Lightweight Cryptography

Shane Brown
Christopher Garrett

School of Computing, University of Utah

December 11, 2015

1 Introduction

In our project we compare 4 different algorithms that have been stated to be lightweight cryptography algorithms[1]. We will run implementations of them on two devices, recording many operational statistics such as time and memory usage. With this data we will then be able to discuss and conjecture about possible uses of the different algorithms.

2 Related Work

Many cryptographic algorithms have been proposed in the area of Lightweight cryptography. The question that we wish to explore is which one is best suited for certain applications. Lightweight cryptography can be defined and any cryptographic algorithm that has been designed or optimized for usage in constrained computational environments. These constrained environments can include items such as refrigerators, thermostats, stoplights, or RFID tags. These are environments were computational power, memory space, and/or electrical power are restricted. Many of the proposed algorithms attempt to reduce all three. It is noted that reducing computational power and memory usage will reduce power consumption. But not all applications are constrained by such demanding restricted. Some applications like appliances or Infrastructure can be a lot more flexible with power consumption than for example a RFID sensor node. These trade-off are what we wish to explore. How different are these algorithms? What potential performance trade-offs do these different algorithms have?
The algoithms that we have decided to look at are RC5, XTea, Simon, and Speck. All of these are Feistal strucuture based algorithms. RC5[7] is the oldest, proposed in 1994. And Simon and Speck are the newest, proposed by the NSA in 2013[2]. XTea[6] is unqiue because it was designed to have a small compiled footprint.

Table 1: The block and key sizes are in bits.

	Block	Key	# rounds
RC5	64	128	15
XTea	64	128	64
Simon	64	128	44
Speck	64	128	27

3 Adversary Model

The need for end-to-end security is one of the main argument for the need of lightweight cryptography. Our world is now flooding with embedded devices that connect to the Internet. They help us in many ways. Smart thermostats let us set our homes temperature from work or vacation. Sensors allow us to get real-time updates on the status of our infrastructure. If these devices are not secure then an advisory listening in on your communication with your thermostat can tell that you are not at home, and are a

perfect target for a robbery.
In our study we are not looking into any authentication, so our adversary model is one that can listen in and can hear all communication. In our study we are also not going to attempt to prove or disprove the security of these algorithms. We will use the keys,block, and rounds that are stated to be secure.
For example the RC5 algroithm has been broken with it running 12 rounds[3]. 18-20 is the recommended number of rounds. XTea has also been broken with it running at 27 and 36 rounds[4][5]. We are running the XTea algorithm at the recommended 64 rounds.

4 Methodology

We selected four specific criteria in order to be able to accurately compare lightweight cryptographic algorithms to each other. Execution time, memory usage, compilation size, and power usage. Execution time is defined as the number of microseconds that given algorithm takes to complete a given task; in our tests it is the amount of time need to encryption one plaintext input of a specified length. Memory usage is the amount of RAM memory on the stack in bytes utilized by the algorithm during runtime. Compilation size is the size in bytes of the object (.o) file of the c++ implementation of the algorithm. Power usage is the estimated power usage of the algorithm given in watts-hours.

5 Experiment

For our experiment we tested the algorithms in two different environments. One environment was the recently released Raspberry Pi 2 model B, this was chosen due to it's availability and because it's resources are more restrictive than laptops and desktops thereby more closely resembling an IoT device. The second environment was a older laptop. The algorithms were each implemented in C++, compiled and measurements recorded. Code can be found on github (github.com/Chondor/LightWeightCryptoTest).

5.1 Raspberry Pi 2 Model B Testbed

Measurements were performed on a Raspberry Pi 2 Model B, which was released in February 2015. The Raspberry Pi 2 has a 900 MHz quad-core ARM Cortex-A7 CPU, 1 GB RAM, OS was load and running off of a SanDisk 32 GB Class 10 micro SD card, and powered by a standard 5 V 2 A power supply. The Ubuntu MATE version 15.10 operating system, which was built specifically for the raspberry pi 2, was loaded on the micro SD card and ran on raspberry pi 2. The C++ code was compiled using the GNU C compiler version 5.2.1 20151010, the c++ standard 11 option was used to compile the code.

5.2 Laptop Testbed

Measurements were performed on a older laptop. The laptop has an Intel Celeron 1.40 GHz processor, 1 GB of RAM. The Ubuntu version 14.04.3 LTS was loaded on the laptop. The GNU C compiler version 4.8.4 was used to compile the code on this system.

5.3 Results

The following graphs illustrate the results of our tests. The exact values of results are show in the appendix.

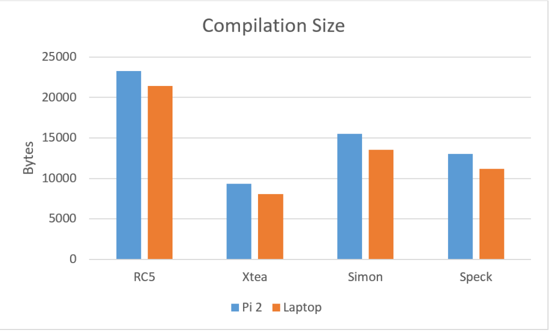

Figure 1:

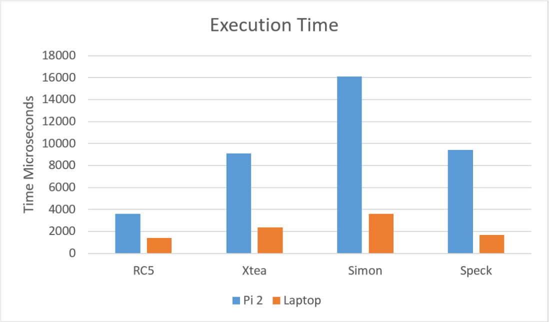

Figure 2:

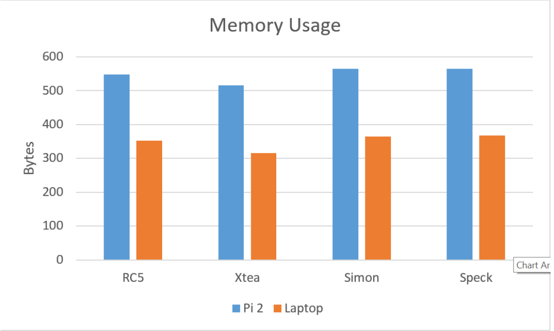

Figure 3:

6 conclusions

One of the first things that I notice when looking at the graphs was that the graph profiles for the Raspberry Pi and the laptop are roughly the same. This can lead us to surmise that the performance on one should roughly scale to another device. This property might allow for the preliminary testing and choosing of lightweight algorithms outside their target environment.
Another thing I noticed was that the memory usage for the 4 algorithms was about the same. This most likely comes from the fact that they are all the same type of algorithm, a Feistal type. More testing will need to be done to see if this is true or do most algorithm have the same or similar memory footprint. The difference in memory sizes between the Raspberry Pi and the Laptop are probably do to differences in cash sizes.

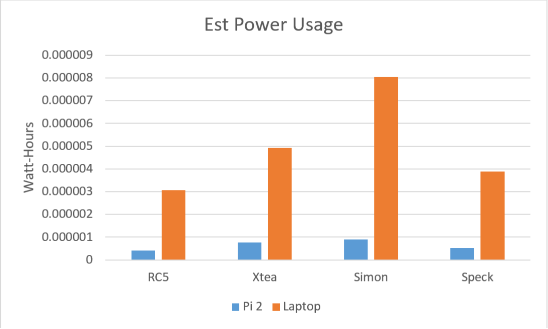

Figure 4:

The execution time and power usage seem to correspond. This is expected longer execution time means more power used. It seems to be more interesting to look at execution time and compilation size. We see that RC5 is the fasted, but it is also the largest. XTea and Speck are about the same in performance, with speck being sightly faster. But in size XTea is smaller. Simon has been stated to be optimized for a hardware implementation. Tests would have to be done on its target hardware class to see its true potential.
We did not seem to see the clear trade-offs that we were looking for. This is mostly likely because of the similarity in the cyphers that we choose to test. The one trade-off we did seem to see is between compilation size and execution speed(and thus power usage).

References

[1] Lightweight block ciphers. `https://www.cryptolux.org/index.php/Lightweight_Block_Ciphers#cite_note-NW97-83`. Accessed: 2015-11-24.

[2] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. The simon and speck families of lightweight block ciphers. *IACR Cryptology ePrint Archive 2013* (2013), 404.

[3] Biryukov, A., and Kushilevitz, E. Improved cryptanalysis of rc5. In *Advances in Cryptology?EUROCRYPT'98*. Springer, 1998, pp. 85–99.

[4] Ko, Y., Hong, S., Lee, W., Lee, S., and Kang, J.-S. Related key differential attacks on 27 rounds of

xtea and full-round gost. In *Fast Software Encryption* (2004), Springer, pp. 299–316.

[5] Lu, J. Related-key rectangle attack on 36 rounds of the xtea block cipher. *International Journal of Information Security 8*, 1 (2009), 1–11.

[6] Needham, R. M., and Wheeler, D. J. Tea extensions. *Report, Cambridge University, Cambridge, UK (October 1997)* (1997).

[7] Rivest, R. L. The rc5 encryption algorithm. In *Fast Software Encryption* (1995), Springer, pp. 86–96.

7 Appendix

Table 2: Compilation Size

System	Algorithm	Size (bytes)
Pi 2	RC5	23288
Pi 2	XTea	9324
Pi 2	Simon	15484
Pi 2	Speck	13012
Laptop	RC5	21436
Laptop	XTea	8068
Laptop	Simon	13536
Laptop	Speck	11208

Table 3: Execution Time

System	Algorithm	Exe Time (microseconds)
Pi 2	RC5	3606
Pi 2	XTea	9098
Pi 2	Simon	16115
Pi 2	Speck	9412
Laptop	RC5	1384
Laptop	XTea	2365
Laptop	Simon	3623
Laptop	Speck	1682

Table 4: Est. Power Usage

System	Algorithm	Power Usage (watt-hours)
Pi 2	RC5	4.00667E-07
Pi 2	XTea	7.58167E-07
Pi 2	Simon	8.95278E-07
Pi 2	Speck	5.22889E-07
Laptop	RC5	3.07556E-06
Laptop	XTea	4.92708E-06
Laptop	Simon	8.05111E-06
Laptop	Speck	3.87794E-06

Table 5: Memory Usage

System	Algorithm	Stack Usage (bytes)
Pi 2	RC5	548
Pi 2	XTea	516
Pi 2	Simon	564
Pi 2	Speck	564
Laptop	RC5	352
Laptop	XTea	316
Laptop	Simon	364
Laptop	Speck	368