

# A Study of the Energy Consumption of Security Encryption Policies in Wireless Devices

D.S.Dawoud  
University of UKZN

Bigondo Alexis  
National University of Rwanda

P.Dawoud  
University of UKZN

## Abstract

Security is becoming an everyday concern for a wide range of electronic systems that manipulate, communicate, and store sensitive data. An important and emerging category of such electronic systems are battery-powered mobile appliances, such as personal digital assistants and cell phones, which are severely constrained in the resources they possess, namely, processor, battery, and memory. This work focuses on one important constraint of such devices—battery life—and examines how it is impacted by the use of various security mechanisms. In this paper, we first present the cryptographic primitives needed to secure wireless communication. We then discuss the impact of the main parameters of the cryptographic algorithms on the overall energy consumption for secure data transactions. We then analyze the performance of different security encryption protocols set in a wireless access point regarding energy consumption. Our research focused mainly on Static WEP, 802.1x-EAP-TLS-WEP128, and 802.1x-EAP-TLS-TKIP. We compared the performance of those security encryption protocols by measuring the power consumed on a laptop.

**Index Terms**— Cryptographic primitives, security policy, battery gap, Static WEP, 802.1x standard

## 1. Introduction

Wireless, mobile, and limited battery-power devices like PDAs with built in WiFi, Cellular, and VoIP accesses are becoming popular now days. These systems are used to store, access, manipulate, or communicate sensitive data. Most of the battery-powered systems use wireless communication for achieving their targets; the matter that introduces security concerns due to the public nature of the physical communication channel.

Secure communication is typically achieved by employing security protocols at various layers of the network protocol stack, e.g., WEP [1] at the link layer, IPSec [2] at the network layer, TLS/SSL [3] and WTLS [4] at the transport layer, SET at the application layer, etc. The building blocks of a security protocol are cryptographic algorithms, which are selected based on the security objectives that are to be achieved by the protocol. They include asymmetric and symmetric encryption algorithms, which are used to provide authentication and privacy, as well as hash or message digest algorithms that are used to provide message integrity. For the system to run these cryptographic primitives for secure communication, the system must consume more power.

While the processing power, memory and network bandwidth of today's mobile and wireless devices are sufficient

(rather increasing exponentially according to Moore's law), their battery power is increasing at a modest pace (linearly). This results in what is called "battery-gap". The power consumption needed for processing the data using the predefined communication protocols and security protocols (cryptographic primitives) can quickly drain the batteries of such devices leaving them useless. Hence, it is essential to find measures that minimize the power consumption of the battery-powered devices.

During a secure wireless session the main sources of energy consumption are:

- transmission and reception of packets (size of actual data transfer),
- number and size of messages required for establishing the session, and
- cryptographic computations (number of operational rounds, key sizes, choice of ciphers (symmetric, asymmetric, hash etc.), modes of cipher (ECB, CBC, CFB etc.), etc.) in that order.

To reduce the energy consumption we must target:

- Energy-efficient communication protocols at several layers of network stack like Application, Transport, MAC, and Network.
- Energy-efficient cryptographic primitives.

To optimize the power consumption needed for cryptographic computations we start by understanding the main factors defining the energy consumptions. These factors are:

- **The security protocol used:** The concern for security in practice is addressed by choosing a security protocol, which achieves all the required security objectives. Security protocols realize the security objectives through the use of appropriate cryptographic algorithms. Any encryption algorithms need two stages: key –establishment (includes authentication and cryptographic primitive exchange) and encryption (encrypting the data before transmission). The number of messages exchanged in the first stages and the computational overhead needed for encryption define the energy requirement of the protocol.

The selection of the cryptographic primitives defines the computation overhead needed. This is because it defines the cryptographic computations; number of operational rounds, key sizes, choice of ciphers (symmetric, asymmetric, hash etc.), modes of cipher (ECB, CBC, CFB etc.), etc.

Commonly used security protocols, like SSL/TLS, IPSec, etc., have the freedom of realizing the desired security objectives by choosing specific cryptographic algorithms from a predefined set. In addition, the communicating parties can also decide upon parameters which influence the mode of operation of the chosen cryptographic algorithm.

We can make the security protocols energy-cognizant by allowing them to alter their operation depending on the operating environment. This adaptation of behavior is guided by rules, which determine the best possible alternative with

respect to energy efficiency under any given input conditions. These changes may involve a conscious and conservative tradeoff between the level of security and energy.

- **Efficiency of executing Cryptographic Primitives:** By making the execution of constituent cryptographic algorithms (cryptographic primitives) efficient through a combination of hardware and software techniques [5], [6], [7], [8], [9], we can improve the performance and energy requirements of security protocols.  
Reducing energy consumption by employing hardware acceleration of crypto-mechanisms (e.g. FPGA implementation of AES) proved to be an effective way. However, it is extremely difficult to provide customized hardware for encryption which would mean more gates and more power consumption. Also, customized encryption hardware is vulnerable to differential power analysis attack. So, a careful investigation is needed before making available encryption through hardware implementations over the mobile wireless devices.
- **Sensitivity of the data (Degree of the security required):** When encryption is used, the data transfer is broken up into sessions. Each session, as mentioned before, comprises two stages: **authentication** and **key-establishment** by using an asymmetric algorithm, followed by transmission of data after encrypting them using a symmetric algorithm with the key established (in the first stage of the session). The amount of data transmitted in a session is referred to as *session length*. The maximum amount of data which can be transferred in each session, i.e., upper bound on the session length, is specified by the security policy and is determined by the sensitivity of the data: The greater the sensitivity of the data, the shorter the session length, and vice versa. Thus, in the case of sensitive data, session lengths are short and the frequency of setting up new sessions is high.

Finding cryptographic primitives that minimize the computational and power-consumption overheads is an important factor for battery-powered embedded system and it was a field of many recent researches. Researchers have recently proposed interesting approaches to the design of lightweight security protocols have been devised for sensor nodes by analyzing the impact of security algorithms on the energy consumption of sensor nodes [12].

Sohail Hirani [10] evaluated various symmetric key such as AES, IDEA, CAST and asymmetric key such as RSA, ElGamal, ECIES algorithms with different key sizes on different devices. He studied also the effect of varying signal to noise ratio and varying packet sizes. He gave also some suggestions for design secure communication systems to handle varying wireless environment. He found that regarding symmetric keys AES is more energy efficient than the other. Concerning asymmetric key, Sohail Hirani found that RSA is more energy efficient [11].

Kerry McKay [13] gives a Trade-offs Between Energy and Security in Wireless Networks; he focused on WEP, WPA and CCMP, measured energy consumed to retrieve web pages over a wireless network. He suggested that there is an increase in energy consumption when increasing a particular protocol, but does not necessarily increase when changing to a more secure protocol. Kerry McKay also found that the cost of the different encryption algorithms did not vary significantly.

Aamer Nadeem, Dr M. Younus Javed [14] studied the performance of DES, 3DES, AES and blowfish by encrypting input files of varying contents and sizes, on different Hardware platforms. And based on their experiment they found that blowfish is the best performing than DES, 3DES, and AES.

P. Prasithsangaree and P. Krishnamurthy [15] compared the performance of RC4 and AES regarding energy consumption, and their found that RC4 is more suitable for large packets and AES for small packets.

The work in [16] evaluated the energy consumption of selected key-exchange protocols on a WINS sensor node and proposed energy-efficient ways for exchanging cryptographic keys, while custom protocols for low-power mutual authentication were proposed in [17], [18]. Energy tradeoffs in the network protocol and key management design space of sensor nodes were explored in [19]. Techniques to minimize the energy consumed by secure wireless sessions have also been proposed in [20]. We believe that comprehensive energy analyses of security protocols, such as the one performed in our work, will facilitate identification of energy bottlenecks and development of energy-efficient security mechanisms.

In this paper we analyze the performance of different security encryption protocols set in a wireless access point regarding energy consumption. Our research focused mainly on Static WEP, 802.1x-EAP-TLS-WEP128, and 802.1x-EAP-TLS-TKIP. We compared the performance of those security encryption protocols by measuring the power consumed on a laptop. Three scenarios were taken into consideration regarding data, where we created a congested network by downloading a big file of 7.5 GB; regarding distances, by changing locations of the laptop, where the access point remain fixed. For each scenario, we compared the power consumed when using the different security encryption protocols. The aim of this research is to study the behavior of security encryption policies regarding batteries consumption on the laptop, thus the accuracy or the perfection of the software was not very important. The test of the software accuracy used for the measurement is not covered in this research.

The paper is arranged as follows: In Section-2 we introduce the security policies used in the test. In section-3 we introduce the experimental setup and the methodology used. Section-4 provides analysis and interpretation of results as well as a Conclusion and discussions.

## II. Security Protocols Used in the Test

The following security policies are considered.

### IEEE 802.1x

802.1x is a flexible framework created for authentication in PPP protocol. Using the existing hardware, 802.1x can also be applied to a wireless network to allow key distribution for TKIP. 802.1x defines the idea of port-based access control; conceptually access control requirement involves two ports: a controlled port and an uncontrolled port. Access to the uncontrolled port should be gained at any time, and leads to the authentication service. The controlled port should only be accessed when authentication and authorization have been processed. In wireless networks, the controlled port is the Access Point's (AP) connection to the network, and the uncontrolled port goes to an authentication server, such as RADIUS.

This authentication scheme presents three part; the supplicant which is the entity that wishes to be authenticated (wireless client), the authenticator which is the entity that the supplicant is trying to authenticate (access point) and the authentication provided by the third party, which is the authentication server, through communication with the authenticator. The supplicant and authenticator send requests or messages over the wireless medium, while the authenticator and authentication server communicate through the wire [5].

## EAP

EAP started to be used in the Point-to-Point Protocol (PPP) as a means of establishing connections over dial-up connection. Since then EAP has been adapted for use in the wireless domain as a means of passing logon credentials between a wireless user and an authentication server. EAP and IEEE 802.1x work together to pass this logon information between the client and the authentication server. IEEE 802.1x is a transport medium for EAP frames. When a client connects to a closed port, IEEE 802.1x opens that port for the transportation of EAP credential frames between the supplicant and the authentication server through the authenticator [3].

## WEP

IEEE 802.11 networks currently have three encryption protocols available for use today: Wired Equivalent Privacy (WEP), Temporal Key Integrity Protocol (TKIP), and Counter Mode/CBCMAC Protocol (CCMP). However this research focused on WEP and TKIP

The WEP protocol was designed to provide a level of security similar to wired network with three goals to achieve: confidentiality, data integrity, and to provide access control to the network. However, after the delivery of the WEP algorithm several vulnerabilities were discovered that severely hamper its ability to perform these functions. WEP was created using the RC4 stream cipher with 40 bits and 104 bits WEP keys with a public 24-bit initialization vector (IV) which gives a total keys length of 64 and 128 bits respectively, used in industry. The initialization vector (IV) provides added security to data as it changes with each packet.

WEP was designed to use up to four different pre-shared keys for every packets sent, which was actually a huge amount of security.

## TKIP

The Temporal Key Integrity Protocol was developed to fix WEP's vulnerabilities; it is used in Wi-Fi Protected Access (WPA). TKIP was developed such a way that, it should be compatible with WEP to prevent the need to replace all hardware that only supported WEP, when migrating to TKIP. TKIP use the RC4 stream cipher but implements a 48-bit initialization vector. This significantly reduces IV reuse and the possibility of a hacker collecting a sufficient number of 802.11 frames to crack the encryption. TKIP has also other numbers of function added such as: New Message Integrity Code (MIC) also called Michael to defeat forgeries, Per-packet key mixing algorithm that provides a RC4 seed for each packet, and active countermeasures to prevent packet modification and replaces the flawed IV system.

The message integrity check was designed with the idea of overcoming WEP's vulnerability ICV by creating a non-linear hash to prevent packet modification. Michael is too weak to stand alone, so TKIP mandates countermeasures: TKIP requires a rekey after detecting a MIC validation error, and limits rekeying to once per minute. With this design the maximum expected number of false positives is about one per year.

TKIP uses extended IV, ICV and MIC. This needs 20 octets which is double the overhead associated with WEP.

Table 1 illustrates the comparison of security protocol between TKIP and WEP.

Table -1 Comparison of security protocols between WEP and TKIP [6]

|                                | WEP   | TKIP  |
|--------------------------------|---|---|
| Cipher Key size (s)            | RC4 40- or 104-bit encryption                       | RC4 128-bit encryption<br>64-bits encryption                |
| Key Lifetime Per-packet Header | 24-bit wrapping<br>IV Concatenate IV<br>to base key | 48-bit IV TKIP mixing<br>function                           |
| Integrity Packet Header        | None  | Source and destination<br>addresses protected by<br>Michael |
| Packet data replay detection   | CRC-32<br>None                                      | Michael<br>Enforce IV sequencing                            |
| Key Management                 | None  | IEEE 802.1X   |

## **III. Experimental Design**

The platform used in this paper is a small WLAN on which we carry out a variety of experiments designed to address the energy consumption of variant security protocols in a wireless AP. The experiments work with MAC layer Protocols such as WEP. In our testbed we used 802.1x/EAP-TLS as authentication means. On application layer we consider the RADIUS protocol, which is based on client-server architecture as described earlier. As it is shown in Figure 1 our test-bed architecture is made of two PCs linked to a Cisco wireless access point, with a FTP cable via a Cisco switch, and two laptops linked to the wireless access point, through wireless connection.

### Hardware selection

The platform is made of two Dell PCs with Pentium IV 3.0 Ghz and 512 MB of RAM. One was used as domain controller server and the other was used as RADIUS server. The two PCs are connected to Cisco access Points (Cisco aironet 1200 series) to provide wireless connectivity. Security over wireless segment in the test-bed is provided by configuring different security protocols. Two laptops one is a TOSHIBA SATELLITE A110 (AMD processor, 1.8 Ghz, 512 MB of RAM) and a HP dv4000 (1.8 Ghz, 512 MB of RAM) but only HP was used for measurement and TOSHIBA was used to create a congested network in one scenario. A Cisco catalyst 2950 series and three FTP cables are used as a network switch and medium to link the two PCs and the access point. The laptop HP dv4000 used for measurements had a lithium ion as type of battery. The testbed topology and the equipments composition are shown clearly Figure 1.

### Software selection

It was important to select a compatible operating system and software to run the tests. This is to avoid any possibility of any problems that may arise when we use them in the test-bed. It was also important to find the equipment required for software installation. The software used is windows 2003 server SP2 installed on each PC and windows XP professional SP2 on each laptop. For power measurement, an open source called BatteryMon which is a battery monitoring was used. For Data analysis we used SPSS 11.5 and Microsoft Excel.

### BatteryMon.

BatteryMon is a battery monitoring software solution. It has been designed as an easy-to-use Windows based application that allows users to monitor the performance of laptop batteries and uninterruptible power supplies (UPS). BatteryMon has the following features:

- Graphically see the charge / discharge rate

- Diagnose problem battery cells with detailed statistics
- Compare and measure your batteries performance with expected discharge rates
- See the critical discharge point for your batteries
- Uniquely identify battery packs for tracking purposes or system inventories
- Log the performance of a battery for later reference
- See the status of each individual battery pack, when multiple batteries are in use

Figure 2 shows the interface used for battery information and battery charge/discharge.

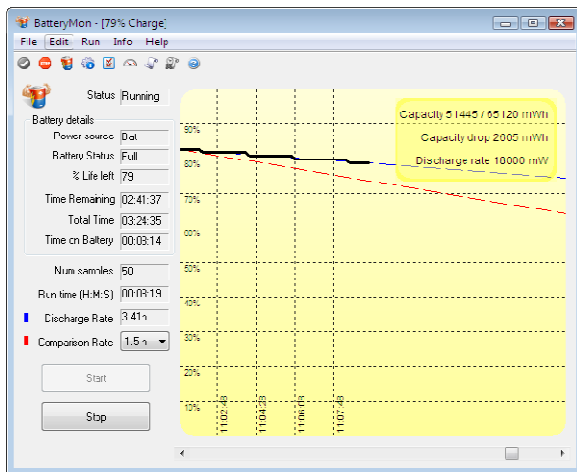
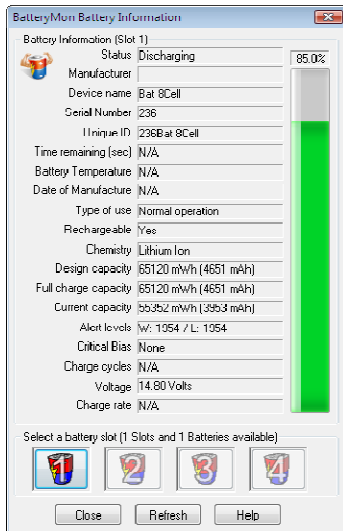


Figure 2 Batterymon interface [12]

**Network scenarios measurement**

To conduct the research, different scenarios were taken into consideration. The intention was to examine how various physical obstacles and distances regarding the AP behave on the energy consumed on the laptop battery. This was used to see the impact of the signal to noise ratio by changing measurement places, creating congested network, regarding performance encryption security policies. Some of the scenarios used in the test are:

**Scenario 1:** In this scenario the test-bed was located entirely within one room, the lab. The client (laptop) used for the test was closer to the wireless equipment (AP). This scenario was used to test the battery consumed in maximum throughput for the link.

**Scenario 2:** The client is located at a distance of about a 100 meters separated by many obstacles.

**Scenario 3:** In this case we use two laptops, where one was used to download a large data file and the other was used to take measurement. This helps in finding the impact of the noise or congested network regarding battery consumption.

**security policies**

Each scenario has been tested for the following security encryption policies:

- “no security”: There is no security services enabled in the network. “no security” policy helps in comparing the energy consumed regarding other security services.
- Static WEP;
- 802.1x-EAP-TLS-WEP 128; and
- 802.1x-EAP-TLS-TKIP

All those policies were set in the access point at every measurement. 802.1x-EAP-TLS was used for the authentication purpose. Static WEP, WEP 128 and TKIP were used for encryption. Table 2 shows the features of security policies.

Table 2: features of security policies

| No  | Security policy        | A | C | I | NR | MA |
|-----|------------------------|---|---|---|----|----|
| P-1 | No security            |   |   |   |    |    |
| P-2 | Static WEP             | Y | Y |   |    |    |
| P-3 | 802.1x-EAP-TLS-WEP-128 | Y | Y |   | Y  | Y  |
| P-4 | 802.1x-EAP-TLS-TKIP    | Y | Y | Y | Y  | Y  |

A – Authentication; C- Confidentiality; I- Integrity; NR-Nonrepudiation; MA- Mutual Authentication

**Methodology**

We downloaded a file of 50 MB from one PC to the laptop whereas a policy is set according to each scenario.

For the first scenario we started by setting a security policy and then download the 50 MB file from the PC to the laptop. Values were taken from batteryMon in terms of percentage dropped. For each policy the measurement was repeated 15 times and the real value was their average (See Table-3).

Table 3: Illustration of measurements

| Number of download | Battery percentage | Percentage dropped |
|--------------------|--------------------|--------------------|
| 1 <sup>st</sup>    | 99.0%              | 1.3%               |
|                    | 97.7%              |                    |
| 2 <sup>nd</sup>    | 96.0%              | 0.9%               |
|                    | 95.1%              |                    |
| 3 <sup>rd</sup>    | 93.8%              | 1.6%               |
|                    | 92.2%              |                    |

Value for the policy:  $(1.3\%+0.9\%+1.6\%) / 3 = 1.266\%$

For the second scenario the measurement was the same apart that the laptop was moved in other room. For the third scenario another laptop was involved to create a congested traffic by using to download another file of 7.5GB. We choose that big file because it downloads would cover the whole 15 repetitions. The rest remain the same as the first scenario. To avoid interference with battery alarm we had to recharge the battery after it reach 40% of energy consumed.

**IV. Results and Conclusions**

For each combination of scenario/security policy we run the test 15 times. In each run we downloaded a file of 50MB. At the end of each run the power consumed and the time taken for

downloading are measured. The power consumed and the time is averaged for each scenario/security policy combination. Table-4 gives the output of our measurements of energy consumed in percentage of workload transfers when varying the security encryption policies for each scenario. Measurement was taken at each set of security encryption policies in the access point (AP).

Table-5 shows the value in terms of times consumed in second for each repetition of each security encryption policy obtained after the measurement in the test lab by downloading a 50MB file.

Tables 3 and 4 are represented graphically in Figure 3 and Figure 4 respectively.

1. From Figure 4 we can see that the security policy has an impact on the battery consumption.

2. The fact that the impact is not very significant can be easily explained by discussing the key initialization used by security algorithms under consideration. All the policies use the same key initialization and initial permutation process; they are using RC4. The key initialization and initial permutation processes depend on the key size, but for smaller key size, the key bytes are simply replicated consequently, the processes in computation and energy consumption are identical [13].

3. If we compare the “no security” with the other three policies, we can note that the difference is not so significant. The reason again is the use of RC4. RC4 does not consume much energy. The power overhead when using RC4 is not very high specially when there is a transmission of large data blocks.

This explains why RC4 became one of the best encryption techniques during the last ten years. RC4 is standardized to provide security in wireless local area networks (WLANs) by using Wired Equivalent Privacy (WEP).

4. The difference in power consumed by TKIP and that consumed by WEP is coming from the way of key processing in the two policies. The energy consumed by WEP encryption is directly linked to two factors which are; the length of the encryption key (the concatenation of IV and shared secret) and the length of the data to be encrypted. The length of the plaintext will always be the addition of the data length and the 32 bits, because there is always a 32-bit CRC appended to the data when encrypting. The key scheduling algorithm (KSA) deals with the key only, and the energy used during this phase increases with the length of the key. The pseudo random generating algorithm (PRGA) uses the result of the KSA to produce a stream of length equal to that of the plaintext. [14]. In case of TKIP the RC4 key used has two phases of computations to incorporate the transmitter address prior to RC4’s KSA and PRGA, therefore there is more computation. TKIP uses Michael which adds 32 more bits to the plaintext. This produces a high amount of PRGA and XOR operations that occur for a fixed amount of data for TKIP than for WEP. Michael is also more computationally expensive than a CRC. According to the way keys are used, TKIP encryption is slightly more secure than WEP encryption.

5. Figure 4 shows the average time spent for each security encryption policy. The curve shows the same results as shown by Figure 3. Time consumed is related to energy consumed as the more it takes times to download the file, the more it consumes much energy. The time and the energy consumed are proportional to the complexity of operations for each security encryption policy.

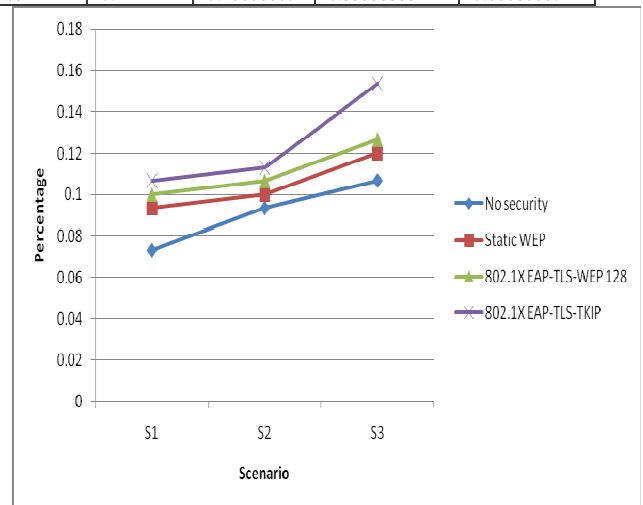
6. The difference in the energy consumed by certain policy in the three scenarios S1, S2 and S3 are the result of the noise and the delay added to the system. As the distance between the lab top and AP increases, the noise and delay increases and accordingly the power consumption.

**Table 4:** Energy consumed by different security policy

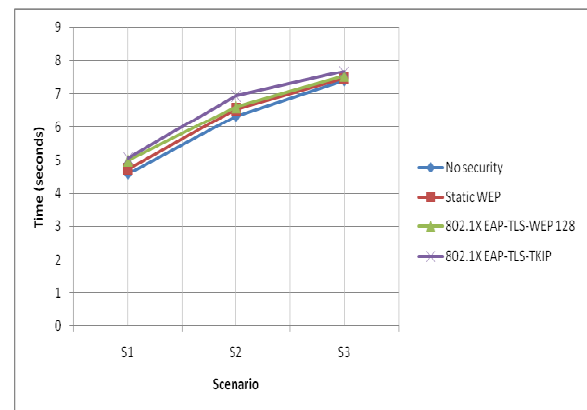
| Scenario | Security Policy Used |            |                       |                     |
|----------|----------------------|------------|-----------------------|---------------------|
|          | No security          | Static WEP | 802.1x-EAP-TLS-WEP128 | 802.1x-EAP-TLS-TKIP |
| S1       | 0.07333333           | 0.09333333 | 0.1                   | 0.10666667          |
| S2       | 0.09333333           | 0.1        | 0.10666667            | 0.11333333          |
| S3       | 0.10666667           | 0.12       | 0.12666667            | 0.13333333          |

**Table 5:** Time consumed

| Scenario | Security Policy Used |            |                       |                     |
|----------|----------------------|------------|-----------------------|---------------------|
|          | No security          | Static WEP | 802.1x-EAP-TLS-WEP128 | 802.1x-EAP-TLS-TKIP |
| S1       | 4.6                  | 4.7333333  | 5                     | 5.0666667           |
| S2       | 6.333333             | 6.5333333  | 6.6                   | 6.9333333           |
| S3       | 7.4                  | 7.4666667  | 7.5333333             | 7.6666667           |



**Fig.3** Percentage energy consumed



**Fig.4** Average consumed time

## V. References

- [1] LAN MAN Standards Committee of the IEEE CS, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: IEEE standard 802.11, 1990.*
- [2] IPSec Working Group, <http://www.ietf.org/html.charters/ipsec-charter.html>, 2000, 138]
- [3] SSL 3.0 Specification, <http://wp.netscape.com/eng/ssl3/>, 1996.
- [4] Wireless Application Protocol 2.0—Technical White Paper, <http://www.wapforum.org/>, Jan. 2002.
- [5] Compaq iPAQ Pocket PC, <http://h20022.www2.hp.com>, 2002.
- [6] J. Goodman, A. Chandrakasan, and A. Dancy, “Design and Implementation of a Scalable Encryption Processor with Embedded Variable DC/DC Converter,” *Proc. Design Automation Conf.*, pp. 855-860, June 1999.

[7] Z. Shi and R. Lee, "Bit Permutation Instructions for Accelerating Software Cryptography," *Proc. IEEE Int'l Conf. Application-Specific Systems, Architectures, and Processors*, pp. 138-148, July 2000.

[8] J. Burke, J. McDonald, and T. Austin, "Architectural Support for Fast Symmetric-Key Cryptography," *Proc. Int'l Conf Architectural Support for Programming Languages and Operating Systems*, pp. 17-189, Nov. 2000.

[9] N. Potlapally, S. Ravi, A. Raghunathan, and G. Lakshminarayan "Optimizing Public-Key Encryption for Wireless Clients," *Proc. IEEE Int'l Conf Comm.*, pp. 1050-1056, May 2002.

[10] S. Ravi, A. Raghunathan, N. Potlapally, and M. Shankaradas, "System Design Methodologies for Wireless Security Processing Platform," *Proc. Design Automation Conf*, pp. 777-782, June 2002

[11] [http://etd.library.pitt.edu/ETD/available/etd-04252003-165520/unrestricted/shirani\\_etd2003.pdf](http://etd.library.pitt.edu/ETD/available/etd-04252003-165520/unrestricted/shirani_etd2003.pdf), 28.12.2007

[12] <http://www.ncjrs.gov/pdffiles/172868.pdf>, 28.12.2007

[13] <http://www.passmark.com/products/batmon.htm>, 28.12.2007

[14] <http://www.ieeexplore.ieee.org/iel5/10652/33619/01598556.pdf>, 26.11.2007

[15] <http://www.ieeexplore.ieee.org/iel5/8900/28134/01258477.pdf?arnumber=1258477>, 26.11.2007

[16] A. Hodjat and I. Verbaauwhede, "The Energy Cost of Secrets in Ad-Hoc Networks," *Proc. IEEE CAS Workshop Wireless Comm. And Networking*, Sept. 2002.

[17] M. Jakobsson and D. Pointcheval, "Mutual Authentication for Low-Power Mobile Devices," *Proc. Financial Cryptography*, pp. 171-195, Feb. 2001.

[18] D.S. Wong and A.H. Chan, "Mutual Authentication and Key Exchange for Low Power Wireless Communications," *Proc. IEEE Military Comm. Conf.*, pp. 39-43, Oct. 2001.

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

[20] R. Karri and P. Mishra, "Minimizing Energy Consumption of Secure Wireless Session with QoS Constraints," *Proc. Int'l Conf. Comm.*, pp. 2053-2057, May 2002.

**D.S.Dawoud** is a Professor in Computer Engineering- University of KwaZulu Natal. His main fields of research are: Computer Engineering, Network Security and Encryption and Embedded Systems.

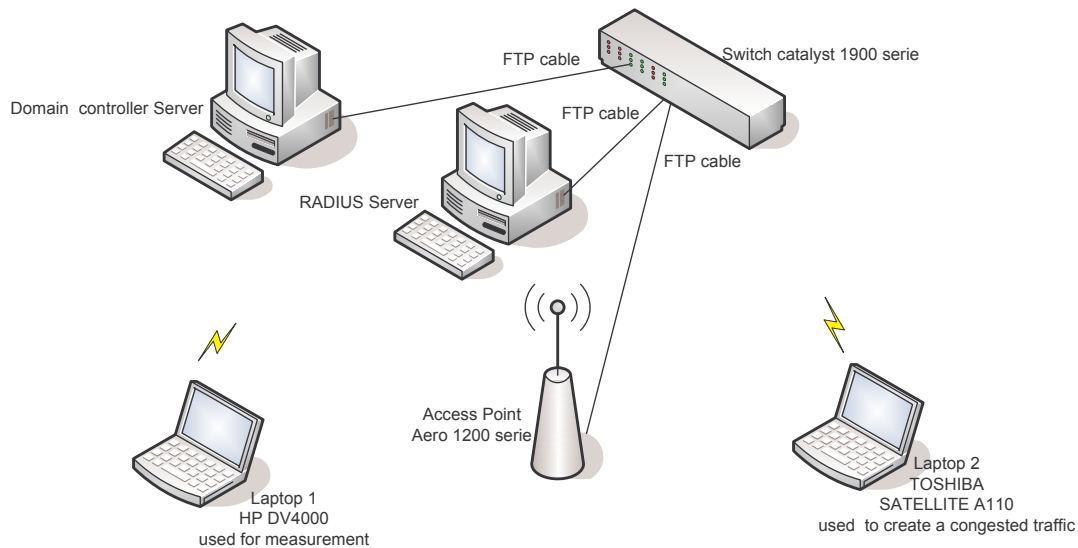


Figure 1 testbed architecture