Simulation Analysis of Session Key Exchange Protocols based on Key Parameters

Pranav Vyas
Smt. Chandaben Mohanbhai Patel Institute of Computer Application
CHARUSAT, Changa, Gujarat, India

Bhushan Trivedi, PhD.
Gujarat Law Society Institute of Computer Technology
Ahmeadabd, Gujarat, India

Atul Patel, PhD.
Smt. Chandaben Mohanbhai Patel Institute of Computer Application 1st line of address
CHARUSAT, Changa, Gujarat, India

ABSTRACT

As the internet grows rapidly, role of security of information of users gain more importance than ever. Key exchange protocols are important in this regard. In this paper we evaluate performance of well known key exchange algorithms. In this paper we analyze their encryption speed and their power consumption on different platforms. We analyze algorithms on both traditional PC and mobile phones. Experiment result shows that protocols which are less computationally intensive and less power consuming but not very secure. We select the protocol which is most suitable for mobile computers.

General Terms

Applied Cryptography, Key Exchange Protocols

Keywords

Key Exchange Protocols, Performance Evaluation, Control Overhead, Power Consumption.

1. INTRODUCTION

Today we see people using applications from mobile computers and accessing internet for online shopping, stock trading, Internet banking and bill payment. Such transaction over wireless public network requires security, data authentication privacy and integrity.

Data encryption plays a vital role in this case to guarantee security of information. When data is encrypted it is transformed from plain text into cipher text which hides the original meaning of text and prevents malicious user from retrieving original data. Exchange of encryption keys makes sure that both parties are authenticated by each other and can understand encrypted data they send to each other. This is done using the key that they exchange using key exchange protocols. Various corporations and individuals sent their data over public network after encrypting it. The intent here is safety of data which can only be decrypted by intended recipient.

To decrypt the encrypted data the recipient needs key with which the data was encrypted. This key is originally generated by sender and needs to be transmitted to receiver. This transmission is done using key exchange protocols. According to [1] [2] key exchange protocols can also be used for encrypting MAC.

The problem of key distribution is solved using anonymous key exchange protocols such as Diffie-Hellman or asymmetric key encryption. It uses both private and public key. Encryption is achieved using public key and decryption is achieved by private key. Public key encryption depends on mathematical functions to generate a strong key. A strong key is calculated with exponential mathematical functions. Size of key also adds to complexity. Standard key sizes are 2048 bits for RSA and 128 bit ASE. Strong key generation needs powerful processor. This powerful processor consumes more power. Here, both processing power and battery power are constraints of mobile computers.

The contrast here is high security requirement of data encryption algorithm. It is also known to be computationally intensive. They require high amount of resources of computer such as processing power, memory and battery power. A wireless device has limited resources in terms such as battery which can be consumed easily due to intensive calculation by encryption algorithm.

These algorithms are very often used on wireless devices. Their performance evaluation is important to determine their domain application. It will also be helpful in optimizing the key exchange algorithm.

In this paper we evaluate performance of eight different key exchange algorithms on wireless devices. We run our algorithms on two laptop computers with different configurations. We analyze them based on their speed and power consumption. We also evaluate performance of key exchange algorithms on smart phones. Based on the results we conclude algorithm best suited for our requirement.

Table 1 describes hardware and software details on implementation of algorithms.

We run each algorithm 40,000 times in 400 cycles for each algorithm where in one cycle the algorithm is executed 100 times. The results that we show here are aggregation of each of cycle. We define key size at the 64 bit for each key.

We have studied these key exchange algorithms in depth at [3] [4]. We concluded that key exchange protocols are designed with no constraints on processing power and power consumption. These protocols were designed with desktop systems in mind. However with advancement in technologies and more and more devices using wireless technology to communicate, we need to redesign these protocols with keeping in mind constraints of constraints of wireless devices like limited processing power, low main memory. Redesigned protocol will address these constraints and try to overcome them by optimizing their performance.

This paper analyzes performance of different key exchange protocols from two important parameters: control overhead and power consumption. Based on results of experiments, we select most appropriate protocol to use on wireless computer devices.

The paper is divided into 4 sections. In section 1 we give introduction to the topic at hand. In section 2 we introduce evaluation methods. Section 3 is of results of evaluation and we provide conclusion to evaluation in section 4.

2. CONTROL OVERHEAD AND POWER CONSUMPTION CALCULATION

Role of key exchange algorithms is vital in protecting network information security. Evaluation of key exchange algorithm includes security analysis, throughput and power consumption. We have provided security analysis in [3] [4].

For computers, to simulate protocol we developed 3 different programs each for sender, receiver and trusted third party namely Bob, Alice and Trent. Trusted third party is used by some algorithm to exchange keys and authenticate identity of each other, other algorithms directly communicate from sender to receiver. We have taken readings from the program which generated the key in some cases it is trusted third party Trent and in other it is sender Bob.

For mobile phones, we developed 2 different programs where each phone has two different programs to communicate. Program one is to receive/send and encrypt/decrypt or request for session key packets and another program two which stores private keys of the parties who are going to communicate and generate session keys based on request by program one.

Table1. System Details

	Computer		Mobile Phone	
Maker	Dell	Dell	Samsung	Sony
Model	Studio 1559	XPS 15	Galaxy Ace	Xperia U
Processor	Intel Centereno 1.7 GHZ	Intel Core i5 2.4 GHZ	ARM 800 MHZ	Cortex 1 GHZ
Main Memory	4 GB	4 GB	278 MB	4 GB
Operating System	Ubuntu Linux 10.10	Ubuntu Linux 10.10	Android 2.3	Android 2.3
Language	C	C	Java	Java
Compiler	GNU C	GNU C	JVM	JVM

In [5] authors study distribution of packet sizes which are transmitted and received by wireless device over a wireless LAN. They used a packet capturing software to capture packets that were sent over wireless network for one hour. They conclude based on analysis of captured packets that the most packets captured are of small size, typically between 64-127 bytes. To evaluate performance of algorithms in wireless network we generated 128-bytes packets with random strings and numbers and added in them the generated key and other algorithm specific control information.

In this section we design methods to evaluate control overhead and power consumption of algorithms that we have studied in [3] [4]. The analysis of related work [6] [5] [7] [8] shows that control overhead and battery consumption is based on structure of algorithm and size of key and time it takes to generate the key.

2.1 Control overhead and key generation evaluation

Wireless devices have limited computation capabilities and there are many processes vying for the processor. Control overhead is the time that it takes in generating key and integrating the key in packet with other control information which will be sent across network to recipient. It is because of this reason that control overhead is considered an important factor of key generation algorithm. It is the time that the algorithm takes to generate key and integrate key into packet.

The core work of this evaluation is about observation of time it took to generate key. Once the key is generated it is integrated into packet with other data which is then encrypted with one of the public key encryption systems and sent over the network to recipient.

Control overhead is usually time taken to generate control information such as rout path and incorporating that information into network packet besides the actual data. Examples of such control overhead are described in [9] [10].

In this paper we calculate control overhead by taking into account time it takes to generate key and accommodating key in the packet besides any control information which may be used to address issues such as authentication, key freshness and reply attacks.

2.2 Power Consumption Evaluation

Power consumption is also a vital performance indicator for key exchange algorithm; especially for an application that will be used in portable wireless device. The power consumption is a widely researched subject. An example study on a wireless device has been conducted in [11]. The study in [11] only shows power consumption by different modules of wireless device under normal circumstances.

Another research on computational complexity of key exchange algorithm has been study on an embedded processor in [8]. It concludes that energy cost is based on authentication and key exchange is based on public key cryptography on an 8-bit microcontroller platform. In result it shows that one of the algorithms is more efficient then the other because of the reduced computation time and amount of data it transmits and stores.

To measure power consumption we charge laptop to its 100 percent battery capacity. Then we remove power cord and run algorithm for 40000 times. After the algorithm finishes we check the remaining percentage in battery and find out the actual consumption during execution.

For mobile phones we use an application called "Juice Defender". We charge mobile phone till it gives message that it is 100 percent charged and then run the application. We repeat this for each algorithm.

3. RESULTS

We ran the each algorithm 40,000 times on laptop and mobile phone without external power supply to obtain the control overhead and power consumption. Table 2 describes time it took for laptop computers and mobile phones in experiment to generate a key and integrate it in packet. Table 3 describes power remaining after execution of algorithms on laptop computers and mobile phones.

3.1 Result of control overhead and key generation time

Table2. Control overhead and key generation results

Key Exchange Protocols	Cycles	Key Generation+ Control Overhead (in seconds)			
		Laptop Computers		Mobile Phones	
		Dell Studio 1559	Dell XPS15	Samsung Galaxy Ace	Sony Xperia U
Wide Mouth Frog Protocol	40,000	7.1	4.7	12.5	9.4
Diffie-Hellman Protocol	40,000	7.8	5.1	13.1	10.2
Needhlam-Schroeder Protocol	40,000	8.4	5.9	13.9	11.1
Otway-Rees Protocol	40,000	8.1	5.5	13.4	10.7
Yahalom Protocol	40,000	8.5	6.1	14.1	11.4
Neuman-Stubblebine Protocol	40,000	8.3	5.7	13.7	10.9
Denning-Sacco Protocol	40,000	8.0	5.3	13.3	10.4

If the control overhead is less, the protocol is said to be more efficient as it is said to use less time and hence less processing and power in generating key and integrating it in packet with other protocol specific control information. The table as we can see is divided into two sections: Laptop computers and Mobile phones.

As we can see in table protocols with least control overhead in both the sections are Wide Mouth Frog Protocol and Diffie-Hellman Protocol. In laptop computer section control overhead were 7.1 and 7.8 seconds respectively on Dell Studio 1559 and 4.7 and 5.1 seconds respectively on Dell XPS15. In mobile phone section control overhead were 12.5 and 13.1 seconds on Samsung Galaxy Ace respectively and 9.4 and 10.2 seconds respectively on Sony Xperia U.

It means that these protocols took the least time in generating key and integrating that key into packet with other control information that is required by particular algorithm to authenticate and perform other functions.

The protocols with highest control overhead in laptop computer section are Needhlam-Schroeder protocol and Yahalom protocol with control overhead of 8.4 and 8.5 seconds respectively for Dell Studio 1559 and 5.9 and 6.1 seconds for Dell XPS15. In mobile phone section highest control overhead were 13.9 and 14.1 seconds for Samsung Galaxy Ace respectively and 11.1 and 11.4 seconds for Sony Xperia U.

That means that these protocols takes most time as compared to other protocols in list to generate key and integrated it into packet with other control information.

3.2 Result of power consumption evaluation

Battery power is one resource is that is available with wireless device in very limited amount, and all the functions of wireless device are dependent of this resource. We can say that this resource is very scarce one and should be used very carefully.

Table3. Power Consumption evaluation results

Key Exchange Protocol	Cycles	Remaining Battery (%)			
		Laptop Computers		Mobile Phones	
		Dell Studio 1559	Dell XPS15	Samsung Galaxy Ace	Sony Xperia U
Wide Mouth Frog Protocol	40,000	95	98	93	92
Diffie-Hellman Protocol	40,000	93	97	89	91
Needhlam-Schroeder Protocol	40,000	93	97	89	91
Otway-Rees Protocol	40,000	90	93	84	86
Yahalom Protocol	40,000	89	92	83	85
Neuman-Stubblebine Protocol	40,000	91	94	85	89
Denning-Sacco Protocol	40,000	90	95	85	89

The less power consumed by protocol, the better the protocol. In laptop computers we can see that two protocols that use least power are Wide Mouth Frog protocol and Diffie-Hellman protocol. After the experiment was completed using these protocols the remaining battery was 95% and 93% respectively in Dell Studio 1559 and 98% and 97% respectively in Dell XPS15. In mobile phone section also the same protocols uses least battery. After execution of algorithms remaining battery was 93% and 89% respectively in Samsung Galaxy Ace and 92% and 91% respectively in Sony Xperia U.

The protocols with highest power consumption in laptop computers are Needhlam-Schroeder protocol and Yahalom protocol. These protocols have battery consumption of 11% each in Dell Studio 1559 and 9% and 10% respectively in Dell XPS15. In mobile phone section, in Samsung Galaxy Ace phone Needhlam-Schroeder Protocol and Yahalom Protocol can be seen consuming highest power of 17% percent each and in Sony Xperia U same protocols executed with remaining battery backup of 84% and 85% respectively.

4. CONCLUSION

From the first look at results we can say that Wide Mouth Frog Protocol is most efficient of key exchange protocol when comparing these protocols parameters of control overhead and power consumption. As compared to other protocol Wide Mouth Frog protocol is very efficient in ideal conditions in theory. However, this protocol has many known vulnerabilities [3]. In real world environment there may not be ideal condition for execution. There are always malicious users who are ready to exploit vulnerabilities of a protocol and compromise security of data.

Experiment results show that in both laptop computers and mobile phone categories most of the protocols have slightly more control overhead then Wide Mouth Frog protocol and Diffie-Hellman protocol. Most protocols use little more battery power when compared to Wide Mouth Frog protocol and Diffie-Hellman protocol. The next closest protocol in terms of least control overhead is Denning-Sacco with control overhead of 8.0 seconds and 5.3 seconds in Dell Studio 1559 and Dell XPS15. In mobile phone section also we can see Denning-Sacco protocol performing next closest with 13.3 seconds and 10.4 seconds in Samsung Galaxy Ace and Sony Xperia U.

In terms of least power consumption also we see that least power consuming protocols are Wide Mouth Frog Protocol and Deiffe-Hellman protocol. The next best protocol in laptop computer is Otway-Rees and Denning-Sacco with 90% in Dell Studio 1559 and Denning-Sacco protocol with 95% battery remaining after all the cycles were completed in Dell XPS15. In mobile phone section protocols consuming next least battery power are Denning-Sacco Protocol and Neuman-Stubblebine protocol with 85% battery remaining for each after all the cycles are executed on Samsung Galaxy Ace phone and on Sony Xperia U the same protocols can be seen consuming next least battery power with Denning-Sacco Protocol and Neuman-Stubblebine protocol with 89% battery remaining for each after execution ends.

Therefore, we can conclude that both Wide Mouth Frog protocol and Diffie-Hellman protocol are good performance wise but they have serious security issues discussed in [3][4].

That takes us to next best protocol base on our parameter which is Denning-Sacco protocol in laptop computer and mobile phone category. Denning-Sacco protocol consumes same amount of power when compared to Otway-Rees protocol on laptop computers. In mobile phone category on battery consumption parameter we can see that both Denning-Sacco protocol and Neuman-Stubblebine protocol consumes same amount of power. However, if we look at control overhead of Denning-Sacco protocol and Neuman-Stubblebine protocol it is 13.3 seconds and 13.7 seconds in Samsung Galaxy Ace and 10.4 seconds and 10.9 seconds in Sony Xperia U phone. These figures show that even when battery consumption is the same, their control overhead times are different. We can clearly see that Denning-Sacco protocol is more efficient.

Based on result of experiment we can say that Denning-Sacco protocol takes less time in doing the same work. It can finish the processing of data in less time and hence drawing power from power source for less time. Based on experiment we can say that Denning-Sacco protocol is more suitable for key exchange over wireless network.

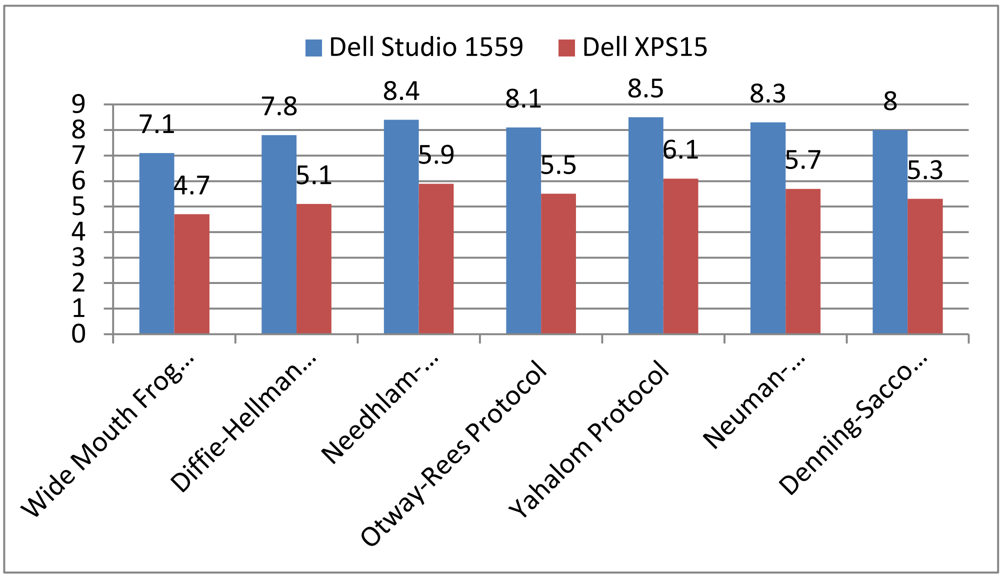

Fig.1 Control Overhead + Key Generation Time for Laptop Computers (in seconds)

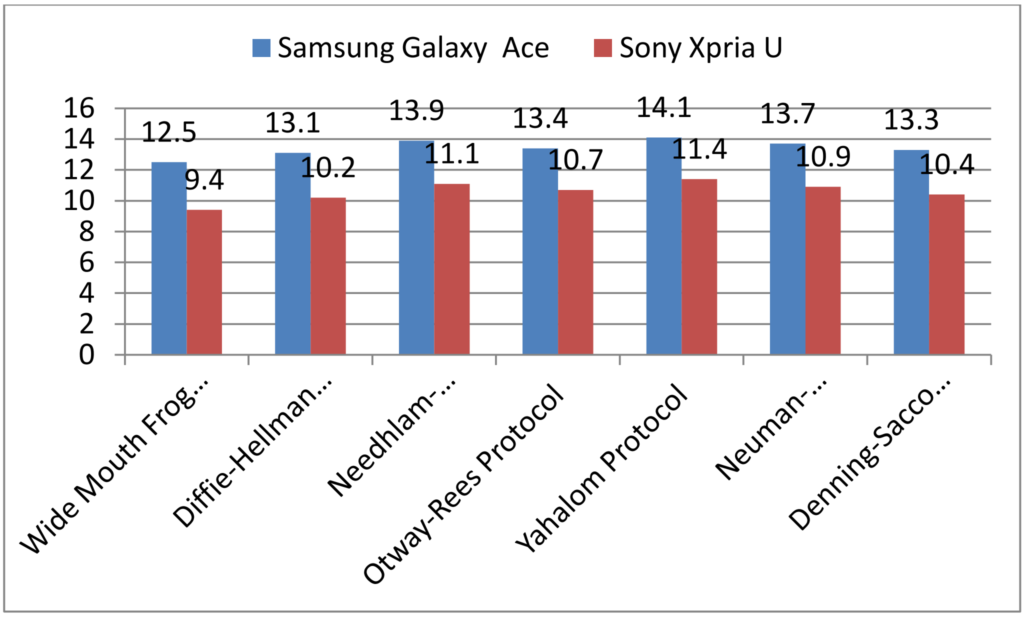

Fig.2 Control Overhead + Key Generation Time for mobile phones (in seconds)

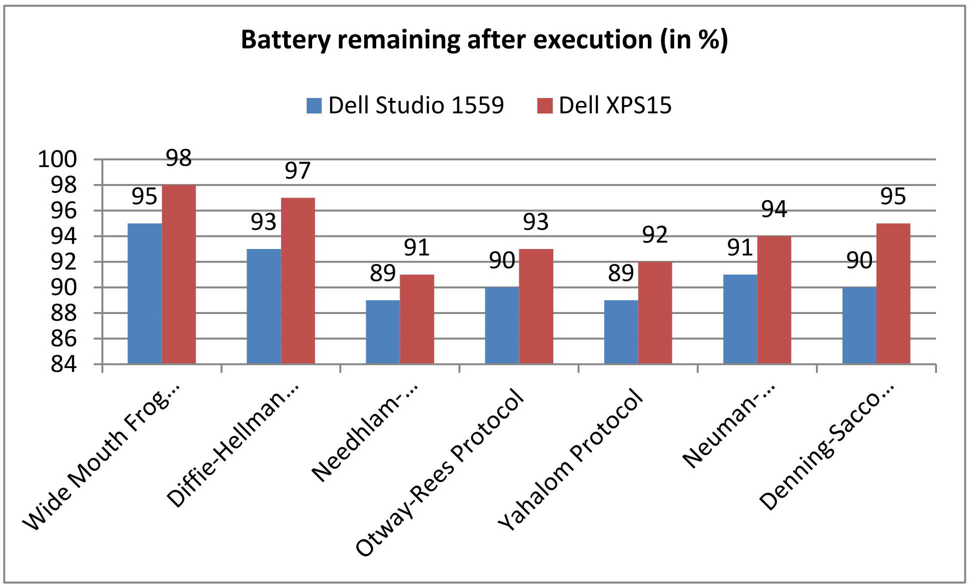

Fig.3 Battery remaining after execution for laptop computer (in %)

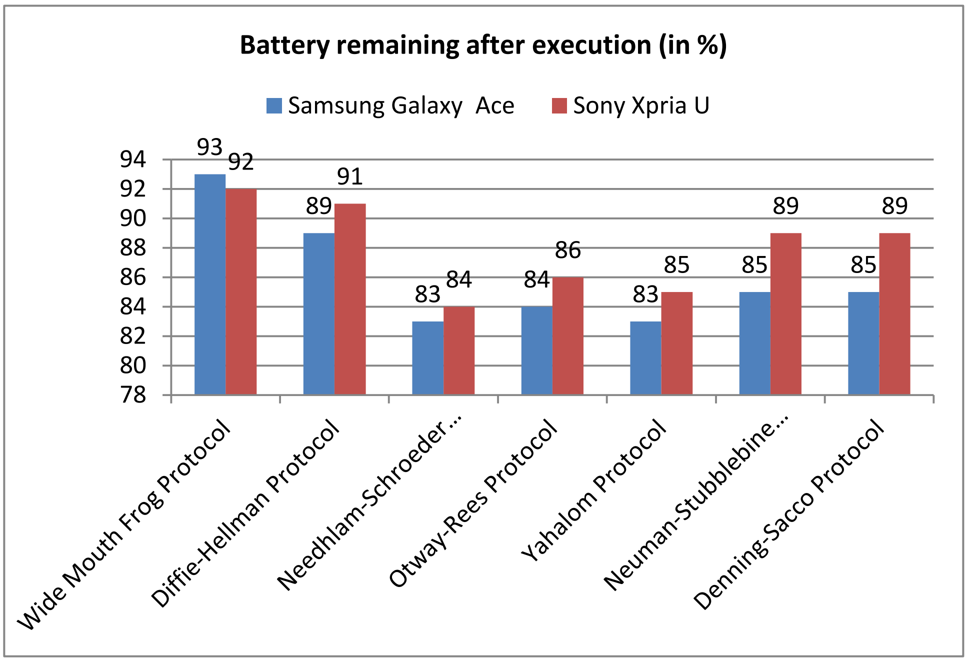

Fig.4 Battery remaining after execution for mobile phones (in %)

5. REFERENCES

[1] Diffie, W., Oorschot, P. C., & Wiener, M. J. (1992). “Authentication and authenticated key exchanges. Designs, Codes and Cryptography”, 2(2), 107-125

[2] Blake-Wilson, S., & Menezes, A. (1999). “Unknown key-share attacks on the station-to-station (STS) protocol”. In Public Key Cryptography (pp. 634-634). Springer Berlin/Heidelberg

[3] Vyas, P. J., & Trivedi, B. H. (2012). “Analysis of Key Exchange Protocols using Session Keys.” International Journal of Applied Information Systems.

[4] Vyas, P., & Trivedi, B. (2012) “An Analysis Of Session Key Exchange Protocols.” International Journal of Engineering Research and Applications.

[5] Prasithsangaree, P., & Krishnamurthy, P. (2003). “Analysis of energy consumption of RC4 and AES algorithms in wireless LANs”. Global Telecommunications Conference.

[6] Elminaam, D. S. A., Kader, H. M. A., & Hadhoud, M. M. (2008). “Performance Evaluation of Symmetric Encryption Algorithms”. IJCSNS International Journal of Computer Science and Network Security.

[7] Nadeem, A., & Javed, M. Y. (2005). “A performance comparison of data encryption algorithms”. In Information and communication technologies.

[8] Wander, A. S., Gura, N., Eberle, H., Gupta, V., & Shantz, S. C. (2005). “Energy analysis of public-key cryptography for wireless sensor networks”. Third IEEE International Conference on Pervasive Computing and Communications.

[9] Zhu, J., & Wang, X. (2011). “Model and Protocol for Energy-Efficient Routing over Mobile Ad Hoc Networks”. IEEE Transactions on Mobile Computing.

[10] Manfredi, V., Hancock, R., & Kurose, J. (2009) "Evaluating the Control Overhead of Routing Protocols in Dynamic Ad Hoc Networks," Annual Conference of the International Technology Alliance.

[11] Viredaz, M. A., & Wallach, D. A. (2001). “Power evaluation of a handheld computer: a case study”. Compaq Western Research Lab, Tech. Rep, 1.