

Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs

P. Prasithsangaree and P. Krishnamurthy

Telecommunications Program

University of Pittsburgh

Pittsburgh, PA 15260

{phongsak,prashant@mail.sis.pitt.edu}

Abstract- Encryption algorithms are known to be computationally intensive. They consume a significant amount of computing resources such as CPU time, memory, and battery power. A wireless device, usually with very limited resources, especially battery power, is subject to the problem of energy consumption due to encryption algorithms. Designing energy efficient security protocols first requires an understanding of and data related to the energy consumption of common encryption schemes. In this paper, we provide the results of experiments with AES and RC4, two symmetric key algorithms that are commonly suggested or used in WLANs. Our results show that RC4 is more suitable for large packets and AES for small packets.

I. INTRODUCTION

Encryption algorithms are known to be computationally intensive. They consume a significant amount of computing resources such as CPU time, memory, and battery power. A wireless device, usually with very limited resources, especially battery power, is subject to the problem of energy consumption due to encryption algorithms. In addition, wireless devices commonly transmit and receive data over a wireless link. The data is protected (confidentiality) before transmission using an encryption algorithm to keep the data secure from an eavesdropper. Encryption is also essential for other security services such as authentication, data integrity and access control [1]. Due to the intensive computation inherent in encryption algorithms, they tend to consume a substantial amount of energy or battery power. The battery can be quickly drained due to encryption, especially for a small wireless device. As shown in [2], encrypting only 13.6 kilobytes of data using 32-bit Blowfish algorithm [3] on a handheld device will drain about 75% of the battery power. However, for sufficient security strength today, it is recommended that key sizes of at least 80 bits be employed. Usually, a longer key implies more operations and the battery can be drained even more quickly. Encryption and transmission seem to be inevitable processes in wireless networks. In the literature, energy efficient communications protocols have been widely studied for wireless transmission [4]. Very few results consider energy efficient security protocols over a wireless link. For instance, a common approach to saving energy in communications protocols is as follows. Transmissions cost energy and consequently it is important to minimize wasted transmissions. So a short probe packet can be sent to discover whether the channel conditions

are good followed by actual bulk data transfer if there is a positive response from the receiver. Similarly, it may be possible to encrypt such probes with a light encryption algorithm (in terms of energy) if the security of the device will not be compromised. However, in order to investigate approaches to designing energy efficient security protocols, there is first a need to understand the energy consumption of different encryption schemes.

In this paper, we investigate the energy consumption of encryption algorithms commonly used or suggested for wireless local area networks (WLANs). We will study two different encryption algorithms known as RC4 and AES over two different protocol layers. Related work is described in Section II. In section III and IV, we explain our experimental design and show the results. We summarize our work and discuss future work in Section V.

II. BACKGROUND AND RELATED WORK

A. Encryption in Wireless Devices

Many encryption algorithms are widely available in wired networks. They can be categorized into symmetric key encryption and asymmetric key encryption. In symmetric key encryption or secret key encryption, only one key is used to encrypt and decrypt data and the key should be distributed before transmission between entities. It is also known to be very efficient since the key size can be small, the functions used for encryption are hardware operations, and encryption time can be very fast. However, in large communication networks, key distribution can be a significant problem. Asymmetric key encryption or public key encryption is used to solve the problem of key distribution. This uses two keys, one for encryption and another for decryption, and there is no need for distributing them prior to transmission. However, public key encryption is based on mathematical functions, computationally intensive and not very efficient for small wireless devices [3].

Generally, most encryptions used in wireless devices are based on symmetric key encryption. One such example is RC4. RC4 is a stream cipher designed by Ron Rivest in 1987 and it is widely used in many applications today and in wireless networks such as IEEE 802.11 WEP [5] and CDPD [6]. With a unique key, a stream of pseudo-random numbers

is generated. Then, the encryption of data using RC4 is simply based on XORing the pseudo-random numbers from the stream with the data. RC4 is known to be fast and efficient. It can be written using only a few lines of codes and requires only 256 bytes of RAM [7]. It is also very fast since it uses only 7 CPU clock cycles per byte of output on Pentium® CPU architecture [7]. Hence, it was one of the best encryption schemes during past decade. RC4 is standardized to provide security services in wireless local area networks (WLANs) using a protocol called Wired Equivalent Privacy (WEP) [8]. However, Fluhrer and many researchers have discovered several vulnerabilities in the RC4 algorithm [9].

The weaknesses in RC4 and loopholes in the WEP protocol have resulted in a new standard for security in WLANs (IEEE 802.11i) proposing a new protocol based on the *Advanced Encryption Standard* (AES). AES (previously called Rijndael) is a block cipher designed by Joan Daemen and Vincent Rijmen that has a variable key length of 128, 192, or 256 bits to encrypt data blocks of 128, 192, or 256 bits long. Both block and key length are extensible to multiples of 32 bits. AES encryption is fast and flexible, and it can be implemented on various platforms especially in small devices and smart card [10]. Also, AES has been rigorously tested for security loopholes for a few years before it was standardized by NIST. AES is considered to be very secure. In terms of the choice of algorithms in WLANs, both RC4 and AES have different tradeoffs. Additionally, a study of their energy consumption is also needed to decide on their use in security protocols.

B. Energy Consumption of Encryption on Wireless Devices

Energy consumption of wireless devices has been extensively studied. In [11], an evaluation of power consumption of an Itsy pocket computer has been conducted. This study is only intended to evaluate power consumption of different parts of the pocket computer under normal operations. In [12], the computational complexity of public key encryption has been studied on an embedded processor. The work concentrates on using several mathematical techniques to improve the performance of public key encryption in the secure socket layer (SSL) protocol. In [13], Law et. al. studied the energy consumption of encryption for sensor networks. In their work, the efficiency of code sizes and algorithms of RC5 and TEA are studied. Yuan and Qu proposed an energy efficient technique using dynamic voltage scaling to reduce energy consumption of public key encryption such RSA, DSA, and ElGamal [14]. In [15], an optimization of the energy consumption of SSL protocol is studied. Its technique is based on using a compression algorithm to reduce the size of the messages exchanged by the protocol in order to reduce the power consumed by encryption and transmission. In our previous work, the energy consumption of CAST, IDEA, and Triple-DES symmetric key encryptions on wireless handheld devices is

studied [2]. From Figure 1 [2], it can be seen that after only 600 encryptions of a 5 MB file using Triple-DES, the remaining battery power is 45% and subsequent encryptions are not possible as the battery dies rapidly. However, these works do not address the energy consumption of AES or RC4.

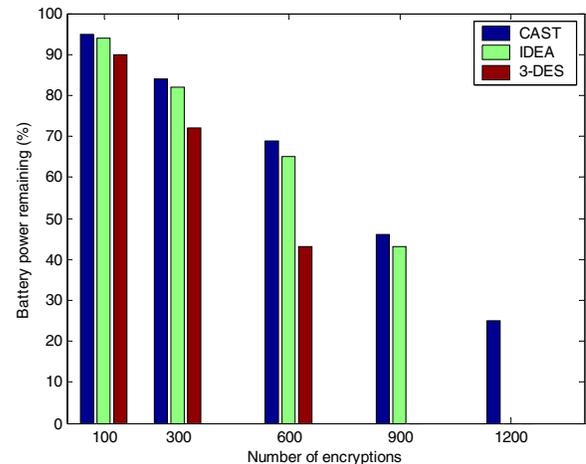


Figure 1 Fall in Battery Power with Symmetric Key Encryption [2].

C. Energy Efficiency Communication Protocols

In the literature, it is possible to find several works on the energy efficiency of transmission protocols over wireless networks at different layers. In [16], Feeney performed measurements of energy consumption of the IEEE 802.11 network interface. It is shown that the significant cost of energy consumption is due to the overhead of the 802.11 MAC protocol for point-to-point links. For sending, a station has to send an RTS packet before sending a data packet, and for receiving, a station needs to send a CTS packet before receiving and an ACK packet after receiving the data packet. In [17], Zorzi and Rao studied the energy consumption of different variants of a TCP protocol. It is shown that the congestion control algorithm of TCP can save energy by backing off when an error burst occurs during the transmission, but it worsens the transmission throughput. A comprehensive survey of energy efficient communication protocols is available in [4]. With the exception of [2] where the idea is suggested, there is no known work on energy efficient security protocols.

III. EXPERIMENTAL DESIGN

For our experiment, we use a laptop with a mobile Pentium III 700 MHz CPU, in which performance data are collected. In the experiments, the laptop encrypts a 5.5 MB file using RC4 and AES encryption algorithms from OpenSSL version

0.9.7a [18]. In our experiment, several performance metrics are collected: encryption time, CPU process time, and CPU clock cycles. The *encryption time* is the time that an encryption algorithm takes to produce a ciphertext from a plaintext. Encryption time is used to calculate the *throughput* of an encryption scheme that indicates the speed of encryption. The throughput of the encryption scheme is calculated as the total plaintext in bytes encrypted divided by the encryption time.

The *CPU process time* is the time that a CPU is dedicated only to the particular process for calculations. It reflects the load of the CPU. The more the CPU time is used in the encryption process, the higher is the load of the CPU. The *CPU clock cycles* is a metric reflecting the energy consumption of the CPU while operating on encryption operations. Each cycle of CPU will consume a small amount of energy.

For computation of the energy cost of encryption, we use the same techniques as described in [19]. We present a basic cost of encryption represented by the product of the total number of *clock cycles* taken by the encryption and the *average current* drawn by each CPU clock cycle. The basic encryption cost is in unit of *ampere-cycle*. To calculate the total energy cost, we divide the ampere-cycles by the clock frequency in *cycles/second* of a processor; we obtain the energy cost of encryption in *ampere-seconds*. Then, we multiply the ampere-seconds with the processor's operating voltage, and we obtain the energy cost in Joule.

To calculate the energy cost, we measure the clock cycles by using an instruction set to set and read the total number of cycles taken by encryption from a register. By using the cycles, the operating voltage of the CPU, and the average current drawn for each cycle, we can calculate the energy consumption of cryptographic functions. For example, on average, each cycle consumes approximately 270 mA on an Intel 486DX2 processor [19] or 180 mA on Intel Strong ARM [22]. However, currently we could not find any energy consumption benchmark for an Intel Pentium III 800 MHz which is used in our measurements; we assume it is close to 200 mA. For a sample calculation, with a 700 MHz CPU operating at 1.35 Volt, an encryption with 20,000 cycles would consume about 5.71×10^{-3} mA-second or 7.7 μ Joule.

IV. RESULTS

A. Encryption Throughput

In Figure 2, we show the encryption throughput of AES and RC4 algorithms. The encryption is performed with different data packet sizes. With a small packet size, AES tends to perform better than RC4. However, with increasing data packet size, the throughput of RC4 is far better than that

of AES. This shows that RC4 is more efficient than AES in encrypting large data blocks. With different key sizes, AES tends to have close performance while RC4 performance is likely independent of the key size. However, a longer key size would provide stronger security of data [20]. Thus, it is preferable to use a long key size to provide data confidentiality without trading off the encryption throughput.

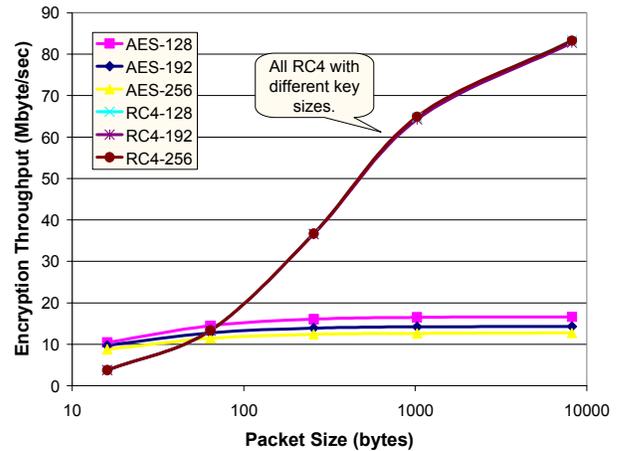


Figure 2: The encryption throughput of AES algorithm with varied key sizes

B. CPU Work Load

In Figure 3, we show the performance of RC4 and AES algorithms in terms of sharing the CPU load. With a small block size, RC4 tends to acquire a far greater CPU time for its processing, and the AES tends to consume much less time. However, RC4 is operating using less CPU processing time and reducing the work load on the CPU when it encrypts large data blocks. Thus, AES is suitable for a device which has low processing power (such as wireless devices) to encrypt small size packets.

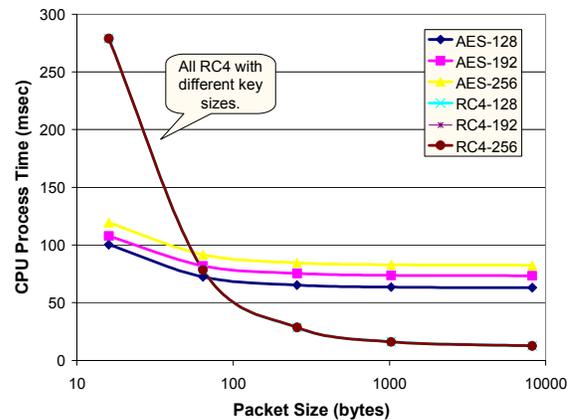


Figure 3 The process CPU time of RC4 and AES with varied key size

C. Energy Cost

In Figure 4, we show the energy cost of each encryption algorithm for different data block sizes. It is shown that AES consumes as little as three times less energy than RC4 when encrypting small data blocks. In contrast, the RC4 consumes less energy than AES for larger data blocks.

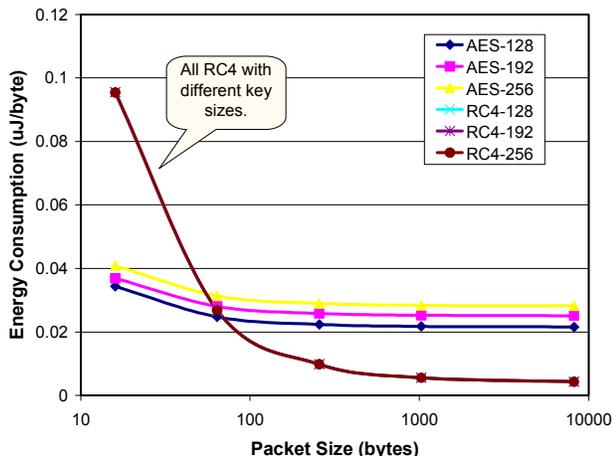


Figure 4 The energy cost of RC4 and AES with varied key size.

D. Encryption Key Size Variation

In Figure 5, we show the energy cost of each algorithms as a function of different key sizes that are used to encrypt packets with a total of one MBytes. We see that varying the key size of AES slightly increases the energy cost. However, RC4 key size variation does not have any effect on energy consumption.

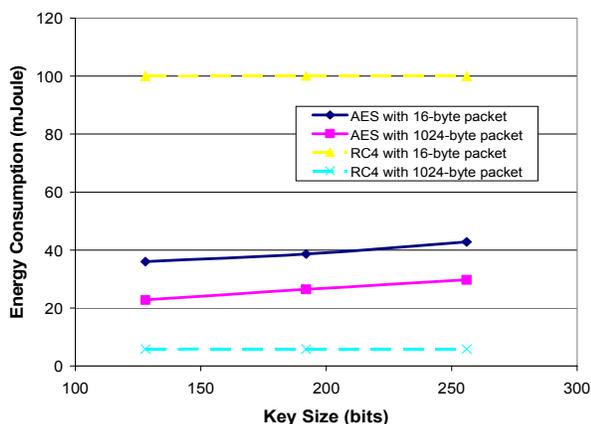


Figure 5 The energy cost of encryption with varied key size

We can conclude from these results that in 802.11 WLANs (where RC4 and AES can be expected to be used), we can save energy by using AES to encrypt small packets such as an 802.11 ACK which is about 14 bytes long, beacon packets

which are about 72 bytes long, and other short 802.11 management packets. To provide strong security and save energy, we could fragment a long packet into smaller packets and use AES to encrypt them. Smaller packets are often less susceptible to wireless channel errors, and hence, we can save much more energy. Of course, the fragmentation would give significant energy efficiency, but it will lower transmission throughput. It is preferable to use RC4 to encrypt data packets whose sizes are about 100 bytes or more on average before transmitting them. In addition, AES would also be appropriate for short probe packets for estimating the channel conditions. However, there are security implications beyond the simple conclusions made here. For instance, if both RC4 and AES used the same key, if RC4 was broken and the key compromised, AES would also be broken. The relationship between security levels, protocols and energy consumption is part of our ongoing work.

V. CONCLUSION AND FUTURE WORK

We have evaluated the performance of RC4 and AES encryption algorithms in this paper. The performance metrics were encryption throughput, CPU work load, energy cost and key size variation. Experiments show that the RC4 is fast and energy efficient for encrypting large packets. However, AES was more efficient than RC4 for a smaller packet size. From our results, it appears that we can save energy by using a combination of RC4 and AES to provide encryption for any packet size. The tradeoffs with security are not completely clear.

For our future work, we are studying the distribution of packet sizes of packets typically transmitted and received by a wireless device over a wireless LAN. We used a wireless sniffer to capture 802.11 network packets over one hour and obtained their packet size distribution of 5 MB captured packet data as shown in Figure 6. It is shown that most of packets have a small size between 64 -127 bytes and they are control and management packets. The data packet is typically about 1024 bytes which also depends on the maximum transfer unit (MTU) size. In our future research, we are looking at a hybrid security protocol that can use both RC4 and AES together to provide an energy efficient security scheme for 802.11 WLANs.

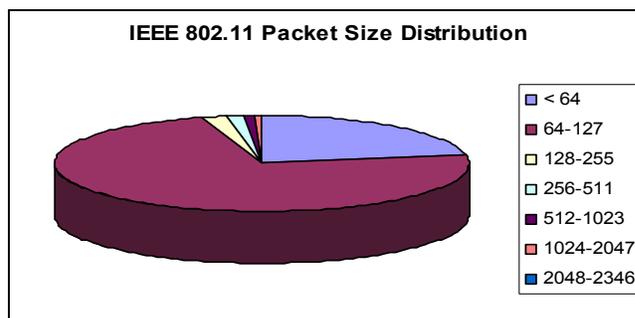


Figure 6 IEEE 802.11 Packet Size Distribution

ACKNOWLEDGMENT

This research has been partially supported by the NIST CIP grant No. 60NANB1D0120 titled "A Survivable and Secure Wireless Information Architecture."

REFERENCES

- [1] D.R. Stinson, *Cryptography Theory and Practice*, CRC Press, 1995
- [2] N. Ruangchajitupon, P. Krishnamurthy, "Encryption and Power Consumption in Wireless LANs," *The Third IEEE Workshop on Wireless LANs*, September 27-28, 2001, Newton, Massachusetts
- [3] B. Schneier, *Applied Cryptography*, John Wiley & Sons, Inc., 1996.
- [4] C.E. Jones, and et. al. "A Survey of Energy Efficient Network Protocols for Wireless Networks," *Wireless Networks*, 7, 343-358, 2001.
- [5] IEEE P802 working group, P802.11i Draft Supplement to Standard for Telecommunications and Information Exchange Between Systems- LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Specification for Enhanced Security, November 2002.
- [6] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks - A Unified Approach*, Prentice Hall, 2002
- [7] B. Schneier and D. Whiting, "Fast Software Encryption: Designing Encryption Algorithms for Optimal Software Speed on the {Intel Pentium} Processor," *Lecture Notes in Computer Science*, vol. 1267, pages 242-259, 1997.
- [8] IEEE P802 working group, IEEE P802.11 Standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999 Edition.
- [9] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," *In Eighth Annual Workshop on Selected Areas in Cryptography*, Toronto, Canada, Aug. 2001, pp. 1-24.
- [10] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," <http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf>.
- [11] M.A. Viredaz and D.A. Wallach, "Power Evaluation of a Handheld Computer: A Case Study," *WRL Research Report*, 2001/1.
- [12] N.R. Potlapally, et. al., "Optimizing Public-Key Encryption for Wireless Clients," *International Conference on Communications (ICC)*, May 2002.
- [13] Y.W. Law, and et. al., "Assessing Security-Critical Energy-Efficient Sensor Networks," *IFIP WG 11.2 Small Systems Security Conf.*, Athens, Greece.
- [14] L. Yuan and G. Qu, "Design Space Exploration for Energy-Efficient Secure Sensor Network," *In proceeding of the 13th IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP'02)*, July 17-19, 2002, San Jose, California.
- [15] R. Karri and P. Mishra, "Optimizing the Energy Consumed by Secure Wireless Sessions – Wireless Transport Layer Security Case Study," *Mobile Networks and Applications*, 8, 177-185, 2003.
- [16] L.M. Feeney, and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," *In Proceedings of IEEE Infocom*, Anchorage AK, April, 2001
- [17] M. Zorzi and R. Rao, "Energy Efficiency of TCP," *In Proceedings of the 7th International Workshop on Mobile Multimedia Communications*, 1999, San Diego, California.
- [18] OpenSSL Software Distribution, <http://www.openssl.org/>
- [19] K. Naik, D. S.L. Wei, "Software Implementation Strategies for Power-Conscious Systems," *Mobile Networks and Applications*, 6, 291-305, 2001.
- [20] A.K. Lenstra and E. R. Verheul, "Selecting Cryptographic Key sizes", *Journal of Cryptology*, vol. 14, no. 4, pp. 255-293, 2001.
- [22] A. Sinha and A.P. Chandrakasan, "JouleTrack- A Web Based Tool for Software Energy Profiling," *Proceedings of the 38th Design Automation Conference, DAC 2001*, Las Vegas, NV, USA, pp. 220-225.