**1.Introduction**

In today’s world, mobile devices are becoming more and more popular. As these devices have begun to spread, the demand for more and better functionality has come with them. However, more functionality leads to more complexity of the operating systems in various mobile devices. However, when involving in an operating system, the mobile devices are much more vulnerable to bugs, crashes, and security holes. When a system adapts to different functions, these functions might mess up with each other unexpectedly and cause it work strangely or improperly. With the plain fact that mobile devices are completely integrated into almost every aspect of our live, they leave a question, “is security an issue?” This question was answered by the first virus for a mobile computer, the cabir worm. Viruses, worms, and other malwares are always concerns since they can steal information and render devices useless. Since the mobile devices always

access to the websites, wirelessly connected to different devices, many severe security issues have been raised. To tackle the security issues, we have to understand different concepts of security. As defined by [1], malware is software designed to infiltrate a computer system without the owner's informed consent.

The expression is a general term used by computer professionals to mean various forms of hostile, intrusive, or annoying software or program codes. When applying this term to mobile devices, it is in essence the same thing, but is even harder to tackle the serious problems caused by it. There are many different operating systems, and even more diverse functionality of each one, it is hard to have a powerful antivirus software that will run on all of the different operating systems and kill all kinds of viruses. It has been thought by the companies that the complexity a virus has to achieve makes it difficult to create a big number of viruses.

This misleading security ignorance creates fundamental security risks for the software systems. Just like people said “If we don’t know a back door exist means we will not look for it”. This idea is the foundation of many the problems in mobile security.

**2.History of mobile malware**

As mentioned in [3], Cabir, a computer worm developed in 2004 is designed to infect mobile phones running Symbian OS [2], which is an operating system designed for mobile devices and smartphones. It is believed to be the first worm that infected mobile phones. When a phone is infected by Cabir, the message "Caribe" is shown on the phone's display, and is appeared every time when the phone is turned on. The worm then attempts to spread out to other phones in the area using Bluetooth technology. The worm was not sent

out into the wild, but sent directly to anti-virus firms, who believed Cabir in its current state is harmless. However, it does prove that mobile phones are also vulnerable to the viruses. Experts also believe that the worm was developed by a group who call themselves 29A, a group of international hackers. They created a "proof of concept" worm in order to catch world’s attention. The worm can attack and replicate on Bluetooth enabled Series 60 phones. It tried to send itself to all Bluetooth enabled devices that support the "Object Push Profile". It can also infect non-Symbian phones, desktop computers and even printers. Cabir does not spread if the user does not accept the file-transfer or does not agree with the installation. Some older phones would keep on displaying popups. Cabir persistently re-sends itself and renders the User Interface until “yes” is clicked. Even though the Cabir virus is credited as the first mobile device virus, it was only regarded as a concept virus. All the virus did was to show that a virus could be created based on the Symbian operating system. The codes were written to spur the development of operating system’s creator, so that the security level of the operating system can be improved. However the source codes were leaked into the internet and modified, which made the virus more malicious than originally intended. About a month after the cabir worm struck, the next

mobile virus, called “Duts” appeared. Duts was the first virus for the windows CE platform, and the first file infector for mobile devices. The duts virus would infect the executables in the root directory of the device if user permitted. Soon after duts, the brador virus came out. The Brador virus was the first backdoor virus for mobile devices. Backdoor is an open port that waits for a remote host to connect to it.

The viruses get into the system through the backdoor without being discovered.

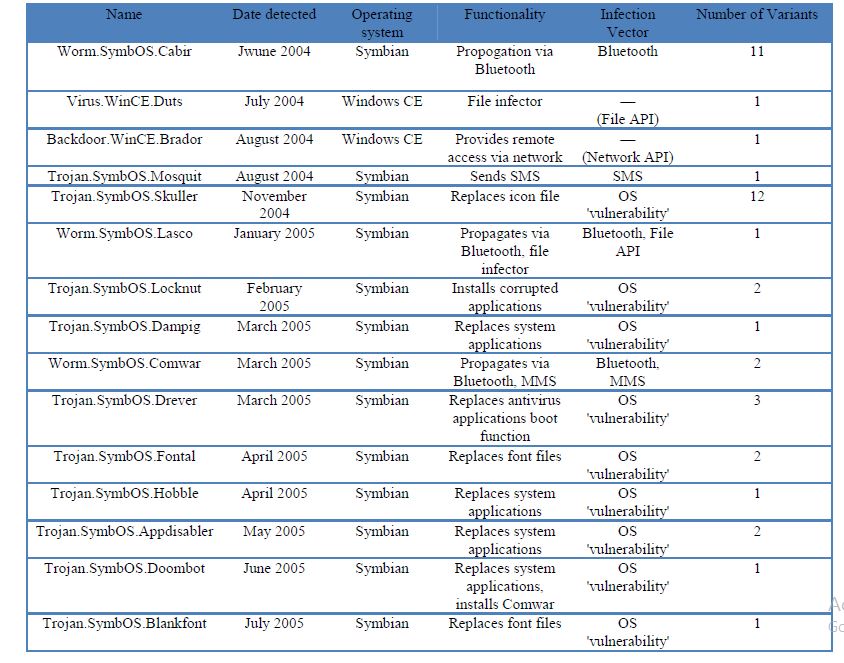


Fig1. The Summary of malware causing security attack

**3.Vulnerabilities and threats of mobile devices**

Mobile devices security is a relatively new technology because there is still not a large focus on it. Sadly enough, the only way that the security is going to develop is by the appearance of a large amount of mobile devices malwares which need to be dealt with immediately without further avoidance. This is not to say that the current devices do not have any form of security, sometimes users are uneducated and render these measures ineffective [9]. Until people are properly taught what to do or what not to do, they will be more aware of security issues. Certain things like Bluetooth or Wi-Fi often time enabled by default on new mobile devices which are huge security risks. There are simple solutions for these problems; installing the newest firmware on devices, turning Bluetooth off when not in use, not connecting to unsecured wireless networks, not opening strange emails, and not running programs that you don’t know what they do. These are the simple precautions people can take that will eliminate the great majority of the mobile device vulnerabilities. This should be regarded as an extreme concern because of the nature of mobile devices. Often time triggered viruses are designed to make money off the ads or the other schemes. It is almost impossible to completely avoid the time triggered viruses if they are put onto a mobile device.

This makes the mobile devices very attractive targets to the hackers. Most threats to mobile devices are in the form of worms, “a self-replicating virus”. This is the biggest issue since mobile devices are designed to communicate with other devices. For this reason, the virus on the compromised mobile

device spreads out, is now in leads to a possibly very devastating virus.

**4. Sources of Threats and Attack Methods**

Mobile devices face a range of cyber security threats. These threats can be unintentional or intentional. Unintentional threats can be caused by software upgrades or defective equipment that inadvertently disrupt systems. Intentional threats include both targeted and untargeted attacks from a variety of sources, including botnet operators, cyber criminals, hackers, foreign nations engaged in espionage, and terrorists. These threat sources vary in terms of the capabilities of the actors, their willingness to act, and their motives, which can include monetary gain or political advantage, among others. For example, cyber criminals are using various attack methods to access sensitive information stored and transmitted by mobile devices.

Botnet operators uses malware distributed to large number of mobile devices and other electronic system to coordinate remotely controlled attack on websites and to distribute phishing schemes ,scams and future malware attacks on individual mobile devices.

Cyber criminals generally attack mobile devices for monetary gain They make use of phishing, spyware, malware to gain access to the information stored on the device which they then commit to make identity theft online fraud. In addition International Criminal organizations pose a threat to corporations, Govt agencies and other institutions.

Foreign intelligent Services may attack mobile devices as a part of their information gathering and espionage operations. Foreign Governments develop information warfare doctrines, Programs and Capabilities that could disrupt the supply chains and mobile communications.

Hackers may attack mobile devices to demonstrate their skills or to gain prestige in the hacker community. While hacking one required a fair amount of skills or computer knowledge, attackers download attack scripts and protocols to carry actions.

These threat sources may use a variety of techniques, or exploits, to gain control of mobile devices or to access sensitive information on them. Common mobile attacks are

as follows

1.Browser Exploits

2.Data Interception

3.Keystroke Logging

4.Malware

5.Unauthorized location tracking

6.Network Exploits

7.Phishing

8.Spamming

Attacks against mobile devices generally occur through four different channels of activities:

• **Software downloads.** Malicious applications may be disguised as a game, device patch, or utility, which is available for download by unsuspecting users and provides the means for unauthorized users to gain unauthorized use of mobile devices and access to private information or system resources on mobile devices.

• **Visiting a malicious website.** Malicious websites may automatically download malware to a mobile device when a user visits. In some cases, the user must take action (such as clicking on a hyperlink) to download the application, while in other cases the application may download automatically.

• **Direct attack through the communication network.** Rather than targeting the mobile device itself, some attacks try to intercept communications to and from the device in order to gain unauthorized use of mobile devices and access to sensitive information.

• **Physical attacks.** Unauthorized individuals may gain possession of lost or stolen devices and have unauthorized use of mobile devices and access sensitive information stored on the device.

**5. A Range of Vulnerabilities Facilitate Attacks**

Mobile devices are subject to numerous security vulnerabilities, including a failure to enable password protection, the inability to intercept malware, and operating systems that are not kept up to date with the latest security patches. While not a comprehensive list of all possible vulnerabilities, the following 10 vulnerabilities can be found on all mobile platforms.

• **Mobile devices often do not have passwords enabled.**

Mobile devices often lack passwords to authenticate users and control access to data stored on the devices. Many devices have the technical capability to support passwords, personal identification numbers (PIN), or pattern screen locks for authentication. Some mobile devices also include a biometric reader to scan a fingerprint for authentication. However, anecdotal information indicates that consumers seldom employ these mechanisms. Additionally, if users do use a password or PIN they often choose passwords or PINs that can be easily determined or bypassed, such as 1234 or 0000. Without passwords or PINs to lock the device, there is increased risk that stolen or lost phones’ information could be accessed by unauthorized users who could view sensitive information and misuse mobile devices.

• **Two-factor authentication is not always used when conducting sensitive transactions on mobile devices.**

According to studies, consumers generally use static passwords instead of two-factor authentication when conducting online sensitive transactions while using mobile devices. Using static passwords for authentication has security drawbacks: passwords can be guessed, forgotten, written down and stolen, or eavesdropped. Two-factor authentication generally provides a higher level of security than traditional passwords and PINs, and this higher level may be important for sensitive transactions. Two-factor refers to an authentication system in which users are required to authenticate using at least two different “factors”—something you know, something you have, or something you are—before being granted access. Mobile devices themselves can be used as a second factor in some two-factor authentication schemes. The mobile device can generate pass codes, or the codes can be sent via a text message to the phone. Without two-factor authentication, increased risk exists that unauthorized users could gain access to sensitive information and misuse mobile devices.

• **Wireless transmissions are not always encrypted.**

Information such as e-mails sent by a mobile device is usually not encrypted while in transit. In addition, many applications do not encrypt the data they transmit and receive over the network, making it easy for the data to be intercepted. For example, if an application is transmitting data over an unencrypted WiFi network using hypertext transfer protocol (http) (rather than secure http),27the data can be easily intercepted. When a wireless transmission is not encrypted, data can be easily intercepted by eavesdroppers, who may gain unauthorized access to sensitive information.

• **Mobile devices may contain malware.**

Consumers may download applications that contain malware. Consumers download malware unknowingly because it can be disguised as a game, security patch, utility, or other useful application. It is difficult for users to tell the difference between a legitimate application and one containing malware. For example, figure 3 shows how an application could be repackaged with malware and a consumer could inadvertently download it onto a mobile device.

• **Mobile devices often do not use security software.**

Many mobile devices do not come preinstalled with security software to protect against malicious applications, spyware, and malware-based attacks. Further, users do not always install security software, in part because mobile devices often do not come preloaded with such software. While such software may slow operations and affect battery life on some mobile devices, without it, the risk may be increased that an attacker could successfully distribute malware such as viruses, Trojans, spyware, and spam, to lure users into revealing passwords or other confidential information.

• **Operating systems may be out-of-date.**

Security patches or fixes for mobile devices’ operating systems are not always installed on mobile devices in a timely manner. It can take weeks to months before security updates are provided to consumers’ devices. Depending on the nature of the vulnerability, the patching process may be complex and involve many parties. For example, Google develops updates to fix security vulnerabilities in the Android OS, but it is up to device manufacturers to produce a device-specific incorporating the vulnerability fix, which can take time if there are proprietary modifications to the device’s software. Once a manufacturer produces an update, it is up to each carrier to test it and transmit the updates to consumers’ devices. However, carriers can be delayed in providing the updates because they need time to test whether they interfere with other aspects of the device or the software installed on it.

In addition, mobile devices that are older than 2 years may not receive security updates because manufacturers may no longer support these devices. Many manufacturers stop supporting smartphones as soon as 12 to 18 months after their release. Such devices may face increased risk if manufacturers do not develop patches for newly discovered vulnerabilities.

• **Software on mobile devices may be out-of-date.**

Security patches for third-party applications are not always developed and released in a timely manner. In addition, mobile third-party applications, including web browsers, do not always notify consumers when updates are available. Unlike traditional web browsers, mobile browsers rarely get updates. Using outdated software increases the risk that an attacker may exploit vulnerabilities associated with these devices.

• **Mobile devices often do not limit Internet connections.**

Many mobile devices do not have firewalls to limit connections. When the device is connected to a wide area network it uses communications ports to connect with other devices and the Internet. These ports are similar to doorways to the device. A hacker could access the mobile device through a port that is not secured. A firewall secures these ports and allows the user to choose what connections he or she wants to allow into the mobile device. The firewall intercepts both incoming and outgoing connection attempts and blocks or permits them based on a list of rules. Without a firewall, the mobile device may be open to intrusion through an unsecured communications port, and an intruder may be able to obtain sensitive information on the device and misuse it.

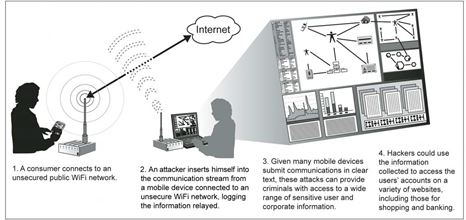


Fig2:Man-in-the-Middle Attack Using an Unsecured WiFi Network

**6.Security Controls for Mobile Devices**

Table 3 outlines security controls that can be enabled on mobile devices to help protect against common security threats and vulnerabilities. The security controls and practices described are not a comprehensive list, but are consistent with recent studies28and guidance from NIST and DHS.

Key Security control to combat common threats and Vulnerabilities

1.Enable User authentication

2.Enable two factor authentication for Sensitive transactions

3.Verify the authenticity of downloaded applications

4.Install antimalware Capability

5.Install a firewall

6.Receive Prompt Security Updates

Additonal Seurity controls Specific to organisations to combat common threats and Vulnerabilities.

1.Adopt Centralized Security management

2.Use mobile device integrity Validation

3.Implement a Virtual Private network

4.Use Public key infrastructure Support

5.Require Conformance to Government Specifications

In addition to using mobile devices with security controls enabled, consumers can also adopt recommended security practices to mitigate threats and vulnerabilities. Table 5 outlines security practices consumers can adopt to protect the information on their devices. The practices are consistent with guidance from NIST and DHS.

Key Security practices to combat common threats and Vulnerabilities

1.Turnoff or set Bluetooth connection capabilities to non discoverable.

2.Limit use of Public Wifi networks when conducting Sensitive transactions.

3.Minimize installation of unnecessary applications.

4.Configure Web accounts to use secure connections

5.Donot follow links set in suspicious emails or web accounts.

6.Establish a mobile device security Policy.

7.Provide mobile device security training.

8.Establish a deployment plan.

9.Perform risk assessment.

10.Perform configuration control and management.

**7.Other Attacks**

1.Bluetooth attack

2.Mobile Denial of Service attack

3.Wifi Security attack i.e Penetration attack

**Bluetooth security modes**

Every Bluetooth device has three major security modes in which it can operate on. The first mode is known as non secure security mode. In this mode, the features such as authentication, encryption, and pairing are not enforced. The second mode is known as the service-level security mode. In this mode, a central security manager restricts access to the device by performing authentication. The last mode is called the link-level security mode. In this mode, authorization and security procedures are enforced and implemented before an establishment of a communication channel. This mode typically involves in using the previously described processes of pairing and binding. Overall, Bluetooth has transformed wireless communication as it is widely implemented and supported. Unfortunately, like many protocols, it suffers from security threats and vulnerabilities.

**Bluetooth attacks**

One of the least serious and harmless Bluetooth attacks is called BlueJacking. This attack takes advantage of a small loophole in the messaging protocol and allows a Bluetooth device to send an anonymous message to a target Bluetooth device. When two Bluetooth devices wish to communicate with each other they must first perform an initial handshake process in which the initiating Bluetooth device must display its name on the target Bluetooth device. Instead, an attacker can send a user-defined field to the target device. BlueJacking takes advantage of this field in order to send the anonymous message.

A much more dangerous case, and one of the best known Bluetooth attacks, is BlueSnarfing. BlueSnarfing is the process in which the attacker connects to the victim’s mobile phone through Bluetooth without the victim’s attention. This attack is dangerous because the attacker can gain access to private information such as the address book, messages, personal photographs, etc. Furthermore, the attacker can initiate as well as forward phone calls. The attacker can complete this BlueSnarfing easily within 10 meters of the victim by using software tools such as Blooover, Redsnarf, and BlueSnarf.

**Countermeasures**

Even though mobile phones face security threats from Bluetooth attacks, there are still effective countermeasures that can be used for protection. The simplest action can be taken is to disable Bluetooth completely on the mobile phone. Alternately, the mobile phone’s Bluetooth settings can be switched to an undiscoverable or hidden mode. It is important to be aware of Bluetooth attacks and take countermeasures, as Bluetooth attacks are one of the primary ways mobile phone data is compromised.

**Mobile denial-of-service**

Compared with Bluetooth attack, Mobile Denial-of- Service (MDoS) attacks can be the worst attacks on a mobile phone. One of the major ways the attack is completed is through a Bluetooth enabled device. An MDoS attack can render a mobile phone useless. MDoS attacks can congest available bandwidth causing all data transfers stop, leading the phone to freeze, crash, or even restart. While there are different types of MDoS attacks, they all usually follow a similar pattern on how the attack is implemented.

The attacker first uses some sort of packet-generation software in order to create infinite and sometimes malicious packets. These packets can then be sent to the victim’s mobile phone using a specified protocol. One reason these attacks are considered dangerous is that they are easy to be executed. MDoS ready-to-go tools can easily be found on the Internet and downloaded. These attacks are possible if there is a loophole found in Bluetooth communication. Bluetooth technology does not have a way to handle incoming packets and therefore does not inspect them at all. Compared with a normal mobile phone user, the problem seems to be more serious to a business mobile phone user, since he or she who depends on the phone for work can be devastated during an MDoS attack. The attack could limit their ability to access important data, significantly slow down their connection speed, and could even cause entire disconnection. Mobile phone users need to be aware that MDoS attacks can and do happen.

**Mobile denial-of-service attacks**

BlueSmacking is a common type of MDoS attack. The basic idea behind the attack is to send oversized data packets to the mobile device. Mobile devices using Bluetooth have a size limit on the packets that they can receive. This size difference depends on the manufacturer and model of the phone. This means that the devices cannot handle packets that are greater than the size limit. The attacker takes advantage of this weakness and sends oversized data packets to the target device. The device will not be able to handle numerous, constant, oversized packets thus resulting in a denial-of service The second MDoS attack, although not very popular, is called Jamming. As described earlier, Bluetooth works in the 2.4GHz frequency range and it handles interferences by frequency hopping. In a Jamming attack, the entire frequency band has to be jammed so that the Bluetooth device has noavailable frequency to use. The amount of work the attacker has to put in for a Jamming attack is not feasible resulting in the attack’s unpopularity.

The third common MDoS attack is called a failed authentication attack. This attack prevents two Bluetooth devices from establishing a connection with each other. In order for the attacker to be successful, the hacker must flood the target device with spoofed packets while the target device is trying to connect with a desired device. In doing so, the target device’s resource becomes congested and the target device is unable to make the connection with the desired device.

**Countermeasures**

Mobile phone users should be aware of MDoS attacks and also realize that there are countermeasures that are available in order to protect themselves from these attacks. One of the simplest things a user can do is to keep their phone up to date by downloading and installing the latest patches and upgrading their mobile phone. Another countermeasure is simply not to accept an unknown incoming message via Bluetooth. Users should only pair their mobile phone with known devices.

**8.End-to-end security implementation for mobile devices using TLS protocol**

End-to-end security has been an emerging need for mobile devices with the widespread use of personal digital assistants and mobile phones. Transport Layer Security Protocol (TLS) is an end-to-end security protocol that is commonly used on the Internet, together with its predecessor, SSL protocol. By implementing TLS protocol in the mobile world, the advantage of the proven security model of this protocol can be utilized. The main design goals of mobile end-to-end security protocol are maintainability and extensibility.

Current trends in mobile applications have been the enterprise-style applications that needed high-capacity, network-connected devices. Financial applications like banking and stock trading are common examples of these kinds of applications. However, some deficiencies prevent its acceptance in e-commerce applications [12]. With more and more network connected applications, security has became one of the most popular concept in mobile community. Mobile security deals mainly with two issues: security of the physical device and its contents and security of the data in network communication. The communication of data in mobile devices is provided by the mobile networks. Mobile networks are open to many kinds of attacks. The open data communication in these networks may cause attacks against the secrecy, integrity and authenticity of data. Many vendors have proposed solutions against these security vulnerabilities. Most of these solutions are vendor-specific proprietary products or libraries.

The alternative to proxy based security solutions is end-to-end security. End-to-end security refers to the securely encoding of data at the source host and decoding at the destination host. The data will not travel unencoded at any part of the communication. Transport Layer Security Protocol (TLS) and its predecessor Secure Sockets Layer (SSL) protocols are end-to-end security solutions that are commonly used in the wired world [1]. There are a number of reasons to use these protocols in the mobile world: – TLS and SSL protocols have been used for many years, so their security is tested by the community and accepted as secure enough to be used by financial data communication. – TLS and SSL protocols are common in wired world and may be accepted as the security protocol of Internet. Using these in mobile applications will make the integration between mobile applications and Internet easier. – TLS and SSL have open specifications and many implementations. It may be relatively easy to implement these for specific needs. Both protocols are known by their resource consuming nature and the idea that they are not suitable for mobile networks. The end-to-end security solution presented in this paper is implemented for mobile devices using Java platform in order to show that it is possible and efficient to use TLS in mobile networks. Kwon et al. [8] worked on the same problem and published an end-toend security model that utilizes transport layer security to overcome WAP security problems. However, implementation of their model and related experiments with resulting values cannot be found.

Application architecture Client/server architecture is the main architecture for mobile network applications. In this architecture, a network-aware application residing on a wireless device, client, connects with back-end applications and servers behind a firewall or proxy gateway over a wireless network and Internet and corresponding communication protocols. Figure 2 shows the client/server architecture of a mobile application. In Figure 2, the wireless devices can be mobile phones, PDAs or two-way pagers. They have the mobile applications on top of MIDP and communicate with an antenna over a wireless communication protocol. These protocols can be Wireless-LAN on a small area or GPRS on a wide area. The antenna is directly connected to a wired network like Internet, which connects the mobile device to the back-end server systems. Mobile applications typically use HTTP as the application-level communication protocol as it is common and can pass firewalls. HTTP is the mandatory protocol in MIDP 1.0 [15] and MIDP 2.0 specifications.

TLS may provide an application-level end-to-end security on top of sockets in this architecture. The TLS implementation developed in this thesis runs on top of pure sockets and provides an application level security between a MIDP application and server back-end application. The MIDP version of the developed TLS protocol communicates with the J2SE version of the protocol and sends object data over the secure connection. The mobile device always runs the client version of TLS protocol and the back-end server always runs the server version of the TLS protocol. Peer To Peer Architecture is an alternative and new architecture for wireless devices. In this architecture, the two devices communicate directly with each other. This communication may be the direct communication of the devices over a communication medium like Bluetooth or it may be an application level communication with the two mobile applications communicating over a central server.

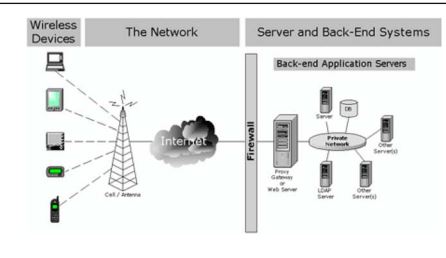


Fig.3 Environment of a typical networked wireless

**Design of mobile devices end-to-end security architecture**

The security protocol presented in this paper is an end to-end security protocol based on TLS 1.0 specifications and adopted to work on J2ME mobile devices as well as standard Java VMs. The protocol implementation itself is also an application, although high-level applications may use it through its APIs to transmit data and objects securely. The security protocol architecture developed covers both TLS protocol architecture and the necessary APIs architecture. TLS protocol architecture is based on TLS 1.0 specification and has some additions and subtractions. The necessary APIs architecture involves cryptography model classes that abstract the real implementations of cryptography packages; socket classes that abstract TCP or UDP based socket implementations and the serialization of object data; XML Serializer architecture that is a standalone API incorporated into the protocol implementation. The main design issues taken into consideration during the definition of the architecture of protocol implementation are J2ME compatibility, mobile adaptation, secure object transmission, full abstraction and complete solution. The main architecture of the mobile end-to-end security protocol is based on the TLS 1.0 protocol specification [3]. TLS is a protocol that consists of several layers of protocols as shown in Figure 3. At the bottom of these layers, there is the TLS Record Protocol. The Record Protocol is responsible for taking messages to be transmitted, fragmenting into blocks, optionally compressing, applying MAC to protect data integrity, encrypting the block and transmitting the result to higher level clients. Received data is then decrypted; applied MAC is verified to protect data integrity; optionally decompressed; reassembled and then delivered to higher level clients by the Record Protocol. TLS Record Protocol.

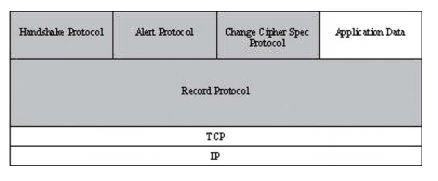


Fig.4 The layered structure of TLS protocol

**Implementation of mobile devices end-to-end security architecture**

The mobile end-to-end security protocol is implemented in Java to be compliant to both J2ME and J2SE platforms. The development environment for mobile endto-end security protocol is JBuilder 8.0 Enterprise MobileSet 3 Integrated Development Environment. The development environment was chosen because it has an integrated MIDP application development support. A project may be compiled with MIDP libraries by changing the project JDK to J2MEWTK from the project properties window. JBuilder IDE was also chosen for its high performance, code completion feature and userfriendly interface. JBuilder IDE uses Sun Microsystem’s J2ME Wireless Toolkit for MIDP development. J2ME Wireless Toolkit (J2MEWTK) is a toolkit that can be used separately or plugged into a development IDE. It provides development tools and emulators for MIDP application development. The 2.1 version of J2MEWTK is used in the development and in the experiements of the developed protocol for mobile environments. J2MEWTK 2.1 supports MIDP 2.0, CLDC 1.1, optionalWireless Messaging API (JSR 120), Mobile Media API (JSR 135) and Web Services Access for J2ME API (JSR 172). J2ME Wireless Toolkit has the tool Ktoolbar that manages MIDP application development. Ktoolbar has menu options for creating a new MIDP application, opening an existing one, compiling and running the application with the set up MIDP version, changing the J2MEWTK and application preferences. J2MEWTK uses device emulators to runMIDP applications. Device emulators are software implementations of mobile devices to test applications before the real deployment. Device emulators provide a faster and easier development for mobile applications. Most mobile phones and PALM devices have their emulator software. Mobile phone emulators are distributed by telephone vendors freely. They have the same operating system and visual interface with the original telephone. PALM emulators emulate PALM OS. A ROM file is taken from a real PALM device with Hotsync connection and loaded into the emulator software. This gives all the features of the PALM device to the emulator software including program installation and uninstallation. Some examples are also run on original PALM M100 PDAs and Nokia 6600 mobile phones. Java has been divided into three platforms: J2EE for server side development, J2SE for standard desktop applications and J2ME for small, embedded devices. J2ME has its own configurations and profiles. Each con- figuration and profile has its own library APIs.

**Mobile device experiment results**

Mobile device experiments were performed according to the configuration explained above. There was no problem in establishing communication with GPRS and all experiments succeeded. Figures 6 and 7 (see Appendix) show the secure communication time span durations when the experimental client program runs on a Nokia 6600 mobile phone, the server program runs on J2SE platform and transport protocol between is TCP. All values presented in the tables are milliseconds (ms). “0” value in a table means that the time duration is below milliseconds.

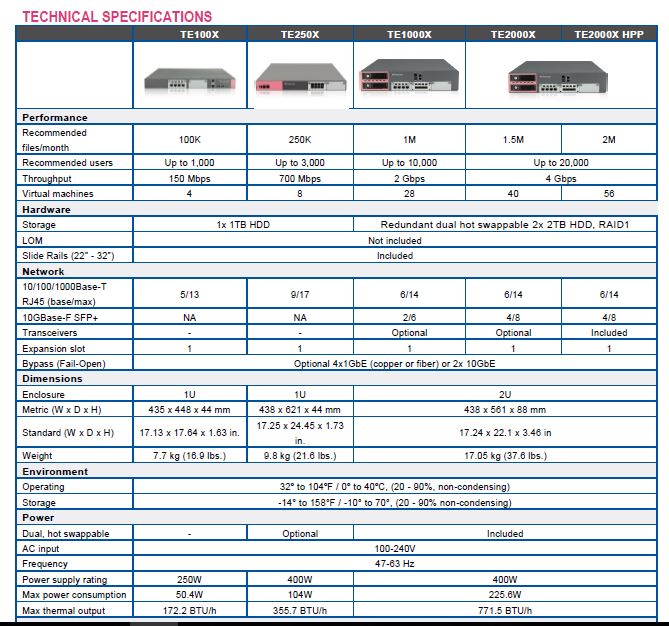
**9.Future development**

The largest problem with mobile security is there is no enough time dedicated to it when designing a mobile device. For the most part, the malware can only access if the user does something to make the system vulnerable in some way or fashion. Be it running a program that has the malware hidden in it, or cracking the system so that the built in security is removed. Many experts argue that the only thing that will make users more aware is a large amount of malware forcing people to become educated or else leave them unable to use their devices. The reason for this is because in the early 2000 there were a large number of viruses that completely debilitated networks. This in turn made people understand the importance of antivirus and their threats that they don’t recognize. Since then, people have been much more careful with their computers. Due to this positive response, many people think this is the only way to make people pay attention to mobile devices security. In example, there have been many proofs of concept viruses that target phones just to show it can be done and explain it could have been even worse; however this generally is circulated through the technical world and never reaches the end users on a large scale.

**10.Conclusion**

End-to-end security is an emerging need for mobile devices. Banking, military and other enterprise applications need more and more security to run on mobile devices. This project aimed at developing an extensible end-to-end security protocol implementation that could be used by both mobile and desktop applications. TLS protocol that is commonly used on the Internet was chosen as the base of the developed protocol implementation. The proposed protocol was designed and implemented. The implementation was run with different experiments and the time span of operations were measured. The protocol implementation run on a real Nokia 6600 mobile phone and established a secure connection with a server computer connected to the Internet over GPRS. All the experiments were successfully completed, which indicated that the protocol was properly designed and implemented with respect to specifications. The implementation guarantees the security of any transmission at most as much as TLS. The implementation did not include optional TLS specifications like client authentication, session resumption and compression.

**Sandbox Specifications**

****

**11.References**

1. Agarwal, A.K.,Gill, J.S.,Wang,W.:An experimental study on wireless security protocols over mobile IP networks. In: IEEE Proceedings of 60th Vehicular Technology Conference (2004)

2. BouncyCastle: Bouncy Castle Documentation. (2006)

http://www.bouncycastle.org/documentation.html

3. Dierks, T., Allen, C.: The TLS protocol version 1.0, IETF RFC

2246 (1999)

4. Reynaud-Plantey, D.: J2ME low level security: implementation

versus specification. (2005) http://prdownloads.source

forge.net/tinapoc/Reynaud\_J2ME.pdf?download.

5. Intel IEEE 802.11b high rate wireless local area networks,

Intel Corporation (2000)

6. Jøsang, A., Sanderud, G.: Security in mobile communications:

challenges and opportunities. In: Australasian Information

Security Workshop (AISW2003), Australia (2003)

7. Knudsen, J.: Introduction toWireless JavaTechnologiesWhite

Paper, Sun Microsystems Inc. (2001)

8. Kwon, E., Cho, Y., Chae, K.: Integrated transport layer security:

end-to-end security model between WTLS and TLS. In:

IEEE Proceedings of 15th International Conference on Information

Networking, pp. 65–71 (2001)

9. Mahmoud, Q.H.: Wireless Application Programming with

J2ME and Bluetooth. (2003) http://developers.sun.com/tech.