SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2012; **5**:1037–1048
Published online 25 October 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.391

RESEARCH ARTICLE

An empirical energy model for secure Web browsing over mobile devices

Sanaa Sharafeddine* and Amal El Arid

Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon

ABSTRACT

Quantifying and modeling energy consumption in mobile devices are essential for developing energy-aware protocols and energy reduction techniques. In this work, we address the energy requirements for secure Web browsing sessions over handheld mobile devices. The contributions of this work are twofold. On one hand, we present a detailed study based on experimental measurements to quantify the energy consumed by the mobile device during secure Web browsing sessions. This includes the energy consumed due to data transmission/reception, encryption/decryption, hashing in addition to browser processing. On the other hand, we derive an empirical energy consumption model for secure Web browsing as a function of various protocol and device parameters. The developed model can be utilized to identify the various components that affect energy consumption during secure Web browsing sessions, to implement application-layer energy models in network simulation tools, and to develop adaptive energy-aware Web browsing protocols. The effectiveness of the developed model is demonstrated via experimental testing on several secure websites. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS
mobile devices; mobile Web browsing; energy modeling; energy consumption measurements; HTTPs; SSL protocol; TLS protocol

***Correspondence**
Sanaa Sharafeddine, Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon.
E-mail: sanaa.sharafeddine@lau.edu.lb

1. INTRODUCTION

Studies demonstrate that the high energy consumption of battery-operated mobile devices is one of the main challenges for emerging mobile computing applications. Mobile devices are shown to consume significant energy because of wireless connectivity activities via their network interfaces (WiFi, cellular, bluetooth, etc.), various data processing tasks, in addition to screen operation. This triggered a wide range of research activities to model energy consumption in mobile devices on the basis of the user behavior [1–3], to analyze energy consumption in mobile devices on the basis of wireless networking protocol design and operation [4–8], and to develop different types of mechanisms to reduce energy consumption [9–13]. It is important to highlight that most existing publications related to energy modeling in mobile devices are based on experimental measurements because of the difficulty of developing analytical models that can accurately capture the operation of the mobile device in relation to the energy drained from the battery.

In this work, we derive two empirical energy models for secure Web browsing over handheld mobile devices on the basis of extensive energy measurements using an experimental test bed. The developed model captures the energy requirements of the Secure Hypertext Transfer Protocol (HTTPS) protocol taking into account HTTP and Transport Layer Security (TLS) exchanged messages. The TLS protocol, which is an enhanced variant of the SSL 3.0 (Secure Sockets Layer 3.0) protocol, is a standard protocol for secure Web transactions over the Internet [14]. We quantify and model the energy requirements in a mobile device due to data transmission/reception, data encryption/decryption/hashing based on different cryptographic algorithms, and browser processing. The importance of the developed model is multifold: It can be used for assessing the additional energy requirements for securing Web transactions, simulating energy consumption for Web browsing applications in network simulation tools, and designing new energy-efficient protocols for secure Web browsing, for example, protocols that can support multiple levels of security from Web servers in real time depending on the actual battery capacity of the mobile device.

Energy consumption studies for Web browsing applications are presented in [15–17]. The authors in [15] proposed a technique for saving energy during Web browsing by switching the network interface to sleep mode when no data is being received. The authors in [16] presented a technique to enhance the performance of the power save mode of

IEEE 802.11 for Web browsing applications. The authors in [17] developed an energy-efficient architecture for Web browsing with cooperation among mobile devices.

Several publications have also considered the performance of security protocols over handheld mobile devices using different measures. The authors in [18] analyze the time taken to perform cryptographic functions for real-time mobile transactions with focus on the SSL, S/MIME and IPsec security protocols. The authors in [19] present a comprehensive analysis of the SSL protocol and its cryptographic algorithms with focus on connection establishment time and data transfer time as performance measures. The authors in [20] analyze the time latency and energy consumption costs of different block ciphers by taking into account encryption and decryption in addition to different file size combinations. The authors in [21] present a comprehensive study on the energy consumption requirements of various cryptographic algorithms by taking into account the trade-offs between energy consumption and the level of security. Moreover, they present an energy analysis for the SSL protocol taking into account various transaction sizes. The authors in [22] focus on designing an energy-efficient mechanism for restarting a secure communication after disruption due to, for example, data loss. The authors in [23] present a comparative performance study of the SSL protocol between mobile devices and laptops with focus on computational complexity and running time as the main performance measures. They present results for each step of the SSL handshaking protocols in order to highlight the existing trade-offs with higher granularity.

This paper is organized as follows. In Section 2, an overview of the HTTPs protocol is presented to highlight key protocol design aspects. In Section 3, a generic energy model for secure Web browsing over mobile devices is derived as a function of various design parameters and HTTPs protocol messages. In Section 4, the implementation of an experimental energy measurement setup is explained and a wide range of energy consumption measurement results are presented and analyzed for data transmission, reception, encryption, decryption, hashing, and browser processing. Moreover, the derived energy model is tested using several secure websites in order to demonstrate its effectiveness. Finally, conclusions are drawn in Section 5.

2. HTTPS PROTOCOL OVERVIEW

The HTTPs protocol is composed of a combination of the HTTP protocol at the application layer and the TLS protocol (or SSL protocol), which runs on top of the transport layer. During secure Web browsing sessions, the operation of HTTPs is divided into two main phases. In the first phase, the TLS handshaking protocols are executed, which include the TLS Handshake protocol, the TLS Change Cipher Spec protocol, and the TLS Alert protocol. In the second phase, the TLS Record Protocol is executed in order to securely exchange encrypted HTTP messages between the mobile device and the Web server.

The main steps of the TLS Handshake Protocol are presented in Figure 1. These steps are mainly used for agreement on a protocol version, exchange of important parameters and keys, generation of shared secrets, selection of a suite of cryptographic algorithms (authentication, encryption, and medium access control (MAC) algorithms), and certificate authentication. Figure 1 includes as well TLS Change Cipher Spec protocol messages, which are used for notification between the mobile device and the server that subsequent messages will be protected on the basis of the negotiated cipher suite. More details on the TLS protocol can be found in [14].

The following are the main steps that take place as part of the TLS handshaking protocols:

- The first step is the connection initiation phase, which involves exchanging hello messages. The hello messages include session identifier, protocol version number, randomly generated numbers, and other information needed to agree on the cipher suite. For example, the cipher suite TLS_RSA_WITH_AES_128_CBC_SHA1 indicates that Rivest–Shamir–Adleman (RSA) algorithm is used for

```
Client                                               Server

ClientHello                 -------->
                                                ServerHello
                                               Certificate*
                                         ServerKeyExchange*
                                        CertificateRequest*
                            <--------       ServerHelloDone
Certificate*
ClientKeyExchange
CertificateVerify*
[ChangeCipherSpec]
Finished                    -------->
                                         [ChangeCipherSpec]
                            <--------              Finished
Application Data            <------->      Application Data
```

Figure 1. Transport Layer Security handshaking protocols. The messages exchanged between the mobile device client and the Web server [14]. The * sign indicates an optional message.

key exchange and authentication, Advanced Encryption Standard (AES) is used in Cipher Block Chaining mode for encrypting application-layer messages with a key size of 128 bits, and SHA-1 is used as a hash function for message integrity verification. The client initiates the secure connection setup by sending the ClientHello message.

- In the second step, the server processes the ClientHello message and responds with the ServerHello message. The server sends its certificate using a ServerKeyExchange message followed by a ServerHelloDone message. These three messages are normally sent together as one TLS record. The protocol also supports optional client authentication; in this case, the client should send its certificate on the basis of a request from the server.
- In the third step, the client uses the information received to authenticate the server. Then, it generates a 48-byte pre-master secret, encrypts it with the server's public key obtained from the server's certificate, and sends it to the server by using a ClientKeyExchange message. The client then sends a ChangeCipherSpec message to inform the server that subsequent messages will be encrypted using the symmetric key; the ChangeCipherSpec message consists of a single byte of value 1. The client ends by sending an encrypted Finished message to indicate that the client part of the handshake phase is finished. The client Finished message is important to verify that the key exchange and authentication processes were successfully set. The ClientKeyExchange, ChangeCipherSpec, and Finished messages are normally sent together as one TLS record.
- In the fourth step, the server uses its private key to decrypt the pre-master secret. The pre-master secret is used by both the client and the server in order to generate the symmetric keys to be used for securing message exchange in the TLS Record Protocol. The server sends a ChangeCipherSpec message to inform the client that subsequent messages will be encrypted using the symmetric key. It then sends a final encrypted server Finished message to indicate that the server part of the handshake phase is finished. The ChangeCipherSpec and Finished messages are normally sent together as one TLS record.
- After the handshake phase is finished, the secure exchange of HTTP application messages can be initiated using the TLS Record Protocol. At the sender side, the TLS Record Protocol fragments the application-layer data into TLS records of size up to 16 384 bytes each, applies a MAC using a hashing function, encrypts the obtained fragment using a symmetric algorithm, adds a 5-byte record layer header (1-byte record type, 2-byte protocol version, and 2-byte length of data in the record), and hands the encrypted fragment to the transport layer for further processing and transmission. It is worth noting that client message boundaries are not preserved in the TLS Record Protocol; that is, multiple client messages may be combined into a single record, or a single message may be fragmented into multiple records. At the receiver side, the TLS Record Protocol extracts the record layer header, decrypts the received fragment, verifies the MAC, reassembles decrypted fragments if applicable, and delivers messages to the HTTP application layer.

3. SECURE WEB BROWSING: ENERGY MODEL DERIVATION

In this section, we derive an energy model for secure Web browsing over mobile devices. We identify the main components that contribute to energy consumption in a mobile device during secure web browsing sessions. These include the energy drained from the battery during the TLS handshaking phase (TLS Handshake Protocol and TLS Change Cipher Spec Protocol operation) and the secure Web data exchange phase (TLS Record Protocol operation). We divide these main components into two classes: energy drained from the battery as a result of communications activities (data transmission and reception) and energy drained from the battery as a result of processing activities (data encryption, decryption, hashing, etc.).

The total energy consumption E_{Total} in a mobile device during a secure Web browsing session can be expressed as follows:

$$E_{\text{Total}} = E_{\text{Handshake}} + E_{\text{Web}} \tag{1}$$

where $E_{\text{Handshake}}$ is the energy consumed during the TLS handshaking phase and E_{Web} is the energy consumed during the HTTP application-layer data exchange phase.

3.1. Transport Layer Security Handshaking Phase

The energy consumed during the handshaking phase can be decomposed as follows:

$$E_{\text{Handshake}} = E_{\text{H-Exchange}} + E_{\text{H-Proc}} \tag{2}$$

$E_{\text{H-Exchange}}$ captures the energy consumed due to communications activities (data transmission and reception) during the handshaking phase. It can be modeled as follows:

The following are the definitions of the parameters in Equation (3):

- E_{T} is the energy consumed per byte during data transmission, E_{R} is the energy consumed per byte during data reception, and E_{I} is the energy consumed per second during idle mode (device active with wireless interface on, but not performing any processing or networking activities).
- B_{H1} is the size of the ClientHello message in bytes.
- B_{H2} is the total size of the ClientKeyExchange, client ChangeCipherSpec, and client Finished messages because they are normally sent together.

DOI: 10.1002/sec

$$\begin{aligned} E_{\text{H-Exchange}} = {} & E_{\text{T}} \cdot \Big(\underbrace{B_{\text{ClientHello}} +}_{B_{\text{H1}}} \underbrace{B_{\text{ClientKeyExchange}} + B_{\text{ChangeCipherSpec}} + B_{\text{Finished}}}_{B_{\text{H2}}} \Big) \\ & + E_{\text{R}} \cdot \Big(\underbrace{B_{\text{ServerHello}} + B_{\text{ServerKeyExchange}} + B_{\text{ServerHelloDone}}}_{B_{\text{H3}}} + \underbrace{B_{\text{ChangeCipherSpec}} + B_{\text{Finished}}}_{B_{\text{H4}}} \Big) \\ & + E_{\text{I}} \cdot \tau_{\text{Handshake}} + E_{\text{T}} \cdot N_{\text{H-Exchange,T}} \cdot B_{\text{TCP/IP}} + E_{\text{R}} \cdot N_{\text{H-Exchange,R}} \cdot B_{\text{TCP/IP}} \\ & + E_{\text{TCP-H}} \end{aligned} \tag{3}$$

- B_{H3} is the total size of the ServerHello, ServerKeyExchange (includes server certificate), and ServerHelloDone messages because they are normally sent together.
- B_{H4} is the total size of the server ChangeCipherSpec and server Finished messages because they are normally sent together.
- $\tau_{\text{Handshake}}$ is the total time of the handshaking phase. Multiplying $\tau_{\text{Handshake}}$ by E_{I} gives the total idle energy consumption during the handshaking phase.
- $N_{\text{H-Exchange,T}}$ is the total number of handshaking protocol packets transmitted from the mobile device during the handshaking phase. This is normally equal to two because B_{H1} and B_{H2} are smaller than the maximum transfer unit (MTU) size and, thus, fit in one link layer frame each.
- $N_{\text{H-Exchange,R}}$ is the total number of handshaking protocol packets received by the mobile device. It can be modeled as follows: $N_{\text{H-Exchange,R}} = 1 + \lceil B_{\text{H3}}/B_{\text{MTU}} \rceil$ because the size of B_{H4} fits within one link layer frame, whereas the size of B_{H3} is relatively large (it contains the server certificate size) and, thus, might require multiple packets. The MTU size is denoted as B_{MTU} with a typical value of 1500 bytes assuming an Ethernet data link layer protocol.
- $B_{\text{TCP/IP}}$ is the total size of the headers added by the transport (TCP), network (IP), and data link (Ethernet) layers. Typical value of $B_{\text{TCP/IP}}$ is 58 bytes (20-byte TCP header, 20-byte IP header, and 18-byte Ethernet header).
- $E_{\text{TCP-H}}$ is the energy consumed due to TCP operation during the handshaking phase. This includes transmission and reception of TCP segments as part of the TCP connection setup phase. Normally, the setup phase requires the exchange of three segments: transmit SYN, receive SYN–ACK, and transmit ACK. Moreover, $E_{\text{TCP-H}}$ includes the transmission of TCP ACK segments to acknowledge the reception of handshake messages from the server, especially as a result of the reception of the server certificate, which is normally divided over multiple packets depending on its size.

$E_{\text{H-Proc}}$ captures the energy consumed as a result of the processing activities in the mobile device during the TLS handshaking phase. It can be modeled as follows:

$$E_{\text{H-Proc}} = E_{\text{KeyGen}} + E_{\text{CertVerify}} + E_{\text{Finished}} \tag{4}$$

where E_{KeyGen} is the energy consumed as a result of key generation-processing activities including pre-master secret encryption using RSA, $E_{\text{CertVerify}}$ is the energy consumed as a result of processing activities to verify the server certificate, and E_{Finished} is the energy consumed as a result of the client Finished message hashing and encryption before transmission in addition to the received server Finished message decryption and MAC verification.

It is important to note that it is common to repeat the handshaking phase for each secure object in the website. The repeated handshaking can be either in full mode or in an abbreviated mode to resume a TLS session (also called restart handshake). In case of a resumed session, only hello, change cipher spec, and finished messages are exchanged, which avoids the processing overhead of key generation and RSA algorithm execution. The model presented in this section can be easily customized to cover the different modes of handshaking (full or abbreviated) and can be scaled by the number of times that the handshaking phase is repeated during a given Web browsing session.

3.2. Secure Web data exchange phase

After the handshaking phase ends, the mobile device starts exchanging encrypted HTTP messages with the Web server. These include sending encrypted HTTP request messages (GET messages) and receiving Web objects. During secure Web browsing sessions, not all Web objects need to be encrypted in order to reduce the communications and processing overhead. For example, confidential information should be sent encrypted, whereas some images or general website content can be sent plain without any encryption. The energy consumed during the Web data exchange phase can also be decomposed into two components as follows:

$$E_{\text{Web}} = E_{\text{W-Exchange}} + E_{\text{W-Proc}} \tag{5}$$

$E_{\text{W-Exchange}}$ captures the energy consumed as a result of communications activities for exchanging HTTP protocol messages. It can be modeled as follows:

$$
\begin{aligned}
E_{\text{W-Exchange}} = & \sum_{i=1}^{N_{\text{ns}}} (E_{\text{T}} \cdot (B_{\text{GET},i} + B_{\text{TCP/IP}}) \\
& + E_{\text{R}} \cdot (B_{\text{Data},i} + \lceil B_{\text{Data},i}/B_{\text{MTU}} \rceil \cdot B_{\text{TCP/IP}})) \\
& + \sum_{j=1}^{N_s} (E_{\text{T}} \cdot (B_{\text{GET},j} + B_{\text{TLS}} + B_{\text{TCP/IP}}) \\
& + E_{\text{R}} \cdot (B_{\text{Data},j} + \lceil B_{\text{Data},i}/B_{\text{MTU}} \rceil \cdot B_{\text{TCP/IP}} \\
& + \lceil \text{B}_{\text{Data},i}/B_{\text{RecordMax}} \rceil \cdot B_{\text{TLS}}) + E_{\text{I}} \cdot \tau_{\text{Web}} + E_{\text{TCP-W}}
\end{aligned} \tag{6}
$$

The following are the definitions of the parameters in Equation (6):

- N_{ns} is the number of non-secure Web objects, and N_{s} is the number of secure Web objects.
- $B_{\text{GET},i}$ is the size of the HTTP GET message for the ith object and $B_{\text{Data},i}$ is the size of the ith Web object. Typically, the size of the GET message is smaller than the MTU size, and thus, only one $B_{\text{TCP/IP}}$ term is added to it to compensate for lower-layer headers' overhead.
- $B_{\text{TLS}} = B_{\text{MAC}} + B_{\text{RecordHeader}}$ where B_{MAC} is the size of the MAC added by the TLS hashing algorithm (typical size: 20 bytes for SHA-1 algorithm), and $B_{\text{RecordHeader}}$ is the size of the TLS record header (size: 5 bytes). It is important to note that this term is added only for the secure objects because non-secure objects are not processed via the TLS Record Protocol.
- $B_{\text{RecordMax}}$ is set to 16 384 bytes because the TLS Record Protocol fragments HTTP messages into records of size up to 2^{14} bytes.
- τ_{Web} is the total time of the Web data exchange phase. Multiplying τ_{Web} by E_{I} gives the total idle energy consumption during the Web data exchange phase.
- $E_{\text{TCP-W}}$ is the energy consumed as a result of TCP operation during the Web data exchange phase. This includes the transmission of TCP ACK segments to acknowledge the reception of HTTP response packets. Moreover, it includes transmission and reception of TCP segments as part of the TCP connection termination phase. Normally, the termination phase requires the exchange of three segments: transmit FIN, receive FIN–ACK, and transmit ACK.

$E_{\text{W-Proc}}$ captures the energy consumed as a result of processing activities for exchanging HTTP protocol messages. It can be modeled as follows:

$$
\begin{aligned}
E_{\text{W-Proc}} = & \sum_{j=1}^{N_{\text{s}}} (E_{\text{H}} \cdot B_{\text{GET},j} + E_{\text{E}} \cdot (B_{\text{GET},j} + B_{\text{MAC}}) \\
& + E_{\text{D}} \cdot (B_{\text{Data},j} + \lceil B_{\text{Data},j}/B_{\text{RecordMax}} \rceil \cdot B_{\text{MAC}}) \\
& + E_{\text{H}} \cdot B_{\text{Data},j}) + E_{\text{Browser}}
\end{aligned} \tag{7}
$$

where E_{H} is the energy consumed per byte as a result of hashing processing operations, E_{E} is the energy consumed per byte as a result of data encryption operations, E_{D} is the energy consumed per byte as a result of data decryption operations, and E_{Browser} is the energy consumed by the Web browser to render and display the website. It is important to note that hashing is applied to the GET messages before encryption; for this reason, the MAC size is added to the GET message size before multiplying by E_{E}. For received data, the decryption is applied for a data size that corresponds to the Web object's total size in addition to multiple MACs depending on the number of records the Web object was fragmented into.

The energy model derived in this section applies to secure and non-secure Web browsing sessions. For typical HTTP Web browsing without TLS, the derived model can be used by setting $E_{\text{Handshake}} = 0$ and setting $N_{\text{s}} = 0$ in Equations (6) and (7). The derived model is generic as it applies to Web browsing sessions with mixed secure/non-secure objects and to various hashing and encryption algorithms. It is particularly suitable when the energy consumed as a result of transmission/reception and as a result of cryptographic algorithms execution has a linear relation with respect to the payload data size.

3.3. Secure Web browsing: simplified energy model

In this section, we present a simplified model for estimating energy consumption for secure Web browsing sessions. This model is based on the number of HTTP messages encrypted/decrypted and the number of packets transmitted/received. This model is particularly applicable when the energy consumed during encryption/decryption/hashing of a message does not vary notably for typical message sizes (maximum size of TLS record is around 15 kB), and the energy consumed during transmission/reception of a packet does not vary notably for typical packet sizes (maximum size of Ethernet packets is around 1500 bytes). In this case, the energy would increase linearly with the number of messages encrypted/decrypted and the number of packets transmitted/received. These conditions are shown to be valid in real scenarios on the basis of the experimental measurement results presented in Section 4.

The total energy consumed as a result of communications activities for exchanging TLS handshaking messages and secure Web data packets can be simplified as follows:

$$
\begin{aligned}
E_{\text{Exchange-S}} = & E_{\text{T-Packet}} (N_{\text{H-Exchange,T}} + N_{\text{W-Exchange,T}} + N_{\text{TCP,T}}) \\
& + E_{\text{R-Packet}} (N_{\text{H-Exchange,R}} + N_{\text{W-Exchange,R}} + N_{\text{TCP,R}})
\end{aligned} \tag{8}
$$

where $E_{\text{T-Packet}}$ is the energy consumed during transmission of a TCP/IP packet, $E_{\text{R-Packet}}$ is the energy consumed during reception of a TCP/IP packet, $N_{\text{H-Exchange,T}}$ is the total number of TLS protocol packets transmitted during the handshaking phase (typical value: two), $N_{\text{H-Exchange,R}}$ is the total number of TLS protocol packets received during the handshaking phase (typical value: one plus the number of packets for the server certificate, which can be calculated as the size of the certificate divided by 1500 bytes), $N_{\text{W-Exchange,T}}$ is the total number of transmitted packets carrying HTTP GET messages (typical value: number of objects), $N_{\text{W-Exchange,R}}$ is the total number of received Web data packets (typical value: sum of the division of the objects' sizes by 1500 bytes), $N_{\text{TCP,T}}$ is the total number of transmitted control TCP segments during the TLS handshaking and Web exchange phases (typically includes ACKs, SYN, and FIN segments), $N_{\text{H-TCP,R}}$ is the total number of received control TCP segments during the TLS handshaking and Web exchange phases (typically includes ACKs, SYN, and FIN segments).

The total energy consumed as a result of processing activities during the TLS handshaking and Web data exchange phases can be simplified as follows:

$$\begin{aligned} E_{\text{Proc-S}} = {} & (E_{\text{E-Record}} + E_{\text{H-Record}})N_{\text{Record,T}} \\ & + (E_{\text{H-Record}} + E_{\text{D-Record}})N_{\text{Record,R}} + E_{\text{H-Proc}} \\ & + E_{\text{Browser}} \end{aligned} \tag{9}$$

where $E_{\text{E-Record}}$ is the energy consumed during encryption of a TLS record message (maximum size of a TLS record is 16 384 bytes), $E_{\text{H-Record}}$ is the energy consumed during hashing of a TLS record message, $E_{\text{D-Record}}$ is the energy consumed during decryption of a TLS record message, $N_{\text{Record,T}}$ is the total number of transmitted TLS record messages (typical value: number of secure objects where each record message encapsulates an HTTP GET message), $N_{\text{Record,R}}$ is the total number of received TLS record messages (typical value: sum of the division of the secure objects' sizes by 16 384 bytes), $E_{\text{H-Proc}}$ is the processing energy during the handshaking phase as modeled in Equation (4), and E_{Browser} is the energy consumed by the browser to render and display the website objects.

4. ENERGY CONSUMPTION MEASUREMENTS AND ANALYSIS

4.1. Energy measurement setup

An experimental measurement setup is built in order to capture the energy consumption from the battery of a mobile device using real-time energy profiling. The setup is composed of four main parts as shown in Figure 2: a mobile device (HP iPAQh6340 [24]), a WiFi access point (IEEE 802.11a), a data acquisition (DAQ) device (NI USB-6008 [25]), and a processing module on a laptop programmed using LabVIEW.

The data acquisition device measures the voltage across a resistor with known resistance wired in series between the mobile device and the battery. The measured voltage is exported to a LabVIEW application, which performs power calculations and plots power consumption profiles in real time. All exchanged packets during the Web browsing session are captured via a packet sniffer (Wireshark [26]). The traces captured are important for analyzing the website content for testing and verification purposes.

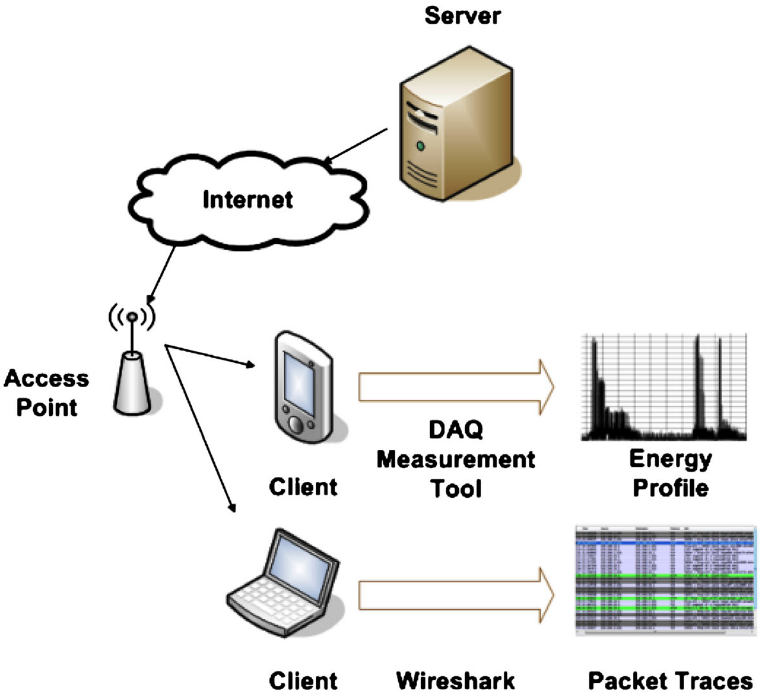


Figure 2. Experimental setup for measuring energy consumption in a mobile device.

DOI: 10.1002/sec

The energy measurement setup is used to analyze the energy consumed by the mobile device during the execution of various cryptographic algorithms. The algorithms are implemented in the .Net Framework under the namespace System.Security.Cryptography.Pkcs, which contains the different needed classes. Moreover, the setup is used to derive empirical expressions to model the energy consumed by the mobile device during data transmission and data reception via its wireless interface.

4.2. Energy consumption due to communications activities

In this section, we derive empirical expressions for the following two parameters: the energy E_{T} consumed by the mobile device per byte during data transmission and the energy E_{R} consumed by the mobile device per byte during data reception. We perform energy measurements for various data sizes ranging from 50 to 1500 bytes (maximum packet size assuming Ethernet data link layer); for each data size, we conduct 50 measurement runs in order to obtain average results. Figure 3 presents typical power consumption results during the different communications states of a mobile device: idle mode, transmit mode, and receive mode. It can be seen that the device consumes more energy when transmitting data compared with receiving data. Averaging the captured power consumption level over a given window of time gives the amount of energy consumed from the battery during that time.

In Figure 4, energy consumption results for E_{T} and E_{R} are presented as a function of the transmitted and received payload data sizes in bytes, respectively. It is worth noting that the total energy consumption during data transmission and reception is shown to increase at a slow rate as the payload data size increases between 50 and 1500 bytes.

The following empirical expressions are derived using curve fitting on the basis of the obtained measurement results:

$$E_{\mathrm{T}} = 1.665e^{-0.022B_{\mathrm{T}}} + 0.262e^{-0.002B_{\mathrm{T}}} \tag{10}$$

$$E_{\mathrm{R}} = 0.438e^{-0.017B_{\mathrm{R}}} + 0.078e^{-0.0016B_{\mathrm{R}}} \tag{11}$$

where B_{T} and B_{R} are the transmitted and received data sizes in bytes, respectively. The accuracy of the curve fitting exceeds 95% in both cases. As the data size increases, the energy consumed per byte is shown to decrease;

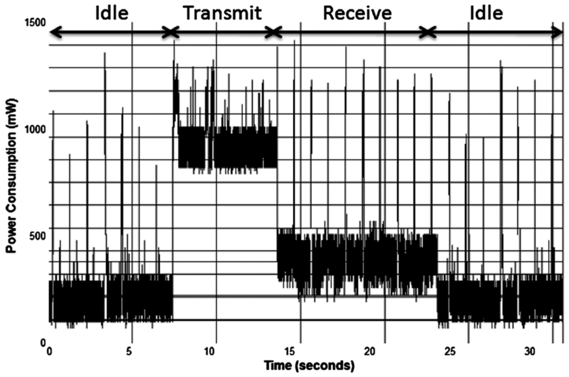
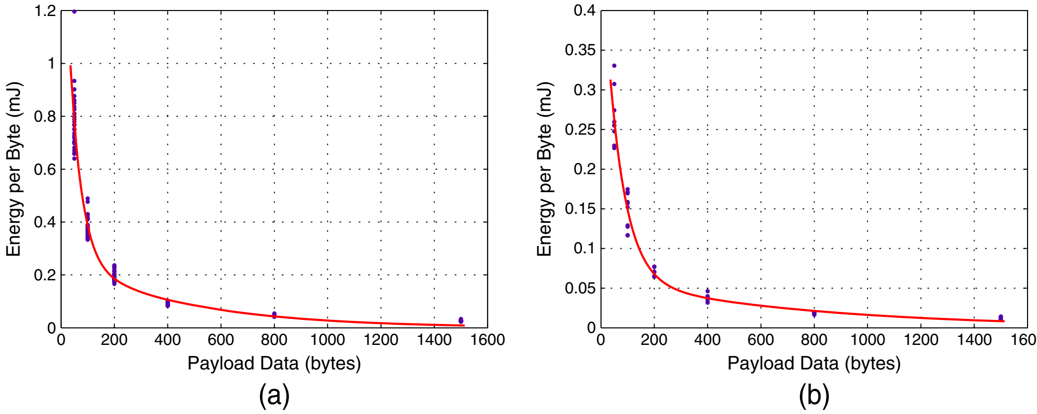


Figure 3. Typical energy measurement results during the following modes of operation: idle, transmit, and receive.



(a)

(b)

Figure 4. (a) Energy consumption per byte E_{T} in mJ during data transmission. (b) Energy consumption per byte E_{R} in mJ during data reception. Dots correspond to measurement points. Solid lines correspond to the outcome of curve fitting.

however, the total energy consumed slightly increases. Similar trends have been also reported in other publications; for example, see [7]. Moreover, the energy consumed per byte during data transmission is notably more than that during data reception. For example, a payload data of 400 bytes consumes around 0.11 mJ per byte to be transmitted and around 0.04 mJ per byte to be received. During data transmission, the mobile device consumes energy for both data processing and transmission power so that the signal arrives at the access point with high enough signal-to-noise ratio. However, during data reception, the mobile device consumes energy only for data processing.

4.3. Energy consumption due to processing activities

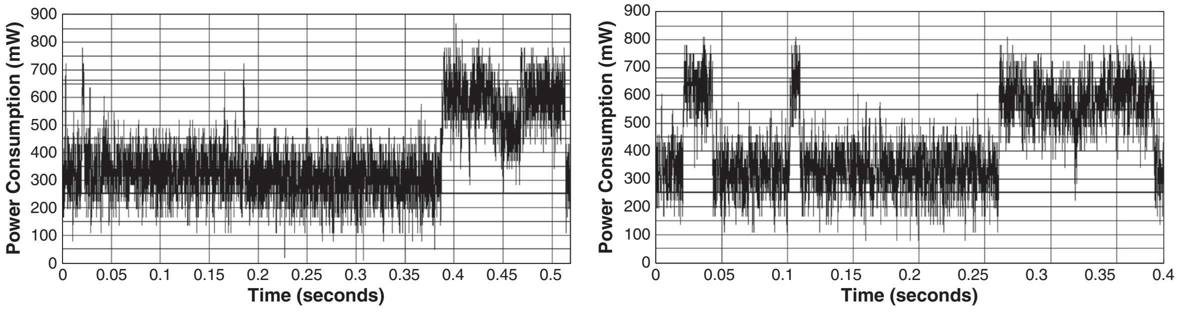
In this section, we present experimental energy measurements for various cryptographic algorithms related to the execution of the TLS protocol in the mobile device. These include the public key encryption algorithm RSA, symmetric key encryption algorithm AES, in addition to the hashing algorithms MD5 and SHA-1. Measurements are performed for various payload data sizes ranging from 250 to 50 000 bytes; for each data size, we conduct several measurement runs in order to obtain average results.

Figure 5 presents sample encryption and decryption energy profile results using the AES algorithm with a key size of 128 bits and data size of 1000 bytes. The peaky intervals in the figure correspond to energy consumption above the idle level as a result of encrypt or decrypt processing activities; the overall energy consumed depends on the algorithm's complexity and computation time. AES encryption as shown in the figure needs more computation time as compared with decryption and, thus, consumes higher energy: 497 mJ and 465 mJ are needed to encrypt and decrypt a given 1000-byte message, respectively. The recorded idle energy during these measurements is around 300 mJ.

Table I presents measurement results for various cryptographic algorithms as a function of the input payload size in bytes. It is shown that the energy consumed per input data block increases slowly as the payload size increases. Moreover, AES with a 256-bit key size consumes slightly more energy that AES with a 128-bit key size. In addition, RSA has notably higher energy consumption compared with AES because of the increased computational complexity of public key encryption algorithms compared with symmetric key algorithms. Finally, it can be noted that both hashing algorithms MD5 and SHA-1 have similar energy consumption for the different payload sizes.

Finally, we performed a wide range of measurements to model the processing energy consumed by the Web browser (Internet Explorer) in order to render and display websites of different sizes. The following empirical expression was derived for E_{mBrowser} in mJ as function of the website size S in kilobytes on the basis of the obtained measurement results with fitting accuracy over 95%:

$$E_{\mathrm{Browser}} = 0.56S + 4.14 \tag{12}$$



(a) (b)

Figure 5. (a) Energy profile for encrypting a 1000-byte message using the Advanced Encryption Standard (AES) algorithm with a key size of 128 bits (time duration: 0.52 s, total energy consumption: 497 mJ). (b) Energy profile for decrypting a 1000-byte message using the AES algorithm with a key size of 128 bits (time duration: 0.4 s, energy consumption: 465 mJ).

Table I. Energy consumption for various cryptographic algorithms in mJ as a function of input data size.

Algorithm	250 bytes (mJ)	500 bytes (mJ)	1000 bytes (mJ)	5000 bytes (mJ)	10 000 bytes (mJ)
AES (128-bit key)	36.71	38.03	45.21	49.33	55.06
AES (256-bit key)	38.04	44.09	46.49	53.19	55.07
RSA (1024-bit key)	56.05	56.55	62.11	64.85	67.11
SHA-1	32.21	33.46	34.52	36.22	39.68
MD5	30.11	33.13	34.31	37.59	39.85

AES, Advanced Encryption Standard; RSA, Rivest–Shamir–Adleman algorithm; SHA, Secure Hash Algorithm; MD, Message Digest.

Table II presents measurement results for the energy consumed as a result of the processing activities during the TLS handshaking phase. These results provide typical values for the different terms in Equation (4).

4.4. Energy consumption during secure Web browsing sessions

In this section, we analyze the energy consumption profiles during secure Web browsing sessions for different websites. Moreover, we demonstrate the effectiveness of the derived simplified energy model in Section 3 by comparing the estimated energy consumption results with measurement based results. Figure 6 presents energy profile results for downloading four secure websites having different sizes and number of objects.

Table II. Processing energy consumption during the Transport Layer Security handshaking phase

Parameter	Typical energy consumption (mJ)
E_{KeyGen}	490
$E_{\mathrm{CertVerify}}$	111
E_{Finished}	108

The results demonstrate interesting observations related to the energy consumption behavior during secure Web browsing sessions. It can be seen that the download duration plays a key role in terms of the overall energy consumption. Therefore, having higher-speed broadband Internet connection leads to significant reduction in energy consumption mainly because of reduced communications interface activity for data transmission and reception. It is shown that websites with more objects have more peaky intervals with higher density per interval. One can roughly deduce the number of objects in the website by observing the real-time energy consumption profile. For example, Figure 6a, 6b, and 6c corresponds to secure websites with 2, 4, and 11 objects, respectively. It is important to highlight that the energy consumption for the Cellular Operator Log In website presented in Figure 6d is significantly higher than the other cases because of the notably larger download duration and the higher number of objects, which leads to more HTTP GET requests, TCP control messages, received HTTP Web data packets, and TLS records to be encrypted, decrypted, and hashed.

We have also analyzed the content of a wide range of secure websites in order to capture common implementation characteristics in terms of the used cipher suites, the number of times the handshaking phase is executed, typical sizes of HTTP GET messages, typical sizes of server

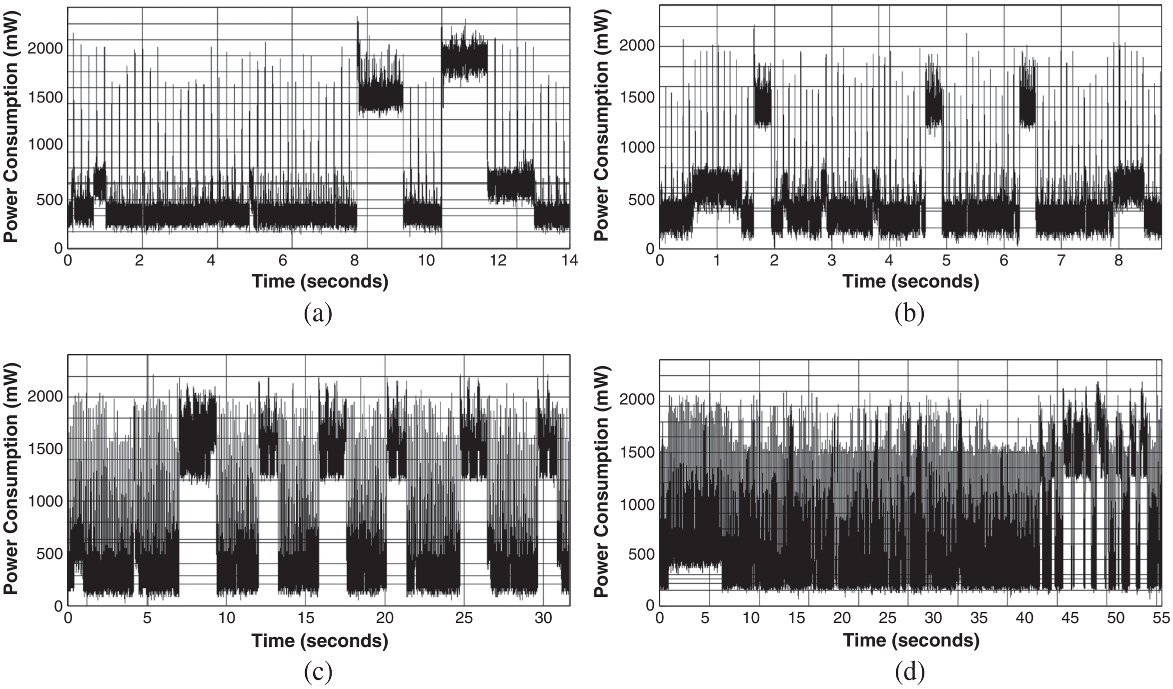


Figure 6. Example energy profiles for downloading four different secure websites (a) Microsoft Online secure website (2 secure objects, website size 4000 bytes, download duration 13.1 s, total energy consumption 5150 mJ), (b) University Email Portal (4 secure objects, website size 5184 bytes, download duration 8.75 s, total energy consumption 6788 mJ), (c) IEEE Sign In (11 secure objects, website size 14 489 bytes, download duration 31.5 s, total energy consumption 23 307 mJ), (d) Cellular Operator Log In (62 secure objects, website size 227 421 bytes, download duration 55 s, total energy consumption 63 034 mJ).

DOI: 10.1002/sec

certificates, and so on. The obtained results are summarized in Table III, which includes the sizes of various messages that are exchanged between the mobile device and the Web server during secure Web browsing sessions. These statistics were collected from 13 secure websites containing a total of 131 secure objects. The list of the tested websites include institution e-mail portals, IEEE secure pages (log in, shop me), Gmail, encrypted Google, Microsoft Download, Amazon, eBay, Yahoo, and others. The direction field indicates whether the message is sent from the mobile client to the Web server (C→S) or from the Web server to the mobile client (S→C).

Table IV presents the obtained typical sizes in bytes for TCP connection management control messages in addition to typical sizes for the HTTP GET message.

Finally, we analyze the accuracy of the derived simplified empirical energy model based on four secure Web browsing sessions, having different characteristics, using the following procedure:

(1) Download a secure website on the mobile device.
(2) Capture the energy consumption profile using the measurement setup. Calculate the total amount of energy consumed and the total download duration during the Web browsing session on the basis of the measured profile.
(3) Capture all the exchanged packets and extract the needed parameters to apply the model; these include the number of Web objects and their sizes, the number of times the handshaking phase is executed, the TLS cipher suite, the total website size, and the total number of transmitted/received packets including TCP connection management and ACK segments.
(4) Apply the derived model to obtain an estimate of the total energy consumed using the extracted website parameters as inputs.
(5) Compare the obtained estimated energy consumption level with the measured value and calculate the percentage difference.

Table V presents a summary of the following main parameters for each tested website: the total website size in bytes, the number of secure objects (the selected websites have all their objects secured), the number of times that the TLS handshaking phase is executed, the total download duration, the average GET message size, the total number of transmitted packets (including TLS handshaking phase transmitted messages, GET requests, in addition to TCP connection management and ACK segments), and the total number of received packets (including TLS handshaking phase received messages, Web data packets, in addition to TCP connection management and ACK segments).

Results show that it is common to execute the handshaking phase in abbreviated mode for each secure object in the website. Moreover, cipher suites based on AES (128 bit or 256 bit) and RC4 encryption with either MD5 or SHA-1 hashing are widely used.The size of the GET message varies between different websites depending on the length of the object's path name and the number of included header lines. The overall numbers of transmitted and received packets are comparable because of the TCP ACKs and are higher for websites containing more objects; however, the total size in bytes of the received packets is normally much more than the transmitted packets. It is important to note that the download duration is not linearly

Table III. Typical sizes in bytes of various messages that are exchanged during secure Web browsing sessions.

Message	Direction	Median (bytes)	Minimum (bytes)	Maximum (bytes)
ClientHello	C→S	157	124	165
ClientKeyExchange	C→S	134	134	262
Client ChangeCipherSpec	C→S	1	1	1
Client Finished	C→S	36	32	48
ServerHello	S→C	74	42	85
ServerKeyExchange (with certificate)	S→C	2872	1425	5188
ServerHelloDone	S→C	4	4	4
Server ChangeCipherSpec	S→C	1	1	1
Server Finished	S→C	36	32	48

Table IV. Sizes of different messages that are transmitted and received during Web browsing sessions.

Message type	Size (bytes)	Message type	Size (bytes)
TCP SYN	66	TCP SYN–ACK	66
TCP FIN	54	TCP FIN–ACK	54
HTTP GET message	500–2500	MTU size	1500

Table V. Website characteristics

Website	Total size (bytes)	Objects	Handshake	Duration (s)	GET size (bytes)	Tx packets	Rx packets
Microsoft Download	4000	2	2	13.10	2368	14	13
Email Portal	5184	4	4	8.75	760	34	47
IEEE Sign In	14 489	11	11	31.50	1920	104	137
Cellular Operator	227 421	62	62	55.00	629	548	588

Table VI. Energy consumption comparison: estimated results versus measurement results.

Website	Profile energy (mJ)	Model Energy (mJ)	Comm. Energy (mJ)	Proc. Energy (mJ)	Idle Energy (mJ)	Difference (%)
Microsoft Download	5150	5610	768	1021	3821	8.95
Email Portal	6788	7031	2112	1335	3584	3.55
IEEE Sign In	23 307	20 056	6352	2821	10 883	13.95
Cellular Operator	63 034	56 462	31 328	8397	16 737	10.43

proportional to the website's total size because it depends as well on the network connection conditions and the route between the client and the server for a given Web browsing session.

Table VI presents a comparison between the measured total energy consumption and the estimated total energy consumption using the proposed simplified energy model.

The following parameters are presented for each tested website: the total energy consumption based on the measured energy profile, the estimated total energy consumption based on the proposed model, the portion of the estimated total energy consumption due to communications activities (data transmissin/reception), the portion of the estimated total energy consumption due to processing activities (TLS cryptographic algorithms and browser operation), and the portion of the estimated total energy consumption that corresponds to idle operation of the device (without any Web browsing or other processing activities). Summing the values for the communications energy component (Comm. Energy), processing energy component (Proc. Energy), and idle energy component (Idle Energy) gives the total estimated energy (Model Energy).

The results demonstrate a high level of accuracy for the proposed model especially that testing was performed on secure Web browsing sessions during real network conditions. The difference between the estimated values and the measured values ranged between 3.55% and 13.95%. The derived model can be used to break down the total energy consumption into different components. Results show that both communications activities and cryptographic processing activities contribute notably to the total energy consumption during secure Web browsing sessions. For example, the energy consumed, as a result of the TLS handshaking messages and cryptographic algorithms execution, exceeds the energy consumed as a result of the exchange of the HTTP messages for the Email Portal website. This demonstrates the notable overhead of Web security especially for websites that are relatively small in terms of total size and number of objects.

5. CONCLUSIONS

We addressed the problem of energy efficiency in handheld mobile devices for secure Web browsing applications. We developed a generic empirical energy model that captures the different energy consumption components during secure Web browsing sessions including communications activities (transmission and reception), cryptographic algorithms (encryption, decryption, and hashing), and device idle mode operation. The derived model captures the key design aspects of the HTTPs protocol. Moreover, we presented a simplified energy model that requires a lower number of parameters and that is particularly applicable to scenarios where the energy consumed does not vary much as the payload data size varies within practical ranges. In order to gain an in-depth insight on the energy requirements of HTTPs and in order to validate the accuracy of the derived model, we presented and analyzed a wide range of energy measurement results that capture both communications activities and cryptographic algorithms execution. Finally, we studied the energy profiles of different selected websites, and we demonstrated the effectiveness of the proposed model by comparing estimated energy results to experimentally measured results.

The derived model can be used to identify the main factors that affect the energy consumption and to quantify their contributions, as a function of various protocol and device parameters during secure Web browsing sessions. This insight can then be utilized to develop energy-aware protocols for secure Web browsing applications. For example, the proposed model can be easily used to assess the impact of the following energy reduction approaches: reducing the number of TCP ACKS, reducing the number of times the TLS handshaking phase is executed, reducing the number of secure objects in the website, using alternative cipher suites, using alternative implementations of specific cryptographic algorithms, using cooperative communications architectures to increase the transmission rate and reduce the download time, and so on.

REFERENCES

1. Shye A, Sholbrock B, Memik G. Into the wild: studying real user activity patterns to guide power optimization for mobile architectures. In *MICRO-42*, December 2009.
2. Falaki H, Mahajan R, Kandula S, Lymberopoulos D, Govindan R, Estrin D. Diversity in smartphone usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, June 2010.
3. Vallina-Rodriguez N, Hui P, Crowcroft J, Rice A. Exhausting battery statistics: understanding the energy demands on mobile handsets. In *Proceedings of the second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld'10)*, August 2010.
4. Feeney L, Nilsson M. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings of IEEE INFOCOM 2001*, April 2001.
5. Mahmud K, Inoue M, Murakami H, Hasegawa M, Morikawa H. Energy consumption measurement of wireless interfaces in multi-service user terminals for heterogeneous wireless networks. *IEICE Transaction Communications* 2005; **E88-B**(3): 1097–1110.
6. Ergen M, Varaiya P. Decompostion of energy consumption in IEEE 802.11. In *Proceedings of the IEEE International Conference on Communication ICC '07*, June 2007.
7. Pedersen M, Perrucci G, Fitzek F, Larsen T. Energy and link measurements for mobile phones using IEEE802.11b/g. In *Proceedings of the 4th International Workshop on Wireless Network Measurements (WiNMEE 2008)—in conjunction with WiOpt 2008*, April 2008.
8. Rice A, Hay S. Decomposing power measurements for mobile devices. *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, April 2010.
9. Havinga P, Smit G. Energy-efficient wireless networking for multimedia applications. *Journal of Wireless Applications and Mobile Computing* 2001; **1**(2): 165–184.
10. Shih E, Bahl P, Sinclair M. Wake on wireless: an event driven energy saving strategy for battery operated devices. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM)*, September 2002.
11. Mayo R, Ranganathan P. Energy consumption in mobile devices: why future systems need requirements-aware energy scale-down. *Lecture Notes in Computer Science, Power-Aware Computer Systems* 2003; **3164**: 26–40.
12. Balasubramanian N, Balasubramanian A, Venkataramani A. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of ACM SIGCOMM Internet Measurement Conference (IMC)*, November 2009.
13. Sharafeddine S, Madah R. A lightweight adaptive compression scheme for energy-efficient mobile-to-mobile file sharing applications. *Journal of Network and Computer Applications* 2011; **34**(1): 52–61.
14. Dierks T, Rescorla E. The Transport Layer Security (TLS) Protocol, Version 1.2. *RFC 5246*, August 2008.
15. Yan H, Krishnan R, Watterson S, Lowenthal D. Client-centered energy saving for concurrent HTTP connections. In *Proceedings of the 14th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, June 2004.
16. Krashinsky R, Balakrishnan H. Minimizing energy for wireless Web access with bounded slowdown. *Wireless Networks* 2005; **11**(1–2): 135–148.
17. Perrucci G, Fitzek F, Zhang Q, Katz M. Cooperative mobile web browsing. *EURASIP Journal on Wireless Communications and Networking* 2009; **2009**(7): 1–9.
18. Argyroudis PG, Verma R, Tewari H, O'Mahony D. Performance analysis of cryptographic protocols on handheld devices. In *Proceedings of the Third IEEE International Symposium on Network Computing and Applications (NCA'04)*, September 2004.
19. Berbecaru D. On measuring SSL-based secure data transfer with handheld devices. In *Proceedings of the 2nd International Symposium on Wireless Communication Systems*, September 2005.
20. Hager C, Midkiff S, Park J, Martin T. Performance and energy efficiency of block ciphers in personal digital assistants. *In Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2005.
21. Potlapally N, Ravi S, Raghunathan A, Jha N. A study of the energy consumption characteristics of cryptographic algorithms and security protocols. *IEEE Transactions on Mobile Computing* 2006; **5**(2): 128–143.
22. Kim K, Hong J, Lim J. Analysis of the power consumption of secure communication in wireless networks. *Lecture Notes in Computer Science, Database and Expert Systems Applications* 2006; **4080**: 894–903.
23. Shin Y, Gupta M, Myers S. A Study of the performance of SSL on PDAs. In *Proceedings of IEEE INFOCOM Workshops*, April 2009.
24. HP iPAQ h6340, http://www.laptop-battery-inc.co.uk/pda-batteries/hp-ipaq-h6340.htm.
25. NI USB 6008, http://sine.ni.com/nips/cds/view/p/lang/en/nid/14604.
26. Wireshark, http://www.wireshark.org/about.html.

DOI: 10.1002/sec