Proposal for Analysis of Security In a Modern Processor

Justin Cox and Tyler Travis
Department of Electrical and Computer Engineering
Utah State University
Logan, Utah 84322
email: justin.n.cox@gmail.com, tyler.travis@aggiemail.usu.edu

***Abstract*— Hardware security is an ever increasing area of study since exploits have been found on computer systems. It has been found that attackers are able to look at the memory of a system to see what is being processed. This paper proposes adding an extra level of security between the memory and the pipeline. This layer of security would encrypt data headed to memory and decrypt the data headed to the pipeline making the data held in memory incomprehensible to an attacker. To make the added cryptographic system more secure, research will be done on the benefits of utilizing a Physically Unclonable Function (PUF). Further analysis will be done on how this will affect the processor's performance.**

***Index Terms*—encryption, decryption, security, pipeline, PUF.**

I. INTRODUCTION

Over the years, there have been many advancements made to computer systems and architecture. These advancements have been primarily focused on improving performance and power consumption. As a result, the security of these architectures has been neglected. Malicious attacks on computer systems are becoming more common and as a result, there is greater need for more secure systems.

The optimal solution to this problem is to completely redevelop the architectures with security as a top priority. However this option is not very practical as it would be far too expensive to replace the current processors and there is a need to support legacy machines. Therefore, we propose to create security modules that can be added to current processor architecture.

This paper shows the authors' process of developing a DES cryptography module designed to be inserted into the gem5 simulation software. The paper also shows how a PUF can be used to increase the security of the DES module. The results of the architecture's performance with the added security module will be simulated and the results will be given.

II. BACKGROUND

Computer security is becoming more important as more systems and devices are being compromised. In particular, there has been an increased number of attacks focused on stealing information stored in memory on computers and servers. This information can be encrypted when not in use, but currently there are very few solutions that allow the computer architecture to work with the encrypted data. As a result, when the architecture is running, the data stored in memory is vulnerable to attackers.

Previous research has been done on the development and implementation of secure processors [1]. Similar to the previous research done, this paper will focus on data encryption and decryption and the toll it takes on processor performance. The objective is to provide as much security as possible without reducing the performance heavily.

III. TECHNIQUE

The following subsections will describe the techniques used to supply current computer architecture with more data security. The first subsection will describe how the data will be protected by cryptography. The last section will describe how the cyptograhic modules will be improved using a PUF.

A. Encryption and Decryption of Data

The memory can be modified by an attacker and as a result the attacker is able to change the execution flow of a program. Since the data is not encrypted, the attacker is also able to steal important information that may be contained in memory. Therefore to prevent the attacker from knowing where to modify or read the data, it is important for the CPU to encrpyt data being stored into memory. Initially, we will encrypt memory that is stored in registers, cache, RAM, and virtual memory [1]. If the performance decreases significantly, encryption will be excluded from the registers and/or cache. Assuming the pipeline is trusted, data being pushed into the pipeline will be decrypted.

These encryption and decryption modules will be built and inserted into the gem5 source code.

B. Data Encryption Standard (DES)

The type of encryption used for the module will be DES. Although DES is not the current encryption standard and is not as secure as the Advanced Encryption Standard AES, DES will allow better processor performance while still making the data more secure.

A brief overview of DES will be given so that the reader has a better understanding of how encryption could take a toll on processor performance. If the reader would like an in-depth understanding of DES, it is recommended that the reader look to other sources.

The DES algorithm takes a 64-bit plain text input. It is then run through an initial permutation that outputs 56-bits when are then split into two halves. The data goes through sixteen rounds that each have a sub-key that is generated for each round based on the original 64-bit DES key. After the

sixteenth round, the output is run through a finial permutation and the algorithm outputs a 64-bit encrypted cipher text. The rounds are illustrated in Figure 1.

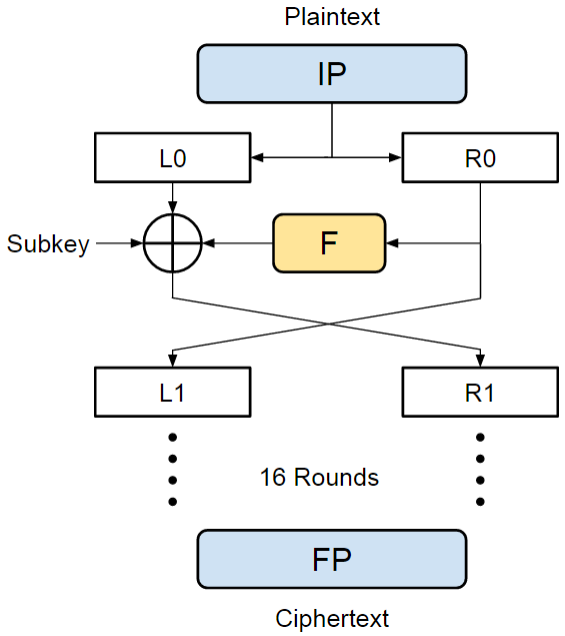

Fig. 1. An illustration of the sixteen round DES algorithm.

During each round, the left 32-bit halve is XORed with the sub key corresponding to the current round as well as the output of the F-function. The inside functionality of the F-function is illustrated in Figure 2. The output of the XOR is used as the next round's right halve and the next round's left halve is the previous round's right halve.

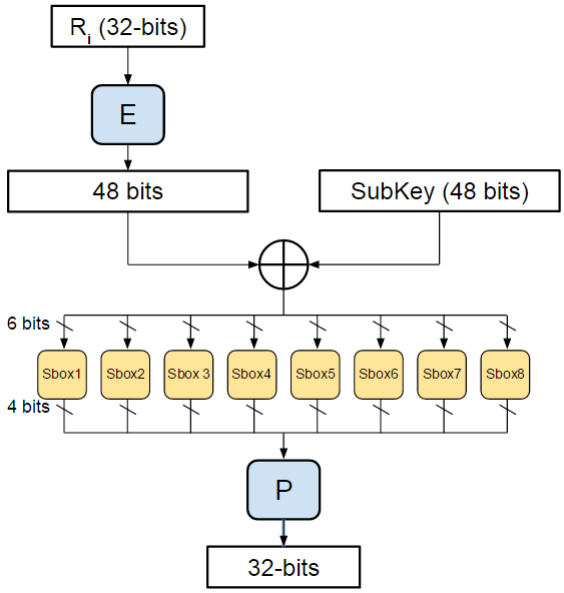

Fig. 2. An illustration of the F-function

C. Physically Unclonable Function (PUF)

A PUF is used to generate secrets extracted from a physical device due to manufacturing differences. This uniqueness can be used for more secure key generation for encryption/decryption [2]. Since the PUF output is generated by a physical device, a predetermined secret will be used to simulate the PUF output.

IV. METHODOLOGY

A. Benchmarking and Performance

The simulator that this project will use is gem5. The security modules will be added on to several different processor benchmarks found in gem5. A chosen application will be run on all benchmarks without the added security modules and again with the added security modules. The performance will be compared between the results.

V. PROGRESS MADE

Over the last few weeks, the team has been working on a DES implementation in C++. The program is able to encrypt and decrypt data in blocks of 16 bytes. The team has started looking into where to put the module inside the gem5 simulation source.

REFERENCES

[1] G. Edward Suh, C. W. O'Donnel, I. Sachdev, and S Devadas. Design and Implementation of the AEGIS Single-Chip Secure Processor Using Physical Random Functions. *Proceedings of the 32nd annual international symposium on Computer Architecture*, 2005.

[2] M. Deutschman, "Cryptographic Applications with Physically Unclonable Functions," M.S. Thesis, Inst. Mathematics, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria, 2010.