# Context Aware and Adaptive Security for Wireless Networks

Creighton T. R. Hager

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Scott F. Midkiff, Chair
Ezra A. Brown
Luiz A. DaSilva
Nathaniel J. Davis, IV
Thomas L. Martin

November 2004
Blacksburg, Virginia

Keywords: block ciphers, analytic hierarchy process, energy efficiency, performance evaluation

# Context Aware and Adaptive Security for Wireless Networks

Creighton T. R. Hager

(ABSTRACT)

This research investigated methods to determine appropriate security protocols for specific wireless network applications. The specific problem being addressed was that there are tradeoffs between security, performance, and efficiency among current and proposed security protocols. Performance and efficiency issues are particularly important in wireless networks which tend to have constrained network capacity and connect to resource-limited nodes. Existing security protocols address problems such as authentication, availability, confidentiality, integrity, and non-repudiation. However, these protocols use resources and limit the efficient use of node resources. Thus, the overall objective of this research is to improve the efficiency of security mechanisms for wireless networks.

A methodology was constructed to satisfy this objective and is an important contribution of this research. The methodology can be used to define the relevant operational parameters of different wireless network applications, classify wireless networks into distinct categories, incorporate appropriate security protocols to a category, and analyze the security protocols through metrics. Three groups of operational parameters were created to classify wireless networks; these are equipment, network topology, and communication characteristics. The wireless network categories include, but are not limited to, fixed broadband wireless networks, wireless local area networks, mobile ad hoc networks, and small device sensor networks. The metrics in the methodology are used to measure end-to-end data throughput and delay, efficiency and overhead, power and energy consumption, and energy consumed per packet transferred.

The main advantage of this methodology is the flexibility of how constraints are considered and suitability is analyzed. This approach can identify problems from manageable categories of networks and find or create solutions for each of them. Another advantage of this methodology is that after suitable security protocols are found or created for each category, any new wireless network application that falls into an existing category may be able to use the security protocols from that category and find that they are the most suitable.

Another key contribution of this research was the implementation and evaluation of a context aware and adaptive security manager (CASM) that selects appropriate protocols in real-time. CASM was developed using the methodology as a guide. Results from a resource analysis of four encryption algorithms were utilized for the design of CASM. A feasibility study of CASM was then completed. Three different experimental scenarios were used to evaluate CASM's operation. The results and analysis of the experiments indicate that the security manager functions properly and security is provided efficiently with different user settings and environments. Three schemes were deemed the best to use for the decision module of CASM.

## Acknowledgments

I would like to express my deep felt gratitude to my advisor, Dr. Scott F. Midkiff, for his patience, advice, and continuous support. The other members of my committee, Dr. Ezra A. Brown, Dr. Luiz A. DaSilva, Dr. Nathaniel J. Davis, IV, and Dr. Thomas L. Martin have also provided valuable suggestions and constructive feedback for this work.

I would like to thank my parents, Albert and Louisiana, and my brother, Kristopher, who encouraged and supported me through all my years in school.

This research was supported in part by a National Science Foundation Integrated Graduate Education and Research Training (IGERT) grant (award DGE-9987586). This research was also supported by Microsoft Research through their "Embedded Systems Request for Proposals" program.

My colleagues and friends, who have worked with me on many projects unrelated to this dissertation, must also be noted. These include but are not limited to Todd Eschler, Michelle Gong, Takeshi Ikuma, Tao Lin, Chris Knestrick, Malcolm Mason, Dan Nash, Christian Rieser, Tom Rondeau, Michael Thompson, and Krishnaraj Varma.

# Contents

# List of Figures

# List of Tables

# Chapter 1. Introduction

## 1.1 Motivation

The use of wireless networks has become more prevalent in commercial and military applications. However, wireless technologies come with security problems. Wireless networks, like most wired networks, may have potential vulnerabilities that can be exploited by intruders. Typical attacks against these types of networks include illicit entry, eavesdropping, unauthorized resource usage, and denial of services [1-4]. Thus, security is one of the most important requirements in wireless communications.

Many protocols and mechanisms have been proposed to alleviate the concern of vulnerable wireless communications; specific examples of these protocols are discussed in the next chapter. While some of these approaches may have merit in minimizing the risk of security attacks, two major shortcomings to securing wireless networks are designing a security protocol that is too broad (meant to be used in any type of wireless communication) and designing security protocols as a patch for a vulnerable system. Most system designs for security do not consider the added processing overhead and energy consumption required for securing a wireless network. In addition, most approaches tend to be too general and may overload wireless networks that have limited or highly variable resources, such as sensor networks or wireless networks composed of handheld devices. Designing wireless networks without security in mind may lead to protocols that have many security vulnerabilities.

## 1.2 Objectives and Approach

The overall objective of this research was to improve the efficiency of security mechanisms for wireless networks. For that to happen, networks must first be divided into manageable categories or applications so that appropriate security designs for each can then be implemented. The approach developed in this work involves defining the operational parameters of different wireless network applications, classifying wireless network applications into distinct categories, applying appropriate security protocols to a category, and analyzing the applied security protocols using suitable metrics.

The first goal of this research was to develop a methodology that can be used to determine the most suitable security protocol for a specific wireless network application. A wireless network application can be described by its operational constraints and characteristics. By using the methodology, we can limit the available security solutions to only the most appropriate ones.

The second goal was to offer a method for wireless network security to adapt based on its context in certain environments. Suitable security algorithms are selected for different scenarios. The purpose is to minimize the resource impact of a security algorithm while maintaining a selected level of security. Additionally, the adaptive method should allow flexibility through user preferences.

This research has led to the following contributions in the field of wireless network engineering and security.

A methodology was developed for evaluating security for wireless networks and satisfies the first goal of this research. Operational parameters from the methodology can be used to represent the characteristics of different wireless networks. Similarities between wireless networks can then be identified more easily and in a formal manner. Wireless networks can be separated into manageable categories. This reduces the space of the problem of securing wireless networks. Metrics can be utilized to determine the resource impact of a security scheme on each category. The resource impact may differ for each security scheme and the metrics help determine suitable security schemes for a wireless network.

Moving beyond the methodology, a context aware and adaptive wireless network security manager was designed and tested. The security manager allows security adaptation depending on environmental circumstances and fulfills the second goal of this research. The security manager gathers context information to make decisions about which security algorithms to use in a given situation. The algorithms for the security manager were also analyzed in terms of resource costs and represent another contribution.

### *1.3 Scope*

This research applies to many commercial and research prototype wireless networks, systems, and devices. The recommendations set forth in this dissertation are intended to assist in designing, implementing, or managing a wireless network to provide services to mobile users or devices. In addition, this work can be used in the project development phase to incorporate appropriate security constraints and protocols and avoid expensive retrofitting after implementation.

This research was not limited to a single operating system (OS), hardware platform, or wireless device. The methodology can be applied to different wireless systems. However, each wireless system requires thorough analysis of the system characteristics, classification, and the security impact. Therefore, I chose some specific applications and devices for case studies, experiments, and quantitative analysis.

It is noted that these procedures may be too time consuming, especially for projects that are on a tight schedule. In addition, a significant change in the system characteristics could necessitate further analysis for a completely different solution. This problem of significant effort at design time is addressed, at least in part, by the context aware and adaptive security manager. The prototype implementation of the adaptive security manager may only operate on specific wireless devices, but it adapts to select and use the most suitable security algorithm in real-time. The adaptive security manager addresses the methodology weaknesses by having the capability to adjust to different environments at run time. The tradeoff with the security manager is that its adaptability is limited to specific devices.

### *1.4 Organization*

This dissertation is divided into seven chapters. Chapter 2 presents an overview of relevant prior and current research on wireless networks. Security mechanisms and adaptive protocols are also reviewed. Chapter 3 discusses the approach employed in this research. Research goals are also discussed. Chapter 4 describes the methodology that was developed as part of this research. The methodology functions as a method to find suitable security mechanisms for different wireless networks. The methods for wireless network classification, incorporating security, and evaluating impact are presented. A novel adaptive security manager design is detailed in Chapter 5. A prototype implementation of the adaptive security manager for a pervasive computing application and associated performance results are discussed in Chapter 6. Chapter 7 summarizes the work, discusses contributions of this research, and presents potential future work. Appendix A lists the procedures and data for the decision making mechanism of the adaptive security manager.

# Chapter 2. Background and Literature Review

An overview of wireless technology, network security, energy efficiency, and adaptive security technology is presented in this chapter. Section 2.1 provides background in wireless networking, while Section 2.2 describes current wireless systems and protocols. Section 2.3 discusses commonly deployed security technology and promising security mechanisms. Section 2.4 describes energy efficiency concepts and issues. Finally, background in adaptive security technology is given in Section 2.5. The chapter is summarized in Section 2.6.

## *2.1 Wireless Network Standards*

This section outlines some of the basic wireless networking standards such as IEEE 802.11 (Section 2.1.1), Bluetooth (Section 2.1.2), IEEE 802.15 (Section 2.1.3), and IEEE 802.16 (Section 2.1.4). Other protocols and standards are not discussed. For more detailed descriptions of some other wireless standards, see Rappaport's book [5]. Additionally, the Defense Information Systems Agency (DISA) has created a Wireless Security Technical Implementation Guide (STIG) [6] that provides useful recommendations and summaries publicly available and is used as the main reference throughout this chapter.

### 2.1.1 IEEE 802.11

The IEEE 802.11 standards specify a data link layer protocol and multiple physical layer protocols defining a wireless local area network (WLAN). The IEEE 802.11 standards can be compared to the IEEE 802.3 wired standards for Ethernet in a local area network (LAN). The 802.11 specifications address both the physical (PHY) and media access control (MAC) layers.

WLANs are generally utilized as extensions to existing wired infrastructures and provide the interface between wireless clients and base stations or access points. Additionally, wireless clients may communicate in a standalone mode. Unfortunately, the mobility that WLANs bring also introduces security issues that must be addressed. For an analysis of potential vulnerabilities in 802.11, see Lough's dissertation [7]. Standards bodies and industry are constantly working on improving data rate, range, and security in wireless networking solutions.

Although there are several components in the 802.11 standard, only subgroups *a*, *b*, *g*, and *i* are summarized below in Sections 2.1.1.1 through 2.1.1.4. For more information on these or other components refer to the IEEE 802.11 standard [8]. For a recent article on WLAN standards, see [9].

#### *2.1.1.1 IEEE 802.11a*

The IEEE 802.11a standard uses the 5 GHz band and operates at data rates from 6 to 54 Mbps [6]. These high data rates are achieved through orthogonal frequency division multiplexing (OFDM).

#### *2.1.1.2 IEEE 802.11b*

The IEEE 802.11b standard is for high speed WLANs and operates in the 2.4 GHz band [6]. Data rates of 1, 2, 5.5, and 11 Mbps are defined in the standard. The physical layer uses direct sequence spread spectrum (DSSS).

#### *2.1.1.3 IEEE 802.11g*

IEEE 802.11g is the "standard for higher rate extensions in the 2.4 GHz band" [6]. The standard uses OFDM and provides data rates up to 54 Mbps.

#### *2.1.1.4 IEEE 802.11i*

Task Group I (802.11TGi) is working to develop enhanced security capabilities for the 802.11 standard. The task group is working on a number of activities to improve WLAN security including adding Advanced Encryption Standard (AES) encryption. See Section 2.3.1.4 for a description of Wi-Fi[1] Protected Access (WPA), an interim implementation of 802.11i. For a recent article on security differences between 802.11i and 802.11b, see [10].

### 2.1.2 Bluetooth

Bluetooth[2] is a wireless specification [11] developed by the Bluetooth Special Interest Group (SIG)[3], a consortium of companies that are interested in promoting Bluetooth wireless solutions. The specification is divided into two parts: Volume 1, Core Specification and Volume 2, Profile Definitions. The Core Specification details requirements for components including the radio, baseband operation, link manager, service discovery protocol, transport layer, and interoperability between different communication protocols [6]. The Profile Definitions part describes the higher-level protocols and procedures needed to implement user-level functions [6].

Bluetooth devices can cooperatively form wirelesss ad hoc networks (Section 2.2.2) called piconets or connect to local access points. This short range system operates at a normal range of 10 m (0 dBm) and an extended range of 100m (+20 dBm). Bluetooth and 802.11b (Section 2.1.1.2) share some characteristics, e.g., "both use 2.4 GHz as their base frequency, and overlap slightly in some usage models, but they serve fundamentally different purposes" [6].

Security for Bluetooth can be found at both the physical and link layers of the protocol. Bluetooth devices transmit on the unlicensed 2.4 GHz Industrial, Scientific, and Medical (ISM) band using Frequency Hopping Spread Spectrum (FHSS) with a high hopping rate of 1600 hops per second. This reduces "casual eavesdropping" by allowing only synchronized devices to be able to communicate in a piconet.

[1] The Wi-Fi Alliance is a nonprofit international association formed in 1999 to certify interoperability of wireless LAN products based on the IEEE 802.11 standard.

[2] The name "Bluetooth" originates from a 10th century Danish king named Harold Blätand.

[3] The Bluetooth Special Interest Group released version 1.2 of the Bluetooth specification in November 2003.

In terms of security the DISA Wireless STIG [6] summarizes that “at the link layer, the Bluetooth specification supports unidirectional or mutual authentication and encryption.” These features are based on a secret link key that is shared by a pair of devices. To generate the secret link key, a pairing procedure is used when the two devices communicate for the first time. “Each Bluetooth device has a unique device address that is also used to authenticate the devices. For encryption, Bluetooth uses an algorithm where the key length is selectable between 8 and 128 bits” [6].

The Bluetooth specification and the IEEE 802.11b standard have similar security problems in that “no process is defined for…issuing, validating, and revoking link keys” [6]. Bluetooth provides built-in encryption and authentication, but like 802.11b, additional security products must be used to mitigate its inherent security shortcomings. See [12] for an analysis of Bluetooth vulnerabilities and [13] for work in demonstration of the vulnerabilities.

### 2.1.3 IEEE 802.15

The IEEE 802.15 Task Group 1 is developing a wireless personal area network (WPAN) standard based on the specification developed through the Bluetooth SIG. The DISA Wireless STIG [6] states that “the goal is to achieve interoperability, e.g., no radio interference, between a WPAN device and any IEEE 802.11 WLAN device. Interference between WLAN technology and Bluetooth (IEEE 802.15 WPAN) networks can be a significant problem, as they both operate in the same frequency band…The IEEE 802.15 Task Group 2 is developing coexistence mechanisms for the two standards. The IEEE 802.15.1 standard defines device-level authentication functionality within the data link layer and data encryption capabilities within the physical layer.” For more detailed information on IEEE 802.15 and its parts, refer to the standard [14].

### 2.1.4 IEEE 802.16

The IEEE 802.16 standard defines interoperability requirements for wireless metropolitan area networks (WMANs) that operate in the 2 to 66 GHz frequency range [15]. “These networks offer subscriber local loop service…and wireless ‘hot-spots’ for Internet connections…and are expected to compete with both public 802.11 and broadband 3G cellular services. WMANs are beginning limited deployment in the United States,” according to the DISA Wireless STIG [6]. Security features of each system depend on each implementation. For more detailed information on IEEE 802.16, see the standard [15].

## *2.2 Wireless Systems and Protocols*

This section covers some of the systems and protocols that use the wireless standards summarized in Section 2.1. Mobile IP (Section 2.2.1) and mobile ad hoc networks (Section 2.2.2) are the two most widely investigated systems.

### 2.2.1 Mobile IP

Mobile IP is a protocol that enables mobile devices to move from one network to another while maintaining permanent Internet Protocol (IP) addresses. It employs mobility agents (routers) to function as a forwarding service. There are two types of mobility agents:  a home agent on the home network and a foreign agent on the visited network. The mobile device, or mobile node, registers with a foreign agent to receive packets from the home network.  In Mobile IPv4[4], the home agent tunnels all traffic for the intended mobile node to the foreign agent which can reach the mobile node.  Mobile IPv6[5] reduces the need for tunneling and, also, includes mechanisms that eliminate the need for foreign agents.  For detailed discussions of Mobile IP, see Perkins's book [16] and Solomon's book [17].

### 2.2.2 Mobile Ad Hoc Networks

"An ad hoc network is the cooperative engagement of a collection of mobile hosts without the required intervention of any centralized access point" [18].  A mobile ad hoc network (MANET) can be a wireless network between two or more mobile hosts, or mobile nodes, where some nodes may not be directly linked to other nodes.  Routing is accomplished by having some nodes function as a mobile router.  Consequently, this leads to many security issues.  For an overview of MANETs, see [19].

## *2.3 Security Technology*

This section describes current deployed security technology and promising security mechanisms.  IEEE 802.11 security solutions are discussed in Section 2.3.1.  Then, Section 2.3.2 explains current MANET specific security schemes.  In Section 2.3.3, some popular security protocols and security mechanisms are reviewed.

### 2.3.1 IEEE 802.11 Security

Like all IEEE 802 standards, the 802.11 standards (802.11a, 802.11b, and 802.11g) focus on the bottom two layers of the Open Systems Interconnect (OSI) model—the physical and data link layers.  The portion of the 802.11 standard that provides security, such as access control and encryption mechanisms, takes place at the data link layer, particularly the MAC sublayer.  The 802.11 MAC can work seamlessly with the standard 802.3 Ethernet, via a bridge or access point, to ensure interoperability between wired and wireless nodes.  Once the access point is reached, the same security standards supported by other 802-compliant LANs for access control and encryption apply [6].  For example, a network operating system login may be used for access control and application level encryption or IP Security (see Section 2.3.3.1) may be used for encryption.

First, Section 2.3.1.1 describes the Service Set Identifier in the standard.  Then, Section 2.3.1.2 explains MAC address filtering as a method for access control, and Section 2.3.1.3 presents the Wired Equivalent Privacy Protocol.  Finally, Section 2.3.1.4 discusses Wi-Fi Protected Access.  The DISA Wireless STIG [6] was the primary reference for this section.

[4] Internet Protocol Version 4

[5] Internet Protocol Version 6

#### *2.3.1.1 Service Set Identifier (SSID)*

Although advertised as a means of simple access control to an access point or group of access points, the SSID should not be considered a safe or reliable access control mechanism.

The DISA Wireless STIG [6] affirms that “this identifier is an alphanumeric code that corresponds to a specific wireless network. The SSID may be advertised as a part of the periodic beacons sent by an access point or it may be requested in a probe-request frame when a wireless client is attempting to associate with a WLAN. Most access points permit the broadcast of their identifier so that wireless stations within range know what is available. It is not good practice to broadcast this identifier since it must be broadcast in clear text.”

The SSID should also be changed from its default setting to something else, such as a pseudo random word consisting of special characters. By having a non-default SSID that is not broadcast, it may be harder to create unauthorized connections to the network.

#### *2.3.1.2 MAC Address Filtering*

A client in a WLAN can be identified by the unique MAC address of its 802.11 wireless network interface card (NIC). Therefore, another type of access control can be implemented based on permitting access to only those MAC addresses that are known to belong to legitimate users.

The DISA Wireless STIG [6] recommends that “for this security to be effective, each access point will need to be configured with a list of authorized MAC addresses. Access to the WLAN will only be permitted to a device that has a MAC address found in the list. MAC related information in the header of a datagram is sent in the clear so it is possible that the MAC address can be obtained by an eavesdropper and spoofed in an attempt to gain access to the WLAN. Although MAC address filtering provides only minimal security, it can be implemented as a deterrent to the casual unauthorized user.”

#### *2.3.1.3 Wired Equivalent Privacy (WEP) Protocol*

There are two types of authentication summarized by the DISA Wireless STIG [6] and defined by the WEP protocol—open system authentication and shared key authentication. “With open system authentication, the access point grants access to stations with an authorized SSID [so, effectively, there is no authentication]. With shared key authentication, both the access point and any station authorized to connect to the access point share a key that is used for both authentication and encryption” [6].

According to the DISA Wireless STIG [6], “the WEP encryption key is comprised of a shared key and a 24-bit initialization vector (IV). The 64-bit WEP key is formed by combining a 40-bit shared key and the IV, while the 128-bit WEP key is formed by combining a 104-bit shared key and the IV. Some WLAN products allow the IV to be changed periodically, including as often as after every transmission.”

However, some known attacks exploit problems with both the encryption and authentication provided by WEP [1, 20, 21]. These flaws result from the current WEP standard “using static, reusable shared secret keys and a poor implementation of the RC4 algorithm” [6]. Several studies have concluded that with minimal hardware and software and statistical analysis (intercepting a minimal amount of wireless traffic), WEP keys can be easily determined [6].

#### *2.3.1.4 Wi-Fi Protected Access*

The DISA Wireless STIG [6] states that “on October 31, 2002, the Wi-Fi Alliance announced that it would require a new interim security specification called Wi-Fi Protected Access (WPA) to be included in devices…WPA relies on an interim version of the new wireless LAN security standard being developed by the IEEE 802.11i task group. WPA also supports…[the Extensible Authentication Protocol (EAP)] to allow simple integration with existing enterprise authentication systems.”

WPA uses a network password that initiates a key rotation every 10,000 bytes of data using the 802.11i Temporal Key Integrity Protocol (TKIP) [6]. However, WPA still uses the RC4 encryption algorithm found in WEP.

### 2.3.2 MANET Security

Three MANET specific security protocols are reviewed in this section. They are Secure-AODV (Section 2.3.2.1), Secure Message Transmission (Section 2.3.2.2), and the Secure Routing Protocol (Section 2.3.2.3). Most of the material in this section can be found in [22], which was the primary reference for MANET security.

#### *2.3.2.1 Secure-AODV*

An extension of the Ad Hoc On-demand Distance Vector (AODV) [23] routing protocol was proposed by Zapata [24] to protect the routing protocol messages. The Secure-AODV (SAODV) scheme assumes that each node has certified public keys of all network nodes, so that intermediate nodes can validate all in-transit routing packets. The basic idea is that the originator of a control message appends an RSA signature [25] and the last element of a hash chain.

Papadimitratos and Haas [22] state that “public-key cryptography imposes a high processing overhead on the intermediate nodes and can be considered unrealistic for a wide range of ad hoc network applications.” Intermediate nodes can also corrupt the route discovery process. They can pretend that the destination is an immediate neighbor, advertising high sequence numbers and changing the actual route length [22]. The IP portion of the S-AODV traffic can also be compromised, since it cannot be protected, unless additional hop-by-hop cryptography and signatures are used [22].

#### *2.3.2.2 Secure Message Transmission*

The Secure Message Transmission (SMT) protocol was proposed by Papadimitratos and Haas [26]. When given a topology view of the network, SMT “determines a set of diverse paths connecting the source and the destination nodes” [22].

Then, Papadimitratos and Haas [22] use "limited transmission redundancy across the paths, by dispersing a message into *N* pieces, so that successful reception of any *M*-out-of-*N* pieces allows the reconstruction of the original message at the destination. Each piece, equipped with a cryptographic header that provides integrity and replay protection along with origin authentication, is transmitted over one of the paths. Upon reception…the destination generates an acknowledgement informing the source of which pieces…were intact… If less than *M* pieces were received, the source retransmits the remaining pieces over the intact routes. If too few pieces were acknowledged or too many messages remain outstanding, the protocol adapts its operation, by determining a different path set, re-encoding undelivered messages and reallocating pieces over the path set. Otherwise, it proceeds with subsequent message transmissions."

SMT also provides limited protection against the use of compromised topological information, although its main focus is to safeguard the data forwarding operation. The use of multiple routes compensates for the use of partially incorrect routing information [27], rendering a compromised route equivalent to a route failure. Nevertheless, the disruption of the route discovery can still be the most effective way for adversaries to consistently compromise the communication of one or more pairs of nodes.

#### *2.3.2.3 Secure Routing Protocol*

Papadimitratos and Haas also proposed the Secure Routing Protocol (SRP) [22] and state that "the scheme guarantees that a node initiating a route discovery will be able to identify and discard replies providing false topological information or avoid receiving them. The novelty of the scheme, as compared with other MANET secure routing schemes, is that false route replies, as a result of malicious node behavior, are discarded partially by benign nodes while in-transit towards the querying node, or deemed invalid upon reception. The security goals are achieved with the existence of a security association between the pair of end nodes only, without the need for intermediate nodes to cryptographically validate control traffic."

This protocol is relatively efficient and scalable, because only the end nodes have to be securely associated and there is no need for cryptographic validation of control traffic at intermediate nodes [22]. SRP places the overhead on the end nodes and contributes to the robustness and flexibility of the scheme. Moreover, SRP does not rely on state stored in intermediate nodes, thus it is immune to malicious acts not directed against the nodes that wish to communicate in a secure manner. Finally, SRP provides one or more route replies, whose correctness is verified by the route "geometry" itself.

### 2.3.3 Security Protocols and Mechanisms

Current security protocols and security mechanisms are reviewed in this section. The IP Security Protocol is covered in Section 2.3.3.1. Next, Section 2.3.3.2 discusses two authentication mechanisms, Kerberos and Radius. Relevant encryption algorithms are discussed in Section 2.3.3.3. The Internet Key Exchange Protocol is presented in Section 2.3.3.4. Finally, two prevalent tunneling protocols are described in Section 2.3.3.5.

#### *2.3.3.1 IP Security Protocol Suite*

IP Security (IPSec) is a set of IP extensions developed by the Internet Engineering Task Force (IETF) to provide security services compatible with IPv4 and IPv6. In addition, IPSec can protect any protocol that runs on top of IP, for instance TCP, UDP, and ICMP. IPSec also provides cryptographic security services that allow for authentication, integrity, access control, and confidentiality. Information exchanged between remote sites can be encrypted and verified with IPSec. Users can create encrypted tunnels, i.e., virtual private networks (VPNs), or just perform encryption between computers.

There are two protocols provided by IPSec, they are the Authentication Header (AH) [28] and the Encapsulated Security Payload (ESP) [29]. AH is a security protocol that provides data authentication and optional anti-replay services. AH is embedded in the data to be protected (a full IP datagram). ESP is a security protocol which provides data privacy services and optional data authentication, and anti-replay services. ESP encapsulates the data to be protected. The two protocols can be used to protect either an entire IP payload or only the upper-layer protocols of an IP payload. An IP host uses transport mode mainly to protect locally generated data, while a security gateway uses tunnel mode to provide IPSec service for other machines lacking IPSec capability [30].

#### *2.3.3.2 Authentication*

##### 2.3.3.2.1 Kerberos

Kerberos[6] is a network authentication protocol[7] [31]. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. A Kerberos domain or realm consists of several entities that cooperate to communicate securely. These are users, servers, and the Key Distribution Center (KDC). The KDC is responsible for coordinating access to services by users and for performing the initial authentication. The KDC is the controller of all secure interactions and, as such, is a trusted entity.

##### 2.3.3.2.2 Remote Authentication Dial In User Service

Remote Authentication Dial In User Service (RADIUS) [32] is currently a standard for remote authentication. RADIUS is a widely used protocol in network environments, especially in embedded network devices such as routers, modem servers, or switches. RADIUS is used in embedded systems because they generally cannot deal with a large number of users with distinct authentication information. This requires more storage than many embedded systems possess.

RADIUS also facilitates centralized user administration. Many Internet service providers (ISPs) have a large number of users and they may be added and deleted continuously throughout the day, and user authentication information changes constantly. Centralized administration of users in this setting is an operational requirement. In addition, RADIUS consistently provides some level of protection against a sniffing, active attacker.

[6] The name Kerberos comes from Greek mythology and represents the three-headed dog that guarded the entrance to Hades.

[7] Find the latest version at http://web.mit.edu/kerberos

#### *2.3.3.3 Encryption*

Encryption is a technique that can be used to address issues regarding confidentiality. Encrypting involves obscuring information through the use of ciphers[8] to make it unreadable without special knowledge [33, 34]. Methods of encryption can be divided into symmetric key algorithms and asymmetric key algorithms. In a symmetric key algorithm, the sender and receiver must have a shared key established in advance and kept secret from all other parties. The sender uses this key for encryption, and the receiver uses the same key for decryption. In an asymmetric key algorithm, there are two separate keys, a public key and a private key. A public key is published and enables any sender to perform encryption, while a private key is kept secret by the receiver and enables decryption.

Symmetric key ciphers can be distinguished into two types, depending on whether they work on blocks of symbols usually of a fixed size or on a continuous stream of symbols. In other words, a block cipher encrypts data in blocks rather than encrypting one bit at a time in a stream, a scheme otherwise known as a stream cipher [34]. Stream ciphers are commonly used with streaming applications or for secure connections, such as Transport Layer Security [35], while block ciphers are typically used for storing data in a database or encrypting files. Since streaming applications, such as streaming audio or video, require constant connectivity and resources that are not always available in personal digital assistants (PDAs) and other small networked devices, block ciphers were chosen for this work which focuses on such devices. However, the methodology could be extended to include stream ciphers as future work. Four block ciphers, described below, were used in this study, specifically RC2, Blowfish, the eXtended Tiny Encryption Algorithm (XTEA), and the Advanced Encryption Standard (AES).

RC2[9] is a 64-bit block cipher with a variable size key. Like most block ciphers, RC2 uses a Feistel network [33] for diffusing the plaintext. A Feistel network is usually a combination of bit-shuffling through permutation boxes (P-boxes), simple non-linear operations using substitution boxes (S-boxes), and linear mixing using the exclusive-OR (XOR) operator. The block cipher iterates plaintext through its Feistel network to generate the ciphertext. Each iteration through the Feistel network is usually counted as one *round* of encryption. In RC2, its 18 rounds are arranged as a source-heavy Feistel network, which means the input to the round function is larger than the output of the round function. The Feistel network involves *mixing* and *mashing* rounds. The mixing rounds consist of sequentially interleaving an expanded encryption key with the plaintext, while the mashing rounds combine different pieces of the expanded key and the results of the mixing rounds. There are 16 rounds of mixing punctuated by two rounds of mashing [36].

Blowfish is a symmetric key (also known as a secret or private key) block cipher designed in 1993 by Schneier [37]. Blowfish has a 64-bit block size and a key length of anywhere from 32 bits to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes.

[8] A cipher is an algorithm for performing encryption.

[9] “RC” stands for “Ron’s Code” or “Rivest Cipher.”

The Tiny Encryption Algorithm (TEA) is a block cipher noted for its simplicity of description and implementation (typically a few lines of code) [38]. However, TEA has a few vulnerabilities [39] and, consequently, XTEA was designed to correct those weaknesses [40]. XTEA is a 64-bit block Feistel network with a 128-bit key and a recommended 64 rounds.

AES, also known as Rijndael, is a block cipher adopted as an encryption standard by the U.S. government [41, 42]. AES is based on a substitution-permutation network and has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits.

#### *2.3.3.4 Key Management*

The Internet Key Exchange (IKE) [43] protocol is a key management protocol standard which is used in conjunction with the IPSec standard (Section 2.3.3.1). IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard. IKE is a hybrid protocol which implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework.

IKE automatically negotiates IPSec security associations (SAs) and enables IPSec secure communications without costly manual preconfiguration. An IPSec SA is a description of how two or more entities will use security services in the context of a particular security protocol (AH or ESP) to communicate securely on behalf of a particular data flow. The IPSec SA is established either by IKE or by manual user configuration. Security associations are unidirectional and are unique per security protocol. So when security associations are established for IPSec, the security associations (for each protocol) for both directions are established at the same time.

When using IKE to establish the security associations for the data flow, the security associations are established when needed and expire after a period of time (or volume of traffic). If the security associations are manually established, they are established as soon as the necessary configuration is completed and do not expire.

IKE provides the following benefits to IPSec [44]:

- Eliminates the need to manually specify all the IPSec security parameters in the crypto maps at both peers.
- Allows setting a lifetime for the IPSec security association.
- Allows encryption keys to change during IPSec sessions.
- Allows IPSec to provide anti-replay services.
- Permits Certification Authority (CA) support for a manageable, scalable IPSec implementation.
- Allows dynamic authentication of peers.

#### *2.3.3.5 Tunneling*

Point-to-Point Tunneling Protocol (PPTP) [45] is a tunneling protocol defined by the PPTP Forum that allows PPP packets to be encapsulated within IP packets and forwarded over any IP network, including the Internet itself. Layer Two Tunneling Protocol (L2TP) [46] is an extension of PPTP used by an ISP to enable the operation of a VPN over the Internet.

## 2.4 Energy Efficiency

This section discusses various aspects of energy efficiency. First, Section 2.4.1 gives a brief introduction to energy efficiency. Section 2.4.2 describes the possible energy consumption sources when using a wireless device. Then, Section 2.4.3 outlines reasons for energy consumed during wireless communications. Finally, some guidelines and mechanisms for energy conservation are presented in Section 2.4.4.

### 2.4.1 Background

Energy efficiency is defined by Havinga and Smit [47] as "the energy dissipation that is essentially needed to perform a certain function, divided by the actually used total energy dissipation." The range of possible functions can be very broad. The function executed can be simple like a multiply operation, but it can also be a complete network protocol. For example, consider a MAC protocol that controls access to a wireless channel. The essential energy consumed is the energy needed to transfer a certain number of bits over the wireless channel. The actual total energy consumed also includes the overhead involved in the data link layer (additional packet headers, error control, etc.) as well as the physical layer overhead (e.g., overhead needed for a frequency hopping scheme). Jones, *et al.* [48] define the energy efficiency of a protocol as "the average number of successful transmissions per energy unit, which can be computed as the average number of successes per transmission attempt."

The energy efficiency of a certain function is independent of the actual implementation and, thus, independent of whether an implementation is low power. Low power implementation represents how the hardware manages power consumption. In contrast, energy efficiency represents the algorithms using the hardware. Thus, it is possible to have two implementations of a certain function that are built with different building blocks: (1) one that has been built with power-hungry components has a high energy efficiency, but dissipates a lot of energy, and (2) one that has low energy efficiency, but is built with low power components.

### 2.4.2 Energy Consumption on a Wireless Device

The sources of energy consumed on a wireless device can be classified into two types: communication related and computation related [49].

Communication related energy consumption involves the usage of the transceiver at the source, intermediate (in the case of ad hoc networks), and destination nodes. The transmitter is used for sending control, route request and response, as well as data packets originating at or routed through the transmitting node. The receiver is used to receive data and control packets, some of which are destined for the receiving node and some of which are forwarded. Communication energy is mainly determined by the signal-to-noise ratio (SNR) requirements and the radio cell diameter. A typical mobile device exists in three modes: transmit, receive and standby. Most of the power is consumed in the transmit mode and least in the standby mode [48].

Conversely, computation related energy consumption factors are concerned with protocol processing aspects. The energy drained by computation mainly involves the usage of the central processing unit (CPU) and main memory. In addition, data compression techniques that reduce packet length (and also energy usage for transmission) may result in increased computation. Computation energy is a function of the hardware and software used for tasks such as compression and forward error correction (FEC) [49]. One area of related work is that of low power software. The consensus is that the potential for power savings in software is greater than the potential for savings in hardware, but that the software savings are more difficult to achieve [50]. A common finding is that energy consumption is tied very closely to execution time [51, 52].

There exists a potential tradeoff between computation and communication costs, as suggested by the discussion of compression above. Techniques that may achieve lower communication costs may also result in higher computational needs, and vice-versa. Therefore, a balance between the two goals must be reached for energy efficiency to be achieved.

### 2.4.3 Energy Consumption on a Wireless Channel

Research by Jones, *et al.* [48] and Lettieri and Srivastava [49] has shown that some of the main causes of unnecessary energy consumption needed for communication over a wireless channel include the following.

- *Collisions which result in retransmissions.* Retransmissions lead to unnecessary power consumption and potentially long delays. Unfortunately, retransmissions cannot be completely avoided due to the nature of wireless communication, the mobility of users, and varying sets of mobile devices in a cell, associated with an access point, or within range of a node in an ad hoc network.

- *Prolonged inactivity.* For applications that have low traffic needs, the transceiver is idle most of the time. Measurements show that on typical applications like a web browser or electronic mail, the energy consumed while the interface is on and idle is more than the cost of actually receiving packets.

- *Constantly active receivers.* The receiver has to be powered on at all times, especially in broadcast environments, to be able to receive messages from a base station, access point, or other nodes in an ad hoc network, resulting in significant energy consumption.

- *Switching between communication modes.* An example of this is switching from idle to transmit mode or transmit mode to receive mode. For example, a protocol that allocates permission on a slot-by-slot basis suffers substantial overhead because of turnaround between transmit and receive modes and vice-versa.

- *Poor channel conditions*. Transmissions under poor channel conditions are likely to result in the reception of frames with many errors.

- *Routing network traffic through nodes with low energy capacity*. This will cause the network lifetime to significantly decrease and will also introduce more network partitions when the nodes exhaust their energy supply.

- *Overhead of the protocol.* More energy is consumed due to the amount of unnecessary control data and required computation for protocol handling. This includes long headers for addressing or control, long trailers for error detection and correction, and a large number of required control messages such as acknowledgement, request-to-send (RTS), and clear-to-send (CTS) messages.

- *High error rate of wireless links*. When the data are not correctly received, the energy that was needed to transport and process the data is wasted. Energy is also used for the error control mechanism.

- *Non-centralized scheduling algorithm*. Having distributed scheduling, wherein each mobile device computes the schedule independently, especially in MANETs, may not be desirable because mobile nodes may not receive reservation requests due to radio and error constraints, and schedule computation consumes energy.

### 2.4.4 Energy Conservation Guidelines and Mechanisms

For a long time, energy conservation and low power design in wireless devices was centered on the physical layer due to the fact that consumption of power in a mobile device is a direct result of the system hardware. Research focused on two different approaches to address the energy problem: (1) increase battery capacity, and (2) decrease the amount of energy consumed at the wireless node.

The primary problem concerning energy in wireless networking is that the battery capacity is extremely limited. Research on battery technology has focused on increasing battery capacity while reducing battery weight and size [48]. Unfortunately, battery technology has not experienced significant advancements; therefore, a more attainable goal of research would be to decrease the energy consumed in the wireless terminal [49].

Numerous energy efficient techniques for the physical layer have been discussed, but it is important to consider other avenues of achieving energy efficiency. One way to achieve energy efficiency is to consider the higher layers of the protocol stack in the design of mobile nodes and maintain energy efficiency as a first class design constraint.

Some of the guidelines that may be adopted for an energy efficient protocol design in the higher layers of the protocol stack are described below. A list of areas in which conservation mechanisms are beneficial is also provided. Adaptability of the protocols is a key issue. Two basic principles that have been suggested to achieving an energy efficient system are to avoid unnecessary actions and reduce the amount of data traffic [47]. The mechanisms suggested include the following.

- Eliminating collisions as much as possible within the MAC layer since they result in retransmissions. For example, in infrastructure networks, new users registering with the base station or access point may have to use some form of random access protocol. In this case, using a small packet size for registration and bandwidth requests may reduce energy consumption.

- Broadcasting a schedule that contains data transmission start times for each mobile. This enables the mobile to switch to standby mode until the receive start time. Alternatively, the transceiver can be switched off whenever the node determines that it will not be receiving data for a period of time.

- Allocating contiguous slots for transmission or reception to reduce turnaround, resulting in lower power consumption. In addition, it would be beneficial for mobile nodes to request multiple transmission slots with a single reservation packet when requesting bandwidth to reduce the reservation overhead.

- Centralizing the scheduling mechanism that computes the system transmission schedule at the base station. This case would be a valid approach in a situation where mobile nodes transmit data requests to the base station or access point, as in an infrastructure network. Therefore, computation of the transmission schedule must be relegated to the base station, which in turn broadcasts the schedule to each mobile.

- The scheduling algorithm at the base station or access point may consider the mobile node's battery power level, in addition to connection priority. This would allow traffic from low power mobile nodes that may be dropped due to depletion of power reserves to be transmitted sooner than traffic from other nodes [48]. Also, under low power conditions, it may be useful to allow a mobile node to rearrange allocated slots among its own flows. This may allow certain high-priority traffic to be transmitted sooner rather than waiting for the originally scheduled time in the context of energy efficiency and channel error compensation.

- Avoiding transmissions when channel conditions are poor. In addition, error control schemes that combine automatic repeat request (ARQ) and forward error correction (FEC) mechanisms may be used to conserve power.

- Establishing routes that ensure that all nodes equally deplete their battery power. This helps balance the amount of traffic carried by each node. In addition, route avoidance can be employed in cases where a node has lower battery power. But, this method requires a mechanism for dissemination of node battery power.

- Suspension of a specific sub-unit, such as a disk, memory, or display, based upon detection of prolonged inactivity [48]. Other techniques studied include power aware CPU scheduling and page allocation. It is important to note that within the application layer, power conservation mechanisms tend to be application specific, such as database access and video processing.

## *2.5 Adaptive Security Technology*

This section reviews literature related to adaptive security techniques. First, Section 2.5.1 briefly introduces adaptive security. Then, related work on adaptive security is presented in Section 2.5.2. Some adaptive approaches to designing security are discussed in Section 2.5.3. A review of the Analytic Hierarchy Process (AHP) as a decision making technique is given in Section 2.5.4. Finally, Section 2.5.5 concludes with remarks about adaptive methods.

### 2.5.1 Background

Modern information security standards include such obligatory requirements for secure systems as self-testing, fault-tolerance and active assessment. However, most traditional security models were developed without taking into consideration the presence of dynamic elements. In practice, adaptive algorithms designed for specific secure information systems must be integrated into the Trusted Computing Base[10] (TCB) of the system. Full access to the assessment tools and system control services must also be provided for the adaptive algorithms. Such algorithms can be implemented at the TCB on the application and hardware layers. In particular, for quantitative assessment of the efficiency and adequacy of miscellaneous security policies the following information can be used [53]: throughput fluctuations in the data channels; processing power loading; time spent on the specific security functions; and quality of service of the entire system pertaining to metrics such as delays, denials, and failures.

Gathering information for adapting complex secure systems can be implemented by using services that register external influences of the environment or by examining internal system states [53]. The combination of these methods provides the best results in terms of accurate adaptation, but requires more resources to accomplish the entire adaptation process. Adequate adaptation methods should be selected depending on the flexibility of secure system components and the complexity of the overall security tasks.

### 2.5.2 Related Work

Son, *et al.* [54] propose a security manager for a distributed real-time database system that has the ability to adapt its behavior during transient overloads in which case transactions can be selected to be executed at a lower security level, thereby, reducing the resource demand on the system. They use a multi-level security classification based on four levels of security. Each level requires a greater degree of protection than the level below it. The heightened security from one level to the next translates to increased demand on system resources. For example, the techniques applied to achieve network security services typically become more complex and time consuming as the desired level of assurance increases. These distinctions can be applied in a multi-level security system to achieve adaptable security.

Schneck and Schwan have developed Authenticast [55], a dynamically configurable user-level communication protocol offering variable levels of security. Authenticast includes

[10] The trusted computing base is everything in a computing system that provides a secure environment. This includes the operating system and its provided security mechanisms, hardware, physical locations, network hardware and software, and prescribed procedures.

multiple heuristics to realize dynamic security levels and to decide when and how to apply dynamic security. The protocol uses a “security thermostat” to enable adaptive security processing. Their work primarily addresses CPU and security issues rather than network resources.

Zou, Lu, and Jin present an intelligent firewall architecture [56]. The firewall is based on a fuzzy adaptive security algorithm. A fuzzy controller is the core module and packet characteristics are “fuzzified” as its inputs. The overall security level of each kind of packet can be discovered and adjusted according to the varying states of the packets. The intelligent firewall can respond to these changes and take different actions accordingly. The intelligent firewall consists of six components: packet capture and data mining module, static packet filter, dynamic packet monitor, address translation gateway, control module, and security policy rules.

Venkatesan and Bhattacharya present a threat-adaptive security policy [57] which enforces a dynamic and individualized security policy mechanism with a trust state machine capturing the different security levels. They discuss a threat-adaptive firewall designed for an electronic commerce application, which adaptively varies the security constraints for each user, thereby improving system performance. The proposed scheme associates a level of trust with every user during run-time and decides an individualized security policy for each user.

### 2.5.3 Adaptive Approaches

An adaptive approach for information security is necessary, especially when the mathematical model of a system and the environmental influences on the system are uncertain after design time or compile time. In decreasing the uncertainty, the adaptation of complex secure systems can be parametrical and structural. In the former, the adaptation is implied as specific variations of the control parameters vector. In the latter, a decrease in uncertainty is achieved by dynamic changes in the structure of the complex information system.

When parametrical adaptation of the complex secure system is chosen, the specific settings and properties of the security tools and methods used are changing during the functioning. For example, depending on the current state of the system and the environment, the rules of the current security policy can become more or less strict; the settings of the network tools, such as firewalls and external authentication devices, can be varied; and the properties of the cryptographic methods, such as algorithms and protocols, can be changed. In general, parametrical adaptation is the simplest method of adaptation and can be relatively easily implemented in new and existing systems.

When structural adaptation of the complex secure system is chosen, both the quantitative and qualitative parameters of the system’s algorithms are utilized. For example, depending on the state of the system and the environment, the security policy itself (and corresponding security model) can be replaced with a new one; the set and the sequence of the security tools used can be varied; and alternative cryptographic methods can be applied. In general, structural adaptation is more complex than parametrical adaptation, but also more flexible. Selection of appropriate adaptation methods should also depend on the overall goals and security requirements of the complex information system.

### 2.5.4 Decision Making using the Analytic Hierarchy Process

Adaptive applications may require selection or decision making among certain alternatives. The method used in this research for decision making in parametrical and structural adaptation is called the Analytic Hierarchy Process[11] (AHP) [58]. It is a flexible decision making process that helps set priorities and make the best decision when both qualitative and quantitative aspects of a decision need to be considered. The method reduces complex decisions to a series of one-to-one comparisons and provides rationale for the results.

Given several choices and *objectives*, the first step in AHP is to decide the relative importance of the objectives. This done by comparing each pair of objectives and ranking them on the following scale.

> Comparing objective *i* and objective *j*, where *i* is assumed to be at least as important as *j*, give a value $a_{i,j}$ as follows in Table 2.1.

**Table 2.1: Pairwise Comparison Values**

| | |
|---|---|
| 1 | Objectives *i* and *j* and of equal importance |
| 3 | Objective *i* is weakly more important than *j* |
| 5 | Objective *i* is strongly more important than *j* |
| 7 | Objective *i* is very strongly more important than *j* |
| 9 | Objective *i* is absolutely more important than *j* |
| 2, 4, 6, 8 | Intermediate values |

A matrix can be created from these preferences.

> Given *n* objectives, for all $a_{i,j}$ with $i = j$, set $a_{i,j} = 1$; and if $a_{i,j} = k_{i,j}$, then $a_{j,i} = 1/k_{i,j}$. Let *O* be an $n \times n$ matrix containing all the elements $a_{i,j}$.

The resulting *objective matrix* created from the preferences is shown in Equation 2.1.

$$O = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ \vdots & a_{3,2} & \ddots & \vdots & \vdots \\ a_{n-1,1} & \vdots & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-1} & a_{n,n} \end{bmatrix} = \begin{bmatrix} 1 & k_{1,2} & \cdots & k_{1,n-1} & k_{1,n} \\ k_{1,2}^{-1} & 1 & k_{2,3} & \cdots & k_{2,n} \\ \vdots & k_{2,3}^{-1} & \ddots & \vdots & \vdots \\ k_{1,n-1}^{-1} & \vdots & \cdots & 1 & k_{n-1,n} \\ k_{1,n}^{-1} & k_{2,n}^{-1} & \cdots & k_{n-1,n}^{-1} & 1 \end{bmatrix} \quad (2.1)$$

[11] "Designed to reflect the way people actually think, AHP was developed in the 1970's by Dr. Thomas Saaty, while he was a professor at the Wharton School of Business" (from http://www.expertchoice.com/customerservice/ahp.htm).

Each entry in the objective matrix, *O*, is then divided by the sum of the column in which it resides. This normalization process is delineated by Equation 2.2, creating the matrix |*O*|. The columns are normalized such that each objective pair is less than 1 and all the values in a single column add up to 1. The normalized values are then averaged across each row to create a column *objective vector* of size *n*, shown in Equation 2.3.

$$\forall a_{i,j} \text{ in } O,\ b_{i,j} = a_{i,j} / \sum_{l=1}^{n} a_{l,j} \ni |O| = O / \sum_{l=1}^{n} a_{l,j} \tag{2.2}$$

$$\forall b_{i,j} \text{ in } |O|,\ o_i = \frac{1}{n}\sum_{l=1}^{n} b_{i,l} \ni \mathbf{o} = \begin{bmatrix} o_1 \\ \vdots \\ o_n \end{bmatrix} \tag{2.3}$$

For a perfectly consistent decision maker, each column should be identical, except for the normalization. Through dividing by the total in each column, identical columns are expected with each entry giving the relative weight of the row's objective. Averaging across each row should then correct any small inconsistencies in the decision making process.

The choices, or *options*, are dealt with next. An *options matrix* is created for each objective. The options are compared to each other with respect to a particular objective, similar to constructing the objective matrix. Again each options matrix is normalized (divided by the sums of the columns) and averaged across rows to obtain the relative weights of each options with regards to a single objective.

> Given *m* options and *n* objectives, there will be *n* column vectors of length *m*. Let **p**(*r*) represent the vertical *option vectors*, where *r* = 0,…,*n*,. Let *P* be a matrix constructed from all the option vectors. Then **d** will be the *decision vector* created from the product of *P* and **o**.

$$\mathbf{p}(1) = \begin{bmatrix} p_{1,1} \\ \vdots \\ p_{m,1} \end{bmatrix}, \cdots, \mathbf{p}(n) = \begin{bmatrix} p_{1,n} \\ \vdots \\ p_{m,n} \end{bmatrix} \tag{2.4}$$

$$P = \begin{bmatrix} \mathbf{p}(1) & \cdots & \mathbf{p}(n) \end{bmatrix} \Rightarrow P = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{bmatrix} \tag{2.5}$$

$$\mathbf{d} = P \cdot \mathbf{o} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{bmatrix} \begin{bmatrix} o_1 \\ \vdots \\ o_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_m \end{bmatrix} \tag{2.6}$$

The procedure for creating the decision vector, **d**, from the option vectors, **p**(*r*), and objective vector, **o**, is shown in Equations 2.4 to 2.6. The element in the decision vector with the largest value corresponds to the most suitable option out of *m* options. In the case of more than one element equal to the largest value, the option can be randomly selected from the largest equal elements.

The AHP is a method for formalizing decision making where there are a limited number of choices, but each has a number of attributes and it is difficult to formalize some of those attributes. Despite the rather arbitrary aspects of the procedure, however, it can provide useful insight into the tradeoffs embedded in a decision making problem.

### 2.5.5 Conclusions

The difficulty of developing universal models describing information security is apparent. In practice, every complex information system represents a unique object which requires an individual approach. Adaptive technologies allow development of solutions that function efficiently in dynamic environments without complete mathematical models of the object that is controlled. The cost of such advantages is the complexity of implementing adaptive systems in comparison with non-adaptive systems.

## *2.6 Summary*

This chapter presented an overview of wireless technology, network security, energy efficiency, and adaptive network technology. Chapter 3 discusses the approach used for this dissertation.

# Chapter 3. Problem Statement and Methodology

This chapter discusses the problem addressed in this research and methodology used. First, Section 3.1 explains the problem statement for this work. Then, Section 3.2 describes the development and verification of an adaptive security methodology. An adaptive security solution is proposed in Section 3.3. Finally, the chapter is summarized in Section 3.4.

## *3.1 Problem Statement*

This research investigates a method to determine appropriate security protocols for specific wireless network applications and, building on these results, an adaptive security manager to select appropriate protocols at run-time. The specific problem being addressed is that currently proposed security protocols often make inefficient use of resources for wireless networks with certain requirements. Existing security protocols address problems such as authentication, availability, confidentiality, integrity, and non-repudiation. However, all of these protocols use resources and limit the efficient use of wireless node resources in several ways, as described below.

- Security protocols significantly increase the amount of overhead required to secure the network, thereby decreasing throughput.
- Security protocols increase the delay between data transmissions due to processing by security algorithms.
- Security protocols decrease the data rates of wireless links because additional traffic is usually needed for authentication or verification services.
- Security protocols increase power and energy consumption at wireless devices since complex security features, such as encryption and decryption, require multiple linear operations and many processing cycles.

This research first focused on developing a methodology that categorizes and evaluates secure networks, which can lead to improved efficiency in wireless nodes after integrating security protocols. The research then considers an adaptive security manager that can utilize context information and decision making to select the "best" security protocol to use for a given situation.

Part of this research focused on both the computational performance and the energy consumption of block cipher encryption algorithms for the PDA platform. Prior work from Ganesan, *et al.* [59] assess the feasibility of different encryption schemes for a range of embedded architectures using execution time overhead measurements. Dhawan [60] describes a study that compared the performance of several encryption algorithms in different network environments. First round AES candidates were compared in terms of efficiency by Bassham [61] using systems with processors in the 200 MHz to 500 MHz range. A study on Palm OS devices [62] also compared performance measurements of AES and other ciphers. Potlapally, *et al.* [63] investigated energy consumption of different ciphers on the Secure Sockets Layer [64].

## *3.2 Methodology Development and Verification*

This section discusses the development and verification of the methodology. Assumptions are stated in Section 3.2.1. Then, the approach for the methodology design is presented in Section 3.2.2. Finally, Section 3.2.3 describes the verification of the methodology.

### 3.2.1 Assumptions

Some assumptions are needed to reduce the scope of the problem. The reasoning behind these assumptions is to manage the complexity of finding suitable security protocols. The assumptions are described below.

First, only wireless networks are considered in this methodology. This does not imply that the methodology is not applicable to other wired or wireless communication systems or protocols, but they are not explicitly considered in its design.

Second, only "complete" security protocols are elements in the set of possible security solutions. In other words, individual algorithms that are used by security protocols are not directly applied to wireless networks.

Third, security protocols are only evaluated based on their resource impact to a wireless network. Vulnerability analysis is not part of this methodology or this research.

Fourth, it is up to the user (e.g., designer, researcher, or project manager) to properly integrate a security protocol into a wireless network when using this methodology. The methodology is intended only to guide in the analysis and selection of suitable security protocols.

### 3.2.2 Methodology Design Approach

The methodology was designed using a systems approach to finding the most suitable security protocol for securing a wireless network. To find the suitability of a security protocol for a given wireless network, the wireless network must first be characterized. Security solutions do not generally apply to every wireless system. Thus, the wireless systems need to be separated into manageable categories. To accomplish this task, a classification scheme is constructed based on the characteristics of common wireless networks. Security protocols have to somehow be incorporated into the wireless networks for evaluation purposes. This can be done through theoretical application, simulation, or direct application. When evaluating the security protocols, metrics are needed to analyze the impact of the security protocols on resources. Finally, suitability analysis is needed to complete the methodology and determine how suitable a security protocol is for a given wireless network.

### 3.2.3 Verification of Methodology

Verification is needed to determine if the methodology functions correctly. To verify the methodology, its components need to be applied to different wireless networks and security protocols. The methodology also needs to be compared to other approaches in this area of research. As discussed in the literature review of Chapter 2, only certain functions of this methodology have been researched or are currently being implemented. Other methodologies [65-72] are more specialized and address only security, energy, or performance problems and not all three in one approach as is done in this research.

The main advantage of this methodology is the flexibility of how constraints are considered and suitability is analyzed. I do not attempt to use the methodology to create a security protocol that is universally applicable to any type of wireless network. My approach can identify problems from manageable categories of networks and find or create solutions for each of them. Another advantage of this methodology is that after suitable security protocols are found or created for each category, any new wireless network application that falls into an existing category may be able to use the security protocols from that category and find that they are the most suitable.

The methodology is described more completely in Chapter 4.

## 3.3 Adaptive Security Solution

Based on the methodology, I have found that there are situations where certain wireless systems are not supported well by a specific security protocol. Even the most suitable security protocols may exhibit poor performance or energy efficiency in many situations. Given a security protocol, its impact will vary depending on the wireless network category and, to a certain extent, the particular type of wireless networks within a category. In addition, the operating environment or context of a wireless networks may vary significantly in terms of operating conditions that affect the performance and energy efficiency of security protocols. This could lead to different security protocols being the most suitable depending on the specific operating environment. For these reasons, I have investigated and developed an adaptive security solution which minimizes the negative impacts while maintaining a certain security level by allowing the security protocol to adapt.

Information about adaptive security methods is given in Section 2.5. In that section, some specific adaptive approaches are also discussed. Initially, a parametrical adaptive approach was used to solve this problem. This was to reduce the complexity of the problem and the potential processing requirements for the adaptation. Modifications of the security manager of Son, *et al.* [54] and threat-adaptive security policy described by Venkatesan and Bhattacharya [57] were investigated and utilized for the adaptive security solution developed in this research.

The adaptive security manager was designed for the "Pervasive Embedded Networks for Ad Hoc Environments" project[12] in Virginia Tech. The embedded systems in the project provided the testbed environment for my research. I directly applied the adaptive

[12] Microsoft Research awarded the Laboratory for Advanced Networking in Virginia Tech a grant involving research with Microsoft embedded systems. Further information is available at the web site http://www.irean.vt.edu/pervasive/ and in the poster "Ad Hoc Networking Support for Pervasive Collaboration" by M. S. Thompson, W. O. Plymale, C. T. Hager, K. Henderson, S. F. Midkiff, L. A. DaSilva, N. J. Davis, and J. S. Park, presented at the International Conference on Ubiquitous Computing (Ubicomp), Nottingham, U.K., September 7-10, 2004 (http://ubicomp.org/ubicomp2004/adjunct/posters/).

security protocol to the testbed and administered performance, energy, and resource consumption experiments. Details about the design and experimental results and discussion of the adaptive security manager are in Chapters 5 and 6.

### 3.4 Summary

This chapter discusses the methodology used for this research. First, Section 3.1 explains the problem statement for this work. Then, Section 3.2 describes the development and verification of the methodology. An adaptive security solution is introduced in Section 3.3. Chapter 4 describes the methodology and addresses related issues, while Chapters 5 and 6 present the design of the adaptive security solution and results.

# Chapter 4. Methodology Description

This chapter describes the methodology developed and used in this research. With this methodology one can focus on designing or modifying a security scheme for a particular wireless network application. These methods are necessary for categorizing wireless networks and minimizing the impact of security schemes on factors such as performance and energy consumption. The classification and impact analysis procedures are presented in this chapter. The methodology consists of three phases. First, Section 4.1 describes the wireless network classification procedure. Next, security protocol incorporation is discussed in Section 4.2. Evaluation of the impact from adding security is related in Section 4.3. A case study is also presented in this chapter in Section 4.5. Finally, Section 4.6 summarizes the results relating to the methodology.

## *4.1 Wireless Network Classification*

To determine a suitable security scheme for a wireless network application, the wireless network application must first be categorized. Every wireless application will have certain network characteristics and constraints (identifying requirements) that can be used for classification. Classifying wireless networks into distinct categories localizes certain security issues to that category and facilitates finding suitable security schemes for a given wireless application. Classification also reduces the set of security schemes for suitability analysis. The characteristics are represented by operational parameters which are discussed in Section 4.1.1. In addition, several common wireless applications that have been categorized by the operational parameters are presented in Section 4.1.3 to illustrate the classification procedure.

### 4.1.1 Operational Parameters

The following subsections explain the operational parameters used in this methodology for classifying wireless network applications into different categories. There are three groups of operational parameters: (1) equipment, (2) network topology, and (3) communication characteristics.

#### *4.1.1.1 Equipment*

Individual nodes, whether they are user devices or base stations, may require different types of equipment that have certain requirements. These requirements are stated in Sections 4.1.1.1.1 through 4.1.1.1.3.

##### 4.1.1.1.1 Energy Requirement

The *energy requirement* parameter defines the energy capacity requirement of a node. For example, small mobile devices may have a low energy capacity; thus, this parameter would be a larger factor in determining an appropriate security protocol for the wireless system. Security protocols might have a significant impact on energy consumption at a node which would directly affect the lifetime of a node.

##### 4.1.1.1.2 User Interface

Depending on the network application, each node may have different *user interfaces* for security or data acquisition purposes. Some common interface elements include passwords, voice recognition, and electronic identification cards. Whether or not a user interface exists for security purposes will determine applicable security protocols, as some require user input for key management.

##### 4.1.1.1.3 End User Device

The *end user device* parameter indicates the actual device utilized by the end user. Some examples of end user devices are workstations, personal digital assistants (PDAs), and notebook computers. Security protocols will have less of an impact on devices that have significantly more processing power.

#### *4.1.1.2 Network Topology*

The network characteristics of a wireless network application can be described by its required *network topology* [73]. Sections 4.1.1.2.1 through 4.1.1.2.4 outline parameters related to the network topology.

##### 4.1.1.2.1 Network Size

The *network size* is the average number of nodes in a wireless system. This parameter relates to the scalability of a security scheme. Since scalability seems to be a limiting factor in many security solutions, wireless networks with a small network size may have a larger set of suitable security protocols available to them than wireless networks with a large network size, for example in the hundreds or more.

##### 4.1.1.2.2 Node Degree

The *node degree* represents the average number of nodes that are directly linked to a given node in the network. Note that this parameter does not specify the average number of neighbors in the physical vicinity (within radio range) of a given node, but the average number of one-hop links to that node. For example, a Bluetooth device is restricted to a maximum of seven direct links with other Bluetooth devices even though there may be hundreds of devices within communication range. The scalability of security protocols will be affected by this parameter. A large node degree in a wireless network may require more data processing at certain nodes when using a particular security protocol.

##### 4.1.1.2.3 Intermediate Nodes

The *intermediate nodes* parameter relates the average number of intermediate nodes between two end nodes. A wireless network is labeled as either a single-hop or multiple-hop network. In a single-hop network, all of the nodes are directly connected to one another. In a multiple-hop network, a particular node might not be able to directly communicate with other nodes in the network. Wireless networks with many intermediate nodes may require tunneling security services. In addition, some security protocols may not be suitable in a large multiple-hop network because they might require authentication at every intermediate node and impose a large delay between end nodes.

#### 4.1.1.2.4 Node Proximity and Node Range

*Node proximity* is the average physical distance between nodes. The maximum communication distance possible between two nodes is the *node range*. These two parameters are related because they affect the necessity of certain security features. For example, an 802.11b ad hoc network with a node proximity of one meter should still require authentication and encryption, because it would have a node range of 100 m, exposing the network to potential intruders at a safe distance. As a counter example, a wearable network with very small node proximity and node range values might not require authentication services, because the devices would always be worn by the user.

### *4.1.1.3 Communication Characteristics*

The following operational parameters describe the communication characteristics of the wireless network.

#### 4.1.1.3.1 Link Capacity

The *link capacity* is the effective link speed measured in bits per second (bps), after accounting for losses due to multiple access, coding, framing, etc. A wireless network with a higher link capacity value should be able to support more security services with less of a performance impact than a wireless network with a lower link capacity value.

#### 4.1.1.3.2 Traffic Pattern

The *traffic pattern* parameter identifies the expected pattern of network traffic and the nature of the data. While the technologies, protocols and network infrastructure supporting wireless data are often complex, most data applications can be simply divided into three main types [74]: bursty, query-response, and batch file. Knowing the traffic pattern of a wireless network will help determine appropriate security protocols that may match certain types of traffic.

In bursty traffic, quick bursts of data are sent from point-to-point. Emerging applications in this area include remote electric power meter readings, wireless burglar alarms, and other remote sensing applications. Bursty traffic is also related to periodic traffic. For example, in a sensor network a node could sense data, send the data, sleep for ten minutes, then wake up and repeat the cycle.

Query-response traffic lies at the heart of new wireless data applications and devices that allow for wireless e-mail and Internet access. Queries can be forwarded throughout the entire network and are usually uniquely identified. Responses usually come in the form of relevant user data, such as weather conditions, stock market quotes, and news headlines. Peer-to-peer applications are a popular type of distributed information sharing system that uses a query-response model.

Batch files are files that contain a sequence, or batch, of commands. Batch files are useful for storing sets of commands that are always executed together because you can simply enter the name of the batch file instead of entering each command individually. Traffic generated from batch files usually comes in the form of multiple services, requests, or file transfers. However, the nature of the traffic may be repetitive but not periodic, as in bursty traffic.

#### 4.1.1.3.3 Network Configuration

A node's *network configuration* is either static (determined *a priori*) or dynamic (changing during operation). Wireless networks with high security requirements will most likely require a static network configuration.

#### 4.1.1.3.4 Network Duration and Session Length

The *network duration* and *session length* parameters represent the expected length of time for network connectivity and for the session. Certain military operations may require their networks to be connected indefinitely, but only maintain short sessions of actual data transmission. These lengths could affect how frequently the keys in a security system need to be distributed or destroyed.

Two main factors that affect network duration and session length are a network topology's rate of change and the fraction and frequency of sleeping nodes. The rate of change of the topology is a function of node mobility and range. For example, a network with a high topological rate of change may contain nodes that are either highly mobile or have a node proximity larger than the node range. A high topological rate of change may lead to small network duration and session length values. The fraction and frequency of sleeping nodes represents the mean and variance of the percentage of nodes sleeping. A high fraction of sleeping nodes may mean a short session length and a long network duration.

#### 4.1.1.3.5 Overhead

The *overhead* parameter indicates the number of bytes or number of messages of overhead necessary to establish and maintain a connection with other nodes. It can include overhead required to initiate services after a route is discovered. Overhead is an important parameter that will affect wireless network performance and energy efficiency. Depending on the overhead value, lightweight security schemes may need to be used to minimize the impact on network and energy resources.

#### 4.1.1.3.6 Quality of Service (QoS)

The *quality of service* parameter determines the allocation of bandwidth and differentiated services required by applications in the wireless network. Different services in a wireless network may require different levels of security or entirely different security protocols. Bandwidth allocation will affect the efficiencies of security protocols and the overall impact of a security protocol on a wireless network.

#### 4.1.1.3.7 Communication Type

A wireless network can communicate via unicast, multicast, or broadcast. Wireless networks can also support two or more communication types for certain services or functions. Some security functions may not support broadcast communication; therefore it is crucial to know the *communication type* of a wireless network before applying a particular security protocol.

#### 4.1.1.3.8 External Networks

The *external networks* parameter indicates whether or not Internet access or some privately controlled network access is necessary for the nodes in the wireless network and for what purpose. Wireless networks requiring Internet access may need to utilize firewalls or a virtual private network (VPN).

### 4.1.2 Security Requirements

The security requirements or level of security for wireless networks are important for determining which security protocols to employ. Specific wireless network applications may require only a few security features against "casual intruders," while some may require elaborate security defenses against determined and capable adversaries that involve many protocols. Sections 4.1.2.1 through 4.1.2.5 describe individual security operational parameters.

#### *4.1.2.1 Availability*

*Availability* is the opposite of the ability to deny information. This criterion ensures the survivability of network services despite denial of service attacks. A denial of service attack could be launched at any layer of an ad hoc network. At the physical and media access control layers, an adversary could employ jamming to interfere with communication on physical channels. At the network layer, an adversary could disrupt the routing protocol and disconnect the network. At higher layers, an adversary could bring down high-level services. One such target might be the key management service, an essential service for any security methodology.

#### *4.1.2.2 Confidentiality*

*Confidentiality* ensures that certain information is never disclosed to unauthorized entities. Network transmission of sensitive information, such as strategic or tactical military information, requires confidentiality. Leakage of such information to enemies could have devastating consequences. Routing information must also remain confidential in certain cases, because the information might be valuable for enemies to identify and to locate their targets in a battlefield. To protect information from unauthorized disclosure, encryption is usually used. In addition, data structures in the operating system and even in the hardware itself may aid in this protection.

#### *4.1.2.3 Integrity*

*Integrity*, or soundness, guarantees that a message being transferred is never corrupted or, if it is, that it can be identified as being corrupted. A message could be corrupted because of benign failures, such as radio propagation impairment, or because of malicious attacks on the network. Information that needs to be constant or information that must only be modified by a certain authorized set of users must have the guarantee that it will not be modified by an unauthorized user. This can be accomplished through the use of cryptographic hashes or message digests.

#### *4.1.2.4 Authentication*

*Authentication* enables a node to ensure the identity of the peer node with which it is communicating. Without authentication, an adversary could spoof a node, thus gaining unauthorized access to resources and sensitive information and interfering with the operation of other nodes.

Networks can employ different authentication mechanisms, some unidirectional, such as 802.11b, and some bidirectional, such as Bluetooth. This parameter determines the authentication requirements needed for the wireless network. Key management protocols are usually applied when an authentication procedure is necessary. In addition, some security protocols use authorization as well as authentication.

#### *4.1.2.5 Non-Repudiation*

*Non-repudiation* ensures that the originator of a message cannot deny having sent the message. Non-repudiation is useful for detection and isolation of compromised nodes. When a node receives an erroneous message from a second node, non-repudiation allows the first node to accuse the second node using this message and to convince other nodes that the second node is compromised.

### 4.1.3 Example Categories of Wireless Network Applications

In this section, several popular wireless networks have been separated into different categories based on operational parameters. The categories are described below in Sections 4.1.3.1 through 4.1.3.4. Note that these categories are meant to illustrate the categorization of wireless network applications using the operational parameters and do not necessarily cover all possible types of wireless network applications. However, new wireless network applications may certainly be formed into new or existing categories using the operational parameters presented in Section 4.1.1.

#### *4.1.3.1 Category 1: Fixed Broadband Wireless Networks*

Fixed broadband wireless systems have long been used for voice and data communications, generally in backhaul networks[13] operated by telephone companies, cable television companies, and government agencies. Frequencies used typically range from 1 GHz to 40 GHz. Fixed broadband wireless networks can be used for almost anything for which a cable is used, whether the cable is an Ethernet cable or a fiber optic cable. Fixed broadband wireless systems are designed so that they emulate cable connections and they use the same type of interfaces and protocols, such as frame relay, Ethernet, and Asynchronous Transfer Mode (ATM). Most new development in fixed broadband wireless networks is data-centric, such as for Internet access, or is flexible in supporting both voice and data communications. Any application that operates over a wired network should be able to operate over a fixed broadband wireless network.

A type of a fixed broadband wireless network is a rapidly deployable broadband wireless network, for example, as described by Bostian, *et al.* [75]. These wireless networks bring communication and network services to areas without existing infrastructure and are typically used for disaster response, where natural or manmade forces have destroyed the area's infrastructure or where remote locations lack underlying infrastructure. The objective of these networks is to provide services to first responders and incident

[13] A backhaul connection is an internal infrastructure connection.

commanders to help them manage the disaster. The operational parameters of the rapidly deployable broadband wireless network are listed in Table 4.1. The composition of the nodes is, at minimum, one hub node and one remote node. The hub node is connected to existing network infrastructure and wirelessly transmits data to a remote node at the disaster site which may be several kilometers away. The remote node may in turn distribute network access and services to response units via a WLAN. The long range, fixed configuration, high data rates, and QoS requirements place this wireless network application in a unique category. Similar wireless network applications include wireless backhaul [76] or 802.16 networks.

**Table 4.1: Rapidly Deployable Broadband Wireless Operational Parameters**

| **Equipment** | |
|---|---|
| Energy Requirement | High (long duration) |
| User Interface | OS |
| End User Device | Workstation |
| **Network Topology** | |
| Network Size | Small (2-6) |
| Node Degree | Pair |
| Intermediate Nodes | Single-hop |
| Node Proximity | Remote (1 to 3 km) |
| Node Range | Far (1 to 3 km or more) |
| **Communication Characteristics** | |
| Link Capacity | 10 to 100 Mbps or higher |
| Traffic Pattern | Bursty (video, voice, messaging, etc.) |
| Network Configuration | Static |
| Network Duration | Hours to days |
| Session Length | Variable |
| Overhead | High (security, secondary nodes, services) |
| QoS | Very high (video, voice) |
| Communication Type | Unicast, multicast |
| External Networks | Required |

#### *4.1.3.2 Category 2: Wireless Local Area Networks*

The IEEE 802.11 standards define a wireless local area network [8]. The IEEE 802.11 standards can be compared to the IEEE 802.3 wired standard for the Ethernet local area network (LAN). The 802.11 specifications address both the physical (PHY) and media access control (MAC) layers.

WLANs are generally utilized as extensions to existing wired infrastructures and provide the interface between wireless clients and base stations or access points. Additionally, wireless clients may communicate in a standalone mode in ad hoc mode. Unfortunately, the use of wireless communications and the resulting mobility enabled by wireless communications introduce security issues that must be addressed. For an analytical description of potential vulnerabilities in IEEE 802.11, see Lough's dissertation [7].

Standards bodies and industry are constantly working on improving data rate, range, and security in wireless networking solutions.

Indoor WLANs normally use the IEEE 802.11a, b, or g PHY standard for providing wireless access. Services are usually Internet related originating from a fixed access point which is wired to existing infrastructure. WLAN hot spots are a slight variation of typical indoor WLANs. The differences between WLAN hot spots and ordinary WLANs are in the *equipment* and *network configuration* parameters. While WLANs may be configured dynamically or statically for security purposes, WLAN hot spots are almost always dynamically configured and consist almost entirely of mobile users. Example WLAN operational parameters are shown in Table 4.2. The required infrastructure, node mobility, and single-hop configuration are the key operational parameters that describe this category.

**Table 4.2: WLAN Operational Parameters**

| | |
|---|---|
| **Equipment** | |
| Energy Requirement | Low to high, or supplied |
| User Interface | None |
| End User Device | PDAs, laptops, PCs |
| **Network Topology** | |
| Network Size | Small to large (several to hundreds) |
| Node Degree | Pair |
| Intermediate Nodes | Single-hop |
| Node Proximity | Short to long (several hundred meters) |
| Node Range | Medium (up to 100 m) |
| **Communication Characteristics** | |
| Link Capacity | 11 Mbps or less |
| Traffic Pattern | Bursty |
| Network Configuration | Static (typically) |
| Network Duration | Minutes to hours |
| Session Length | Minutes to hours |
| Overhead | Medium |
| QoS | None |
| Communication Type | Unicast |
| External Networks | Required |

#### *4.1.3.3 Category 3: Mobile Ad Hoc Networks*

A mobile ad hoc network (MANET) is an autonomous system of multiple nodes that can communicate without infrastructure. Nodes that are not within radio range must communicate through intermediate nodes that act as routers. The network topology will change as nodes move. A MANET, as widely envisioned, consists of mobile nodes that are end user computing devices or terminals, such as notebook computers or PDAs. An alternative form of a MANET uses the wireless network as a backbone network for mobile platforms. Such a platform could be an edge router that is connected to multiple hosts via a subnet and to other subnets via the MANET backbone. The mobile platforms may be located in or on airplanes, ships, trucks, cars, or, perhaps, even people to connect a personal area network of small devices to a larger network.

The Virtual Operations Network (VON), investigated in a task of the Navy Collaborative Integrated Information Technology Initiative (NAVCIITI) at Virginia Tech[14], is a somewhat unique example of a MANET that serves as a backbone network for mobile platforms [77]. The project focuses on interoperability between heterogeneous networks that may belong to and be managed by different organizations, including allies and coalition partners. A group of naval vessels represent the mobile nodes in this wireless network. A gateway node at each ship will link to other ships which will create the MANET. Naturally, the security requirements for this wireless network application are, for many applications, very high. The operational parameters for the NAVCIITI VON are displayed in Table 4.3.

**Table 4.3: NAVCIITI VON Operational Parameters**

| **Equipment** | |
|---|---|
| Energy Requirement | High or supplied |
| User Interface | OS |
| End User Device | Control station, PCs |
| **Network Topology** | |
| Network Size | Small to Medium (less than a hundred) |
| Node Degree | Small |
| Intermediate Nodes | Single-hop or multi-hop |
| Node Proximity | Long to remote (up to several kilometers) |
| Node Range | Far (several kilometers) |
| **Communication Characteristics** | |
| Link Capacity | 64 Kbps to 11 Mbps |
| Traffic Pattern | Bursty (video, voice, sensor data) |
| Network Configuration | Static |
| Network Duration | Indefinite |
| Session Length | Mission time |
| Overhead | High (security, services) |
| QoS | Very high (video, voice, sensor data) |
| Communication Type | Unicast, multicast |
| External Networks | None |

A Wireless Soldier Network based on the Land Warrior System [78] falls into another unique military related application of MANETs. The operational parameters in Table 4.4 represent the Wireless Soldier Network. The wireless equipment, e.g., a PDA, a networked rifle, and a networked helmet, are linked together to form the MANET. The routing tables in the MANET are static since the equipment should never leave the

[14] The NAVCIITI project was funded by the U.S. Navy’s Office of Naval Research. More information on this particular NAVCIITI task is at http://www.irean.vt.edu/navciiti/.

soldier even though his location may change. Thus, some functions in this application might not need security, such as route discovery, and a reduction in overhead or resource consumption may be gained.

**Table 4.4: Wireless Soldier Network Operational Parameters**

| **Equipment** | |
|---|---|
| Energy Requirement | High (duration requirement) |
| User Interface | Sensors, OS, biometrics |
| End User Device | PDAs, guns, helmets, etc. |
| **Network Topology** | |
| Network Size | Small (devices on a soldier) |
| Node Degree | Small (2-6) |
| Intermediate Nodes | Single-hop |
| Node Proximity | Personal (less than 1 m) |
| Node Range | Short (1 to 10 m) |
| **Communication Characteristics** | |
| Link Capacity | Less than 10 Mbps |
| Traffic Pattern | Periodic, query/response |
| Network Configuration | Static |
| Network Duration | Minutes to hours |
| Session Length | Minutes to hours |
| Overhead | Medium (security) |
| QoS | None |
| Communication Type | Unicast, multicast |
| External Networks | None |

A more conventional use of MANETs would be in networks in pervasive or ad hoc computing environments [79-82], since MANETs can also be created for the duration of a meeting or conference presentation. If a presenter wishes to give contact information to the audience, the data can be sent and exchanged through the MANET. Such data exchanges in meetings or conferences are becoming commonplace. The wireless devices involved in this application tend to be portable, such as PDAs, notebook computers, or cellular telephones. Operational parameters for a pervasive network are listed in Table 4.5.

Similarities between these three MANET applications include the network size and node degree. MANETs do not scale well and only a small number of nodes can be in a network if it is to maintain high data rates. In addition, MANETs are usually created from devices that are battery powered and, thus, have a limited supply of energy. The duration of the sessions can also vary depending on the type of data transmitted. One other parameter common between the three applications is the capability of communications without the use of external networks, such as the Internet. This is the primary factor that differentiates MANETs from other categories. Security has to be established without the use of any centralized services.

**Table 4.5: Pervasive Network Operational Parameters**

| | |
|---|---|
| **Equipment** | |
| Energy Requirement | Medium (duration) |
| User Interface | OS |
| End User Device | PDAs, laptops |
| **Network Topology** | |
| Network Size | Small (2-6) |
| Node Degree | Small (2-6) |
| Intermediate Nodes | Single-hop |
| Node Proximity | Personal, PAN (1 to 10 m) |
| Node Range | Short (1 to 10 m) |
| **Communication Characteristics** | |
| Link Capacity | Less than 10 Mbps |
| Traffic Pattern | Query/response (objects, files, messaging) |
| Network Configuration | Dynamic |
| Network Duration | Minutes to hours |
| Session Length | Minutes to hours |
| Overhead | Low (requested data) |
| QoS | None |
| Communication Type | Unicast, multicast |
| External Networks | None |

#### *4.1.3.4 Category 4: Small Device Sensor Networks*

Recent advances in wireless communications and electronics have enabled the development of low cost, low-power, multifunctional sensor nodes that are small in size and communicate over short distances. These tiny sensor nodes have capabilities to sense, process data, and communicate. Typically they are densely deployed in large numbers and are prone to failures and frequent topology changes. They have limited power, computational capacity, bandwidth, and memory. As a result of the properties of such networks, traditional protocols cannot be applied. A sensor network is composed of a large number of sensor nodes that are densely deployed near the phenomenon. Sensor nodes that make up the sensor network are randomly deployed in inaccessible terrains, such as near a volcano, or as part of disaster relief operations, such as following a flood or fire, which means that sensor network protocols and algorithms must possess self-organizing capabilities and exhibit cooperative higher-level behavior. Sensor networks have wide applications in areas such as health care, military, collecting information in disaster prone areas and surveillance applications [83].

Small device sensor networks can be used in almost any environment. For example, small sensors may be embedded into the structure of a building to collect seismic data over a period of several months. Sensor networks may also be used for a short *network duration* and for single purposes. After a forest fire, for instance, sensors could be dropped into the area to collect temperature readings to determine the safety of any rescue missions. A network of small sensors can be described with the operational parameters in Table 4.6.

Small device sensor networks are typically deployed without any direct user interface or end user devices [83]. The energy requirements can be high because the sensors may need to operate for long periods of time. The nature of the data being collected may require hundreds of nodes, especially if a large area is being monitored. Furthermore, the sensors will typically be in close proximity to one another, or at least within their wireless communication range. The bandwidth of the sensor devices is usually limited to conserve energy and lower the cost of devices. Data is collected periodically and will usually be packaged efficiently to reduce the payload size of the packets. Transmissions usually occur at regular intervals either after each data collection or after the sensor has accumulated data from a series of collections. The session length is only as long as the time required to transmit one collection of data. However, the duration of the network can be indefinite, as the cost of reestablishing a network connection for each packet transmission would use more resources than just maintaining the network.

**Table 4.6: Small Device Sensor Network Operational Parameters**

| | |
|---|---|
| **Equipment** | |
| Energy Requirement | Low to high (duration requirement) |
| User Interface | None |
| End User Device | None |
| **Network Topology** | |
| Network Size | Small to very large (several to thousands) |
| Node Degree | Small to medium (2-20) |
| Intermediate Nodes | Multiple-hop |
| Node Proximity | Short (1 to 10 m) |
| Node Range | Short (1 to 10 m) |
| **Communication Characteristics** | |
| Link Capacity | Less than 1 Mbps |
| Traffic Pattern | Periodic |
| Network Configuration | Static or dynamic |
| Network Duration | Indefinite |
| Session Length | Short |
| Overhead | Low (periodic measurements) |
| QoS | None |
| Communication Type | Unicast, multicast, and broadcast |
| External Networks | None |

## *4.2 Incorporating Security*

In the first phase of this methodology, as described in Section 4.1, the operational parameters are identified and the wireless network of interest is categorized. In this second phase, appropriate security protocols are selected. Depending on the operational parameters of a wireless network category, many security protocols may be eliminated from the set of possible choices.

### 4.2.1 Operational Parameter Influence

Each operational parameter of a wireless network may have some influence on the suitability of a given security protocol. Some parameters have more of an effect than others on a certain factor related to the selection of a security protocol. Given the parameters associated with a particular wireless network category, it may be possible to restrict the number of security protocols that are appropriate for that category.

**Table 4.7: Equipment Influence**

| Equipment | Security Influence |
|---|---|
| Energy Requirement | Services, Complexity |
| User Interface | Complexity |
| End User Device | Complexity |

In Table 4.7, the operational parameters derived from the Equipment Parameter Group of a wireless network category would affect the complexity and, therefore, the strength of the algorithms involved in realizing a security protocol. A wireless network with a low *energy requirement* would be significantly affected by computationally intense algorithms that some strong security protocols employ. In addition, the number of security services may need to be restricted with a small *energy requirement*, because more services might consume more energy without much added security benefit. The *user interface* will also have an effect on the complexity of the algorithms in a security protocol. Passwords, tokens, and other user interactions can increase the randomness of keys used in a protocol. An *end user device* that has a high processing capability would be able to handle more complicated security algorithms with less of a performance penalty than devices with low processing resources.

**Table 4.8: Network Topology Influence**

| Network Topology | Security Influence |
|---|---|
| Network Size | Scalability |
| Node Degree | Scalability |
| Intermediate Nodes | Delay, Tunneling, Scalability |
| Node Proximity | Authentication |
| Node Range | Authentication |

The Network Topology Parameter Group impacts the scalability and authentication factors of security protocols, which are outlined in Table 4.8. A large *network size* implies that security protocols would need to be scalable and work with a large number of nodes efficiently and, preferably, without a considerable impact on network performance. A wireless network category with a large *node degree* would, likewise, require security protocols that are scalable, especially if public-key cryptography is used. Section 4.1.1.2.3 describes how the *intermediate nodes* parameter may influence security protocols. Authentication service inclusion may be determined by the *node proximity* and *node range* parameters. Low values in these two parameters may not require authentication and, thus, the wireless network may gain some potential performance improvement and energy savings.

**Table 4.9: Communication Characteristics Influence**

| Communication Characteristics | Security Influence |
|---|---|
| Link Capacity | Services |
| Traffic Pattern | Key Management |
| Network Configuration | Key Management, Tunneling |
| Network Duration | Key Management |
| Session Length | Key Management |
| Overhead | Services, Complexity |
| Quality of Service | Services |
| Communication Type | Key Management |
| External Networks | Firewalls, Tunneling |

As indicated in Table 4.9, the Communication Characteristics Parameter Group mainly influences the key management and number of services in a security protocol. The *link capacity*, *overhead*, and *quality of service* parameters affect the number of services a wireless network may employ without a substantial performance or energy penalty. The *traffic pattern*, *network configuration*, *network duration*, *session length*, and *communication type* parameters influence the key management properties of a security protocol. Finally, the *network configuration* and *external networks* parameters determine the tunneling requirement.

**Table 4.10: Security Factors Influence**

| Factors Influenced | Operational Parameters |
|---|---|
| Services | Energy Requirement, Link Capacity, Overhead, Quality of Service |
| Scalability | Network Size, Node Degree, Intermediate Nodes |
| Complexity | Energy Requirement, User Interface, End User Device, Overhead |
| Key Management | Traffic Pattern, Network Configuration, Network Duration, Session Length, Communication Type |
| Tunneling | Intermediate Nodes, Network Configuration, External Networks |
| Authentication | Node Proximity, Node Range |

The factors in the Security Parameter Group affected by the operational parameters are delineated in Table 4.10. Each operational parameter can be assigned positive or negative weights and then summed to “quantitatively” help determine the importance of the factors involved in a security protocol. Nevertheless, the factors influenced should be compared against the security requirements of a wireless network. For example, if a category has no *intermediate nodes*, a static *network configuration*, and no *external networks*, then tunneling would not be necessary for an applicable security protocol. However, if for some reason tunneling is required as part of the integrity and confidentiality security requirements, then that category will not gain any resource savings, because tunneling is necessary.

### 4.2.2 Security Integration Approach

A security protocol may be integrated into a wireless system for impact evaluation, as described in Section 4.3, through theoretical application, simulation, or direct application.

#### *4.2.2.1 Theoretical Application*

A security protocol can be mathematically modeled and parameterized. Performance or energy cost functions can then be expressed in terms of security parameters and wireless network characteristics (operational parameters). Each security protocol may have parameters that significantly affect wireless network resource consumption. Analytical solutions typically offer less accuracy than simulation, but are also less costly and time consuming.

#### *4.2.2.2 Simulation*

A wireless network application may incorporate different security protocols through simulation. This can be done entirely with software or hardware or a combination of both. Performing a simulation is more economically advantageous than actually implementing a design and testing it. The iterative process of designing, implementing, and analyzing can increase expenses for a project. Simulations can use the model created in the design phase for multiple experiments and analysis. A simulation can also provide results that are not experimentally measurable or would require many actual experiments.

However, a disadvantage in simulating is simulation errors. Programming the simulations using theories and algorithms will not guarantee accurate depictions of wireless network operations. Not everything may be accounted for in terms of an actual deployment. Thus, actual experimental results need to be verified against simulated results for the simulation to be generally accepted. If the two data sets compare, then the simulation design will have some credibility.

#### *4.2.2.3 Direct Application*

Actually constructing the wireless network and implementing a given security protocol would be a direct application of incorporating the protocol into the network. With this approach the experimental results would be more likely to closely match real world situations. Unfortunately, implementing security protocols for large scale networks may be expensive and time consuming, especially if the security protocol investigated results in poor network performance or energy efficiency.

## *4.3 Impact Evaluation*

Evaluating the resource impact of a security protocol is a vital part of this methodology. Metrics used for evaluating a security protocol are described in Section 4.3.1. Section 4.3.2 discusses suitability analysis for different security protocols applied to a wireless network category.

### 4.3.1 Metrics

The following subsections explain the metrics used in this methodology for evaluating the impact of a security protocol on a wireless network category. More information on these metrics can be found in a paper by Jones, *et al.* [48] and Request for Comments (RFC) 2501 [73].

#### *4.3.1.1 End-to-End Data Throughput and Delay*

Statistical measures of data routing performance, e.g., means, variances, and distributions, are important to measure the effect of security on the communications system. In terms of secure routing, these are the measures of a security protocol's effect on the underlying routing protocol or in the case of a secure routing mechanism, the efficiency of the mechanism.

#### *4.3.1.2 Efficiency and Overhead*

To achieve a given level of performance, two different policies can expend differing amounts of overhead, depending on their efficiency. Protocol efficiency may or may not directly affect data routing performance. If security control and data traffic must share the same channel, and the channel's capacity is limited, then excessive control traffic often impacts performance. Three metrics, described below, can be used to measure efficiency and overhead.

- *Mean data bits transmitted/data bits delivered* measures the efficiency, at the bit level, of delivering data within the network. Indirectly, it also gives the average hop count taken by data packets.

- *Mean control bits transmitted/data bit delivered* measures the efficiency, at the bit level, of the protocol in expending control overhead to deliver data. Note that this should include not only the bits in the control packets, but also the bits in the header of the data packets. In other words, anything that is not data is control overhead, and should be counted in the control portion of the algorithm.

- *Mean data and control bits transmitted/data packets delivered* captures the channel access efficiency of a protocol, as the cost of channel access can be high in contention-based link layers.

#### *4.3.1.3 Power and Energy Consumption*

Power and energy consumption metrics indicate how much power and energy a security protocol consumes when used in a wireless network. Power consumption would be most useful as a rate, e.g., Joules per second, to determine the battery lifetime of a system. Two different security protocols can then be compared to at least roughly determine which is more power or energy efficient. The measurements from mobile nodes in a given network can be used to maximize the lifetime of the network.

#### *4.3.1.4 Energy Consumed per Packet*

If the energy consumed per packet is minimized, then the total energy consumed is also minimized. Under light loads, the shortest-hop path will most likely result in a relatively low value for this metric. As network load increases linearly, the metric value may increase exponentially, as packets are routed around areas of congestion in the network. This metric can be used to compare security protocols at a packet level in terms of energy efficiency.

### 4.3.2 Suitability Analysis

In this research, the suitability of a security protocol to be used in a particular wireless network is determined by the level of security and the performance, e.g., the throughput, or energy efficiency. A security protocol that maximizes these two factors, for example, for a particular wireless network will yield a high suitability result. The suitability potential is the sum of weighted factors which describe the suitability for the security protocol. Using the metrics described in Section 4.3.1, security protocols may be analyzed in terms of performance and energy efficiency with respect to a given wireless network category. The operational parameters and metrics are inputs to the analysis and are examined to determine their influence on the wireless network. Using one approach, the security protocols and their relative metric values may be judged through a comparison matrix. The eigenvector of this matrix can be calculated to determine the weights for the input factors.

Using a theoretical approach to incorporate security protocols into a wireless network system would provide accurate results in terms formulas and algorithms. However, the actual impact of a security protocol may differ from analytical results. Simulation of the resource impact of a security protocol applied to a wireless network will yield results that are comparable to an actual deployment. The accuracy of a simulation will depend on its composition. Whether a simulation uses software, hardware, or both, and how many internal and external experimental factors are included will affect the results.

With a small set of manageable categories, like the ones in Section 4.1.3, appropriate security protocols can be found for each category. Subsequently, the second and third procedures of this methodology can be omitted for any wireless network application classified into the set. Thus, if a new wireless network application falls into an existing wireless network category, then suitable security protocols for that category should also be appropriate for the new application. In other words, the performance analysis may be streamlined since the application has already been classified and minimal simulation or experiments would be needed to assess the suitability of various security protocols.

### 4.3.3 Vulnerability Consideration

Once metrics have been measured from applied security protocols, vulnerabilities should be considered, if they exist. The Suitability Analysis process specifically takes into account performance and energy efficiency when choosing appropriate security protocols, but not security vulnerabilities. Even though algorithms of security protocols may be cryptographically strong and sound, if vulnerabilities around the algorithms exist, then the security protocol may be useless in some or even most cases. Disregarding brute force or physical security attacks, successful attacks against security protocols usually result from poor implementations of the protocol or a flaw in the security protocol [33, 34]. For a thorough methodology for discovering security vulnerabilities in wireless networks, refer to Lough's dissertation [7].

When comparing two or more security protocols applied on a category of wireless networks, the metrics may reveal one protocol to be more suitable than another. However, better suitability may come at a price of including vulnerabilities in the protocol. Resource efficiency versus the level of security need to be balanced in terms of the security requirements for the wireless network. Obviously, if there is not much difference in performance or energy efficiency, but a significant difference in level of security, then the most suitable security protocol should be the one that offers the most security.

### *4.4 Methodology Operation Summary*

Properly utilizing the methodology requires a series of procedures to be followed. These were described earlier and are summarized in this section. The diagram in Figure 4.1 shows the flow through the different processes of the methodology. Given a wireless network application, the operational parameters and security requirements are first determined. From this information the wireless network is placed into an existing category or a new category is created containing the wireless network.

If there are no suitable security protocols for the category, then more steps are needed before any security protocol is applied to the wireless network. The security requirements analysis initially filters out any security protocols that do not meet the determined requirements. The operational parameters are then used to reduce the number of potential security protocols that need to be integrated for impact evaluation. Once individually integrated in separate instances of the wireless network application, the metrics are used to measure the impact of each security protocol. Vulnerabilities are also identified at this phase of the methodology. The measured results and vulnerabilities, if any, are used for the suitability analysis procedure. In this procedure, the security protocols are finally compared to determine the most suitable protocol for the wireless network application.

If a wireless network category has been identified, then suitable security protocols for that category should also be appropriate for a new application falling into that category. In this case, the security incorporation and impact evaluation processes may be simplified or even removed, as explained in the suitability analysis procedure in Section 4.3.2. If no suitable security protocols are found, then the operational parameters and security requirements can be redefined and the entire process can begin from choosing the categories as shown in Figure 4.1.

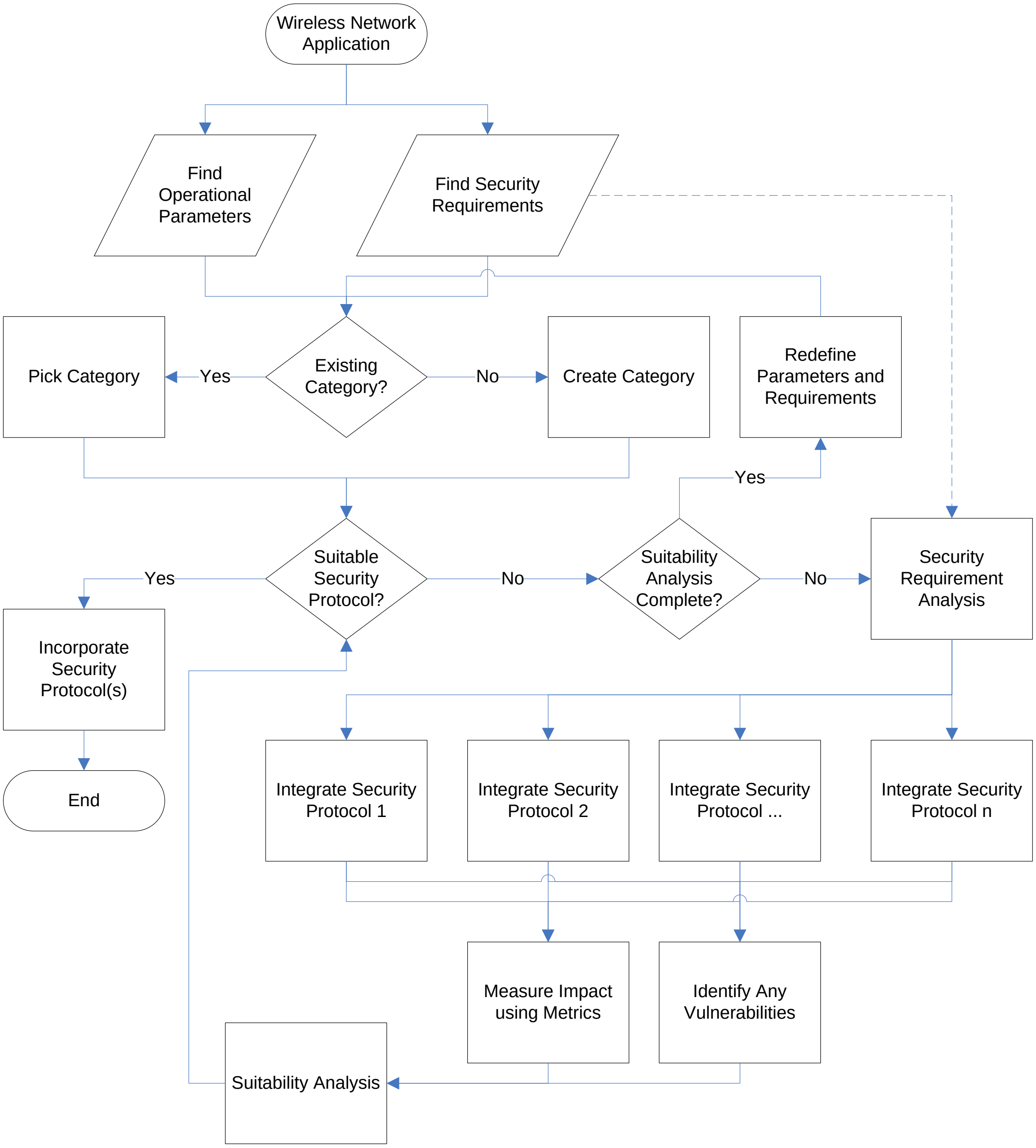

Wireless Network Application
Find Operational Parameters
Find Security Requirements
Pick Category
Yes
Existing Category?
No
Create Category
Redefine Parameters and Requirements
Yes
Yes
Suitable Security Protocol?
No
Suitability Analysis Complete?
No
Security Requirement Analysis
Incorporate Security Protocol(s)
End
Integrate Security Protocol 1
Integrate Security Protocol 2
Integrate Security Protocol ...
Integrate Security Protocol n
Measure Impact using Metrics
Identify Any Vulnerabilities
Suitability Analysis

Figure 4.1. Flowchart of methodology operation.

## *4.5 Case Study: Small Device Sensor Network*

To illustrate the use of the methodology, a case study is presented here. This case study considers selection of security protocols for a sensor network of small sensor nodes, as described briefly in Section 4.1.3.4.

### 4.5.1 Assumptions and Experimental Setup

The purpose of this case study is to determine if IPSec AH or IPSec ESP is a suitable security protocol for a small device sensor network. Several assumptions were necessary to limit the scope of the case study. The intent of these limiting assumptions was to keep the simulation complexity manageable.

The selection of a particular evaluation technique can significantly impact the outcome of a performance evaluation. Three possible techniques of performance evaluation are analytic, simulation, and measurement. As discussed in Section 4.2.2, these methods differ in terms of accuracy, cost, and required time. Based upon these factors, simulation was the most appropriate technique for this case study. Measurement was ruled out as a feasible technique based upon both cost and required time. Analytical solutions typically offer less accuracy than simulation, but are also less costly and time consuming. In this case, the cost of simulation was negligible because the software required was already available. Therefore, simulation was used to conduct this performance analysis.

The ns-2 network simulation tool[15] was used in this case study. Code was written to create new ns-2 applications and protocols. A network configuration of three stationary wireless nodes with a two-hop configuration was used. The network topology is shown in Figure 4.2. One end node continuously transmitted data intended for the other end node. However, since the two end nodes were not directly connected, the middle node always forwarded the data. Only constant bit rate (CBR) User Datagram Protocol (UDP) traffic was used. Ten hours of time were simulated for each run of the sensor network simulation model, with ten replications for each of the three experiments. A correlogram [84] was used to determine the minimum lag such that observations separated by this lag are approximately uncorrelated. The models seemed to reach steady state in less than 50 observations, thus a warm-up period of 600 seconds was deemed more than enough for each replication. Each device began with 1,000 Joules (J) of energy before the simulation. Idle power consumption and processing power consumption were not considered in the simulation.

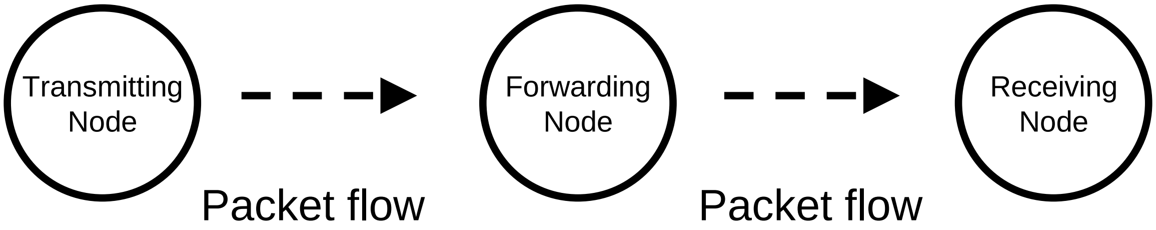


Figure 4.2. Small device sensor network configuration.

The input parameters for this experiment are listed in Table 4.11. The transmission rate was chosen as 128 bits per second to represent the low power and low data rate signals sent by small sensor devices. The packet length signifies the data payload of each transmission. For small wireless sensors, 16 bytes was deemed more than enough to transmit periodic information updates to the receiver node. The packet header length is the security overhead needed for each packet, which is 52 bytes for IPSec AH and ESP

[15] The Network Simulator (ns-2) is a discrete event simulator targeted at network simulation, especially for research. More information is available at http://www.isi.edu/nsnam/ns/.

[28, 29]. The delay factor is the number of seconds required to secure each byte of plaintext imposed by the cryptographic functions of IPSec. The processing delay is simply the product of the delay factor and the packet length. This parameter represents the additional delay before each packet transmission.

**Table 4.11: Parameter Values for No Security, IPSec ESP, and IPSec AH Cases**

| Input Parameters | None | IPSec ESP | IPSec AH |
|---|---|---|---|
| Transmission Rate (bps) | 128 | 128 | 128 |
| Packet Length (bytes) | 16 | 16 | 16 |
| Packet Header Length (bytes) | 0 | 52 | 52 |
| Delay Factor (sec/byte) | 0 | 2.7796250E-07 | 5.94370E-08 |
| Processing Delay (ms) | 0 | 0.00444740 | 0.0009509920 |

### 4.5.2 Categorization

The sensor network was first classified into a category based on its operational parameters. For the purposes of this illustrative case study, I derived the values of these operational parameters using what I considered to be typical sensor network deployment characteristics. The operational parameters are specified in Table 4.6 and were discussed in Section 4.1.3.4. The security requirements are given in Table 4.12.

The security requirements of availability, confidentiality, and integrity are optional in sensor networks. This reasoning can be justified by the number of nodes in the network and the importance of the data collected. Having a large number of sensors has advantages. If some sensors fail to operate, cannot transmit their collected data, or transmit corrupted data, the overall outcome may not significantly be affected. Whether or not the data requires privacy protection also depends on the type of data being collected. Authentication is useful because it confirms that data is transmitted or received by nodes within the network.

**Table 4.12: Small Device Sensor Network Security Requirements**

| Security Parameters | Requirement |
|---|---|
| Availability | Optional |
| Confidentiality | Optional |
| Integrity | Optional |
| Authentication | Required |
| Non-Repudiation | None |

### 4.5.3 Results

Three simulation experiments were performed and analyzed. The first experiment was a simulation of the sensor network described in Section 4.5.1 without any security mechanisms included. The second experiment was a simulation of the sensor network with IPSec ESP incorporated. The last experiment was with IPSec AH integrated into the sensor network. Verification procedures include examining the trace files of the simulations, checking code to ensure proper operation, and visually checking the ns-2 running models for any errors.

The results of these simulations are presented in Table 4.13. The throughput metric represents the speed at which the sensors receive and process data. The offered load metric can be associated with utilization of the bandwidth. The average delay is a measure of latency between nodes. Finally, the energy metric denotes the average of the total energy consumptions for each run.

**Table 4.13: Simulation Results**

| **Output Metrics** | **None** | **IPSec ESP** | **IPSec AH** |
|---|---|---|---|
| Throughput (bps) | 127 | 30 | 30 |
| Offered Load | 1.0000 | 0.2353 | 0.2353 |
| Avg. Delay (seconds) | 0.0039 | 0.0048 | 0.0048 |
| Avg. Energy Consumed (J) | 48.69216633 | 14.64116467 | 14.64116467 |

The simulation can be validated analytically, since the sensor network topology consists of only three nodes. Without using security, a throughput close to the maximum data rate can be achieved. Approximately one packet is sent every second without using IPSec. The average delay is slightly less than 4 ms and may be a result of forwarding delay. When using IPSec, the packet size increases from 16 bytes to 68 bytes. Since the transmission rate can only handle about 16 bytes per second, a packet secured by IPSec requires more than four seconds to be completely received by the receiving node. The offered load can be found by dividing the size of the original packet by the size of the secured packet, with the quotient being 0.235294. This value is close to the offered load values found through the simulations. Multiplying the offered load by the data rate would result in the throughput. The calculated throughput is approximately 30.12 bps, which is less than a 1% difference between the throughput results from the simulations.

### 4.5.4 Conclusions

IPSec ESP is deemed to be more suitable than IPSec AH for this small device sensor network because the results show that it has the same “cost” in terms of throughput reduction, delay increase, and energy consumption, yet offers more security. The offered load in both cases was 0.2353, while without security the offered load is close to the maximum data rate. After incorporating the security protocols, the throughput, average delay, and average energy consumed remained about the same for the two cases using IPSec. Thus, IPSec ESP would be a more suitable protocol because more security is provided than with IPSec AH. Even though the overall performance of the sensor network has diminished and IPSec has reduced the wireless network’s effectiveness in sensing data, the requirements have been met and the network is considered reasonably secure.

An interesting result is the average energy consumed during the simulation. Intuitively the energy consumed on the wireless device should be significantly higher when utilizing a security protocol. However, all the simulations ended after a certain time period and not after transferring a specified amount of data. Since the throughput was reduced, the data transmitted were also reduced, thus, less energy was consumed. Another way to simulate the sensor network, for possible future study, would be to run simulations until a certain amount of data is transferred. Additionally, the energy consumed from processing should also be included in future work. If energy consumption from processing is included, the simulation results may be significantly different since ESP requires more processing time and energy because the payload must be encrypted and decrypted.

### *4.6 Summary*

This chapter describes the methodology used for this research and represents the preliminary work for this dissertation. The methodology consists of three steps: (1) wireless network classification, (2) security protocol incorporation, and (3) impact evaluation. A wireless network is classified through operational parameters. Security protocols can be integrated into the wireless networks and then undergo impact evaluation through theoretical analysis, simulation, or direct application. Application of this methodology to the selection of a security mechanism for a small device sensor network was also presented.

# Chapter 5. Design of a Context Aware and Adaptive Security Manager

The previous chapter describes a methodology meant for users or system developers to determine suitable security protocols for different categories of wireless networks. This chapter describes a realization of a mechanism to select a security scheme at run-time. This mechanism is based on the methodology and is geared towards a specific wireless network environment. An application is designed to adapt to different situations, denoted by the context, and by selecting a suitable security protocol. In other words, the application performs the processes of the methodology as needed for a given situation.

This chapter discusses the design of our application, the Context aware and Adaptive Security Manager (CASM[16]). First, Section 5.1 briefly reviews the premises of contexts and context aware applications and how they relate to CASM. Section 5.2 then describes the specific wireless networking environment for which CASM is designed. Next, Section 5.3 presents an analysis of security performance and energy costs. Various block ciphers are considered in this analysis for use in CASM. Section 5.4 gives a brief high-level overview of CASM. Then, Section 5.5 defines the contexts that are used in CASM. The decision making process is discussed in Section 5.6, and the security procedures used in CASM are outlined in Section 5.7. Finally, Section 5.8 summarizes the overall design.

## *5.1 Background*

Context is defined in this work as information that can be used to describe the situation of an entity. Schilit, *et al.* [85] use "where you are," "who you are with," and "what resources are nearby" as aspects of context. Dey and Abowd [86] state that location, identity, time, and activity are the main dimensions of context. In CASM, we use six different components or dimension of context representing the system device, network, and environment. These are described in Section 5.5.2.

Context aware applications rely on contexts to provide manual or automatic triggers to perform certain tasks. What sets CASM apart from most context aware applications is that we focus only on adapting security instead of general-purpose tasks, for example related to human-computer interfaces. CASM requires context to provide itself with relevant information to perform its task of appropriately securing shared objects. In other words, the contexts are essential for CASM to make decisions on using the most suitable security protocols in different situations. Examples of other context aware applications and systems are discussed in references [85-88].

Sun, *et al.* [67] list some functional requirements for the design of a context aware system. These requirements include context collection, context storage and management, context

[16] Pronounced as "chasm."

subscription and delivery, and context analysis and composition ability. We apply these functional requirements to the design of CASM.

### 5.2 Environment

The pervasive collaboration project at Virginia Tech, mentioned earlier in Section 3.3, assumes a pervasive embedded network environment. The project objectives are to investigate session and content management, ad hoc session access control, and Microsoft® Windows CE and Pocket PC usage in pervasive environments. The architecture consists of a *global space* representing a virtual world that in turn houses *session spaces*. Each session could be a single meeting of several pervasive devices. Pervasive devices can either belong to a user or provide standalone services. User devices can share or acquire objects, which we may define as files, Uniform Resource Locators (URLs), or services.

The pervasive collaboration environment depicted in Figure 5.1 is an example of how wireless devices can interact with each other in sessions and share objects, such as location information, digital images, audio recordings, and video clips. Currently our prototype environment enables PDAs and notebook computers to communicate with each other via IEEE 802.11b interfaces in either peer-to-peer (P2P) or infrastructure mode.

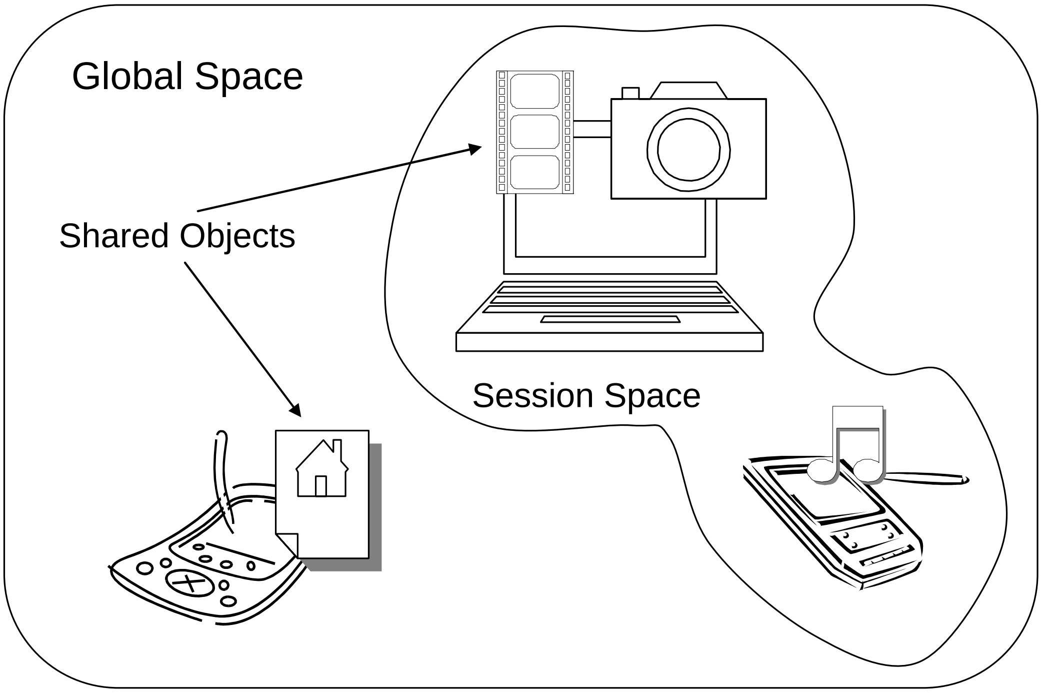


Figure 5.1. Pervasive collaboration environment.

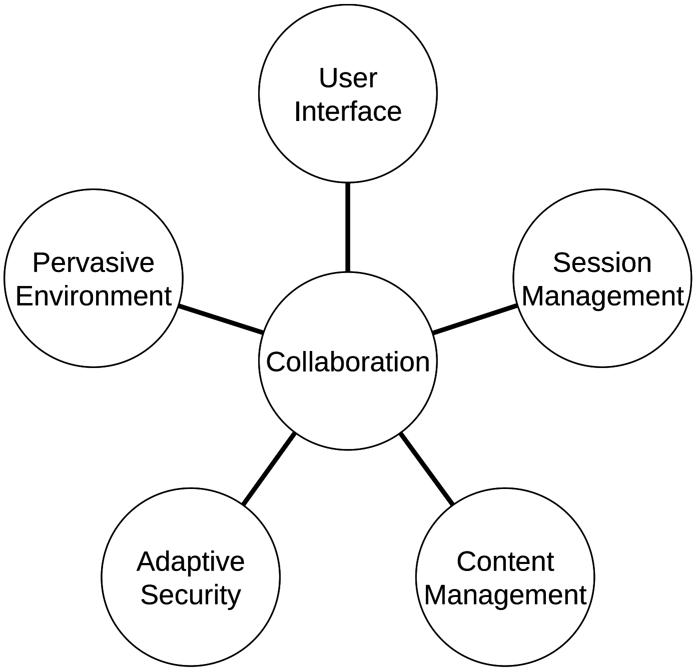


Figure 5.2. Pervasive collaboration project components.

The pervasive collaboration project has five components, which are shown in Figure 5.2. The user interface allows users to easily share and interact with other devices in the network. The session management component controls the addition and removal of entities to and from sessions. Location and publishing of objects are controlled by the content management component. The pervasive environment allows interaction with different devices. Finally, the adaptive security component controls system parameters and secures data based on context. The design of the adaptive security component, which is identified here as CASM, is the focus of this chapter.

### 5.3 An Analysis of Resource Costs for Security

Information privacy has become a major concern for all users, even personal users. Thus, our pervasive collaboration applications require capabilities for secure wireless transmission. A variety of algorithms are available for a user to encrypt data and prevent eavesdroppers from obtaining critical or private information. However, the execution of encryption algorithms consumes both time and energy. While certain encryption algorithms may be less vulnerable to compromise than others, constantly using cryptographically strong algorithms may result in severely reduced lifetimes for battery-powered devices such as PDAs. In other words, utilizing stronger encryption algorithms may consume more energy and drain the PDA battery faster than less secure algorithms. Due to the processing requirements and the limited computing power in many PDAs, using strong cryptographic algorithms may also significantly increase the delay between data transmissions.

The analysis presented here answers questions regarding energy consumption and execution time for various encryption algorithms executing on a PDA platform with the goal of helping software and system developers design more effective applications and systems and of allowing end users to better utilize the capabilities of PDA devices. The knowledge gained from this study also assists in the design of CASM. In particular, we experimentally measure and compare the energy consumption and computation time for four different block cipher encryption algorithms – RC2, Blowfish, XTEA, and AES – executing on contemporary PDAs. We measure energy consumption for one device and latency and throughput for three different devices. The experiments consider different transfer sizes from one kilobyte to one megabyte. Based on the results, we develop observations that are intended to increase the awareness of and insight into the costs associated with using encryption algorithms.

The objective of this study is to determine the time and energy costs associated with specific encryption algorithms for PDAs. Several assumptions are necessary to limit the scope of this objective. The intent of these assumptions is to keep the implementation and analysis complexity manageable. Moreover, many case studies would be needed to consider all of the issues involved with different encryption algorithms and systems. We present a somewhat simple, but informative, set of case studies involving four encryption algorithms on three different PDA platforms, to gain insight into the cost of utilizing encryption algorithms in a PDA environment.

### 5.3.1 Assumptions and Experimental Setup

The selection of a particular evaluation technique can significantly impact the outcome of a performance evaluation study. Three techniques for performance evaluation are analytical, simulation, and measurement. These methods differ in terms of accuracy, cost, and required time. Based on these factors, experimental measurement was the most appropriate technique for this case study. Analytical evaluation was ruled out due to the precision required in analytical models and discrepancies between different types of PDAs. Simulations typically offer less accuracy than actual measurements, but are also less costly and time consuming. In this case, the cost of implementing the experiments was relatively low because the required hardware and parts of the software base are readily available. Therefore, actual measurements with controlled workloads were used in this performance analysis.

Only block ciphers were considered in this research due to their frequent use, encryption speed, and ease of implementation. The methodology and, in a general sense, the results of this study are still applicable to other ciphers, algorithms, and security protocols with appropriate modification. More detailed consideration of techniques other than block ciphers is a possible area of future research. The study focuses on the resource implications of block ciphers. Cryptanalysis of the ciphers is not part of this research, but, of course, vulnerabilities and other cryptographic weaknesses must be considered by users and system designers.

Three commercially available PDAs running two major versions of Microsoft's Pocket PC (PPC) were used for the experiments, the Compaq (now Hewlett-Packard) iPAQ 3850, the Hewlett-Packard iPAQ 4150, and the Dell Axim X30. These devices were chosen because of their availability and differences in architecture and operating system (OS) versions. Relevant specifications of the Pocket PCs used in this study are listed in Table 5.1. The iPAQ 3850 is the oldest device and uses an Intel® StrongARM® processor. The more contemporary HP iPAQ 4150 and Dell Axim use Intel XScale® processors. For its power source, the iPAQ 3850 uses a lithium polymer battery while the other two PDAs use lithium ion batteries. Latency and throughput were determined for all three devices. Energy consumption was examined only for the HP iPAQ 4150 because power measurements are difficult for the other devices that do not have a removable battery or any other direct way to access the battery terminals. Note that the research methodology is not limited to these devices and may be extended to include other handheld devices, such as other PPC devices, PalmOS PDAs, Blackberry PDAs, smart phones, or even embedded devices for sensor networks.

**Table 5.1: Specifications of Devices Used in Experiments**

| Device | CPU | Battery | OS[17] |
|---|---|---|---|
| HP iPAQ 3850 | 206 MHz Intel StrongARM | 1400 mAh | MS PPC 2002 |
| HP iPAQ 4150 | 400 MHz Intel PXA255 | 1000 mAh | MS PPC 2003 |
| Dell Axim X30 | 624 MHz Intel PXA270 | 950 mAh | MS PPC 2003 SE |

We used the Microsoft's Visual Studio .NET and Compact Framework (CF) as the software development environment. Code was written in C# to create the applications and block ciphers for the experiments. Rather than creating implementations "from scratch," existing block cipher implementations [89-92] written in C, C++, and C# were ported for use in this study. Validation was needed to determine if the block ciphers function correctly. To verify proper implementation, the block ciphers were applied to their corresponding standard test vectors.

**Table 5.2: Block Cipher Operating Parameters for the Experiments**

| Block Cipher | Key Size (bits) | Block Size (bits) | Rounds |
|---|---|---|---|
| RC2 | 40 | 64 | 18[18] |
| Blowfish | 448 | 64 | 16 |
| XTEA | 128 | 64 | 64[19] |
| AES | 256 | 128 | 14 |

The operating parameters for the block ciphers in our experiments are presented in Table 5.2. Most of the parameters, such as key size and number of rounds, were maximized when using the ciphers for our experiments. Since longer keys and more rounds require

[17] Microsoft Pocket PC 2002, 2003, and 2003 Second Edition

[18] 16 mixing and 2 mashing

[19] 64 Feistel rounds, or 32 "cycles"

more time and effort to attack the cipher, this could enable us to measure the performance and energy consumption when the algorithm offers the maximum security. The block ciphers operated in electronic code book (ECB) mode [34]. ECB mode was mainly chosen due to its straightforward implementation. The ECB mode does not require any additional operations for each algorithm to encrypt a file. Cipher Block Chaining (CBC) overcomes the problems of repetition and order independence in ECB mode, but requires extra memory and operations involving an Initial Vector (IV). Cipher FeedBack (CFB) and Output FeedBack (OFB) modes treat plaintext as a stream of bits, however, we do not use stream oriented data for our experiments. Moreover, the objective is to compare the inherent algorithms of the block ciphers not the different cipher operating modes.

For the plaintext data to be encrypted, five different files of different sizes were generated with random content. File sizes in bytes are $2^{10}$ (1,024 bytes or 1 KB), $2^{12}$ (4,096 bytes or 4 KB), $2^{15}$ (32,768 bytes or 32 KB), $2^{17}$ (131,072 bytes or 128 KB), and $2^{20}$ (1,048,576 bytes or 1 MB). Each plaintext file was encrypted 100 times with each block cipher using the operating parameters specified in Table 5.2. There are twenty combinations of different file sizes and different block ciphers. For example, one combination uses AES with a 256-bit key, 128-bit block, and 14 rounds to encrypt $2^{12}$ bytes (4 KB) of plaintext data 100 times.

For each combination, the selected file was loaded into the PDA's memory. The unencrypted data blocks were kept statically in memory for the duration of the experiments. Static keys were used for encrypting the files. In addition, the encrypted memory contents were flushed after each encryption process to eliminate the effects of caching. To minimize the current drawn from the battery by other causes, each Pocket PC device used its dimmest backlight setting, any wireless devices were disabled, and all other active background programs were terminated before running each test.

### 5.3.2 Metrics and Measurement Methods

Evaluating the impact of encryption algorithms on processor and battery resources is the focus of this study. The following sections explain the metrics and methods used in the study. The metrics below have been adapted from metrics in the methodology, described in Section 4.3.1.

#### *5.3.2.1 Encryption Latency and Encryption Throughput*

Metrics for block cipher performance indicate the speed and efficiency of encrypting plaintext. To measure the encryption latency, the query performance frequency and query performance counter functions [93] were invoked[20] from the "CoreDll.dll" library in the operating system. These functions measure the frequency and current clock value of the CPU. The query performance frequency function was used to determine the CPU frequency of the device. The query performance counter function was called before and after each encryption process to measure elapsed clock cycles. The difference between the counter values was divided by the CPU frequency to obtain the encryption time in seconds. Encryption latency represents the encryption time per byte or file. Encryption

[20] The .NET platform invoke facility (P/Invoke) can expose functions in any DLL.

throughput is calculated by dividing the number of bytes encrypted by the time required to encrypt that number of bytes.

### *5.3.2.2 Energy Consumption*

Other metrics indicate how much energy is consumed by a block cipher for encryption. Two different ciphers can be compared (at least roughly) to determine which is more power or energy efficient. To measure the current power level, the battery of a device is first removed. A resistor is placed in series with the battery and the device. A multimeter is used to measure the voltage drop across the resistor.

We conducted power measurements using an HP iPAQ 4150, a very low resistance (0.025 Ω) precision resistor in series between the battery and the device itself, and an Agilent 3458A 8½ Digit Multimeter, as shown in Figure 5.3. It would have been preferable to use a benchtop power supply instead of the battery, but due to difficulties with the iPAQ's battery detection mechanisms, the battery was used instead. However, for the fluctuations of current for this iPAQ, battery voltage variations were negligible, less than 1%. The multimeter measured the voltage across the resistor 10,000 times per second. The resulting voltage measurements were then multiplied by the 4.1 V input voltage and divided by the 0.025 Ω resistance to calculate the power values. This procedure is indicated by Equation 5.1.

$$P(t) = V(t) \cdot V_{input} / R \tag{5.1}$$

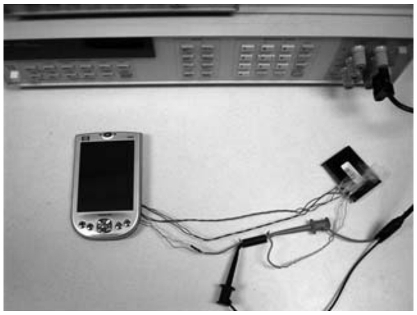


Figure 5.3. Experimental setup to measure energy consumption.

Energy consumption for each encryption task, $E_{task}$, is computed using Equation 5.2.

$$E_{task} = \sum_{i=0}^{n} [P(t_i) - P_{idle}] \cdot T \tag{5.2}$$

An encryption task begins at time $t_0$, which is the time when the measured power level significantly increases as a cipher begins encrypting a file. The task ends at time $t_n$, which is the time when the measured power level significantly decreases as the task completes encrypting the file. As indicated in Equation 5.2, $E_{task}$ is based on the $n$+1 multimeter measurements from $t_0$ to $t_n$. Each measured power value during the encryption task, $P(t_i)$, $i$ = 0,…, $n$, is reduced by the average power level measured when the device is idle, $P_{idle}$. The sum of the adjusted power values from $t_0$ to $t_n$ is multiplied by the sampling interval, T = 100 $\mu s$ (since there are 10,000 samples per second), to determine $E_{task}$, the energy consumed by the task. A MATLAB script was written to process the data collected from the multimeter.

Note that the interval $t_n - t_0$ offers an alternative way to calculate encryption time. This method is not used, however, since it is not as accurate as the procedure using clock cycle measurements that was described in Section 5.3.2.1. Results computed using the two methods do generally agree as described in Section 5.3.3.

#### *5.3.2.3 Energy-Latency Product and Throughput/Energy Ratio*

The power measurements and encryption performance results can be combined to determine the energy-latency product and the throughput/energy ratio. These metrics can be used to compare the block ciphers in terms of energy efficiency at the byte or packet level.

The energy-latency product is the encryption latency multiplied by power and represents the amount of energy required for the encryption task. It can be divided by the number of bytes encrypted to determine the Joules of energy consumed per byte of data. This can be thought of as the energy cost per unit of data encrypted.

The throughput/energy ratio is the encryption throughput result divided by power and indicates the number of bytes encrypted per joule of energy expended. This can be thought of as the benefit or utility for encryption that can be obtained from a joule of energy.

### 5.3.3 Results and Discussion

#### *5.3.3.1 Latency and Throughput*

The experiments to determine latency and throughput consisted of twenty different combinations of cipher algorithm and file size, as described in Section 5.3.1. For each combination, the specific file encryption task was replicated 100 times. Using the methods described in Section 5.3.2.1, the encryption times were calculated for each task and then averaged for the 100 replications. Encryption times were also calculated using the hardware measurements from the multimeter on the iPAQ 4150 for general verification. Even though Table 5.3 shows that the hardware measurements generate relatively similar encryption times, the software measurements from the clock cycle function calls are considered to be more accurate for computing encryption times and are used for the rest of the analysis. Average encryption times for each combination of cipher scheme and file size are used to compute average encryption latency and average encryption throughput for each of the three devices. These results for latency and throughput are shown in Tables 5.4 and 5.5, respectively. Variance in the measurements typically fell under 0.1% of the average values. The algorithm with the largest variance in measurements was XTEA, which had variances approximately 0.5% of the average values. These were caused by a few outliers in the raw data.

**Table 5.3: Average Encryption Times from Software and Hardware Measurements for the iPAQ 4150**

| Cipher | File Size (bytes) | Software (s) | Hardware (s) | % Difference |
|---|---|---|---|---|
| RC2 | $2^{10}$ | 0.0020 | 0.0021 | 6.55% |
| | $2^{12}$ | 0.0067 | 0.0071 | 6.87% |
| | $2^{15}$ | 0.0496 | 0.0497 | 0.35% |
| | $2^{17}$ | 0.1983 | 0.1962 | 1.06% |
| | $2^{20}$ | 1.5799 | 1.5614 | 1.17% |
| Blowfish | $2^{10}$ | 0.0066 | 0.0066 | 1.42% |
| | $2^{12}$ | 0.0090 | 0.0091 | 0.89% |
| | $2^{15}$ | 0.0307 | 0.0313 | 1.86% |
| | $2^{17}$ | 0.1117 | 0.1097 | 1.81% |
| | $2^{20}$ | 0.7973 | 0.8276 | 3.80% |
| XTEA | $2^{10}$ | 0.0034 | 0.0036 | 5.45% |
| | $2^{12}$ | 0.0139 | 0.0132 | 5.06% |
| | $2^{15}$ | 0.1077 | 0.1067 | 0.98% |
| | $2^{17}$ | 0.4330 | 0.4290 | 0.91% |
| | $2^{20}$ | 3.4615 | 3.4243 | 1.07% |
| AES (Rijndael) | $2^{10}$ | 0.0110 | 0.0114 | 2.83% |
| | $2^{12}$ | 0.0420 | 0.0420 | 0.06% |
| | $2^{15}$ | 0.3283 | 0.3260 | 0.73% |
| | $2^{17}$ | 1.3115 | 1.3042 | 0.55% |
| | $2^{20}$ | 10.4708 | 10.3645 | 1.02% |

**Table 5.4: Average Block Cipher Encryption Latency**

| Cipher | File Size (bytes) | Encryption Latency (μs/byte) | | |
|---|---|---|---|---|
| | | iPAQ 3850 | iPAQ 4150 | Axim X30 |
| RC2 | $2^{10}$ | 3.24 | 1.95 | 1.45 |
| | $2^{12}$ | 2.60 | 1.63 | 1.13 |
| | $2^{15}$ | 2.45 | 1.51 | 0.99 |
| | $2^{17}$ | 2.36 | 1.51 | 1.01 |
| | $2^{20}$ | 2.32 | 1.51 | 1.01 |
| Blowfish | $2^{10}$ | 10.08 | 6.40 | 4.65 |
| | $2^{12}$ | 3.68 | 2.20 | 1.46 |
| | $2^{15}$ | 1.52 | 0.94 | 0.64 |
| | $2^{17}$ | 1.25 | 0.85 | 0.59 |
| | $2^{20}$ | 1.10 | 0.76 | 0.51 |
| XTEA | $2^{10}$ | 11.57 | 3.32 | 2.31 |
| | $2^{12}$ | 11.40 | 3.39 | 2.29 |
| | $2^{15}$ | 11.88 | 3.29 | 2.23 |
| | $2^{17}$ | 11.79 | 3.30 | 2.24 |
| | $2^{20}$ | 12.36 | 3.30 | 2.24 |
| AES (Rijndael) | $2^{10}$ | 19.82 | 10.78 | 7.19 |
| | $2^{12}$ | 19.88 | 10.24 | 6.92 |
| | $2^{15}$ | 21.63 | 10.02 | 6.54 |
| | $2^{17}$ | 21.78 | 10.01 | 6.54 |
| | $2^{20}$ | 20.81 | 9.99 | 6.57 |

**Table 5.5: Average Block Cipher Encryption Throughput**

| Cipher | File Size (bytes) | Encryption Throughput (KBps) | | |
|---|---|---|---|---|
| | | iPAQ 3850 | iPAQ 4150 | Axim X30 |
| RC2 | $2^{10}$ | 301 | 501 | 675 |
| | $2^{12}$ | 376 | 598 | 861 |
| | $2^{15}$ | 398 | 646 | 984 |
| | $2^{17}$ | 415 | 645 | 968 |
| | $2^{20}$ | 421 | 648 | 970 |
| Blowfish | $2^{10}$ | 97 | 153 | 210 |
| | $2^{12}$ | 266 | 445 | 669 |
| | $2^{15}$ | 644 | 1043 | 1526 |
| | $2^{17}$ | 780 | 1146 | 1657 |
| | $2^{20}$ | 891 | 1284 | 1914 |
| XTEA | $2^{10}$ | 84 | 294 | 423 |
| | $2^{12}$ | 86 | 288 | 426 |
| | $2^{15}$ | 82 | 297 | 438 |
| | $2^{17}$ | 83 | 296 | 436 |
| | $2^{20}$ | 79 | 296 | 437 |
| AES (Rijndael) | $2^{10}$ | 49 | 91 | 136 |
| | $2^{12}$ | 49 | 95 | 141 |
| | $2^{15}$ | 45 | 97 | 149 |
| | $2^{17}$ | 45 | 98 | 149 |
| | $2^{20}$ | 47 | 98 | 149 |

The average encryption latency and throughput values remain relatively stable when encrypting different file sizes, except for the Blowfish cipher which seems to work more efficiently with larger file sizes. This could be due to the way Blowfish uses one function to process the entire buffer of plaintext and to encrypt the blocks with only a few operations in each round. The exclusive-or (XOR) operation is used frequently within Blowfish, compared to the other ciphers that mostly use addition. For file sizes larger than 4 KB, the Blowfish cipher is the fastest, followed by RC2, XTEA, and AES, in that order. This ordering appears to be the same across all three Pocket PC devices. However, there are some discrepancies with smaller file sizes. When encrypting the two smallest file sizes (1 KB and 4 KB), the ordering of ciphers from fastest to slowest is RC2, Blowfish, XTEA, and AES for the iPAQ 3850. However, XTEA encrypts faster than Blowfish on the other two Pocket PCs and the ordering from fastest to slowest is RC2, XTEA, Blowfish, and AES.

The RC2 cipher is fastest for small files, most likely because it loads only one P-box for key expansion into memory, versus one P-box and four S-boxes in Blowfish. The delay of loading the P- and S-boxes into memory becomes more significant when the size of the plaintext approaches the size of the boxes. RC2 also uses fewer rounds than XTEA.

The AES cipher was always the slowest of the four block ciphers in our experiments. The series of linked mathematical operations with P-boxes and S-boxes gives AES its strong cryptographic properties, but at the cost of more memory and processing requirements compared to the other three block ciphers.

The XScale processor of the HP 4150 and Dell Axim X30 appears to favor the XTEA cipher. RC2 is about twice as fast as XTEA for the iPAQ 4150 and Axim X30, while RC2 is about 4.5 times faster than XTEA for the iPAQ 3850. XTEA is about as fast to 1.75 times faster compared to AES on the iPAQ 3850, while XTEA is about as fast to three times faster than AES on the HP iPAQ 4150 and Dell Axim X30.

#### *5.3.3.2 Energy*

All energy-related metrics are based on experiments with the HP iPAQ 4150. The graphs in Figures 5.4 through 5.7 each show power consumption for a single execution of each block cipher encrypting a $2^{15}$-byte file. The voltage across the resistor in the experimental setup described in Section 5.3.2.2 was measured for 0.5 seconds in each case. The surges of power at the beginning and end of each encryption task are likely due to the allocation and deallocation of memory. Activity before the encryption task in Figures 5.5 and 5.6 is a result of loading the plaintext file into memory.

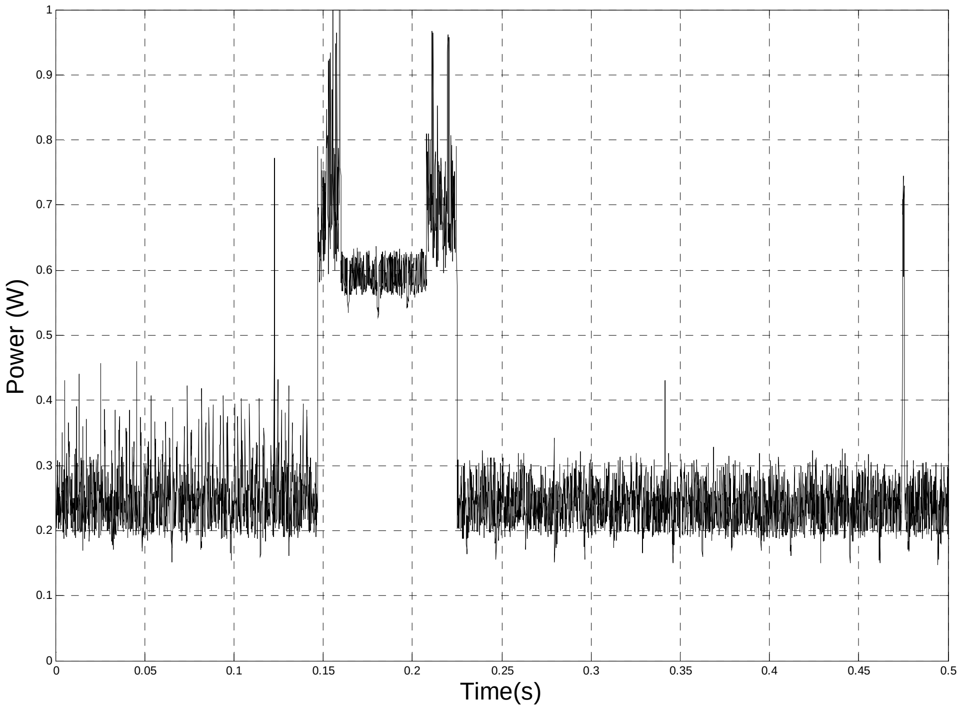


Figure 5.4. Power for RC2 encrypting a $2^{15}$-byte file.

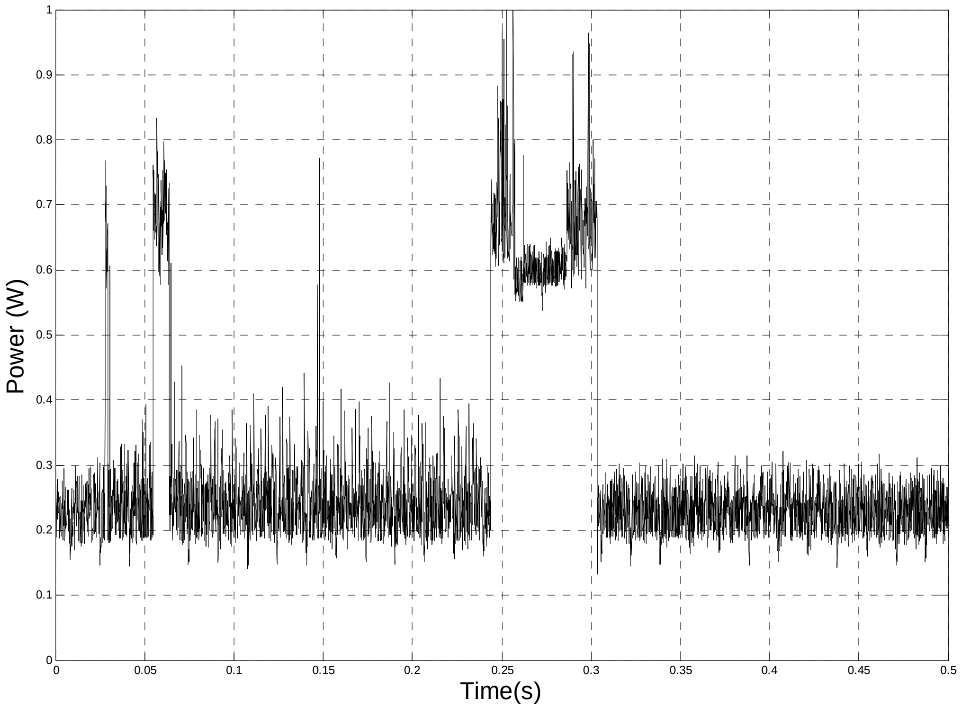


Figure 5.5. Power for Blowfish encrypting a $2^{15}$-byte file.

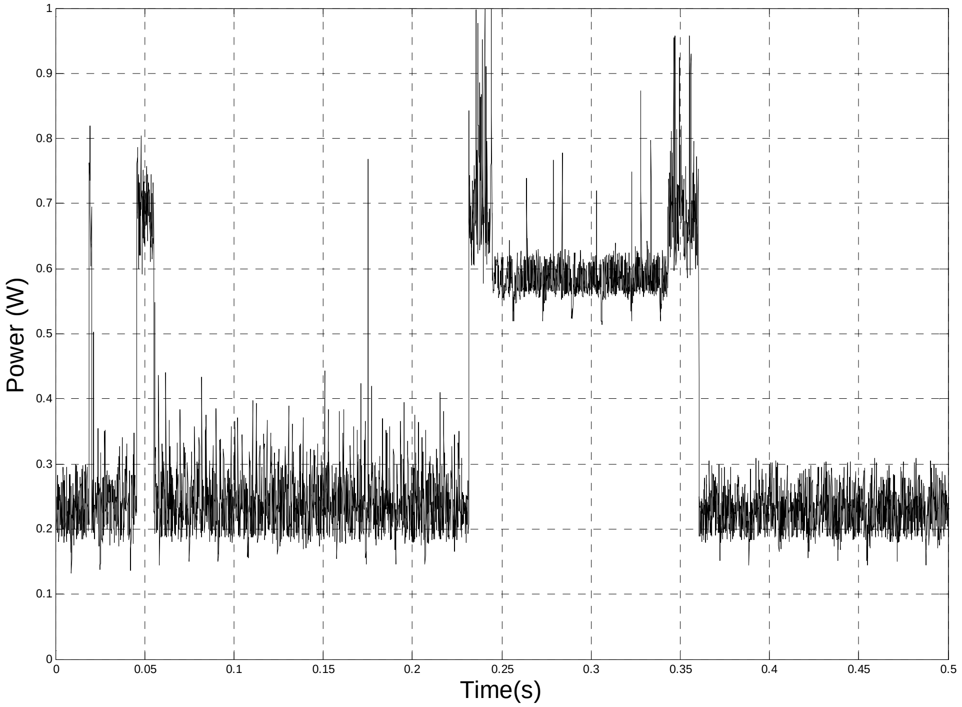


Figure 5.6. Power for XTEA encrypting a $2^{15}$-byte file.

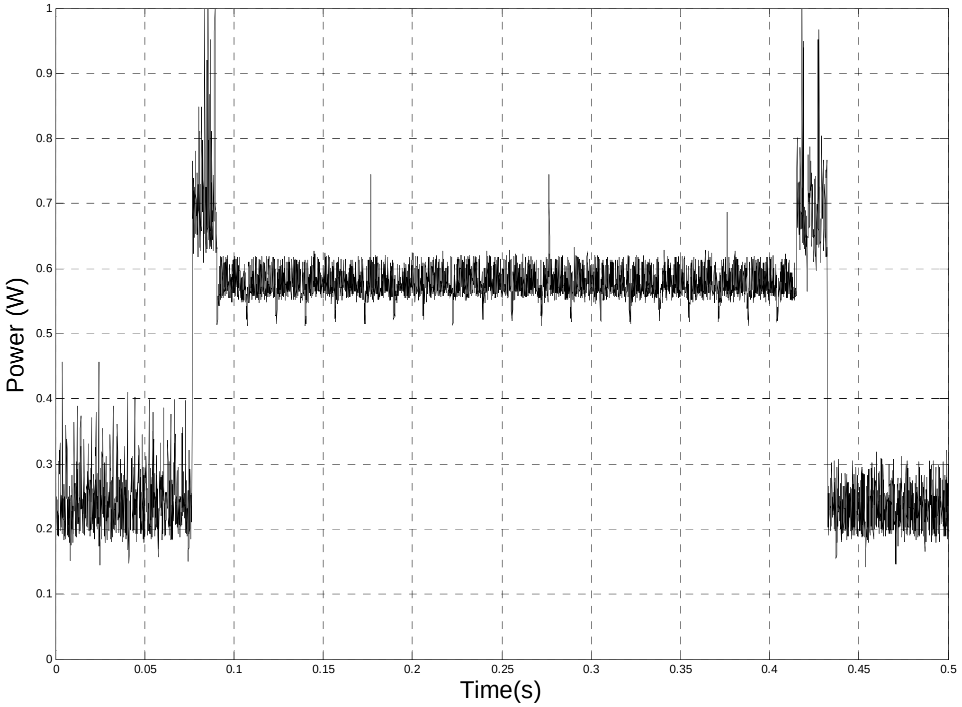


Figure 5.7. Power for AES encrypting a $2^{15}$-byte file.

The graphs show the relationship between the block ciphers in terms of the durations required to complete the encryption task. Blowfish exhibits slightly higher speed (shorter duration) compared to RC2, followed by XTEA, and then AES, by a wide margin. The encryption latency in the graphs is consistent with the performance of the ciphers shown in Tables 5.4 and 5.5. The power levels during the encryption task itself are relatively stable and are about the same for all four ciphers, implying that the time spent performing encryption affects energy efficiency more than differences in power drain while performing encryption.

Average energy consumption for the encryption tasks was calculated using the procedure described in Section 5.3.2.2. The energy-latency product and throughput/energy ratio were determined from the energy consumption values for different ciphers and file sizes. Results are presented in Table 5.6. The performance and energy metrics are highly correlated with one another since, as stated above, energy consumption depends mostly on the time required for the encryption task.

Note that the energy consumption values do not include energy consumption associated with an idle device. The energy consumption values in Table 5.6 consider only the additional energy required for encryption. In other words, the total energy consumed from the battery during an encryption task will be the energy consumed for encryption plus the energy consumed during idle mode.

In the iPAQ 4150, AES has the highest average values for the energy-latency product of the four block ciphers, which means that it consumes the most energy per byte when encrypting plaintext. These high values are due to the large number of sequential operations required for encrypting each block of data in AES. The high energy-latency product values of AES are consistent with its high encryption latency results shown in Table 5.5. XTEA consumes less energy per byte followed by RC2. The Blowfish cipher has the lowest average energy-latency product when encrypting large files, but this value increases as the plaintext file size decreases. A similar trend can be found in the average encryption latency values for Blowfish. The throughput/energy ratios of all the ciphers are also correlated to the encryption throughput values in Table 5.5 and yield similar trends and relative comparisons.

**Table 5.6: Energy Consumption Results for the iPAQ 4150**

| Cipher | File Size (bytes) | Energy Consumption (J) | Energy-Latency Product (uJ/B) | Throughput/Energy Ratio (MB/J) |
|---|---|---|---|---|
| RC2 | $2^{10}$ | 0.0006 | 0.633 | 1.507 |
| | $2^{12}$ | 0.0020 | 0.497 | 1.919 |
| | $2^{15}$ | 0.0131 | 0.401 | 2.381 |
| | $2^{17}$ | 0.0508 | 0.387 | 2.463 |
| | $2^{20}$ | 0.4033 | 0.385 | 2.480 |
| Blowfish | $2^{10}$ | 0.0019 | 1.858 | 0.513 |
| | $2^{12}$ | 0.0026 | 0.643 | 1.484 |
| | $2^{15}$ | 0.0091 | 0.276 | 3.451 |
| | $2^{17}$ | 0.0320 | 0.244 | 3.909 |
| | $2^{20}$ | 0.2483 | 0.237 | 4.027 |
| XTEA | $2^{10}$ | 0.0010 | 1.003 | 0.951 |
| | $2^{12}$ | 0.0035 | 0.857 | 1.113 |
| | $2^{15}$ | 0.0299 | 0.912 | 1.046 |
| | $2^{17}$ | 0.1163 | 0.887 | 1.075 |
| | $2^{20}$ | 0.9458 | 0.902 | 1.057 |
| AES (Rijndael) | $2^{10}$ | 0.0029 | 2.829 | 0.337 |
| | $2^{12}$ | 0.0106 | 2.599 | 0.367 |
| | $2^{15}$ | 0.0827 | 2.524 | 0.378 |
| | $2^{17}$ | 0.3285 | 2.506 | 0.381 |
| | $2^{20}$ | 2.7205 | 2.594 | 0.368 |

### 5.3.4 Summary

This section describes experiments and analysis intended to increase awareness of and insight into the processing and energy costs associated with utilizing certain block ciphers on PDAs and other resource-limited devices. In addition, results from this analysis are used in the design of the CASM decision module. Results for both performance, including latency and throughput, and energy consumption are presented for different ciphers applied to plaintext files of different sizes. For the handheld devices used in our experiments, we found that RC2 is faster than XTEA, which in turn is faster than AES for all files sizes. The relative performance of Blowfish depends on the size of the plaintext file, but, overall, Blowfish is a fast encryption algorithm. Our results also indicate that all ciphers consume a similar amount of power while executing, so faster algorithms consume less energy because they operate at an elevated level of power for less time. The methodology of these experiments can be applied to characterize other encryption algorithms, including stream ciphers, and other devices, for example with other operating systems and processors. Such extensions are left as future work.

In addition to the results of this research, users, software developers, and system designers need to consider a broad range of operating conditions when employing encryption. The experiments reported here consider the performance and energy consumption of encryption algorithms in isolation. Other operations, possibly using other resources, such as sending and receiving data over a wireless interface, playing audio over speakers, displaying video and graphics on a display, and running background applications may have more of an affect on battery life or system performance than encryption.

### 5.4 Top Level CASM Functionality

CASM was developed to secure our proposed pervasive environment. Its goal is to provide devices in the shared environment with improved performance and energy efficiency while maintaining desired levels of security. The manager selects appropriate security algorithms for an entity in the environment. The selection is based on the device's current context and user preferences. A change in context could cause a decision component of CASM to select a different security scheme.

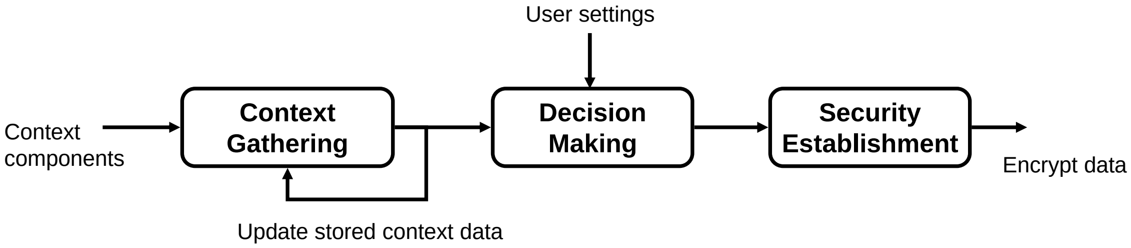


Figure 5.8. Top level CASM architecture.

The architecture of CASM is divided into three modules as shown in Figure 5.8. These modules include the context gathering, decision making, and security establishment processes. When an object requires security, CASM relies on these modules to fulfill the request. First, the context gathering module collects information from the surrounding environment in the form of the current context, further described in Section 5.5. Some context information is stored and is updated every time new context information is gathered. The output of the context module is used for the decision module, as described in Section 5.6. In addition to relying on user settings, the decision module contains algorithms, or *engines*, that are used to select a single security protocol. This selection result is then passed to the security module, which acts upon the decision and enacts the selected security scheme.

The current implementation of CASM is a prototype that is intended for experimentation. However, CASM's functionality could be incorporated into the OS of a wireless device. The OS could utilize loaded services to provide requests to CASM for file transfers or other internal operations. CASM could be used in combination with firewalls to provide privacy along with protection from unauthorized access.

CASM could also function as a middleware component used by different applications. For example, a file transfer protocol (FTP) server could use CASM to efficiently encrypt files before or as they are downloaded. CASM would also be useful for online transaction processing (OLTP) that requires wireless connections. In these applications, CASM can efficiently secure client requests by determining the context of the wireless network, and by deciding suitable security algorithms for each transaction.

## 5.5 Context Module

The context module is responsible for collecting context information from the environment. Context information comes in the form of six context components. These context components are then quantified for use in the decision module.

### 5.5.1 Functionality

A group of values or conditions that define the current context is used as input for the decision algorithm before it selects a cipher. Context components include desired security level, energy, location, network communication properties, object size, and user interactions or familiarity. The security level component of the current context is a user setting that influences the desired encryption strength for encrypting an object. The energy component is represented by the current battery level of the user device. SSIDs and the signal strength observed at the wireless network interface card affect the location and communications components. Finally, user familiarity is denoted by the number of times the user and a peer have interacted with each other.

The context values can be checked periodically for changes or assessed on demand. On some devices, especially those with small energy capacities or low processing capabilities, there exists a significant tradeoff between periodically collecting context data and only collecting them when they are needed. The advantage of periodic context collection is that the overhead can be reduced when actually performing an encryption task. In other words, if the relevant context values are not likely to change within an appropriate time interval, then decisions can be made from stored context data without having to acquire the current context data. The disadvantage to this method is that there are periodic surges of power consumption and processing overhead when acquiring the context values and these will result in overall reduced battery life and processing delays. Conversely, the advantage of collecting the context on demand potentially reduces total energy consumption, but requires additional overhead to perform every encryption task.

After a collection of context values, numerical weights are assigned to each context component, represented as *context metrics*, and supplied to the decision maker process. The context metrics are usually assigned a value between 1 and 9, inclusive, and quantitatively represent the current status of a single context value. For example, an energy context value with a context metric value of 9 could mean that the system's battery level is very low and a decision could be made to use the most energy efficient security protocol. However, should a context metric have a value of 0, then that means the associated context component should have no impact on the decision making process. This situation may arise for several reasons, which are explained in the context component descriptions in Section 5.5.2.

### 5.5.2 Context Components

Sections 5.5.2.1 through 5.5.2.6 indicate the components of context that are used in the CASM context module. Each context component is briefly summarized.

#### *5.5.2.1 Security Level*

The security level component is a user (or application) setting of the desired security level. The user has three options: Low, Medium, and High. These context values correspond to context metric values of 1, 5, and 9, respectively. Setting the security level to High causes the decision maker to be more likely to select a stronger security algorithm for encrypting data.

#### *5.5.2.2 Energy*

The energy component is actually a combination of two context components. The first is the status of the alternating current (AC) input. If the system is powered by an external power source, then, for the moment, there is no need to be concerned about remaining energy in the device and the context metric value is set to 0. If the AC line status is negative, then the device is solely functioning on its internal power supply and the context metric is now based on the second context component, remaining battery life.

The battery life information is gathered from the operating system's power management function as a percentage and the context metric is computed directly from this percentage value. Using the "remaining battery life" value, the *consumed battery life* value is calculated. The tens digit of this new percentage is used as the context metric. For example, a remaining battery life of 57% corresponds to a consumed battery life of 43% and a context metric of 4. If the consumed battery life is less than 10%, the context metric is set to 1. If the consumed battery life is 100%, which in theory should not be possible, then the context metric is set to 0. High context metric values bring about the selection of more energy efficient security algorithms. Low context metric values mean that the decision maker process is less concerned about energy efficiency and more concerned about security.

#### *5.5.2.3 Location*

Location is a difficult context component to quantify with respect to security. Usually when people refer to location, they mean position. Locations can be inferred from positions, such as "the corner of Stanger Street and Alumni Mall," "the office in 2040 Torgersen Hall," or "the longitude and latitude coordinates 37° 13.815' N, 80° 25.232' W supplied by a global positioning system (GPS) device." However, when given information such as "at home," "in the office," or "in the lab," position is not necessarily known or even needed. For example, a "home" location usually entails an environment where there exists a certain degree of trust between residents of the home. Wireless communications from a device located "at home" might not need the strongest security schemes to protect data transmitted (assuming, of course, that there aren't any less trusted neighbors in close vicinity).

In our design we use SSID to indicate location. A change in the SSID indicates a change in location. While there is a certain dependency on infrastructure, this is justified by the information provided by the location of the access points. Access points participating in our pervasive environment will have SSIDs that directly state or correspond to specified locations. The possible locations for this context component include Home, Office, Lab, and Unknown. The Home and Office locations have a context metric value of 3, the Lab location has a context metric value of 6, and the Unknown location has a context metric value of 9. The Home and Office environments were rationalized as relatively secure compared to the changing environment of the Lab. Any SSIDs not recognized are automatically classified as Unknown locations. When operating in ad hoc mode it is safer to classify all SSIDs as Unknown locations, because of the potential mobile nature of devices in ad hoc mode. A location with a high context metric value could mean the device is in an insecure environment and, thus, would result in selecting a strong encryption algorithm to secure shared data.

#### *5.5.2.4 Communications*

The communications context component can be represented by several types of information, including packet overhead, throughput, link capacity, quality of service (QoS), and signal strength. Our design uses signal strength to determine the communications context value with respect to security. Signal strength will be affected if the wireless device or access point is placed near metal surfaces and solid high-density materials. If there are obstacles in the radio signal path between an access point and wireless device or between two directly communicating wireless devices, the radio signal may either be absorbed or reflected. The coverage will, hence, be decreased. In addition, other devices that share the 2.4GHz radio spectrum with IEEE 802.11b, including microwave ovens, some cordless phones, and competing systems such as Bluetooth, may cause interference. Low signal strength could also be a result of an intruder attempting to jam the communications link. The data rate will also drop if the signal strength is weak.

Signal strength possibilities for this context component include Very Low, Low, Good, Very Good, Excellent, and No Signal. The corresponding context metric values are 9, 7, 5, 3, 1, and 0. For this context, we have placed a priority on how signal strength affects transmission rates. High context metric values suggest low signal strengths and, thus, low data rates. In our design, a device communicating with low data rates will use security protocols that are more efficient in terms of energy and performance. Communications with good signal strengths, which follows with potentially high data rates, can afford to use security algorithms that are stronger and not necessarily as efficient.

#### *5.5.2.5 Object Size*

An object requiring security, whether it is a document, audio clip, or URL, will have a size value representing the number of bytes needed to represent and store the object. The information collected from this context is simply the number of bytes necessary for object storage.  The study of security resources presented in Section 5.3 experimented with five different file sizes.  These five sizes are used as boundaries for ranges of object sizes in this context component.  Objects that have a size less than 4 KB have a context metric value of 1; sizes between 4 KB and 32 KB have a metric value of 3; sizes between 32 KB and 128 KB have a metric value of 5; sizes between 128 KB and 1 MB have a metric value of 7; and sizes greater than or equal to 1 MB have a metric value of 9. Large objects require a longer time to encrypt than small objects, accordingly more efficient security algorithms are needed for larger objects.

#### *5.5.2.6 User Interactions*

The user interactions component of context represents the number of different sessions in which a user and a peer have participated.  Strangers creating a session for the first time will begin with a value of 0.  Every subsequent session created between the same two users will increase the number of user interactions by 1.  In other words, the importance of this context component is inversely related to the number of interactions between two users.  Fewer than 3 interactions gives a context metric value of 9; 3 or 4 interactions give a metric value of 7; 5 or 6 interactions give a metric value of 5; 7 or 8 interactions give a metric value of 3; and 9 or more interactions give a context metric of 1.  The more times two users have interacted with each other in sessions, the more lenient the security level will be in favor of other objectives, such as rapid communications and reduced energy consumption.

#### *5.5.2.7 Summary*

This section described six different contexts that are used in the context module of CASM.  The context metric values and the matching context components are summarized in Table 5.7.

**Table 5.7: Context Metric Values and Context Components**

| Context Metric Values | Security Level | Energy | Location | Communications | Object Size | User Interactions |
|---|---|---|---|---|---|---|
| 0 | – | AC Line | – | No Signal | – | – |
| 1 | Low | > 80% | – | Excellent | < 4 KB | ≥ 9 |
| 2 | – | 70-79% | – | – | – | – |
| 3 | – | 60-69% | Home / Office | Very Good | 4 KB to 32 KB | 7 or 8 |
| 4 | – | 50-59% | – | – | – | – |
| 5 | Medium | 40-49% | – | Good | 32 KB to 128 KB | 5 or 6 |
| 6 | – | 30-39% | Lab | – | – | – |
| 7 | – | 20-29% | – | Low | 128 KB to 1 MB | 3 or 4 |
| 8 | – | 10-19% | – | – | – | – |
| 9 | High | < 10% | Unknown | Very Low | ≥ 1 MB | ≤ 2 |

## *5.6 Decision Module*

The decision module uses the numeric context data acquired from the context module to produce decisions. The decision module contains more than one decision maker. Each decision of the most appropriate encryption algorithm is output from this module. The resulting decisions are sent as input to the security module.

### 5.6.1 Functionality

The decision module selects a suitable block cipher for encrypting object data. The cipher selection criteria are based on the context metrics and the metrics from the cost analysis experiments of the block ciphers. The decision module in CASM contains three *decision engines*.

- One decision engine relies on the Analytic Hierarchy Process (AHP), as described in Section 2.5.4. In this engine, decision weights are based on general results in the open literature and are not "tuned" to the particular devices used in this study.

- Another engine uses the analytical results of the block cipher study presented in Section 5.3. This engine effectively makes decisions based on simple thresholds, with the thresholds determined from the experimental results. Note that these results are for the particular device types on which CASM is run.

- The third engine is a hybrid of the previous two engines. It uses AHP, but with decision values based on the same experimental results used in the second engine. Thus, it uses AHP for decision making, but is “tuned” to the particular devices used in this research.

Note that a “production” version of CASM would likely include only one decision engine. We have included three engines to allow experiments to compare and assess the three different decision engines. Note, also, that for our design we focused on AHP as part of the decision maker. We chose AHP for its flexibility and its capability to formalize the decision process from quantitative and qualitative aspects of the decision criteria. This does not imply that AHP is the only method to handle decision making. System designers, project developers, and even users may utilize other decision mechanisms as they see fit.

### 5.6.2 Decision Engines

The AHP decision engine is explained in more detail in Section 5.6.2.1. A Deterministic decision engine based on the security resource cost-analysis is described in Section 5.6.2.2. Finally, the Modified AHP decision engine is discussed in Section 5.6.2.3.

#### *5.6.2.1 AHP Decision Engine*

Before the context metrics are used, *context weights* must be applied. Context weights represent the importance of a single context component to the decision process. A context weight of 9 means the represented context component is extremely important and the decision will most likely be entirely based on this context. A context weight of 1 means the context component will have little significant effect, if any, on the final outcome.

Furthermore, there are four settings for the AHP engine. These settings include Balanced, Energy, Performance, and Security. The Balanced setting allows CASM to make decisions based on equal weights for context components. The Energy and Performance settings use weights in favor of the energy and communications contexts, respectively. The last setting emphasizes strength of security and targets the location and user familiarity contexts. At present, these are user-selected settings in CASM. In a “production” implementation of CASM the settings might also be selected by the end user to specify his or her preference or they might be configured as part of the system software for an application-specific device.

The block ciphers used to secure the objects are represented as matrices in terms of relative importance to a specific context component. In other words, each context component is associated with an options matrix. A context metric affects the block cipher weights used in the respective options matrix. The weights in the options matrices are derived from the literature (summarized in Section 2.3.3.3) and may not accurately represent actual software implementations of the block ciphers’ behaviors or their operation on any given device.

An objective matrix is then constructed from preferences for each context component derived from the user selection of Balanced, Energy, Performance, or Security setting. The objective matrix and options matrices are evaluated to create the decision vector. This vector contains overall importance values for each block cipher. The block cipher with the highest value is chosen to encrypt the data.

The following example illustrates the use of AHP in CASM. For simplicity, only three block ciphers – RC2, XTEA, and AES – and two context components – security level and energy – are enumerated in this example. In addition, both context components have equal context weights. (The full set of four ciphers and all context components are considered in the CASM implementation.)

AHP Example:

1. Let RC2, XTEA, and AES represent the options.

2. Let security level, *S*, and energy, *E*, represent the context components, with context metrics Medium and 76%, respectively.

3. Then, the context metric value of *S* is 5 and *E* is 2, and *S* is weakly more important than *E*. This was calculated by taking the difference between the context metric values. If the difference is positive, then that value is chosen, otherwise, the negative reciprocal is used. Thus, comparing *S* to *E* results in a pairwise comparison value of 3, but comparing *E* to *S* results in a pairwise comparison value of $3^{-1}$ or 1/3.

4. The objective matrix, *O*, can now be constructed. After the normalization and averaging procedure the objective vector, *o*, is computed.

$$O = \begin{array}{c} S \\ E \end{array}\begin{bmatrix} 1 & 3 \\ 3^{-1} & 1 \end{bmatrix} \Rightarrow o = \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix}$$

5. The options matrix with respect to *S* is denoted as $P_S$ and the options matrix with respect to *E* is denoted as $P_E$.

$$P_S = \begin{array}{c|ccc} & \text{RC2} & \text{XTEA} & \text{AES} \\ \hline \text{RC2} & 1 & 5^{-1} & 1 \\ \text{XTEA} & 5 & 1 & 5 \\ \text{AES} & 1 & 5^{-1} & 1 \end{array} ; P_E = \begin{array}{c|ccc} & \text{RC2} & \text{XTEA} & \text{AES} \\ \hline \text{RC2} & 1 & 5^{-1} & 9^{-1} \\ \text{XTEA} & 5 & 1 & 5^{-1} \\ \text{AES} & 9 & 5 & 1 \end{array}$$

6. The option vectors are created as *p*(*S*) and *p*(*E*) with the overall options matrix *P* as follows.

$$p(S)=\begin{bmatrix}0.105\\0.524\\0.105\end{bmatrix};p(E)=\begin{bmatrix}0.029\\0.138\\0.333\end{bmatrix}\Rightarrow P=\begin{bmatrix}0.105 & 0.029\\0.524 & 0.138\\0.105 & 0.333\end{bmatrix}$$

7. The decision vector, *d*, is the product of the overall options matrix and the objective matrix.

$$d=P\cdot o=\begin{bmatrix}0.105 & 0.029\\0.524 & 0.138\\0.105 & 0.333\end{bmatrix}\cdot\begin{bmatrix}0.75\\0.25\end{bmatrix}=\begin{array}{c}\begin{bmatrix}0.086\\0.428\\0.162\end{bmatrix}\end{array}\begin{array}{l}\text{RC2}\\\text{XTEA}\\\text{AES}\end{array}$$

8. The largest element in the decision vector corresponds to the XTEA block cipher. Thus, for a Medium security level and 76% remaining battery life, AHP deems XTEA as the most suitable cipher for encryption.

Refer to Appendix A for a complete description of the AHP decision engine, including the context component weights and options matrices.

#### *5.6.2.2 Deterministic Decision Engine*

The Deterministic decision engine is composed of a set of cases that utilize the results from the block cipher analysis. Only two context components are used to help make the decision, the security level and the object size. There are two settings for Deterministic decision engine, Flexible and Rigid. These are selected by the user in the current implementation of CASM. The thresholds for the two settings were created from the resulting metrics in Tables 5.4 and 5.5. For each file size, the cipher with the best performance is chosen.

If the Rigid setting is selected, the block cipher chosen is mostly dependent on the level of security and not the object size. A High security level would mean AES is chosen. A Low security level leads to selection of RC2. If the user selects a Medium security level, then either XTEA or Blowfish is selected, depending on the object size. XTEA has lower encryption latency and higher encryption throughput than Blowfish for objects smaller than 4 KB. Thus, small objects, like URLs or contact information, will be encrypted with XTEA under the Medium security level in this decision engine. Larger objects will use Blowfish for encryption, because Blowfish improves its encryption efficiency for larger objects.

If the Flexible setting is selected, block ciphers are selected mostly based on their efficiency with regard to object size. For security levels of Low and Medium, RC2 is chosen if the object size is less than 32 KB, and Blowfish is selected for objects larger than 32 KB. With a High security level and object size less than 4 KB, XTEA is selected. Objects larger than 4 KB at a High security level will use Blowfish for encryption.

Using the Deterministic engine versus the AHP engine has several advantages. The cipher selections are adjusted to maximize efficiency for any situation within the constraints of the environment. This means faster and more energy efficient block ciphers are chosen over slower ones. Data would be encrypted as quickly as possible. The disadvantages are that security is not regarded as highly as encryption performance and that user settings are limited. In other words, the Deterministic engine chooses efficiency over strength of security. The AHP engine has more flexibility in terms of user settings, but it is not optimized to work with the particular environment which is reflected in the parameters of the deterministic model.

#### *5.6.2.3 Modified AHP Decision Engine*

The Modified AHP decision engine functions similarly to the AHP decision engine described above in Section 5.6.2.1. There are the same four settings of Balanced, Energy, Performance, and Security that can be selected by the user in the current implementation of CASM. The main difference between the Modified AHP engine and the original AHP engine is the assignment of the block cipher weights in the options matrices. For this engine, the options matrices favor block ciphers that encrypt efficiently. The values in the options matrices are assigned on the basis of the block cipher experiments described in Section 5.3. The Blowfish cipher is especially preferred, due to its reasonable security strength and encryption speed.

The Modified AHP engine attempts to garner the advantages of both the original AHP and the Deterministic decision engines. Decision parameters are altered to incorporate results from the block cipher experiments. The efficiency of the Deterministic engine and the flexibility of the AHP engine are combined to select ciphers that are efficient and secure. The Modified AHP engine would work well in our environment, but might not be as flexible as the original AHP engine in other environments or implementations. The Modified AHP engine also needs to gather more context components than the Deterministic engine to make its decisions. More context components could potentially mean more overhead, especially for encrypting small files.

Refer to Appendix A for the options matrices used in the Modified AHP decision engine.

## *5.7 Security Module*

Encryption algorithms are contained in the security module. The security module acts on the decisions that are output from the decision module. The decisions are the chosen algorithms that are used to encrypt data. Each chosen algorithm is initialized and performs its encryption task accordingly.

### 5.7.1 Functionality

The purpose of the security module in CASM is to activate the block ciphers selected by the decision module and use them to encrypt the objects. Four blocks ciphers are incorporated into the security module. These block ciphers are RC2, Blowfish, XTEA, and AES.

An object that needs to be encrypted in the pervasive collaboration environment will use CASM to fulfill its request. After context acquisition and deciding on the appropriate block cipher, the security module uses the block cipher to encrypt the object contents, which are stored in memory. The encrypted object is then ready to be transmitted to another user or “shared” in the environment.

### 5.7.2 Cryptographic Algorithms

As stated previously, four block ciphers are used to encrypt objects in CASM. The operating parameters of the block ciphers in the security module are equivalent to the parameters used in the security resources cost-analysis described in Section 5.3. The operating parameters are listed above in Table 5.2. The block ciphers function in ECB mode.

## *5.8 Summary*

This chapter describes the design of CASM. The intended environment for its operation is briefly discussed. Four different block ciphers were analyzed for their resource costs. The internal architecture of CASM is also explained. The architecture consists of three modules: context, decision, and security. The context module gathers information from the environment. The decision module selects a particular block cipher based on the information retrieved by the context module. Finally, the security module initiates the parameters required for the block cipher to encrypt an object.

# Chapter 6. Implementation and Evaluation of the Context Aware and Adaptive Security Manager

This chapter describes the implementation of CASM. Experiments involving the CASM implementation are also discussed in this chapter. The experiments and analysis demonstrate the feasibility, benefits, and limitations of CASM and its associated concepts.

The chapter is divided into four sections. First, assumptions and the experimental setup are explained in Section 6.1. Next, Section 6.2 describes the different experiments. The results and analysis of the CASM experiments are related in Section 6.3. Then, Section 6.4 discusses the results. Finally, Section 6.5 summarizes the chapter.

## *6.1 Assumptions and Experimental Setup*

The experiments in this work were evaluated through implementation and collection of measurements. Analytical evaluation was ruled out due to the precision required in analytical evaluation. Even though simulations are less costly and time consuming, they typically offer less accuracy than actual measurements. Additionally, the cost of implementing the experiments was relatively low because the required hardware and parts of the software base are readily available. Therefore, actual measurements with controlled workloads were used for this research.

This section presents the assumptions and setup used in the experiments used to evaluate CASM. Instead of separating the assumptions and experimental design, they are explained jointly for clarity. Sections 6.1.1 through 6.1.5 depict the components of the experimental setup, including the algorithms used, the hardware and software required, the environment, and the implementation of CASM.

### 6.1.1 Algorithms

Block ciphers were chosen for the CASM security module because of their frequent use, encryption speed, and ease of implementation. After some adjustments, other ciphers, algorithms, and security protocols could also be incorporated. However, for this study, only the four block ciphers used in the security resources cost-analysis in Chapter 5 are implemented. These ciphers are RC2, Blowfish, XTEA, and AES.

The operating parameters for the block ciphers employed in the CASM security module are presented in Table 5.2. Most of the parameters, such as key size and number of rounds, were maximized when using the ciphers for the experiments. This is done in accordance with the parameters used in the block cipher study described in Section 5.3.1. Since the results of the block cipher study affect the operation of the CASM decision module, the parameters for the CASM implementation need to be identical to those used in the block cipher study.

The block ciphers operated in electronic code book (ECB) mode. ECB mode was mainly chosen due to its straightforward implementation. The ECB mode was also picked to match the settings used in the block cipher analysis done earlier.

Cryptanalysis of CASM and the ciphers that it selects is not part of this research. However, users and system designers must consider vulnerabilities and other cryptographic weaknesses before implementing security mechanisms.

### 6.1.2 Hardware

We used the iPAQ 4150 for the CASM experiments. Note that power measurements are difficult for devices that do not have a removable battery or other easy way to access the battery terminals. Relevant specifications for this Pocket PC are listed in Table 5.1.

The iPAQ 4150 also interfaced with a Dell™ Inspiron™ 4000 notebook using a Dell TrueMobile™ 1150 Series wireless LAN card. The pertinent system specifications of the notebook are the CPU, memory, and OS. The Inspiron 4000 uses a 900 MHz Intel Pentium® III processor. It also contains 256 MB of random access memory (RAM) and uses Microsoft Windows XP Professional Service Pack 1 for its operating system.

Power measurements were conducted using the iPAQ 4150, a 0.025 Ω precision resistor in series between the battery and the device itself, and an Agilent 3458A 8½ Digit Multimeter, as shown in Figure 5.3. Note that this is the same power measurement setup used in the security resources cost-analysis described in Section 5.3.2.2.

### 6.1.3 Software

We used the Visual Studio .NET and Compact Framework as the software development environment. Code was written in C# to create the CASM application for the experiments. Existing block cipher implementations in C, C++, and C# were ported for use in the CASM security module [89-92]. The block ciphers were applied to their corresponding standard test vectors to verify proper implementation. More details of the CASM application components are given in Section 6.1.5.

### 6.1.4 Environment

Communications were achieved via 802.11b wireless network cards and an access point. The iPAQ 4150 has an internal 802.11b wireless card and the Dell Inspiron 4000 used the Dell TrueMobile wireless card. The notebook acted as a server, while the Pocket PC functioned as the client. The distance between the client and server was approximately one meter. The distance between the nodes and the nearest access point was within 30 meters. The nominal data rates were 11 Mbps. Encrypted files are transmitted from the client to the server. The setup is shown in Figure 6.1.

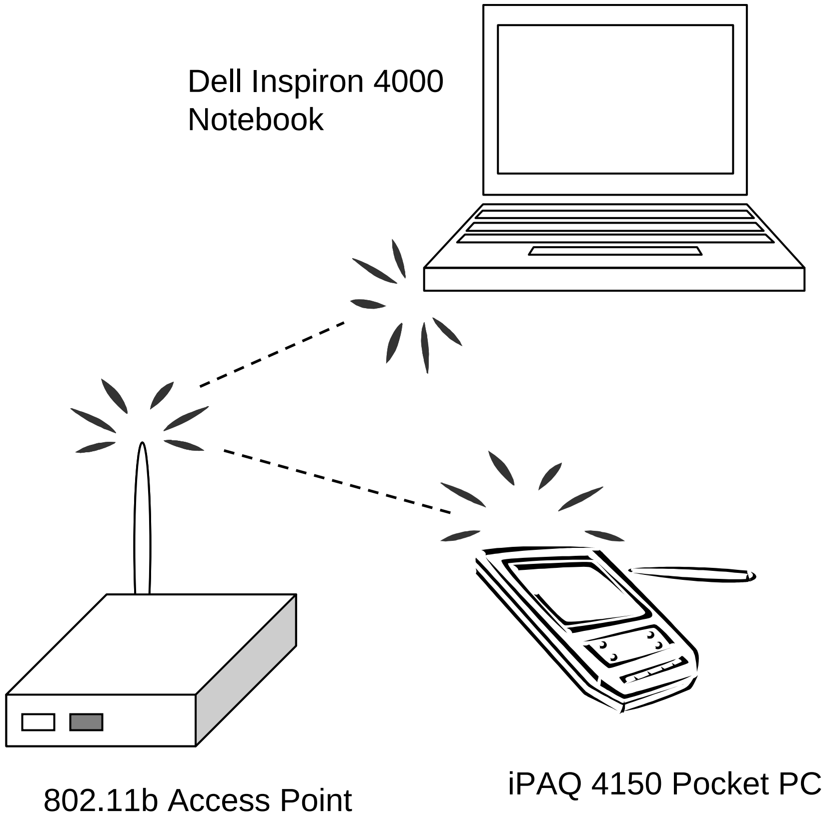


Figure 6.1. Configuration for the CASM experiments.

### 6.1.5 Pocket PC Client Application

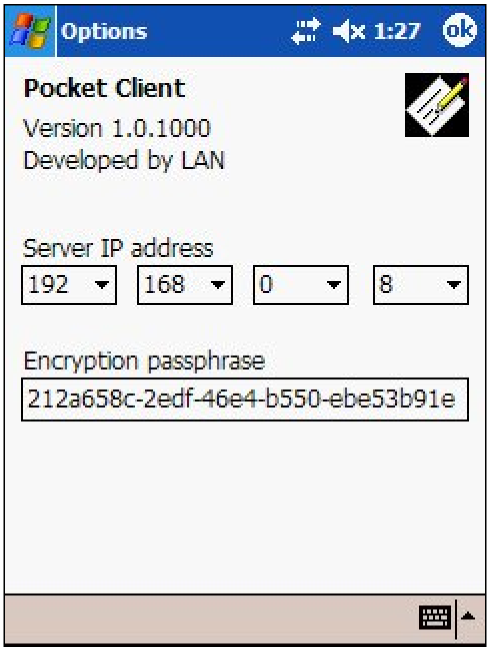
The client application gives the user the ability to send encrypted files to a server application using an IEEE 802.11b wireless connection. Requests are processed through CASM, which is contained within the client application. CASM accepts each request and decides on suitable encryption algorithms for different situations, then encrypts the file.

#### *6.1.5.1 User Interface*

The user interface of the Pocket PC client allows a user to control the CASM operation to a certain degree. The current user interface is designed to support experimental investigation of the CASM concept. A menu bar, a status bar, a few buttons, a few tab groups[21], and a dialog box constitute the entire user interface. The menu bar contains one item, the File menu. Within the File menu are the items Reset, Options, and Exit. If the Reset item is selected, then the application resets all its internal components, closes any active network connections, and flushes its memory. Selecting the Exit item causes the application to respond similar to a Reset command, except the application also closes and stops its associated system processes. If the Options item is clicked, the dialog box is

[21] A tab is a small rectangular box usually containing a text label or graphical icon associated with a view pane. A tab group is a collection of more than one tab. For information regarding the Adobe Systems and Macromedia patent dispute over tabs see http://www.iboost.com/build/design/articles/adomac/1148.htm.

displayed to the user. The dialog box contains the Internet Protocol (IP) address of the server and can be altered by the user if needed. The dialog box also holds the passphrase that is used to generate the encryption keys of the block ciphers. The passphrase itself is a randomly created Pocket PC globally unique identifier (GUID) [94]. There are 122 random bits (128 total bits – 2 bits for variant – 4 bits for version) in a GUID, so this results in $2^{122}$ possible combinations. The Options dialog box is illustrated in Figure 6.2.



Pocket Client
Version 1.0.1000
Developed by LAN

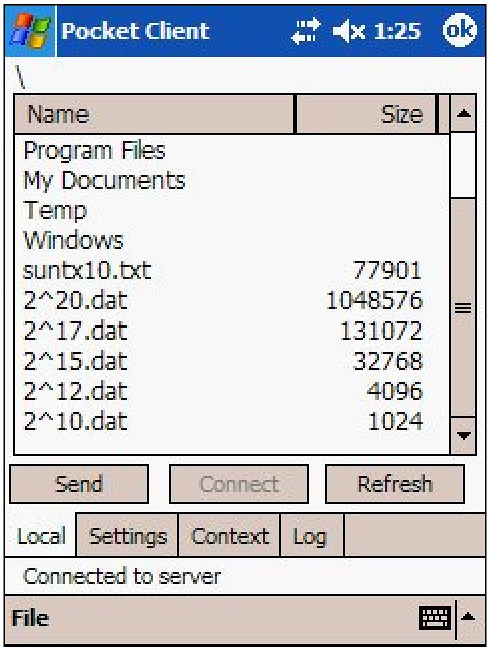
Server IP address
192 168 0 8

Encryption passphrase
212a658c-2edf-46e4-b550-ebe53b91e

Figure 6.2. Options dialog box of Pocket PC client user interface.

The status bar displays useful information regarding the current state of the client application. Text messages such as “Connected to server” or “Encrypting content…” are updated on the status bar.

The tab group contains four tabs labeled as Local, Settings, Contexts, and Log. The Local tab resembles a file explorer interface that lists files stored on the device and their associated size in bytes. The path is shown at the top of the tab. In addition, there are three buttons, Send, Connect, and Refresh. The Send button initiates a request to encrypt a selected file and send it to the server. This action causes CASM to perform its services. Pressing the Connect button causes the application to attempt to create a Transmission Control Protocol (TCP) connection to the IP address contained in the dialog box. Using the Refresh button simply updates the file explorer display. The Local tab is displayed in Figure 6.3.



\

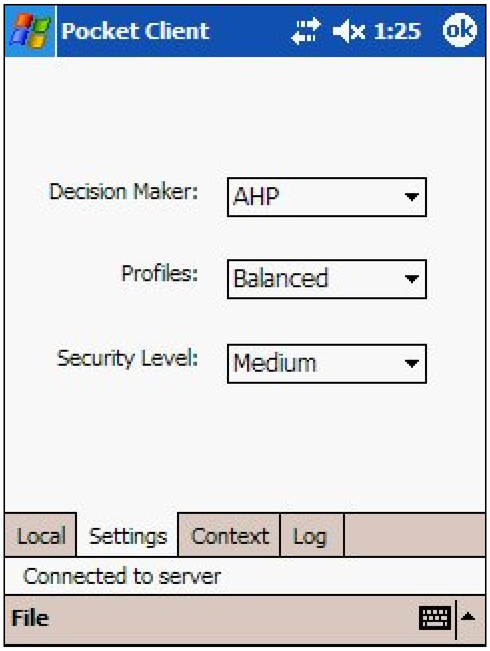
| Name | Size |
|---|---|
| Program Files | |
| My Documents | |
| Temp | |
| Windows | |
| suntx10.txt | 77901 |
| 2^20.dat | 1048576 |
| 2^17.dat | 131072 |
| 2^15.dat | 32768 |
| 2^12.dat | 4096 |
| 2^10.dat | 1024 |

Connected to server

Figure 6.3. Local tab of Pocket PC client user interface.

The Settings tab, shown in Figure 6.4, contains user profiles that allow the user to affect how CASM operates. The three “combo” boxes shown are Decision Maker, Profiles, and Security Level. The Decision Maker combo box allows the user to choose between the three decision engines that can be used by CASM. The Profiles combo box contains the different settings for each decision engine (e.g., Balanced, Energy, Rigid, etc.). Changing the Security Level combo box directly affects the context metric for the security level context. In other words, the user has direct control over this particular context and can determine whether the security level of CASM should be Low, Medium, or High.

The Context tab is actually another tab group that contains four other contexts. These contexts are energy, location, communications (denoted by the Network tab), and user interaction (denoted by the Familiarity tab). For example, the Energy tab in Figure 6.5 is active within the Context tab group. Each of these context tabs has an Update button so that the user may manually update the context metrics to view their current statuses. The user can also manually control the context metrics by selecting the Manual check box in the upper right corner of the context tabs. This can be done for one or more of the four contexts. Note that the final context, size, is determined by the size of the file selected in the Local tab.



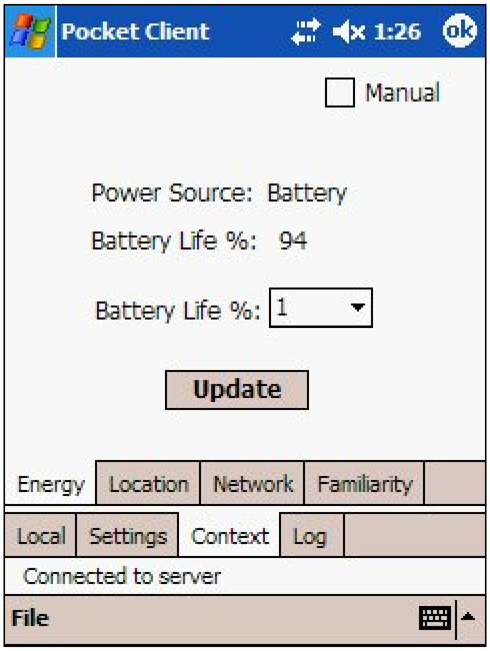
Decision Maker: AHP

Profiles: Balanced

Security Level: Medium

Connected to server

Figure 6.4. Settings tab of Pocket PC client user interface.



□ Manual

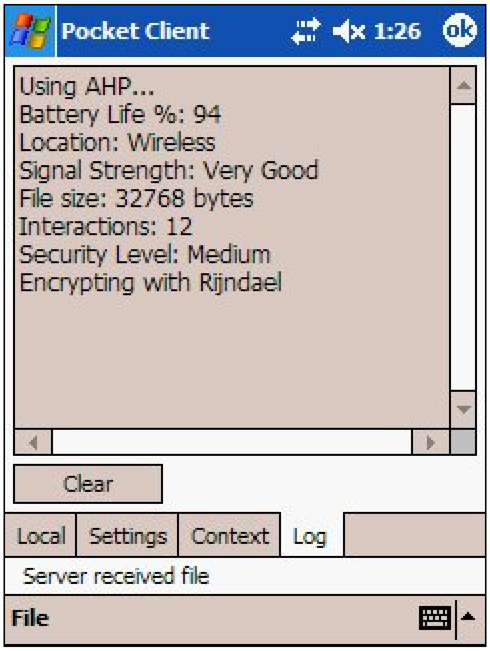
Power Source: Battery

Battery Life %: 94

Battery Life %: 1

Connected to server

Figure 6.5. Context tab of Pocket PC client user interface.



```
Using AHP...
Battery Life %: 94
Location: Wireless
Signal Strength: Very Good
File size: 32768 bytes
Interactions: 12
Security Level: Medium
Encrypting with Rijndael
```

Server received file

Figure 6.6. Log tab of Pocket PC client user interface.

Finally, the Log tab contains a text box that keeps the user informed of events occurring within the application. These events include file transmissions, CASM output, and potential errors. A Clear button in the Log tab can be used to erase text messages displayed in the log window. The Log tab is shown in Figure 6.6.

### *6.1.5.2 Device Communications*

Upon starting the application, the client attempts to connect to the server. The client application uses the IP address stored in a configuration file encoded using the eXtended Markup Language (XML). If an IP address does not exist in the configuration file, then the client attempts to find the IP address of the server by reading information from the registry. If the address cannot be found, then the client prompts the user to enter the server IP address via the Options dialog box. The port number for communications is also stored in the XML configuration file. The port number used for the experiments is 10200.

Data is exchanged between the client and server over TCP sockets by using the *Net.Sockets.Socket* class. The client sends the encrypted file to the server and the server sends an acknowledgment back to the client that indicates it received the file. The client uses asynchronous sockets when communicating with the server. The user interface is not blocked since socket operations occur in separate system-supplied threads. An *AsyncCallback* delegate is passed to all of the asynchronous socket methods and is invoked when the socket operation completes.

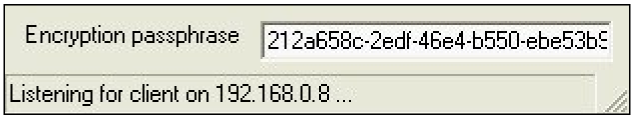
A one-byte *Crypto Header* is added to each encrypted file before transmission. The Crypto Header indicates which security algorithm was used to encrypt the file. Encoded values are as follows: RC2 = 1, Blowfish = 2, XTEA = 3, AES = 4. In addition, the file is padded with a unique terminator that is not encrypted. Adding these extra bytes assists the server in decrypting the file.

### 6.1.6 Server Application

The server executes on a standard Microsoft Windows platform, as described in Section 6.1.2. It receives the encrypted file over a TCP socket and decrypts the data using part of the CASM security module. The server application does not check for context or make decisions regarding encryption algorithms. Furthermore, only the security module of CASM is included in the notebook server. Once the server decrypts a received file, it stores the file in its local path.

#### *6.1.6.1 User Interface*

The server user interface is much simpler than the client user interface and has only two items. The first item is a status bar that operates similarly to the status bar in the client. Text messages such as "Connected to client" are updated on the status bar. The second item in the server user interface is a text box containing a passphrase. Like the client, the passphrase in the server is used to generate the encryption keys for decrypting the received files. The user interface of the notebook server application is shown in Figure 6.7.



Encryption passphrase 212a658c-2edf-46e4-b550-ebe53b9

Listening for client on 192.168.0.8 ...

Figure 6.7. Server user interface.

#### *6.1.6.2 Device Communications*

When the server application starts, it begins listening for client connection requests for its IP address and port. The port number is determined by user specification in an XML configuration file for the server. The server listens indefinitely until it is terminated or a client requests a connection and successfully connects to the server. The server uses synchronous sockets so it is responsible for creating its own worker thread to process socket operations. Thus, socket operations execute in a different thread than the user interface. The server application uses an event to notify the user interface when socket operations complete.

The server receives encrypted files from the client. As the server is receiving the file, it checks for the terminator that was added at the end of the encrypted file by the client. Once the terminator is received the server discards it from the received file and sends an acknowledgement back to the client. The first byte received is also parsed because it is the Crypto Header. The server uses the Crypto Header to identify the proper algorithm for decrypting the file. Once the file is decrypted, a user can access its contents on the server.

### 6.1.7 CASM Implementation

The implementation of CASM is a goal for this research because it demonstrates the feasibility of the context aware adaptive security approach and it allows experimental evaluation of CASM. CASM itself runs within the Pocket PC client application. Sections 6.1.7.1 through 6.1.7.3 explain the software design of the CASM modules.

#### *6.1.7.1 Acquiring Context Information*

There are six context components in the CASM context module. All the context components are assessed on demand per file encryption request. Two context components are already accounted for in the Pocket PC client application. The security level context component is directly manipulated by the user and, when a file is selected, the associated size is used for the object size component of the context. This leaves the energy, location, communications, and user interactions context components.

The energy component is derived from the percentage of battery life remaining and the AC line status. To determine the energy context metric, the system power status functions [95] were invoked from the “CoreDll.dll” library in the OS. These functions measure the different properties of the internal battery, including remaining battery life percentage, present voltage, chemistry, and AC line status.

The location and communications context components require an external library to gather the context values. A library from the OpenNETCF.org Smart Device Framework [96] was used. The “OpenNETCF.Net.dll” library was slightly modified to suit the Pocket PC device. This library enables application software to access information from network adapters in a PDA. The PDA must support the .NET CF and at least the Microsoft Pocket PC 2003 OS. Using functions from the library, the current SSID and signal strength related to the Pocket PC wireless network adapter can be determined.

The user interactions context component is determined by the number of times the server and client have successfully established a connection. This number is stored externally in the XML configuration files of the client and server. Each time a client and server have successfully connected, the user interaction number is incremented by one. The context metric is acquired directly from the configuration file.

#### *6.1.7.2 Decision Making*

The decision module contains three decision engines, each with their own user settings. CASM operates using only one decision engine with one setting for a single file transmission. In other words, only one decision maker is active at a time. The client application allows the user to select a specific decision engine and user settings for that engine. When an encryption request is made, CASM checks the status of the Settings tab in the client application to determine which decision maker to use. The context metric values are input into the active decision engine.

Regarding context weights, the Balanced setting used a value of 1 for all its weights. The Energy setting used context weights of 9 for the energy and object size context components and 1 for the rest of the context components. The Performance setting used context weights of 9 for the communications and object size context components and 1 for the rest of the context components. Finally, the Security setting used context weights of 9 for the location and user interactions context components, and 1 for the rest of the context components. The context weights of the different settings in the AHP and Modified AHP decision engines are summarized in Table 6.1.

**Table 6.1: Context Weights per User Setting**

| Contexts | Balanced | Energy | Performance | Security |
|---|---|---|---|---|
| Security Level | 1 | 1 | 1 | 1 |
| Energy | 1 | 9 | 1 | 1 |
| Location | 1 | 1 | 1 | 9 |
| Communications | 1 | 1 | 9 | 1 |
| Object Size | 1 | 9 | 9 | 1 |
| User Interactions | 1 | 1 | 1 | 9 |

The bulk of the decision module software implementation involves matrix operations or a deterministic selection to find suitable block ciphers for encrypting a file. The AHP engines use many two dimensional arrays to store the options matrices, as documented in Appendix A.

#### *6.1.7.3 Cryptography*

Four block ciphers are implemented in the security module of CASM. When the appropriate block cipher is selected for encryption, an instance of the linked block cipher class is created. The object instance is then initialized according to the operating parameters in Table 5.2. The encryption object performs its encryption task on the file contents. The client application is notified of which block cipher was chosen and then properly sets the value of the Crypto Header.

Since the security module contains methods for decryption, it is also used in the server application for this purpose. In other words, the server uses the CASM security module to decrypt the files received from the client. The encryption keys are generated from the passphrase in the Options dialog box of the client or the text box of the server. If the passphrases are different, then the server will not be able to generate the proper key for decryption.

## 6.2 Experiments

There are three main experiments conducted using the CASM implementation. These experiments are used to show that the operation and use of CASM is feasible. The experiments described in this section include the baseline operation, an evaluation of the context metrics population list, and a study of a random sample of context metrics. Results are presented in Section 6.3.

### 6.2.1 Baseline Operation

The baseline operation experiment simply verifies the functionality of the CASM context module, the Pocket PC client application, and the server. The client interacts with the server in this experiment. Each of the options or items is triggered on the client application. Responses from the server side and client side are then noted. Important operations include proper file transmission, encryption and decryption, acquisition of context, use of correct decision engine and setting, and manual intervention of contexts. Files of different sizes are also used for this experiment.

### 6.2.2 Context Metrics Population

This experiment involves a predetermined population of context metrics. The context metrics are used to evaluate CASM's decision module. Each context metric combination is applied to each decision engine and each setting in an engine. The different settings for a decision engine are shown in Table 6.2.

**Table 6.2: Decision Engine Settings**

| Decision Engine | Settings |
|---|---|
| AHP | Balanced, Energy, Performance, Security |
| Deterministic | Rigid, Flexible |
| Modified AHP | Balanced, Energy, Performance, Security |

The context module design was described in Section 5.5 and the possible context metric values and context components were listed in Table 5.7. A modified version of Table 5.7 is shown as Table 6.3 to indicate values used in this experiment. Eliminating extremes, such as using an AC line or no signal present, the modified table contains the entire population of context metrics. The Total Metrics column indicates the number of context metrics for each context. The product of all the numbers in the Total Metrics column yields the total number of context metric combinations, which is 10,125. An example of a context combination is a situation where the user sets a Medium security level, the device has 70% battery life remaining, the device is in an Unknown location, there is Very Good signal strength, the object to be encrypted is 128 KB in length, and there have been 3 user interactions.

**Table 6.3: Context Metric Population List**

| Context Components | Context Metrics | Total Metrics |
|---|---|---|
| Security Level | Low, Medium, High | 3 |
| Energy | 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% | 9 |
| Location | Unknown, Lab, Office | 3 |
| Communications | Very Low, Low, Good, Very Good, Excellent | 5 |
| Object Size | 1 KB, 4 KB, 32 KB, 128 KB, 1 MB | 5 |
| User Interactions | 1, 3, 5, 7, 9 | 5 |

Since only the CASM decision module is utilized in this experiment, results can be gathered through a console application. Thus, the Pocket PC client and server are not needed for this experiment. The block cipher that the decision module chose for a given context combination represents one result.

### 6.2.3 Random Sample of Context Metrics

This experiment evaluates both the decision and security modules of CASM. A random sample taken from the context metric population list in Table 6.3 is used for this experiment. The sample size is 100 combinations. Each combination is chosen independently from the population and repetitions are possible. Each combination of context metrics from the random sample is applied to each decision engine and each setting in an engine.

For a single combination, the security module activates the chosen block cipher according to the operating parameters specified in Table 5.2. The security module then uses the block cipher to encrypt a file corresponding to the object size context component. For the plaintext data to be encrypted, five different files of different sizes were generated with random content. File sizes in bytes are $2^{10}$ (1,024 bytes or 1 KB), $2^{12}$ (4,096 bytes or 4 KB), $2^{15}$ (32,768 bytes or 32 KB), $2^{17}$ (131,072 bytes or 128 KB), and $2^{20}$ (1,048,576 bytes or 1 MB). For example, if the object size context metric was 32KB, then the selected block cipher encrypted the file that requires 32KB of data storage.

For each combination, the selected file was loaded into the memory of the iPAQ 4150. The unencrypted data blocks were kept static in memory for the duration of the experiments. Static keys were used for encrypting the files. In addition, the encrypted memory contents were flushed after each encryption process to eliminate the effects of caching. To minimize the current drawn from the battery, the iPAQ 4150 used its dimmest backlight setting, disabled its wireless adapter, and terminated all other active background programs before running each test.

The metrics and measurement methods described in Section 5.3.2 are used to evaluate each encryption task for a given context metric combination. The query performance frequency and query performance counter functions are first used to measure the encryption times. Then, the Agilent multimeter is used to measure the current drawn from the iPAQ 4150 battery, allowing us to compute the energy consumed for each encryption task. These measurements are required to calculate the metrics of encryption latency, encryption throughput, energy-latency product, and throughput/energy ratio.

## *6.3 Results and Analysis*

The three experiments detailed in Section 6.2 evaluate CASM's three modules. The experiments also help verify the operation and feasibility of CASM. The results and analysis for each of the three experiments are discussed in this section.

### 6.3.1 Baseline Operation

The client and server applications were used for this experiment. Each option functioned properly in the user interface. The Log tab in the Pocket PC client tracked context collection and algorithms chosen. The context values were retrieved successfully: battery life percentages displayed the correct amounts, signal strengths and SSIDs matched those displayed on the Pocket PC device, and the number of user interactions was correct. Encrypted files transmitted by the client were properly received and decrypted by the server. If the passphrase of the client was changed, while the passphrase at the server remained the same, then the server would still receive the encrypted file from the client, but would not be able to successfully decrypt the file. Overall, there were no problems with the baseline operation experiment and the context module functioned correctly.

### 6.3.2 Context Metrics Population

The context metrics from the population list in Table 6.3 were applied to the CASM decision module. All the settings and decision engines were employed for each context metric combination to determine the block cipher decision. The block ciphers were tallied according to the number of times that they were selected from a given decision engine and user setting. The percentages in Tables 6.4 through 6.6 report the frequency that a particular block cipher is used in a given setting of a decision engine.

**Table 6.4: Block Cipher Usage with the AHP Engine**

| Cipher | Balanced | Energy | Performance | Security |
|---|---:|---:|---:|---:|
| RC2 | 30.25% | 42.07% | 40.52% | 10.14% |
| Blowfish | 7.71% | 14.01% | 14.54% | 11.37% |
| XTEA | 10.79% | 21.06% | 21.01% | 16.05% |
| AES | 51.25% | 22.85% | 23.93% | 62.44% |

The block cipher usage for the AHP decision engine is summarized in Table 6.4. The AES cipher is used more frequently under the Balanced and Security settings than in the Energy and Performance settings. This is due to the fact that the AES cipher is always given more importance when contexts allow for ciphers that are slower and consume more resources. Combinations of high battery life percentages, strong signal strengths, secure locations, and small file sizes typically lead to selection of AES. RC2 is chosen when efficient operation is needed. Hence, the Energy and Performance settings yield the highest percent usage values of RC2.

**Table 6.5: Block Cipher Usage with the Deterministic Engine**

| Cipher | Flexible | Rigid |
|---|---:|---:|
| RC2 | 26.67% | 33.33% |
| Blowfish | 66.67% | 26.67% |
| XTEA | 6.67% | 6.67% |
| AES | 0.00% | 33.33% |

The decision results in Table 6.5 are from the Deterministic decision engine. Percentages are a direct reflection of how efficient the block ciphers are with respect to object size and security level. The Rigid setting shows that when the security level is set to Low, only RC2 is used, and when the security level is on High, only AES is chosen. At a Medium security level, Blowfish and XTEA are both used, with an emphasis on Blowfish because of its efficiency when encrypting files larger than 4 KB.

The Flexible setting allows for more block ciphers to be available at each security level, so decisions are mostly influenced by object size. Since the security resources cost-analysis deems Blowfish as an overall fast and efficient security algorithm and the Deterministic engine is based on the cost-analysis results, the Blowfish cipher is used the most in the Flexible setting. The RC2 cipher is used for almost one-third of the context situations, because the cipher is usually second to Blowfish in terms of speed and efficiency. XTEA is not used as frequently and AES is never used, because these algorithms are slower and less efficient.

**Table 6.6: Block Cipher Usage with the Modified AHP Engine**

| Cipher | Balanced | Energy | Performance | Security |
|---|---:|---:|---:|---:|
| RC2 | 0.64% | 2.24% | 2.46% | 4.65% |
| Blowfish | 52.12% | 77.25% | 75.89% | 26.00% |
| XTEA | 1.25% | 0.76% | 0.55% | 10.45% |
| AES | 45.99% | 19.74% | 21.10% | 58.89% |

The percent usage of a cipher operating under a particular user setting for the Modified AHP decision engine is shown in Table 6.6. Note that the Blowfish and AES ciphers are used the most regardless of the setting. This is due to the speed of Blowfish and the cryptographic strength of AES. Certain context combinations are in favor of using the strongest cipher, whereas other combinations require a faster one.

The block cipher choices are a result of both the objective matrix and options matrices in the decision engine. The options matrices for the AHP engine were derived from the literature. However, the options matrices for the Modified AHP engine were obtained from the security resources cost-analysis described in Section 5.3. For the iPAQ 4150, RC2 was found to not be as efficient as Blowfish in terms of encryption throughput and encryption latency. In addition, RC2 is more vulnerable to attacks than Blowfish, thus, Blowfish is ranked higher in importance in the options matrices for the Modified AHP decision engine. These rankings in the options matrices are reflected in the percent usage values in Tables 6.4 and 6.6.

Another interesting observation is the similarity between the results of the Energy and Performance settings of both AHP and Modified AHP engines. The difference between the Energy and Performance settings is the emphasis on the energy context and the communications context. The weights of object size are also a priority in these settings, but the weights have the same value in both settings. The primary reason for the similarities between the results is that the options matrices for the energy, communications, and object size contexts are all the same in the AHP engine. The Modified AHP engine has slightly adjusted weights within the options matrices, but the matrices are still identical for the three components of context. This leads to the conclusion that equal context weights on different context components with identical options matrices will lead to similar results.

### 6.3.3 Random Sample of Context Metrics

A random sample of 100 context metric combinations was generated from the context metric population list in Table 6.3. Each independent combination is applied through the

CASM decision module which then activates the CASM security module. Each user setting within a decision engine is tested with the random sample, implying 100 encryption tasks per user setting. There are a total of ten different user setting variations in the CASM decision module, thus, a thousand encryption tasks in total were performed.

#### *6.3.3.1 Raw Metrics*

The four metrics detailed in Section 5.3.2 are used to evaluate the decision module in this experiment. The four metrics utilized are based on the two raw metrics, encryption time and energy consumption, that measured the security module responses. These metrics are considered "raw" because they are used to calculate the metrics defined in the next section. However, to begin understanding how the results are characterized, the raw metrics need to be investigated first.

The plot and chart in Figure 6.8 show the distribution of the encryption times that were collected under the AHP Balanced setting. The chart is a histogram of the encryption times with bins logarithmically spaced apart and frequency readings on the left of the figure. Each bar represents the number of tasks completed by the time range in its bin. The histogram is roughly divided into five groups of bars. This division reflects the approximate times needed to encrypt the five different file sizes using the slowest block cipher. The rightmost bar indicates that four files required more than ten seconds to encrypt. Checking the raw results, four of the largest sized files (1 MB) were indeed encrypted using AES.

The line that is plotted is a cumulative distribution function (cdf) of the encryption times. It is also plotted on a logarithmic time scale. Percentages of the total time are labeled on the right side of the figure. Note that less than half a second is needed to complete about 70 of the 100 encryption tasks.

The distribution of the energy consumed per task in the random sample is displayed in Figure 6.9. The histogram bars are also generally separated into five groups. Each bar group is associated with the amount of energy needed to encrypt a file using the slowest block cipher. Furthermore, according to the line plot, approximately 70 of the 100 encryption tasks require less than 0.02 J of energy.

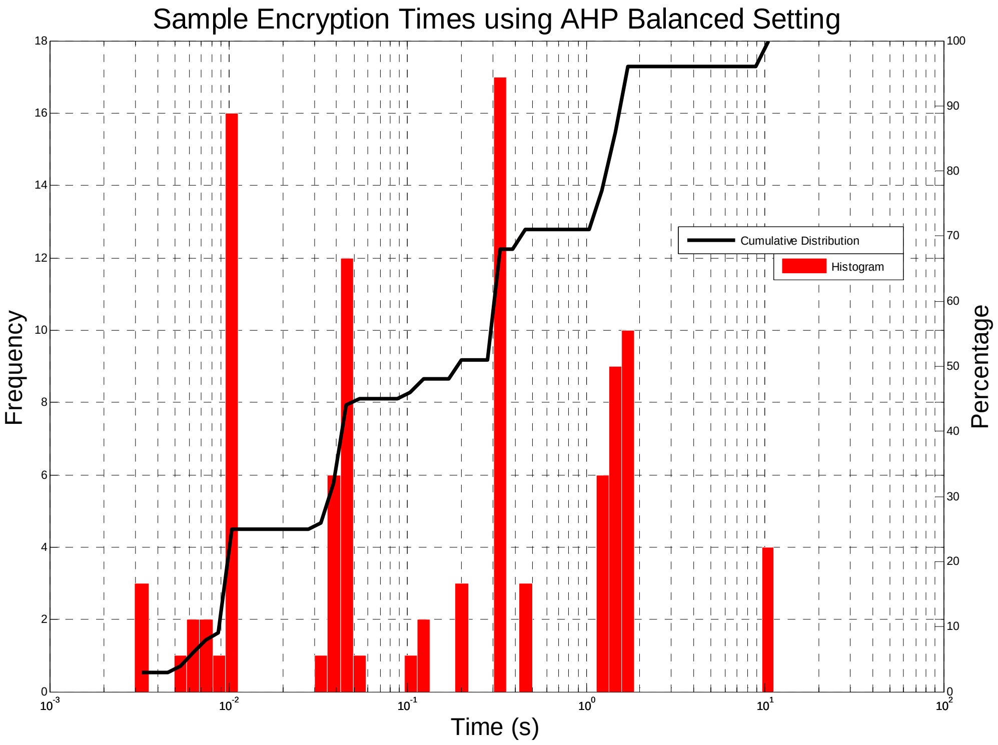


Figure 6.8.  Sample encryption times using AHP Balanced setting.

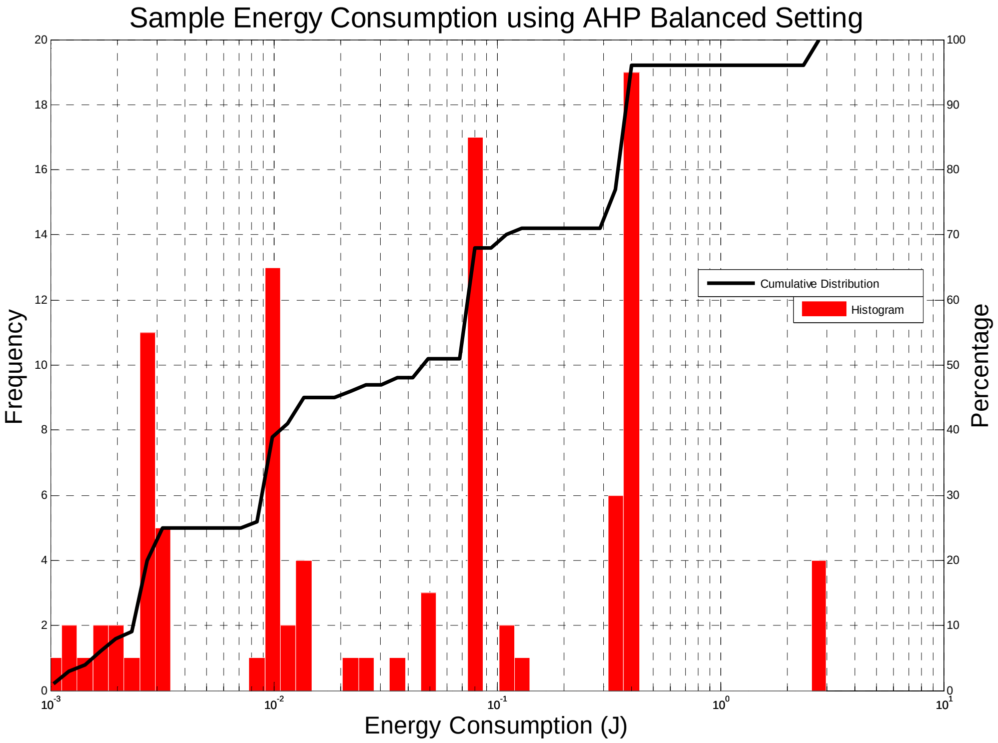


Figure 6.9.  Sample energy consumption using AHP Balanced setting.

Using the cumulative distribution as a guideline, the other user settings can also be plotted and visually compared. The plots in Figures 6.10 through 6.12 show cumulative distribution functions (CDFs) of encryption times for the three decision engines. Each figure has plots representing the different user settings of a decision engine. Of the four CDFs in Figure 6.10, the Energy and Performance CDFs reach their 100% limits earliest, with the Energy CDF slightly outperforming the Performance cdf. The Balanced and Security CDFs are relatively close to each other. It appears that there is a significant difference between the Energy/Performance and Balanced/Security plots. However, whether or not the difference is statistically significant requires analysis of variance (ANOVA) tests.

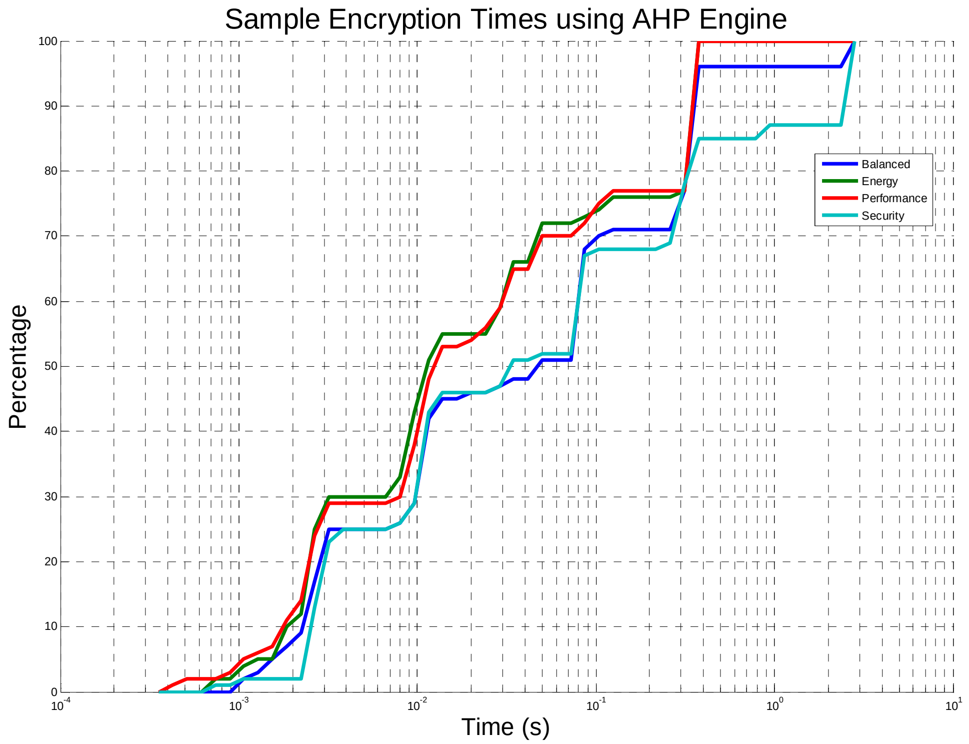


Figure 6.10. Sample encryption times using AHP engine.

There are only two CDFs for the Deterministic engine, which are shown in Figure 6.11. There is clearly a visually difference between the CDFs for the Flexible and Rigid settings. Given that the Flexible setting allows for more block cipher choices at a designated security level, it makes sense that the fastest ciphers will be selected. The Rigid setting restricts the use of the block ciphers based on their level of security. The graph reveals that all 100 encryption tasks in the experiment's sample require less than 0.8 seconds under the Flexible setting, but less than 11 seconds under the Rigid setting.

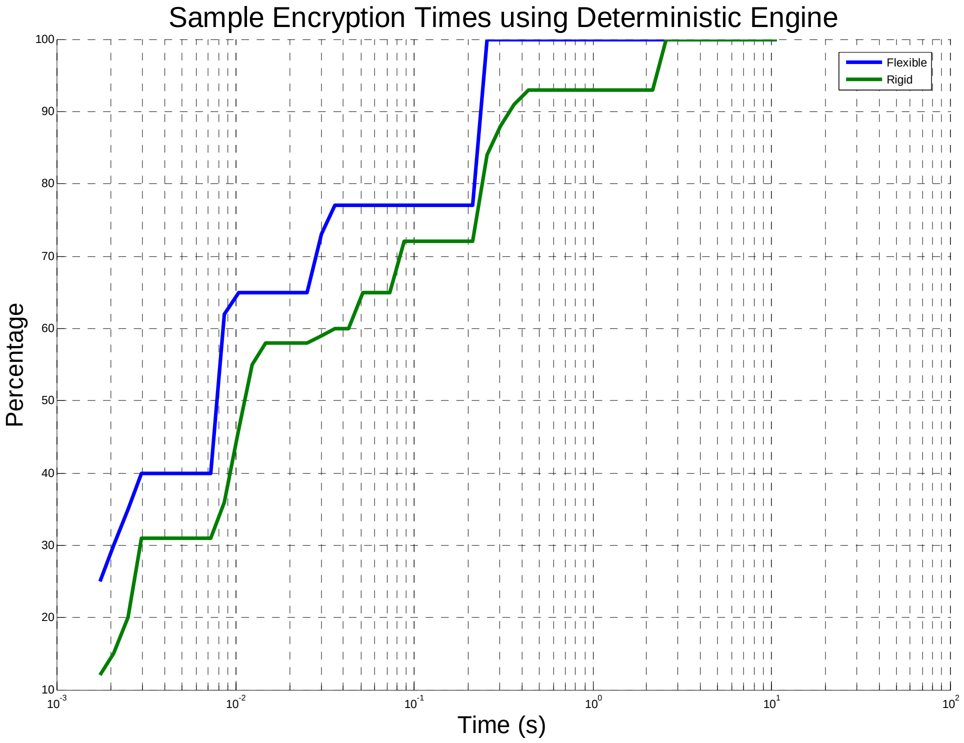


Figure 6.11. Sample encryption times using Deterministic engine.

The four CDFs in Figure 6.12 characterize the four settings in the Modified AHP decision engine. The behavior of the functions is similar to that of the CDFs representing the AHP decision engine. The Energy and Performance plots are further to the left than the Balanced and Security plots. Again, whether or not this difference is statistically significant requires ANOVA testing.

The cumulative distributions of the energy consumption values for each decision engine are illustrated in Figures 6.13 through 6.15. The energy consumption CDFs have characteristics that are similar to the CDFs for the encryption time. Since the task's energy consumption is calculated by multiplying the measured current from the multimeter by nominal voltage and encryption task time, the CDFs for energy consumption and encryption time should be highly correlated. In fact, the Pearson product-moment correlation coefficients between corresponding encryption time and energy consumption CDFs are all over 0.99.

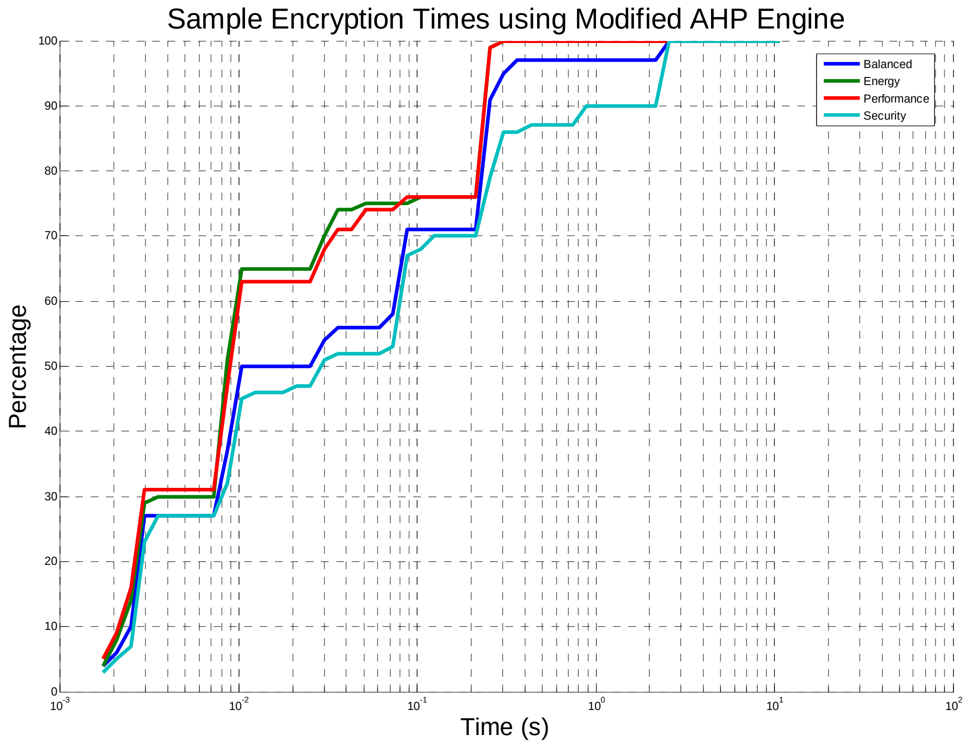


Figure 6.12. Sample encryption times using Modified AHP engine.

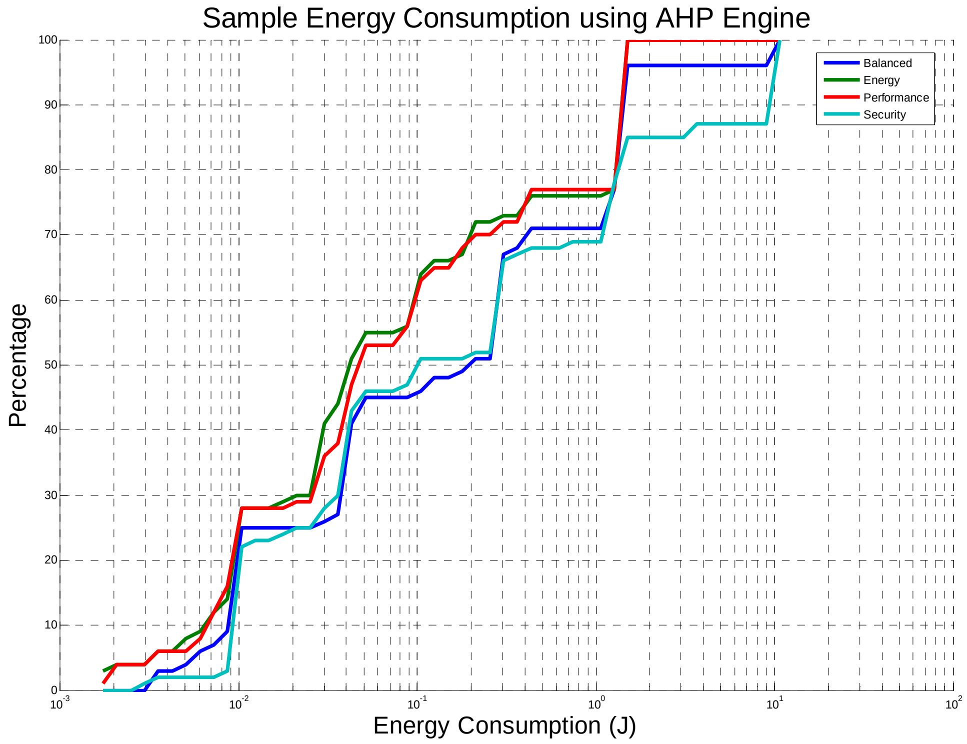


Figure 6.13. Sample energy consumption using AHP engine.

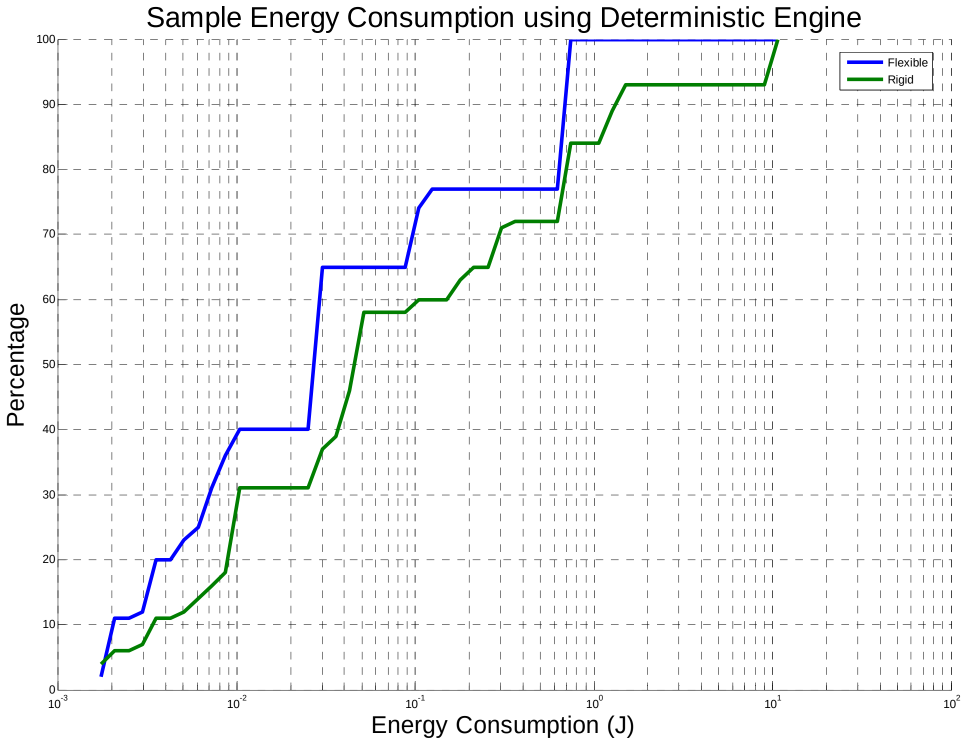


Figure 6.14. Sample energy consumption using Deterministic engine.

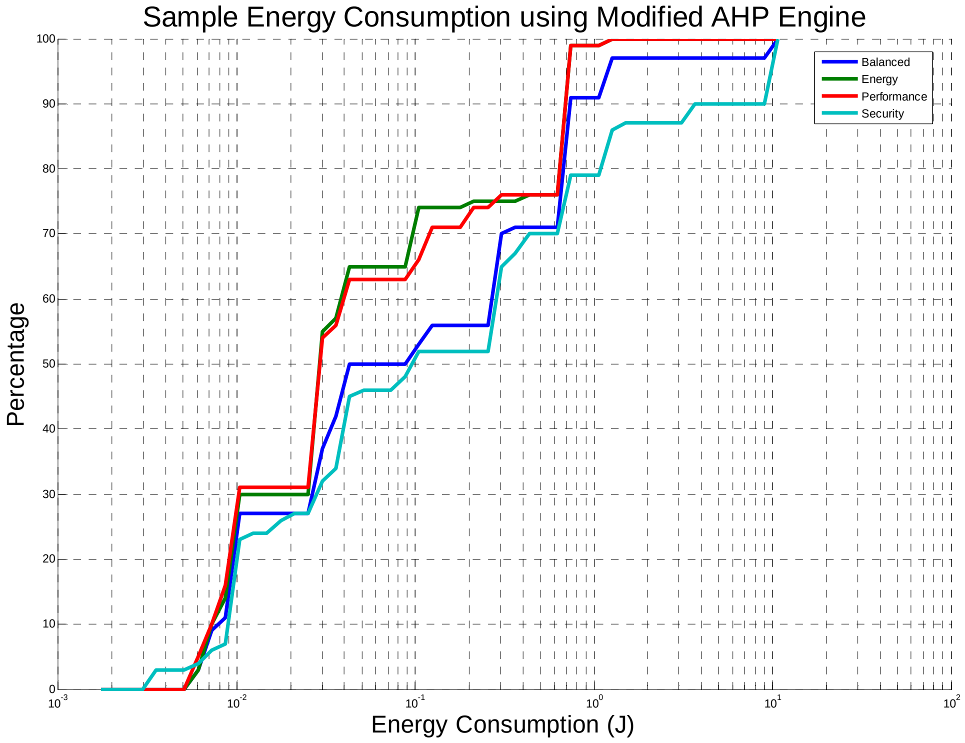


Figure 6.15. Sample energy consumption using Modified AHP engine.

### *6.3.3.2 Calculated Metrics*

The two raw metrics, encryption time and energy consumption, are used to calculate the four metrics described in Section 5.3.2. These four metrics are then used to evaluate the different decision engines and different user settings. In addition, a new metric, called *security strength*, is computed based on the block cipher selected for an encryption task. If RC2 is chosen, the security strength value is 1, if Blowfish or XTEA are selected, the security strength value is 2, and if AES is used, the security strength value is 3. These security strength values are based on the strengths of block ciphers as reported in the literature [33, 34, 36-38, 40, 41, 60-62]. RC2 is considered the most vulnerable of the four ciphers. Blowfish and XTEA, though not as vulnerable, have not been as extensively analyzed as AES. Additionally, AES is considered secure enough by the U.S. government to be its encryption standard for now.

After obtaining the security strength for each encryption task, its average value can be found for each user setting. The means for the other four calculated metrics – encryption latency, encryption throughput, the energy-latency product, and the throughput/energy ratio – are also calculated. These average values are tabulated in Table 6.7.

Although there are clearly differences in the averages between certain user settings, these differences can be more clearly seen in charts. Latency and throughput metrics are also counter to each other. For example, low latency and high throughput may be a result of a highly efficient block cipher, but a cipher with a low latency and low throughput would not be considered as efficient. The average latency and throughput are separated into two bar charts, shown in Figures 6.16 and 6.17. Each metric is normalized by first summing all the metric values across the ten user settings and then dividing each metric by the sum. The security strength metric is also normalized and included in the chart in Figure 6.16. The security strength and throughput metrics are grouped together, because high values in these metrics indicate a more efficient and secure security manager. In contrast, the latency metrics are grouped in a separate bar chart, shown in Figure 6.17, because low latency values suggest a security manager that responds more quickly to user encryption requests.

**Table 6.7: Metric Averages for Each User Setting**

| User Setting | Security Strength | Encryption Latency (μs/byte) | Encryption Throughput (KBps) | Energy-Latency Product (μJ/byte) | Throughput /Energy Ratio (MB/J) |
|---|---|---|---|---|---|
| AHP Balanced | 2.27 | 6.83 | 291 | 1.757 | 1.095 |
| AHP Energy | 1.82 | 4.01 | 489 | 1.094 | 1.693 |
| AHP Performance | 1.79 | 4.14 | 461 | 1.098 | 1.658 |
| AHP Security | 2.59 | 7.86 | 236 | 2.027 | 0.852 |
| Deterministic Flexible | 1.78 | 1.41 | 885 | 0.446 | 2.887 |
| Deterministic Rigid | 2.05 | 4.77 | 515 | 1.292 | 1.687 |
| Modified AHP Balanced | 2.51 | 6.06 | 521 | 1.587 | 1.704 |
| Modified AHP Energy | 2.22 | 3.70 | 752 | 0.995 | 2.446 |
| Modified AHP Performance | 2.20 | 3.72 | 729 | 1.006 | 2.388 |
| Modified AHP Security | 2.58 | 7.11 | 327 | 1.835 | 1.125 |

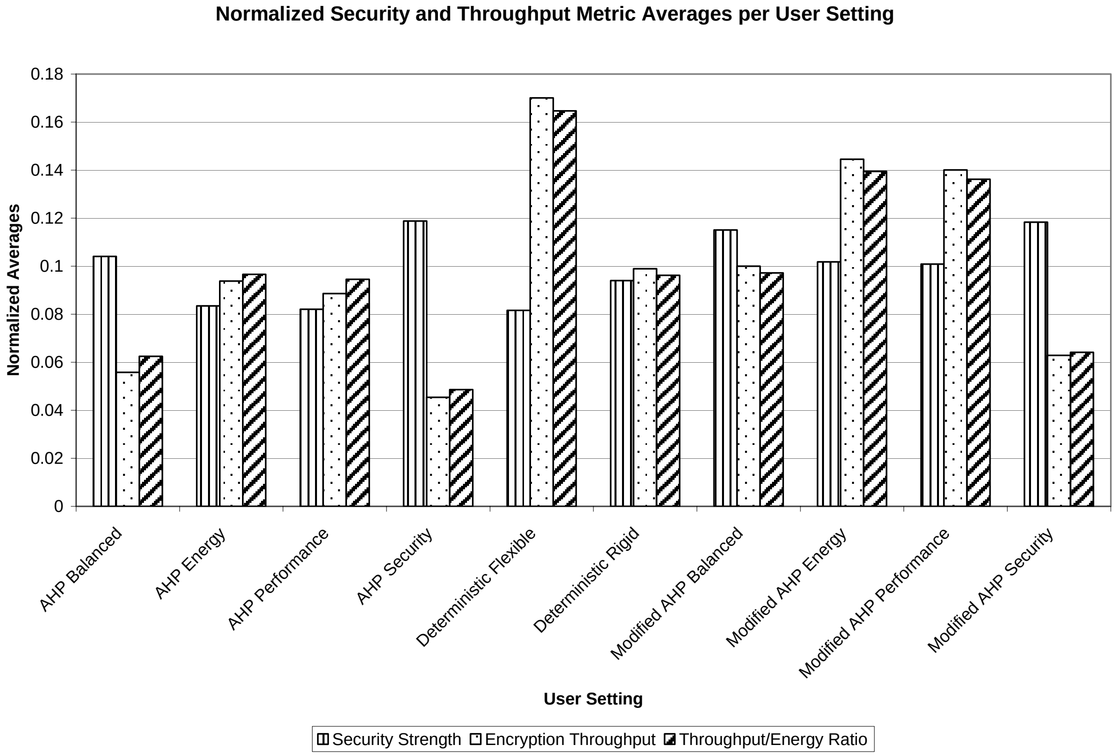


Figure 6.16. Normalized security strength and throughput metrics for each user setting.

There is a definite tradeoff between throughput and strength of security, as revealed in Figure 6.16. The Deterministic decision engine's Flexible user setting leads to the highest relative encryption and throughput/energy ratio among the ten user settings, but it also has the weakest security rating. Conversely, the AHP decision engine's Security setting has the highest relative security strength, but the lowest throughput values. User settings with balanced, but high, security and throughput averages include the Balanced, Energy, and Performance settings of the Modified AHP decision engine. The Deterministic decision engine's Rigid user setting is also relatively well balanced and has the next highest security and throughput metric averages.

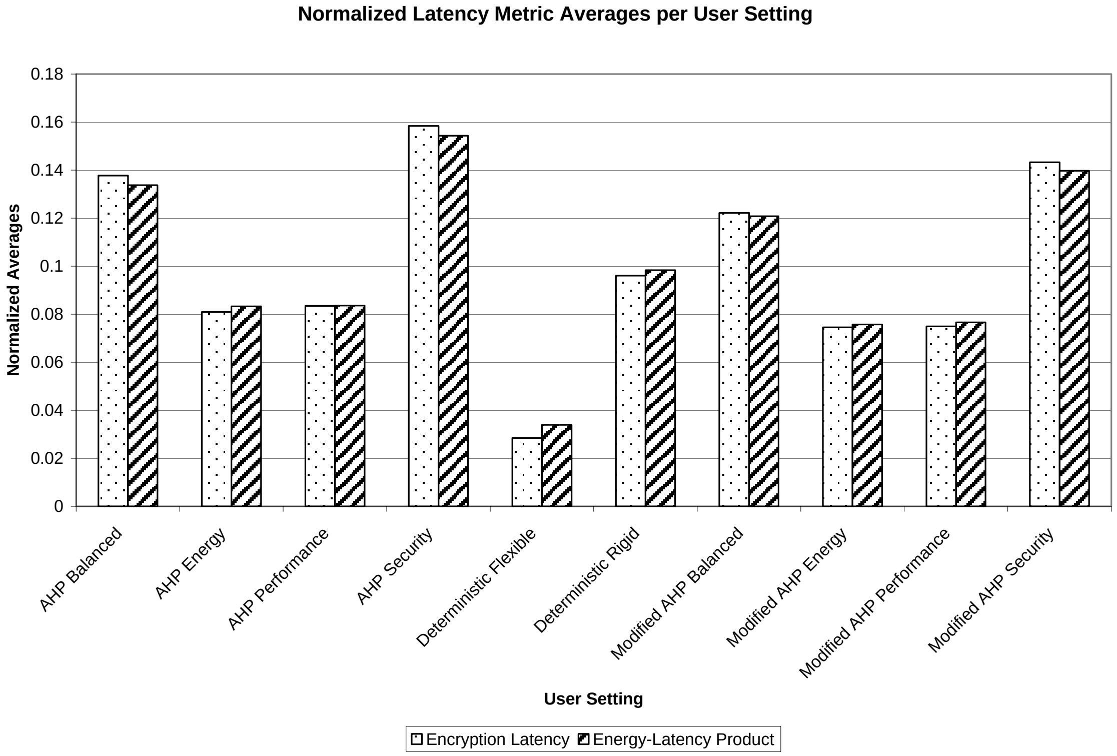


Figure 6.17. Normalized latency metrics for each user setting.

The bar chart in Figure 6.17 displays the normalized latency metric averages for each of the ten user settings. For this chart, low values denote a user setting that would allow CASM to perform more efficiently. The Deterministic decision engine's Flexible user setting has the lowest relative encryption latency and energy latency product values, followed by the Modified AHP decision engine's Energy and Performance settings. Even though the Deterministic decision engine's Flexible setting appears to be the best in terms of latency metrics, if security strength is considered, the overall rating may differ. For this purpose, a table was created to show relative ranking of the five calculated metrics for all user settings.

Table 6.8: Normalized Metric Rankings for Each User Setting

| User Setting | Security Strength | Encryption Latency | Encryption Throughput | Energy-Latency Product | Throughput /Energy Ratio |
|---|---|---|---|---|---|
| AHP Balanced | 4 | 8 | 9 | 8 | 9 |
| AHP Energy | 8 | 4 | 6 | 4 | 5 |
| AHP Performance | 9 | 5 | 7 | 5 | 7 |
| AHP Security | 1 | 10 | 10 | 10 | 10 |
| Deterministic Flexible | 10 | 1 | 1 | 1 | 1 |
| Deterministic Rigid | 7 | 6 | 5 | 6 | 6 |
| Modified AHP Balanced | 3 | 7 | 4 | 7 | 4 |
| Modified AHP Energy | 5 | 2 | 2 | 2 | 2 |
| Modified AHP Performance | 6 | 3 | 3 | 3 | 3 |
| Modified AHP Security | 2 | 9 | 8 | 9 | 8 |

The normalized metrics in Table 6.8 are ranked for each user setting. The ranks were created from the order of the best to worst metric values from Table 6.7. A ranking value of 1 means the user setting has the best value for that particular metric, whereas a rank of 10 means the user setting has the worst value.

I have labeled the Energy and Performance settings of the Modified AHP as the best choices for the CASM decision module. Both rank high in the throughput and latency metrics and fair in security strength. Referring back to Table 6.7, the security strengths of these settings are on average around 2.2. This indicates that XTEA or Blowfish were used the most for the sample combinations of context values, with some use of AES. Since the design was based on the block cipher cost-analysis, the rankings are justified. Blowfish was given the most priority in the options matrices of the Modified AHP engine when efficiency was needed and AES was prioritized when security was needed. However, the Modified AHP decision engine's Balanced and Security settings do not rank as well overall. Even though the Security setting has the second strongest ranking in security, the other metrics come in near last. The Modified AHP decision engine's Balanced setting would have almost been a good choice, but the latency metric values are approximately 1.5 times as high as the Modified AHP decision engine's Energy and Performance latency metrics.

The Deterministic decision engine yields average security strength for the Rigid setting and the lowest security strength for the Flexible setting. Having the lowest security strength is why the Flexible setting is not considered a good choice, even though it has the highest throughput and lowest latency. Another reason to disregard the Flexible setting is that AES is never used throughout the context metric population (see Table 6.5). The Rigid setting would be a good second choice behind the Energy and Performance settings of the Modified AHP engine. The security choices are well balanced and metrics are, overall, about average.

The AHP decision engine's Energy setting would also be a good second choice for the CASM decision module. Security strength may rank below average, but the throughput and latency metrics rank slightly above average. The AHP decision engine's Performance rankings are slightly lower, thus, it would not be a viable choice for the decision module. The AHP decision engine's Balanced and Security settings both have the worst rankings in efficiency, regardless of the high security strength values. These two settings were also not chosen to be used in the CASM decision module.

## 6.4 Discussion

The primary use of CASM, as envisioned in this research, is as middleware for applications operating in a pervasive networking environment. The previous sections in this chapter described the implementation, experiments, and analysis of CASM. This section discusses some open issues and attempts to answer some potential questions.

The CASM security module contains four block ciphers. In the analysis of the CASM experiments no comparisons were made between the performance of the decision engines and performance of the individual block ciphers. Comparisons could be made by altering the context metrics for the decision engines while keeping the block cipher selections static. However, comparisons are not deemed necessary because analysis of the block ciphers has already been completed in Section 5.3. Additionally, the Modified AHP decision engine is based on the results of the block cipher cost-analysis and comparing it to the other decision engines would be similar to comparing the results of the individual block ciphers.

Another issue arises from the comparison discussion, which is the number of user settings and engines. There are a total of ten user setting and decision engine combinations. This provides the user with variability in CASM operation, but too many settings may be confusing for the user. The random sample analysis in Section 6.3.3 regards the Modified AHP decision engine's Energy and Performance settings as, overall, yielding the most efficient and secure operation. However, this analysis was based only on relative comparisons of means. A few simple statistical tests are needed to give more credibility to the random sample analysis. In addition, the number of engines and settings can be reduced.

I used a one-way multivariate analysis of variance (MANOVA) [97] procedure to compare the multivariate means of the calculated metrics, grouped by the ten user settings. The function returns *d*, an estimate of the dimension of the space containing the group means. The MANOVA procedure tests the null hypothesis that the means of each group are the same five-dimensional multivariate vector and that any difference observed in the sample metrics is due to random chance. If $d = 0$, there is no evidence to reject that hypothesis. If $d = 1$, then the null hypothesis can be rejected at the 5% level, but we cannot reject the hypothesis that the multivariate means lie on the same line. Similarly, if $d = 2$ the multivariate means may lie on the same plane in five-dimensional space, but not on the same line. The estimate returned from the MANOVA function was $d = 2$, which implies the multivariate means of the metrics lie on the same plane. The *p* values of the null and first dimensions were close to zero, granting statistically significant results.

Next, a dendrogram plot [98-103] of the group means was generated using the MANOVA results. The dendrogram plot of the user settings is shown in Figure 6.18. A dendrogram consists of many U-shaped lines connecting objects in a hierarchical tree. The height of each U represents the distance between the two objects being connected. The clusters are computed by applying the average linkage method[22] to the matrix of Mahalanobis distances between group means. The Mahalanobis distance is a multivariate measure of the separation of a data set from a point in space. It is the criterion minimized in linear discriminant analysis.

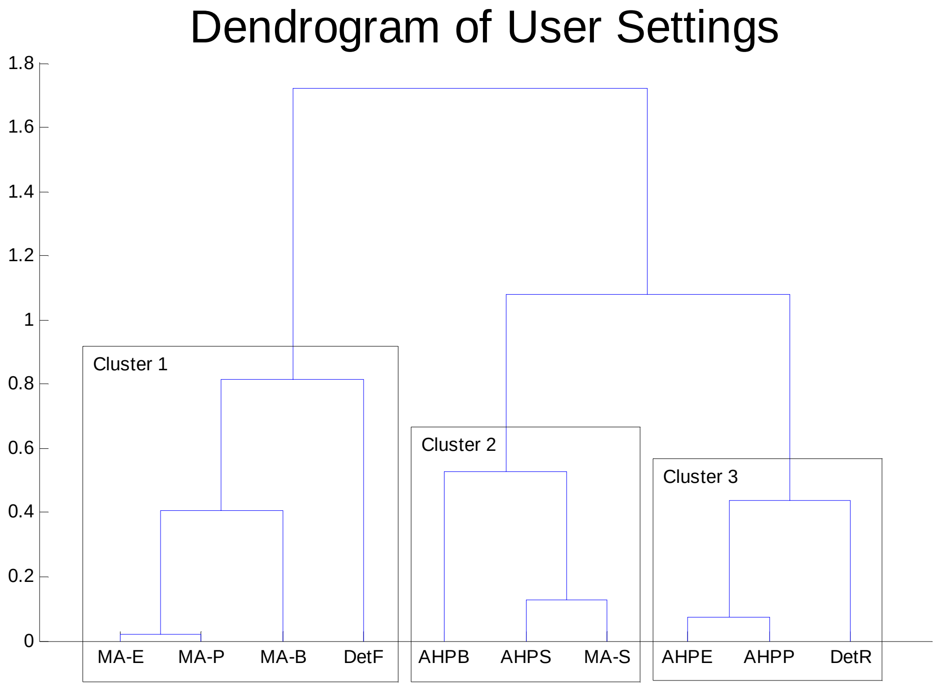


Figure 6.18. Dendrogram of user settings.

[22] Unweighted Pair Group Method with Arithmatic Mean (UPGMA)

Three clusters are labeled in the dendrogram. The first cluster contains the Deterministic decision engine's Flexible (DetF) setting and Modified AHP decision engine's Balanced (MA-B), Energy (MA-E), and Performance (MA-P) settings. The second cluster contains the AHP decision engine's Balanced (AHPB) and Security (AHPS) settings and the Modified AHP decision engine's Security (MA-S) setting. The final cluster contains the AHP decision engine's Energy (AHPE) and Performance (AHPP) settings and the Deterministic decision engine's Rigid (DetR) setting. User settings in Cluster 1 provide the most efficiency, the settings in Cluster 2 give the most security, and the user settings in Cluster 3 are somewhere in between. We can combine the statistical results of the MANOVA with the comparative results in Section 6.3.3 to simplify the number of useful settings available to the user and reduce repetition of settings that may produce similar results. Using the combined analysis, the number of settings can be reduced to three: the Modified AHP decision engine's Energy setting from Cluster 1, the Deterministic decision engine's Rigid setting from Cluster 3, and the AHP decision engine's Security setting from Cluster 2. These three settings range from most efficient operation with average security strength to the least efficient with good security strength.

However, a bias may exist when using only the Modified AHP and Deterministic decision engines because they are both "tuned" to the particular Pocket PCs used in these experiments. If the same code, including values in decision matrices, were to be run on different Pocket PCs or other devices, the settings choices may be different after analysis. Even if the resource cost experiments were re-run on each new device and matrix values updated for the Deterministic and Modified AHP decision engines, the settings choices may still differ. In other words, the three choices from the combined analysis may not be the best three choices for other devices. Future work on different devices and environments needs to be done to address this potential limitation.

Nevertheless, the combined analysis reduces the burden on users by reducing the number of user settings to adjust, but there is another issue with the choice of block ciphers in a multiple user environment. If one user wishes to share or transmit an object to a peer, the block cipher selection needs to be resolved before encrypting the object. Otherwise, the peer will not be able to decipher the object. The resolution process can be done in a few different ways. For the implementation in the CASM experiments, the easiest method is used, which is sending a header containing the encryption algorithm choice and allowing the receiver to decrypt the object using the information from the header. This method is simple, because the object provider controls the level of security based on the provider's environment. Receivers of the objects, or object requestors, have to comply with the algorithm choice. The disadvantage to this method is that the decision is based on the object provider's environment of which the object provider is aware. The object receiver or requestor's context is not considered, although some aspects of the receiver or requestor's context are the same as the object provider's. Another method that can resolve the security selection between users is through a negotiation mechanism. There would be requests from the object provider and object requestor for their preferred algorithm from their own contexts, then a resolution function would determine the final decision. Finally, another way to choose the security among a group of users is by using a game theory approach. Game theory deals with multi-person decision making, in which each decision maker tries to maximize his utility. Each user cooperates to maximize the performance and efficiency of securing shared objects. Each of these three methods has its own merits and weakness, but implementing and comparing them is beyond the scope of this research.

### *6.5 Summary*

This chapter describes the implementation of CASM and related experiments. The experiments and analysis mostly constitute a proof of concept or feasibility study of CASM. Three different experimental scenarios were used in this research. The experiments were used to evaluate the three modules in CASM. The results and analysis of the experiments indicate that CASM functions properly. Security is provided efficiently with different user settings. Three schemes were deemed the best to use for the CASM decision module, which are the Modified AHP engine with the Energy setting, the Deterministic engine with the Rigid setting, and the AHP engine with the Security setting.

# Chapter 7. Conclusions

This dissertation contributed to the field of wireless network security by developing a methodology to efficiently secure wireless networks. Moving beyond the methodology, a context aware and adaptive security manager (CASM), with application to a pervasive networking environment, was designed and evaluated.

## 7.1 Summary and Discussion

This research had two main goals. The first goal was to provide a methodology that can be used to determine the most suitable security protocol for a specific wireless network application based on certain operational constraints and characteristics. The second goal, which built on the first, was to realize and evaluate a method for wireless network security to adapt based on its context in certain environments.

One security solution will not operate efficiently for all wireless networks and all computing devices. Thus, wireless network applications needed to be separated into broad categories, such that an appropriate security solution can be identified for each category. Background and a literature review of wireless network standards, wireless systems and protocols, security technology, energy efficiency, and adaptive security technology was presented in Chapter 2.

On reviewing the different types of wireless networks, it was found that some characteristics are similar between certain wireless networks. This led to the approach of constructing the methodology by first categorizing the wireless networks. Certain security schemes may operate more efficiently for different categories of wireless networks. However, we realized that sometimes it is not appropriate to use a certain security algorithm in a specific type of network for all possible situations. Thus, the idea of adapting security to the current context as an evolution of the methodology was introduced in Chapter 3.

The methodology, described in Chapter 4, begins with classifying different types of wireless networks into different categories. Operational parameters and security requirements are used to create the categories. By separating wireless networks into broad categories, they become more manageable in terms of assessing how to best secure them. It becomes easier to evaluate the impact of different security mechanisms. The impact evaluation is needed to determine the most suitable security mechanism for a given category of wireless network.

The methodology is useful for wireless networks operating in environments that are static or have low variability with respect to the characteristics of their deployment. However, if a wireless network operates in an environment that rapidly fluctuates or switches between extreme or even substantially different conditions, then an adaptive solution is required. The design of such a mechanism that is context aware and adapts based on the changes in context was described in detail in Chapter 5. This mechanism is called the

Context aware Adaptive Security Manager (CASM). CASM incorporates three modules. The first module collects the context, the second module makes decisions based on the context, and the third module secures the wireless network.

In Chapter 6, I discussed how I implemented CASM, specifically for application in a pervasive networking environment. I used three different experimental scenarios to evaluate the operation of CASM. This was done to study feasibility. Different user settings and different decision engines were compared to each other in terms of security, performance, and energy efficiency. Finally, an appendix describes the core process and mathematics behind the CASM decision engines.

## 7.2 Contributions

This research contributes in the following ways to the fields of network engineering and wireless security.

1. This research defined operational parameters that can be used to represent the characteristics of different wireless networks with respect to the performance and efficiency of data encryption. Every wireless application will have certain network characteristics and constraints that can be used for classification. The combination of all the operational parameters for wireless network characterization is a novelty approach proposed in this dissertation. Similarities between wireless networks can be found with these parameters. New applications of wireless networks can also be easily classified with the operational parameters.

2. This research demonstrated how wireless networks can be separated into manageable categories based on their properties with respect to data encryption. Classifying wireless networks into distinct categories localizes certain security issues to that category and facilitates finding suitable security schemes for a given wireless application. Wireless network categorization has been attempted on smaller scales [48, 74] and for specific applications [19, 48, 72, 104, 105]. However, this methodology encompasses a larger set of parameters and can potentially create more diverse categories. The classification process can even be extended to previously undiscovered wireless network applications or standards.

3. This research led to a methodology that defines and utilizes metrics to determine the resource impact of a security scheme on each category. The metrics defined in the methodology are typically used separately for assessing performance, overhead, energy consumption, or rate of energy consumed [48, 73]. Using the metrics together can improve analysis of security protocols. Additionally, four block ciphers (RC2, Blowfish, XTEA, and AES) were investigated in terms of resource consumption by using the metrics in this methodology. Results for both performance and energy consumption were presented for different ciphers applied to plaintext files of different sizes. The experiments and analysis are intended to increase awareness of and insight into the processing and energy costs associated with utilizing certain block ciphers on PDAs and other resource-limited devices.

4. These parameters, categories, and metrics were combined into a methodology for evaluating security for wireless networks. This methodology is used to categorize wireless networks and evaluate security protocols while addressing security, energy, and performance problems together, unlike other methodologies [65-72] that assess the problems individually. The procedure first identifies the operational parameters of a wireless network application. These parameters are compared to the parameters of existing wireless network categories to determine any major similarities. Different security protocols can be incorporated individually to a specific category and analyzed using the metrics in the methodology. A follow-up analysis is done to ascertain how suitable a particular security protocol is for a given category. An advantage of the methodology is that previously analyzed categories would already have information on suitable security protocols for new wireless network applications that are classified into that category.

5. A context aware and adaptive wireless network security manager, CASM, was designed, implemented, and evaluated. The difference between CASM and most context aware applications [67, 85-88] is that we focus only on adapting security instead of general-purpose tasks. CASM requires context to provide itself with relevant information to perform its task of appropriately securing data. The context is used to decide on the most suitable security protocol in different situations. The benefits of using CASM in a system are gains in performance and energy efficiency when using security algorithms to encrypt data. Different settings also provide CASM with flexibility towards user preferences. These user settings were evaluated and compared. This resulted in three groups of settings, from the most efficient to the most secure.

These contributions are important because wireless network security needs to be suited to its environment and not just the wireless network deployment requirements. Protocol and system developers can use the methodology to better understand particular wireless networks and determine how to secure them appropriately. Security protocols can also be used more efficiently when the context is known. Finally, adaptive methods can help conserve resources that may be utilized elsewhere in wireless network devices.

## *7.3 Applications of this Research*

This research can be applied to most commercial wireless networks, systems, and devices. The methodology in this dissertation is intended to assist in designing, implementing, or managing a wireless network to provide services to mobile users or devices. This work can also be used in the development phase to incorporate appropriate security constraints and protocols and avoid expensive retrofitting after implementation.

The application of the three CASM modules as a system is an original contribution from this research. CASM's functionality could be implemented as middleware for handheld wireless network devices. CASM can also be advantageous to implement in other types of battery-powered network devices, such as notebook computers and small sensor nodes. Other wireless systems, like MANETs or IEEE 802.16, can use the wireless channel conditions as context to determine security levels and allocation of resources. In telecommunications, CASM can be part of an automated configuration management system that could provide dynamic functions and configuration for security and, more broadly, information assurance services. In short, there are many ways that CASM is of use to the fields of wireless networks, network engineering, and security.

## *7.4 Future Research*

This dissertation has created a methodology for securing wireless networks and a context aware and adaptive security manager for selecting a suitable encryption scheme, especially for a pervasive computing environment. Several components and aspects of this research could be extended as interesting future research projects.

First, wireless networks are deployed in many different configurations. The number of categories presented in this dissertation represents only a handful of the most popular wireless network applications. Other wireless networks have yet to be categorized or even invented. The list of categories can be expanded. More case studies can be done to further validate the operation of the methodology.

Although individual standards, systems, and protocols may be cryptographically sound by design, implementing and incorporating them in practice may introduce unexpected weaknesses. Vulnerabilities in a system could be severe enough for an intruder to completely bypass the suitable security protocols found by the methodology. Thus, the methodology itself could be further developed to include vulnerability analysis. This process can be added as the last step of the methodology.

Several other potential areas of research involve CASM and its components. For the context module, increasing or decreasing the number of context-related parameters will have a certain effect on the decisions or actions, as prior research in adaptive schemes has mentioned [53-57]. The number of context components can be altered to determine the amount of information that needs to be processed at the decision maker. Context components that are related to the operational parameters of the methodology can be included to improve the feedback of a changing environment with many characteristics. The number of context metrics within a context can also be varied.

The CASM decision module can potentially be improved by integrating other decision engines. More decision engines would allow users to have more options, but the decision engines first need to be analyzed. Decision making in a multi-user environment could pose some interesting problems. The scalability of CASM has not been tested and many decisions or negotiations being made for each user could reduce the efficiency of the entire system. This study needs to be done to improve the credibility of CASM.

Security in CASM only relies on encryption and this is further restricted to four block ciphers. Different authentication and authorization mechanisms could be included to further secure object data and maintain access rights. Stream encryption may also be integrated for different types of objects, such as streaming audio or position information. Public-key cryptography can be used in situations where resources are abundant for a wireless device, for instance, having a high battery life and fast wireless connection.

Adding more security features can increase the number of security levels and decrease the differences between adjacent security levels, but decisions may not always be the most suitable. Thus, more research in this area can improve the quality of the decisions.

Finally, although CASM is an adaptive system, the settings and modules are static and to change them requires user intervention or recompiling of libraries. Dynamic settings and automatic acquisition of context would place the user in a more immersive and pervasive application. A type of *cognitive security* is needed, not in the psychological sense, but similar to the concept of a cognitive radio [106]. A system using cognitive security would be able to sense the user environment and learn how to adapt to threats, volatile environments, and surges of resource utilization. Distribution of security-related cognition could improve the overall security of a wireless network, better than an adaptive security mechanism. The approach taken for cognitive security would be similar to that of cognitive radios. Different learning methods would first need to be researched for this topic to reach fruition.

## 7.5 Summary

This dissertation has reviewed past literature on wireless networks, energy efficiency, security, and adaptive technology. I have created a methodology that can be exercised to find encryption schemes suitable for different types of wireless networks. I have also developed a security manager called CASM that adapts based on its context. Finally, I implemented CASM for use in a pervasive computing environment and performed a proof of concept study.

# References

[1] R. A. Stanley, "Wireless LAN Risks and Vulnerabilities," White Paper, Information Systems Audit and Control Foundation, 2002. Available at http://www.isaca.org/.

[2] T. Karygiannis and L. Owens, "Wireless Network Security: 802.11, Bluetooth, and Handheld Devices," National Institute of Standards and Technology, U.S. Department of Commerce, November 2002. Available at http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf.

[3] D. Welch and S. Lathrop, "Wireless Security Threat Taxonomy," in *Proc. IEEE Systems, Man and Cybernetics Society Information Assurance Workshop*, 2003, pp. 76-83.

[4] B. Potter, "Wireless Security's Future," *IEEE Security & Privacy Magazine*, vol. 1, no. 4, pp. 68-72, 2003.

[5] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed., Upper Saddle River, NJ, Prentice Hall, 2001.

[6] Defense Information Systems Agency, "Wireless Security Technical Implementation Guide," Version 3, Release 1, April 2004. Available at http://csrc.nist.gov/pcig/STIGs/Wireless-STIG-V3R1.zip.

[7] D. L. Lough, "A Taxonomy of Computer Attacks with Applications to Wireless Networks," Ph.D. dissertation, Blacksburg, Virginia, Virginia Polytechnic Institute and State University, 2001.

[8] IEEE 802.11 Working Group, "IEEE 802.11 Wireless Standard," 1999. Available at http://standards.ieee.org/getieee802/802.11.html.

[9] J.-H. Yeh, J.-C. Chen, and C.-C. Lee, "WLAN standards," *IEEE Potentials*, vol. 22, no. 4, pp. 16-22, 2003.

[10] B. Brown, "802.11: the Security Differences between b and i," *IEEE Potentials*, vol. 22, no. 4, pp. 23-27, 2003.

[11] Bluetooth Special Interest Group, "The Bluetooth Specification, v1.1," 2001. Available at http://www.bluetooth.com/dev/specifications.asp.

[12] C. T. Hager and S. F. Midkiff, "An Analysis of Bluetooth Security Vulnerabilities," in *Proc. IEEE Wireless Communications and Networking Conference*, 2003, pp. 1825-1831.

[13] C. T. Hager and S. F. Midkiff, "Demonstrating Vulnerabilities in Bluetooth Security," in *Proc. IEEE Global Communications Conference*, 2003, pp. 1420-1424.

[14] IEEE 802.15 Working Group, "IEEE 802.15 Wireless LAN Standard," 2002. Available at http://standards.ieee.org/getieee802/802.15.html.

[15] IEEE 802.16 Working Group, "IEEE 802.16 Broadband Wireless MAN Standard," 2001. Available at http://standards.ieee.org/getieee802/802.16.html.

[16] C. E. Perkins, *Mobile IP: Design Principles and Practices*, Boston, MA, Addison-Wesley, 1998.

[17] J. D. Solomon, *Mobile IP: The Internet Unplugged*, Upper Saddle River, NJ, Prentice Hall, 1998.

[18] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequence Distance-Vector Routing (DSDV) for Mobile Computers," in *Proc. ACM SIGCOMM*, 1994, pp. 234-244.

[19] Z. J. Haas, J. Deng, B. Liang, P. Papadimitratos, and S. Sajama, "Wireless Ad Hoc Networks," in *Wiley Encyclopedia of Telecommunications*, J. Proakis, Ed., New York, NY, John Wiley & Sons, 2002.

[20] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11," in *Proc. 7th International Conference on Mobile Computing and Networking*, 2001, pp. 180-188.

[21] A. Stubblefield, J. Ioannidis, and A. Rubin, "Using the Fluhrer, Mantin, and Shamir attack to break WEP," in *Proc. 9th Network and Distributed System Security Symposium*, 2002, pp. 17-22.

[22] P. Papadimitratos and Z. J. Haas, "Secure Routing for Mobile Ad Hoc Networks," in *Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2002, pp. 27-31.

[23] C. E. Perkins and E. M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," in *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90-100.

[24] M. G. Zapata, "Secure Ad Hoc On-Demand Distance Vector (SAODV) Routing," IETF Internet Draft, draft-guerrero-manet-saodv-01.txt, August 2004. Available at ftp://ftp.ietf.org/internet-drafts/draft-guerrero-manet-saodv-01.txt.

[25] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[26] P. Papadimitratos and Z. J. Haas, "Secure Data Transmission in Mobile Ad Hoc Networks," in *Proc. ACM Workshop on Wireless Security*, 2003, pp. 41-50.

[27] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24-30, 1999.

[28] S. Kent and R. Atkinson, "IP Authentication Header," IETF RFC 2402, November 1998.

[29] S. Kent and R. Atkinson, "IP Encapsulating Security Payload," IETF Internet Draft, draft-ietf-ipsec-esp-v3-08.txt, March 2004. Available at ftp://ftp.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v3-08.txt.

[30] ZyXEL Communications Corp., "IPSec FAQ," 2002. Available at http://www.zyxel.com/support/supportnote/zywall10/faq/vpn_faq.htm.

[31] J. G. Steiner, C. Neuman, and J. I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," in *Proc. USENIX Winter Conference*, 1988, pp. 191-202.

[32] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service," IETF RFC 2865, June 2000.

[33] A. J. Menezes, P. C. v. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, 5th ed., Boca Raton, FL, CRC Press, 2001.

[34] B. Schneier, *Applied Cryptography*, 2nd ed., New York, NY, John Wiley & Sons, 1996.

[35] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," IETF RFC 2246, January 1999.

[36] R. Rivest, "A Description of the RC2(r) Encryption Algorithm," IETF RFC 2268, March 1998.

[37] B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," in *Proc. Fast Software Encryption, Cambridge Security Workshop*, 1993, pp. 191-204.

[38] R. Needham and D. Wheeler, "TEA, a Tiny Encryption Algorithm," in *Proc. Fast Software Encryption, Cambridge Security Workshop*, 1994, pp. 363-366.

[39] J. Kelsey, B. Schneier, and D. Wagner, "Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X NewDES, RC2, and TEA," *Lecture Notes in Computer Science*, 1334, 1997, pp. 233-246.

[40] R. Needham and D. Wheeler, "Tea Extensions," Technical Report, Computer Laboratory, University of Cambridge, 1997. Available at http://www.cl.cam.ac.uk/ftp/users/djw3/xtea.ps.

[41] National Institute of Standards and Technology, U.S. Department of Commerce, "Advanced Encryption Standard (AES)," FIPS PUB 197, November 2001. Available at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[42] J. Daemen and V. Rijmen, "AES Submission Document on Rijndael," June 1998. Available at http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael-ammended.pdf.

[43] D. Carrel and D. Harkins, "The Internet Key Exchange," IETF RFC 2409, November 1998.

[44] Cisco Systems, Inc., "About the Internet Key Exchange," IPSec User Guide for the Cisco Secure PIX Firewall Version 5.3, 2004. Available at http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v53/ipsec/ike.htm.

[45] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn, "Point-to-Point Tunneling Protocol," IETF RFC 2637, July 1999.

[46] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter, "Layer Two Tunneling Protocol," IETF RFC 2661, August 1999.

[47] P. J. M. Havinga and G. J. M. Smit, "Energy-Efficient Wireless Networking for Multimedia Applications," *Wiley Wireless Communications and Mobile Computing*, vol. 1, no. 2, pp. 165-184, 2001.

[48] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J.-C. Chen, "A Survey of Energy Efficient Network Protocols for Wireless Networks," *Wireless Networks*, vol. 7, no. 4, pp. 343-358, 2001.

[49] P. Lettieri and M. B. Srivastava, "Advances in Wireless Terminals," *IEEE Personal Communications*, vol. 6, no. 1, pp. 6-19, 1999.

[50] S. Kiaei and S. Devadas, "Which Has Greater Potential Power Impact: High-Level Design and Algorithms or Innovative Low Power Technology," in *Proc.*

*1996 International Symposium on Low Power Electronics and Design*, 1996, pp. 175.

[51] T. Simunic, L. Benini, and G. D. Micheli, "Energy-Efficient Design of Battery-Powered Embedded Systems," in *Proc. International Symposium on Low Power Electronics and Design*, 1999, pp. 212-217.

[52] V. Tiwari, S. Malik, and A. Wolfe, "Power Analysis of Embedded Software: A First Step Towards Software Power Minimization," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 4, no. 2, pp. 437-445, 1994.

[53] A. Shnitko, "Adaptive Security in Complex Information Systems," in *Proc. 7th Korea-Russia International Symposium on Science and Technology*, 2003, pp. 206-210.

[54] S. H. Son, R. Zimmerman, and J. Hansson, "An Adaptable Security Manager for Real-Time Transactions," in *Proc. 12th Euromicro Conference on Real-Time Systems*, 2000, pp. 63-70.

[55] P. A. Schneck and K. Schwan, "Dynamic Authentication for High-Performance Networked Applications," in *Proc. 6th International Workshop on Quality of Service*, 1998, pp. 127-136.

[56] J. Zou, K. Lu, and Z. Jin, "Architecture and Fuzzy Adaptive Security Algorithm in Intelligent Firewall," in *Proc. MILCOM*, 2002, pp. 1145-1149.

[57] R. M. Venkatesan and S. Bhattacharya, "Threat-Adaptive Security Policy," in *Proc. IEEE International Performance, Computing, and Communications Conference*, 1997, pp. 525-531.

[58] T. L. Saaty, *The Analytic Hierarchy Process*, New York, NY, McGraw-Hill, 1980.

[59] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu, "Analyzing and Modeling Encryption Overhead for Sensor Network Nodes," in *Proc. 2nd ACM International Conference on Wireless Sensor Networks and Applications*, 2003, pp. 151-159.

[60] P. Dhawan, "Performance Comparison: Security Design Choices - Building Distributed Applications with .NET," Microsoft Developer Network, October 2002. Available at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/bdadotnetarch15.asp.

[61] L. Bassham, National Institute of Standards and Technology, "Efficiency Testing of ANSI C Implementations of Round1 Candidate Algorithms for the Advanced Encryption Standard," October 1999. Available at http://csrc.nist.gov/encryption/aes/round1/r1-ansic.pdf.

[62] D. S. Wong, H. H. Fuentes, and A. Chan, "The Performance Measurement of Cryptographic Primitives on Palm Devices," in *Proc. 17th Computer Security Applications Conference*, 2001, pp. 92-101.

[63] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha, "Analyzing the Energy Consumption of Security Protocols," in *Proc. International Symposium on Low Power Electronics and Design*, 2003, pp. 30-35.

[64] "SSL 3.0 Specification," November 1996. Available at http://wp.netscape.com/eng/ssl3/.

[65] T. Nakajima, "A Framework for Building Environment-Aware Software," in *Proc. 2nd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, 1999, pp. 237-240.

[66] T. Heath, E. Pinheiro, J. Hom, U. Kremer, and R. Bianchini, "Application Transformations for Energy and Performance-aware Device Management," in *Proc. International Conference on Parallel Architectures and Compilation Techniques*, 2002, pp. 121-130.

[67] A. Daftari, N. Mehta, S. Bakre, and X.-H. Sun, "On Design Framework of Context Aware Embedded Systems," Monterey Workshop, Chicago, Illinois, 2003. Available at http://www.cs.uic.edu/~shatz/SEES/sun.paper.pdf.

[68] F. Chang and V. Karamcheti, "Automatic Configuration and Run-Time Adaptation of Distributed Applications," in *Proc. 9th International Symposium on High-Performance Distributed Computing*, 2000, pp. 11-20.

[69] W. H. Sanders, C. Polychronopoulos, T. Huang, T. Courtney, D. Daly, D. Deavours, and S. Derisavi, "Overview: an Integrated Framework for Performance Engineering and Resource-Aware Compilation," in *Proc. International Parallel and Distributed Processing Symposium*, 2002, pp. 173-180.

[70] Y. Takahashi, "A Mathematical Framework for Solving Dynamic Optimization Problems with Adaptive Networks," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 28, no. 3, pp. 404-416, 1998.

[71] K. Takashio, M. Mori, and H. Tokuda, "m-P@gent: a Framework of Environment-Aware Mobile Applications for Small, Networked Appliances," in *Proc. 4th International Workshop on Networked Appliances*, 2001, pp. 257-266.

[72] G. Welling and B. R. Badrinath, "A Framework for Environment Aware Mobile Applications," in *Proc. 17th International Conference on Distributed Computing Systems*, 1997, pp. 384-391.

[73] S. Corson and J. Macker, "Mobile Ad hoc Networking: Routing Protocol Performance Issues and Evaluation Considerations," IETF RFC 2501, January 1999.

[74] J. R. Vacca, *Wireless Broadband Networks Handbook*, New York, NY, McGraw-Hill, 2001.

[75] C. W. Bostian, S. F. Midkiff, W. M. Kurgan, L. W. Carstensen, D. G. Sweeney, and T. Gallagher, "Broadband Communications for Disaster Response," *Space Communications*, vol. 18, no. 3/4, pp. 166-177, 2002.

[76] V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks," in *Proc. 10th International Conference on Mobile Computing and Networking*, 2004, pp. 287-301.

[77] L. A. DaSilva, S. F. Midkiff, J. S. Park, G. C. Hadjichristofi, N. J. Davis, K. S. Phanse, and T. Lin, "Network Mobility and Protocol Interoperability in Ad Hoc Networks," *IEEE Communications Magazine*, vol. 42, no. 11, pp. 88-96, 2004.

[78] Board on Army Science and Technology, National Research Council, Washington, D.C., "Energy-Efficient Technologies for the Dismounted Soldier," 1997. Available at http://www.nap.edu/.

[79] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Communications*, vol. 8, no. 4, pp. 10-17, 2001.

[80] T. Starner, "The Challenges of Wearable Computing: Part 1," *IEEE Micro*, vol. 21, no. 4, pp. 44-52, 2001.

[81] T. Starner, "The Challenges of Wearable Computing: Part 2," *IEEE Micro*, vol. 21, no. 4, pp. 54-67, 2001.

[82] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Communications of the ACM*, vol. 36, no. 7, pp. 75-84, 1993.

[83] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, 2002.

[84] B. D. Ripley, *Spatial Statistics*, New York, NY, Wiley, 1981.

[85] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, 1994, pp. 85-90.

[86] A. K. Dey and G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness," Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology, 1999. Available at ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf.

[87] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The Anatomy of a Context Aware Application," in *Proc. 5th ACM/IEEE International Conference on Mobile Computing and Networking*, 1999, pp. 59-68.

[88] T. Selker and W. Burleson, "Context-Aware Design and Interaction in Computer Systems," *IBM Systems Journal*, vol. 39, no. 3 & 4, pp. 880-891, 2000.

[89] G. Anescu, "A C++ Implementation of the Rijndael Encryption/Decryption method," The Code Project, 2002. Available at http://www.codeproject.com/cpp/aes.asp.

[90] M. Hahn, "Blowfish.NET 1.01," 2004. Available at http://www.hotpixel.net/bfnet101.zip.

[91] pbrooks, "Tiny Encryption Algorithm (TEA) for the Compact Framework," The Code Project, 2004. Available at http://www.codeproject.com/netcf/teaencryption.asp.

[92] W. Dai, "Crypto++® Library 5.2.1," 2004. Available at http://www.eskimo.com/~weidai/cryptlib.html.

[93] J. D. Meier, S. Vasireddy, A. Babbar, and A. Mackman, "How To: Time Managed Code Using QueryPerformanceCounter and QueryPerformanceFrequency," Microsoft Developer Network, 2004. Available at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/scalenethowto09.asp.

[94] P. J. Leach and R. Salz, "UUIDs and GUIDs," IETF Internet Draft, draft-leach-uuids-guids-01.txt, February 1998. Available at http://www.webdav.org/specs/draft-leach-uuids-guids-01.txt.

[95] G. Schwab, "How To: Get the Device Power Status," Excell Data Corporation and Microsoft Corporation, October 2003. Available at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncfhowto/html/getpowstat.asp.

[96] "Smart Device Framework v1.2," OpenNETCF.org, July 2004. Available at http://www.opennetcf.org/.

[97] D. J. J. Hand and C. C. Taylor, *Multivariate Analysis of Variance and Repeated Measures*, London, U.K., Chapman & Hall, 1987.

[98] M. R. Anderberg, *Cluster Analysis for Applications*, New York, NY, Academic Press, 1973.

[99] J. Hartigan, *Clustering Algorithms*, New York, NY, Wiley, 1975.

[100] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Englewood Cliffs, NJ, Prentice Hall, 1988.

[101] N. Jardine and R. Sibson, *Mathematical Taxonomy*, London, U.K., Wiley, 1971.

[102] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy*, San Francisco, CA, Freeman, 1973.

[103] R. C. Tryon and D. E. Bailey, *Cluster Analysis*, New York, NY, McGraw-Hill, 1973.

[104] S. Sharma, A. R. Nix, and S. Olafsson, "Situation-Aware Wireless Networks," *IEEE Communications Magazine*, vol. 41, no. 7, pp. 44-50, 2003.

[105] P. Lettieri and M. B. Srivastava, "A QoS-Aware, Energy-efficient Wireless Node Architecture," in *Proc. IEEE International Workshop on Mobile Multimedia Communications*, 1999, pp. 252-261.

[106] J. Mitola and G. Q. Maguire, "Cognitive Radio: Making Software Radios More Personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13-18, 1999.

# Appendix A. AHP-Based Decision Engines

In this appendix, the AHP decision engine of CASM is described in detail, focusing on the procedures and mathematics. Note that only six context components and four security algorithms were implemented in the feasibility study described in Chapter 6. The equations and matrices presented here are generalized for any number of security algorithm choices or context components. The core process of the AHP engine is illustrated in Section A.1. The Modified AHP engine also uses the same core process as the AHP engine. The differences between the AHP and Modified AHP engines lie in the matrices used to make the decisions. The values of the options matrices are depicted in Section A.2 for both engines. Discussion of the Deterministic decision engine is not included in this appendix as the operation is straightforward and is described in detail in Section 5.6.2.2.

## *A.1 Core Process*

Given $m$ options and $n$ context components, the first step in the AHP engine is to create the objective matrix. Each context component is associated with a context metric value, and each value is compared to one another to construct the objective matrix. There are a total of $2n$ pairwise comparisons.

Let **c** be a vector of length $n$, where each element represents a single context metric value that is an integer between 1 and 9, inclusive. Let $O$ be an $n \times n$ matrix containing $n^2$ rational number elements $a_{i,j}$, where $i$ and $j$ are integers that range from 1 to $n$.

For all pairs of $i < j$: If $c_i > c_j$, then $k_{i,j} = c_i - c_j$. If $c_i = c_j$, then $k_{i,j} = 1$, otherwise $c_i < c_j$ and $k_{i,j} = \dfrac{1}{c_j - c_i}$. For all $a_{i,j}$ with $i = j$, set $a_{i,j} = 1$; and if $a_{i,j} = k_{i,j}$, then $a_{j,i} = 1/k_{i,j}$. The objective matrix is shown in Equation A.1.

$$O = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ \vdots & a_{3,2} & \ddots & \vdots & \vdots \\ a_{n-1,1} & \vdots & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-1} & a_{n,n} \end{bmatrix} = \begin{bmatrix} 1 & k_{1,2} & \cdots & k_{1,n-1} & k_{1,n} \\ k_{1,2}^{-1} & 1 & k_{2,3} & \cdots & k_{2,n} \\ \vdots & k_{2,3}^{-1} & \ddots & \vdots & \vdots \\ k_{1,n-1}^{-1} & \vdots & \cdots & 1 & k_{n-1,n} \\ k_{1,n}^{-1} & k_{2,n}^{-1} & \cdots & k_{n-1,n}^{-1} & 1 \end{bmatrix} \qquad \text{(A.1)}$$

Each entry in the objective matrix, $O$, is then divided by the sum of the column in which it resides. This normalization process is represented by Equation A.2, creating the matrix $|O|$. The columns are normalized such that each objective pair is less than 1 and all the values in a single column add up to 1. The normalized values are then averaged across each row to create a column objective vector of size $n$, shown in Equation A.3.

$$\forall a_{i,j} \text{ in } O,\ b_{i,j} = a_{i,j} / \sum_{l=1}^{n} a_{l,j} \ni |O| = O / \sum_{l=1}^{n} a_{l,j} \tag{A.2}$$

$$\forall b_{i,j} \text{ in } |O|,\ o_i = \frac{1}{n}\sum_{l=1}^{n} b_{i,l} \ni \mathbf{o} = \begin{bmatrix} o_1 \\ \vdots \\ o_n \end{bmatrix} \tag{A.3}$$

An options matrix is then created for each context component. The options are compared to each other with respect to a particular component, similar to constructing the objective matrix. The weights of each option range from 1 to 9, inclusive. Again each options matrix is normalized (divided by the sums of the columns) and averaged across rows to obtain the relative weights of each options with regards to a single objective.

Each context component is also associated with a weight of importance defined by the user settings. There are *n* context weights, where each value represents a single context weight that is an integer between 1 and 9, inclusive. The weights are compared to each other and a weights matrix is created. The procedure for creating the weights matrix is the same as the one used above to create the objective matrix. The weights matrix is vertically normalized and then averaged across rows to create a weights vector, **w**.

For each options matrix there are *n* column vectors of length *m*. Let **p**(*r*) represent the vertical option vectors, where *r* = 0,…, *n*. Let *P* be a matrix constructed from all the option vectors. The weights vector, **w**, is multiplied by *P* as a scalar to create a weighted matrix *Q*. Then, **d** is the decision vector created from the product of *Q* and **o**.

$$\mathbf{p}(1) = \begin{bmatrix} p_{1,1} \\ \vdots \\ p_{m,1} \end{bmatrix}, \cdots, \mathbf{p}(n) = \begin{bmatrix} p_{1,n} \\ \vdots \\ p_{m,n} \end{bmatrix} \tag{A.4}$$

$$P = \left[\mathbf{p}(1) \quad \cdots \quad \mathbf{p}(n)\right] \Rightarrow P = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{bmatrix} \tag{A.5}$$

$$\begin{aligned} Q = P|\mathbf{w}| &= \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{bmatrix} \begin{vmatrix} w_1 \\ \vdots \\ w_n \end{vmatrix} \\ &= \begin{bmatrix} w_1 \cdot p_{1,1} & \cdots & w_n \cdot p_{1,n} \\ \vdots & \ddots & \vdots \\ w_1 \cdot p_{m,1} & \cdots & w_n \cdot p_{m,n} \end{bmatrix} = \begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ \vdots & \ddots & \vdots \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix} \end{aligned} \tag{A.6}$$

$$\mathbf{d} = Q \cdot \mathbf{o} = \begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ \vdots & \ddots & \vdots \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix} \begin{bmatrix} o_1 \\ \vdots \\ o_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_m \end{bmatrix} \tag{A.7}$$

The procedure for creating the decision vector, **d**, from the option vectors, **p**($r$), objective vector, **o**, and weights vector, **w**, is shown in Equations A.4 to A.6. The element in the decision vector with the largest value corresponds to the most suitable option out of $m$ options. In the case of more than one element equal to the largest value, the option can be randomly selected amongst the largest equal elements.

## *A.2 Options Matrices*

Each options matrix represents the impact a particular context component will have on the decision choices. The options matrices are of $m \times m$ dimensions, where $m$ was defined above as the number of options. The matrices are identical for some context components. In addition, as the context metric values change, so do the values of the pairwise comparisons. The options matrices for the AHP engine are displayed in Section A.2.1 and the options matrices for the Modified AHP engine are listed in Section A.2.2.

### A.2.1 AHP Decision Engine

The options matrices for the Security Level, Location, and User Interactions context components are the same and are shown in Figures A.1 through A.5. The relative importance for each pair-wise comparison is derived from the literature, including [33, 34, 36-38, 40, 41, 60-62].

$$\begin{array}{ccccc} & \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \\ \text{RC2} & 1 & 1/5 & 1/5 & 1/9 \\ \text{BF} & 5 & 1 & 1 & 1/5 \\ \text{XTEA} & 5 & 1 & 1 & 1/5 \\ \text{AES} & 9 & 5 & 5 & 1 \end{array}$$

Figure A.1. Options matrix with context metric values of 1.

$$\begin{array}{ccccc} & \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \\ \text{RC2} & 1 & 1 & 1 & 1/5 \\ \text{BF} & 1 & 1 & 1 & 1/3 \\ \text{XTEA} & 1 & 1 & 1 & 1/3 \\ \text{AES} & 5 & 3 & 3 & 1 \end{array}$$

Figure A.2. Options matrix with context metric values of 2 or 3.

$$\begin{array}{c|cccc} & \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \\ \text{RC2} & 1 & 5 & 5 & 1 \\ \text{BF} & 1/5 & 1 & 1 & 1/5 \\ \text{XTEA} & 1/5 & 1 & 1 & 1/5 \\ \text{AES} & 1 & 5 & 5 & 1 \end{array}$$

Figure A.3. Options matrix with context metric values of 4 or 5.

$$\begin{array}{c|cccc} & \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \\ \text{RC2} & 1 & 3 & 3 & 5 \\ \text{BF} & 1/3 & 1 & 1 & 1 \\ \text{XTEA} & 1/3 & 1 & 1 & 1 \\ \text{AES} & 1/5 & 1 & 1 & 1 \end{array}$$

Figure A.4. Options matrix with context metric values of 6 or 7.

$$\begin{array}{c|cccc} & \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \\ \text{RC2} & 1 & 5 & 5 & 9 \\ \text{BF} & 1/5 & 1 & 1 & 5 \\ \text{XTEA} & 1/5 & 1 & 1 & 5 \\ \text{AES} & 1/9 & 1/5 & 1/5 & 1 \end{array}$$

Figure A.5. Options matrix with context metric values of 8 or 9.

The options matrices for the Energy, Communications, and Object Size context components are the same and are shown in Figures A.6 through A.10.

$$\begin{array}{c|cccc} & \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \\ \text{RC2} & 1 & 5 & 5 & 9 \\ \text{BF} & 1/5 & 1 & 1 & 5 \\ \text{XTEA} & 1/5 & 1 & 1 & 5 \\ \text{AES} & 1/9 & 1/5 & 1/5 & 1 \end{array}$$

Figure A.6. Options matrix with context metric values of 1.

$$\begin{array}{cc} & \begin{array}{cccc} \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \end{array} \\ \begin{array}{c} \text{RC2} \\ \text{BF} \\ \text{XTEA} \\ \text{AES} \end{array} & \begin{bmatrix} 1 & 3 & 3 & 5 \\ 1/3 & 1 & 1 & 1 \\ 1/3 & 1 & 1 & 1 \\ 1/5 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Figure A.7. Options matrix with context metric values of 2 or 3.

$$\begin{array}{cc} & \begin{array}{cccc} \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \end{array} \\ \begin{array}{c} \text{RC2} \\ \text{BF} \\ \text{XTEA} \\ \text{AES} \end{array} & \begin{bmatrix} 1 & 5 & 5 & 1 \\ 1/5 & 1 & 1 & 1/5 \\ 1/5 & 1 & 1 & 1/5 \\ 1 & 5 & 5 & 1 \end{bmatrix} \end{array}$$

Figure A.8. Options matrix with context metric values of 4 or 5.

$$\begin{array}{cc} & \begin{array}{cccc} \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \end{array} \\ \begin{array}{c} \text{RC2} \\ \text{BF} \\ \text{XTEA} \\ \text{AES} \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1/5 \\ 1 & 1 & 1 & 1/3 \\ 1 & 1 & 1 & 1/3 \\ 5 & 3 & 3 & 1 \end{bmatrix} \end{array}$$

Figure A.9. Options matrix with context metric values of 6 or 7.

$$\begin{array}{cc} & \begin{array}{cccc} \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \end{array} \\ \begin{array}{c} \text{RC2} \\ \text{BF} \\ \text{XTEA} \\ \text{AES} \end{array} & \begin{bmatrix} 1 & 1/5 & 1/5 & 1/9 \\ 5 & 1 & 1 & 1/5 \\ 5 & 1 & 1 & 1/5 \\ 9 & 5 & 5 & 1 \end{bmatrix} \end{array}$$

Figure A.10. Options matrix with context metric values of 8 or 9.

### A.2.2 Modified AHP Decision Engine

Some of the options matrices in the Modified AHP engine are based on the block cipher experiments. Other options matrices, specifically for the Security Level, Location, and User Interactions, are exactly the same as the ones used in the AHP engine. The Energy, Communications, and Object Size context components have the different values based on experimental results and are listed in Figures A.11 through A.15.

$$\begin{array}{cc} & \begin{array}{cccc} \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \end{array} \\ \begin{array}{c} \text{RC2} \\ \text{BF} \\ \text{XTEA} \\ \text{AES} \end{array} & \begin{bmatrix} 1 & 5 & 4 & 7 \\ 1/5 & 1 & 1/2 & 3 \\ 1/4 & 2 & 1 & 5 \\ 1/7 & 1/3 & 1/5 & 1 \end{bmatrix} \end{array}$$

Figure A.11. Options matrix with context metric values of 1.

$$\begin{array}{cc} & \begin{array}{cccc} \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \end{array} \\ \begin{array}{c} \text{RC2} \\ \text{BF} \\ \text{XTEA} \\ \text{AES} \end{array} & \begin{bmatrix} 1 & 3 & 2 & 5 \\ 1/3 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 2 \\ 1/5 & 1 & 1/2 & 1 \end{bmatrix} \end{array}$$

Figure A.12. Options matrix with context metric values of 2 or 3.

$$\begin{array}{cc} & \begin{array}{cccc} \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \end{array} \\ \begin{array}{c} \text{RC2} \\ \text{BF} \\ \text{XTEA} \\ \text{AES} \end{array} & \begin{bmatrix} 1 & 6 & 5 & 1 \\ 1/6 & 1 & 1/2 & 1/5 \\ 1/5 & 2 & 1 & 1/4 \\ 1 & 5 & 4 & 1 \end{bmatrix} \end{array}$$

Figure A.13. Options matrix with context metric values of 4 or 5.

$$\begin{array}{cc} & \begin{array}{cccc} \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \end{array} \\ \begin{array}{c} \text{RC2} \\ \text{BF} \\ \text{XTEA} \\ \text{AES} \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1/3 \\ 1 & 1 & 1/3 & 1/5 \\ 1 & 3 & 1 & 1 \\ 3 & 5 & 1 & 1 \end{bmatrix} \end{array}$$

Figure A.14. Options matrix with context metric values of 6 or 7.

$$\begin{array}{cc} & \begin{array}{cccc} \text{RC2} & \text{BF} & \text{XTEA} & \text{AES} \end{array} \\ \begin{array}{c} \text{RC2} \\ \text{BF} \\ \text{XTEA} \\ \text{AES} \end{array} & \begin{bmatrix} 1 & 2 & 1/3 & 1/5 \\ 1/2 & 1 & 1/5 & 1/7 \\ 3 & 5 & 1 & 1/2 \\ 5 & 7 & 2 & 1 \end{bmatrix} \end{array}$$

Figure A.15. Options matrix with context metric values of 8 or 9.

# Vita

Creighton Tsuan-Ren Hager

Creighton Hager received his M.S. degree in Electrical Engineering from Virginia Tech in December 1999 and his B.S. in Electrical Engineering, also from Virginia Tech, in May 1998. He received scholarships from the Virginia Tech Alumni Association, Marshall Hahn, NHA, and AFCEA. His professional memberships include IEEE and Eta Kappa Nu.

Creighton has also consulted overseas in Manila, Philippines, improving a private high school, Samantabhadra Institute, into a technical university. He developed the technical university by improving the classrooms, installing computers, developing curriculums, hiring and training instructors, teaching the first classes, and managing the school.

## Publications

C. T. Hager and S. F. Midkiff, "An Analysis of Bluetooth Security Vulnerabilities," *Proceedings IEEE Wireless Communications and Networking Conference (WCNC)*, 2003, pp. 1825-1831.

C. T. Hager and S. F. Midkiff, "Demonstrating Vulnerabilities in Bluetooth Security," *Proceedings IEEE Global Telecommunications Conference (Globecom)*, 2003, pp. 1420-1424.

M. S. Thompson,W. O. Plymale, C. T. Hager, K. Henderson, S. F. Midkiff, L. A. DaSilva, N. J. Davis, and J. S. Park "Ad Hoc Networking Support For Pervasive Collaboration" (poster presentation and abstract), *Adjunct Proceedings of the Sixth International Conference on Ubiquitous Computing (UbiComp)*, 2004.

C. T. Hager, S. F. Midkiff, J.-M. Park, and T. L. Martin, "Performance and Energy Efficiency of Block Ciphers in Personal Digital Assistants," *Proceedings IEEE International Conference on Pervasive Computing and Communications (Percom)*, 2004. (in press)