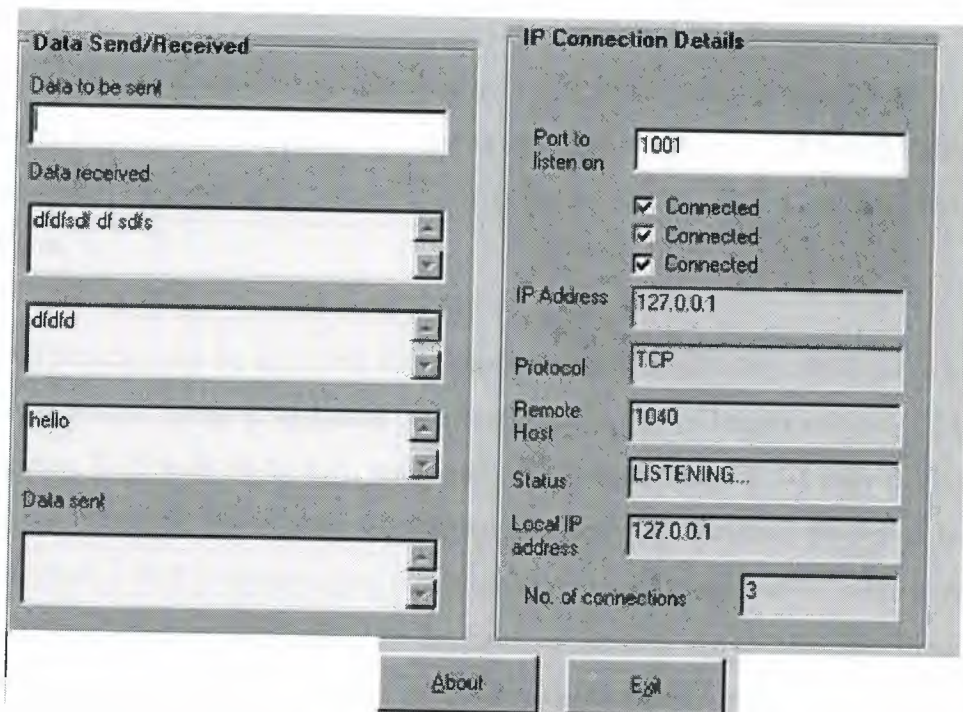


Figure 5.15 Server at work

## 5.4 Summary

In this chapter, the cryptographic algorithms had been revised, then two practical programs had been introduced with their inputs and outputs that had been captured after running the programs. The first program was a simple RSA calculator program, while the second was a multifunctional Client/Server program. The two programs had been done under the visual basic compiler.

## CONCLUSION

There has been a lot of speaking about security in the WAP, some of it justified, but most of it being misinformation and misunderstanding. Some had mentioned that there is no security in WAP, but the fact is there was always security in the WAP while the problem was whether that security is sufficient and can be trusted or not. WTLS was part of WAP 1.1 and is almost unchanged in WAP 1.2 assures that security has been there all the time. Even if there was no WTLS in an implemented WAP gateway security can be obtained from some vendors by encryption.

WTLS based on some common protocols like the TLS version 1.0, and SSL version 3.0. TLS has some properties that help to be implemented in the wireless world. The WTLS also, supports a span of algorithms that meet the requirements of authentication, integrity and privacy. Privacy is implemented using block ciphers, such as DES\_CBC, IDEA, and RC5\_CBC. The authentication is supplied by using of RSA-and Diffie-Hellman-based key exchange. Integrity is implemented with SHA-1 and MD5 MAC algorithms.

Sufficient security can be achieved if the supported algorithms are combined in suitable way. For example, using RSA-based certificates and a key of length at least 1024 bits, a block cipher, RC5 with 56-bit key, and a full MAC algorithm, SHA-1 may not be a sufficient solution for the security threads because the WTLS allows the server or the client to select a NULL-cipher algorithm which causes communication without ensuring authentication, integrity or privacy. So the balance between the chosen algorithms should happen. It would be better if a public key exchange algorithm which uses only 512 bits is chosen with a block cipher uses 168 bits.

Conclude that WTLS contains some significant security holes and there is a need to some drastic architectural changes, since we had identified some of these problems in the WAP WTLS protocol like a chosen data attack of a plaintext, data truncation attack, a message forgery attack and key-search shortcut for some exportable keys.



Another thing to be mentioned is preventing man-in-the-middle attack by denying the anonymous authentication at least from the server side. Regardless of the defects in the SSL and the TLS, they have gained a remarkable popularity providing security for the connection-oriented communication.

In order to manage the problem of security in the WAP a server/client program had been implemented and tested successfully. The program has many uses among of them is the WAP. It may act as any server or any client. Also, another small program had been programmed in order to calculate RSA. The two programs worked successfully to form a step towards the solution of the security in WAP.

Cryptographic algorithm is a WAP server security where application cryptography to RSA algorithm is carried out.



## REFERENCES

- [1] WAP Forum, Wireless Application Environment overview, 16.6.1999-2.10.1996. <http://www.Wapforum.org>.
- [2] Cryptography and Network Security  
Principles and Practice Second Edition by William Stallings
- [3] Symmetric cryptographic system for data encryption by C. ADAMS in Apr 1996.
- [4] Amoroso, E., Fundamentals of Computer Security Technology, PTR Prentice Hall, Englewood Cliffs, New Jersey, P. 403, 1993
- [5] Dierks, T. and Allen C., the TLS Protocol, January 1999.
- [6] Freier A. O. & Karlton P. & Kocher P.C., the SSL Protocol Version 3.0, 18.10.1996.
- [7] Internet FAQ Consortium, Cryptography FAQ (03/10: Basic Cryptography)
- [8] Saarinen, Marku-Juhani, Attacks against the WTLS Protocol, 20.09.1999.
- [9] Schneider, B., Applied Cryptography, Second Edition, John Wiley & sons, Inc, 758, 1996-1999.
- [10] Proceedings of the Internet Society Symposium on Network and Distributed System Security by IDUP and SPKM in 1996.
- [11] Random sources for cryptographic systems by G.B. AGNEW in 1988.
- [12] An implementation for a fast public-key cryptosystem by G.B. AGNEW, R.C. MULLIN, I.M. ONYSZCHUK & S.A. VANSTONE in 1991.
- [13] A secure key distribution system by N. ALEXANDRIS, M. BURMESTER & V. CHRISSIKOPOULOS.
- [14] A Weakness in the 4.2BSD UNIX TCP/IP Software by R.T. Morris in 1985.
- [15] Security Problems in the TCP/IP Protocol Suite Vol. 19 by S.M. Bellovin in April 1989.
- [16] Handbook of Applied Cryptography by A. Menezes, P. van Oorschot, and S. Vanstone in 1996.
- [17] Symmetric cryptographic system for data encryption by C. ADAMS in Apr 1996.

- [18] Proceedings of the Internet Society Symposium on Network and Distributed System Security by IDUP and SPKM in 1996.
- [19] Random sources for cryptographic systems by G.B. AGNEW in 1988.
- [20] An implementation for a fast public-key cryptosystem by G.B. AGNEW, R.C. MULLIN, I.M. ONYSZCHUK & S.A. VANSTONE in 1991.
- [21] A secure key distribution system by N. ALEXANDRIS, M. BURMESTER & V. CHRISSIKOPOULOS.
- [22] A Weakness in the 4.2BSD UNIX TCP/IP Software by R.T. Morris in 1985.
- [23] Security Problems in the TCP/IP Protocol Suite Vol. 19 by S.M. Bellovin in April 1989.
- [24] Handbook of Applied Cryptography by A. Menezes, P. van Oorschot, and S. Vanstone in 1996.
- [25] Virtual Private Networks Second Edition by Charlie Scott, Paul Wolf And Mike Ervin.
- [26] The Wassenaar Arrangement, List of Dual-Use Goods and Technologies - Category 5 - Part 2 - "Information Technology", 3.12.1998, [Referred 10.11.1999].
- [27] The Graduate Studies: A Complete Reference, Near East University, Lefkosa, TRNC, Mersin-10, Turkey edited by: Senol Bektas, Fakhreddin Mamedov, and Adnan Khasman, ISBN 8359-06-01, Nicosia 2001.

## APPENDIX-A

In this part of the thesis lie source codes of the programs that had been done and tested.

### TCP Client

```
Private Sub show_status()
    If (myTCPClient.State = sckClosed) Then
        status.Text = "CLOSED"
    ElseIf (myTCPClient.State = sckOpen) Then
        status.Text = "OPEN"
    ElseIf (myTCPClient.State = sckListen) Then
        status.Text = "LISTENING..."
    ElseIf (myTCPClient.State = sckConnecting) Then
        status.Text = "CONNECTING"
    ElseIf (myTCPClient.State = sckConnected) Then
        status.Text = "CONNECTED"
    ElseIf (myTCPClient.State = sckError) Then
        status.Text = "ERROR"
    Else
        status.Text = myTCPClient.State
    End If
End Sub

Private Sub cmdConnect_Click()
    If (myTCPClient.State <> sckClosed) Then myTCPClient.Close ' close existing
connection
    'Connect to the server
    myTCPClient.Connect
    server_address.Text = AddressIP.Text
    Call show_status
End Sub

Private Sub cmdDisconnect_Click()
    'Disconnect from the server
    myTCPClient.Close
    myTCPClient.RemoteHost = AddressIP.Text
    myTCPClient.RemotePort = AddressPort.Text
    Call show_status
End Sub

Private Sub Command1_Click()
    Call show_status
End Sub

Private Sub Form_Load()
    portnamec.AddItem ("Test")
    portnamec.AddItem ("Echo")
    portnamec.AddItem ("Daytime")
    portnamec.AddItem ("FTP")
    portnamec.AddItem ("SMTP")
    portnamec.AddItem ("Telnet")
    portnamec.AddItem ("Char. gen.")
    portnamec.AddItem ("Port 37")
    portnamec.AddItem ("WWW")
```



```

portnamec.Text = "Test"
AddressIP.AddItem "127.0.0.1"
AddressPort.Text = "1001"
localipaddress.Text = myTCPClient.LocalIP
Call show_status
End Sub
Private Sub HelpClient_Click()
    If (myTCPClient.State <> sckClosed) Then myTCPClient.Close ' close existing
connection
    myTCPClient.RemoteHost = "www.neu.edu.tr"
    AddressIP.Text = "www.neu.edu.tr"
    AddressPort.Text = "13"
    portnamec.Text = "Daytime"
    myTCPClient.Connect
End Sub
Private Sub Label2_Click()

End Sub
Private Sub PortNameC_Click()
    'Choice of the port (name)
    If portnamec.Text = "Test" Then AddressPort.Text = "1001"
    If portnamec.Text = "Echo" Then AddressPort.Text = "7"
    If portnamec.Text = "Daytime" Then AddressPort.Text = "13"
    If portnamec.Text = "FTP" Then AddressPort.Text = "21"
    If portnamec.Text = "Telnet" Then AddressPort.Text = "23"
    If portnamec.Text = "SMTP" Then AddressPort.Text = "25"
    If portnamec.Text = "Char. gen." Then AddressPort.Text = "19"
    If portnamec.Text = "Port 37" Then AddressPort.Text = "37"
    If portnamec.Text = "WWW" Then AddressPort.Text = "80"
End Sub
Private Sub myTCPClient_DataArrival(ByVal bytesTotal As Long)
    'Display incoming data
    Dim str1 As String, str2 As String, str As String 'declare old, new, to-tal data
    str1 = ShowText.Text 'old data
    myTCPClient.GetData str2 'incoming data (new data)
    str = str1 + str2 'total data to display
    ShowText.Text = str 'display to ShowText
End Sub
Private Sub myTCPClient_Close()
    If (myTCPClient.State = sckClosed) Or (myTCPClient.State = sckClosing) Then
        Call show_status
    Else
        myTCPServer.Close
    End If
End Sub
Private Sub AddressIP_Click()
    If (myTCPClient.State <> sckClosed) Then myTCPClient.Close ' close existing
connection
    'Choose IP Address

```

```

    myTCPClient.RemoteHost = AddressIP.Text
End Sub
Private Sub AddressIP_Change()
    If (myTCPClient.State <> sckClosed) Then myTCPClient.Close ' close existing
connection
    'Enter IP or DNS address
    myTCPClient.RemoteHost = AddressIP.Text
End Sub
Private Sub AddressPort_Change()
    If (myTCPClient.State <> sckClosed) Then myTCPClient.Close ' close existing
connection
    'Change port number directly in the AddressPort box (manually)
    myTCPClient.RemotePort = AddressPort.Text
End Sub
Private Sub CloseC_Click()
    'Return to main menu
End
End Sub
Private Sub SendTextData_KeyPress(KeyAscii As Integer)
    'When you press the ENTER key the contain of the top box is sent
    If KeyAscii = 13 Then
        myTCPClient.SendData SendTextData.Text + vbCrLf
        show_text_sent = show_text_sent + SendTextData.Text + vbCrLf
        SendTextData.Text = ""
    End If
End Sub
Option Explicit
Private Sub Form_Click()
    Unload Me
End Sub
Private Sub Form_KeyPress(KeyAscii As Integer)
    Unload Me
End Sub
Private Sub Form_Load()
    lblwisam.Caption = "Wisam Thanks Mr. Rahib "
    Load Me
End Sub
Private Sub Form_Unload(Cancel As Integer)
    myClient.Show
End Sub
Private Sub Frame1_Click()
    Unload Me
End Sub
Private Sub lblCompanyProduct_Click()
    Unload Me
End Sub
Private Sub lblPlatform_Click()
    Unload Me
End Sub

```

```

Private Sub lblProductName_Click()
    Unload Me
End Sub
Private Sub Picture1_Click()
    Unload Me
End Sub
Private Sub Timer1_Timer()
    Unload Me
End Sub
Option Explicit
' Reg Key Security Options...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
    KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
    KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL
' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1 ' Unicode nul terminated string
Const REG_DWORD = 4 ' 32-bit number
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"
Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA"
    (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal
    samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias
    "RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal
    lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData
    As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As
    Long
Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub
Private Sub cmdOK_Click()
    Unload Me
End Sub
Private Sub Command1_Click()
    frmBrowser.Show
End Sub
Private Sub Form_Load()

```



```

Me.Caption = "About " & App.Title
lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
App.Revision
lblTitle.Caption = App.Title
lblDescription.Caption = App.Comments
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr
    Dim rc As Long
    Dim SysInfoPath As String
    ' Try To Get System Info Program Path\Name From Registry...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
        ' Try To Get System Info Program Path Only From Registry...
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
            ' Validate Existence Of Known 32 Bit File Version
            If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
                SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
                ' Error - File Can Not Be Found...
            Else
                GoTo SysInfoErr
            End If
        ' Error - Registry Entry Can Not Be Found...
    Else
        GoTo SysInfoErr
    End If
    Call Shell(SysInfoPath, vbNormalFocus)
    Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As
String, ByRef KeyVal As String) As Boolean
    Dim i As Long                ' Loop Counter
    Dim rc As Long                ' Return Code
    Dim hKey As Long              ' Handle To An Open Registry Key
    Dim hDepth As Long           '
    Dim KeyValType As Long        ' Data Type Of A Registry Key
    Dim tmpVal As String          ' Tempory Storage For A Registry Key
    Value
    Dim KeyValSize As Long        ' Size Of Registry Key Variable
    '-----
    ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
    '-----
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open
Registry Key

```

```

If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle Error...

tmpVal = String$(1024, 0)                          ' Allocate Variable Space
KeyValSize = 1024                                  ' Mark Variable Size

'-----
' Retrieve Registry Key Value...
'-----
rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
    KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value

If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle Errors

If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then      ' Win95 Adds Null
Terminated String...
    tmpVal = Left(tmpVal, KeyValSize - 1)          ' Null Found, Extract From String
Else                                                ' WinNT Does NOT Null Terminate String...
    tmpVal = Left(tmpVal, KeyValSize)              ' Null Not Found, Extract String
Only
End If

'-----
' Determine Key Value Type For Conversion...
'-----
Select Case KeyValType                            ' Search Data Types...
Case REG_SZ                                       ' String Registry Key Data Type
    KeyVal = tmpVal                               ' Copy String Value
Case REG_DWORD                                   ' Double Word Registry Key Data
Type
    For i = Len(tmpVal) To 1 Step -1              ' Convert Each Bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Build Value Char. By Char.
    Next
    KeyVal = Format$("&h" + KeyVal)                 ' Convert Double Word To String
End Select

GetKeyValue = True                                ' Return Success
rc = RegCloseKey(hKey)                            ' Close Registry Key
Exit Function                                       ' Exit

GetKeyError: ' Cleanup After An Error Has Occured...
    KeyVal = ""                                     ' Set Return Val To Empty String
    GetKeyValue = False                             ' Return Failure
    rc = RegCloseKey(hKey)                          ' Close Registry Key
End Function

Option Explicit

Public startingaddress As String
Dim mbDontNavigateNow As Boolean

```

```

Private Sub Form_Load()
    On Error Resume Next
    Me.Show
    tbToolBar.Refresh
    Form_Resize

    cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15
    startingaddress = "http://www.neu.edu.tr/~wisam/vb_winsock.htm"
    If Len(startingaddress) > 0 Then
        cboAddress.Text = startingaddress
        cboAddress.AddItem cboAddress.Text
        'try to navigate to the starting address
        timTimer.Enabled = True
        brwWebBrowser.Navigate startingaddress
    End If
End Sub

Private Sub brwWebBrowser_DownloadComplete()
    On Error Resume Next
    Me.Caption = brwWebBrowser.LocationName
End Sub

Private Sub brwWebBrowser_NavigateComplete(ByVal URL As String)
    Dim i As Integer
    Dim bFound As Boolean
    Me.Caption = brwWebBrowser.LocationName
    For i = 0 To cboAddress.ListCount - 1
        If cboAddress.List(i) = brwWebBrowser.LocationURL Then
            bFound = True
            Exit For
        End If
    Next i
    mbDontNavigateNow = True
    If bFound Then
        cboAddress.RemoveItem i
    End If
    cboAddress.AddItem brwWebBrowser.LocationURL, 0
    cboAddress.ListIndex = 0
    mbDontNavigateNow = False
End Sub

Private Sub cboAddress_Click()
    If mbDontNavigateNow Then Exit Sub
    timTimer.Enabled = True
    brwWebBrowser.Navigate cboAddress.Text
End Sub

Private Sub cboAddress_KeyPress(KeyAscii As Integer)
    On Error Resume Next
    If KeyAscii = vbKeyReturn Then
        cboAddress_Click
    End If
End Sub

```



```

Private Sub Form_Resize()
    cboAddress.Width = Me.ScaleWidth - 100
    brwWebBrowser.Width = Me.ScaleWidth - 100
    brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top + picAddress.Height) -
100
End Sub
Private Sub timTimer_Timer()
    If brwWebBrowser.Busy = False Then
        timTimer.Enabled = False
        Me.Caption = brwWebBrowser.LocationName
    Else
        Me.Caption = "Working..."
    End If
End Sub

```

```

Private Sub tbToolBar_ButtonClick(ByVal Button As Button)
    On Error Resume Next

```

```

    timTimer.Enabled = True
    Select Case Button.Key
        Case "Back"
            brwWebBrowser.GoBack
        Case "Forward"
            brwWebBrowser.GoForward
        Case "Refresh"
            brwWebBrowser.Refresh
        Case "Home"
            brwWebBrowser.GoHome
        Case "Search"
            brwWebBrowser.GoSearch
        Case "Stop"
            timTimer.Enabled = False
            brwWebBrowser.Stop
            Me.Caption = brwWebBrowser.LocationName
    End Select
End Sub

```

### **HTTP Client**

```

Private Sub button_get_Click()
    Inet1.AccessType = icUseDefault
    Inet1.Protocol = icHTTP
    Inet1.Execute "http://" + text_url.Text + "/" + text_file.Text, "GET"
    command_window.Text = "http://" + text_url.Text + "/" + text_file.Text + ", GET"
End Sub

```

```

Private Sub button_clear_Click()
    status.Text = ""
End Sub

```

```

Private Sub button_head_Click()

```

```

    Inet1.AccessType = icUseDefault
    Inet1.URL = "http://" + text_url.Text + "/" + text_file.Text
    Inet1.Execute Inet1.URL, "HEAD"
End Sub

Private Sub button_help_Click()
    frmBrowser.Show
End Sub

Private Sub button_exit_Click()
    End
End Sub

Private Sub http_command_text_box_Change()
    Inet1.Execute , http_command_text_box.Text
    command_window.Text = command_window.Text + Text4.Text
End Sub

Private Sub Inet1_StateChanged(ByVal State As Integer)
    Select Case State
        Case icNone
            ' Ignore, no error
        Case icHostResolvingHost
            status.Text = status.Text + "Resolving host... " + vbCrLf
        Case icHostResolved
            status.Text = status.Text + "Resolved host... " + vbCrLf
        Case icConnecting
            status.Text = status.Text + "Connecting to host... " + vbCrLf
        Case icConnected
            status.Text = status.Text + "Connected to host... " + vbCrLf

        Case icError
            status.Text = status.Text + "ErrorCode: " & Inet1.ResponseCode & " : " &
Inet1.ResponseInfo
        Case icResponseCompleted
            Dim data As Variant
            data = Inet1.GetChunk(1024, icString)
            Do While LenB(data) > 0
                status.Text = status.Text + data
                data = Inet1.GetChunk(1024, icString)
            Loop
            status.Text = status.Text + vbCrLf
    End Select

End Sub

Private Sub Label5_Click()

End Sub

```

```

Private Sub status_Change()

End Sub
Option Explicit

Private Sub Form_Click()
    Unload Me
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    Unload Me
End Sub

Private Sub Form_Load()

    lblProductName.Caption = "HTTP Client"
End Sub

Private Sub Form_Unload(Cancel As Integer)
    HTTPclient.Show
End Sub

Private Sub Frame1_Click()
    Unload Me
End Sub

Private Sub lblCompanyProduct_Click()
    Unload Me
End Sub

Private Sub lblPlatform_Click()
    Unload Me
End Sub

Private Sub lblProductName_Click()
    Unload Me
End Sub

Private Sub lblVersion_Click()
    Unload Me
End Sub

Private Sub Picture1_Click()
    Unload Me
End Sub

Private Sub Timer1_Timer()
    Unload Me
End Sub

```



## Option Explicit

```
Public startingaddress As String
Dim mbDontNavigateNow As Boolean
Private Sub Form_Load()
    On Error Resume Next
    Me.Show
    tbToolBar.Refresh
    Form_Resize

    cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15
    startingaddress = "http://www.neu.edu.tr/~wisam/vb_winsock_http.htm"
    If Len(startingaddress) > 0 Then
        cboAddress.Text = startingaddress
        cboAddress.AddItem cboAddress.Text
        'try to navigate to the starting address
        timTimer.Enabled = True
        brwWebBrowser.Navigate startingaddress
    End If
End Sub
```

```
Private Sub brwWebBrowser_DownloadComplete()
    On Error Resume Next
    Me.Caption = brwWebBrowser.LocationName
End Sub
```

```
Private Sub brwWebBrowser_NavigateComplete(ByVal URL As String)
    Dim i As Integer
    Dim bFound As Boolean
    Me.Caption = brwWebBrowser.LocationName
    For i = 0 To cboAddress.ListCount - 1
        If cboAddress.List(i) = brwWebBrowser.LocationURL Then
            bFound = True
            Exit For
        End If
    Next
    imbDontNavigateNow = True
    If bFound Then
        cboAddress.RemoveItem i
    End If
    cboAddress.AddItem brwWebBrowser.LocationURL, 0
    cboAddress.ListIndex = 0
    mbDontNavigateNow = False
End Sub
```

```
Private Sub cboAddress_Click()
    If mbDontNavigateNow Then Exit Sub
```

```
timTimer.Enabled = True
brwWebBrowser.Navigate cboAddress.Text
End Sub
```

```
Private Sub cboAddress_KeyPress(KeyAscii As Integer)
    On Error Resume Next
    If KeyAscii = vbKeyReturn Then
        cboAddress_Click
    End If
End Sub
```

```
Private Sub Form_Resize()
    cboAddress.Width = Me.ScaleWidth - 100
    brwWebBrowser.Width = Me.ScaleWidth - 100
    brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top + picAddress.Height) -
100
End Sub
```

```
Private Sub timTimer_Timer()
    If brwWebBrowser.Busy = False Then
        timTimer.Enabled = False
        Me.Caption = brwWebBrowser.LocationName
    Else
        Me.Caption = "Working..."
    End If
End Sub
```

```
Private Sub tbToolBar_ButtonClick(ByVal Button As Button)
    On Error Resume Next

    timTimer.Enabled = True

    Select Case Button.Key
        Case "Back"
            brwWebBrowser.GoBack
        Case "Forward"
            brwWebBrowser.GoForward
        Case "Refresh"
            brwWebBrowser.Refresh
        Case "Home"
            brwWebBrowser.GoHome
        Case "Search"
            brwWebBrowser.GoSearch
        Case "Stop"
            timTimer.Enabled = False
            brwWebBrowser.Stop
            Me.Caption = brwWebBrowser.LocationName
    End Select
```

End Sub

### **FTP Client**

Dim filetransfer As Boolean

Private Sub button\_connect\_Click()

Inet1.AccessType = icUseDefault

Inet1.Protocol = icFTP

Inet1.URL = text1.Text

Inet1.UserName = Text2.Text

Inet1.Password = Text3.Text

filetransfer = False

End Sub

Private Sub button\_help\_Click()

frmBrowser.Show

End Sub

Private Sub button\_exit\_Click()

End

End Sub

Private Sub button\_dir\_Click()

Inet1.Execute, "LS"

command\_window.Text = command\_window.Text + "LS:" + vbCrLf

End Sub

Private Sub button\_close\_Click()

Inet1.Execute, "CLOSE"

command\_window.Text = command\_window.Text + "CLOSE:" + vbCrLf

End Sub

Private Sub button\_pwd\_Click()

Inet1.Execute, "PWD"

command\_window.Text = command\_window.Text + "PWD:" + vbCrLf

End Sub

Private Sub button\_size\_Click()

Inet1.Execute, "SIZE ."

command\_window.Text = command\_window.Text + "SIZE:" + vbCrLf

End Sub

Private Sub button\_mkdir\_Click()

Inet1.Execute, "MKDIR TEMP"

command\_window.Text = command\_window.Text + "Mkdir temp:" + vbCrLf

End Sub

Private Sub button\_rmdir\_Click()

Inet1.Execute, "RMDIR TEMP"



```
command_window.Text = command_window.Text + "Rmdir temp:" + vbCrLf  
End Sub
```

```
Private Sub button_status_Click()  
    status.Text = ""  
End Sub
```

```
Private Sub Inet1_StateChanged(ByVal State As Integer)  
    Select Case State  
        Case icNone  
            ' Ignore, no error  
        Case icHostResolvingHost  
            status.Text = status.Text + "Resolving host..." + vbCrLf  
        Case icHostResolved  
            status.Text = status.Text + "Resolved host..." + vbCrLf  
        Case icConnecting  
            status.Text = status.Text + "Connecting to host..." + vbCrLf  
        Case icConnected  
            status.Text = status.Text + "Connected to host..." + vbCrLf  
  
        Case icError  
            status.Text = status.Text + "ErrorCode: " & Inet1.ResponseCode & " : " &  
Inet1.ResponseInfo  
  
        Case icResponseCompleted  
            Dim data As Variant  
            If (filetransfer = True) Then  
                Open text_get_filename.Text For Binary Access Write As #1  
            End If  
  
            data = Inet1.GetChunk(1024, icString)  
            Do While LenB(data) > 0  
                If (filetransfer = True) Then  
                    Put #1, , data  
                End If  
                status.Text = status.Text + data  
                data = Inet1.GetChunk(1024, icString)  
            Loop  
            If (filetransfer = True) Then  
                Put #1, , data  
            End If  
            status.Text = status.Text + vbCrLf  
            filetransfer = False  
        End Select  
End Sub
```

End Sub

Private Sub Label6\_Click()

End Sub

Private Sub status\_Change()

End Sub

Private Sub text\_ftp\_command\_KeyPress(KeyAscii As Integer)

    If KeyAscii = 13 Then

        Call Inet1.Execute(, text\_ftp\_command.Text)

        command\_window.Text = command\_window.Text + text\_ftp\_command.Text

    End If

End Sub

Private Sub text\_get\_filename\_KeyPress(KeyAscii As Integer)

Dim str As String

    If KeyAscii = 13 Then

        str = "RETR " + text\_get\_filename.Text + " " + text\_get\_filename.Text

        Call Inet1.Execute(, str)

        command\_window.Text = command\_window.Text + vbNewLine + str

    End If

    filetransfer = True

End Sub

Private Sub text\_put\_filename\_KeyPress(KeyAscii As Integer)

Dim str As String

    If KeyAscii = 13 Then

        str = "Put c:\" + text\_put\_filename.Text + " " + text\_put\_filename.Text

        Inet1.Execute , str

        command\_window.Text = command\_window.Text + vbNewLine + str

    End If

End Sub

Option Explicit

Private Sub Form\_Click()

    Unload Me

End Sub

Private Sub Form\_KeyPress(KeyAscii As Integer)

    Unload Me

End Sub

Private Sub Form\_Load()

```
    lblProductName.Caption = "FTP Client"  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
    FTPclient.Show  
End Sub
```

```
Private Sub Frame1_Click()  
    Unload Me  
End Sub
```

```
Private Sub Label2_Click()  
  
End Sub
```

```
Private Sub lblCompanyProduct_Click()  
    Unload Me  
End Sub
```

```
Private Sub lblPlatform_Click()  
    Unload Me  
End Sub
```

```
Private Sub lblProductName_Click()  
    Unload Me  
End Sub
```

```
Private Sub Picture1_Click()  
    Unload Me  
End Sub
```

```
Private Sub Timer1_Timer()  
    Unload Me  
End Sub  
Option Explicit
```

```
Public startingaddress As String  
Dim mbDontNavigateNow As Boolean  
Private Sub Form_Load()  
    On Error Resume Next  
    Me.Show  
    tbToolBar.Refresh  
    Form_Resize
```

```
cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15  
startingaddress = "http://www.neu.edu.tr/~wisam/vb_winsock_ftp.htm"
```



```

    If Len(startingaddress) > 0 Then
        cboAddress.Text = startingaddress
        cboAddress.AddItem cboAddress.Text
        'try to navigate to the starting address
        timTimer.Enabled = True
        brwWebBrowser.Navigate startingaddress
    End If

End Sub

Private Sub brwWebBrowser_DownloadComplete()
    On Error Resume Next
    Me.Caption = brwWebBrowser.LocationName
End Sub

Private Sub brwWebBrowser_NavigateComplete(ByVal URL As String)
    Dim i As Integer
    Dim bFound As Boolean
    Me.Caption = brwWebBrowser.LocationName
    For i = 0 To cboAddress.ListCount - 1
        If cboAddress.List(i) = brwWebBrowser.LocationURL Then
            bFound = True
            Exit For
        End If
    Next i
    mbDontNavigateNow = True
    If bFound Then
        cboAddress.RemoveItem i
    End If
    cboAddress.AddItem brwWebBrowser.LocationURL, 0
    cboAddress.ListIndex = 0
    mbDontNavigateNow = False
End Sub

Private Sub cboAddress_Click()
    If mbDontNavigateNow Then Exit Sub
    timTimer.Enabled = True
    brwWebBrowser.Navigate cboAddress.Text
End Sub

Private Sub cboAddress_KeyPress(KeyAscii As Integer)
    On Error Resume Next
    If KeyAscii = vbKeyReturn Then
        cboAddress_Click
    End If
End Sub

```

```

Private Sub Form_Resize()
    cboAddress.Width = Me.ScaleWidth - 100
    brwWebBrowser.Width = Me.ScaleWidth - 100
    brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top + picAddress.Height) -
100
End Sub

```

```

Private Sub timTimer_Timer()
    If brwWebBrowser.Busy = False Then
        timTimer.Enabled = False
        Me.Caption = brwWebBrowser.LocationName
    Else
        Me.Caption = "Working..."
    End If
End Sub

```

```

Private Sub tbToolBar_ButtonClick(ByVal Button As Button)
    On Error Resume Next

```

```

    timTimer.Enabled = True

```

```

    Select Case Button.Key
        Case "Back"
            brwWebBrowser.GoBack
        Case "Forward"
            brwWebBrowser.GoForward
        Case "Refresh"
            brwWebBrowser.Refresh
        Case "Home"
            brwWebBrowser.GoHome
        Case "Search"
            brwWebBrowser.GoSearch
        Case "Stop"
            timTimer.Enabled = False
            brwWebBrowser.Stop
            Me.Caption = brwWebBrowser.LocationName
    End Select

```

```

End Sub

```

## APPENDIX-B

### Program of the Server

```
Private Sub About_Click()  
    frmAbout.Show  
End Sub
```

```
Private Sub exit_Click()  
    End  
End Sub
```

```
Private Sub Form_Load()  
    ' Set the local port to 1001 and listen for a connection  
    listenport.Text = "1001"  
    localipaddress.Text = myTCPServer.LocalIP ' Show local IP address  
    Call show_status  
End Sub  
Private Sub show_status()
```

```
    If (myTCPServer.State = sckClosed) Then  
        status.Text = "CLOSED"  
    ElseIf (myTCPServer.State = sckOpen) Then  
        status.Text = "OPEN"  
    ElseIf (myTCPServer.State = sckListening) Then  
        status.Text = "LISTENING..."  
    ElseIf (myTCPServer.State = sckConnecting) Then  
        status.Text = "CONNECTING"  
    ElseIf (myTCPServer.State = sckConnected) Then  
        status.Text = "CONNECTED"  
    ElseIf (myTCPServer.State = sckError) Then  
        status.Text = "ERROR"  
    Else  
        status.Text = myTCPServer.State  
    End If  
End Sub
```

```
Private Sub listenport_Change()  
    If myTCPServer.State <> sckClosed Then myTCPServer.Close  
    myTCPServer.LocalPort = listenport.Text  
    myTCPServer.Listen  
    Call show_status  
End Sub
```

```
Private Sub myTCPServer_Close()  
    If myTCPServer.State <> sckClosed Then myTCPServer.Close
```



```

myTCPServer.LocalPort = listenport.Text
myTCPServer.Listen
ipaddress.Text = ""
iphost.Text = ""
remoteport.Text = ""
Call show_status
End Sub

```

```

Private Sub myTCPServer_ConnectionRequest(ByVal requestID As Long)
' Check state of socket, if it is not closed then close it.
If myTCPServer.State <> sckClosed Then myTCPServer.Close
' Accept the request with the requestID parameter.
myTCPServer.Accept requestID
ipaddress.Text = myTCPServer.RemoteHostIP
If (myTCPServer.Protocol = 0) Then
    iphost.Text = "TCP"
Else
    iphost.Text = "UDP"
End If
remoteport.Text = myTCPServer.remoteport
Check1.Value = 1 ' show that remote has connected
Call show_status
End Sub

```

```

Private Sub myTCPServer_DataArrival(ByVal bytesTotal As Long)
' Read incoming data into the str variable,
' then display it to ShowText
Dim str As String
myTCPServer.GetData str
ShowText.Text = ShowText.Text + str
Call show_status
End Sub

```

```

Private Sub SendTextData_KeyPress(KeyAscii As Integer)
Call show_status
If (KeyAscii = 13) Then
    myTCPServer.SendData SendTextData.Text + vbCrLf
    show_text_sent = show_text_sent + SendTextData.Text + vbCrLf
    SendTextData.Text = ""
End If
End Sub

```

Option Explicit

```

Private Sub Form_KeyPress(KeyAscii As Integer)

```

```
Unload Me
End Sub
```

```
Private Sub Form_Load()
    lblversion.Caption = "wisam "
    lblProductName.Caption = App.Title
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    myServer.Show
End Sub
```

```
Private Sub Frame1_Click()
    Unload Me
End Sub
```

```
Private Sub Picture1_Click()
    Unload Me
End Sub
```

```
Private Sub lblPlatform_Click()

End Sub
```

```
Private Sub lblversion_Click()
    Unload Me
End Sub
```

```
Private Sub Timer1_Timer()
    Unload Me
End Sub
Option Explicit
```

```
Public startingaddress As String
Dim mbDontNavigateNow As Boolean
Private Sub Form_Load()
    On Error Resume Next
    Me.Show
    tbToolBar.Refresh
    Form_Resize
```

```
    cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15
    startingaddress = "http://www.neu.edu.tr/~wisam/vb_winsock.htm"
    If Len(startingaddress) > 0 Then
        cboAddress.Text = startingaddress
        cboAddress.AddItem cboAddress.Text
        'try to navigate to the starting address
        timTimer.Enabled = True
        brwWebBrowser.Navigate startingaddress
```

End If

End Sub

```
Private Sub brwWebBrowser_DownloadComplete()  
    On Error Resume Next  
    Me.Caption = brwWebBrowser.LocationName  
End Sub
```

```
Private Sub brwWebBrowser_NavigateComplete(ByVal URL As String)  
    Dim i As Integer  
    Dim bFound As Boolean  
    Me.Caption = brwWebBrowser.LocationName  
    For i = 0 To cboAddress.ListCount - 1  
        If cboAddress.List(i) = brwWebBrowser.LocationURL Then  
            bFound = True  
            Exit For  
        End If  
    Next i  
    mbDontNavigateNow = True  
    If bFound Then  
        cboAddress.RemoveItem i  
    End If  
    cboAddress.AddItem brwWebBrowser.LocationURL, 0  
    cboAddress.ListIndex = 0  
    mbDontNavigateNow = False  
End Sub
```

```
Private Sub cboAddress_Click()  
    If mbDontNavigateNow Then Exit Sub  
    timTimer.Enabled = True  
    brwWebBrowser.Navigate cboAddress.Text  
End Sub
```

```
Private Sub cboAddress_KeyPress(KeyAscii As Integer)  
    On Error Resume Next  
    If KeyAscii = vbKeyReturn Then  
        cboAddress_Click  
    End If  
End Sub
```

```
Private Sub Form_Resize()  
    cboAddress.Width = Me.ScaleWidth - 100  
    brwWebBrowser.Width = Me.ScaleWidth - 100  
    brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top + picAddress.Height) -  
    100  
End Sub
```



```

Private Sub timTimer_Timer()
    If brwWebBrowser.Busy = False Then
        timTimer.Enabled = False
        Me.Caption = brwWebBrowser.LocationName
    Else
        Me.Caption = "Working..."
    End If
End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As Button)
    On Error Resume Next

    timTimer.Enabled = True

    Select Case Button.Key
        Case "Back"
            brwWebBrowser.GoBack
        Case "Forward"
            brwWebBrowser.GoForward
        Case "Refresh"
            brwWebBrowser.Refresh
        Case "Home"
            brwWebBrowser.GoHome
        Case "Search"
            brwWebBrowser.GoSearch
        Case "Stop"
            timTimer.Enabled = False
            brwWebBrowser.Stop
            Me.Caption = brwWebBrowser.LocationName
    End Select

End Sub

Option Explicit

' Reg Key Security Options...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
    KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
    KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

```

```

' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1           ' Unicode nul terminated string
Const REG_DWORD = 4        ' 32-bit number

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA"
    (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal
    samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias
    "RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal
    lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData
    As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As
    Long

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Me.Caption = "About " & App.Title
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
    App.Revision
    lblTitle.Caption = App.Title
    lblDescription.Caption = App.Comments
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr

    Dim rc As Long
    Dim SysInfoPath As String

    ' Try To Get System Info Program Path\Name From Registry...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
    gREGVALSYSINFO, SysInfoPath) Then
        ' Try To Get System Info Program Path Only From Registry...
    
```

```

ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
    ' Validate Existence Of Known 32 Bit File Version
    If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
        SysInfoPath = SysInfoPath & "\MSINFO32.EXE"

        ' Error - File Can Not Be Found...
    Else
        GoTo SysInfoErr
    End If
' Error - Registry Entry Can Not Be Found...
Else
    GoTo SysInfoErr
End If

Call Shell(SysInfoPath, vbNormalFocus)

Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

```

```

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As
String, ByRef KeyVal As String) As Boolean
    Dim i As Long                ' Loop Counter
    Dim rc As Long              ' Return Code
    Dim hKey As Long            ' Handle To An Open Registry Key
    Dim hDepth As Long          '
    Dim KeyValType As Long       ' Data Type Of A Registry Key
    Dim tmpVal As String         ' Tempory Storage For A Registry Key
Value
    Dim KeyValSize As Long       ' Size Of Registry Key Variable
    '-----
    ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
    '-----
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open
Registry Key

    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError    ' Handle Error...

    tmpVal = String$(1024, 0)                ' Allocate Variable Space
    KeyValSize = 1024                        ' Mark Variable Size

    '-----
    ' Retrieve Registry Key Value...
    '-----
    rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
        KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value

```



```

If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle Errors

If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then      ' Win95 Adds Null
Terminated String...
    tmpVal = Left(tmpVal, KeyValSize - 1)          ' Null Found, Extract From String
Else                                              ' WinNT Does NOT Null Terminate String...
    tmpVal = Left(tmpVal, KeyValSize)              ' Null Not Found, Extract String
Only
End If
'-----
' Determine Key Value Type For Conversion...
'-----
Select Case KeyValType                          ' Search Data Types...
Case REG_SZ                                     ' String Registry Key Data Type
    KeyVal = tmpVal                             ' Copy String Value
Case REG_DWORD                                ' Double Word Registry Key Data
Type
    For i = Len(tmpVal) To 1 Step -1            ' Convert Each Bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Build Value Char. By Char.
    Next
    KeyVal = Format$("&h" + KeyVal)              ' Convert Double Word To String
End Select

GetKeyValue = True                             ' Return Success
rc = RegCloseKey(hKey)                         ' Close Registry Key
Exit Function                                  ' Exit

GetKeyError: ' Cleanup After An Error Has Occured...
    KeyVal = ""                                ' Set Return Val To Empty String
    GetKeyValue = False                        ' Return Failure
    rc = RegCloseKey(hKey)                    ' Close Registry Key
End Function

```

```

Private Sub lblVersion_Click()

```

```

End Sub

```

## APPENDIX-C

### Program of the RSA Calculator

Option Explicit

```
Private Sub Form_Click()  
    Unload Me  
End Sub
```

```
Private Sub Form_KeyPress(KeyAscii As Integer)  
    Unload Me  
End Sub
```

```
Private Sub Form_Load()  
  
    lblProductName.Caption = "RSA Program"  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
    Form1.Show  
End Sub
```

```
Private Sub Frame1_Click()  
    Unload Me  
End Sub
```

```
Private Sub Label2_Click()  
  
End Sub
```

```
Private Sub lblCompanyProduct_Click()  
    Unload Me  
End Sub
```

```
Private Sub lblPlatform_Click()  
    Unload Me  
End Sub
```

```
Private Sub lblProductName_Click()  
    Unload Me  
End Sub
```

```
Private Sub lblVersion_Click()  
    Unload Me  
End Sub
```

```
Private Sub Picture1_Click()  
    Unload Me
```

End Sub

Private Sub Timer1\_Timer()

    Unload Me

End Sub

Option Explicit

Public startingaddress As String

Dim mbDontNavigateNow As Boolean

Private Sub Form\_Load()

    On Error Resume Next

    Me.Show

    tbToolBar.Refresh

    Form\_Resize

    cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15

    startingaddress = "http://www.neu.edu.tr/~wisam/c\_rsa.htm"

    If Len(startingaddress) > 0 Then

        cboAddress.Text = startingaddress

        cboAddress.AddItem cboAddress.Text

        'try to navigate to the starting address

        timTimer.Enabled = True

        brwWebBrowser.Navigate startingaddress

    End If

End Sub

Private Sub brwWebBrowser\_DownloadComplete()

    On Error Resume Next

    Me.Caption = brwWebBrowser.LocationName

End Sub

Private Sub brwWebBrowser\_NavigateComplete(ByVal URL As String)

    Dim i As Integer

    Dim bFound As Boolean

    Me.Caption = brwWebBrowser.LocationName

    For i = 0 To cboAddress.ListCount - 1

        If cboAddress.List(i) = brwWebBrowser.LocationURL Then

            bFound = True

            Exit For

        End If

    Next i

    mbDontNavigateNow = True

    If bFound Then

        cboAddress.RemoveItem i

    End If

    cboAddress.AddItem brwWebBrowser.LocationURL, 0



```
cboAddress.ListIndex = 0
mbDontNavigateNow = False
End Sub
```

```
Private Sub cboAddress_Click()
    If mbDontNavigateNow Then Exit Sub
    timTimer.Enabled = True
    brwWebBrowser.Navigate cboAddress.Text
End Sub
```

```
Private Sub cboAddress_KeyPress(KeyAscii As Integer)
    On Error Resume Next
    If KeyAscii = vbKeyReturn Then
        cboAddress_Click
    End If
End Sub
```

```
Private Sub Form_Resize()
    cboAddress.Width = Me.ScaleWidth - 100
    brwWebBrowser.Width = Me.ScaleWidth - 100
    brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top + picAddress.Height) -
100
End Sub
```

```
Private Sub timTimer_Timer()
    If brwWebBrowser.Busy = False Then
        timTimer.Enabled = False
        Me.Caption = brwWebBrowser.LocationName
    Else
        Me.Caption = "Working..."
    End If
End Sub
```

```
Private Sub tbToolBar_ButtonClick(ByVal Button As Button)
    On Error Resume Next

    timTimer.Enabled = True
```

```
    Select Case Button.Key
        Case "Back"
            brwWebBrowser.GoBack
        Case "Forward"
            brwWebBrowser.GoForward
        Case "Refresh"
            brwWebBrowser.Refresh
        Case "Home"
            brwWebBrowser.GoHome
```

```

Case "Search"
    brwWebBrowser.GoSearch
Case "Stop"
    timTimer.Enabled = False
    brwWebBrowser.Stop
    Me.Caption = brwWebBrowser.LocationName
End Select

End Sub

Function check_prime(ByVal val As Long) As Boolean
Dim primes
    primes = Array(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163,
167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257,
263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359,
367, 373, 379, 383, 389, 397)
    check_prime = False

    For i = 0 To 78
        If (val = primes(i)) Then
            prime = True
        End If
    Next i
    check_prime = prime
End Function

Function decrypt(ByVal c, ByVal n, ByVal d As Long)

Dim i, g, f As Long

On Error GoTo errorhandler

If (d Mod 2 = 0) Then
    g = 1
Else
    g = c
End If

    For i = 1 To d / 2

        f = c * c Mod n
        g = f * g Mod n
    Next i
    decrypt = g

Exit Function
errorhandler:
    Select Case Err.Number ' Evaluate error number.
        Case 6

```

```

        status.Text = "Calculation overflow, please select smaller values"
    Case Else
        status.Text = "Calculation error"
    End Select

```

```
End Function
```

```

Function getD(ByVal e As Long, ByVal PHI As Long) As Long
    Dim u(3) As Long
    Dim v(3) As Long
    Dim q, temp1, temp2, temp3 As Long

```

```
    u(0) = 1
```

```
    u(1) = 0
```

```
    u(2) = PHI
```

```
    v(0) = 0
```

```
    v(1) = 1
```

```
    v(2) = e
```

```
    While (v(2) <> 0)
```

```
        q = Int(u(2) / v(2))
```

```
        temp1 = u(0) - q * v(0)
```

```
        temp2 = u(1) - q * v(1)
```

```
        temp3 = u(2) - q * v(2)
```

```
        u(0) = v(0)
```

```
        u(1) = v(1)
```

```
        u(2) = v(2)
```

```
        v(0) = temp1
```

```
        v(1) = temp2
```

```
        v(2) = temp3
```

```
    Wend
```

```
    If (u(1) < 0) Then
```

```
        getD = (u(1) + PHI)
```

```
    Else
```

```
        getD = u(1)
```

```
    End If
```

```
End Function
```

```

Function getE(ByVal PHI As Long) As Long
    Dim great, e As Long

```

```
    great = 0
```

```
    e = 2
```

```
    While (great <> 1)
```

```
        e = e + 1
```

```
        great = get_common_denom(e, PHI)
```



```

Wend
getE = e
End Function

```

```

Function get_common_denom(ByVal e As Long, ByVal PHI As Long)
Dim great, temp, a As Long

```

```

    If (e > PHI) Then
        While (e Mod PHI <> 0)
            temp = e Mod PHI
            e = PHI
            PHI = temp
        Wend
        great = PHI
    Else
        While (PHI Mod e <> 0)
            a = PHI Mod e
            PHI = e
            e = a
        Wend
        great = e
    End If
    get_common_denom = great
End Function

```

```

Private Sub show_primes()
    status.Text = "1"
    no_primes = 1
    For i = 2 To 400
        prime = True
        For j = 2 To (i / 2)
            If ((i Mod j) = 0) Then
                prime = False
            End If
        Next j

        If (prime = True) Then
            no_primes = no_primes + 1
            status.Text = status.Text + ", " + Str(i)
        End If
    Next i
    status.Text = status.Text + vbCrLf + "Number of primes found:" + Str(no_primes)
End Sub

```

```

Private Sub Command1_Click()
Dim p, q, n, e, PHI, d, m, c As Long

```

```

p = Text1.Text
q = Text2.Text
If (check_prime(p) = False) Then
    status.Text = "p is not a prime or is too large, please re-enter"
ElseIf (check_prime(q) = False) Then
    status.Text = "q is not a prime or is too large, please re-enter"
Else
    n = p * q
    Text3.Text = n

    PHI = (p - 1) * (q - 1)
    e = getE((PHI))
    d = getD((e), (PHI))
    Text4.Text = PHI
    Text5.Text = d
    Text6.Text = e
    m = Text7.Text

    c = (m ^ e) Mod n
    Text8.Text = c
    m = decrypt(c, n, d)
    Text9.Text = m
    Label12.Caption = "Decrypt key =<" + Str(d) + "," + Str(n) + ">"
    Label13.Caption = "Encrypt key =<" + Str(e) + "," + Str(n) + ">"
End If
End Sub

Private Sub Command2_Click()
    End
End Sub

Private Sub Command3_Click()
    frmBrowser.Show
End Sub

Private Sub Command4_Click()
    Call show_primes
End Sub

Private Sub Form_Load()

End Sub

```