Benchmarking Security Computations on Wireless Devices

Divyesh Shah   Sheng Zhong

Department of Computer Science and Engineering
University at Buffalo
{dpshah,szhong}@cse.buffalo.edu

Abstract

Many applications using cryptographic solutions on wireless devices choose one cryptographic technique over another with the assumption of it being more battery power efficient than the other. We present extensive benchmark test results for various cryptography techniques measuring the battery consumption of each technique on wireless devices in terms of number of computations per unit percent battery drain so that many of these assumptions could be better supported with strong evidence. The benchmark tests are performed on two wireless devices: a HP iPaQ PDA and a Dell Inspiron notebook. The cryptography techniques considered include DES, RC2, Rijndael , RSA encryption, MD5 and SHA hashing, and DSA, RSA signature algorithms. We present analysis of our results and a very useful comparison of these techniques enabling future research efforts to make a better decision while selecting cryptographic methods based on concrete results.

1 Introduction

In recent years, cryptographic solutions have been proposed for many applications involving wireless devices, which includes key distribution in wireless sensor networks [12,13,18,19,20] and incentive based routing in mobile ad hoc networks [1-11]. Security issues in ad hoc networks and sensor networks have been addressed using public-key cryptography [1,2,3,4,5], as well as symmetric key cryptography [6,7,8].

One common property between wireless sensor and ad hoc networks is wireless devices with limited battery capacity and proposed protocols strive to achieve minimal battery consumption. To achieve this, protocol designers have had to choose one cryptographic method over another on the premise that one is computationally efficient than other considering the battery drain [eg., 9,12]. However, generally speaking, such assumptions have not been supported by strong evidence and in many cases these assumptions rule out the possibility of considering some cryptographic techniques for a given problem.

In this paper, we attempt to provide a practical view of how expensive each cryptographic technique is on wireless devices in terms of the battery consumption. We hope that these results would help future protocol designers make a better decision when considering various cryptographic techniques for a similar application where the goal is to save on battery consumption.

The remainder of the paper is organized as follows: We describe the benchmarking environment and certain assumptions in Section 2. In Section 3, we discuss the test parameters for symmetric encryption benchmark tests and present our results and analysis. Section 4,5 and 6 follow the same structure as for Section 3 for hashing techniques, digital signature algorithms and asymmetric encryption, respectively. Finally, in section 7 we conclude with pointers to possible future work.

2 System Setting and Assumptions

We choose two most typically used wireless devices - a PDA and a laptop. We conducted the benchmark tests on a HP iPAQ hx4700 series Pocket PC 2003 PDA(Personal Digital Assistant) and a Dell Inspiron 6000 notebook(laptop). Detailed specifications for the PDA are: 64 MB primary memory, 624 MHz Intel PXA270 processor, 1800maH Lithium-Ion battery and Windows Mobile Operating System. Specifications for the laptop are: 512MB primary memory, 1.7GHz Intel Pentium M Processor, a 6-cell Lithium Ion battery (53 WHr) and Windows XP Media Edition Operating System.

Another parameter that is of importance is the static/idle discharge time for the two computing devices. The laptop static discharge time was 110 minutes while the same being 340 minutes for the PDA.

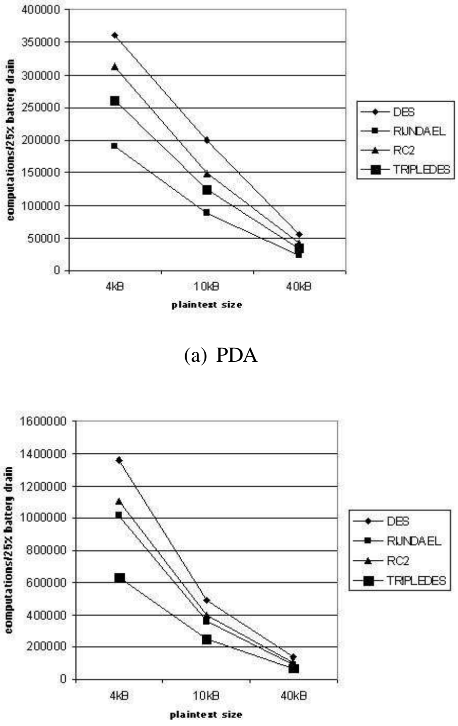
The cryptographic library used for the benchmark tests was Windows (.NET Framework) System.Security.Cryptography API. The reason for this selection was that we wanted to have a fair comparison by using similar cryptographic libraries on both devices so that the impact of the implementation differences on both devices is minimal. The tests were programmed in C# using Microsoft Visual Studio 2005 and the .NET (and .NET Compact)Framework.

We compute the number of cryptographic computations performed by each of these wireless devices for a battery drain of 25%. Most of the tests are repeated to work with plaintexts of different size to help us also understand the effect of message size on the power drain for a particular computation.

3 Symmetric Encryption

We conducted benchmarking tests for the following widely used symmetric encryption algorithms: Data Encryption Standard(DES), Rijndael, RC2 and TripleDES. All the tests were conducted using three different plaintexts of size 4kB, 10kB and 40kB. The DES implementation used a key size of 64 bits. All other results are based on 128-bit key size. The reason for the difference is that DES is a fixed- block size cipher technique with 64 bit keys and we did not wish to restrict other encryption techniques to 64 bit key size but to rather benchmark them at key sizes that are normally used in practical implementations.

Figure 1: Symmetric Encryption Algorithms



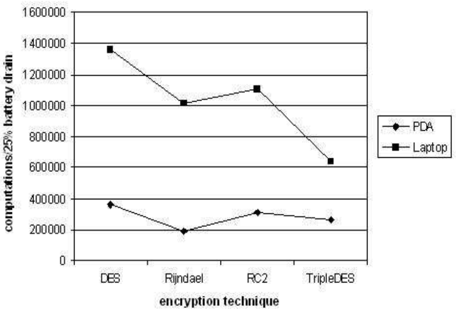
(a) PDA

(b) Laptop

The results clearly indicate that 64-bit DES is the least expensive symmetric encryption operation among these. However, the security of DES has come under question of late as small key size is vulnerable to even brute force attacks. For some other more efficient attacks refer [15,16]. Thus, RC2 seems to be the next best candidate for saving on battery power.

An interesting result from this is that on the PDA, TripleDES encryption algorithm seems to perform much better than Rijndael, it is not the same case for the laptop where Rijndael encryption is more battery efficient than TripleDES.

Figure 2: Symmetric Encryption PDA-Laptop Comparison



From the raw results in Figure 2, we can see that the laptop performs at least twice as good as the PDA. However, considering the difference between the static discharge times for both devices we can derive another set of results(Table 1) which show that the laptop outperforms PDA by a factor of 8-10 for symmetric computations.

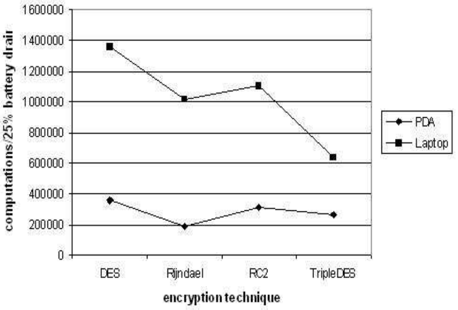


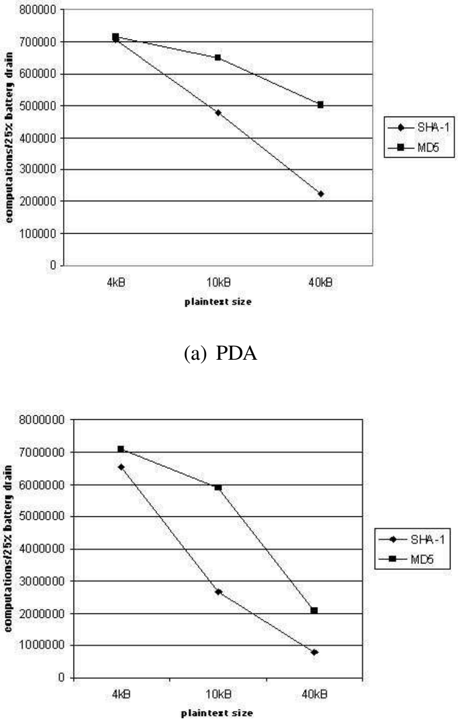
Table 1: Symmetric encryption (derived results)

	PDA		Laptop	
	4KB	40KB	4KB	40KB
DES	361150	56428	4199321	414719
Rijndael	189958	18850	3138311	290582
RC2	313618	42054	3419083	325534
TripleDES	251850	32942	1958925	193974

4 Hashing Algorithms

We conducted benchmark test for two most widely used hashing algorithms SHA-1 and MD5. Once again, these tests were repeated for plaintext sizes of 4, 10 and 40 kB. The results are shown in Figure 3.

Figure 3: Hashing Algorithms



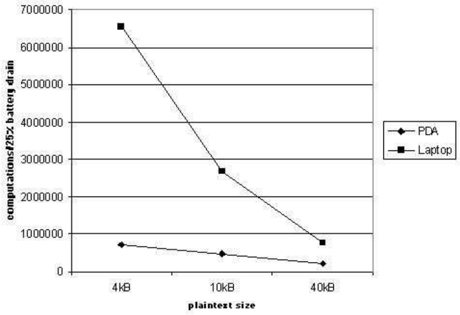
(a) PDA

(b) Laptop

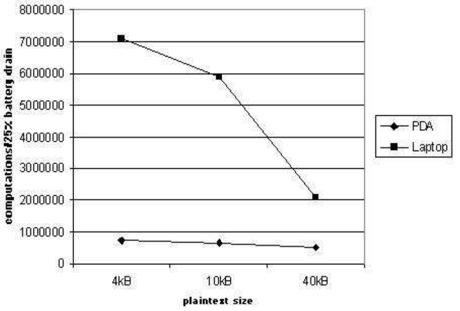
For smaller message sizes, both algorithms perform nearly equally well. However, with the increase in the message size, MD-5 clearly starts outperforming SHA-1 on both platforms.

Thus, for larger messages(10k and above) MD-5 saves a lot on battery consumption.

Figure 4: Hashing Algorithms PDA-Laptop Comparison



(a) SHA-1



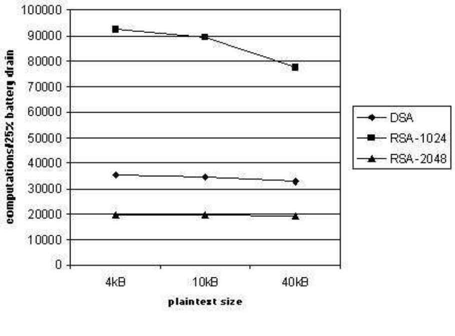
(b) MD5

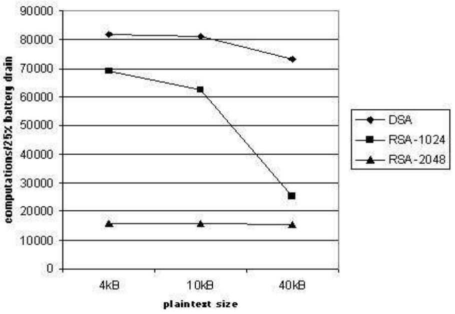
An interesting trend is noticed in the raw data results (Figure 4). For smaller sizes the laptop computations per % battery drain are better by almost a factor of 10 but this drops down to a factor of 3-4 as the message size increases.

5 Digital Signature Algorithms

We performed benchmark tests for DSA signature algorithm with 1024-bit keys and RSA signature with 1024-bit and 2048-bit keys on both devices.

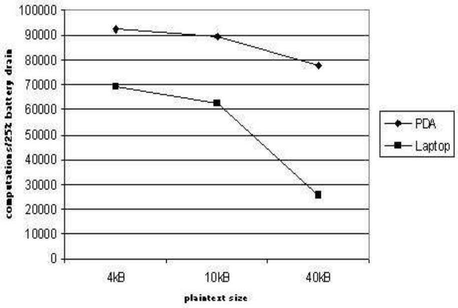
Figure 5: Digital Signature Algorithms



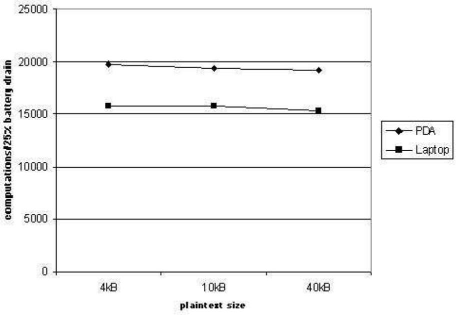


From Figure 5, we can see that compared to DSA(1024-bit key), RSA(1024-bit key) consumes less battery power.

Figure 6: Digital Signature Algorithms PDA-Laptop Comparison(1)



(a) RSA-1024(raw results)



(b) RSA-2048(raw results)

A very interesting phenomenon is noticed in Figure 6 where the PDA outperforms the laptop for RSA signature algorithm when viewing the raw test results. For 1024-bit key size the PDA performs much better than the laptop for larger plaintexts whereas the difference is almost constant for the 2048-bit case.

Table 2: Digital Signature Algorithms PDA-Laptop Comparison(2) - derived results

	PDA		Laptop	
	4KB	40KB	4KB	40KB
DSA	35182	32648	253708	226205
RSA-1024	92512	77648	213442	78540
RSA-2048	19756	19162	48913	47337

On factoring in the static battery drain, the laptop is better than PDA for both 1024-bit and 2048-bit key RSA signature implementation. But, for 1024-bit key size the laptop is only slightly better than PDA for larger plaintexts even after considering the static drain ratio. Also, on the PDA, RSA with 1024-bit keys is more battery efficient than DSA.

But when we implemented RSA using a supplementary BigInteger Class in C# [21] on both the laptop and PDA, the results showed the laptop performing better than PDA even for

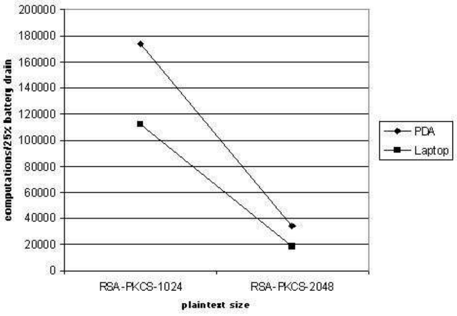
128-bit key case. This means that the alarmingly low values in the raw results for the laptop compared to the PDA were only due to implementation details and not an implementation-independent phenomenon.

Another interesting result from this data set is that for larger key sizes the difference in the number of computations per unit battery drain is marginal even for increasing plaintext sizes.

6 Asymmetric Encryption

We performed benchmark tests for RSA asymmetric encryption algorithm with 1024-bit and 2048-bit keys.

Figure 7: Asymmetric Encryption PDA-Laptop Comparison - Raw results



A trend similar to the RSA signature algorithms is noticed here, where the laptop lags behind the PDA for smaller key-sizes. However, on factoring in the static battery drain, laptop does better for both key-sizes.

7 Conclusion

This paper presents an extensive comparison of widely-used cryptographic techniques based on their battery consumption on wireless devices. With the help of the results and analysis presented here researchers will be in a better position to decide when choosing one cryptographic technique over another for application in above mentioned research. These results can now be used to analyze existing protocols and determine how expensive it really is in terms of battery drain and be used to support or disprove their earlier claims about one method being more expensive than another.

Possible future work in this direction could be similar work comparing different cryptographic libraries like Crypto++, OpenSSL, Java Security package, etc. This would serve as a useful guide for selection of cryptographic techniques for practical implementation. Also, results for more types of hardware platforms would be useful as well.

8 References

[1] L. Zhou and Z. Haas, Securing ad hoc networks, IEEE Network Magazine, vol 13,# 6, Nov/Dec 1999.

[2] J. Hubaux, L. Buttyan, and S. Capkun, The quest for security in mobile ad hoc networks, in Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2001), 2001.

[3] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, Providing robust and ubiquitous security support for mobile ad-hoc networks, in ICNP, 2001, pp. 251260.

[4] M. G. Zapata, Secure ad-hoc on-demand distance vector (SAODV) routing, IETF MANET Mailing List, Message-ID: 3BC17B40.BBF52E09@nokia.com, Available at ftp://manet.itd.nrl.navy.mil/pub/manet/2001-10.mail, October 8, 2001.

[5] H. Luo, P. Zefros, J. Kong, S. Lu, and L. Zhang, Self-securing ad hoc wireless networks, Seventh IEEE Symposium on Computers and Communications (ISCC 02), 2002.

[6] Y.-C. Hu, A. Perrig, and D. B. Johnson, Ariadne: A secure on-demand routing protocol for ad hoc networks, Department of Computer Science, Rice University, Tech. Rep. TR01-383, December 2001.

[7] S. Basagni, K. Herrin, E. Rosti, and D. Bruschi, Secure pebblenets, in ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001), October 2001, pp. 156163.

[8] P. Papadimitratos and Z. Haas, Secure routing for mobile ad hoc networks, in SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), January 2002.

[9] Naouel Ben Salem, Levente Buttyan, Jean- Pierre Hubaux, Markus Jakobsson, A charging and rewarding scheme for packet forwarding in multi-hop cellular networks, in proceedings of MobiHoc 2003, pp. 13-24.

[10] Sheng Zhong, Jiang Chen, and Yang Richard Yang, Sprite: A simple, Cheatproof, Credit-based System for Mobile Ad hoc Networks, in Proceedings of IEEE Infocom ’03, San Francisco, CA, April 2003.

[11] S. Zhong, L. Li, Y. Liu, Y. R. Yang. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks, MOBICOM 2005, Germany

[12] Security in wireless sensor networks, Mayank Saraogi

[13] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. SPINS: security protocols for sensor networks. Wireless Networks, 8(5):521534, 2002.

[14] Microsoft Studio Developer Network Library, 2005

[15] E. Biham and A. Shamir, Differential cryptanalysis of the full 16-round DES, Advances in Cryptology - Crypto ’92, Springer-Verlag (1993), 487-496.

[16] M. Matsui, The first experimental cryptanalysis of the data encryption standard, Advances in Cryptology - Crypto ’94, Springer-Verlag (1994), 1-11.

[17] Chew Keong TAN, C# BigInteger Class, http://www.codeproject.com/csharp/BigInteger.asp.

[18] G. Jolly, M.C. Kuscu, P. Kokate, and M. Younis., A Low-Energy Key Management Protocol for Wireless Sensor Networks., IEEE Symposium on Computers and Communications(ISCC’03)., June 2003

[19] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, J. Zhang., Fast Authenticated Key Establishment Protocols for Self-Organizing Sensor Networks., WSNA’03, September 2003

[20] S. Zhu, S. Setia, and S. Jajodia., LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks., CCS’03, October 2003