

Electronics and Computer Science
Faculty of Physical and Applied Sciences
University of Southampton

Author: Lewis Smith

December 5, 2017

Low Power Hardware Accelerated Internet of Things Cryptography

Project Supervisor: Mark Zwolinski
Second Examiner:

A project progress report submitted for the award of MEng
Electronic Systems with Computer Systems

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Contents

Abstract	1
Contents	2
1 Introduction	3
2 Background Research and Literature	5
2.1 Asymmetric & Symmetric Cryptography	5
2.1.1 Asymmetric Key	5
2.1.2 Symmetric Key	6
2.2 Conventional Algorithms	7
2.2.1 Standardization	7
2.2.2 Other Algorithms	8
2.3 Lightweight Algorithms	8
2.3.1 PRESENT	8
2.3.2 PRINCE	8
2.3.3 SIMON & SPECK Algorithms	8
3 Progress	9
3.1 Hosted C	9
3.2 System Verilog	9
4 Future Plan	10
References	11
Appendix	12

Chapter 1

Introduction

Over the last few years there has been a shift in type of [devices](#) connected to the internet from just servers, PC's and later smartphones, to small embedded processors that can control many [devices](#). The idea of connecting such [devices](#) to internet has been dubbed "Internet of Things" or "IoT" and has the aim to make our lives simpler. These IoT [devices](#) can range from industrial applications, such as automated factories, to public infrastructure, like a smart electrical grid, or more homely [devices](#) in a smart connected home. Thus due to the wide range of products that a connected IoT device can be applied to improve the efficiency and/or usefulness, it has been predicted that billions of [devices](#) will be in use by [\[insert year/reference\]](#). This also means that the complexity of the [devices](#) varies greatly with simple light switches and even kettles being given the IoT treatment[\[insert reference\]](#), but at the other end of the scale a network of connected self-driving cars is being considered[\[insert reference\]](#).

The one thing that all of these [devices](#) have in common though is that they need to be secure as they communicate sensitive and private data through an open channel on the internet between the user and the device. Hence, to keep the potential adversaries from accessing the data and possibly controlling numerous connected [devices](#), maliciously or not, an encryption algorithm can be used with only the specified users having access to the decryption key. There are many encryption algorithms that perform this function and most can be implemented in both software and dedicated hardware such as a Application Specific Integrated Circuit (ASIC) or a Field Programmable Gate Array (FPGA). As a majority of IoT [devices](#) are implemented on small embedded processors which have limited resources so the hardware option might possibly be a better solution for IoT [devices](#). Dedicated hardware can also, theoretically, perform the algorithms faster and more accurately. However, due to the fact that most IoT devices are always on, power consumption is a very important factor when considering options for adding hardware accelerated encryption and for battery powered devices is often more critical than the actual encryption.

The goals of this project are to explore various encryption algorithms and compare their performance based on data throughput, accuracy, security and power consumption when implemented in software and hardware. To evaluate these parameters the same algorithms can be coded in C or C++ for the software versions and a Hardware Development Language (HDL) such as System Verilog can be first simulated

in ModelSim before programming a FPGA for the hardware version. These comparisons can then be used to match the algorithms to the appropriate IoT device as they all have different requirements for relative security level and power consumption, as for example a light switch does not necessarily need to be protected from the same level of attack as a set of digital locks or private data storage. In order for the hardware to work with IoT devices it will also need a communication protocol like I^2C or SPI to work with embedded processors, and possibly Ethernet or WiFi to act as the gateway to internet for the device. Some of these protocols are available on FPGA development boards but can be implemented in System Verilog code.

The scope of this project is therefore to develop encryption algorithms in System Verilog for a FPGA and compare their performance to the equivalent software before applying the hardware to actual IoT devices. Also if more time is available then a layout, based on the System Verilog synthesis, could be developed for ASIC fabrication in the future which could increase efficiency of the hardware due to the fact that only the required circuitry would be present on the chip.

Chapter 2

Background Research and Literature

The primary objective of cryptography is to convert, or encrypt, a readable message known as plaintext into an unreadable form, ciphertext, so that adversaries cannot read the contents, but over the years the scope of cryptography has widened. Throughout history encryption has been used allow people and groups to exchange secret messages, especially in times of war. Since the early transposition and substitution ciphers, where each character in a message are rearranged and replaced by others a certain number further down the alphabet respectively, encryption has evolved to include techniques for identity authentication, integrity checks and much more. Cryptography is therefore the study of encryption and these techniques, its counterpart: the study of breaking the encryption to find the original message, is known as cryptanalysis. Eventually, for most encryption techniques a weakness is found, and subsequently exploited, so more complex techniques are conceived and with the invention of computers the complexity of the algorithms has increased greatly. However, the computer power is also available for cryptanalysis so the cycle of continuous improvement of the algorithms hasn't stopped.

2.1 Asymmetric & Symmetric Cryptography

In cryptography there are two main concepts that the algorithms are based on: symmetric and asymmetric keys. In asymmetric key cryptography a unique key is used to encrypt data and different, but a related key, is used to decrypt it. The relationship between the two keys is often defined by maths problem that is very difficult to solve which is the basis of the encryption. On the other hand, symmetric key cryptography uses the same key for both encryption and decryption, hence symmetric, with the security usually provided by a combination of simple logic operations.

2.1.1 Asymmetric Key

Asymmetric key cryptography can be referred to as public key due to the fact that one of the related keys can be publicly available without compromising the security

of the encrypted data. This is because the keys are usually generated based on mathematical problems that have no solution or the solution is impossible for a computer to solve efficiently, such that solving it takes longer than an exhaustive key search[insert reference]. There are many problems that fit this criteria but the most popular in use today are the integer factorization and elliptic curve problems used by the RSA[insert reference] and the family of Elliptic-curve cryptography (ECC) techniques[insert reference] respectively. Public key cryptography can be used in two different modes as if data is encrypted with the intended recipients public key only they can decrypt it with their private key, thus encryption. However, if a private key is used for encryption then using the public key to decrypt it ensures the senders identity, authentication[insert reference].

Due to the fact that these algorithms are hard to solve they require complex hardware or software to implement which is undesirable for this project. Also, with the exception of ECC the key sizes needed for the security can be very large so with the limited IO pins available on FPGAs they could prove difficult to program.

2.1.2 Symmetric Key

Similar to the symmetric/private comparison symmetric key cryptography is also known as private key as in order to keep the encrypted data secure the key used must be kept secret. There are two main types of private key algorithms that operate on the plaintext differently: block ciphers which uses a fixed number of bits, block; or stream ciphers which encrypts data bit by bit.

Modern block ciphers are based on Claude Shannons work on product ciphers in [insert reference], in which he suggested that iterating a cipher for multiple rounds, with subkeys, improves the security. Hence, the cipher to be iterated didn't need to be complex operations and simple logic operations such as XOR, substitution or permutation of the plaintext could be used[insert reference]. The base cipher that is iterated is known as the round function and it takes as an input a block of plaintext and a subkey, which is generated from the main key by a separate key expansion function, and outputs a block of ciphertext. The round functions are mostly designed using either a Feistel network (F network) or a Substitution Permutation network (SP network).

The Fesitel network was named after physicist Horst Feistel who was a integral part of the team at IBM that developed the early block cipher Lucifer, which of course used a Feistel network[insert reference]. The F network works by splitting the input plaintext into two equal words, known as the left (MSB) and right (LSB) words. The round function is then applied to the right word before the result is XORed with the left word and then the words are swapped over and iterated as in equation [insert equation], with the ciphertext being equal to (R_{n+1}, L_{n+1}) where n is the number of rounds iterated. The advantage of using a F network is that decryption is just applying the same algorithm but in reverse as in [insert equation], with the ciphertext (R_{n+1}, L_{n+1}) and the plaintext (L_0, R_0) returned.

$$L_{i+1} = R_i$$

$$R_{i+1} = L_i \oplus F(R_i, K_i)$$

$$R_i = R_{i+1}$$

$$L_i = R_{i+1} \oplus F(L_{i+1}, K_i)$$

The other structure used in block ciphers is Substitution Permutation network and unlike F networks an SP network operates on the whole plaintext block using substitution, in the form of a S-box, to provide Shannon's confusion and permutation, a P-box, for diffusion[insert reference]. Individually, these operations aren't particularly strong as a S-box and a P-box can be thought of as simple substitution and transposition ciphers respectively. Although if combined in a SP network over multiple rounds the security can be very strong due to Shannon's confusion and diffusion properties being satisfied. The S-boxes usually take in a certain number of bits and outputs the same number of bits but of a different value. P-boxes are then used to spread the bits around such that the output of the S-boxes are used by as many S-boxes in the next round. After the S-boxes and P-boxes and before the next round occurs the block is XORed with the round key so the round equation is [insert equation]. Decryption is achieved using inverted S-boxes and P-boxes and the round keys in reverse order which means that different hardware or operations are needed.

$$B_{i+1} = F(B_i) \oplus K_i$$

$$B_i = F'(B_{i+1}) \oplus K_i$$

Stream ciphers

2.2 Conventional Algorithms

There are many block ciphers that are considered very secure and therefore popular, they include: DES, Blowfish, Twofish and AES[insert reference].

2.2.1 Standardization

DES, Data Encryption Standard, is one the earliest block ciphers used in the computer age and it was developed by IBM in the 1970s based on their earlier cipher Lucifer[insert reference]. As with Lucifer it was designed around a Feistel network but the round function used also has a SP network structure to it[insert reference], but the S-boxes aren't a one-to-one function but rather output 4 bits from a 6 bit input. DES operates on a block of 64 bits for 16 rounds using a key length of 64 bits but it has an effective key length of 56 bits as 8 bits were used for parity[insert reference]. It has the name Data Encryption Standard as it was accepted as the standard encryption algorithm by the US National Bureau of Standards (NBS), now the National Institute of Standards and Technology (NIST), in 1977 after it was altered by the National Security Agency (NSA), which caused some controversy[insert reference].

DES

2.2.2 Other Algorithms

2.3 Lightweight Algorithms

2.3.1 PRESENT

2.3.2 PRINCE

2.3.3 SIMON & SPECK Algorithms

Chapter 3

Progress

3.1 Hosted C

3.2 System Verilog

Chapter 4

Future Plan

Bibliography

Appendix

DRAFT