Polytechnic UNIVERSITY

Optimizing IPSEC for Energy Efficient Secure Wireless Sessions

Ramesh Karri, Piyush Mishra

April 2002

WICAT TR 02-003

OPTIMIZING IPSEC FOR ENERGY-EFFICIENT SECURE WIRELESS SESSIONS

Ramesh Karri and Piyush Mishra
Department of Electrical and Computer Engineering
Polytechnic University, Brooklyn, 6 Metrotech Center, NY, US 11201

Abstract: Deployment of security protocols in battery-powered mobile devices has elevated energy consumption to an important design metric of network design. In this chapter we identified the various sources of energy consumption during the setup, operation and tear down of a secure wireless session. We used Internet Security Protocol (IPSec) for our analysis and developed techniques based on information compression, session negotiation protocol optimization and choice of cryptographic primitives to reduce the energy consumed by a secure wireless session. A mobile test bed was developed to validate our energy management schemes which showed that the proposed schemes were able to reduce the session establishment energy by more than 6.5× and the secure data communication energy by more than 25× during data transmission and by more than 3.8× during data reception.

Key words: Mobile, Security protocols, Wireless, Energy-efficiency, IPSec, Secure session.

1. INTRODUCTION

Rapidly evolving personalized network applications, such as online health, commerce and education services, have fueled the need for securing data communication networks [CL00]. Security protocols used to provide the required security mechanisms employ session negotiation protocols for establishing and managing secure sessions and secure data exchange protocols for secure data communications. In addition to data confidentiality, authenticity and integrity security protocols also provides features such as system resource protection and audit-trails.

Traditionally, researchers have focused on security, complexity, and throughput metrics while designing security protocols for wired networks. Due to the increasing size of the mobile computing and communication device market deployment of computation and communication intensive security protocols in battery-powered mobile devices has elevated energy consumption to another important design metric [SE00]. Further, lack of well-defined boundaries in wireless networks and other physical constraints imply stronger security requirements, which in turn increase the energy consumed by security protocols operating on these mobile devices.

There are two main sources of energy consumption during a secure wireless session: (i) cryptographic computations used to establish the secure session and to support encryption and authentication during secure data transaction and (ii) message exchanges during secure session establishment and data transfers during secure data transactions. We considered a Symbol PPT2800™ Pocket PC device running Windows CE™ 3.0 operating system and equipped with an 11 Mbps Spectrum24™ wireless LAN adapter card as the mobile test bed[1] to measure the energy consumed by a secure wireless session while transmitting 64 KB data over an 802.11b WLAN channel. Secure data transaction used 3DES encryption [DE], SHA-256 message authentication code [SH] and consumed 952

[1] A detailed description of the mobile test bed and experimentation methodology is outlined in section 3.1

milli Joules. Figure 1 shows that idle system consumes 44% of the system energy followed by data transmission and cryptographic computations which consume 35% and 21% of the system energy respectively. Idle system energy corresponds to the energy consumed by the mobile test bed between the transmissions of packets and is determined by the sustained throughput of the system, which in turn depends upon the network conditions, application requirements, etc.

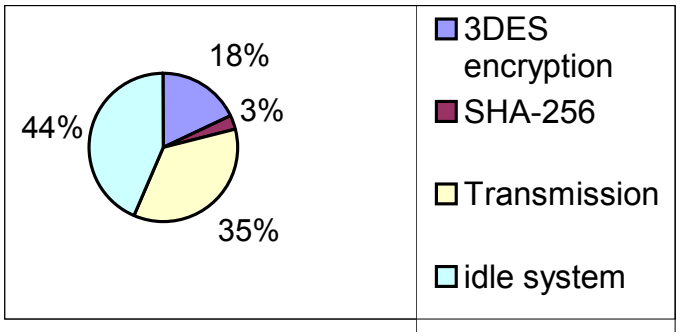

Figure 1. Energy consumed by a secure wireless session while transmitting 64 KB data over an 802.11 wireless channel using IPSec.

In this chapter we will accurately measure the energy consumed by various components of a security protocol used for establishing and managing a secure wireless session, present techniques to minimize their energy consumption and investigate the associated energy *vs.* security trade-offs. We selected Internet Security Protocol (IPSec) for our study since network layer security protocols are comprehensive, mobile and transparent to users.

1.1 Related Research

Techniques developed to reduce the energy consumed by wireless data communication include adaptive network interface control, data compression, optimized network protocols and adaptive error control techniques.

Woesner et. al. studied the power saving features of wireless LAN standards and presented simulation studies for energy-efficient ad-hoc configurations [WE98$^+$]. Ebert et. al. presented a packet length dependent power control mechanism for optimal RF power level [ES00$^+$], while Rulnick and Bambos considered autonomous transmitters to determine optimal level of transmit power given a set of quality-of-service constraints and information on the nature of channel and the level of interference at the receiver [RB96].

Kravets and Krishnan presented a transport level protocol to selectively suspend communication and shut down the communication device and studied the tradeoff between power consumption and data delay [KK99]. Kravets et. al. presented a pay-off adaptation based energy-efficient scheme for distributed applications [KC98$^+$]. Rohl et. al. studied the influence of typical parameters on the power saving mechanisms in IEEE 802.11 [RW98$^+$]. Singh and Raghavendra developed a power-efficient multi-access protocol for ad-hoc radio networks and showed that the idea can be easily extended to other access protocols [SR98a, SR98b]. Zorzi and Rao developed probing schemes for energy-efficient error control protocols and a formal approach to track complex models for power sources including dynamic charge recovery in batteries [ZR97a, ZR97b]. Lettieri et. al. presented another energy-efficient error control protocol with hybrid combination of an appropriate forward error correction (FEC) code and automatic repeat request (ARQ) protocol that adapts over time for each data stream [LF97$^+$].

Our study differs from these works in two aspects: (1) it focuses on the energy consumption characteristics of computation-intensive security protocols which, till now, have received little attention

and (2) unlike most of the other simulation based studies, we will use real-life mobile test bed to validate the techniques presented for reducing the energy consumed by secure wireless sessions.

1.2 Outline

In section 2 we describe negotiation, management and operation of secure wireless sessions within the IPSec framework. In section 3 we describe the experimental setup and methodology and use the results of energy measurement experiments to estimate the energy consumed during IPSec secure sessions. In section 4 we present the techniques for optimizing the energy consumed by secure wireless sessions and validate these techniques using IPSec framework. Section 5 concludes this chapter.

2. IPSEC SECURITY PROTOCOL

Security protocols first negotiate security associations (SAs) between the communicating parties using a session negotiation protocol. Secure session negotiation entails (1) *mutual authentication* of the communicating parties, (2) *parameter negotiation* to agree upon the SAs primitives (for example, a list of data encryption and authentication algorithms and their parameters) and the key exchange and management protocol and its parameters, (3) *key exchange and management* to generate a set of secret keys shared exclusively among the communicating parties, and (4) *SAs establishment* to establish a secure session using the negotiated SAs. Secure session negotiation protocols should ensure perfect forward secrecy and be scalable. After successfully establishing a secure session security protocols carry out secure data communication using the agreed upon SAs.

IPSec session negotiation uses Internet Key Exchange (IKE) protocol [HC98] to establish and manage SAs. Design of IPSec session negotiation protocol is influenced by Station-to-Station (STS) [DO92^{+}], SKEME [KR96], and OAKLEY [OR98] protocols. It uses Diffie-Hellman key exchange and management [DH76] to generate the shared secret keys and one of four authentication mechanisms - pre-shared secret, public key signature, public key encryption and revised public key encryption- for mutual authentication (see [HC98] or [CH01] for detailed description of each mechanism).

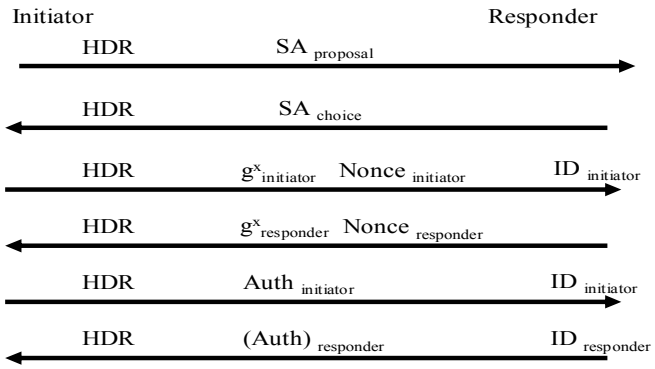

Figure 2. Messages exchanged by IPSec session negotiation during *first* SA negotiation.

IPSec session negotiation protocol first authenticates the communicating parties and runs a public-key based key exchange and management protocol (Diffie-Hellman) to generate the *first* shared SA using either main or aggressive mode. Aggressive mode collapses the 6 messages of the main mode into 3 messages at the expense of constrained negotiation space and unsecured identities. Then, under the protection of this *first* SA, it runs a private-key based key exchange and management protocol *frequently*

to establish and manage multiple IPSec SAs. In general, the *first* SA is negotiated between the gateways and the proxies, while IPSec SAs are negotiated on behalf of the end clients. In a mobile environment the mobile client may act as its own proxy.

Figure 2 shows the protocol messages exchanged during *first* SA negotiation in the main mode and Figure 3 shows the protocol messages exchanged during IPSec SA negotiation. HDR contains session-specific information (such as session ID etc), SA contains a list of cryptographic algorithms, nonces are large random numbers, and IDs are the unique identities of the communicating parties. Parameters in '[]' are optional.

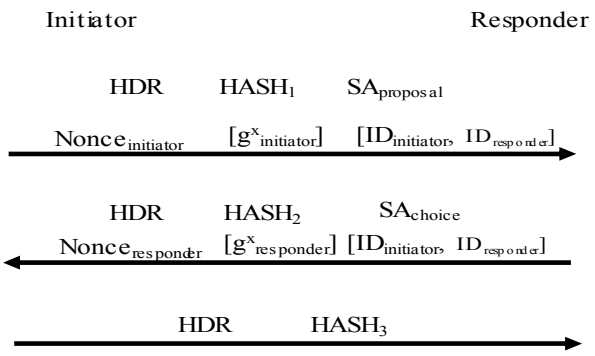

Figure 3. Messages exchanged by IPSec session negotiation during IPSec SA negotiation.

After successfully establishing the secure session IPSec (either at the client or at the server) accepts plain text messages, computes the MAC, encrypts the data and transmits it. At the other end, received data is decrypted and verified. Security of a session is enhanced by periodically refreshing the SAs.

3. ENERGY CONSUMPTION OF A SECURE WIRELESS SESSION

We computed the total energy consumed during a secure session negotiation as the energy consumed for authenticating the communicating parties (sign and verify certificates) + energy consumed for negotiating the parameters (exchange SA primitives) + energy consumed for key exchange and management (generate shared secrets) + energy consumed for SA establishment (generate secret keys for encryption + generate secret keys for message authentication (MAC) and the total energy consumed for secure data exchange as the energy consumed to authenticate data (sign and verify) + energy consumed to secure data (encrypt and decrypt) + energy consumed to exchange data (transmit and receive) + energy consumed to manage SAs.

3.1 Test Bed for Energy Measurements

The mobile test bed shown in Figure 4 consists of a PPT2800™ Pocket PC device [SP] equipped with an 11 Mbps Spectrum24™ wireless LAN Adapter card (IEEE 802.11b) [SSb], both from Symbol Technologies Inc.

Pocket PC is running WindowsCE™ 3.0 operating system on a 32-bit, 206 MHz StrongArm™ SA-1110 processor with 16 MB flash ROM, 16 MB RAM, 16 KB instruction cache and 32-way set associative 8 KB data write back data cache. The WLAN card is operating in polling mode P1[2]. We measure the current drawn by an application executing on the mobile test bed using a Tektronix TCP202

[2] See section 3.2.1

current probe (DC to 50 MHz, Min sensitivity: 10 mA/div, DC accuracy: ±1% with probe calibrator) and a Tektronix TDS 3054 oscilloscope (4 channel, 500 MHz, 5 GS/s) [TI]. Voltage was held constant at 4 V, the nominal operating voltage of the mobile test bed. In order to ensure consistency and accuracy of our results, we averaged each of our results over several thousand iterations for each application with data residing in the main memory and data and instruction caches enabled.

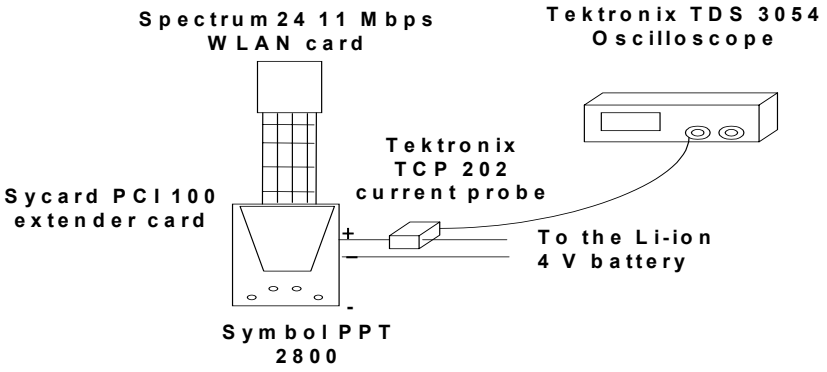

Figure 4. Mobile test bed for performance and energy measurements.

3.2 Sources of Energy Consumption

This section presents the results of our energy measurement experiments for the various sources of energy consumption.

3.2.1 Wireless Transceiver Subsystem

The 11 Mbps Spectrum24® LA-4121 wireless LAN card operating at 5 V supports a continuous access mode (CAM) and five polling modes, P1 to P5 [SSb]. Current consumed by the WLAN card during active transmission and reception is approximately uniform across all polling modes. Table 1 shows that the current, and hence power, consumed by LA-4121 wireless LAN card depends upon its mode of operation. Transmission energy is the product of power consumed in transmit-mode and the time to transmit data.

Table 1. Power consumed by 11 Mbps Spectrum24® LA-4121 WLAN card.

	Continuous access mode		P1 polling mode	
	Current (mA)	Power (W)	Current (mA)	Power (W)
Sleep	-	-	10	0.05
Idle	170	0.85	30	0.15
Receive	190	0.95	190	0.95
Transmit	410	2.05	410	2.05

3.2.2 Message Authentication Code

We used SHA-256, a variation of a 256-bit symmetric block encryption algorithm, as the message authentication code (MAC) [SH]. SHA-256 encrypts the intermediate hash values using the message blocks as the keys. Figure 5 shows the energy consumed by optimized ‘C’ implementation of SHA-256 for different data sizes.

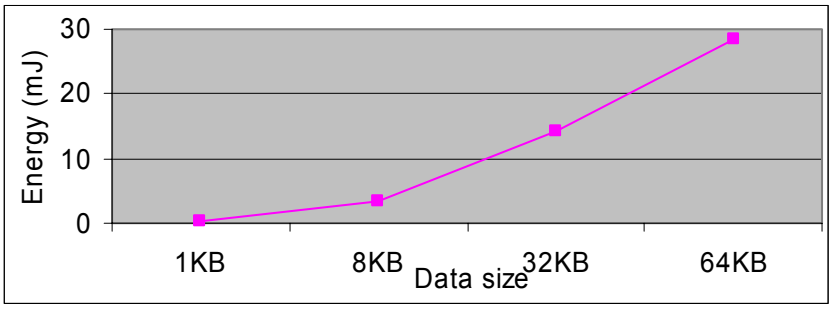

Figure 5. Energy consumed by SHA-256 MAC.

3.2.3 Data Encryption

Data Encryption Standard (DES) and triple-DES (3DES) are private key encryption algorithms that have been widely used for more than 20 years now to ensure data privacy [DE]. Recently National Institute of Standards and Technologies (NIST) selected Rijndael as the Advanced Encryption Standard (AES) to replace DES and 3DES in systems with higher security and performance requirements [AE]. AES supports multiple user key lengths (128, 192, or 256 bits) and multiple data block sizes (128, 192 or 256 bits) [DR]. Security level and the energy consumption of a private key encryption algorithm increase with the number of encryption rounds and the length of the user key. Table 2 summarizes the energy consumed by round-key generation and encryption of optimized 'C' implementations of AES. Energy consumed by AES key generation increases with the key size at a rate greater than the energy consumed by its encryption due to an increase in the complexity of key generation and an increase in the number of round keys to be generated.

Table 2. Energy consumed by AES key-schedule and data encryption.

	AES					
	Key-schedule			Encryption		
Key-size (bits)	128	192	256	128	192	256
Energy (μJ)	10.44	13.70	17.44	0.067	0.07	0.075
Time (μS)	7.48	9.82	12.47	0.0385	0.0407	0.0415

3.3 Energy Consumed by a Secure Wireless Session

3.3.1 Negotiating a Security Association

Table 3 summarizes the energy consumed during IPSec secure session parameter negotiation and mutual authentication as a function of size of messages exchanged and cryptographic computations performed. We assume that client is the initiator and server is the responder. Energy values in bold correspond to the client. Client SA proposal includes DES [DE], 3DES [DE] and AES [AE] encryption algorithms for Encapsulating Security Payload (ESP) protocol [KE02] and SHA-256 [SH] message authentication code for Authentication Header (AH) protocol [KE98] and server selects 3DES and SHA-256 for the SA. Mutual authentication is pre-shared secret based and the *first* SA negotiation is carried out in the main mode.

Table 3. Energy consumed by IPSec mutual authentication and parameter negotiation.

Mutual Authentication, Parameter negotiation (Main mode, Pre-shared secret)				
Messages Exchanged		Energy consumed (milli Joule)		
		Cryptographic computations	Transmit	Receive
M1	HDR	**0.01**	**2.5**	1.25
	$SA_{proposal}$	-	**5**	2
M2	HDR	0.01	2.5	**1.25**
	SA_{choice}	-	4	**1.5**
M3	HDR	-	**2.5**	1.25
	$g^{x}_{initiator}$	**60.21**	**337.2**	145.1
	$Nonce_{initiator}$	**0.01**	**2**	1
M4	HDR	-	2.5	**1.25**
	$g^{x}_{responder}$	60.21	337.2	**145.1**
	$Nonce_{responder}$	0.01	2	**1**
M5	HDR	-	**2.5**	1.25
	$ID_{initiator}$	**0.01**	**2.6**	1.3
	$Hash_{initiator}$	**12.3**	**3**	2
M6	HDR	-	2.5	**1.25**
	$ID_{responder}$	0.01	2.6	**1.3**
	$Hash_{responder}$	12.3	3	**2**
Client Total		**72.54**	**357.3**	**154.65**
Server Total		72.54	356.3	154.15

Table 4 shows the energy consumed by the IPSec key exchange to generate the shared secret and derive the secret keys for establishing IPSec SAs.

Table 4. Energy consumed by IPSec key exchange.

Key exchange and management (Main mode, Pre-shared secret)	
Cryptographic computations	Energy consumed (milli Joule)
SKEYID	**0.01**/0.01
$SKEYID_{sub\text{-}keys}$	**12.3**/12.3
$SKEYID_{authentication}$	**12.3**/12.3
$SKEYID_{encryption}$	**12.3**/12.3
TOTAL	**36.91**/36.91

Table 5 shows the energy consumed by IPSec *SA establishment*. More than 90% of the energy consumed during SA establishment is due to the exchange of large size certificates. Table 6 summarizes the total energy consumed by the IPSec session negotiation protocol at the client and at the server in main and aggressive modes corresponding to the various authentication mechanisms. From Table 6 we can see that although aggressive mode exchanges only three messages it consumes approximately the same amount of energy as the main mode that exchanges six messages. This is because both modes exchange approximately the same amount of information and perform identical cryptographic computations.

Table 5. Energy consumed by IPSec SA establishment.

IPSec SA Establishment				
Messages exchanged		Energy consumed (milli Joule)		
		Cryptographic computations	Transmit	Receive
M1	HDR	-	**2.5**	1.25
	$SA_{proposal}$	**0.01**	**5**	2
	$Hash_1$	**12.3**	**3**	2
	$Nonce_{initiator}$	**0.01**	**2**	1
	$g^x_{initiator}$	**2**	**337.2**	145.1
	$ID_{initiator}$	**0.01**	**2.6**	1.3
	$ID_{responder}$	**0.01**	**2.6**	1.3
M2	HDR	**-**	2.6	**1.3**
	SA_{choice}	0.01	5	**2**
	$Hash_2$	12.3	3	**2**
	$Nonce_{responder}$	0.01	2	**1**
	$g^x_{responder}$	2	337.2	**145.1**
	$ID_{initiator}$	0.01	2.6	**1.3**
	$ID_{responder}$	0.01	2.6	**1.3**
M3	HDR	**-**	**2.5**	1.25
	$Hash_3$	**0.01**	**5**	2
Key generation		**12.3**/12.3	**-**	**-**
Client Total		**26.65**	**362.4**	**154**
Server Total		26.64	355	157.2

As far as authentication schemes are concerned, IPSec clients using public-key encryption and pre-shared secret based authentication consume equivalent energy while the energy consumed by client supporting revised public key encryption based authentication is comparable to the one supporting public-key signature based authentication. Of the 1521 milli Joules consumed by a client supporting revised public-key encryption based authentication 89% is consumed by the message exchanges and 11% is consumed by public key cryptographic computations.

Table 6. Energy consumed by IPSec session negotiation protocols.

Mode	Authentication method	Client	Server
Main	Pre-shared secret	1165	1159
	Public key signature	1646	1640
	Public key encryption	1185	1179
	Revised public key encryption	1521	1319
Aggressive	Pre-shared secret	1164	1159
	Public key signature	1642	1636
	Public key encryption	1185	1179
	Revised public key encryption	1521	1319

3.3.2 Secure Data Transaction

Following successful secure session establishment, secure data exchange protocol accepts plain text messages, computes the MAC, encrypts the data and transmits it. At the other end, received data is decrypted and verified. Security of a session is enhanced by periodically refreshing the SA and the encryption and the MAC keys. Table 7 summarizes the energy consumed during secure wireless data transmission assuming following security association: 3DES encryption, SHA-256 MAC, key refresh

rate that entails re-computing encryption and MAC keys every 128 KB of data and SA refresh every 2 MB of data.

Table 7. Energy consumed by secure wireless data transmission.

Mobile-to-server secure data communication energy (milli Joule)		
	2560 KB data	8 KB data
SHA-256 MAC	1130	3.53
3DES encryption	6858	21.43
Transmission	13480	42.13
Key refresh	245	-
Idle system	16604	51.87
Total	38317	118.96

It does not include the energy consumed by SA refreshes which is essentially the same as the energy consumed during secure session negotiation. Energy consumed by idle system is inversely proportional to the sustained throughput of the system, where sustained throughput of a system is determined by a variety of factors including the network condition, efficiency of the wireless protocols and the throughput requirements of the application. Energy consumed by data transmission and reception increases linearly with the input data size while the energy consumed by cryptographic computations increases linearly with both data size and security level. For example, while transferring an 8 KB of data does not entail any key refresh overhead, transferring 2560 KB of data entails 19 key refreshes, consuming 245 milli Joules each at the client and at the server. Refreshing the secret keys entails generating new encryption and MAC keys and generating fresh round keys using the new encryption key.

Similarly, large encryption keys and higher number of encryption rounds also improves the level of security at the cost of extra energy consumption and performance degradation. Cryptographic computations (key refresh, data encryption and MAC authentication) consume 7.7% of the total energy for transferring 8 KB data, and this increases to 8.4% of the total energy for transferring 2560 KB data.

4. DESIGNING ENERGY-EFFICIENT SECURITY PROTOCOLS

We will now describe techniques to optimize the energy consumed by security protocols by systematically considering the various sources of energy consumption - authentication, parameter negotiation, key exchange and management and secure data exchange - and study the impact of these techniques on the energy consumption characteristics of the protocols providing these services.

4.1 Reducing the Energy Consumed by Data Exchanges

Data exchanges account for a considerable fraction of the energy consumed by session negotiation (more than 90%) and secure data communication (more than 40%). Therefore, reducing the amount of data exchanged during a secure wireless session can significantly reduce its energy consumption. This can be achieved by reducing either the size of the data or the number of data exchanges or some combination thereof.

4.1.1 Data Compression

Compressing data before encryption and transmission and decompressing it after reception and decryption reduces the energy consumed during secure wireless data communication if the energy

savings due to the reduced data size are more than the extra energy consumed by compression and decompression. Similarly, compressing session negotiation messages before transmission and decompressing them after reception may reduce the energy consumed by secure session negotiations.

Table 8. Energy consumed by DEFLATE compression[3].

Data size		CL = 9 ML = 9	CL = 1 ML = 9	CL = 9 ML = 1	CL = 5 ML = 5
64 KB	Energy (milli Joule)	1004.15	132.59	2785.15	395.79
	Compression ratio	4.482	3.5884	4.0042	4.3256
8 KB	Energy (milli Joule)	55.71	46.11	65.52	31.69
	Compression ratio	3.5085	3.1059	3.1035	3.4782
1 KB	Energy (milli Joule)	26.62	34.57	14.24	14.76
	Compression ratio	2.8679	2.7861	2.4805	2.8254

To study the impact of compression we used optimized 'C' implementation of DEFLATE loss-less data compression algorithm [DC]. Compression level, history window size, and memory-level are the three important parameters that affect the energy consumed by DEFLATE compression. Table 8 summarizes the energy consumed by DEFLATE while compressing 1KB, 8 KB and 64 KB size benchmarks from Calgary corpus [BE90]. It can be seen that matching the compression block size to the data cache size (8 KB for our mobile test bed) saves significant energy. Further, increasing either the compression level or the memory level results in a proportional increase in the energy consumed without a corresponding increase in the compression ratio, while increasing the size of the history window yields a proportional increase in the compression ratio without a corresponding increase in the energy consumed. We found that medium compression level (level 5), medium memory level (level 5), and maximum history window size (32 KB) combination achieves a compression ratio close to the best while consuming significantly less energy. Decompression energy is very small compared to the energy for compression (~10× less for DEFLATE) since decompression involves fewer and simpler computations. Hence, energy consumed by the client can be reduced significantly if the server/gateway compresses all the data following session initiation messages.

Table 9. Energy saved by compressing the session negotiation protocol messages.

Energy consumed by secure session negotiation protocol at client (milli Joule)		
	IPSec (Main mode, Revised public key encryption)	
	first SA	IPSec SA
Uncompressed	978	543
Compressed	912	460
Energy saving	1.1×	1.18×

Table 9 shows a 1.13× reduction in the energy consumed by IPSec session negotiation protocol by compressing the messages that exchange authentication certificates and shared secret.

Energy consumed during transmission and reception of uncompressed and compressed, 3DES encrypted 2560 KB data is summarized in Table 10. Energy consumed by data transmission is 2× the energy consumed by 3DES encryption which in turn is 0.65× the energy consumed by DEFLATE data compression for 8 KB compression block size. Besides reducing the size of the data to be encrypted and transmitted, compression also reduces the energy consumed by key refreshes. For example, if encryption and MAC keys are refreshed after exchanging every 128 KB of data, an uncompressed 2560 KB secure

[3] CL: Compression level, ML: Memory level

data communication requires 19 key refreshes and consumes 245 milli Joules, while the compressed 736 KB data (=2560 KB÷compression ratio of 3.48) requires only 5 key refreshes and consumes 64.5 milli Joules.

Table 10. Energy consumed during a secure wireless data communication.

Energy consumed while transmitting 2560 KB data (milli Joule)		
	Uncompressed	Compressed
Compression	-	10141
3DES encryption	6858	1971
3DES decryption		
SHA-256 sign	1130	324.9
SHA-256 verify		
Transmit	13480	3876
Receive	5803	1668
Decompression	-	1014
Key refresh	245	64.45
Idle system	16604	4773
Total transmit energy	38317	21150
Transmit energy saving factor	-	1.81×
Total receive energy	30640	9815
Receive energy saving factor	-	3.12×

Therefore, data compression reduces (a) transmission, reception, encryption and decryption energy during secure data communication (b) number of key refreshes required and the corresponding energy, and (c) energy consumed by the idle system. It allows either a higher level of security (large-size certificates and keys) within an energy budget or minimizes the energy consumed while maintaining a minimum level of security.

4.1.2 Refreshing the Security Association

Wireless networks are inherently unreliable and discontinuous resulting in frequent secure session negotiations. Network services may become unavailable due to bad radio coverage, shortage of resources or network roaming. Therefore, session resumption and transaction recovery, together with fast secret key generation are key to energy efficiency of mobile clients. As shown in the last section, compressing the messages exchanged during secure session negotiation reduces the energy consumed by the mobile client by more than 10%. Client energy can also be reduced by modifying the session negotiation protocols such that the server looks up the client's certificate from its own source (*session negotiation variant 1*). Embedding the client's shared secret in its certificate can further reduce this energy. When establishing a new session, a client-server pair can exchange new client and server nonces and combine these with previously negotiated security association (*session negotiation variant 2*). Finally, implanting the shared secret in the server and mobile client eliminates the energy consumed by shared secret exchange messages (*session negotiation variant 3*). Table 11 shows the energy consumed by these variants of IPSec secure session negotiation protocol.

We propose an *adaptive session negotiation protocol* that uses *session negotiation variant 1* to establish a new secure session and *session negotiation variant 2* to refresh the security association by exchanging new client and server random numbers if the session lasts beyond a certain number of messages determined by the security requirements of the session or if the session is disrupted abruptly due to bad channel conditions or temporary network outage.

Table 11. Energy consumed by variants of IPSec secure session negotiation protocol.

IPSec revised public key encryption based session negotiation protocol		
Variants	Operation	Energy (milli Joules)
Basic	Server and client exchange certificates	1520
Variant 1	Server looks up client's certificate from its own source or provided URL, Uses compression	1033
Variant 2	Combine previous SAs with newly exchanged random values to generate new session parameters	51

4.2 Reducing the Energy Consumed by Cryptographic Computations

Cryptographic computations constitute as much as 10% of the energy consumed during session negotiation and 60% of the energy consumed during secure data communication. Energy consumed by cryptographic computations can be reduced by (1) selecting energy-efficient cryptographic mechanisms, (2) trading-off security for energy, and (3) energy-efficient hardware acceleration. This chapter does not discuss the last technique which has already been covered extensively in literature [GC01, PR02$^+$, KM02].

4.2.1 Choice of Cryptographic Mechanism

Choice of authentication, key exchange and management, and data encryption mechanisms can reduce the energy consumed by security protocols at the mobile clients. IPSec session negotiation protocol uses Diffie-Hellman key exchange and management for exchanging shared secrets and RSA public-key scheme for mutual authentication. In wired networks, these two operations are isolated to ensure perfect forward secrecy; otherwise anybody who somehow gains access to the RSA private keys can obtain access to the future as well as the past communications. On the other hand, in a mobile wireless environment such risks are minimal due to the relatively shorter life-span of secure wireless sessions. For example, IPSec session negotiation protocol derives all secret keys using the ephemeral nonces which expire as soon as the wireless session terminates. Therefore, energy consumed by the IPSec session negotiation can be reduced by using RSA protocol for both exchanging the shared secrets and for mutual authentication. This observation is confirmed by results in Table 12 where replacing Diffie-Hellman key exchange and management protocol with the RSA key exchange and management protocol reduces the session negotiation energy consumption by more than 1.5×.

Table 12. Energy saved by the choice of key exchange and management protocols.

Energy consumed by session negotiation protocol at the client (milli Joule)		
Key exchange and management protocol	IPSec (Main mode, Revised public key encryption) first SA	IPSec SA
Diffie-Hellman	978	543
RSA	601	208
Energy saving	1.63×	2.61×

Similarly, Table 13 shows that replacing 64-bit 3DES encryption with 128-bit key AES encryption reduces the energy consumed by secure data communication by approximately 1.2×. This is due to the elegant design of AES to better exploit features like pipelining and parallel processing and due to the larger data block size.

Table 13. Energy saved by choice of data encryption mechanism.

Mobile-to-server secure 2560 KB data communication energy (milli Joule)		
	3DES encryption	128-bit key AES encryption
Total transmit energy	38317	32831
Energy savings	-	1.17×
Total receive energy	30640	25154
Energy savings	-	1.22×

4.2.2 Optimizing the Security Association

There exists an inherent trade-off between the security level and the energy consumption of a secure wireless session. Energy consumption increases with the increasing level of security. For example, energy consumed by Diffie-Hellman key management and exchange protocol depends upon the algebraic group used. Size of parameters exchanged between the communicating parties can have substantial impact on the security and energy consumption characteristics of a secure session negotiation protocol.

A client also has a choice of either reducing the encryption key size or the number of encryption rounds while increasing the key refresh rate or vice versa to reduce the system energy while maintaining a desired level of security. Table 14 shows the performance characteristics of AES encryption as a function of the user key size. The tradeoff depends upon the relative energy consumption of the key refreshes and the data encryption mechanism. For example, for a secure session transmitting 2.56 MB data using 3DES encryption, reducing the number of rounds of encryption by 2×, and correspondingly increasing the key refresh rate by 2× reduces the session energy by 1.05×. On the other hand, energy consumed by a secure wireless session using 128-bit key AES encryption is reduced by 1.05× by increasing the encryption key size to 256 bits and reducing the key refresh rate by 2×.

Table 14. Performance characteristics of optimized software implementations of AES encryption as a function of the user key size.

AES Encryption			
	128-bit	192-bit	256-bit
Energy/bit (micro Joule)	0.0666	0.07	0.075
Throughput (Mbps)	25.963	24.58	24.1

4.2.3 Reducing the Energy Consumed by Authentication Protocol

Mutual authentication of the communicating parties during secure session negotiation consumes significant energy. Table 6 shows that public key signature based authentication consumes maximum energy since it entails computation of public-key signatures and exchange of large certificates. Public key encryption based authentication is comparatively energy-efficient due to the absence of the certificate exchanges, even though it requires 2 public-key encryption operations at the client. Revised public key encryption based authentication is almost as energy-inefficient as the public key signature based authentication since it also involves exchange of large certificates. Pre-shared key based authentication scheme consumes least energy but is unsuitable for large-sized networks and mobile operations. An optimized session negotiation protocol based on revised public key encryption is most suitable for wireless environment since it allows the client to send the URL of its certificate to the server instead of the certificate. Therefore, it is energy-efficient, scalable, involves only one public-key

encryption at the client, does not carry out private-key encryption and transmission of the large certificate and uses ephemeral shared secrets (based on the exchanged nonces).

In general, most mobile transactions either do not need user authentication or can use simple password based authentication schemes. With the increasing mobility of wireless devices across heterogeneous wireless networks frequency of session establishment, and hence authentication is increasing. A provision for unidirectional authentication together with simple password based schemes can significantly reduce the energy consumed by a mobile client during secure wireless sessions.

5. SUMMARY

A security protocol offers three levels of security: refreshing the security associations, refreshing the encryption and the MAC keys and varying the length of the secret keys and the number of rounds of encryption. Our proposed schemes affect security protocols at all three levels of security, as shown in Table 15.

Table 15. Improving the secure session energy consumption at various security levels.

Security level	Scheme
Refreshing the secure session	Protocol optimization, Compression
Refreshing the secret keys	Compression
Increasing the length of keys and number of encryption rounds	Hardware acceleration of cryptographic computations

Let us examine the impact of the presented techniques on the energy consumption characteristics of a secure wireless session while transmitting 20 MB data over an 11 Mbps WLAN channel with key refreshes every 128 KB of data and session refreshes every 2 MB of data.

Table 16. Energy savings for the client due to secure session optimization.

	Un-optimized secure session		Optimized secure session	
	Parameter	Energy (milli Joule)	Parameter	Energy (milli Joule)
Handshake	Basic (9)	9558	Var. 1 (1), Var. 2 (1)	335
Compress	-	-	DEFLATE	79225
SHA-256 sign, SHA-256 verify	Software	8828	Hardware	2538
AES-128 encrypt, AES-128 decrypt	Software	53578	Hardware	3082
Transmit	-	105313	-	30278
Receive	-	45336	-	13034
Decompress	-	-	DEFLATE	7988
Key refresh	(150)	1934	(43)	554
Idle system	-	129719		37295
Total transmit	-	308930	-	153307
Transmit energy saving factor				2.02 ×
Total receive	-	248953	-	64826
Receive energy saving factor				3.84 ×

Table 16 compares the energy consumed by an un-optimized scheme using *basic session negotiation* protocol, supporting 3DES encryption and SHA-256 MAC in software and no compression scheme with

the energy consumed by an optimized scheme using *adaptive session negotiation* protocol, supporting 128-bit key AES encryption and SHA-256 MAC in hardware and DEFLATE data compression. We assume a compression ratio of 3.48 corresponding to the data block size of 8 KB. Details of hardware implementation of cryptographic mechanisms can found in [KM02]. Figures in '()' refer to the frequency of the corresponding operations. For example, un-optimized scheme performs 9 session negotiations and 150 key refreshes.

Optimized scheme reduces the energy consumed during data transmission by more than 2× and the energy consumed during data reception by more than 3.8×. Since energy saved due to hardware acceleration is not very significant, adopting a performance, energy consumption and area tradeoff based software-hardware co-design approach is more beneficial.

In this chapter we investigated the energy consumption characteristics of computation-intensive security protocols. We presented techniques to reduce the energy consumed by security protocols during a secure wireless session and used a real-life mobile test bed to demonstrate the energy savings obtained by implementing these techniques. This research can be extended by developing a comprehensive communication security architecture that will integrate these and other accurate energy-models for various components of the security protocols with accurate energy models of other networking components.