Tradeoffs Between Energy and Security in Wireless Networks

Fernando C. Colón Osorio, Emmanuel Agu, and Kerry McKay

WPI System Security Research Laboratory (*WSSRL*),
Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA

email: fcco@cs.wpi.edu, emmanuel@cs.wpi.edu, kerrym@wpi.edu

Abstract

In recent years, several researchers have studied the vulnerabilities present in the encryption protocols and authentication mechanisms associated with 802.11-based networks. This research has led to the creation of protocol extensions and replacement proposals such as WPA, 802.11i, and 802.1X. In addition, Denial-of-Service attacks that can be launched against 802.11-based networks, with relative ease and impunity, have been studied. Simultaneously, researchers studying the limitations of wireless networks have turned their attention to one of the inherent limitation of wireless devices, namely, power consumption. Research in this area has been focused in understanding the impact of the network interface card, and its effect on the overall power consumption. The main research result has been the design and implementation of adaptive power management algorithms that complement the power saving modes of 802.11 devices. Unfortunately, study of wireless networks protocols from the perspective of their security profile, that is, how do the power consumption limitations of wireless devices affect security, is less well understood.

In this manuscript, we will first review the current limitations of security protocols associated with 802.11 networks. We will develop a general model that will help us understand how the current set of security related protocols affect the energy consumption of the devices. This model is general enough to cover the security energy tradeoffs at different layers of wireless network protocols in use. In the model, we use a decision-theoretic framework. This framework requires both an energy cost function, called, C^E and a security-reliability measure, R_M. The energy cost function, C^E, is the cost, both in energy and other system resources, of applying a countermeasure M_k against a specific protocol vulnerability V_i. The security-reliability measure, R_M, represents the level or measure of the security-reliability attained by countermeasure M_k on the overall security of the system. Having defined such a framework, we present our initial analysis of popular security protocol, such as WEP, TKIP. Preliminary results showed that significant improvements can be obtained by constraining the time frame where security needs to be guaranteed.

Based on these results, a new wireless encryption protocol, calledε-sec, or Energy efficient secure protocol is introduced. This protocol has the potential to minimize power consumption while maximizing the security profile of the network as a whole.

Key words: Vulnerabilities, Wireless Networks, 802.11 Networks, Denial Of Service, Operational Security, Energy Efficient Cryptographic Algorithms.

1 Introduction

In recent years, several researchers have studied the vulnerabilities present in the encryption protocols and authentication mechanisms associated with 802.11-based networks. This research has led to the creation of protocol extensions and replacement proposals such as WPA, 802.11i, and 802.1X. Security attacks on wireless networks are harder to prevent than attacks on wired networks for several reasons. Wireless signals leak beyond the confines of buildings in which wireless LANs are installed, the mobility of users on a wireless network makes perpetrators of security attacks difficult to track down and the cooperative nature of most ad hoc networking protocols makes it easy to perpetuate man-in-the-middle types of attacks.

A key limitation of mobile devices is their limited battery power. The effect of the power consumption of wireless devices on their performance has received renewed interest in the last few years. Research in this area has focused primarily on measuring and understanding energy utilization on the network interface card, its impact on the overall power consumption of the mobile systems, and power management techniques at various layers of the protocol stack. The principal results from such investigations has been the design and implementation of adaptive power management algorithms that complement the power saving modes of 802.11 devices. The majority of these investigations consider only energy utilization in the absence of malicious users. Unfortunately, within this context, the study of wireless networks protocols power consumption from the perspective of their security profile, and more specifically how the power consumption limitations of wireless devices affect their security is less well understood. Entire classes of security attacks which involve draining the batteries of mobile devices are now possible.

In this manuscript, we will first review the current limitations of security and network protocols associated with 802.11 devices. We will next use a model proposed in [CO04] to understand protocols elements affect the energy consumption of the device. More specifically, we attempt to quantify how much additional power is expended by a mobile device in order to achieve a given security profile. The model will be used to evaluate current and proposed wireless security protocols such as WEP, WPA, 802.1x/EAP, Counter CBC-MAC Protocol (currently under review by the IEEE as the next wireless security protocol), and $\varepsilon - sec$ (a new wireless encryption protocol capable of minimizing power consumption while maximizing the security profile). These analytical evaluations will serve as the basis for future comparisons against actual empirical measurements.

2 Previous Work

A careful review the wireless security literature shows that four general areas of wireless security research have emerged in the last few years. These are:

(1) Security of the Wireless Channel;
(2) Denial of Service Attacks on Wireless Network Protocols;
(3) Trust and Trust Extensions to the Wireless Secure Infrastructure; and
(4) Identification and demonstration of specific attacks on wireless network protocols

While all of these are important, in this manuscript, we are primarily concerned with the first item, that is, the security of the wireless channel.

2.1 Security of the Wireless Channel

The weaknesses of the current 802.11 security standard (WEP), WEP2, and other protocol extensions has been explored recently [AR01]. Scott Fluhrer, et. al. explored the weakness of the underlying encryption algorithm used by WEP, RC4 [FL01], Fluhrer showed that in a common mode of operation used by WEP, RC4 is completely insecure. Further work by Nikita Bosrisov, Ian Goldberg, and David Wagner [BO01] identified several WEP protocol flaws including its vulnerability to dictionary based attacks (so called Decryption Dictionary flaw), and the problems associated with key management and message authentication. In their paper, several practical attacks were constructed, and their work showed that WEP does not achieve its security goals.

In order to deal with these limitations, a set of extensions have been proposed that attempt to ameliorate 802.11 security weakness by:

(1) using greater keys lengths
(2) decomposing the problem into three phases: authentication, authorization, and access control ; and
(3) modifying key distribution and management methods to use a trusted certificate authority.

One key limitation of this approach is that it ignores the financial cost associated with their implementation (such as the cost of a trusted certificate), as well as the practical limitations of wireless devices such as their limited battery life. In effect wireless networks are significantly different that their wired counterpart in this area. Specifically, mobile nodes and wireless networks have a lower amount of memory, battery power, and bandwidth. This means that attacks on system resources will affect wireless devices quicker and have more pronounced effects than their wired counterparts.

Furthermore, by separating authentication, authorization, and access control, the proposed protocols increase the overhead required per packet of data transfered. This, of course, leads to greater utilization of scarce resources. As we point out in section 5, an approach to get around this limitation is to investigate security from the perspective of effective resource utilization. For example, if we apply the Principle of Adequate Protection, i.e., *Computer items/data must be protected only until they loose their value* , then, we can construe different scenarios where limited extensions of WEP are indeed optimal. Optimality, in this context, means that the confidentiality, integrity, and availability of the system can be guaranteed for a specific period of time $[t_0, t_0 + \Delta]$ while minimizing energy consumption, or some other resource.

3 Summary of Current, and Proposed, Wireless Security Protocol's Limitations

In this section, we present a detailed summary of the current limitations of the proposed wireless protocols from a security perspective.

3.1 WEP

The Wired Equivalent Privacy (WEP) protocol was created as a way to ensure the same level of privacy for wireless communication as there is for wired communications. Its goals, as with any security mechanism, is to provide confidentiality, integrity, and availability to the wireless network. Unfortunately, WEP accomplishes none of these goals. It is a very poor protocol and was nearly removed from the 802.11 standard in a vote by the IEEE in June 2001 (54%-46%)[NA02].

3.1.1 WEP Encryption

The encryption scheme used in WEP is a very simple one: it uses the RC4 stream cipher to generate a pseudo-random keystreams which it XORs with the plaintext to encrypt. To decrypt, XOR the keystreams with the ciphertext. In WEP, the RC4 key is the concatenation of a 24-bit initialization vector (IV) and the shared secret key common to the access point and all its users.

$$keystream = RC4(IV + key)$$

$$C = P \oplus keystream$$

$$P = C \oplus keystream$$

RC4 is a keyed stream cipher containing two different functions - the key scheduling algorithm (KSA) and the pseudo-random generator algorithm (PRGA)[FL01]. The same RC4 key will always produce the same keystream. Since the only varying piece of this is the IV, that means that there will only be 2^{24} different keystreams generated per shared secret key. This small space causes keystreams to repeat, which is in violation of a key concept in the security of stream ciphers - the same keystream should never be used twice.

To help alleviate this problem, it was proposed that the IV space be increased to 128 bits. Unfortunately, this does not solve the problem, since IV's are still reused. In fact, it was never enforced that more than one IV had to be used in the first place. Vendors could set their devices to only go between 0 and 2^{24}, and the WEP protocol has no way of preventing or detecting this.

The RC4 cipher itself is insecure. The key scheduling algorithm has been shown to leak information about the key, one byte at a time. By collecting about 60 messages of a special form, an attacker can guess the secret with a high probability of being correct [FL01].

3.1.2 Integrity Check

The WEP integrity check is also weak. WEP uses a cyclic redundancy check like the one used to detect *random* errors in networking. The distinction between random and intentional changes is very important. The output space of this integrity check value (ICV) is only 32 bits, which is poor for collision resistance. Because it is unkeyed and linear, zero knowledge of the shared secret is needed in order to compute it. For instance, an attacker could easily change or spoof a packet and it would go undetected because the attacker would make sure that the ICV was appropriate for the hacked packet.

3.1.3 Authentication

WEP uses a simple challenge/response protocol that is also quite poor. The challenge exchange goes as follows[AR01]:

$$AP \rightarrow client : challenge$$
$$client \rightarrow AP : IV, \{challenge, ICV\}_{wepKey}$$

This is completely unacceptable as an authentication scheme. By capturing the clear challenge, the encrypted challenge, and the IV, an unauthorized user can gain access by making a simple calculation.

$$keystream = C \oplus P$$

With this (keystream, IV) pair, an attacker can gain access to the network without knowing the shared secret.

3.2 WPA

The main reasons for WEP remaining in the 802.11 standard is its wide deployment and implementation in hardware. WI-FI Protected Access (WPA) is a set of improvements over WEP that are compatible with existing 802.11 devices.

3.2.1 The Temporal Key Integrity Protocol

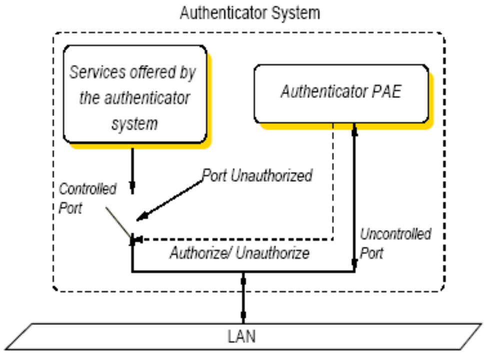
The Temporal Key Integrity Protocol (TKIP) is a modified version of WEP's encryption scheme. Like WEP, it uses the RC4 stream cipher to generate a keystream which is then XORed with the plaintext. TKIP's major contribution is a way of ensuring that keystreams are unique to each packet. This is done by mixing the transmitter address (TA) into the key, giving each user a unique shared key per session, and by using the IV as a counter. If an IV value is received out of sequence, then it is discarded. When the IV space is almost exhausted, a new key is negotiated.

3.2.2 Michael

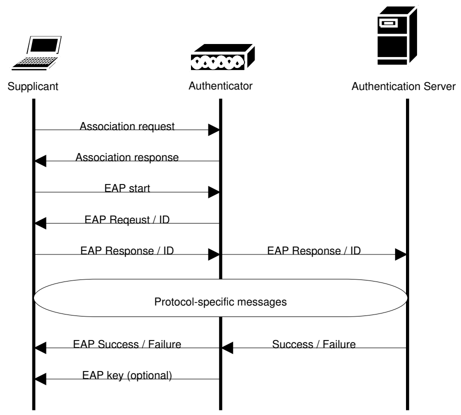
The TKIP specification also names a new message integrity code (MIC) called Michael. Michael is a nonlinear hash function that produces a 64-bit output. Unlike the CRC used in WEP, Michael is keyed. Only those who know the secret can compute a valid hash. However, it should be noted that the output space is still small, allowing the possibility of finding or guessing a valid hash in a feasible amount of time.

3.2.3 802.1x/EAP Authentication

802.1x is a flexible framework which has been created for authentication in PPP (point-to-point protocol). This framework can also be applied to a wireless network to allow a key distribution for TKIP while still using existing hardware. 802.1x defines the concept of port-based access control. This is achieved by having two types ports: a controlled port and an uncontrolled port. Access to the uncontrolled port can be gained at any time, as this port leads to the authentication service. The controlled port can only be accessed after authentication and authorization have taken place, as denoted by the switch in figure 1(a). In wireless networks, the controlled port is the AP's connection to the network, and the uncontrolled port goes to an authentication server, such as RADIUS (remote authentication dial-in user service).



(a) Controlled and uncontrolled ports in the authenticator



(b) Basic EAP messages

Fig. 1. EAP/802.1x Authentication

There are three parties identified in this authentication scheme. The supplicant is the entity that wishes to be authenticated (wireless client). The authenticator is the entity with which the supplicant is trying to authenticate (access point). Authentication is provided by the third party, the authentication server, through communication with the authenticator. The supplicant and authenticator send messages over the wireless medium, while the authenticator and authentication server communicate over a wire (or may even reside in the same machine). The separation of services here is interesting because it is something that was borrowed from the wired world. It is also interesting to note that a wire has actually been introduced into the wireless authentication process.

The extensible authentication protocol (EAP) is an outline for authentication that sits beneath a higher protocol (figure 1(b)). For instance, SSL could be used on top of EAP. Protocols which are currently available from vendors deploying TKIP and EAP (Cisco Systems, for example) include protocols such as EAP-TLS (transport layer security), LEAP (Cisco's lightweight EAP), EAP-FAST (Flexible Authentication via Secure Tunneling), EAP-TTLS (tunneled transport layer security), and PEAP (protected EAP). Each variant has its own methods, such as mutual authentication vs. client-only authentication, and certificates vs. username/password.

3.3 The Next Standard

The IEEE views TKIP as a temporary solution and is currently developing a new standard. Presently, Counter CBC-MAC Protocol (CCMP) is the frontrunner. This new protocol is based around a trusted block encryption algorithm in CCM mode.

3.3.1 AES

The advanced encryption standard (AES) has been selected as the block cipher for CCMP. This algorithm, called Rijndael, has been widely accepted and selected to replace DES as *the* encryption standard. AES takes block and key sizes of 128, 192, and 256 bits. Different block size/key size combinations change the key schedule and number of rounds. AES can be optimized for energy and time efficiency by pre computing the matrix multiplications and storing them in look-up tables (LUTs). Of course, this does sacrifice some storage space. In order to do this, 2560 bytes of storage are needed. Hardware vendors can also optimize AES in hardware with specialized chips.

3.3.2 Counter CBC-MAC Mode

CCM is a new mode that performs confidentiality and integrity by combining counter mode (confidentiality) and a cipher block chaining message authentication code, or CBC-MAC (integrity). In the documentation, they say that CCM provides authentication, not integrity, but what they mean is that there is a MAC which determines if the data has been tampered with. This is synonymous with a message integrity code. To keep the terminology consistent, we shall say integrity. CCM was submitted to NIST (National Institute of Standards and Technology) in 2002. It is still undergoing revisions, with the last public draft published in September 2003. The activity that we have seen regarding this protocol displays some unease with the CBC-MAC. There is a weakness in CBC mode that allows blocks to be swapped without altering the resulting MAC. CBC is still widely used despite this flaw; however, some still do not approve of it supplying the integrity check.

3.3.3 Authentication

CCMP authentication and key management will use the 802.1x framework. There will most likely be a set of recommended EAP types, which may include an implementation of EAP-Kerberos (which is currently not formally defined).

4 Power Limitations of Mobile Devices and their Impact on Security

The primary sources of power consumption on an 802.11 network device are: the duration of radio transmission while sending packets, the power level at which the radio transmits packets, the amount of power consumed by the radio while it is idling and waiting to receive packets, and the amount of power spent receiving packets addressed to it. In addition, protocol efficiencies affect power utilization. This is to say the information-theoretic measure of each packet is the ratio of information content versus the total number of bits, or packets, transmitted. This power consumption affects the utility of wireless networks, especially when ad-hoc networks in a battlefield experience are considered. Techniques that have been implemented in the past to limit the duration of transmission have made use of both compression and aggregation. However, this addresses only one of the four factors limiting the utility of wireless networks due to energy consumption. A second area of research interest is that of improving protocol efficiencies. The basic efficiency metric used to evaluate such networks has been *the mean rate of a word of data successfully arriving at its destination per the power used to support the network, i.e., the average number of bytes successfully transmitted per Joule of energy consumed*[GA03]. In order to optimize this metric of energy efficiency, researchers have studied two key techniques. namely,

- The use of power modulation algorithms at the network card to improve the energy utilization at the transmitter; and
- The design and implementation of energy efficient and topological-aware protocols.

In both of these approaches, changes to the protocol stack at the data link MAC layer and the routing layers have been proposed. In considering power modulation techniques, the primary approach has been to design power-saving strategies that make use of the sleep mode present in 802.11 devices. While variations of sleep mode modulation have been somewhat effective, adaptive strategies that attempt to dynamically trade off power versus network activity[**?**] while providing a guarantee on the maximum RTT of a connection show the most promise. A different approach to power saving modulation is presented in [SH03]. In their work, instead of trying to adapt the sleep duration of the NIC, the authors attempted to eliminate the power consumed during sleep mode altogether by incorporating a separate low-power consumption channel for control. The basic idea is to shutdown both the device and the network card, while keeping the control channel/device alive. In their work, they showed that for an iPAQ PDA the battery lifetime with a low-power control channel approaches the lifetime of an iPAQ without any wireless LAN card.

Research on energy efficient network layer protocols is not new. Earlier explorations, see Raghavendra and Singh, [RA98], proposed protocols where by intelligently powering down nodes that are not actively transmitting, energy can be preserved. They showed, using simulation, that for an Ad Hoc network of 10-20 nodes power savings of up to 60 % could be attain if a special purpose protocol called PAMAS was used. Recently, Xu et.al. developed a similar protocol that is topologically aware in Ad-Hoc networks. In their protocol, redundant nodes are identified using their physical location and an estimate of their radio range, and then they are selectively turn off.

While the approaches investigated thus far are useful in reducing the power and resource consumption of wireless devices, the additional power and resource utilization drain that security and security protocols imposed as well as the energy drainage profiles of successful attacks, are less understood. For example, if known security techniques from the "Wired-World", such as Authentication and Ticketing servers (e.g., Kerberos IV, V) are used, then, power utilization of the device will necessarily go up. Upon such a consideration, it becomes clear that there exist a tradeoff between security, as measured by some metric, S, which captures the security or protection provided by protocol and the incremental energy consumption required to provide such protection.

Although several recent studies have proposed energy efficient protocols, [HO02], [JA01], [WO01], [LA02], with one the notable exception [PO03]there has not been a comprehensive energy analysis of security protocols across multiple levels of the protocol stack. We observed that [PO03] for the first time attempts to remedy this situation, and more specifically, they studied the energy consumption requirements of the most popular transport-layer security protocol SSL (Secure Sockets Layer). In addition, Potlapally, et.al., considers a parametric approach to energy utilization. The one missing element of the works cited is an attempt to provide an analytic model across multiple protocols layers that can effectively explained the energy wastage imposed and measured.

5 Energy-Security Tradeoff Model

From the previous literature survey, it is clear that battery power is one of the most precious resources to a mobile client. Thus, it is important to understand the relevant energy and battery trade-offs involved in any protocol attack or its associated countermeasure. More specifically, each class of protocol attack leads to potential loss in efficient battery use. Similarly, any proposed countermeasure can provide a given level of security-reliability but will also requires an additional expenditure in energy by mobile nodes. At this point, we will refer to the security-reliability goal simply as security. In effect the classical definition of security encompasses the concepts of reliability pertinent to our discussion, namely, security is the protection of assets from harm, or:

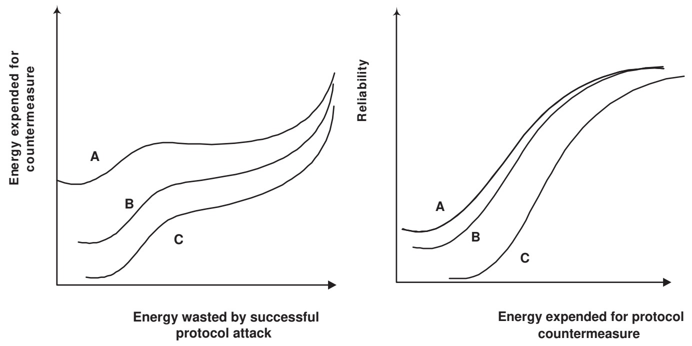
- **Confidentiality**: assets are used/access only by authorized parties
- **Integrity**: assets can be modified only by authorized parties and only in authorize ways
- **Availability**: assets are available to authorized parties when requested.

We claim that there is a direct relationship between a given attack countermeasure and the level of security-reliability it can provide, and also a relationship between the energy spent in carrying out a countermeasure and the energy level that is potentially lost if a given attack is successful. This three-dimensional security-reliability tradeoff is the basis on which we propose a security-energy model for protocol vulnerabilities. Figure 2(a) depicts a hypothetical relationship between a given protocol vulnerability and the potentially wasted energy if the attack is successful. Figure 2(b) is our hypothetical model which illustrates the amount of energy expended in a given attack countermeasure in order to guarantee the desired level of security.

In the figures three different countermeasures are shown. These countermeasures may be at different layers of the protocol stack. However, by using our hypothetical model, the effectiveness of somewhat dissimilar attacks and their countermeasures can be compared, even across multiple protocol layers. Therein lies the power of our model. It becomes possible to decide areas of maximal yield in terms of energy expenditure and what level of the security is achieved. A given target application may decide what energy levels its nodes must expend in order to guarantee a certain level of security. For instance, a highly sensitive military application may choose a high level of security which requires countermeasures that kill mobile device batteries in half an hour while casual email between friends may choose a lower level of security which allows longer battery life. While security levels are harder to quantify, except in the most simple cases such as the information theoretical measure of Equivocation in the case of cryptographic algorithm [SH46], energy expended or wasted can be more easily quantified. Finally, as a practical note, rigorous experimentation and measurement is required to parameterize currently proposed attacks and countermeasures and fit them to our model.

In addition, this model, from a security perspective, maybe considered naive, in the sense that all vulnerabilities are considered to have the same effect on the security profile of the protocol. Clearly this is not the case. For example, a vulnerability on the cryptographic algorithm that leads to "masquerading attacks", such as in the case of the "The Denning-Sacco" disaster, see [AN95] will have a significantly reduced impact than those associated with the key regeneration weaknesses of WEP, see [AR01], [FL01], [BO01] which affects every message exchange in the network. Theoretically, we could remedy this limitation of the model by associating an effect/impact measure $I(V_i)$ that quantifies the effect on the security profile of the protocol when vulnerability V_i is exploited by an attacker. The exact energy profile of V_i is dependent on the specific attack and needs to be evaluated per vulnerability. However, at this point, this extension will unnecessarily complicate our analysis. Here we will assume that each exploit of different vulnerability classes V_i have the same equal effect on the security of the protocol at hand.

To formalize our model, we use a decision-theoretic framework similar to that in [HO97]. First, we define an energy cost function, C^E, of applying a countermeasure M_k against a protocol vulnerability V_i as $C^E(M_k, V_i)$. For simplicity, we lump the costs of applying the countermeasure with the overhead of successful recovery from an attack. These can be separated but will not affect our results. As a practical note, these countermeasures may vary across protocol layers such as including more FEC bits in transmitted packets at the link layer, or a high maximum retransmission threshold in 802.11 MAC protocols. The total energy consumed by all countermeasures are then given as



(a) Trade-off between protocol vulnerability and potential loss in energy

(b) Energy expenditure required for a given security level

Fig. 2.

$$C^E = \sum_i C^E(M_k, V_i) \tag{1}$$

Combinations of countermeasures may not be additive as suggested by equation 1 since some countermeasures may perform multiple functions and countermeasures may be correlated or interdependent. We now introduce a variable, A, which takes into account a specific attack on a vulnerability V_i. The energy consumed given in Equation 1 changes to $C^E(M_k, V_i, A).p(A_i^V|E)$ is the probability that the attack A on vulnerability V_i has occurred given some evidence, E. This evidence in practice could be incorrect checksums or protocol timeouts. Thus the expected energy consumption for all countermeasures is:

$$C^E = \sum_i p(A^{V_i}|E) C^E(M_k, V_i) \tag{2}$$

The above model is for single attacks on specific vulnerabilities. However, in real life, entire classes of attacks are possible on a given vulnerability. Thus, these classes of attacks are somewhat correlated and the model should reflect this. So, we define a group of attacks S_j which are possible on a given protocol vulnerability such that

$$C^E = \sum_i \sum_j p(A^{V_{ij}} | A^{S_j}, E) p(A^{S_j} | E) C^E(M_k, V_i) \quad (3)$$

Finally we define R_M, a measure of the security-reliability of the system by implementing a set of countermeasures. Further the *Countermeasure Energy Quotient (CEQ)*, Q_M, as the ratio of the security-reliability from a set of countermeasures divided by the energy required to implement them. Hence,

$$Q_M = \frac{R_M}{C^E} \quad (4)$$

Equation 4 is our security-energy model. We seek to find a set of countermeasures which yield the highest values of Q_M. The choice of sets of countermeasures is indeed a complex operation requiring extensive experimentation and measurements.

5.1 Security-Energy Model - An Instance

Embedded in the Security-Energy model represented by Equations 2, 3, and 4 is the general concept of real time adaptive protocols. That is, faced with an attack on a specific vulnerability, $V_(i)$, the protocols described by the said equations are capable of detecting the attack in real time, isolating the source of the attack, and launching a set of countermeasures whose energy costs are given by $C^E(M_k, V_i, A)$. We know of no such protocols in existence today. In general, most protocols in used are static in nature. That is, in protocols such as WEP and TKIP the energy expenditure to counteract a given vulnerability attack is constant, or $C^E(M_k, V_i)$. This energy expenditure is fixed upon the definition of the protocol itself, and it is configured based on a set of parameters, such as key length, upon initialization. In order to make our model concrete, we will now turn our attention to one such instance and apply the model above to it.

5.2 Static Protocols - From an Energy consumption sense

Consider a simple protocol such as WEP or TKIP. These wireless protocols were designed to protect the system from three classical vulnerabilities, $V_1, V_2, and V_3$, where

- V_1 = Confidentiality or robustness of the cryptographic algorithm;
- V_2 = Robustness of the authentication protocol; and
- V_3 = Robustness of the authorization and access protocol.

Further, the energy expenditure function associated with each countermeasures $M_1, M_2, and M_3$, $C^E(M_k, V_i)$ is defined by the protocol itself and the parameters used. For example, in WEP, the countermeasure against V_1 is simply the RC4 cryptographic algorithm. In this case, the energy expenditure to achieve the desire level of security is simply $C^E(K_{length}, V_i) = f(\#computations in RC4)$. In this example, C^E can be easily calculated by multiplying the Number of computations required by RC4 given a key of length K_{length} times the energy consumed in joules by a single computation.

In addition to the security-energy tradeoffs expressed by Equations 2, 3, and 4, it is often useful to represent the energy consumed to achieve a level of security as an overhead measure on the total energy consumed to achieve a particular protocol task. To accomplish this, we borrow some of the concepts first introduced by [ST97]. Simply stated, we break down the total energy consumed to complete a single bulk file transfer of b bytes as follows.

$$\begin{aligned} Energy_{Total} &= E_{SendRecv} + E_{idle} \\ &\quad + (C^E = \sum_i C^E(M_k, V_i)) \end{aligned} \quad (5)$$

where,

$$\begin{aligned} C^E &= \sum_i C^E(M_k, V_i) = \alpha_1 E_{cryp} \\ &\quad + \alpha_2 E_{SendRcvd_{ap}} + \alpha_3 E_{SendRcvd_{tgs}} \end{aligned} \quad (6)$$

and,

$$Energy_{cryp} = e_i \times C_{cry} \quad (7)$$

Here, the energy consumed by a device includes the energy consumed to complete a bulk transfer absent of security protocol overhead, SendRcvd (steady state or intrinsic energy consumed), the energy consumed by the device while in the Idle state, Idle, and the overhead energy consumed by encryption algorithm and cryptographic protocols, namely the energy consumed per encryption/decryption pair on messages, $Energy_{cryp}$, the energy consumed by all authentication message exchange, $E_{SendRcvd_{ap}}$, and the energy consumed by the ticketing granting services, $E_{SendRcvd_{tgs}}$. The overhead energy associated with encryption is reflected by the term $e_i \times C_{cry}$, where e_i is the fixed energy consumed per constant encryption (using encryption algorithm i), and C_{cryp} refers to the number of encryptions required by the protocol, exclusive of the SendRcvd encryptions. Our goal here, is to understand how the different elements in the energy equation change as additional features are included to enhance the security of the protocol. In our analysis, we consider WEP as the base case and denote its energy consumption as $E_{SendRcvd_{WEP}}$. For the purpose of our work, both the SendRcvd and Idle energy consumption are constant on a per single bulk transferred, and energy Equation 6 can be simplified, as shown below in Equation 8.

$$\begin{aligned} Energy_{Total} = K_0 + \alpha_1 E_{cryp} + \alpha_2 E_{SendRcvd_{ap}} \\ + \alpha_3 E_{SendRcvd_{tgs}} \end{aligned} \tag{8}$$

6 Major Contributions

The work proposed here formalizes the concept of operational security as a function of energy consumption by a wireless device in a wireless network. Operational security within the larger context is similar to the concept of "practical secrecy", first introduced by Shanon in 1946, [SH46]. This concept is rather simple. That is, given a bounded time period $[\ t_0\ ,\ t_0 + \delta\ ]$, the system under consideration is operationally secure, iff, it can guarantee the confidentiality, integrity, and availability of the system and its resources with a probability, P_s, where, $P_s = 1 - P\{"BreakingtheSystem"\} = 1 - \epsilon$. Or conversely, if $P\{"BreakingTheSystem"\} = \epsilon$, where $\epsilon \rightarrow 0$.

Consider the following example. In the design of a secure communication channel using cryptographic algorithms, "Breaking The System" corresponds to "Breaking the code". In this context, Shannon's definition of operational secrecy corresponds to "operational security", and he demonstrated that operational security approximates "perfect security" when the cryptographic algorithm generates a sequence of statistically independent keys per time period $[t_1, t_2), [t_2, t_3), \cdots, [t_{n-1}, t_n)$. Here, it becomes relatively simple to define a measure of how secure the system is, and subsequently, evaluate design tradeoffs between the different cryptographic algorithms, and the energy consumed, as we have shown in section 5. The problem of defining such tradeoffs across multiple layers of protocols is significantly more difficult. The difficulty lays on the definition of what does "operational security" mean?, and how to model, analyze, and quantify it. For example, if "the system" under consideration provides a set of services such as authentication, key distribution, and access to a set of distributed resources, then, "Breaking The System" will corresponds, at the very least, to "Breaking the Cryptographic Protocol". Hence, in order to apply the model described in section 5, one needs to answer the question of how secure is the cryptographic protocol? A good example that illustrates how difficult it is to answer such a question is the "Denning-Sacco" disaster. In "Denning-Sacco", a protocol deemed secure was found to be fragile twelve years after it was first introduced, see [AB94].

Given such challenges, our approach here is to first understand the model in terms of the energy utilization. Specifically, we will investigate the energy consumption and wastage as it relates to security features. Two distinct and complementary approaches will be taken. In the first approach, we will study current and proposed extensions to security protocols for wireless networks and evaluate the energy consumption associated with different services and attributes that the protocol provides using our energy-security model described in section 5, and Equations 5, 6, 7, and 8. We will call this, intrinsic energy evaluations. However, in order for our analysis to be useful, we and in accordance with the *Countermeasure Energy Quotient (CEQ)*, Q_M, of Equation 4, we will need a methodology for computing the security profile of a given wireless security protocol. Here, and a first approximation in our work, we will use the concept of "perceived security". Perceived security would based on the following criteria:

- equivocation measure of the encryption algorithm used;
- known weaknesses
- effort required to break protocol
- key usage (lifetime, keystream reuse, etc.)

Secondly, we will explore, model, analyze, and empirically quantify the impact that well know attacks across multiple protocol layers have on the battery life of a wireless network device. The hope here is that by studying such an impact better protocols, which will be potentially adaptive, can be developed. This work will be presented in a separate manuscript later this year.

6.1 Intrinsic Energy Model - 1^{st} Results

6.2 Methodology

In section 5, we introduced the Security-Energy model first presented by Colon Osorio et.al. in [CO04]. In order to effectively use such model, we would like to apply the closed-form analytic solutions presented in Equations 1,2, 3, 4, and further simplified as in Equations 6, and 7 to a set of current and proposed wireless security protocols such as WEP, TKIP, TKIP enhanced by CISCO proprietary authentication protocol LEAP, and others. As a first step, and in accordance to Equations 6, 7, we need to understand the energy consumed on a per block transfer for each one of the protocols under consideration. Here, we break down each protocol under consideration in terms of the primitive operations required to accomplished a single transfer. This was accomplished by reviewing the Standards in question: [FI01], drafts: [CC03], [CC02], RFCs: [JK93], [RF99], [**?**], papers: [SC98], [SC99][CI02][**?**][**?**] and textbooks[KA02][JE03]. Available pseudo-code and explanations from these sources were used to create tables recording the number of occurrences of operations used by each protocol.

However, and as it is well known, data dependencies greatly affect the number of operations used to accomplish a block transfer. For this reason "real world" parameters were needed in order to establish a bound on the number of computations. One such case, where real data was required, is EAP-TLS. In this particular case, we used the firefox web browser with TLS enabled and SSL disabled while a secure connection to amazon.com was established. This transaction was captured with the Ethereal network protocol analyzer. The length of each message was then used to compute the number of operations of the corresponding TLS message during EAP-TLS authentication phase.

Using the information provided by these tables, and the energy consumed on a $\frac{joules}{computation}$, we can readily compute the total energy overhead per block of information transferred, E_{total}, as given in Equation 6. The exact value of $\frac{joules}{computation}$ varies depending on several critical parameters. These are,

- Type of computation used in a particular encryption algorithm;
- The specific implementation of both the wireless network card and access point;
- The hardware/software tradeoff selected by the particular vendor to implement the encryption algorithm; and
- other.

Here, and as a first approximation, we will use the industry standard metric of $\frac{joules}{mac}$, as shown here in Figure 3, see **??**. Figure 3 depicts the improvements over time of most modern DSP processors. From this Figure, we can see that today a state of the art DSP spends about one-(1) milliwatt per million of MAC's (multiply and accumulate) operations or 10^{-15} joules per single MAC operation. Using, modern DSP processors as the basis for energy consumption in our analysis, and our earlier estimates of the number primitives operations, we can now compute the total energy utilization as required by Equation 8.

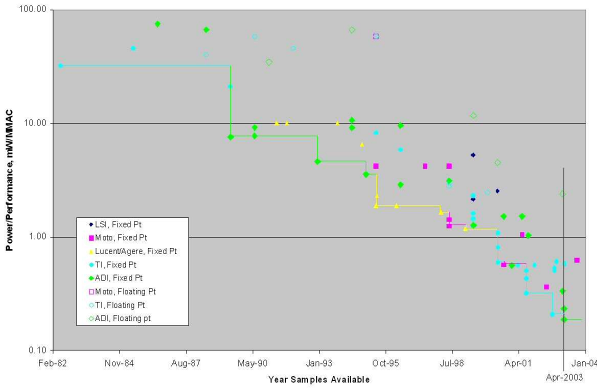


Fig. 3. Fento-joules per MAC - Modem DSP Processors

7 Analytical Evaluation - Results

The model presented in section 5 was applied to the following wireless encryption protocol using the methodology described in section 6.2 above.

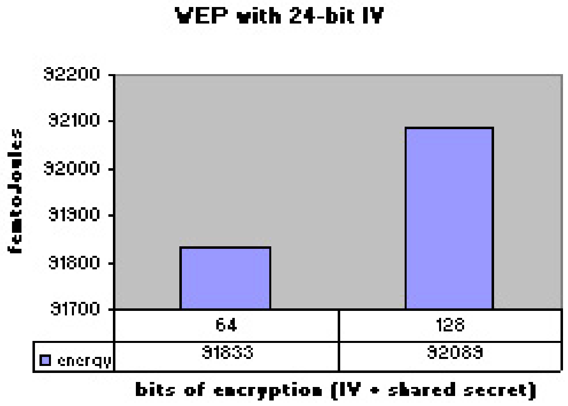
- WEP
- TKIP
- AES

  and several variants of authentication schemes, such as
- EAP-TLS, and
- EAP-Kerberos

In the following sections, the results from our evaluations are presented.

7.1 Wired Equivalent Privacy

The energy consumed by WEP encryption is directly linked to two things: the length of the encryption key (concatinatation of IV and shared secret) and the length of the data to be encrypted. The length of the plaintext will always be *data.length + 32* bits because a 32-bit CRC is appended to the data prior to encryption. The key scheduling algorithm (KSA) only deals with the key, and the energy used during this phase increases with the key length. The pseudo-random generating algorithm (PRGA) uses the result of the KSA to produce a stream of length equal to that of the plaintext. Therefore, both WEP encryptions will use the same amount of energy during this phase. As



th

Fig. 4. Estimated energy consumed by WEP with 24-bit IV

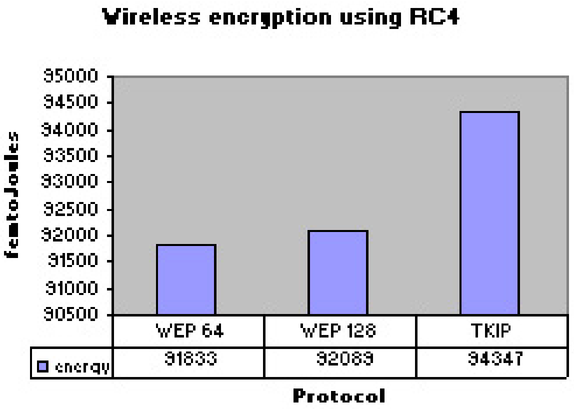


Fig. 5. Estimated energy consumed by RC4 encryption

expected, increasing the security (keysize) consumes more energy.

7.2 Temporal Key Integrity Protocol

There is only one key size specified for TKIP, which is 128 bits. Because of this, we need only to compare one value with that of the previous results. TKIP uses the same cipher for encryption as WEP. The difference in energy consumption is attributed to the key processing. The RC4 key used by TKIP goes through two phases of computations to incorporate the transmitter address prior to RC4's KSA and PRGA, which means more computation. In addition, TKIP uses a different integrity check (Michael) which adds 32 more bits to the plaintext. This causes the amount of PRGA and XOR operations that occur for a fixed amount of data to be higher in TKIP than in WEP. Michael is also more computationally expensive than a CRC. Overall, TKIP adds 2% more overhead to 128-bit WEP encryption.

TKIP encryption is more secure than WEP encryption because of the way keys are used. In WEP, all users share a secret with the access point, and with the same IV, will produce the same keystream. This is not the case with TKIP due to the key processing and distribution. Mixing in the transmitter address creates a different key for each user, even if the shared key is the same. The shared key is changed periodically so that keystreams are not reused. TKIP also adds rules for IV reuse, but that does not affect the number of computations or our results.

Nr	Nb = 4	Nb = 6	Nb = 8
Nk = 4	10	12	14
Nk = 6	12	12	14
Nk = 8	14	14	14

Fig. 6. Round combinations

7.3 Advanced Encryption Standard

The advanced encryption standard (AES) is a block cipher that can encrypt data in 128, 192, or 256 bit blocks, with 128, 192, or 256 bit encryption. The AES standard[FI01] only specifies the algorithm for block sizes of 128-bits. This is important to note because the number of rounds that occur vary by block size just as they do with different key sizes. The proposal submitted by Daemen and Rijmen[DA99] provided a chart (figure 6) that defined the number of rounds that need to take place for every key and block size combination. This chart was used in the construction of the results.

There are three variables in this algorithm that will dictate how each keylength/blocksize combination performs, with respect to energy, per block encryption. Nb represents the size of the block to be encrypted, and Nk is the key size. Both of these have three possible values, which are equivalent to the number of 32-bit words they have, or 4, 6, and 8. Nr is the number of rounds that executed. This attribute also as three possible values which are 10, 12, and 14.

It is helpful to note that in block ciphers, the message is divided into chunks specified by the block size. If any chunk is less than the block size, it is padded. This becomes important when encrypting messages of different sizes, as different block sizes are more efficient for different data lengths. Also, all encryptions shown are using ECB (electronic code book) mode, which does not link the blocks in any way.

Figure 7 shows the estimated energy consumed to encrypt 128-bits of data with each key and block length combination. You'll notice that this is equivalent to the

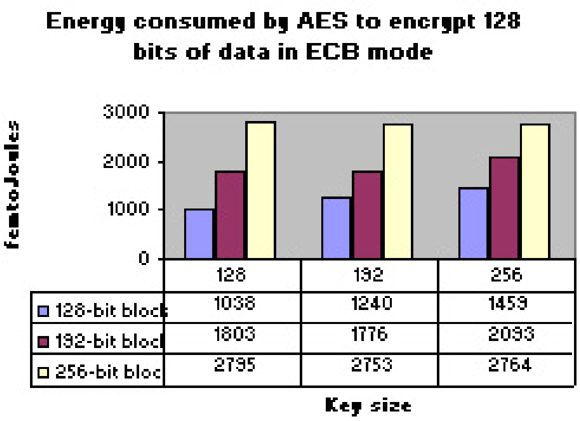
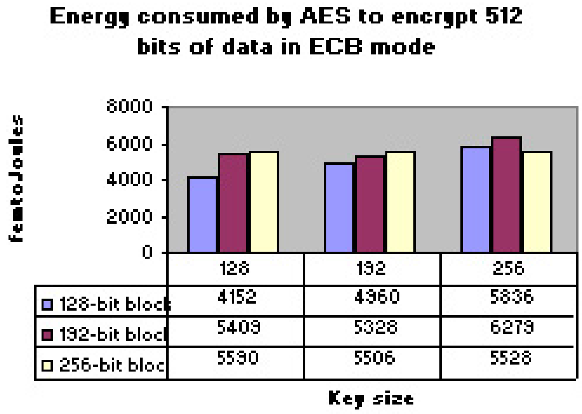
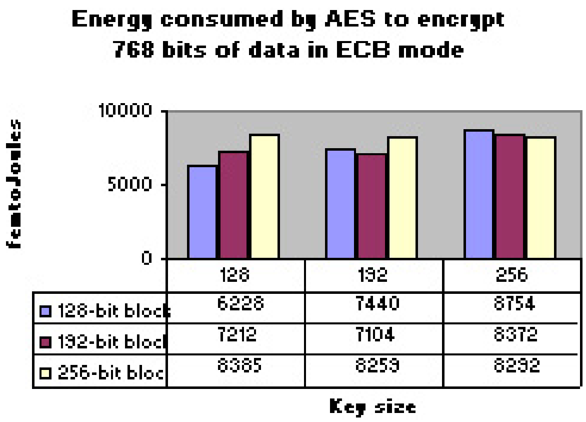


Fig. 7. Estimated energy used to encrypt 128 bits of data with AES



th

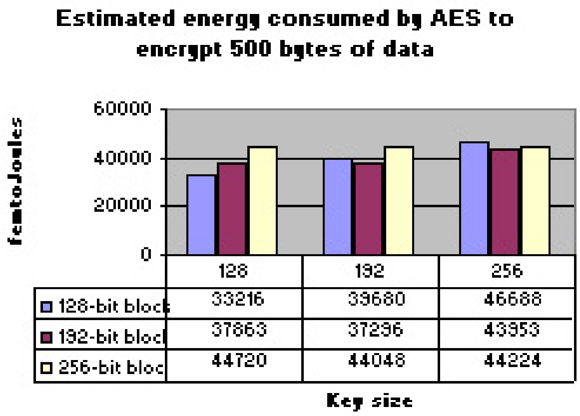
Fig. 8. Estimated energy used to encrypt 512 bits of data with AES



th

Fig. 9. Estimated energy used to encrypt 768 bits of data with AES

energy used to encrypt a block of each size. In this example, 128-bit block sizes are most energy efficient, as expected since the other lengths waste energy encrypting extra zeros.



th

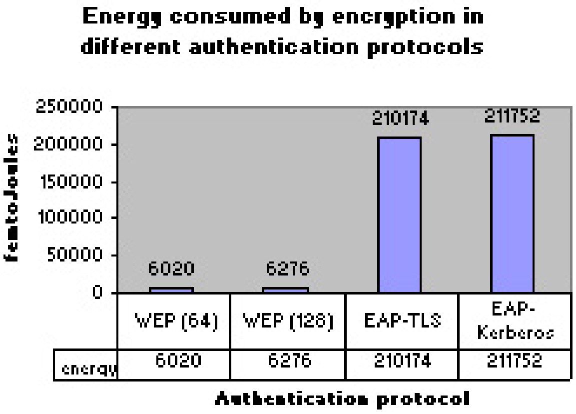
Fig. 10. Estimated energy used to encrypt 500 bytes of data with AES

Figures 8 and 9 show energy used computing 512 and 768 bits of data, respectively. These results are interesting because the curve for each block size remains the same, but their positions relative to each other change. This suggests that a particular block length could be optimal with a given key size if the average data length were known. The previous three charts were nice, but they are not accurate representation of the average packet. Figure 10 is a more realistic example. Here, the 128-bit key length is optimal when using a 128-bit block. The other two key sizes both use the least amount of energy when paired with 192-bit encryption.

AES was approved by NIST in 2001[FI01] after going under heavy scrutiny by the cryptography community. It has passed the test of many trained eyes, and still there are currently no plausible attacks known. Because of this, AES will be rated higher in security than the previous two RC4 encryption scheme.

7.4 Authentication

There are only two major forms of authentication that need to be considered: WEP authentication and EAP authentication. This is because both TKIP and CCMP use EAP. However, because there are a variety of EAP protocols, a number of them need to be evaluated. In figure 11, we have estimated energy consumed during authentication (includes encryptions and decryptions) for WEP with 24-bit IV and two variants of EAP. Note that this graph differs from those seen previously in that it is bar graph and not a curve. This was done intentionally because it is harder to determine levels of security. For example, in this context, both WEP authentications have the same security level because they use the same protocol, while it is harder to say whether Kerberos or TLS is more secure than the other. The protocols are not necessarily listed in order of security, so it makes no sense to create a curve.



th

Fig. 11. Energy consumed by different authentication schemes

It is easy to see that WEP requires far fewer encryptions than the two more secure protocols, EAP-TLS and EAP-Kerberos. It is also well-known that WEP authentication is absolutely insecure (no knowledge of the key is needed to authenticate), so we will focus on the others. The numbers for EAP-TLS and EAP-Kerberos were obtained using plausible transactions, but the numbers for both will vary for each instance of authentication as different amounts of data will be encrypted and transmitted.

EAP-TLS is a public key protocol, incorporating the popular Transport Security Layer protocol commonly used on the web as its upper layer. The TLS handshake uses public keys to encrypt the negotiation of a shared secret key and the cipher-suite (encryption algorithm, hash algorithm, mode, etc). This scheme allows some flexibility as far as what a particular supplicant supports, as well as allows session key negotiations be carried out with different cipher-suites. The handshake is also truncated for a supplicant who wishes to change the key that they currently have.

This method is very effective when mutual authentication takes place. However, when server-only authentication occurs, the entire thing is vulnerable to man in the middle attacks. Some administrators may steer away from mutual authentication EAP-TLS because it requires every client to have its own certificates, as well as the authentication server. The use of certificates is very helpful though. Certificates, unlike password schemes, are not subject to dictionary attacks.

EAP-Kerberos is still not specified formally, although there are a few opinions on what it would like[JE03]. The number of messages to authenticate will vary by structure (AS and TGS in AP or separate), but the computations for authentication should be similar in both.

Kerberos uses tickets to allow access to different resources. All messages are exchanged via private key cryptography. A pre-shared secret is used only once, to transmit a new session key. The idea is that if an attacker manages to get Alice's session key, they can only impersonate Alice for as long as that key is valid. Session length is specified by the administrator. The pre-shared secret is in the form of a password. As with all password-based authentication, it is vulnerable to dictionary attacks. This can be countered with the enforcement of a strong password policy, but one is not enforced by default.

From these tables, two things are immediately apparent. These are:

- There is very little difference across existent and proposed wireless protocols, from the perspective of the cryptographic algorithm, in terms of energy consumed per crypto operation. That is, energy consumption on a per cryptographic computation is dependent primarily on key sizes and not the algorithm selected. Of course, the security profile of different algorithms is significantly different.
- Authentication and Authorization protocols have a significant impact on the total energy consumed by the protocol at hand.

This last observation is critical when one considers that in wireless networks, the number of authentications and authorizations required can increase dramatically as the number of disassociation with the access point increase. Simply, as the wireless node looses connectivity due to weather, distance from the Access Point, topographical limitations, and roaming, the corresponding energy costs for authentications and authorizations will increase linearly. This last observation led the researchers to consider alternative cryptographic protocols that will minimize the number of messages exchanged per authentication and authorization. One such protocol is given here in **??**, called ε-sec. We believe, at first glance, that this protocol is optimal in the sense that it maximizes "security" while minimizing energy consumption.

8 Summary and Future Work

In this manuscript, we reviewed the current limitations of security protocols associated with 802.11 networks. We further developed a general model that helps with the understanding on how the current set of security related protocols, and protocol extensions, affect the energy consumption of the devices. The model, based on a decision-theoretic framework, requires both an energy cost function, called, C^E and a security-reliability measure, called R_M. The energy cost function, C^E, is the cost, both in energy and other system resources, of applying a countermeasure M_k against a specific protocol vulnerability V_i. The security-reliability measure, R_M, represents the level or measure of the security-reliability attained by countermeasure M_k on the overall security of the system. We showed that the model is general in the following senses:

- It can be used to analyze both static as well as adaptive security protocols. In static security protocols, such as WEP, the energy expended to counteract a particular class of attacks is fixed, $C^E(M_k, V_i)$, and it is determined a priori by a set of configurable parameters such as key size. Real time adaptive security protocols, on the other hand, when faced with an attack on a specific vulnerability, $V_(i)$ are capable of detecting the attack, real time, isolating the source of the attack, and launching a set of countermeasures whose energy costs are given by $C^E(M_k, V_i, A)$.
- Can be applied across multiple protocol layers.

Finally, and having define such a framework, we present our initial analysis and assessment of popular security protocol and protocol extensions, such as WEP, TKIP, AES, as well as several authentication schemes being proposed. Preliminary results showed that significant improvements can be obtained by the corresponding energy costs for authentications and authorizations.

Based on these preliminary results, a new wireless encryption protocol, calledε-sec, or Energy efficient secure protocol has been designed and it is the subject of a separate manuscript, see [CO04a]. This protocol has the potential to minimize power consumption while maximizing the security profile of the devices as well as the overall power consumption of the network.

8.1 Further Work

The work presented raises more questions than it answers. Fundamental to this work, is the basic idea of cost/benefit analysis. Unfortunately, as discussed in section 5, while several mechanisms exist (analytical tools, simulation, and empirical measurement) to quantify the costs (in terms of energy), measuring the benefits is significantly more difficult, except perhaps in the most simple of cases. For example, how does one go about answering the question how secure is the system, or how secure is the cryptographic protocol (not the algorithm itself)? Clearly, formal proofs can help in this area. One of the first challenges that we are tackling is precisely how to proof ε-sec formally. In addition,we are currently pursuing the following set of problems:

- Through experimental measurement, in both a Campus-wide and Corporate wireless network, measure the average number of lost connections with the Access Point.
- Based on this average number of lost connections, refine both the model, the analytic equations, Equations 7 and 8, and Figures 4 thru 10 in order to accurately compute the energy usage per security protocol class.
- Establish a detail implementation standard for ε-sec.
- Create a set of NS models that correctly represent the behavior of ε-sec in a network environment. Use such simulation experiments to validate our analytical results.
- Implement ε-sec using off-the-shelf components readily available from such vendors as LYNKSIS, CISCO, and others. Base on this reference implementation measure the energy costs associated with the protocol, and validate against our models, both analytic and simulation.
- Formally verify ε-sec for protocol correctness and vulnerability avoidance.
- Used the theoretical framework defined in section 5, and WEP as a baseline, empirically (through measurements), compare different security protocols (and protocol extensions) in terms of the energy consumption associated a single bulk file transfer, and the Security-Energy tradeoffs implied.

- Base on the results of the above evaluation, propose backwards compatible protocol extensions to 802.11 X.
- Finally, and in order to deal with the different effects of different attacks, we must parameterize the effect/impact measure V_i, for specific MAC layer, ad hoc routing and Internet (TCP/IP) protocol vulnerabilities.

References

[AB94] M Abadi, R. M. Needham, *"Prudent Engineering Practice for Cryptographic Protocols"*, DEC SRC Research Report no 125 (June 1 1994).
/bibitem [AB99] ab:99 B. Aboba and D. Simon, "PPP EAP TLS Authentication Protocol", RFC 2716, October 1999.

[AN95] Ross Anderson and Roger Needham *Programming Satan's Computer* , In Jan van Leeuwen, editor, Computer Science Today Recent Trends and Developments, volume 1000 of ln-cs, pages 426440. 1995.

[AR01] W.A. Arbaugh, N. Shankar, J. Wang and K. Zhang. *Your 802.11 Network has No Clothes* . First IEEE International Conference on Wireless LANs and Home Networks, Suntec City, Singapore, December 2001.

[BO01] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting Mobile Communications; The Insecurity of 802.11. Seventh Annual International Conference on Mobile Computing and Networking, Rome, Italy, July 2001.

[BE03] Bellardo J and Savage S, "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions", in Proc. USENIX Security Symposium, August 2003.

[CC02] , Doug Whiting, Russ Housley, and Niels Ferguson, "Counter with CBC-MAC (CCM): AES Mode of Operation", Submission to NIST, 2002.

[CC03] Morris Dworkin, "Recommendation for Block Cipher Modes of Operation: The CCM Mode For Authentication and Confidentiality (Draft)", NIST Special Publication 800-38C, September, 2003.

[CO04] F.C. Colon Osorio, E. Agu, and K. McKay, *Energy Trust Models for Wireless Security Protocols - Tradeoffs and Optimality* , WSSRL Technical Report, TR-WSSRL-2004-02-01, February, 2004.

[CO04a] F.C. Colon Osorio and K. McKay, *ε-sec, an Energy Efficient Protocol in Support of Wireless Networks"*, WSSRL Technical Report, TR-WSSRL-2004-03-02, March, 2004.

[CI02] Cisco Systems, "A Comprehensive Review of 802.11 Wireless LAN Security and the Cisco Wireless Security Suite", Cisco Systems, Inc., 2002.

[DA99] Joan Daemen and Vincent Rijmen, AES Proposal: Rijndael, March 1999.

[FE01] Feeney L M, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment", in Proc. INFOCOM 2001

[FE03] F. Ferrari, M Bernaschi, and L. Valcamonici, *"Access Point Vulnerabilities to DoS attacks in 802.11 networks"* to appear in Proceedings of WCNC2004, IEEE Wireless Communications and Networking Conference, Atlanta (Georgia-U.S.), 2004.

[FI01] , FIPS, Announcing the Advanced Encryption Standard (AES) - Federal Information Processing Standards Publication 197, Nov, 2001.

[FL01] Scott Fluhrer, Itsik Mantin, and Adi Shamir. *"Weakness in the Kay Scheduling Algorithm of RC4"* , IEEE 802.11-1999, http://www.drizzle.com/ãboba/IEEE.

[FL02] Scott Fluhrer, Wireless Lan Security Framework: void11. http://www.wlsec.net/void11, 2002.

[FU04] Sam Fuller, Vice President Analog Devices, "The Future of Computer Architecture: Connecting the real world to the digital world", Presented at Worcester Polytechnic Institute Department of Computer Science Colloquium: Distinguished Lecture Series,
Mar 2004).

[GA03] David Garmire and Stephen Sorkin, *"Power Conscious Routing for Wireless Devices: Theory and Simulation* UC Berkeley, 2003.

[GU02] Gupta V, Krishnamurthy S, Faloutsos M, "Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks", in Proc. IEEE MILCOM '02.

[HO97] Horvitz E and Lengyel J, "Perception, Attention and Resources: A Decision-Theoretic Approach to Graphics Rendering", in Proc. 13th Conf. on Uncertainty in Artificial Intelligence, August 1997, pp. 238-249

[HO02] Alireza Hodjet and Ingrid Verbauwhede, *The Energy Cost of Secrets in Ad Hoc Network* , In 2002 IEEE CAS Workshop, Pasadena, CA, September 5-6, 2002.

[IE99] *IEEE802.11b/D3.0, Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layer Specification: High Speed Physical Layer Extensions in the 2.4 GHz Band* , IEEE, 1999.

[JA01] M. Jakobsson and D. Pointcheval, *Mutual authentication for low-power mobile devices* , in Proc. Financial Cryptography, pp. 178195, Feb. 2001.

[JK93] , J. Kohl and C. Neuman, "The Kerberos Authentication Service (V5)", RFC 1510, September 1993.

[KA02] Charlie Kaufman, Radia Perlman and Mike Speciner, "Network Security: Private Communication in a Public World", Prentice Hall, 2002.

[KR02] R. Krashinsky and H. Balakrishnan, *Minimizing Energy for wireless Web Access with Bounded Slowdown* , in Proceedings of the Eight Annual ACM conference on mobile Computing and Networking, Atlanta, GA, September 2002.

[LA02] Y. W. Law, S. Dulman, S. Etalle,, and P. J. M. Havinga, *Assessing Security-Critical Energy-Efficient Sensor Networks* , Univ. of Twente, The Netherlands, Tech. Rep. TR-CTIT-02-18, July 2002.

[LO01] Lough M L, "A Taxonomy of Computer Attacks with Applications to Wireless", PhD Thesis, Virginia Polytechnic Institute, April 2001.

[LO03] Lou W and Fang Y, *"A Survey of Wireless Security in Mobile Ad Hoc Networks: Challenges and Available Solutions"*, in Ad Hoc Wireless Networking, Kluwer Academic Publishers, 2003.

[LY03] Lynksis Documentation, Wireless-G Cable/DSL Router, 2003.

[MA90] Ueli Maurer, *"A Probably-Secure Strongly-Randomized Cipher Advances in Cryptology"* - EUROCRYPT '90, Lecture Notes in Computer Science, Springer-Verlag, vol. 473, pp. 361-373, 1990.

[MA03] Stefan Dziembowski and Ueli Maurer, http://www.crypto.ethz.ch/research/itc/samba/

[MI03] Denial of Service Attacks Project at Mitsubishi Electric Research Laboratory (MERL), Cambridge, Massachusetts, http://www.merl.com/projects/DenialServiceAttacks.

[NA02] Nanda, Soumandra, "Wireless Insecurity / How Johnny can hack your WEP protected 802.11b Network!", Dartmouth College, 2002.

[NE95] RJ Anderson, RM Needham, *Robustness Principles for Public Key Protocols* , Crypto 95 pp 236-247.

[PO03] , Nachiketh R. Potlapally, Srivaths Ravi, Anand Raghunathan and Niraj K. Jha, *"Analyzing the Energy Consumption of Security Protocols"* , Proceedings of the 2003 International Symposium of Low Power Electronics and Design, August 25-27, 2003, Seoul, KOREA.

[RA98] C.S. Raghavendra and Suresh Singh, *PAMAS - Power Aware Multi-Access protocol with Signaling in Ad Hoc Networks* , ACM Computer Communications Review, 28(3):5-26, July 1998.

[SH46] C.E. Shannon, Confidential Memorandum, now de classified, *A Mathematical Theory of Cryptography* , dated Sept.1, 1946.

[SH02] E. Shih, V. Bahl, and M. Sinclair, *Wake-On-Wireless: An Event-Driven Energy Saving Strategy for Battery Operated Devices* in Proceedings of the Eighth Annual ACM conference on mobile Computing and Networking, Atlanta, GA, September 2002.

[SH03] Eugene Shih, Victor Bahl, Michael Sinclair, *Reducing Energy Consumption of Wireless, Mobile Devices Using a Secondary Low-Power Channel* , MIT Laboratory for Computer Science, March 2003.

[SC98] Bruce Schneier and Mudge, "Cryptanalysis of Microsoft's Point-to-Point Tunneling Protocol (PPTP)", ACM Conference on Computer and Communications Security", pp. 132-141, 1998, pages = "132-141", url = citeseer.nj.nec.com/article/schneier98cryptanalysis.html.

[SC99] Schneier and Mudge and Wagner, "Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2)", CQRE: International Exhibition and Congress on Secure Networking – CQRE [Secure]", 1999, url = "citeseer.nj.nec.com/article/schneier99cryptanalysis.html".

[ST97] M. Stemm and R. H. Katz, *Measuring and reducing energy consumption of network interfaces in hand-held devices* , IEICE Transactions on Communications, E80-B(8):1125-1131, Aug. 1997.

[VA03] Salil Vadhan, *"Locally Computable Extractors and Cryptosystems in the Bounded Storage Model"*, Lecture Notes, Harvard University, 2003, http://eecs.harvard.edu/s̃alil.

[WO98] woe:98 H. Woesner et al., *Power-saving mechanisms in emerging standards in wireless LANs: the MAC level perspectives* , IEEE Personal Communications: The Magazine of Wireless Communication and Networking, Vol. 5, no. 3, pp. 4048, June 1998.

[WO01] D. S. Wong and A. H. Chan, *Mutual authentication and key exchange for low power wireless communications* , in Proc. IEEE MILCOM Conf., pp. 3943, October 2001.

[WO02] Wood A D and Stankovic J A, "Denial of Service in Sensor Networks", in IEEE Computer, October 2002, pp. 48-56.

[JE03] Jon Edney and William A. Arbaugh, "Real 802.11 Security: Wi-Fi Protected Access and 802.11i", Addison Wesley, July 2003, isbn = 0-321-13620-9.

[Jy98] Chen J-C, Sivalingam K M, Agrawal P and Kishore S, "A Comparison of MAC Protocols for Wireless Local Networks Based on Battery Power Consumption", in Proc. INFOCOM '98

[Jy99] Chen J-C, Sivalingam and Agrawal P, "Performance Comparison of Battery Power Consumption in Wireless Multiple Access Protocols", Wireless Networks 5 (1999), pp 445-460.

[Ky03] Kyasanur P and Vaidya N, "Detection and Handling of MAC Layer Misbehavior in Wireless Networks", in Proc. 2003 International Conference on Dependable Systems and Networks (DSN '03).

[Mi02] Michiardi P and Molva R, "Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks", in Proc. European Wireless 2002 Conference.

[Ha02] Handy M and Timmermann D, "Simulation of Mobile Wireless Networks with Accurate Modelling of Non-Linear Battery Effects", in Proc. 4th International Conference on Applied Simulation and Modelling 2003, Marbella

[Ra01] Rakhmatov D and Vrudhula S, "An Analytical High-Level Battery Model for Use in Energy Management of Portable Electronic Systems", in Proc. International Conference on Computer Aided Design (ICCAD '01), 2001.

[Ra02] Rakhmatov D, Vrudhula S and Wallach D, "Battery Lifetime Prediction for Energy-Aware Computing", in Proc. International Symposium on Low Power Electronics and Design (ISLPED '02), 2002.

[RF99] T. Derks and C. Allen, The TLS Protocol, RFC 2246, January 1999.

[SC98] Bruce Schneier and Mudge, "Cryptanalysis of Microsoft's Point-to-Point Tunneling Protocol (PPTP)", ACM Conference on Computer and Communications Security", pp. 132-141, 1998, pages = "132-141", url = citeseer.nj.nec.com/article/schneier98cryptanalysis.html.

[Si02a] Singh H and Singh S, "Energy Consumption of TCP Reno, Newreno and SACK in multi-hop wireless networks", in Proc. SIGCOMM 2002

[Si02b] Singh H, Saxena S and Singh S, "Energy Consumption of TCP in Ad Hoc Networks", in Proc. MSWiM 2002.