Data Protection and Secure Encryption using RSA and DES with IOT Devices

Steven Granquist
Adviser: Shimin Li
Computer Science, Winona State University, 175 W Mark St, Winona, MN 55987, United States
1- 651-900-3654
[1]sgranquist15@winona.edu
[2]shiminli@winona.edu

Abstract— **This paper examines different methods of encryption and Data Protection on Internet of Things (IoT) devices. As IoT devices become more popular, they need to be properly secured. Specific comparisons are made between RSA and DES encryption and how those methods handle the processing of information and data across different types of devices. The goal of the paper is to determine which method is the most efficient and computationally effective to be used with IoT devices. RSA and DES were imitated using a Raspberry Pi, a Laptop, and a Desktop PC. OpenSSL was used to encrypt those devices with RSA and DES. Based off the experiments done, RSA encryption is the method that would be best suited for implementation on IoT devices from a computational perspective.**

Keywords— **IoT, Encryption, Data Protection, RSA, DES, Security.**

I. Introduction

Internet of Things (IoT) devices are becoming more prevalent as we step into the future of technology. These devices are becoming ingrained in culture and the daily life of millions of people. Where there is a large amount of people, there is bound to be a security issue. IoT devices post a large amount of risks in their current state, many devices are un-patchable and incredibly vulnerable to different types of intrusions.

According to predictions from Statista, the total number of IoT devices will reach around 31 billion devices in 2020. By 2025, this number is estimated to reach around 75 billion [1]. This massive variety of interconnected devices serves to be a potential threat if not handled properly. “Everything from the physical or virtual world will possible be connected by the IoT. Connectivity between the things shall be available to all with low cost and may not be owned by private entities. For IoT, intelligent learning, fast deployment, best information understanding and interpreting, against fraud and malicious attack, and privacy protection are essential requirements” [2].

Properly encrypting these devices is the next step we must take to ensure we set ourselves up for success in the future. In a study done with the most popular IoT devices, and average of 25 vulnerabilities were found [3]. As we use more and more IoT devices over time, we need to make sure we are taking the proper precautions to protect and properly secure these devices and their users, our devices should have 0 vulnerabilities for them to be deemed secure and ready for use.

There are many ways these devices can be secured, and many types of encryption currently exist that could possibly be used to solve this problem. One problem that arises when trying to address this issue is the relationship that all these devices have with and to each other. Many IoT devices use a variety of software, operating systems, and protocols to perform their functions, this makes it harder to develop a universal solution, therefore, harder to enable a standard set of encryptions. There are many different encryption styles that can be considered as a possible solution for this issue, however, this study will focus specifically on two different methods and how they compare to each other. Based on these comparisons, we can determine which method would be ideal to implement in environments similar or identical to IoT devices. The two methods of interest being explored in this paper are RSA and DES. While both methods are very different, comparisons can still be made between them. Utilizing OpenSSL, RSA and DES encryption was implemented and extensively tested. For RSA, different key lengths are

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee if copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Proceedings of the 18thWinona Computer Science Undergraduate Research Seminar, May 1, 2019, Winona, MN, US.

compared. For DES, 4 different methods of block ciphers will be compared. When comparing the key lengths, results can be a little bit misleading, [4] an accurate comparison between two types of encryption may need to use two completely different key lengths, in order to be fair to the different lengths of RSA. Lower levels of RSA, such as 1024 and 2048 length methods, can be compared to the DES methods shown later in the paper because of the similar level of security. Although RSA and DES are two different types of encryption, they still can be accurately compared to each other.

II. HYPOTHESIS

How do the different methods perform on different devices of varying hardware and software configurations? Do some OS's handle the encryption better than others?

Between RSA and DES, which method could be the best solution to properly secure and protect IoT devices and their critical data? And how can we verify this method is better than the other?

III. METHODS

The two methods of encryption chosen, RSA and DES, have been implemented to run on multiple devices with a variety of hardware and software constraints. These devices consist of the following; a Raspberry Pi running Raspbian OS, a HP Laptop running Debian on Windows 10, and a Desktop PC running Ubuntu. With these devices, the two methods can be tested with a variety of constraints and options to narrow down which encryption style offers the best performance and overall protection.

The devices vary in operating system, which was considered. While a bias towards a certain OS was likely to exist and be found [5], no observable evidence was found to prove this point.

A. Experiment on Device

For an individual experiment on a device, the process was as follows: In a project file, five different sizes of files were created. These files each had a different amount of characters within them, this allows an average to be computed for the overall effectiveness of the method. The five different ranges of character selected were, 6,000, 18,000, 42,000, 63,000 and 280,000. Using OpenSSL, the encryption method was specified, either RSA or DES. The above five files were encrypted, then decrypted. Time to complete both tasks were measured, in addition to the total overall time to complete. The encryption and decryption process were done five times for each of the five files. Given the results from the five trials, an average was computed, this average was said to be the average encrypt time and the average decrypt time. A total of 25 trials for each file, were summed and averaged to determine the average time for a given method to run across a variety of files.

The above steps were followed using RSA (1024), RSA (2048), RSA (3072) and RSA (4096) for RSA. These key values selected are in bits. For DES, DES, DES-CBC, DES-ECB, and DES-EDE block ciphers were tested. These RSA and DES methods were tested on all three of the test devices. The results can be viewed below in the results section.

B. DES

Pertaining to DES, four different block cipher methods were utilized. A brief explanation is needed to describe the differences between the four DES methods. DES uses two fundamental attributes of cryptography: Substitution (confusion) and transposition (Diffusion). [6]. DES is a block cipher; it encrypts the data in a block of 64 bits. The key length is 56 bits, after certain bits are discarded as they are unneeded [6].

C. DES-CBC

DES-CBC uses DES as the base of the algorithm, with Cipher Block Chaining, Like DES, DES-CBC uses a 64-bit key. 56 of those 64 bits are used directly, the other 8 are odd parity bits [7]. DES-CBC also required a 64-bit Initialization Vector (IV). For each encrypted message or data, a new IV is generated. This IV is sent with the message within the header field, included with the 64-bit key [7].

D. DES-ECB

DES-ECB uses DES as the base of the algorithm, with what is described as an Electronic Codebook, hence the ECB. In this mode, the data being encrypted is divided into different, individual variable-sized blocks. These separate blocks are then encrypted on their own. The blocks do not need each other in order to decrypt the encrypted message, a specific block just needs its own individual key to decrypt its information [8]. All the blocks hold the entire original message, but they each hold their piece of the message. Later, these blocks will be recombined to remake the original message. While this method does extra encryption within the message, it is not very good at it. Most instances of DES-ECB do not hide data patterns very well. Below is an example of how clear the message can be viewed despite being encrypted.

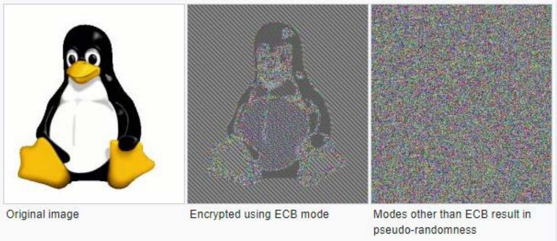

Fig. 1 DES-ECB example [8].

E. DES-EDE

DES-EDE uses an Encrypt-Decrypt-Encrypt, EDE, multiple encryption mode [9]. It uses multiple pairs of 64-bit keys, instead of just one like normal DES. The pair of keys is later passed in the header field to be used later [9].

F. RSA

Using the RSA key lengths was easy to implement, unlike what was originally predicted. As expected, with the RSA key length increase, computation time also increased. RSA 2048 was previously considered to be the standard for RSA encryption, that is now moving more towards the large key lengths, 3072 and 4096. The NSA has determined the RSA with a minimum key length of 3072 is deemed safe enough to protect information up to Top Secret information [10]. This decision was made based off the NIST's document, SP 800-56B Rev. 1, published to demonstrate the level of security that RSA with 3027 and 4096 provides [11].

RSA algorithm is an asymmetric based cryptography method [12]. This means that there are two different keys, the Public Key and the Private Key. An example of this relationship is as follows [12]:

1. A client (a web browser) sends its public key to the server and requests for some information.

2. The server encrypts the data using the client's public key and send the encrypted data.

3. The client receives this data and decrypts it.

Even though the client gives up its key, it is not helpful to anyone listening in hopes of stealing information, since the public key is not used to decrypt the information. This means that information can be safely sent without worrying if someone in the middle is intercepting the encrypted message [13]. However, RSA is not perfect, "methods such as brute-force are simple but lengthy and may crack a message, but not likely an entire encryption scheme. It cannot be verified that RSA is unbreakable… Despite years of attempts, no one has been known to crack the RSA algorithm" [14]. The reason this algorithm is so effective is because of how it is made. "RSA cryptosystem is based on the dramatic difference between the case of finding large primes and the difficulty of factoring the product of two large prime numbers" [15].

In the experiment, a Public and Private key were generated in order to encrypt and decrypt the original data file. The client and server were both on the same device, no actual transmission was made with the data files, but the files were encrypted with the Public/Private key pair.

Table 1. RSA Encryption

Device	**Type (in bits)**	**AVG Total Time (Seconds)**	**AVG Total Time (Milliseconds)**
Pi	1024	0.460	460
Pi	2048	0.471	471
Pi	3072	0.482	482
Pi	4096	0.489	489
Laptop	1024	0.165	165
Laptop	2048	0.168	168
Laptop	3072	0.175	175
Laptop	4096	0.184	184
Desktop	1024	0.732	73.2
Desktop	2048	0.756	75.6
Desktop	3072	0.796	79.6
Desktop	4096	0.843	84.3

Table 2. DES Encryption

Device	**Type (in bits)**	**AVG Total Time (Seconds)**	**AVG Total Time (Milliseconds)**
Pi	DES	0.9024	902.4
Pi	DES-CBC	0.9055	905.5
Pi	DES-ECB	0.9329	932.9
Pi	DES-EDE	1.0568	1056.8
Laptop	DES	0.316	316
Laptop	DES-CBC	0.3188	318.8
Laptop	DES-ECB	0.3246	324.6
Laptop	DES-EDE	0.3511	341.1
Desktop	DES	0.0916	91.6
Desktop	DES-CBC	0.0939	93.9
Desktop	DES-ECB	0.0946	94.46
Desktop	DES-EDE	0.1032	103.2

IV. Analysis

DES: Based on the result from the experiment, basic DES has been determined to be the standout choice between the DES methods as the most efficient method. Across all the test cases and devices, DES always had the lowest Average Total Time in both seconds and milliseconds. When DES times are compared to the other DES block cipher methods, DES beats them all in terms of runtime. In most test cases, DES-CBC came in a very close second, and seemed like a possible choice, however, after extra testing between DES and DES-CBC, DES proved to have the fastest average time. DES-EDE was the method that took the longest time on all three of the devices. This is caused by the fact that DES-EDE is a multiple encryption mode, with multiple 64-bit keys, which causes the average total time to be larger. While all the DES methods ran in under 1 second, DES is still the best choice out of the 4. Below is a graph that shows this relationship:

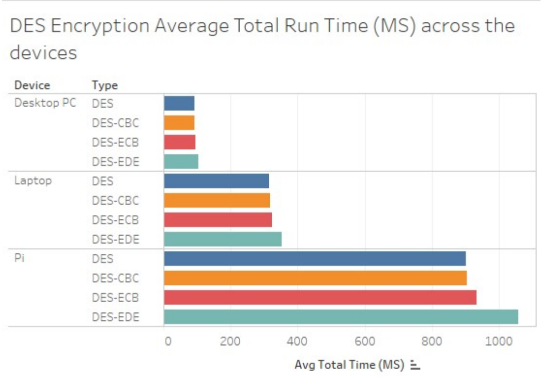

Fig. 2 DES Encryption Average Total Run Time

RSA: Based on the result from the experiment, all methods of RSA can be used as a viable choice as the most efficient method. Across all the test cases and devices, all key lengths in RSA had relatively low Average Total Times. In all cases, the results had a very similar relationship. Regardless of key length, the times for RSA were the same on each device. For the Desktop PC, all the tests resulted in a time (in MS) between 70-80. For the Laptop, all the tests resulted in a time (in MS) between 160-180. For the PI, all the tests resulted in a time (in MS) between 460-490. Technology with more processing power can run RSA more efficiently and obtain more smaller times. As the level of processing power goes down, there is an increase the Average Total Time range. Below is a graph that shows this relationship:

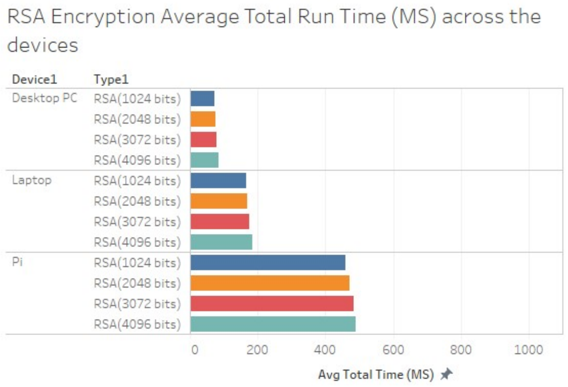

Fig. 3 RSA Encryption Average Total Run Time

RSA vs DES: When comparing RSA vs DES, it becomes clear that RSA is faster at all levels when compared to DES. Even though DES encryption is simpler than RSA, RSA still runs faster in every test case. In some cases, like the Desktop PC, the differences are very slim. In the Laptop and Pi tests, the differences are very drastic. For the Laptop and Pi, DES takes twice as long compared to RSA. It becomes clear when comparing the Average Total Times between RSA and DES, that RSA is the better choice in every test case. Based off this information, RSA encryption is the method that would be best suited for implementation on IoT devices from a computational perspective. Below is a graph that shows this relationship:

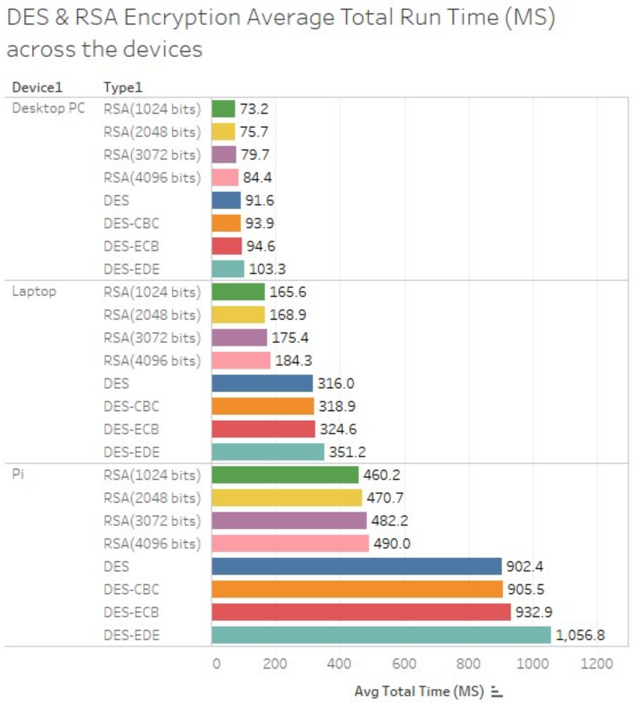

Fig. 4 RSA vs DES Encryption Average Total Run Time

V. Conclusion

Based on the results from the experiments, the best method to implement on IoT devices was able to be reliably determined. RSA encryption method is the standout choice based on the results from the experiments. RSA was the most efficient at runtime on each of the test devices. In every case, RSA performed faster than DES in terms of completion time. This means that in smaller IoT devices, RSA encryption is viable at initial setup of the device. This method is the most effective across the various hardware and would be good to use in environments with large transportation of data. There was not a bias between the OS's of the devices. This can be verified by the fact that all the test results followed a similar pattern. All DES methods were close between each other; however, RSA was significantly faster than DES. Based on all these parameters, it can be reliably determined that RSA is the best solution to properly secure and protect IoT devices and their critical data.

References

[1] "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)". Statista, Nov-2016. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities with China Perspective," IEEE Xplore, 09-Jul-2014. [Online]. Available: https://ieeexplore.ieee.org/document/6851114.

[3] E. Bertino, "Data Security and Privacy in the IoT," OpenProceedings.org, 15-Mar-2016. [Online]. Available: https://openproceedings.org/2016/conf/edbt/paper-a.pdf.

[4] M. Suarez-Albela, P. Fraga-Lamas, and T. M. Fernandez-Carames, "A Practical Evaluation on RSA and ECC-Based Cipher Suites for IoT High-Security Energy-Efficient Fog and Mist Computing Devices," ResearchGate, Nov-2018. [Online]. Available: https://www.researchgate.net/publication/328890429_A_Practical_Evaluation_on_RSA_and_ECC-Based_Cipher_Suites_for_IoT_High-Security_Energy-Efficient_Fog_and_Mist_Computing_Devices.

[5] K. Deshpande, P. Singh, "Performance Evaluation of Cryptographic Ciphers on IoT Devices," arxiv.org, 5-Dec-2018. [Online]. Available: https://arxiv.org/pdf/1812.02220.pdf.

[6] Nirm Aljeet Kaur, S. Sodhi, "Data Encryption Standard Algorithm (DES) for Secure Data Transmission," research.ijcaonline.org. [Online]. Available: https://research.ijcaonline.org/icaet2016/number2/icaet036.pdf

[7] "DES in CBC Mode (DES-CBC)", freesoft.org. [Online]. Available: https://www.freesoft.org/CIE/RFC/1423/2.htm

[8] "Block cipher mode of operation", wikipedia.org. [Online]. Available: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Electronic_Codebook_(ECB)

[9] "DES in EDE Mode (DES-EDE)", freesoft.org. [Online]. Available: https://www.freesoft.org/CIE/RFC/1423/8.htm

[10] National Cryptography Solutions Management Office, "Commercial National Security Algorithm Suite", nsa.gov, 19-Aug-2015. [Online]. Available: https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm

[11] Computer Security Division (Information Technology Laboratory), "Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography", nist.gov, Mar-2019. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Br1.pdf

[12] "RSA Algorithm in Cryptography", geeksforgeeks.org. [Online]. Available: https://www.geeksforgeeks.org/rsa-algorithm-cryptography/

[13] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", csail.mit.edu. [Online]. Available: https://people.csail.mit.edu/rivest/Rsapaper.pdf

[14] Evgeny Milanov, "The RSA Algorithm", math.washington.edu, 3-Jun-2009. [Online]. Available: https://sites.math.washington.edu/~morrow/336_09/papers/Yevgeny.pdf

[15] M. Preetha, M. Nithya, "A STUDY AND PERFORMANCE ANALYSIS OF RSA ALGORITHM", ijcsmc.com, 6-Jun-2013. [Online]. Available: https://www.ijcsmc.com/docs/papers/June2013/V2I6201330.pdf