ITEC 810 Project:

Security Issues in Mobile (Wireless) Ad Hoc Networking

Christopher Levari

Student ID: 41264681

Supervisor: Michael Hitchens

23rd May 2009

TABLE OF CONTENTS

1. Abstract

Technology and communication have rapidly evolved over the last decade to permeate all facets of our lives. We now use technology in our homes, our work, and at play. With the advent of these devices to fulfill our need, we've discovered that we now want these devices to communicate and share our information. Since we are a mobile society, these communications need to be wireless. This need has helped expand the field of wireless ad hoc communications (networking). Expanding in parallel is the need to secure these communications and keep our data private. Security in ad hoc networks is handled differently; therefore the security issues are different. This paper will define the main security issues in ad hoc networking using Bluetooth as a primary example of ad hoc technology. After which, this paper will offer insight into these issues and how to combat them. Finally, this paper will offer some solutions to the remaining problems.

2. Introduction

Mobile ad hoc networks are entirely self-configuring networks that may have any number of devices attached to them. Because of the constantly changing topology each device is required to keep track of these changes and update its routing information. How the device keeps or uses the routing information is dependent on which routing protocol is used. Currently, ad hoc networks are deployed in military tactical operations, disaster management and rescue situations. In order to support the delivery and routing of critical applications as well as to meet the demands of next-generation business applications, security is vital in MANET architecture. This review of sources will take you through some important topics in networking. Specifically, we will cover the current state of wireless networking . You will learn why wireless networking is important and some of the current applications of the technology. This paper will give an overview of some popular wireless technologies, show how they work and the differences between them. Once we have set that foundation, we will cover ad hoc routing protocols, the different types and explain how they work. Next, we will detail some of the issues in ad hoc networking including the security issues. To tie this project together, we will examine Bluetooth in detail as a representative ad hoc networking technology. Once we understand Bluetooth and its security issues, we will apply some of the solutions for ad hoc security to the Bluetooth specification.

3. Why is it needed?

Networking is in constant evolution. Contemporary society is always on the move and technology has to move/change along with it. Traditional networks cannot keep up, so mobile wireless devices and communication are essential. As with any evolution it comes in stages, first wireless networks had a typical infrastructure topology. This solved some of the issues wired networks had. Wired networks had one major drawback: wires. Devices were restricted by the wires. Consumers and businesses alike had to spend money and time to wire houses and offices to facilitate networking. If changes to the topology were needed these wires were wasted.

Scalability was also greatly affected, making the process slow and expensive. Wireless networks helped enable cheaper scalability, as new devices could easily be added. This still didn't solve the mobility issue. Wireless devices are still reliant on wireless routers and access points. If a device was out of range of the access point, networking and communications ceased. The next phase in the evolution is truly mobile networking. Enter: Mobile Ad Hoc Networks.

4. Definition of Mobile Ad Hoc Networks (MANETs)

A mobile ad hoc network is a system of wireless mobile devices (nodes) that can freely and dynamically self-organize in arbitrary and temporary network topologies without the need of a wired backbone or a centralized administration. People and devices can be seamlessly internetworked in areas without any preexisting communication infrastructure or when the use of such infrastructure requires wireless extension. (Aggelou, 2005).

Now that we know what MANETs are, we can see this gives devices flexibility and mobility. This opens the door to a myriad of different devices communicating; PDAs, cell phones, laptops, and any other network enabled devices. Since we now have so many different types of devices sharing information we need a common language/ communication technology to easily share data. Next, we will cover some of the common wireless technologies used in ad hoc networking.

5. Routing Protocols

Ad hoc routing protocols are just like those used in standard infrastructure routing. The difference is what device contains and mediates routing information. In infrastructure architecture a router or access point handles data routing. In ad hoc architecture each device acts as a router and is only responsible for that routing information for as long as that device is part of the network. These routing protocols fall into three main categories: Proactive, Reactive, and Flow Control. We will look at these categories and examples of each.

5.1. Proactive

Proactive routing protocols, (also called table driven protocols) maintain fresh lists of destinations and their routes by periodically distributing routing tables throughout the network. (Ilyas, 2003). These tables are created by flooding the network with routing information requests until a table is created. The table can contain one or both of the following information:

Distance Vector (DV)- How many hops a given device is away from another device/destination. For example, Device A is three hops from Device D.

Link State (LS)- Flooding the network with queries to find out whether a neighboring node is active.

These protocols have an advantage, in that the sender has an immediate path to the receiving node. This allows data to be transmitted quickly without waiting for a path to the destination to

be plotted. The main disadvantage to this type is that the routing tables have to be stored and updated on each device in the network. This ultimately will eat up storage space on the device. This a problem as current mobile/wireless devices are getting smaller and smaller, so storage becomes a premium.

5.1.1. Wireless Routing Protocol

The Wireless Routing Protocol (WRP) calculates the shortest path to the Nodes using the length (hop count) and the second to last hop to each destination. (Aggelou, 2005) To calculate the routes, WRP uses four different tables: Distance Table, Routing Table, Link-Cost Table, and a Message Retransmission List (MRL).

Distance Table - Node (n), for this example, Node A contains a table listing Node A's neighbors and the neighbor's distance to the destination. (IIyas, 2003) (Aggelou, 2005)

Routing Table - Node (n), for this example, Node A contains a table listing the predecessor and successor Nodes on the path to the destination. (IIyas, 2003) (Aggelou, 2005)

Link Cost Table - The number of hops to a destination along a given route from a source Node. There may be many routes to a destination Node, this helps identify the shortest path. (IIyas, 2003) (Aggelou, 2005)

Message Retransmission List (MRL) - The MRL contains a list of Nodes that have not acknowledged a route update, or responded to a "Hello" message. This helps each Node identify when another Node has left the network. (IIyas, 2003) (Aggelou, 2005)

5.1.2. Optimized Link State Routing Protocol (OLSR)

OLSR is interesting in the fact that it only keeps track of link state and hop count in its tables. Each device floods the network with a "Hello" packet to determine its one and two hop count neighbors. The device then elects a set of *multipoint distribution relays* (MPRs) that are responsible for relaying the data to a given destination. This information is then flooded into the network to synchronize tables. Flooding the topology data only happens often enough to make sure that the database does not remain unsynchronized for extended periods of time.

Since OLSR has to constantly flood the network using the radio link this protocol is unsuitable for low power wireless devices, because it puts too much strain on battery life. Also of note, constant broadcasting of routing data can use up the network bandwidth.

5.1.3. Destination-Sequenced Distance-Vector Routing (DSDV)

DSDV is a table driven routing protocol that includes sequence numbers in the table as well as the traditional source, destination and hop count fields. The sequence numbers are even, if a link is active and odd if it is not. Full Routing tables are broadcast throughout the network infrequently, to save bandwidth, so incremental updates are sent more often than not. If an identical sequence number is found the entry is replaced by the one with the fastest route. This

routing protocol was created to address the infinite loop problem (Wikipedia, 2009). using the routing algorithm and sequence numbers combat this.

Infinite Loop

In the simplest version, a routing loop of size two, node A thinks that the path to some destination (call it C) is through its neighboring node, node B. At the same time, node B thinks that the path to C starts at node A. Thus, whenever traffic for C arrives at either A or B, it will loop endlessly between A and B, unless some mechanism exists to prevent that behavior. (Wikipedia, 2009).

With this type of routing protocol regular updates of the routing tables occur even when the network is idle. This takes up precious bandwidth and drain battery power to operate the wireless radio. In portable ad hoc devices bandwidth and power are at a premium, so this protocol is not the best option to use.

5.2. Reactive

To combat the storage problem with proactive type protocols, reactive protocols can be used. This type of protocol finds a route on demand by flooding the network with Route Request packets. (Ilyas, 2003) Reactive (on demand) protocols calculate a path from sender to destination before sending the actual data. Since this routing information is not stored, three main tasks have to be constantly carried out.

Path Discovery - Based on a question/answer cycle achieved by flooding the network with queries.

Path Maintenance- How long to keep the routing information before deletion. This method varies depending on the actual protocol used.

Path Deletion- Removing the routing information from the device.

The disadvantage of this type is that the constant flooding takes up bandwidth space leaving less for the actual data transmission ultimately slowing the network down. Also a key factor in using reactive protocols is that the device's radio is constantly in use reducing battery life in the device.

5.2.1. Ad Hoc On-Demand Distance Vector Routing Protocol (AODV)

This routing protocol is a reactive protocol because it only requests a route when needed. This is what meant by "on-demand". What is interesting is it doesn't require nodes to maintain routes that are not actively used. Routes are only maintained between nodes that need to communicate. (IIyas, 2003) This helps keep the route tables smaller. Next we will go through the steps of the protocol. When Node A wants to send data to Node B, Node A checks its route table for a route to Node B, if it does it sends the data to the next hop in the route. This process is repeated until the data reaches the destination. If the Node does not have a route to the destination, route discovery is initiated.

Route Discovery: When Node A needs route information it floods the network with a Route Request Message (RREQ). The RREQ packet contains the following: Source address, destination address, sequence number (keeps track of repeat messages), and broadcast ID. After flooding the network, Node A sets a timer to wait for a reply. When a Node receives a RREQ message it checks to see if has received a previous RREQ it checks the source address and ID pair, if it has, it discards the message. If not it sets up a reverse path pointing to the source. (Ilyas, 2003)

Route Reply: A Node that receives a RREQ message can send a Route Reply Message (RREP) if it has an entry for the destination in its route table. If it does not a new RREQ message is forwarded along to the next Node with an updated sequence number. Ultimately the destination Node may receive the RREQ message and send a RREP message with updated route information back to the intermediate Nodes until it reaches the original source Node. Meanwhile, all the intermediate Nodes also update their route tables to include this information since this is a useable route. (Ilyas, 2003)

Route Error: If a Node leaves the network during communication a Route Error (RERR) Message is sent by the previous node before the break back to the source Node and route discovery is initiated again. Also, to keep an updated list of neighboring Nodes, each Node transmits a "Hello message" and awaits a reply. (Ilyas, 2003)

5.2.2. Dynamic Source Routing (DSR)

DSR is similar to AODV in that it uses source routing to handle the data and only creates a route when it is needed. This protocol only has 2 main phases Route Discovery and Route Maintenance. Route Reply would only be generated if the message has reached the intended destination node. If the source node does not have a route to the destination it initiates a Route Request packet. This Route Request is flooded throughout the network. Each node that receives a Route Request packet, rebroadcasts the packet to its neighbors, provided that the node is not the destination node and that the packet's time to live (TTL) counter has not expired. A Route Request has a sequence number that is created by the source node and the path it has traveled. A node that receives a Route Request packet, checks the sequence number on the packet before forwarding it. The packet is forwarded only if there is no duplicate Route Request. The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions of the same Route Request by an intermediate node (just like DSDV) that receives it through multiple paths. Therefore, all nodes except the destination forward a Route Request packet during the route construction phase. After a destination node receives the first Route Request packet, it replies to the source node through the reverse path the Route Request packet has traveled.

Some of the benefits of this protocol are the same as any reactive protocol. It only creates a route when needed therefore storage eating tables are not necessary. DSR is unique in that it does not require periodic "Hello" messages to be sent out as it does not track link state. The major drawback is the time needed to create the new route slows down the transmission of data.

5.2.3. Dynamic MANET On Demand Routing (DYMO)

In this protocol determining routes is a simple 2 step process. This first step is to send a Route Request (RREQ) to a specific destination. As the RREQ passes through the intermediate nodes the ID of those nodes gets added into the Route Request. By doing this, the destination node can immediately send a Route Reply back to the source because the route has already been recorded in the RREQ. On its way back to the source, an RREP message can simply backtrack the way the RREQ message took and simultaneously allow all hosts it passes to record a complementary route back to where it came from. (Wikipedia, 2008). After these routes have been recorded communication of the actual data can occur.

With this protocol you have the benefit of saving storage space because tables are not maintained or created. On top of this, it also saves bandwidth because there is no constant broadcast of hello messages or polling for link state.

5.3. Flow Control

This type of protocols finds a route on demand by following present flows. (Aggelou, 2005) This means the route tables only get populated with the most popular routes. For example, Device A is sending data to Device E, but trends say Device A never sends to Device C, Device C will not be polled for routing information. The advantage is the routing tables are shorter and data transmission is faster. The disadvantage is that it can take a long time to generate tables if new routes are needed.

5.3.1. Multipath On-Demand Routing Protocol

Multipath On-Demand Routing Protocol (MOR) is different in that each Node keeps multiple paths to each destination. The Node keeps track of each next hop Node to a given destination Node. The Node, say Node A, sends a packet to each of the next hop nodes in sequence. If that next hop Node doesn't reply, Node A moves on to the next hop in sequence. If a Node doesn't reply then Node A knows that route is unusable. The disadvantage to this method is it takes time to build the initial table. The advantage is increased reliability because there are multiple paths to a destination. (Biagioni, Chen, 2004).

5.3.2. Multipoint Relay Distance Vector Routing Protocol (MPRDV)

MPRDV is similar to OLSR when it come to neighbor discovery. In MPRDV, each node periodically sends out "Hello" packets across the network. These packets contain information about who that node's immediate neighbors are, as well their link status. This information then gets updated in one for the four following tables:

1-hop table - This table stores information about its one-hop neighbors
2-hop table - This table stores information about its two-hop neighbors
MPR table - This table stores nodes addresses which are elected as MPRs (Multipoint Relays) by the node.
MPRS table - This table stores nodes addresses which have elected the node as MPR (Multipoint Relays).

Once these tables are populated, MPRDV uses Route Packets to perform route requests and route repairs. A Route Packet basically contains addresses of nodes for which a route should be created. This information is then updated in a table. The lifetime of this information is specified as a protocol parameter (ROUTE HOLD TIME) and is refreshed every time a data packets are transmitted along this path. Each node also maintains a Sequence Number which is incremented every time a ROUTE message needs to be sent. Including this Sequence Number in ROUTE messages is a common way to provide loop free route creation and to only consider the most recent messages.
Upon receiving data packets, Node A checks its routing table for an entry for the destination node. If such an entry exists, data packet is transmitted to the specified next hop. On the other hand, if destination is unknown, Node A broadcasts a ROUTE packet containing the address of the requested destination. Data packets are buffered until a route is created for this destination. When a Node A receives a ROUTE message, it first creates or updates a reverse route to the Source Node. Next hop is set to the node from which the ROUTE message have been received (last hop). A route's lifetime is initially set with the parameter ROUTE HOLD TIME. Then, Node A checks the message to determine if the packet contains its address. If Node A finds its address in the message, it broadcasts an empty ROUTE message (*i.e.* a ROUTE message containing no requested addresses) and then erased its address from the message. (Allard et al., 2003).

5.3.3. Lightweight Mobile Routing (LMR)

LMR is a on demand protocol (reactive) that only maintains routes that are needed. In this protocol there are two main parts: Route Discovery, and Route Maintenance. In the first part, if a node cannot find a route to the destination node is sends out a route query (QRY) packet. This packet contains a sequence number, the destination node ID, the original source node ID, and the ID of the node that currently has the packet. The node that currently has the packet sends a reply packet (RPY) back to the source. This reply packet contains the ID of the current node, and the ID of the original source node. The RPY packet travels back along the path that the original QRY packet took. So, the QRY packet gets passed along until one of the nodes knows a path to the requested destination. That node that has the answer sends a RPY packet back to the original source node. The original source node can now transmit its data to the destination. (University, 2006)
The second part, route maintenance only occurs if a node needs to reuse a previous route to a destination. A node which has lost its last link to a still active destination issues a Failure Query Packet (FQ). This packet gets passed along just like the original QRY packet, and eventually a RPY packet is sent and data communication commences.
This protocol is very streamlined, it does not unnecessarily broadcast packets which saves battery power. The only potential drawback is the time it takes to find a suitable route to the destination. However, this is not so much of a problem in small networks.

After examining the three main types of routing protocols with examples of each, we can conclude the best type of protocol for ad hoc networks. For this paper we are focusing on small devices in an ad hoc network. So we know the following things: The devices are small and

portable so battery life is a premium. Second, the devices are small so they have limited storage space. With this in mind we can cross out proactive protocols, because they take up too much storage maintaining the routing tables. Flow control based protocols are not ideal because they are essentially a hybrid of proactive and reactive protocols. Also, flow control is best suited for larger ad hoc networks where the time it takes to reach a node is noticeable. So, for the rest of this paper we will be referring to reactive routing protocols, as seem to be the best solution for our purposes.

6. Security in Ad Hoc Networks

Security in ad hoc wireless networks is just as important as in traditional wired networks; perhaps more so. Wireless networks are more vulnerable because data transmission is broadcast through the air, and interceptable by anyone within range. Attacks that intercept the transmission and just "listen" are known as passive attacks. Attacks which intercept the transmission and modify or delete the data are called active attacks.

6.1. Attacks

As I have stated above listening/eavesdropping attacks are passive attacks, an attack that fits into this category is called a disclosure attack. Exposure of confidential data exchanged, as well as critical data the nodes store is called disclosure. This critical data the nodes store could be status information (i.e. Standby or Connection as with Bluetooth), secret keys like those used for encryption, and passwords. (Mishra, 2008) An example of an active attack would be impersonation. An impersonation attack occurs when an outside (not part of the network in question) node eavesdrops on traffic and finds identification information of a node, and now the attacking node pretends to be an existing node on the network. Now the attacking node is on the network it can just listen and gather up more data, or broadcast malicious/false into the network.

6.2. Defense

There are four main categories to defending attacks on an ad hoc system or any system for that matter. Confidentiality. Integrity. Authentication. Non-Repudiation. To adequately defend a system or network a combination of these are needed. No system is 100% secure all the time; if someone wants to gain access to your network it is only a matter of time. To understand the mechanisms for security let us define the four categories.

Confidentiality - Assurance that data/information is not disclosed to an unauthorized recipient.

Integrity - Assurance that the data has not been modified.

Authentication - Assurance that the sender and receiver are who they say they are.

Non-Repudiation - Assurance that the sender and/or the receiver cannot deny sending or receiving the data.

6.3. Mechanisms of Defense

Encryption - Encryption comes in two forms: Public Key Encryption and Symmetric Encryption. In public key encryption there is a private key that is never shared and a public key that is shared. For example, Node A needs confidential communication to Node B. So Node A encrypts the data with an encryption algorithm and Node B's public key. Node B then decrypts the data with its private key to read the message. In symmetric encryption, Node A and B each have their own key and share keys before confidential communication is started. Then Node A encrypts the data sent to Node B, with Node B's key. Node B then decrypts the data with its key.

Challenge-Response/Passwords - This is used to authenticate a node. Node A receives a connection request from Node B. So Node A requests the password from Node B, and Node B replies. Node A verifies the password against the one it has stored, if it matches connection occurs. This is only useful if the integrity of the stored password has not been compromised.

Timestamps - Timestamps are added to data transmissions to verify the freshness of the message transmission. This guards against replay attacks, where the attacking node resends the data to illicit a response. Timestamps are checked against that nodes internal clock. This is only effective if all the clocks for each node in the network have been synchronized. Sequence numbers can also be used in the same way timestamps are, plus they have the added benefit of being able to tell the order of the packets, and the way they should be re-assembled.

Authentication Keys - Public key encryption can also be used to authenticate a message and provide non-repudiation. Node A encrypts/signs the data with its private key and sends the message to Node B. Node B then decrypts the message with Node A's public key. If the message can be read then Node B knows it came from Node A, and Node A cannot argue that it did not send the message.

7. Other Security Issues

The attacks and defenses we have examined above are common in any computing system not just ad hoc networks. Ad hoc networks, however, have their own unique security issues; since ad hoc networks have nodes leaving and joining on the fly. There are three main issues with security in ad hoc networks. Next, we will cover the topic and give an example solution. These issues are: Key Management, Trust Management, and Secure Routing.

7.1. Trust Management (Mishra, 2008) (Ilyas, 2003)

An important issue for ad hoc networks is an issue of trust. We need a method to tell whether a node/device has permission to join the network. Is that device trustworthy? Does it have permission to access the data over the network?

Trust management is traditionally handled in one of two ways, a hierarchical trust model, or a web of trust model. In a hierarchical trust model (See Figure 1) is very structured and lends

itself well to a client/server infrastructure. In this model a root certificate authority (CA) is used to delegate subordinate CA's that issue certificates to end users. This method does not work well in ad hoc networks because there rarely a static device in the network that could be assigned as the root CA. The next trust model is called the web of trust model (See Figure 2). In a web of trust model, the end users (devices) are responsible for issuing certificates and vouching for the security of other users/devices.

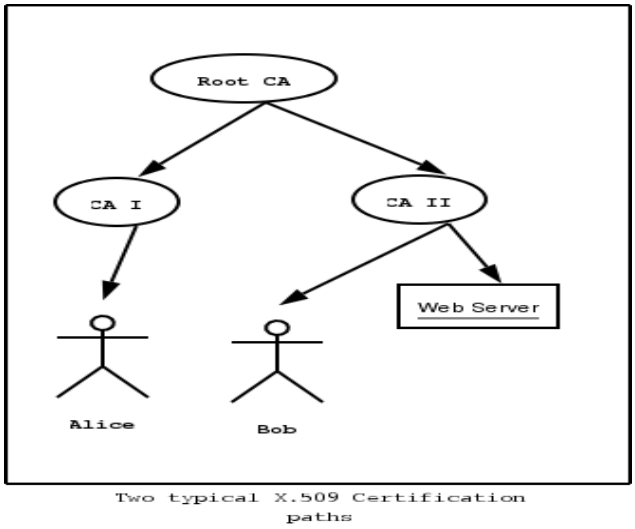

Figure 1: Hierarchical Trust Model

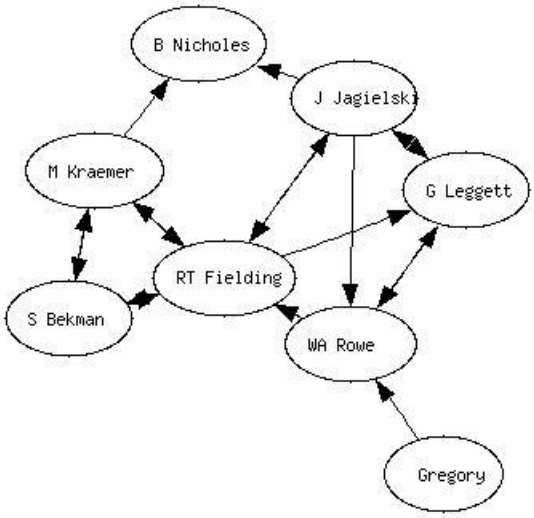

Figure 2: Web of Trust Model

(Ilyas, 2003), has an interesting approach to trust management that is more of a hybrid of the two schemes above. According to (Ilyas, 2003), one way to establish trust in ad hoc networks is to use certificates and a master/slave framework to enforce the security policy. This method lends itself to Bluetooth piconets since there is always one master in the network. The master node changes in a round robin fashion, so that at some point every node has been the master/certificate authority. To have this method work properly an Administrator has to install the certificates on each device. Another method is to establish trust as the piconet is formed. As Node A joins with Node B they share keys and say they "trust" each other. Now as another Node "C" joins with Node B and shares its key and around to the other Nodes. Once the network is complete every node has everyone else's key. This brings us to our next issue, Key Management.

7.2. Key Management

To be able to protect nodes against eavesdropping by using encryption, it is necessary that the nodes must have made a mutual agreement on a shared secret key or have exchanged public keys. (Mishra, 2008)

Therefore, it is necessary to have a secure means of creating, storing and distributing keys. In a typical infrastructure based network this is relatively easy. Infrastructure based networks have a central authority on the network that manages the keys (a server). However, when using ad hoc networks there is no centralized server on the network. So each of the individual devices

have to manage all of the keys on the network. This can be a problem especially in large networks because all of those keys would take up storage on the device. Key management also ties in neatly with Trust management. Does the network trust the device that just joined to have everyone else's key?

Mishra has an interesting option for key management that addresses the trust issue as well. He calls this solution "distributed asynchronous key management service". "Asynchronous key" is the same as Public Key Encryption that I have discussed above. In Mishra's scheme every node in the network carries the public key (K), but the private key (k) is split up evenly between the nodes. Each node can encrypt the message with their piece of the private key (k1,k2, or k3, etc) as long as a certain number of nodes (the threshold) sign the message with their key piece; when the messages are fed through the "combiner" a completed signed message emerges. This message can then be decrypted with the public key and read.

7.3. Secure Routing (Mishra, 2008) (IIyas, 2003) (Toh, 2002)

Secure routing is an important topic in ad hoc networking. Even if a Node does not read the data, does that Node have the appropriate trust and security capabilities to relay the data? This brings up the area of security aware routing. The authors we encountered during our research suggest some sort of metric is needed that is attached to each node. For example, Trust Level- high, medium, low; and Security Capabilities- a metric assigned to a node based on whether the node can process any of the security mechanisms (like the ones I stated above). A security aware protocol would take these metrics into account when calculating routes. For instance, the header on the data packet has a requirement of high trust and a security capability requirement of 5 (meaning it needs to be able to do all of the mechanisms, encryption, timestamp, etc). So when a route discovery message is sent, the nodes give a route reply with each node's security metric. The protocol would then calculate the route only sending the data through nodes with a metric of "high, 5". It is important to note that the calculated route may not be the shortest route. Also, the device must have the processing power to calculate the route in a timely fashion.

8. Ad Hoc Wireless Technologies

Ad hoc wireless technologies have expanded in the last few years to fit the needs of different applications. Some of these technologies include: 802.11 and HiperLAN, which are both used for typical network sizes (2 or more nodes). They are both suitable for laptops and desktops where running physical wires are not feasible. Bluetooth and Zigbee, which are used for small, short range networks (2-15 nodes) and suitable for portable low power devices such as smart phones and sensor nets. For the purposes of this project we will only cover 802.11 and Bluetooth.

8.1. Wireless LANs 802.11

802.11 Classes

Class	Frequency	Data Rate	Range
A	5GHz (Gigahertz)	54 Mbits (Megabits)	~120m
B	2.4GHz	11 Mbits	~140m
G	2.4HGHz	54 Mbits	~140m
N	2.4GHz or 5GHz	300 Mbits	~250m

Figure 3: (Aggelou, 2005) (Wikipedia, 2009)

In 802.11 architecture there are two separate modes: Infrastructure, and Ad Hoc. With infrastructure mode an access point sends its beacon so that each 802.11 device can join and form a network. In ad hoc mode, when two or more devices come together they form a Basic Service Set (BSS). A device by itself is an Independent Service Set (IBSS). The first to join the IBSS sets the beacon interval. Every other station has a back-off interval; the amount of time it listens for a beacon before it sends out its own beacon. In infrastructure mode the access point sets its own beacon interval and uses its own clock for synchronization. In ad hoc mode synchronization is achieved with these two steps:

Synchronization Acquisition- For a device to join an existing IBSS it will scan different frequencies to search for a beacon. If no beacon is found, the device may initialize its own IBSS and send a beacon.

Synchronization Maintenance- Once joined to a network a device will send out a beacon to check for synchronization.

8.2. Bluetooth

We will focus on Bluetooth technology instead of some of the others (i.e. 802.11 WiFi) because this is where technology is headed. Devices are more and more portable as the years go by. Portability is a key benefit of ad hoc networking, devices can join or leave a network on a whim. So, as you can see it is beneficial for portable devices to be ad hoc. The problem with portable devices is the lack of static power, so they run off of batteries. Bluetooth is designed to be low power, so it is ideal for portable devices.

Bluetooth Classes

Class	Data Rate	Range
Class 1	1 -3 Mbits (Megabits)	~100m
Class 2	1-3 Mbits	~10m
Class 3	1-3 Mbits	~1m

Figure 4:(Aggelou 2005) (Ilyas, 2003)

Bluetooth was created in 1994 by Ericsson as a replacement for short range ad hoc networks. (IIyas, 2003) These small short range networks are called piconets. In a Bluetooth piconet there are two types of nodes or devices attached to the network. These nodes are either a master or a slave; the piconet has one master and up to seven slaves. Two or more piconets create a scatternet. In order to connect piconets, a slave node in one network is shared between a second network and becomes the master in that second network. All communication to the different nodes only goes through the master. Hearing this method, it sounds more like infrastructure architecture than ad hoc architecture. Bluetooth is more ad hoc than not, because the protocol is designed to allow nodes to join or leave the piconet with ease. The master node is nominated in round robin fashion based on a timer. If the master node leaves the network another node is nominated as the master node.

8.2.1. Creating a Piconet

A Bluetooth node is in either of two states: Standby or Connection. Standby mode means the device is waiting for a connection and not part of a piconet. Connection mode means simply that the device is connected to a piconet. To form a piconet the master transmits an ID packet over 32 of the 79 channels. Devices in the Standby state periodically scan for this packet. If it hears it, the device sends its address and timing info to the master. The device then waits for the master to page it. When the master is satisfied that it has identified all the devices in its range it starts to form the piconet. It pages each device with its own device access code (DAC) using a frequency hopping sequence based on the slaves address. When the slave hears this it sends a confirmation packet. On the next slot the master sends the slave the master DAC. The slave then enters the Connection state. The master does this for all the slaves in the piconet then it enters the Connection state itself. (Dunne, Roche, O'Loghlin, Rhatigan, 2009)

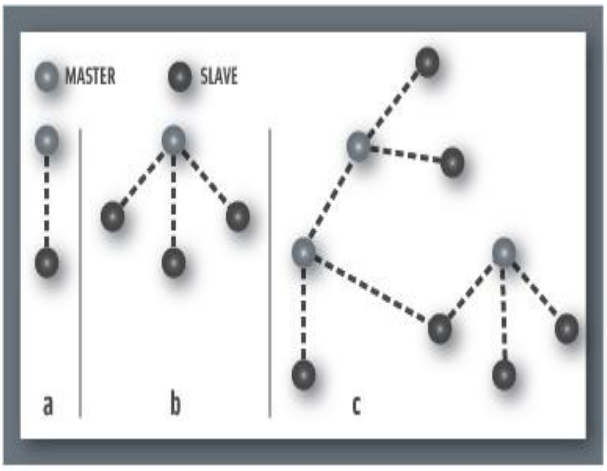

Figure 5: Piconet and Scatternet

Part A shows a typical paring of two devices. Part B shows a piconet and Part C shows a scatternet.

8.2.2. Bluetooth Transmission Methods

Once the piconet is formed it is essential to define the methods of data transmission. With the Bluetooth protocol there are two methods to achieve data transmission:

Synchronous Connection Oriented (SCO)- This transmission is a circuit switched connection used for traffic that is delay sensitive such as audio streaming. (Aggelou 2003) The drawback to this is

that the Bluetooth radio is constantly transmitting to keep the connection. This type shortens battery life quicker than Asynchronous Connection Less.

Asynchronous Connection Less (ACL)- This transmission type is a packet switched connection between devices. The radio is only used when a data packet needs to transferred. This configuration help conserve battery life. (Aggelou, 2003)

8.2.3. The Bluetooth Data Packet

In Bluetooth technology, data is transmitted in the form of packets. This section will describe the main data packet and its components. The common Bluetooth packet is made up of three main components: Access Code, Header, and Payload. (See Figure 6).

Access Code	**Header**	**Payload**
72 bits	54 bits	0 - 2745 bits

Figure 6: A Bluetooth Data Packet

This packet can be deconstructed further, by breaking down the individual segments. First, we will examine the Access Code segment. The Access Code is used to detect a packet, and identify the Master device the data is headed to, or coming from. The Access Code segment can be seen in Figure 7.

Preamble	**Synchronization Word**	**Trailer**
4 bits	64 bits	4 bits

Figure 7: Expanded view of the Access Code Segment

In this segment we see 3 parts, the preamble, the synchronization word, and the trailer. The preamble and the trailer act as padding to flag that tells the device the beginning and end of important data. This important data is called the synchronization "word". This word synchronizes the communicating devices with regards to noise in the radio signal.

Types of Access Codes (Hole, 2009)

Channel Access Code (CAC) - derived from Master's LAP and is used by all devices in a piconet during data exchange.

Device Access Code (DAC) - derived from a specific device's LAP. It is used when paging a specific device and by that device in Page Scan while listening for paging messages to itself.

General Inquiry Access Code (GIAC) - used by all devices during the inquiry procedures.

Dedicated Inquiry Access Code (DIAC) - range of reserved codes for inquiry procedures between specific devices.

The segment we will examine next is the packet header. In the figure below you can see the expanded view of the header.

LT_ADDR	Packet Type	Flow	ARQN	SEQN	Header Error Check (HEC)
3 bits	4 bits	1 bit	1 bit	1 bit	8bits

Figure 8: Expanded View of the Header Segment

LT ADDR - Master assigns Logical Transport Address (LT ADDR) to Slave. 3-bit field for up to 7 Slaves. This ID field shows the intended recipient in the piconet.

Flow - Flag asserted when device is unable to receive any more data due to full receiver buffer.

Packet Type - There are 6 packet types, ACL, SCO, Null Packet (used for flow control), Poll Packet, ID Packet (contains just the access code), and a FHS Packet (Frequency Hopping Synchronization). The FHS packet is used to synchronize device clocks.

ARQN and SEQN - SEQN toggled each time new packet with CRC is transmitted, ACK represented by ARQN=1 and NAK by ARQN=0

Header Error Check (HEC) - 8-bit CRC. Used for error checking.

The final segment in the Bluetooth packet is the payload segment. In the payload segment there is a header which contains the length of the payload data, the actual payload data, and the CRC code which checks for errors.

9. 802.11 vs. Bluetooth

These two wireless technologies are perfectly suited for ad hoc wireless/mobile communications. The difference is the size. The size/distance between nodes, the size of the device, and the size of the network. Bluetooth is suited for short distances (10m) between devices. 802.11 can handle longer distances (100m) between devices. As far as size of the devices, Bluetooth takes less power to function than 802.11. So, Bluetooth is best suited for small low power devices such as mobile phones, and PDAs. 802.11 requires more power to function so that it is best suited for laptops and Ultra Mobile Personal Computers (UMPC). Now that we understand ad hoc networks and some of the principles behind wireless technologies, we will focus solely on Bluetooth and examine Bluetooth security. (Aggelou, 2005) (Ilyas, 2003) (Toh, 2002)

10. Bluetooth Security

The Bluetooth (IEEE 802.15.1) specification has mechanisms built-in to handle authentication and encryption. In order to facilitate this, each device is given a "Bluetooth Device Address

(BDA*)" at manufacture time. This 48 bit ID is akin to a network card's MAC address. This unique hexadecimal number identifies the manufacturer and a unique ID for each device. The BDA is part of the basis for key generation.

10.1 Bluetooth Attacks

Bluetooth technology is susceptible to any of the attacks that affect any normal network. Stated below are some of the other attacks that specifically relate to Bluetooth technology.

Bluesnarfing . Bluesnarfing attacks involve a hacker covertly gaining access to your Bluetooth-enabled device for the purpose of retrieving information, including addresses, calendar information or even the device's International Mobile Equipment Identity. With the IMEI, a hacker could route your incoming calls to his cell phone. (Soto, 2005).
In order to combat this Firmware updates have been released. In addition, placing your phone in a non-discoverable mode makes it harder on the attacker.

Bluebugging. Bluebugging means hacking into a Bluetooth device and using the commands of that device without notifying or alerting the user.A hacker could eavesdrop on phone conversations, place phone calls, send and receive text messages, and even connect to the Internet. (Soto, 2005).
This type of attack is lessened by keeping up with device Firmware updates.

Bluejacking. Bluetooth devices have the ability to send so-called wireless business cards. Bluejacking is sending anonymous business cards with offensive messages but doesn't put data in jeopardy. (Soto, 2005).
To avoid this attack, put your phone on non-discoverable mode.

Denial of Service. Bluetooth DoS attacks occur when an attacker uses his Bluetooth device to repeatedly request pairing with the victim's device. (Soto, 2005).In a normal network DoS attacks can be devastating, causing services/servers to crash. However, with Bluetooth it is more of an inconvenience, since no information can be transferred, copied or attained by the attacker.

10.2 Keys

The Bluetooth standard has two categories of keys: Link keys and Encryption keys (Used for end to end communication). The link keys are used for just that, creating a link between two or more devices. Below are the link keys followed lastly by the encryption key.

Unit key, K_A, is created the first time device A is used. The Unit key is created by combining a 128 bit random number and the 48 bit BDA into an algorithm to create the 128 bit Unit key. This key is often used when device has little memory, instead of using/creating K_{AB}.

Combination key, K_{AB}, The first step to create this key is each device (A,B) create their respective Unit keys and then they exchange their keys and the random number that was generated. Each device combines the Unit keys of A and B with the two random numbers to

create K_{AB}. This key is only used for communication between A and B. If device A wants to communicate and authenticate to C, a new combination key K_{AC} has to be created and stored. This key (128 bit) is generated for each pair of devices and is used when more security is needed. This requires more memory, since device has to store one combination key for each connection it has.

Master key, K_{master}, is used when the master device wants to transmit to several devices at ones. It over rides the current link key only for one session. The Master key (128 bit) is created by feeding two random numbers into the same algorithm used to create the initialization key. The algorithm can be found at (Standards, p. 458).

Initialization key, K_{init}, is used in the initialization process. This key protects initialization parameters when they are transmitted. This key is formed from a random number, a passkey or PIN code, and the BDA of the initiating device.

Encryption key, K_E, Encryption key is derived from the current link key. Each time encryption is needed the encryption key will be automatically changed. The purpose of separating the authentication key and encryption key is to facilitate the use of a shorter encryption key without weakening the strength of the authentication procedure. (Standards, 2005)

10.3. Authentication and Authorization

Bluetooth has three different levels of security, which are called service levels. These levels are:

Service Level 1 - Requires authorization and authentication. Automatic access is granted only to trusted devices; untrusted devices need manual authorization.
Service Level 2 - Requires authentication only; authorization is not necessary. Access to an application is allowed only after an authentication procedure.
Service Level 3 - Open to all devices, with no authentication required. Access is granted automatically. (Standards, 2005).

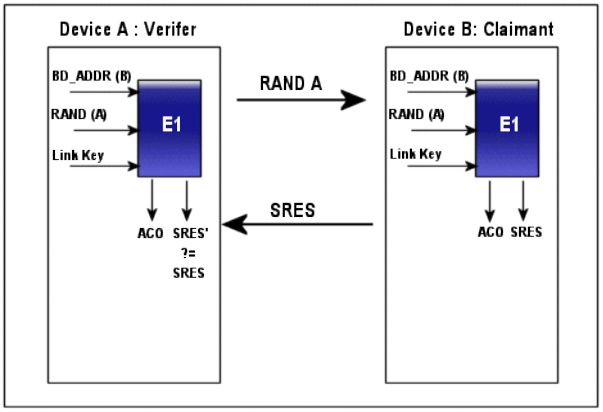

Figure 9: Challenge Response Mechanism

Note that BD_ADDR is the same as BDA, and SRES stands for the response.

Authentication is handled by a challenge and response mechanism (See Figure 9 above). In this mechanism, Device A sends a random number to Device B. Device B then calculates a response by feeding the random number from A, its BDA*, and the key it has into an algorithm. Device B then sends the calculated response back to A. Device A then calculates the response by feeding that same random number, the BDA of B, and the key into the algorithm. If the two responses match, Device B is authenticated and they both share the same key. If they do not match, Device B must wait a given time interval before trying again. This time interval increases exponentially with each attempt.
Bluetooth handles authorization very simply. If a device is authorized to access the content then it is known as "trusted", if not, then it is known as "untrusted". When a device is trusted it has been authenticated and its link key is stored.

10.4. Data Encryption

Bluetooth data encryption is handled by a stream cipher. This cipher first calculates a "payload key" this key is generated using K_E, the BDA, the synchronization clock value, and a random number. After that the payload key is fed into the key stream with the data to be transmitted and out comes the encrypted text/cipher text. The receiving device then decrypts the data by feeding it back through the stream cipher (See Figure 10).

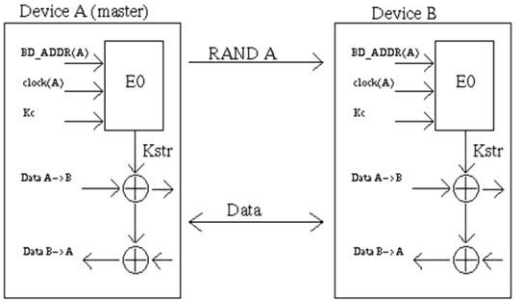

Figure 10: Stream cipher for end to end encryption

Note that BD_ADDR is the same as BDA.

11. Implementation and Observation

Now that now that we have examined the technical details of Bluetooth and understood them, let's look at how it does or does not address the three main security issues of ad hoc networking.

11.1 Bluetooth Implementation: Key Management

Bluetooth does handle key management efficiently since the keys for each device are stored only on a semi-permanent basis. Although this does eliminate the problem of constantly authenticating for every communication attempt. The semi permanency of the keys helps alleviate the burden of storing all keys for every device even if that device never joins the network again. The problems start when first "pairing" the device. Bluetooth pairing is

equivalent to joining the network. The new device pairs with the master node in the piconet and the initialization key is created once a PIN is shared. How does the PIN get shared? Most likely from a company Network Administrator. The device PIN has to be entered by the user or the PIN is installed on the device by the Administrator. This method is not "truly" ad hoc, as centralized management of the PIN/"key" is handled by the Admin. The other issue I see, is a big security issue. If the device uses its Unit key for communication it becomes a security threat. The Unit key is created for the life of the device, this means the same key will be used over and over for communication. The more times a key is used, the more it has to become compromised and used by a hacker. Also, if Device A is communicating with multiple other devices, for example Device B and C, then Device B could eavesdrop on Device C because they are both using Device A's Unit Key.

11.2 Bluetooth Implementation: Trust

The Bluetooth specification does handle the issue of trust but only on a very basic level. Either a device is trusted or it is not, there are no provisions for levels of trust such as top secret, secret, classified, or non-classified.

Bluetooth gets around the issue of trust by sharing a secret PIN. If the PIN is known and entered into the device, the device is trusted. This is a weak method of trust because the PIN could be compromised. If this happens then the proof of trust is void. This method also does not address the trust level of the user. Any user could randomly use a device where the PIN has already been entered. Therefore some other layer of authentication which establishes trust needs to be implemented.

11.3 Bluetooth Implementation: Secure Routing

The Bluetooth protocol has no provisions written in for secure routing. This needs to be rectified to make it more robust and suitable for secure business applications. It has been suggested by (Varadharajan, et al, 2009) to secure AODV by signing the data packets with a key. This method would work nicely with Bluetooth's Unit key for each device. Combining this with the capabilities model suggested in section 7.3 would go a long way to applying secure routing to the Bluetooth protocol. In this next section we will combine our observations with expert's suggestions to improve Bluetooth ad hoc networking.

12. Proposed Solution

First, we need to re-examine what we've covered up to this point. The ad hoc networking literature this paper covers states that the main security issues are:

- *How do we manage keys in an ad hoc environment, and who is responsible for those keys?*
- *How do we prove a device is "trusted" and allowed to join our network? Once they have joined our network, how do we handle different levels of trust?*
- *How do we insure that our data is secure while en-route to its destination?*

Bluetooth answers some of these questions in a few different ways. Bluetooth handles our first problem quite well. A device pairs with the Master node in the piconet by entering and sharing a predefined PIN code. This PIN is factored in with other unique data to create security keys. Once a device is paired it is free to leave or join the piconet in an ad hoc fashion. These security keys are stored by each device in the piconet. Since a piconet has up to 7 devices (1 Master, 6 Slaves), each device needs to store up to 23 different keys to make up the number of possible different pairings. This takes care of the combination keys, now we need to add in each device's Unit Key (1 per device), and potentially 1 Master key per device (Since the Master role rotates to a new node after a predefined interval). That is now 25 total keys. Lastly, we add in the encryption keys, that's 23 different permutations of pairs. That gives us a total of 48 keys each device has to potentially store. As stated in section 10.2 the amount of keys can be lessened by only using the Unit key instead of the combination keys to free up storage space. This lessens the number of required keys to 25. In our proposed solution, however, will require the use of combination keys to increase security. With ever increasing storage in new devices, storage space will not be as much of an issue. The management of these 48 keys is agreed upon during the initialization phase of the paring of two devices. In this management, they agree on key length, and the time frame keys are valid.

In our second problem is where we see Bluetooth is lacking a bit. Bluetooth assigns trust to a device if the PIN has successfully been entered during the pairing process. Handling trust this way is binary, either they're trusted or they're not. Bluetooth does not address multi-level trust, this paper proposes two solutions to this dilemma. First, you could rewrite the Bluetooth specification to accept multiple PINs per device pair. Each PIN would be the equivalent of a different trust level. This would then be used in multiple keys for different trust levels, giving the device decent access control to files and services. The drawback to this is each device would be responsible for storing even more keys. On top of that, the Administrator would then be responsible for handing out even more PINs to the users in a network. With these drawbacks in this paper suggests the second solution to the problem. In the second solution, we suggest an extension to the current Bluetooth packet. In this new packet would be a 3 bit field that labels trust. The 3 bit allows for up to 8 levels/degrees of trust. This trust level would be assigned by the administrator of the network. This field will remain static, as it assigns trust level to the device. Our extended packet header will now look like this:

LT_ADDR 3 bits	**Device Trust Level** 3 bits	**Packet Type** 4 bits	**Flow** 1 bit	**ARQN** 1 bit	**SEQN** 1 bit	**Header Error Check (HEC)** 8bits

Figure 11: Expanded View of the Extended Header Segment

It is important to note that this only assigns trust to the device and not the user of the device. To handle user trust levels, it would have to be implemented at the application level. This implementation is outside the scope of this paper. Handling trust in 3 different ways gives the network administrator greater granularity of control. A device can join a piconet if the correct PIN is entered. Once joined with the network the user logs in at the application level giving the

user access to data at his trust level. However, the device become the limiting factor, the user can only access data on the same trust level as the device and no higher. If the user's trust level is lower than the device, the login application should deny access. This also inherently prevents a user from using someone else's device of a higher access level.

The third and final issue is securing data during routing. Bluetooth has nothing in its specification to handle this problem. Our proposed solution takes advantage of our extended header segment. It also applies the suggestion found during our research and discussed in Section 7.3. The first step to this solution is to add another field to the header segment. This field is 4 bits long and addresses the device's security capabilities. See Figure 12 below.

4 Bit Capabilities Matrix			
1	2	3	4
128 Bit Keys	256 Bit Keys	Auto - Generated PIN	Sequence Numbers

Figure 12: Capabilities Assigned to Each Bit

As you can see, each bit in the field is assigned a specific capability. If the corresponding bit has been turned on, the device has that capability. Some of the other security capabilities not mentioned (Encryption, Challenge - Response Mechanism) are available in any device because that capability is inherent in the Bluetooth Specification.

With this final solution in effect, the resulting header segment will look like this:

LT_ADDR 3 bits	**Security Capabilities** 4 bits	**Device Trust Level** 3 bits	**Packet Type** 4 bits	**Flow** 1 bit	**ARQN** 1 bit	**SEQN** 1 bit	**Header Error Check (HEC)** 8bits

Figure 13: Final Proposed Extended Header Segment

These two address the multi-level trust issue as well as the secure routing issue. For example, once the device has paired with the Master node in the piconet and the user has logged in, the device is successfully on the network. Let's call the device Node C. If Node A wants to send data to Node D and we assume we are using a reactive routing protocol like AODV, but with a security aware extension. Node A sends a Route Request. In this Route Request it would contain the Source Node's Capability and Trust level. In this protocol, Nodes that meet or exceed this Capability and Trust level *and* know a route to Node D would respond with a Route Reply. In this example, Node C meets those requirements and the data is relayed through to the destination. Every other portion of the protocol remains the same, including "no routes available" scenarios.

13. Conclusion

In this project we see that mobile/wireless ad hoc networks are advantageous and allow devices flexibility and mobility. Bluetooth and 802.11 wireless LANs (Local Area Networks) are two of the technologies that make this happen. We have examined how the network communicates and correctly routes information to the nodes. We have explained based on our personal knowledge the need for security in ad hoc networks and outlined some of the security basics. These security basics are what are used to secure ad hoc networks. Our research into ad hoc networking have uncovered some important security issues. They are: Key Management, Trust Management, and Secure Routing. To understand these security issues we analyzed Bluetooth technology and dissected how these issues apply to this ad hoc networking technology. In the final section we applied our proposed solutions to the given problems. By reading this paper you will understand the principles of these issues and see that our solutions could solve the problems if applied to a future release of the Bluetooth specification.

SUMMARY

Introduction

Mobile ad hoc networks are entirely self-configuring networks that may have any number of devices attached to them. Currently, ad hoc networks are deployed in military tactical operations, disaster management and rescue situations. In order to support the delivery and routing of critical applications as well as to meet the demands of next-generation business applications, security is vital in MANET architecture. This paper will take you through some important topics in ad hoc networking. Specifically, we will cover the current state of wireless networking . You will learn why wireless networking is important and some of the current applications of the technology. Once we have set that foundation we will cover ad hoc routing protocols, the different types and which is best. Next, we will detail some of the issues in ad hoc networking including the security issues. To tie this project together, we will examine Bluetooth in detail as a representative ad hoc networking technology. Once we understand Bluetooth and its security issues, we will apply some of the solutions for ad hoc security to the Bluetooth specification.

First off, MANETs are: "A mobile ad hoc network is a system of wireless mobile devices (nodes) that can freely and dynamically self-organize in arbitrary and temporary network topologies without the need of a wired backbone or a centralized administration. People and devices can be seamlessly internetworked in areas without any preexisting communication infrastructure or when the use of such infrastructure requires wireless extension." (Aggelou, 2005).

Routing

Ad hoc routing protocols are just like those used in standard infrastructure routing. The difference is what device contains and mediates routing information. In infrastructure architecture a router or access point handles data routing. In ad hoc architecture each device acts as a router and is only responsible for that routing information for as long as that device is part of the network. These routing protocols fall into three main categories: Proactive, Reactive, and Flow Control. Proactive and most Flow Control based routing protocols use tables to keep track of routing information. This is unacceptable because ad hoc devices are small adn portable with limited storage capabilities. Also these types of protocols require regular updates of the routing tables occur even when the network is idle. This takes up precious bandwidth and drain battery power to operate the wireless radio. In portable ad hoc devices bandwidth and power are at a premium, so this protocol is not the best option to use. To combat the storage problem with proactive type protocols, reactive protocols can be used. This type of protocol finds a route on demand by flooding the network with Route Request packets. (Ilyas, 2003) Reactive (on demand) protocols calculate a path from sender to destination before sending the actual data.

Ad Hoc Technologies

Ad hoc networking has 2 prevalent technologies: 802.11 WiFi and 802.11.15 Bluetooth. We will focus on Bluetooth technology instead of some of the others because this is where technology is headed. Devices are more and more portable as the years go by. Portability is a key benefit of ad hoc networking, devices can join or leave a network on a whim. So, as you can see it is beneficial for portable devices to be ad hoc. The problem with portable devices is the lack of static power, so they run off of batteries. Bluetooth is designed to be low power, so it is ideal for portable devices.

Bluetooth networks are small mobile networks call piconets. These piconets can have up to 7 devices networked together. In Bluetooth technology, data is transmitted in the form of packets. This section will describe the main data packet and its components. The common Bluetooth packet is made up of three main components: Access Code, Header, and Payload. (See Figure 1).

Access Code	**Header**	**Payload**
72 bits	54 bits	0 - 2745 bits

Figure 1: A Bluetooth Data Packet

This packet can be deconstructed further, by breaking down the individual segments. First, we will examine the Access Code segment. The Access Code is used to detect a packet, and identify the Master device the data is headed to, or coming from. The Access Code segment can be seen in Figure 2.

Preamble	**Synchronization Word**	**Trailer**
4 bits	64 bits	4 bits

Figure 2: Expanded view of the Access Code Segment

In this segment we see 3 parts, the preamble, the synchronization word, and the trailer. The preamble and the trailer act as padding to flag that tells the device the beginning and end of important data. This important data is called the synchronization "word". This word synchronizes the communicating devices with regards to noise in the radio signal.

The segment we will examine next is the packet header. In the figure below you can see the expanded view of the header.

LT_ADDR	**Packet Type**	**Flow**	**ARQN**	**SEQN**	**Header Error Check (HEC)**
3 bits	4 bits	1 bit	1 bit	1 bit	8bits

Figure 3: Expanded View of the Header Segment

LT ADDR - Master assigns Logical Transport Address (LT ADDR) to Slave. 3-bit field for up to 7 Slaves. This ID field shows the intended recipient in the piconet.

Flow - Flag asserted when device is unable to receive any more data due to full receiver buffer.

Packet Type - There are 6 packet types, ACL, SCO, Null Packet (used for flow control), Poll Packet, ID Packet (contains just the access code), and a FHS Packet (Frequency Hopping Synchronization). The FHS packet is used to synchronize device clocks.

ARQN and SEQN - SEQN toggled each time new packet with CRC is transmitted, ACK represented by ARQN=1 and NAK by ARQN=0

Header Error Check (HEC) - 8-bit CRC. Used for error checking.

The final segment in the Bluetooth packet is the payload segment. In the payload segment there is a header which contains the length of the payload data, the actual payload data, and the CRC code which checks for errors.

Bluetooth Security

Bluetooth has some security built in. It handles security with the use of security keys, either for encryption or identification. The Bluetooth standard has two categories of keys: Link keys and Encryption keys (Used for end to end communication). The link keys are used for just that, creating a link between two or more devices. Below are the link keys followed lastly by the encryption key.

Unit key, K_A, is created the first time device A is used. The Unit key is created by combining a 128 bit random number and the 48 bit BDA into an algorithm to create the 128 bit Unit key. This key is often used when device has little memory, instead of using/creating K_{AB}.

Combination key, K_{AB}, The first step to create this key is each device (A,B) create their respective Unit keys and then they exchange their keys and the random number that was generated. Each device combines the Unit keys of A and B with the two random numbers to create K_{AB}. This key is only used for communication between A and B. If device A wants to communicate and authenticate to C, a new combination key K_{AC} has to be created and stored. This key (128 bit) is generated for each pair of devices and is used when more security is needed. This requires more memory, since device has to store one combination key for each connection it has.

Master key, K_{master}, is used when the master device wants to transmit to several devices at ones. It over rides the current link key only for one session. The Master key (128 bit) is created by feeding two random numbers into the same algorithm used to create the initialization key. The algorithm can be found at (Standards, p. 458).

Initialization key, K_{init}, is used in the initialization process. This key protects initialization parameters when they are transmitted. This key is formed from a random number, a passkey or PIN code, and the BDA of the initiating device.

Encryption key, K_E, Encryption key is derived from the current link key. Each time encryption is needed the encryption key will be automatically changed. The purpose of separating the authentication key and encryption key is to facilitate the use of a shorter encryption key without weakening the strength of the authentication procedure. (Standards, 2005).
Authentication is handled by a challenge and response mechanism. In this mechanism, Device A sends a random number to Device B. Device B then calculates a response by feeding the random number from A, its BDA^*, and the key it has into an algorithm. Device B then sends the calculated response back to A. Device A then calculates the response by feeding that same random number, the BDA of B, and the key into the algorithm. If the two responses match, Device B is authenticated and they both share the same key. If they do not match, Device B must wait a given time interval before trying again. This time interval increases exponentially with each attempt.
Bluetooth handles authorization very simply. If a device is authorized to access the content then it is known as "trusted", if not, then it is known as "untrusted". When a device is trusted it has been authenticated and its link key is stored.

Bluetooth data encryption is handled by a stream cipher. This cipher first calculates a "payload key" this key is generated using K_E, the BDA, the synchronization clock value, and a random number. After that the payload key is fed into the key stream with the data to be transmitted and out comes the encrypted text/cipher text. The receiving device then decrypts the data by feeding it back through the stream cipher.

Ad Hoc Security Issues

Now that we understand Bluetooth technology and how it handles security, we can move on to the main security issues in ad hoc networking, how they relate to Bluetooth, and ultimately our proposed solutions. The three main security issues in ad hoc networking are:

- *How do we manage keys in an ad hoc environment, and who is responsible for those keys?*
- *How do we prove a device is "trusted" and allowed to join our network? Once they have joined our network, how do we handle different levels of trust?*
- *How do we insure that our data is secure while en-route to its destination?*

Proposed Solution

Bluetooth answers some of these questions in a few different ways. Bluetooth handles our first problem quite well. A device pairs with the Master node in the piconet by entering and sharing a predefined PIN code. This PIN is factored in with other unique data to create security keys. Once a device is paired it is free to leave or join the piconet in an ad hoc fashion. These security keys are stored by each device in the piconet. Since a piconet has up to 7 devices (1 Master, 6 Slaves), each device needs to store up to 23 different keys to make up the number of possible different pairings. This takes care of the combination keys, now we need to add in each device's Unit Key (1 per device), and potentially 1 Master key per device (Since the Master role rotates

to a new node after a predefined interval). That is now 25 total keys. Lastly, we add in the encryption keys, that's 23 different permutations of pairs. That gives us a total of 48 keys each device has to potentially store. As stated in section 10.2 the amount of keys can be lessened by only using the Unit key instead of the combination keys to free up storage space. This lessens the number of required keys to 25. In our proposed solution, however, will require the use of combination keys to increase security. With ever increasing storage in new devices, storage space will not be as much of an issue. The management of these 48 keys is agreed upon during the initialization phase of the paring of two devices. In this management, they agree on key length, and the time frame keys are valid.

In our second problem is where we see Bluetooth is lacking a bit. Bluetooth assigns trust to a device if the PIN has successfully been entered during the pairing process. Handling trust this way is binary, either they're trusted or they're not. Bluetooth does not address multi-level trust, this paper proposes two solutions to this dilemma. First, you could rewrite the Bluetooth specification to accept multiple PINs per device pair. Each PIN would be the equivalent of a different trust level. This would then be used in multiple keys for different trust levels, giving the device decent access control to files and services. The drawback to this is each device would be responsible for storing even more keys. On top of that, the Administrator would then be responsible for handing out even more PINs to the users in a network. With these drawbacks in this paper suggests the second solution to the problem. In the second solution, we suggest an extension to the current Bluetooth packet. In this new packet would be a 3 bit field that labels trust. The 3 bit allows for up to 8 levels/degrees of trust. This trust level would be assigned by the administrator of the network. This field will remain static, as it assigns trust level to the device. Our extended packet header will now look like this:

LT_ADDR 3 bits	**Device Trust Level** 3 bits	**Packet Type** 4 bits	**Flow** 1 bit	**ARQN** 1 bit	**SEQN** 1 bit	**Header Error Check (HEC)** 8bits

Figure 11: Expanded View of the Extended Header Segment

It is important to note that this only assigns trust to the device and not the user of the device. To handle user trust levels, it would have to be implemented at the application level. This implementation is outside the scope of this paper. Handling trust in 3 different ways gives the network administrator greater granularity of control. A device can join a piconet if the correct PIN is entered. Once joined with the network the user logs in at the application level giving the user access to data at his trust level. However, the device become the limiting factor, the user can only access data on the same trust level as the device and no higher. If the user's trust level is lower than the device, the login application should deny access. This also inherently prevents a user from using someone else's device of a higher access level.

The third and final issue is securing data during routing. Bluetooth has nothing in its specification to handle this problem. Our proposed solution takes advantage of our extended header segment. It also applies the suggestion found during our research and discussed in

Section 7.3. The first step to this solution is to add another field to the header segment. This field is 4 bits long and addresses the device's security capabilities. See Figure 12 below.

4 Bit Capabilities Matrix			
1	2	3	4
128 Bit Keys	256 Bit Keys	Auto - Generated PIN	Sequence Numbers

Figure 12: Capabilities Assigned to Each Bit

As you can see, each bit in the field is assigned a specific capability. If the corresponding bit has been turned on, the device has that capability. Some of the other security capabilities not mentioned (Encryption, Challenge - Response Mechanism) are available in any device because that capability is inherent in the Bluetooth Specification.

With this final solution in effect, the resulting header segment will look like this:

LT_ADDR 3 bits	**Security Capabilities** 4 bits	**Device Trust Level** 3 bits	**Packet Type** 4 bits	**Flow** 1 bit	**ARQN** 1 bit	**SEQN** 1 bit	**Header Error Check (HEC)** 8bits

Figure 13: Final Proposed Extended Header Segment

These two address the multi-level trust issue as well as the secure routing issue. For example, once the device has paired with the Master node in the piconet and the user has logged in, the device is successfully on the network. Let's call the device Node C. If Node A wants to send data to Node D and we assume we are using a reactive routing protocol like AODV, but with a security aware extension. Node A sends a Route Request. In this Route Request it would contain the Source Node's Capability and Trust level. In this protocol, Nodes that meet or exceed this Capability and Trust level *and* know a route to Node D would respond with a Route Reply. In this example, Node C meets those requirements and the data is relayed through to the destination. Every other portion of the protocol remains the same.

Conclusion

In this project we see that mobile/wireless ad hoc networks are advantageous and allow devices to have flexibility and mobility. Bluetooth and 802.11 wireless LANs (Local Area Networks) are two of the technologies that make this happen. We have examined how the network communicates and correctly routes information to the nodes. We have explained based on our personal knowledge the need for security in ad hoc networks and outlined some of the security basics. These security basics are what are used to secure ad hoc networks. Our research into ad hoc networking have uncovered some important security issues. They are: Key Management, Trust Management, and Secure Routing. To understand these security issues we analyzed Bluetooth technology and dissected. how these issues apply to this ad hoc networking technology. In the final section we applied our proposed solutions to the given problems. By reading this paper you will understand the principles of these issues and see that our solutions could solve the problems if applied to a future release of the Bluetooth specification.

References

Aggelou, George. Mobile Ad Hoc Networks: From Wireless LANs to 4G Networks. McGraw Hill: New York, 2005.

Allard, G. & Jacquet, P. & Viennot, L. *Ad Hoc Routing Protocols with Multipoint Relaying.* Institut National de Recherche en Informatique et en Automatique: France, 2003.

Biagioni, E. & Chen, S. *A Reliability Layer For Ad Hoc Wireless Sensor Network Routing*. University of Hawaii: Hawaii, 2004.

Bluetooth Special Interest Group. *Bluetooth Specifications.* Bluetooth SIG: 2009. Retrieved on 19 April 2009 from http://www.bluetooth.org

Davis, Carlton R. *A Localized Trust Management Scheme for Ad hoc Networks*. McGill University: Montreal, 2009.

Duryee, Trisha. *New Bluetooth Spec Will Enable Speedy Transfers Of Photos, Videos.* moconews.net 13 April 2009. Retrieved on 30 April 2009 from http://www.moconews.net/entry/419-new-bluetooth-spec-will-enable-speedy-transfers-of-photos-videos/

Hole, Kjell Jorgen. *Bluetooth: Part 2 Baseband and Java ME.* UiB, 22 Feb. 2009.

Ilyas, Mohammad. The Handbook of Ad Hoc Wireless Networks. CRC Press: New York, 2003.

Mishra, Amitabh. Security and Quality of Service in Ad Hoc Wireless Networks. Cambridge University Press: Cambridge, 2008.

Muller, Thomas. *Bluetooth Security Architecture*. Nokia: USA, 1999.

Palo Wireless. *Bluetooth Tutorial - Packet Formats.* Palo Wireless: Australia, 2009. Retrieved on 18 May 2009 from http://www.palowireless.com/bluearticles/packets.asp.

Papadimitratos, P. & Haas, Z. *Secure Routing for Mobile Ad Hoc Networks*. Cornell University: Ithaca, 2002.

Roche, E. & Dunne, K. & O'Loghlin, D. *Bluetooth For Ad Hoc Networking*. Retrieved on 23 March 2009. from http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group3/index.html

RSA Laboratories. RSA Laboratories - 4.1.1 What is Key Management? RSA Security: USA, 2009. Retrieved on 15 May 2009 from http://www.rsa.com/rsalabs/node.asp?id=2262.

Scarfone, K. & Padgette, J. *NIST: Guide to Bluetooth Security*. NIST Special Publication: Gaithersburg, 2008.

Soto, Carlos. *A Menu of Bluetooth Attacks.* Government Computer News, 20 July 2005. Retrieved on 19 May 2009 from http://gcn.com/Articles/2005/07/20/A-menu-of-Bluetooth-attacks.aspx?Page=1.

Standards Committee. *IEEE Standard 802.15.1: Revision 2005*. IEEE: New York, 14 June 2005. (pp. 437-459)

Toh, C. K. Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall: New Jersey, 2002.

Traskback, Marjaana. *Security of Bluetooth: An Overview of Bluetooth Security*. Helsinki: 2009

University of Luxembourg. *Lightweight Mobile Routing*. University of Luxembourg: 2006. Retrieved on 20 May 2009 from http://wiki.uni.lu/secan-lab/Lightweight+Mobile+Routing.html

Varadharajan, V. & Shankaran, R. & Hitchens, M. *Securing the Ad Hoc On-demand Distance Vector Protocol.* Macquarie University: Australia, 2009.

Varadharajan, V. & Shankaran, R. & Hitchens, M. *Securing the Ad Hoc Dynamic Source Routing Protocol.* Macquarie University: Australia, 2009.

Wikipedia. *Destination-Sequenced Distance Vector Routing.* Wikipedia Foundation. 22 February 2009. Retrieved on 18 May 2009 from http://en.wikipedia.org/wiki/DSDV.

Wikipedia. Dynamic Source *Routing.* Wikipedia Foundation. 30 April 2009. Retrieved on 18 May 2009 from http://en.wikipedia.org/wiki/Dynamic_Source_Routing.

Wikipedia. L*ist of Ad Hoc Routing Protocols*. Wikipedia Foundation. 18 May 2009. Retrieved on 2 May 2009 from http://en.wikipedia.org/wiki/Ad_hoc_routing_protocol_list.

Wikipedia. *Routing Loop Problem*. Wikipedia Foundation. 16 May 2009. Retrieved on 18 May 2009 from http://en.wikipedia.org/wiki/Routing_loop_problem.

Illustrations

Figure 1 obtained at: http http://www.gnu.org/software/gnutls/manual/html_node/gnutls-x509.png

Figure 2 obtained at: http http://people.apache.org/~henkp/trust/httpd-grey.jpg

Figure 5 obtained at:http://www.easycom.com.ua/data/netlan/712162057/img/piconets1.jpg

Figure 9 obtained at:
http://www.palowireless.com/bluearticles/cc1_security1_files/image002.gif

Figure 10 obtained at:
http://www.palowireless.com/bluearticles/cc1_security1_files/image003.gif