Lappeenranta University of Technology

Faculty of Technology Management

Department of Information Technology

Authentication and authorization in PeerHood System

The subject of this Master's thesis was approved by the department council of the Department of Information Technology on _____.

Supervisors: Professor Jari Porras

M.Sc. Arto Hämäläinen

Helsinki, 30th of October, 2007

_____ / _____

Yevgeniy Bondarenko

Lounaisväylä 2 B 7, 00200, Helsinki

Phone: +358 509233091

ABSTRACT

Lappeenranta University of Technology

Department of Information Technology

Yevgeniy Bondarenko

Authentication and authorization in PeerHood System

Thesis for the Degree of Master of Science in Information Technology

2007

46 pages, 7 figures, 3 tables, 9 listings

Examiner: Professor Porras Jari,

M.Sc. Hämäläinen Arto

Keywords: mobile technology, authentication, authorization, peer-to-peer communication, mobile Ad-Hoc network, PeerHood

The goal of this work is to design and implement authentication and authorization section to PeerHood system. PeerHood system is developed in Lappeenranta University of Technology. It provides functions of discovering devices compatible with PeerHood and listing services offered by those devices; based on wireless technologies: Wi-Fi, Bluetooth and GPRS. The thesis describes implementation of the security approach into mobile Ad-Hoc environment and includes both authentication and authorization processes.

Table of contents

List of Abbreviations

3G is the **third generation of mobile phone standards** [1] is based on the International Telecommunication Union (ITU) family of standards. It increases efficiency of using wavelengths range, allows providing more services for users and higher capacity. 3G systems use radio wider-bandwidth channels and use Wideband Code Division Multiple Access (WCDMA).

Adaptive Frequency-hopping spread spectrum (AFH).

The **Advanced Research Projects Agency Network** (**ARPANET**) [2] was a first large wide-area network (created in 1969). It was formed by the United States Defense Advanced Research Project Agency. ARPANET was a training ground for new networking technologies, connecting many universities and research centers.

DataBase (DB).

Deoxyribonucleic acid (DNA).

Denial-of-service attack (DoS attack).

Enhanced Data Rate (EDR).

Frequency-hopping spread spectrum (**FHSS**) is a technique of transmitting radio signals by fast switching a carrier between many frequency channels, using a pseudorandom sequence acknowledged by both transmitter and receiver sides.

General Packet Radio Service (**GPRS**) is a Mobile Data Service utilized in IS-136 and GSM mobile phones. GPRS can be used for Internet communication services.

Host Controller Interface (HCI) is an essential part of Bluetooth subsystem. HCI is the interface which links a Bluetooth host to a Bluetoth controller.

Hotspots are basic devices that provide Wi-Fi access. One can use a WiFi mobile phone, laptop, or other suitable portable device to access sources provided by the wireless network.

Hypertext Transfer Protocol (**HTTP**) is a protocol used to convey or transfer information on the World Wide Web.

Hypertext Transfer Protocol over Secure Socket Layer (**HTTPS** or HTTP over SSL) is a Uniform Resource Identifier scheme used to set up a secure HTTP connection.

Information technology (**IT**) is "the study, design, development, implementation, support or management of computer-based information systems, particularly software applications and computer hardware."

The **Infrared Data Association** (**IrDA**) is a protocol for the short-range transfer of data over infrared light.

The **Institute of Electrical and Electronics Engineers** (**IEEE**) is an international organization for the development of technology related to electricity.

The **Internet Engineering Task Force** (**IETF**) promotes and develops Internet standards. It deals with standards of the Internet protocol suite and TCP/IP.

Internet Research Task Force (**IRTF**) promotes research of significance to the improvement of the future Internet by creating small, focused and long-term Research Groups operation on topics related to Internet technology, protocols, architecture and applications.

Mobile Ad-hoc network (MANET).

Orthogonal Frequency-Division Multiplexing (OFDM).

A **peer-to-peer** (**P2P**) computer network uses mainly resources (computing power and bandwidth) of its participants rather than concentrating it in servers and used for sharing content files.

Personal Area Network (PAN).

Personal digital assistant (PDA).

Personal identification number (PIN).

Point-to-point telecommunications is a type of connection restricted to two endpoints, typically host computers.

Received Signal Strength Indication (**RSSI**) is a measurement of the received radio signal strength (not the quality). RSSI is common radio receiver technology metric (IEEE 802.11 protocol family)

Secure And Fast Encryption Routine (SAFER).

Secure Sockets Layer (SSL).

The **Transmission Control Protocol** (**TCP**) is one of the core protocols of the Internet protocol suite. TCP/IP protocol provides reliable and in-order delivery of data.

Transport Layer Security (**TLS**) is cryptographic protocol that provides secure communications on the Internet.

A **Universal Asynchronous Receiver/Transmitter** (**UART**) is a piece of computer hardware that transforms data between serial and parallel interfaces. It converts bytes of data to/from asynchronous start-stop bit streams during transformation.

User Datagram Protocol (UDP).

1. INTRODUCTION

Mobile devices[1] have become a very popular nowadays. This is a growing area of modern technologies. Telecommunication systems such as Bluetooth, Wi-Fi and GPRS are very common standards used in mobile devices. However, utilization of these technologies is still incomplete. Plenty of mobile devices are still disconnected. Number of software products [3], [4] provides communication network in the mobile world. Nevertheless, these solutions are very limited and do not cover all problems of network construction. One of the challenges for these applications is a security problem. Because the radio interface of wireless networks is accessible to everyone, the security is more difficult to implement, as attackers can more easily interact. [5]

The master thesis describes a possible solution for this problem. The implementation of the security is done for PeerHood system developed by Lappeenranta University of Technology. That is the example of application for creating a mobile ad-hoc network. The security approach for such networks is very important and yet opens issue. Authentication and authorization is only a part of the security in computer science but very efficient for building a protected system. The goal of this work is to design and implement authentication and authorization section to PeerHood system. The thesis contents following chapters:

- Wireless and Ad-Hoc networking;
  - Wireless technologies;
  - Introduction to the security (authentication and authorization theory);
- PeerHood system;
- Design and implementation of authentication and authorization techniques into PeerHood;
- Conclusion and future work (recommendations).

[1] In this thesis ”mobile device” term means a device such as Pocket PCs, laptops, mobile communicators and so on, with any version of linux pre-installed. This device also has to support one or number of the following technologies: Bluetooth, Wi-Fi or GPRS

“Wireless and Ad-Hoc networking” chapter describes technologies used to build up wireless networks, advantages and disadvantages of different approaches. It also includes overview for Ad-Hoc principle of inter-organization such networks. “PeerHood system” chapter continues discussion about Ad-Hoc networking and gives an example of tool which provides certain functionality for creating Ad-Hoc networks. Next chapter – “Design and implementation of authentication and authorization techniques into PeerHood”, describes algorithms and security protocols designed and implemented in PeerHood system to provide authentication and authorization functionality. The last chapter - “Conclusion and future work”, gives recommendations concerning future developing of the PeerHood system.

2. WIRELESS AND AD-HOC NETWORKING

The history of information technology (IT) sector in last few decades demonstrates that the improvement and appearance of new technologies are unpredictable. None has even thought integration of four universities' networks to single ARPANET utilized 50 Kbit/s capacity could become epochal in the early seventies. Internet boom-growth in the late ninetieths was the next phase. Telecommunication sector stops on the brink of 3G mobile networks in the present. The cost of 3G-licensies hold up the rapid rise, yet the potential of Internet and mobile communication is not in doubt. [2]

Two or three years ago nobody could assume the future for technologies that made the IT-sector to grow again. Of course, I am talking about license-free wireless technologies of transferring data. The brand name for this approach is Wi-Fi. Initially these technologies were developed for creating rapid and cheap wideband networks. The networks were supposed to be small and designed only for usage inside office or house. However, they become very popular soon and hotspots began to appear one after another with amazing rate. Sometimes, new Wi-Fi hotspots covered entire blocks or even regions. Spreading of Wi-Fi is just got under way though. [2]

2.1. Wireless technologies

Wireless technology is a set of standards, protocols and tools provide communication services that are impractical or impossible to realize with the use of wires. That is telecommunications systems which use some form of energy (e.g. radio frequency, infrared light, laser light, etc.) to transfer information without wires. Information can be transferred in this manner over both short and long distances. This chapter gives a general overview for three ways of communication without the use of wires. Depending on the standard, spectrum usage, and frequency wireless networks can be split based on the access technology used. There are Wireless Fidelity, Bluetooth and General Packet Radio Service. [5]

2.1.1. Wireless Fidelity

This chapter is about wireless technology that is named Wi-Fi. Wireless Fidelity or simply Wi-Fi is a brand for the embedded technology of wireless local area networks based on the IEEE 802.11 specifications. Wi-Fi is used for mobile computing devices, such as mobile phones, laptops, digital cameras and others. Wi-Fi was licensed by the Wi-Fi Alliance.

There are several specifications of wireless local area network (WLAN). They are defined by a set of WLAN standards developed by working group 11 of the IEEE (Institute of Electrical and Electronics Engineers). Originally the 802.11 standard provided speed up to 2 Mbit/s. However, the latest version of the standard gets over this number. The maximum throughput of the recently developed standard is going to be even to 74 Mbit/s. 802.11n is that new multi-streaming modulation technique, although the standard is still under draft development. [6]

All currently used multiple over-the-air modulation techniques utilize the same basic protocol. The most popular techniques are those defined by the 802.11b and 802.11g. 802.11b was the first WLAN standard developed and accepted by IEEE in June 1999. The 802.11 family also includes other standards such as 802.11a, 802.11y, 802.11h and a lot of others. Those standards are service amendments and extensions or corrections to earlier specifications. [6]

The necessity of new standards is a growing interest to wireless techniques. It poses a problem of using uncommon operational frequencies in different countries and competition in throughput and range with wire techniques. Though, specified speed of the wireless communication is quite high. For instance, 802.11a supports bandwidth up to 54 Mbps, because of Orthogonal Frequency-Division Multiplexing (OFDM) and other improvements usage. This figure depends on lot of factors: walls, radio-frequency noises and utilized capacity of a radio channel and a real number is 21 Mbit/s. The same approach works for the range. In practice, hybrid solutions are often used such as combination of 802.11b and 802.11g or even 802.11a/b/g. [6]

Many companies deploy Wi-Fi networks and try to bring it in our everyday life. Perspectives of Wi-Fi are appointed on commercial and free basis by following companies: [7]

- Ozone and OzoneParis in France; the main idea of Ozone is to create a Wi-Fi network covering whole Paris. Ozone Pervasive Network evaluates on a national scale.

- Sify set-up 120 Wi-Fi hotspots in Bangalore. These Wi-Zones are extended across locations such as the IIM Bangalore, Vidhana Soudha including hotels, restaurants and coffee shops. It makes Bangalore India's first Wi-Fi-enabled city.

- A network of hotspots distributed over Brazil is provided by Vex. Telefonica Speedy Wi-Fi has run its services in a new and increasing network spread over the state of Sao Paulo.

- Golden Telecom offers telecommunications and Internet services in major population areas throughout Russia and other Commonwealth of Independent States' countries. The Company provides data, voice and Internet services to individuals, operators and corporations in major cities, including Moscow, St. Petersburg, Almata, Tashkent, Odessa, Kiev, Samara, Nizhny Novgorod, Krasnoyarsk and Kaliningrad.

There are two flows in the word that deploying Wi-Fi networks today. They are commercial services and volunteers. While commercial companies are making money by providing Wi-Fi access, a lot of individuals, and private bodies setting up free of charge Wi-Fi networks. Free wireless networks are considered the future of the Internet and often adopted to use common peering contract in order to share with each other openly. [7]

At the moment many municipalities and even entire countries already provide free access to Wi-Fi networks and access to Internet for everyone including Kingdom of Tonga and Estonia which have a large number of free Wi-Fi hotspots all over the countries, OzoneParis in France that provides a free Internet to everyone who allows using their rooftop for placing the Wi-Fi equipment.

Another example is Annapolis (USA). The company, Annapolis Wireless Internet, provides free Wi-Fi networks to all city residents. All expenses are repaying by advertisements. It means that any local company can order advertisement publishing through the city-wireless network and Wi-Fi users will see these advertisements upon accessing the network. Philadelphis is one more example of the cities that provides access to Wi-Fi network for free public usage. [7]

A lot of hotspots belong to schools and universities. Often universities are owners of big IT infrastructure and can provide Internet access (including Wi-Fi) to their students, staff and sometimes even to anyone in the range of Wi-Fi network. Many schools and universities also rent a capacity from other providers but have own local network infrastructure that may include Wi-Fi access points. The same way, some commercial companies provide access to free wireless networks. Some communities such as universities can provide their services for free to members and guests of the community, but at the same time they can let to use the services for everyone for money. A good example is Sparknet in Turku, Finland. Sparknet supports a wireless network that covers the area of entire Turku and offer free Internet access for students and Universities staff. [7]

The security of Wi-Fi networks was originally purposefully weak due to multi-governmental meddling on export requirements and was later improved via the 802.11i amendment after legislative and governmental changes.

2.1.2. Bluetooth

Another noteworthy wireless standard is Bluetooth - a radio standard and communications protocol considered for low power consumption. Bluetooth is a specification for wireless Personal Area Network (PAN). Bluetooth interface replaces wires and IrDA. Main distinguish is a short range based on low-cost transceiver chips which is built-in each device. Bluetooth provides exchange information and connection between two or more devices. It is used to connect secondary devices into single system with PC or laptop in its center (as a rule). For example, it allows using a mobile phone for Internet connection, digital photo or camera for copying multimedia data, Bluetooth printer for printing and

lot of others. To make all it possible the devices should be at short range, up to 10 meters though according specification Bluetooth provides the communication of these devices when they are placed on distance 10-100 meters from each other. [8]

First versions (1.0 / 1.0B) of the devices under different vendors were incompatible. Another disadvantage is that they sent its address (BD_ADDR) on stage of setting connection up. It made an impossible to create an anonymous connection. In the next version (1.1) of the protocol a lot of problems were solved. This generation also offered support for insecure channels and added RSSI. The following version of Bluetooth (1.2) provided an Adaptive Frequency-hopping spread spectrum (AFH) technology that increased resistance to interference. Increased capacity and quality of transferring multimedia information (such as voice) of new version influenced to popularity of Bluetooth. It is also known as first protocol included UART interface and duplicating of damaged data packets. Last generation of Bluetooth, version 2.0, includes support for Enhanced Data Rate (EDR) that allow raising capacity up to 2.1 Mbit/s. [9], [10]

Some Bluetooth kits also include point-to-multipoint way of communication in a Piconet's topology (star-shaped point-to-multipoint formation with a single link between each slave and a master) and Scatternet functionality. Point-to-multipoint communication provides nodes to participate in more than one Piconet at the same time (any node can be a master in only one Piconet) while Scatternet (Figure 1) is a structure formed by intercommunication of two or more overlapping Piconets. These techniques are very important in Ad Hoc Networking since they provide high flexibility and robust, however only few of the commercially available Bluetooth kits support them. [10]

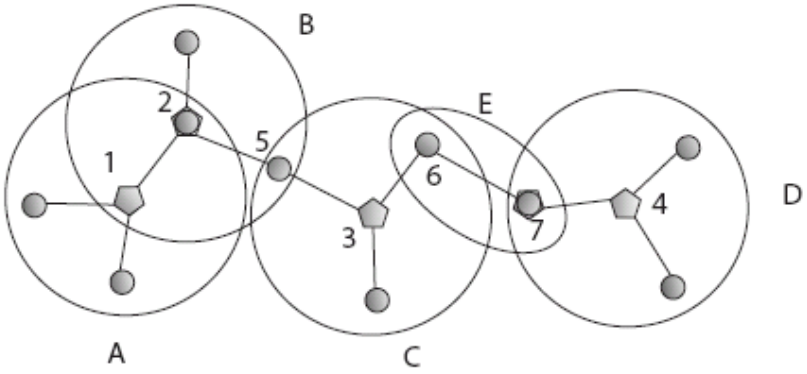

Figure 1. Scatternet formed from five Piconets, [10]

Following list enumerates the most important Bluetooth's profiles accepted by Bluetooth Special Interest Group: [8]

- Personal Area Networking Profile (PAN)
- Common ISDN Access Profile (CIP)
- Cordless Telephony Profile (CTP)
- Dial-up Networking Profile (DUN)
- File Transfer Profile (FTP)
- General Audio / Video Distribution Profile (GAVDP)
- Generic Access Profile (GAP)
- Intercom Profile (ICP)
- Service Discovery Application Profile (SDAP)
- SIM Access Profile (SAP, SIM)
- Wireless Application Protocol Bearer (WAPB)

For security purpose Bluetooth utilizes "Secure And Fast Encryption Routine" (SAFER+) authentication algorithm. For derivation initiation and main key it uses E22. Though, encryption in Bluetooth does not use SAFER+. Stream cipher

E0 is used instead. In general that approach makes listening devices connected by Bluetooth relatively hard.

2.1.3. General Packet Radio Service

General Packet Radio Service (GPRS) is an addition for technology of mobile communication – GSM. It provides transfer data in packets. GPRS offers to user of mobile phone to exchange data with other devices in the GSM and external networks (including Internet). GPRS also allows tariffing by value of sent/received data instead of time. [5]

The principles of functioning of GPRS are quite similar to Internet. Data are split into packets and then send to receiver. It does not limit the path of packet delivery though. Every device labels with unique address (analogously to IP address in Internet) during the process of obtaining connection. It makes each device like a server. The protocol of GPRS is transparent for TCP/IP. That is why the integration GPRS and Internet is easy but still very important for end-users. The packets can be in format IP or X.25, yet any standard protocol can be used on top of IP. Such protocols of transport and applied level as TCP, UDP, HTTP, HTTPS, and SSL can be utilized. When mobile phone uses GPRS the phone is a client of external network and have own IP-address (static or dynamic).

GPRS Applications:

- Mobile access to Internet; It provides acceptable capacity, rapid obtaining a connection and relatively small cost;
- Mobile and secure access for employees to corporate networks, remote databases, mail and information servers of the company;
- Telemetry; the device can be active and does not require a reserved channel. This system is important for police, security (signalling), banks (Automatic Teller Machines), in production (sensors, counters) and for personal usage (transport).

2.2. Mobile ad-hoc networks (MANETs) as an extension of wireless

Mobile Ad Hoc Network is a wireless network that can be created randomly and dynamically without the need for infrastructural settings. Such networks are self adaptive and can reconfigure themselves on the fly according to changing network topologies. The example of MANET is demonstrated at Figure 2. [10]

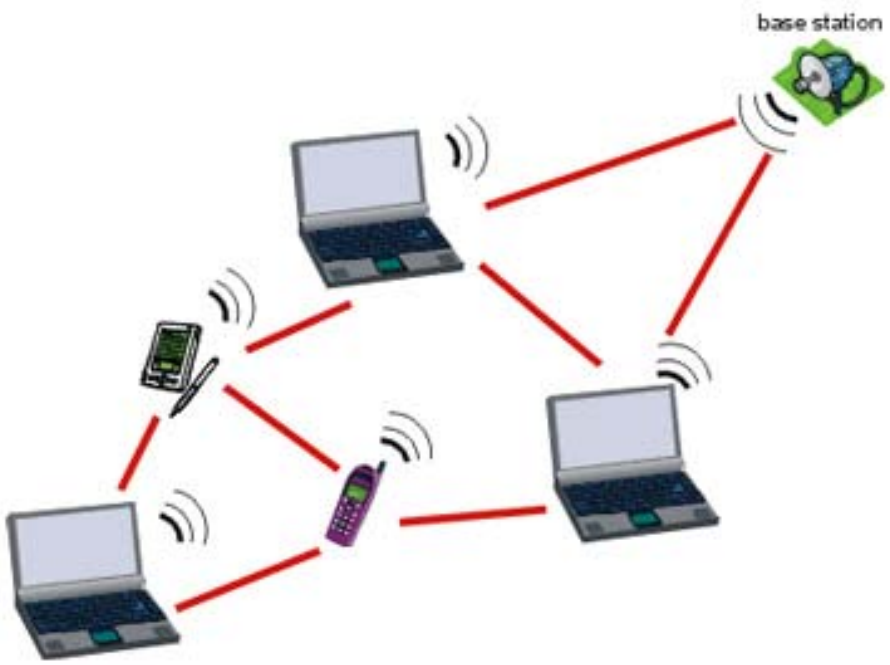

Figure 2. Mobile ad-hoc network example, [11]

A wireless infrastructure depends on centrally deployed hub-and-spoke networks while in ad hoc networks the devices themselves are the network. They are autonomously self-organizing in networks. This allows seamless communication, at low cost, in a self-organized manner and with easy deployment. That is the reason why MANETs completely different from any other networking solution. Only this type of networks provides the large degree of freedom and the self-organizing capabilities. For the first time, users have the chance to build their own network, which can be deployed easily and cheaply. However, complex technology solutions are a price for all those features. [10]

Therefore, mobile ad hoc networking is one of the more innovative and challenging areas of wireless networking. Moreover, this technology is going to

become increasingly present in everybody's life. Ad hoc networks are very important in the evolution of wireless networks. They inherit the traditional problems of wireless and mobile communications, such as bandwidth optimization, power control and transmission quality enhancement. In addition, the multihop nature and the lack of fixed infrastructure bring new research problems such as device discovery, network configuration and topology maintenance, as well as ad hoc self-routing and addressing. A lot of different approaches and protocols have been proposed and there are multiple standardization efforts within the Internet Research Task Force and the Internet Engineering Task Force, as well as industrial projects and academic. [10]

The following set of characteristics [10] can be generalized for all Ad-Hoc networks.

- nodes should communicate wirelessly;
- the network is temporary and forms dynamically in an arbitrary manner by a collection of nodes;
- the network should not depend on any centralized administration or infrastructure;
- nodes operate as routers;
- nodes are free to move and the topology of such network is dynamic.

2.3. Security

Information security is a part of science studying and developing methods of protecting data from illegal access and modifications, from leak, damage, or disruption. Information security cares about data in any format and storing form. It could be printed or handwrote information, or information in electronic form.

The question about information security is extremely important nowadays. Both government and businesses are trying to keep and protect their secrets such as technical details or/and source codes about its products, customers' and employees' confidential information and lot of others. The information itself is often a subject of businesses. Individuals or companies are building their

business on selling information. This market grows, because of Internet. The success of such businesses is completely depends on security of the information. For example the electronic book bought in Internet shop should be protected from further redistribution. After all, the information privacy is a business and an ethical requirement. [12]

At least three levels of protecting should stand between the information and the one who wants to obtain the access to the data (human, device, information system etc.). They are an identification of requesting side, clarifying the permits level to the data and defending the communication channel among computer systems or devices. [12]

There are following mechanisms for access control:

- Authentication
- Authorization
- Encryption

2.3.1. Authentication

Authentication is a process of identification a personality of the device, person or any kind of computer system by another side. The system checks that an authenticated part is a certain identity during the authentication process. Person's authentication factors are generally classified into three cases: [13]

- The user is or does something

There are physical features such as DNA sequence, palm prints, fingerprint, retinal pattern, unique bio-electric signals, or another biometric identifier. It includes also signature and voice recognition.

- The user has something

The most popular is ID card, a magnetic swipe card, security or software tokens. Identification card often stores some personal information about the user, private and public keys or/and some other relevant information while a Software token is a form of two-factor authentication security device. The token operates on a

general purpose electronic device like a mobile phone, PDA, laptop or desktop computer, while hardware tokens store the user's credentials on a dedicated device. Software tokens are weaker than hardware tokens, because they are exposed to threats such viruses and malicious software attacks.

- The user knows something

It could be a password, pass phrase, or PIN (personal identification number). The password authentication is still often in use today, this way of authentication is the most common method that requires the matching of a username with its associated password. The information about username and password often stored in database and sometimes encoded. User's password can be replaced with its hash to increase security.

It is very popular to use combination of enumerated approaches for authentication today. This way allows making the process very secure. One can even use all methods together to reach the highest security level.

2.3.2. Authorization

Once a person, device or program has successfully been identified and authenticated it has to be determined what kind of informational resources they are permitted to access and what actions they are allowed to perform (run, create, view, delete or modify). It is called an authorization procedure.

Authorization to access information or services begins with administrative policies and procedures. This polices set what information and services can be accessed under what conditions, and by whom. The access control mechanisms are then configured to put into effect these policies. [14]

2.3.3. Wireless security

Wireless networks have many security issues. They are easy to break into if compare with wired networks, and even can be used to crack into more secured networks. Growth of popularity of wireless services causes to increasing a risk for users of such networks. There are a large number of security challenges associated with the current protocols of authentication, authorization and encryption algorithms in wireless networks. [11]

A lot of early access points could not detect if or not a particular user had authorization to access the network. In wired networks where this problem was discovered during long history of its evolution, this did not cause to a significant problem, because of physical security. However, the fact that radio signals leaks out makes physical security completely irrelevant. [15]

2.3.4. Ad-hoc wireless security

Ad-hoc networks cause a security threat. They are defined as peer-to-peer networks between wireless devices that do not have an access point among them. These types of networks still have slight protection. [11]

In the chapter 2.3.3 described that wireless networks are weaker to information security threats than fixed-wire-line networks. For mobile ad hoc networks the security approach is even more critical, since it is a distributed within unfixed infrastructure network, but it also relies on individual security solution from each mobile node and used application. Moreover, it is almost impossible to implement centralized security logic. Source [10] describes following security requirements in ad hoc networking:

- confidentiality, has a deal against passive listening;
- access control, protects access to wireless network infrastructure;
- data integrity, prevents fabrication of traffic;
- protecting against denial-of-service (DoS) attacks by malicious nodes.

The access control requirement is quite similar to wire-line networks. That is the starting step in wireless network security. Authentication techniques are mainly used to protect access to wireless network infrastructure and to prevent fabrication of traffic. The best known way to prevent fabrication of routing information and data traffic is using digital signatures. Certification authority function could not be implemented in mobile ad hoc network, and this role must be distributed over all nodes in the network. In addition to authentication, encryption should be used to raise confidentiality, to protect information during transmission, to prevent passive eavesdropping and data integrity attacks. [11]

Advanced security mechanisms include techniques for intrusion-resistant ad hoc routing algorithms and intrusion detection. These methods use characteristic "training" data and intelligent protocols for early detection of intrusion into the network. [11]

3. PEERHOOD SYSTEM

PeerHood [16] is a system, which offers a communication environment where mobile devices communicate in a peer-to-peer manner. PeerHood provides applications with a common interface to several wireless network technologies: Wi-Fi, Bluetooth and GPRS.

The applications can register their services to the local PeerHood service database, which includes all local services registered to the PeerHood system. The PeerHood daemon is the component, which manages the database, as well as takes care of discovering other devices and services. When the application wants to discover PeerHood services in other devices, it connects to the local PeerHood daemon and requests the device and service records for other PeerHood devices. If a desired service is found on another nearby device, a connection can be established between devices using PeerHood connection architecture. This connection can be used to transfer data between devices regardless of the used network technology.

In short, PeerHood is a system, which provides:

- Device discovery;
- Service discovery;
- Service sharing;
- Connection management using different wireless network technologies.

Essential component of such system is security. Security involves things such as authentication, authorization and encryption. Authentication is the first and important part of the security process. We must be able to proof, that the person is who he claims to be. After a successful *authentication*, he can use a certain service, if he's *authorized* to do so. The authorization process sets certain

requirements for the system. We must be able to assign persons or devices a set of rights for different services and resources and check those rights when he's trying to connect to a service or resource. Then, if required, the data transfer between the devices is *encrypted*.

3.1. A plan of improvements

The task is to design and to include all modifications in existing software needed to control incoming and outcoming connections as well as access to services and any objects on mobile devises. This security approach is founded on user permission database.

Very briefly, there are three main proposals. Firstly, a database to describe foreign permitted devices to a local one with user rules; secondly, a system of mobile device owner authentication (local authentication, Client-daemon and Service-daemon); thirdly, a system of remote (Daemon-Daemon, Client-Service) user authentication.

I have split the development process into stages whose result will be completed versions, i.e. software work with gathered functionality, although the whole set of the features is not implemented.

3.1.1. Design of the whole system

First step in development of every system is in-depth examining of a current system. I have only to understand a work of communication part of PeerHood. It includes all types of connections provided by the system. The way how incoming connections are proceed what kind of information are sent and received, what sort of signals are there. When the description of the system is ready I can design the system of authentication that can be imposed on PeerHood application structure.

3.1.2. Database design & implementation (Version 1)

This is the largest stage in the project. It includes building a logical database structure according to the authentication system that designed. Database should store all information required for building authentication and authorization. The

implementation has to be able to create a full database structure in case it is not exist, and to get a complete set of function to manage records of the database.

3.1.3. Local authentication (Version 2)

The main issue of this step is to recognize a user started a client or service on the device. Before the client or service application can use Daemon services it must pass through authentication at the Daemon. This procedure is done just once in compare with low-level authentication that is performed every time the connection is needed.

3.1.4. Remote authentication (Version 3)

Remote or low level authentication provides authentication between two daemons, two devices. It is the important part from security point of view. It offers protected connection for local clients to remote services.

3.2. An ideal example of user connection and work

We have two devices and one of them (let's call it - "Tikki") wants to use an Internet service provided by another one.

Tikki has already completed a process of device discovery and knows that there is only one device. Tikki starts a process of service discovery. It sets a connection to the first device using Bluetooth technology and tries to authenticate itself and remote device. Tikki sends request for authentication and waits for a random number and name of the user of the first device. When Tikki receives both values it sends its name and hash based on password corresponds to remote user and received random number. First device checks the information Tikki sent and if it is successful generates a new unique token for this connection. As reply for authentication first device sends this token.

Later, when a client started on Tikki uses the token to connect to remote services. The first device also uses token system to control permits for the user.

4. IMPLEMENTATION

The purpose of this part is to describe the protocol that is used for setting a security in PeerHood system. It describes all changes that were performed with

original software, provided interfaces, functionality and some other important issues. First of all let's cover ground of authentication and alteration of PeerHood modules.

4.1. Basic principles of authentication in PeerHood

Building on the authentication system is completely depends on original architecture of the system. The fact, that the system is split into three strong independent parts makes this task extremely challenging. I describe the modules of PeerHood below and give short notices. PeerHood system consists of three separated modules: Daemon, Client and Service:

- The **Daemon** is PeerHood itself in this thesis. The description given in 3 PEERHOOD SYSTEM is about the Daemon and its branches such as plugins and external libraries that provide functional methods for third party software.

- A **Client** is a kind of third party application that allows user to obtain access to remote services. It uses interfaces of PeerHood libraries and communicates with the local Daemon via Unix sockets (sort of network connection).

- A **Service** is a third party application that offers services to remote users. To make service available through PeerHood the application has to register it on the local Daemon. It uses interfaces of PeerHood libraries and communicates with the local Daemon through Unix sockets.

The implementation of security issue is done by modifications built on the Daemon part. It means that communication between any two Daemons (communication between PeerHood systems) is protected. However, the client and the service don't connect with each other through PeerHood system; they use their own communication channel for that goal. PeerHood provides functionality for third party software but it has to use them properly to reach a real security. I also want to note that the way how authentication and authorization is implemented in both client and service applications are

completely defined by their producer. It makes this solution flexible but not uniform.

4.2. Modifications of PeerHood modules

PeerHood modules were improved to introduce authentication and authorization approaches to the system. There are two essential changes. One of them is adding a database manager to provide storing information about users, groups, devices, services and their relationships. The second change is about Daemon part and plugins.

Database manager is represented in the CDBManager class. This class was completely designed and implemented in the context of this Master Thesis. You can read a full description of it in the 4.3 DB Design and 4.5 Implementation of DB interface chapters.

CDaemon class included implementation of the PeerHood's Daemon subsystem has been fundamentally changed as well. (Full description of the changes described in the 4.8 Description of the Daemon interface for policy management chapter) Handlings of following events were added:

- Event of creating new connection from client/service;

Cleaning of information about a current state of the connection (initiation phase).

- PH_ADD_NEWUSER

Processing of the event connected with attempt to create a completely new user on the local system.

- PH_ADD_NEWREMOTEUSER

Processing of the event connected with attempt to create a new remote user on the local system. A new local user also will be created during this operation.

- PH_ADD_NEWGROUP

The event generates when a request for creating a new user group arrives. The way of organizing groups is outside the scope of PeerHood system. This is

configured by software (clients and services) which is going to use functionality of the system. Though it is possible to disable PH_ADD_NEWGROUP event to prevent creating new groups and to make third party software to use only limited amount of groups predefined in the daemon initialization function (CDaemon()). The examples of creating such groups are presented in the code:

```
m_db.NewGroup("personal");
```

This example calls the method *NewGroup* of the build-in *m_db* object (CDBManager class) to create a new group with name "personal".

- PH_GET_LTOKEN

The event is a request for local authorization token. This token is used by local service to check if remote client is authorized to use it. This event provides functionality for third party services; however they also have to support this technology.

- PH_GET_RTOKEN

The event is a request for remote authorization token. This token is used by local client to obtain a secured connection with remote service. This event provides functionality for third party clients; however they also have to support this technology.

- PH_LOCAL_AUTH_REQUEST

The event means a request for authentication from local client or service. The Daemon expects some exchange of the authentication information after handling this event. Unauthenticated client or services could not use Daemon external interfaces and functionality. Any attempt to generate another event before authentication leads to disconnection. Detailed description of the exchange protocol is in the 4.7 Client and Service authentication chapter.

Methods and functions that were changed in the latest implementation of CDaemon are represented below:

- CDaemon constructor;

Reading Linux environment (PH_LOCAL_USER_NAME) and database initialization (given as an example, commented in the latest release) are added at this part.

- Run() method;

In “Run” method added handling of the events described above.

- HandleLocalAuthRequest;

Local authentication request is a request for authentication from local client or service. This new procedure validates connection of third party software to the Daemon. It runs when PH_LOCAL_AUTH_REQUEST even arrives.

- HandleGetLocalServiceList;

The system had this method before modifications. The purpose of that is to send to a client application a list of available services on a local device. The updated version of the method returns only services to which a user of the client has a read permit (authorized) according to local security policy.

- HandleInsertService;

The system had this method before modifications. The purpose of the HandleInsertService function is to add a new registered service on a local device. The updated version of the method sets a full set of permits on added service for the owner (user who provides the service).

- HandleRemoteAuthRequest;

That is a new method. HandleRemoteAuthRequest is one of the core procedures of the authentication system. The function is called to introduce a local user to remote Daemon and used as opposite part for SendAuthRequest() method.

- SendAuthRequest;

That is a new method. SendAuthRequest is one of the core procedures of the authentication system. It is called to ask remote Daemon about its authentication

(to send authentication data to local Daemon) and used as opposite part for HandleRemoteAuthRequest () method.

- HandleUserPermit;

That is a new method. It allows to set a new set of permits to {user; service} or {user; all services} processions.

- SendServiceList.

That is one of the original PeerHood's methods. It sends all local services to by the remote request. The new version of the method sends only services to which a user of the remote client has a read permit according to local security policy.

As an example of using these methods Bluetooth and WLAN plugins were modified. *AdvertThread* and *FetchInformation* functions of *CBTPlugin* and *CWLANPlugin* classes were updated. After setting a new connection these methods are doing/expecting authentication actions in addition. If exchange packets among these entities are invalid or the phase is missing Daemon breaks the connection, otherwise new tokens are generated. The last note means that the third party software should update used security tokens before every request to remote device.

There is also a new function in CPeerHoodImpl class. This is *MakeSHA* procedure. This method should be used by third party software to convert its authentication data according to policy of PeerHood system (protecting data during transmission).

4.3. DB Design

The main aim of this database is the storing of permanent information about mobile devices, its users and groups with some settings of privileges given them by a local user. Representation of DB format is MySQL database. Figure 3 shows the model of this database. Internal DB Interface interacts with DB and gives the opportunity to edit, add or remove users, groups or rules from database content.

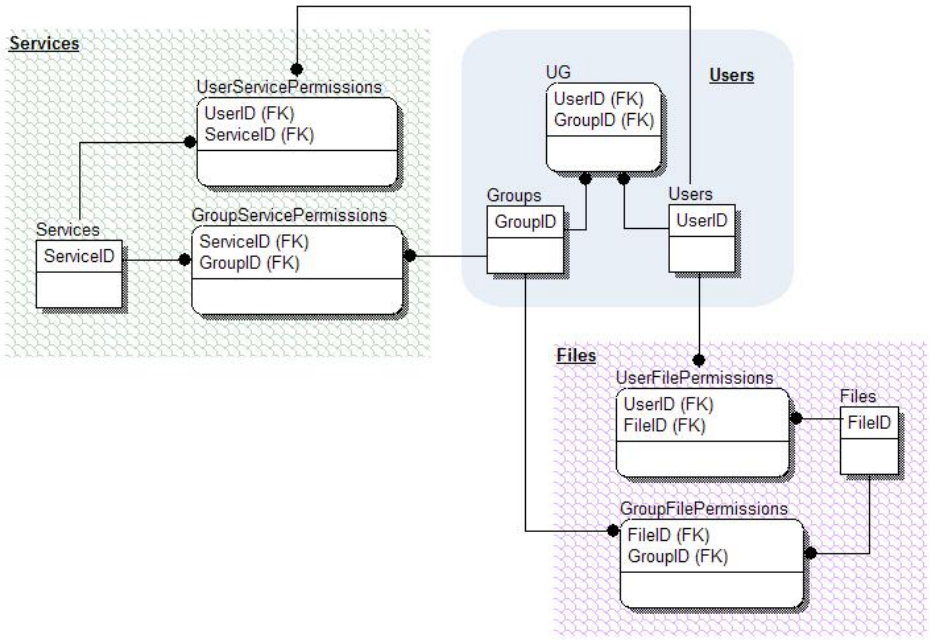

Figure 3. DB Description

4.4. Authentication approach

A general chart of the authentication procedure is demonstrated at Figure 4. The whole system contents three types of authentication: Daemon-Daemon, Client-Daemon and Service-Daemon. Once all required authentication processes are done successfully, a local client can obtain access to a remote service using a token generated by the local daemon for a certain connection to the remote device. Besides, there is need to protect the application from the local malaises. The client's application part as well as service's should prove their personality to the Daemon. Successful authentication allows to third party applications:

- Registering new services;
- Getting list of devices;
- Getting list of services;
- Controlling permits for its services;
- Receiving authentication tokens to authenticate remote connections;

- Using information needed for authorization.

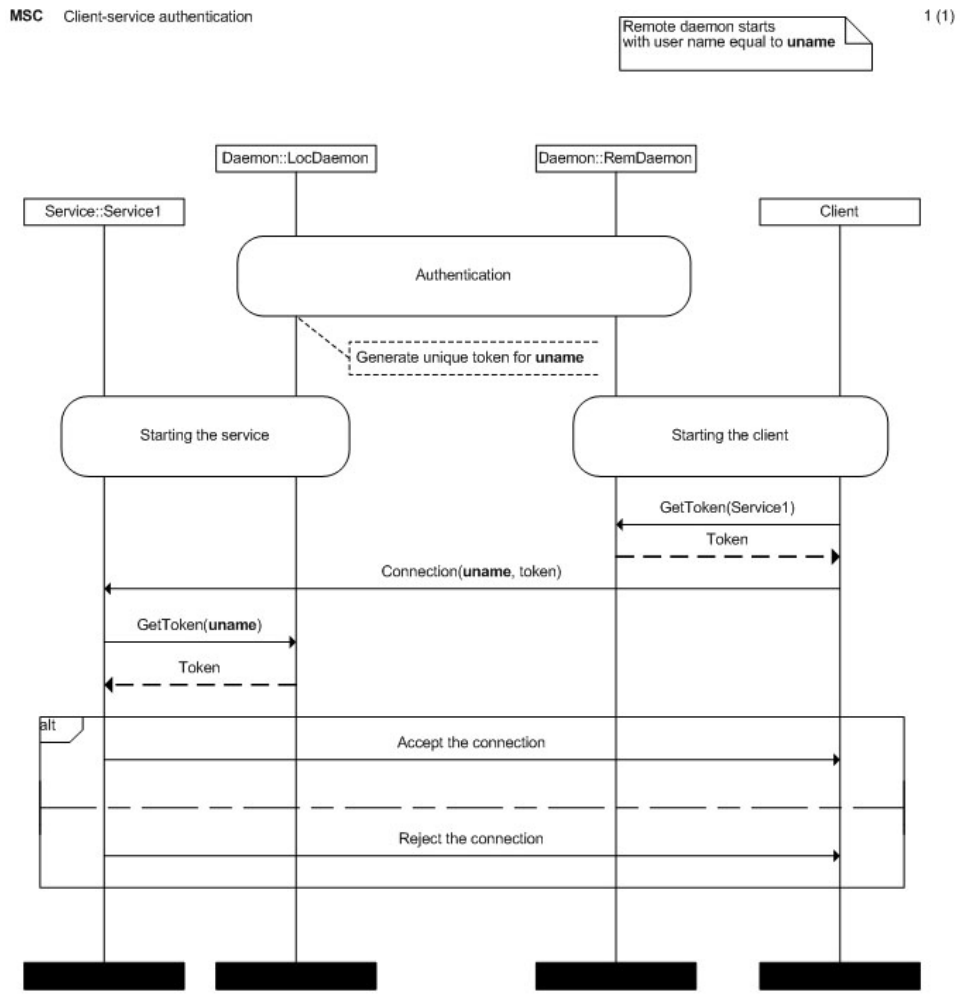

Figure 4. Authentication in PeerHood system

According to the security approach used in the project the daemon part is responsible for managing user accounts and key storages. Device authentication is done by daemon and described below at Figure 5

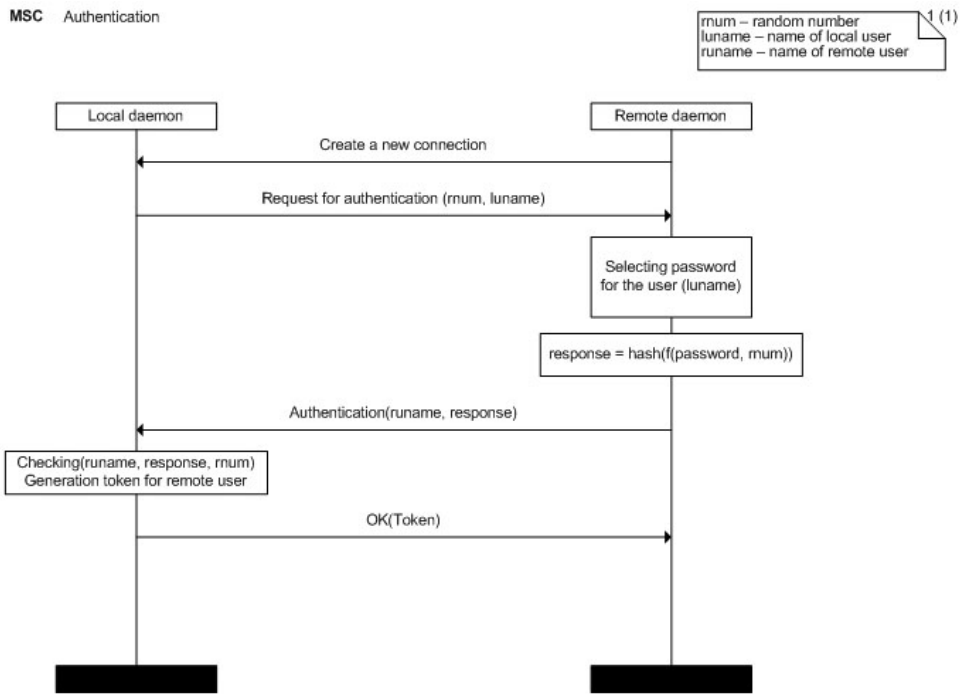

Figure 5. Device authentication

There is a process of authentication among remote and local daemons. The procedure started from request for creating a new connection by remote device. The local Daemon sends request for authentication then. This request contents user name and random number used for creating a unique hash sequence from the password by the remote system. One can see from Figure 3 DB Description, every Daemon has a table of records about registered remote users. The records also store a password to each user. It allows using a separate password for every remote system. This password together with random number is used to create the hash. A remote computer sends this hash along with the user name of remote Daemon. In case of successful authentication it performs in the back order.

The authentication of the application provided services is shown below at Figure 6. This scheme was implemented at the service application with purpose to test work of the PeerHood authentication and can be very different in real. Authentication data in this case pushed in the service application as parameters of command line and marked as attributes of Start event in the figure. The start of the application runs a process of searching a local Daemon and an attempt to connect to it. Once the connection is established Service application sends

PH_AUTH request. The request shows a wish of the Service to pass authentication at the Daemon. The Daemon replies with a random number and waits for authentication data. If authentication is not succeeded the Daemon breaks the connection.

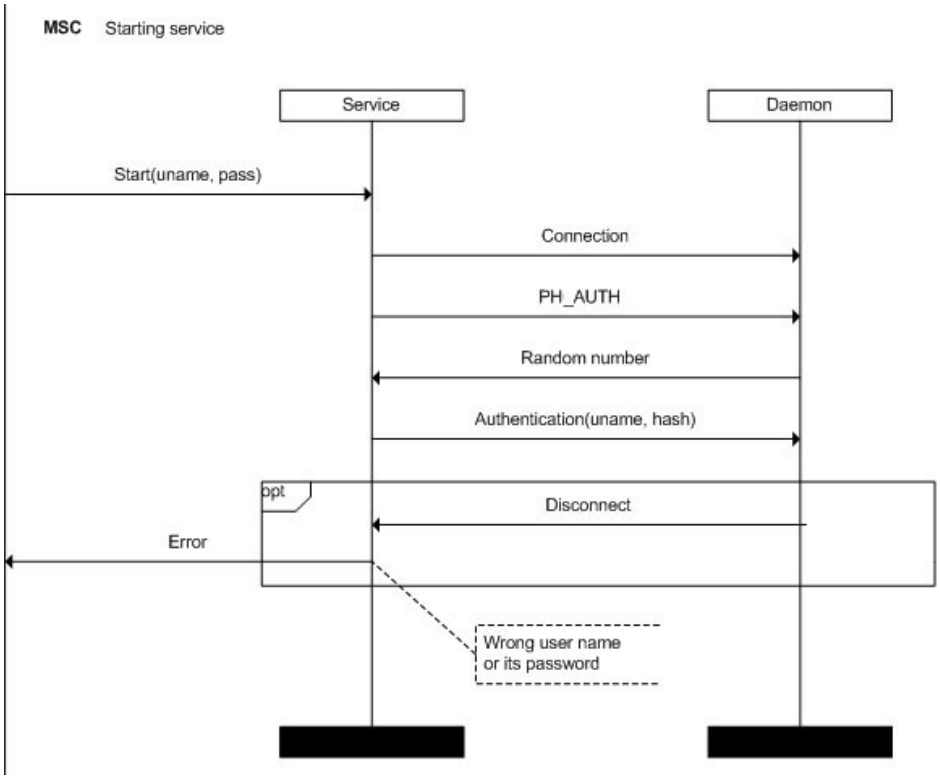

Figure 6. Service-Daemon authentication

The very similar case is for the Client application that is demonstrated at Figure 7

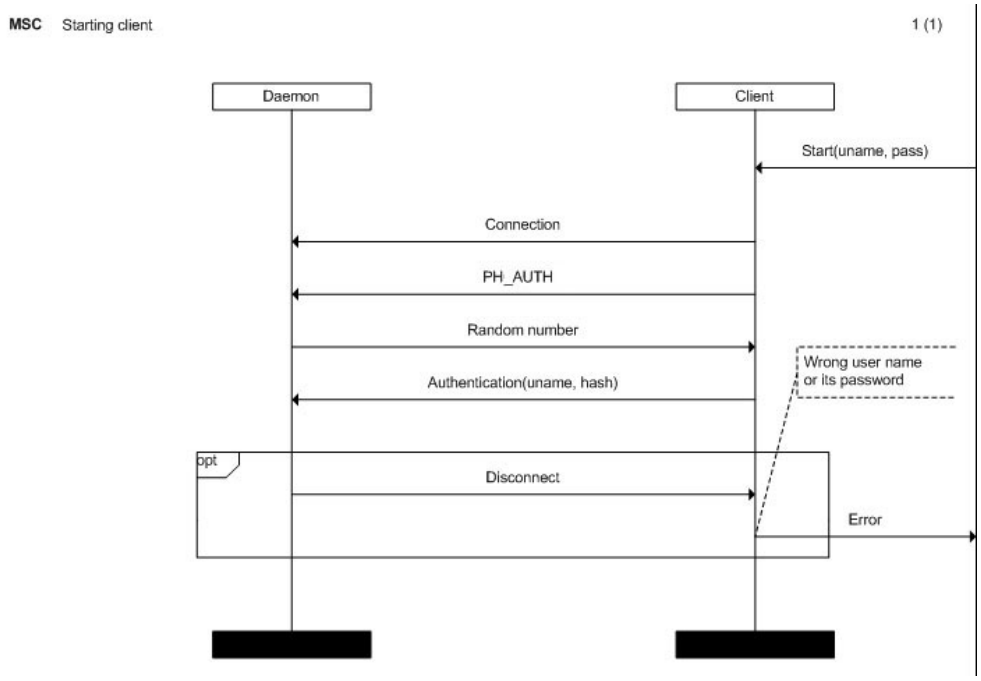

Figure 7. Client-Daemon authentication

4.5. Implementation of DB interface

To make the authentication possible the following list of methods and functions is implemented in CDBManager class:

Table 1. CDBManager methods of manipulation with entities

CreateDBStructure	Rebuilding database structure in case of damages or if first start
AddRemoteUser	Adding new remote user to the database, it allows to set a name and password of a user
NewUser	Adding new local user to the database, it allows to set a name
SetUserKey	Set the password for the local user
NewGroup	Adding new group to the database, it allows to set a title of the group
AddUserToGroup	Adding user to the certain group

Table 1. CDBManager methods of manipulation with entities

NewService	Adding new service to the database, It allows to set service's name, owner's name, device's iPid, service's port
NewObject	Adding new object (file) to the database, It allows to set object's path (second common attribute), object's name (personal attribute), type of the object (first common attribute)

Table 2. CDBManager methods of changing users' and groups' permits

SetServicePermitForUser	Adding or changing rules in database about user permit to the certain service
SetServicePermitForGroup	Adding or changing rules in database about group permit to the certain service
SetObjectPermitForUser	Adding or changing rules in database about user permit to the certain object (is not specified, can be used for any purpose by third party software; to control access to files or different functions of the same service; the external interface in the Daemon part does not represent this sort of methods)
SetObjectPermitForGroup	Adding or changing rules in database about group permit to the certain object
GetAuthTokenL	Returns local authentication token by user name (for services)
GetAuthTokenR	Returns remote authentication token by user name (for clients)
GetAuthTokenRbyService	Returns remote authentication token by owner of service

Table 2. CDBManager methods of changing users' and groups' permits

SetAuthTokenR	Setting a new remote authentication token; The token is obtained from remote Daemon after successful authentication phase.
GetRemPassword	Returns the password of the remote user (registered in the local database); This password should be used to get access to a local device from this certain user.
UserValidation (v1.0)	*Deprecated function;* The parameters are {const string name, const string key} The function was used in the first version of the PeerHood system; It uses a password as it is for authentication; This approach is low secure and not used anymore; see next function.
UserValidation (v2.0)	*The latest version of user validation function.* The parameters are {const string name, const unsigned char *hkey, const int randnum} This function is base on checking a hash of the password (stored in database) modified according to a random number with the hash (hkey) given as a parameter. This function is used for both local and remote user validation with no difference. In case of remote registration a new authentication token will be generated.
MakeSHA	Converts given string (the password is expected) according to the given random number and SHA hash is calculated from the result.

Table 3. CDBManager methods of obtaining users' and groups' permits

GetServicePermit (by user name)	Calculates and returns permit of the user for certain service; this function accumulates both permissions to user and to groups where user included. Function returns 1 byte value that contents access to the service; It means that all prohibitions are taken into account. (denies have priority)
GetServicePermit (by token)	Calculates and returns permit of the remote user (defined by its token) for certain service; this function accumulates both permissions to user and to groups where user included. Function returns 1 byte value that contents access to the service; It means that all prohibitions are taken into account. (denies have priority)
GetServicePermitForUser (by user name)	Returns two bytes value where the first byte is reserved to grant an access and the second one is to deny some options; The value includes only parameters of access specified for the user and does not include group privileges.
GetServicePermitForGroup (by group name)	Returns two bytes value where the first byte is reserved to grant an access and the second one is to deny some options; The value includes only parameters of access specified for the group.
GetServicePermitForUserGroup (by user name)	Returns two bytes value where the first byte is reserved to grant an access and the second one is to deny some options; The value accumulates permits of all groups where user

Table 3. CDBManager methods of obtaining users' and groups' permits

	included; the logical OR operation is used. (if some operation is allowed in any group it is allowed in result)
GetServicePermitForUser (by token)	Returns two bytes value where the first byte is reserved to grant an access and the second one is to deny some options; The value includes only parameters of access specified for the remote user and does not include group privileges.
GetServicePermitForUserGroup (by token)	Returns two bytes value where the first byte is reserved to grant an access and the second one is to deny some options; The value accumulates permits of all groups where remote user included; the logical OR operation is used as well.

4.6. Low level authentication

The authentication between Daemons is performed by pluging modules as a phase of new connection creating. Methods *AdvertThread* and *FetchInformation* are updated to include activation of the authentication process. They execute *SendAuthRequest* and *HandleRemoteAuthRequest* functions of Daemon's instance. Below you can find a description of the algorithms for both methods. That is not a source code, but pseudo code.

Listing 1. SendAuthRequest

```
Receive(randnum); // receiving a random number
Receive(length);  // receiving a length of
                  // a following packet
if(length > 0 && length < MAX_REQUEST_LENGTH) {
  Receive(data);  // expecting data of [length] size
  data[length] = 0;
```

Listing 1. SendAuthRequest

```
  name = data;    // the first part of
                  // the data is a user name
  // getting a password for the user (by [name]),
  // result is stored to [key]
  // m_db is an object of CDBManager class
  if(m_db.GetRemPassword(name, key)) {
    // filling [hkey] with hash for [key]
    // modified according to [randnum]
    m_db.MakeSHA(key, randnum, hkey);
    // calculating length of the packet for sending
    full_length = local_uName.size()+SHA_DIGEST_LENGTH+1;
  }
}

if(full_length <= MAX_REQUEST_LENGTH && full_length > 1) {
  Send(full_length); // Sending length of the auth packet
  // creating a new [data] array with auth. information
  strcpy(data, local_uName.c_str());
  data[local_uName.size()] = 0;
  memcpy(&data[local_uName.size()+1],
         hkey, SHA_DIGEST_LENGTH);
  // sending auth. data
  Send(data);

  // waiting for a token of us
  Receive(Token);
  m_db.SetAuthTokenR(name, Token);
  retval = Token > 0;
}

return retval;
```

Listing 2. HandleRemoteAuthRequest

```
number = rand();
Send(number);         // sending a random number first
// defining a length of a local user name
```

Listing 2. HandleRemoteAuthRequest

```
length = local_uName.size();
Send(length);         // sending the length
Send(local_uName);    // sending the name
Receive(full_length); // reading reply's length
if(full_length <= MAX_REQUEST_LENGTH && full_length > 1) {
  Receive(data);      // reading a packet
  // taking the user name from the packet
  part_length = strlen(data);
  data[full_length] = 0;
  name = std::string(data);
  if(full_length - part_length - 1 == SHA_DIGEST_LENGTH) {
    // taking a password's hash from the packet
    bcopy(&data[part_length+1], hkey, SHA_DIGEST_LENGTH);
    // checking received data: [name] – the user name;
    // [hkey] – password's hash; [number] – random number
    // m_db is an object of CDBManager class
    m_db.UserValidation(name, hkey, number);
    // getting a token for the connection; if auth.
    // is successful the token is a positive number
    Token = m_db.GetAuthTokenL(name);
    Send(Token);
    retval = Token;
  }
}
```

Token generated in UserValidation is a random number in the range from 0 to 32767.

Listing 3. MakeSHA(const string str, const int randnum, unsigned char *hash)

```
// initiation
retval = false;
  // a length of the password ([str]) is into [length]
int length = str.length();
  // copying the password to [oldstr]
oldstr = str.c_str();

// converting the random number from int to 2 byte char
```

Listing 3. MakeSHA(const string str, const int randnum, unsigned char *hash)

```
bcopy(&randnum, randnumS, sizeof(int));

// going through the password
for(int i = 0; i < length; i += sizeof(randnumS)) {
  // putting binary multiplication of password and
  // random number word by word (two bytes) into [newstr]
  op = &oldstr[i];
  np = &newstr[i];
  for(int k=0; k<sizeof(randnumS) && i+k<length; k++) {
    np[k] = (unsigned char) op[k] & randnumS[k];
  }
}
// calculating hash from [newstr]
retval = (SHA1(newstr, length, hash) > 0);

return retval;
```

4.7. Client and Service authentication

Both Client and Service authentication is started after PH_LOCAL_AUTH_REQUEST event that is processed by Daemon in run() method. This event is generated as a reaction for corresponding command message from the Client or Service applications.

Listing 4. Run()
PH_LOCAL_AUTH_REQUEST event

```
// executing HandleLocalAuthRequest method
// with pointer to the information about connection
if(HandleLocalAuthRequest(**i)) {
  // Validating received data
  if(m_db.UserValidation(name, hkey, randnum) != DBU_OK) {
    // if authentication is failed closing
    // the connection and removing the record
    // about the Client or Service
    close(newFd);
    iClients.erase(i);
  }
```

Listing 4. Run()

```
PH_LOCAL_AUTH_REQUEST event
}
```

Listing 5. HandleLocalAuthRequest

```
retval = true;

randnum = rand();
Send(randnum);        // sending a random number first
Receive(full_length); // Reading a length of a reply

if(full_length <= MAX_REQUEST_LENGTH && full_length > 1) {
  Receive(data);      // Reading the reply data
  // extracting a user name of remote device
  // from the reply
  part_length = strlen(data);
  data[full_length] = 0;
  name = std::string(data);

  if(full_length - part_length - 1 == SHA_DIGEST_LENGTH) {
    // copying a hash from the reply to [hkey]
    bcopy(&data[part_length+1],
           hkey, SHA_DIGEST_LENGTH);
  }
}

return retval;
```

The implementation of this technique on a remote side (clients / services) is done by hooking initiation method in PeerHood library used by third party applications. To initialize the library a client and a service have to give user authentication data as a parameter. In current examples of clients and services the command line parameters applied for that. Expected names of the parameters:

- "-user" – is used to set a user name;
- "-key" – is used to set a password.

The following description demonstrates the algorithm in detail:

Listing 6. CPeerHoodImpl Init(int aArgc, char aArgv)**

```
// reading user authentication data from input parameters
for (int i = 0; i < aArgc; i++) {
  string arg = string(aArgv[i]);
  // defining the pointer to user name
  if (arg.find("-user=", 0) == 0)
    user = aArgv[i] + 6;
  // defining the pointer to user password
  if (arg.find("-key=", 0) == 0)
    key = aArgv[i] + 5;
}
// informing PeerHood that we want to pass authentication
SendCommand(PH_LOCAL_AUTH_REQUEST);
Receive(randnum);    // receiving a random number
// calculating hash into [hkey]
MakeSHA(key, randnum, hkey);

// calculating a length of the authentication packet
length = strlen(user) + SHA_DIGEST_LENGTH + 1;
Send(length);        // sending the length
// Forming the authentication packet
strcpy(auth_data, user);
auth_data[strlen(user)] = 0;
// copying the password hash at the end of the packet
memcpy(&auth_data[strlen(user)+1],hkey,SHA_DIGEST_LENGTH);
Send(auth_data);     // sending the authentication packet
```

4.8. Description of the Daemon interface for policy management

This chapter describes interfaces offered by PeerHood to third party applications. It is given in the two-column tables where the left side represents PeerHood part and the right - expecting client actions. The title of tables lists events for which description is applicable. All data in packets are separated by zero HEX code (0x00h).

Listing 7. PH_ADD_NEWUSER, PH_ADD_NEWREMOTEUSER	

`Receive(Packet Length);`
`Receive(Packet);`
`{ new user name, password } = Packet;` [2]
	`Send(Packet Length)`
	`Packet = { new user name, password };` [3]
	`Send (Packet);`

Listing 8. PH_ADD_NEWGROUP	

`Receive(Packet Length);`
`Receive(Packet);`
`{ title } = Packet;`
	`Send(Packet Length)`
	`Packet = { title };`
	`Send (Packet);`

Listing 9. PH_GET_LTOKEN, PH_GET_RTOKEN	

`Receive(Packet Length);`
`Receive(Packet);`
`{ user name } = Packet;`
`Send(Token;)` **`// 4 bytes`**
	`Send(Packet Length)`
	`Packet = { user name };`
	`Send (Packet);`
	`Receive(Token;)` **`// 4 bytes`**

Algorithms presented in Listing 7 - Listing 9 are quite similar with each other and very simple. They content just couples of receive-send functions. It allows easy to implement template for corresponding methods in client's and service's software.

[2] Operation "{…} = Packet" means extraction data from [packet]
[3] Operation "Packet = {…}" means wrapping data into [packet]

5. CONCLUSIONS AND RECOMMENDATIONS

This work is very good experience in two rapidly-developing and important fields. The Information security and mobile communication obtain more and more importance in the business and private life. Mobile device are not used by small group of people anymore. I have a personal experience when a boy of six years old used a laptop for education purpose. Not every family can offer so high efficient way of knowing the outward things to their children. However, this is a future and only strong security system can protect private life of each and every user in modern world. Adding mobility to IT world make the task much more complex. We cannot build a security wall around every mobile device, because no wires inside which it suppose to be build. It means new issues and new challenges but new possibilities and novel opportunities as well.

This project does not claim to originality. There are many solutions for the problem made by such monsters in IT industry as Nokia, Siemens and other corporations. But this work shows the alternative way of resolving the puzzle and our view to the question. It is more flexible and this is open source that means every user can create its own service or client application or even to improve a PeerHood part.

I can give some recommendations for future improvements of the security in PeerHood system:

- Developing services and clients with full support of the security interfaces of the PeerHood Daemon;
- Developing new security interfaces in the Daemon part to use all functionality of the database, 4.5 Implementation of DB interface chapter;
- To improve the security system by replacing password-token authentication with digital certificates.

An implementation of new services and powerful client gives to PeerHood system chance for future improvements. It also shows the potential of this tool.

Creating of the client can be even separated in individual project. At the same time new functionality can be implemented in daemon of PeerHood systems.

REFERENCES

[1] The Evolvement of 3G Mobile, Introduction of Thrid Generation Cell Phones: http://www.planetomni.com/ARTICLES-The-Evolvement-of-3G-Mobile.shtml (Reliable data: 19 September 2007)

[2] ARPANET -- The First Internet, http://www.livinginternet.com/i/ii_arpanet.htm (Reliable data: 19 September 2007)

[3] Bryan Ford, Jacob Strauss, Chris Lesniewski-Laas, Sean Rhea, Frans Kaashoek, Robert Morris: "Persistent Personal Names for Globally Connected Mobile Devices", Massachusetts Institute of Technology;

[4] Anna Hayes, David Wilson: "Peer-to-Peer Information Sharing in a Mobile Ad Hoc Environment", University College Dublin, University of North Carolina at Charlotte

[5] Adam Kornak, Gaining Business, Jorn Teutloff, Michael Welin-Berger: "Enterprise Guide to Gaining Business Value from Mobile Technologies", Wiley Publishing

[6] Wi-Fi, From Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Wifi (Reliable data: 19 September 2007)

[7] WiFiHelps.Com, http://www.wifihelps.com/ (Reliable data: 19 September 2007)

[8] Bluetooth, From Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Bluetooth (Reliable data: 19 September 2007)

[9] Bluetooth Basics, http://www.bluetooth.com/Bluetooth/Learn/ (Reliable data: 19 September 2007)

[10] Stefano Basagni, Marco Conti, Silvia Giordano, Ivam Stojmenovich: "Mobile AD HOC Networking", IEEE Press, WILEY-INTERSCIENCE

[11] MANET (Mobile adhoc network), http://www.4ellene.net/tt/1139 (Reliable data: 19 September 2007)

[12] Information security, http://en.wikipedia.org/wiki/Information_security (Reliable data: 06 October 2007)

[13] Authentication, http://www.objs.com/survey/authent.htm (Reliable data: 19 September 2007)

[14] Matt Bishop: "Introduction to Computer Security", Prentice Hall PTR

[15] Richard Bejtlich: “The Tao of Network Security Monitoring Beyond Intrusion Detection”, Addison Wesley

[16] Jari Porras, Petri Hiirsalmi and Ari Valtaoja: "Peer-to-peer Communication Approach for a Mobile Environment", 37th Annual Hawaii International Conference on System Sciences, 2004.