Benchmarking SHA-2 and SHA-2 Hash Functions

Roberto Martinezcardozo   
*Electrical and Computer Engineering Department*  
*California Polytechnic Pomona*  
Pomona, USA  
rcardozo@cpp.edu  
  
  
Alec Voong  
*Electrical and Computer Engineering Department*  
*California Polytechnic Pomona*   
Pomona, USA   
alvoong@cpp.edu  
  
  
Jasmine Mirbasoo  
*Electrical and Computer Engineering Department*  
*California Polytechnic Pomona*   
Pomona, USA   
jlmirbasoo@cpp.edu  
  
  
Avesta Najaf  
*Electrical and Computer Engineering Department*  
*California Polytechnic Pomona*   
Pomona, USA   
anajaf@cpp.edu

*Abstract*— This technical report presents a detailed benchmarking study of SHA-2 and SHA-3 hash functions implemented on various softcore processors. The primary objective is to assess the performance and efficiency of these cryptographic algorithms across diverse hardware configurations. Utilizing OpenSSL for benchmarking, the study includes processors such as Intel i7/i9 series, ARM Cortex-A72, and Raspberry Pi 4B. The report provides an in-depth analysis of the results, comparing metrics such as the number of blocks processed and throughput in 1000 bytes per second. Conclusions from this study contribute to a understanding of the interplay between hardware characteristics and the computational efficiency of SHA-2 and SHA-3 hash functions.

Keywords—performance, hash functions, benchmark

# Introduction

Understanding system performance and the efficiency of cryptographic algorithms is crucial for ensuring cybersecurity. This report provides a detailed examination of SHA2 and SHA3 hash functions when applied to softcore processors. The objective of this research is to assess the performance of these hash functions on a variety of softcore processors commonly found in everyday devices.

To conduct this benchmarking study, we've utilized a diverse set of hardware configurations. These configurations range from high-powered desktop CPUs to the more energy-efficient Raspberry Pi setups, allowing us to explore how different types of processors handle cryptographic tasks. Here's a list of the hardware configurations used:

Roberto Martinezcardozo’s Hardware

Desktop 1

CPU: Intel i7-9700 @ 3.0 GHz

RAM: 32 GB DDR4 2400 MHz

Raspberry Pi 4B

CPU: ARM Cortex-A72 @ 1.5 GHz

RAM: 4 GB LPDDR4 948 MHz

Laptop  
CPU: Intel i7-11800H @ 2.30 GHz

RAM: 16 GB DDR4 3200 MHz

Alec Voong’s Hardware

Desktop 2

CPU: Intel i5-1035G7 @ 1.20 GHz

RAM: 8 GB 3733 MHz

Avesta Najaf’s Hardware

Desktop 3

CPU: Intel i9-12900K @ 3.20 GHz

RAM: 32 GB DDR5 5200MHz

Jasmine Mirbasoo’s Hardware

Desktop 4

CPU: Intel i7-1165G7 @ 2.80GHz

RAM: 16 GB

The report will begin with a brief overview of the background of SHA, explaining the basics of shared hash algorithms and their evolution. Following this, we'll dive into the operational mechanics of SHA-2 and SHA-3 hash functions to set the stage for our experimentation. The experimentation and results section outlines the methods we used, employing the OpenSSL library for benchmarking, and presenting detailed analyses of SHA2-256/512 and SHA3-256/512 benchmarks.

Moving forward, we'll dissect the results obtained from our experiments, focusing on block processing and throughput rates across the various processors. By making detailed comparisons, we aim to uncover how different hardware specifications influence the efficiency of cryptographic algorithms.

This report provides a comprehensive guide to understanding the outcomes of our benchmarking study. It not only facilitates a comparison of softcore processors but also offers valuable insights into the broader world of cryptographic algorithm optimization. As technology continues to advance, the findings of this research contribute to the ongoing conversation about the relationship between hardware design and the performance of cryptographic algorithms.

# Background on SHA

## What is SHA?

SHA stands for secure hash algorithm, and it was first developed by the National Institute of Standards and Technology. SHA is a type of hashing algorithm used in computer security to perform tasks such as data verification creating digital certificates, and act as a password storage. In the SHA family, there are many types of versions and each one differs depending on the block size of the algorithm. Some examples of commonly used SHA include SHA-1, SHA-2, SHA-256, etc. When identifying the complexity and type of SHA, the higher the block size that follows SHA will produce a more secure output. As technology is still evolving, SHA-2 has overtaken SHA-1 as the industry standard.

## How does SHA work?

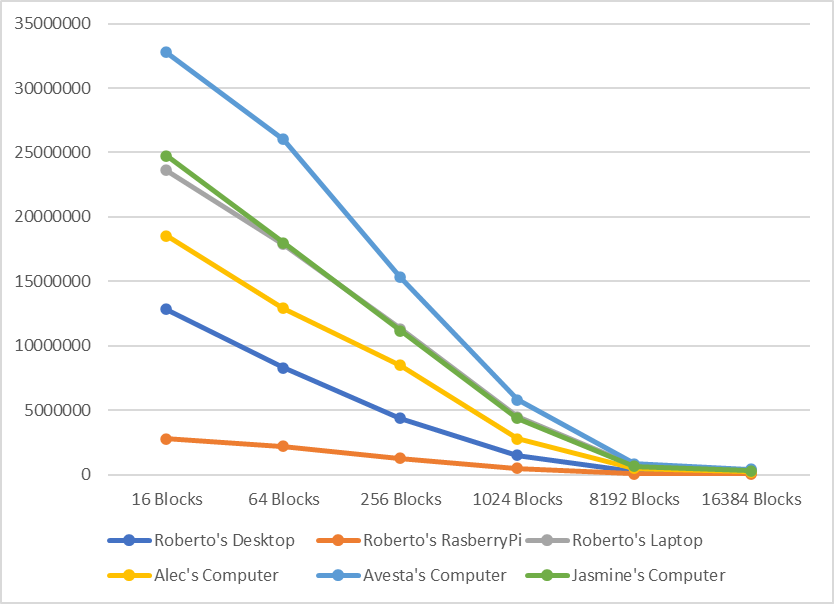
The SHA algorithm works by taking in a message and encrypting it with the algorithm to create a hash value of a certain size. The first step in the process of SHA is padding the message to make the length become a multiple of the block size. When padding occurs, a ‘1’ but is added at the end of the message followed by ‘0’s until the length is a multiple of the block size. Next, the message with padding is broken up into blocks of the desired block size. There is another vector that is initialized, and it contains buffers. In the following step, the hash value is processed by combining the message blocks and the initializing vector. The process of combining involves the use of logical and arithmetic operations. After each of the rounds during the combination process, the value is stored in the initialization vector. When producing an output using a type of SHA, the result is non-reversable, and this is important in maintaining the security of the information.

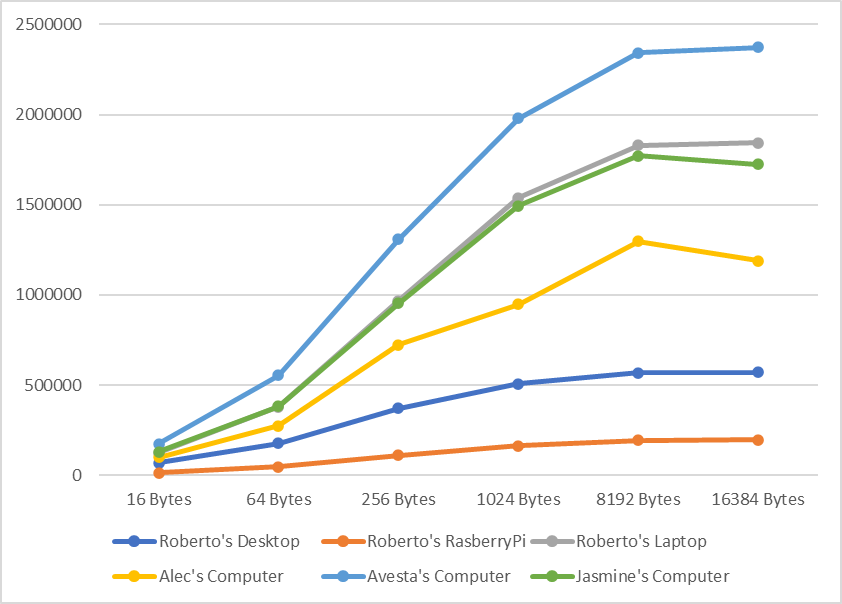
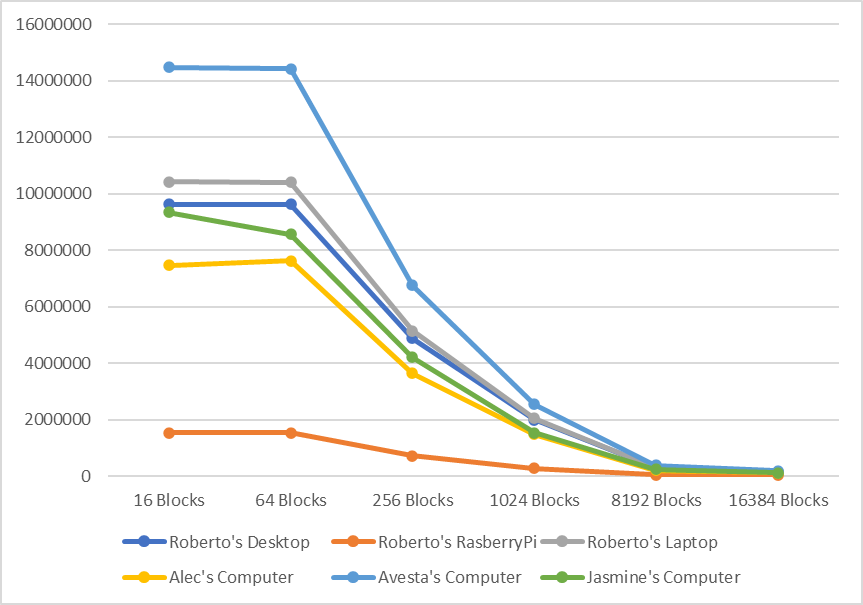
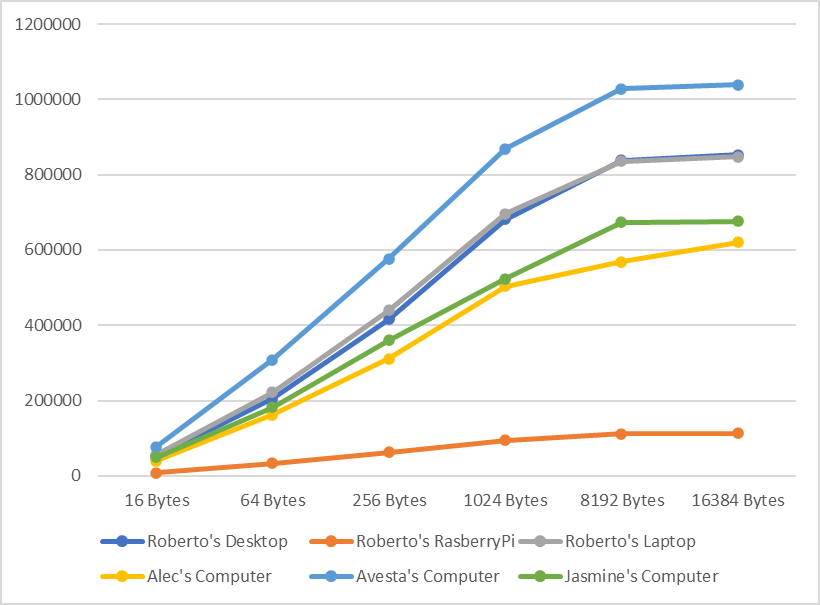
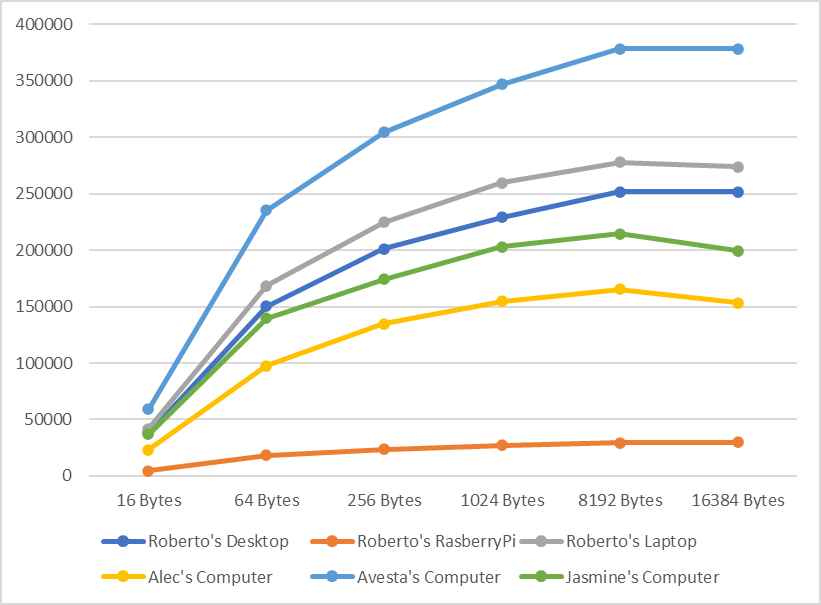
# experimentation and Results

For our benchmarking we used a library called OpenSSL. This is an open-source software library for implementing Secure Sockets Layer (SSL) and Transport Layer Security (TLS) Protocols. It also includes a general-purpose cryptography library which we will be using to benchmark all our different processors.

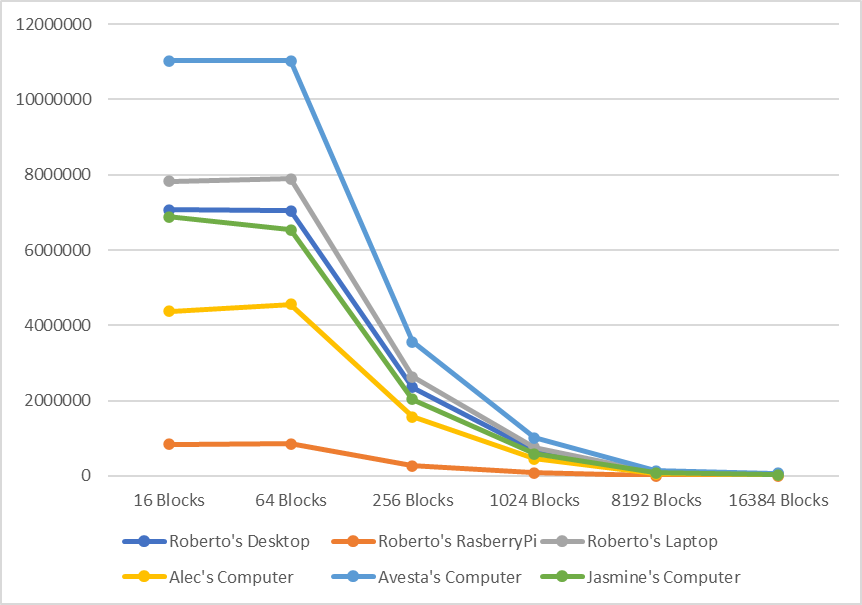
Using the OpenSSL library, we benchmarked our processors SHA2 and SHA3 hash functions using a native benchmarking tool. We ran two different benchmarks for both hash functions, the first was a 256-bit hash output and the second was a 512-bit hash output. Each benchmark gives 2 different outputs, the first is the number of blocks in a given time which in this case is three seconds. The second is the throughput in 1000 bytes per second.

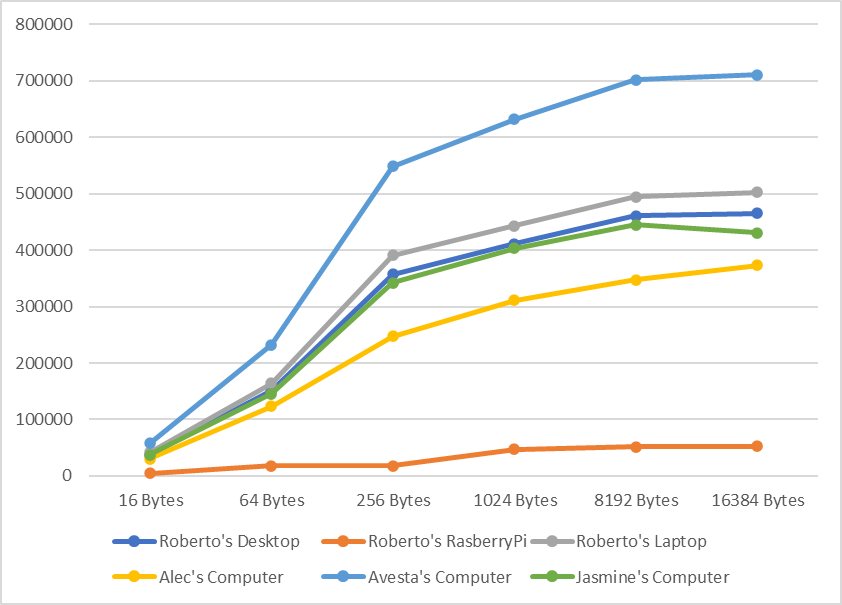
## SHA2-256 & SHA2-512 Benchmark

For our benchmark a time of three seconds was used across all the runs. The native benchmark runs the specified hash function repeatedly for a set number of blocks and bytes. It then reports that results which we analyzed and organized into different graphs showing our different processors and how they compare to each other in terms of performance metrics.  
  
Fig. 1. SHA2-256 Number of Blocks in a Given Time Graph

  
  
Fig. 2. SHA2-256 Throughput in 1000 Bytes Per Second Graph  
  
  
Fig. 3. SHA2-512 Number of Blocks in a Given Time Graph  
  
  
  
  
Fig. 4. SHA2-512 Throughput in 1000 Bytes Per Second Graph

## SHA3-256 & SHA3-512 Benchmark

We used the same method of benchmark for the SHA3 family of algorithms that we used on the SHA2 family.

Fig. 5. SHA3-256 Number of Blocks in a Given Time Graph  
  
  
Fig. 6. SHA3-256 Throughput in 1000 Bytes Per Second Graph

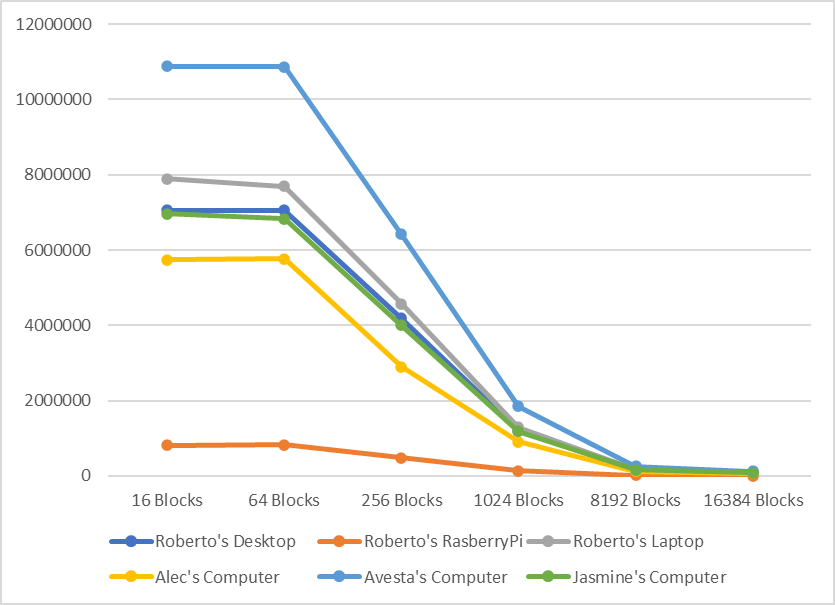
  
Fig. 7. SHA3-512 Number of Blocks in a Given Time Graph

Fig. 8. SHA3-512 Throughput in 1000 Bytes Per Second Graph

# Analysis of results

Before diving into the analysis, we will be giving a brief overview of some of the words we will be using. When referring to blocks this is the size of the data chunks or ‘blocks’ that are being processed by the hash function. Blocks tell us how many operations can be performed on the data within the given time which in this case would be three seconds. When referring to ‘bytes’ this is the throughput of the hash function in terms of data processed in the rate of time given. This gives insight into the processors overall data processing speed of the hash function. The benchmark used 6 different block sizes to give us better insight on performance for the processor.

## SHA2-256 and SHA2-512 comparison

When looking at both graphs we can infer a few things from the results. The highest and lowest performing processors are consistent throughout all the graphs. The highest performing processor is the Intel i9-12900K processor that was accompanied by two 16 gigabyte DDR5 sticks of RAM. It outclasses all of the rest of processors in clock speed, L2 Cache size and accompanying RAM. This was an expected result as was the lowest performing processor. This was the ARM Cortex-A72 on the raspberry pi 4 model B. This is due to its very low clock speed as well as its RAM being considerably slower than the rest, having a clock speed of 948 MHz.

Next, we will be looking at the metrics for Roberto’s Desktop and Roberto’s Laptop. When looking at the graph in Figure 1 and Figure 3 the margin between the two is different, however, in Figure 1 the margin is a lot bigger than the margin in Figure 3. The large margin in Figure 1 can be explained by the laptop’s higher clock speed. Figure 1 is the SHA2-256 algorithm which operates on 32-bit words. This means that the processor will be accessing data more frequently, because of this a processor with a faster clock speed will be able to access the memory faster and be able to process more hashes. This is what makes the margin large, now the reason why the margin is smaller in Figure 3 is because of the difference between SHA2-256/512. SHA2-512 works using 64-bit words instead of 32, this causes the processors to access the memory less frequently than it did before and because it is accessing memory less the advantage of having higher clock speeds is lessened. This is what causes the margin between the desktop and the laptop to shrink.

Next, we will be looking at the through put of the devices in Figure 2 and Figure 4. Similarly, to the other figures we saw the highest and lowest devices were the intel i9 processor and the raspberry pi respectively. We see the same margin between Roberto’s Laptop and Desktop that we saw in Figures 1 and 3. This is still for the same reason.

## SHA3-256 and SHA3-512 comparison

Now we will be moving onto the SHA3 family, this algorithm is often slower in software due to its more complex operations. This explains why fewer total blocks are being processed in the same 3 second benchmark than its counterpart in the SHA2 family.

The next thing I will be talking about are the initial two block sizes and why the graph in Figure 1 looks different than the ones in Figures 3, 5, and 6. SHA2-256 is built to process using 32-bit words, this is why when moving from 16 blocks to 64 there is a drop on blocks processed in 3 seconds. The rest of the figures are algorithms that are built to process using 64-bit words, this causes there to be a considerable amount more of blocks to be processed in 3 seconds and explains why the graphs in Figure 1 and 3 to be so different. This difference is not seen in the SHA3 family because both are built to process 64-bit words, this explains why there is a significant similarity between the two graphs.

The next thing we will be analyzing is the throughput per second. We noticed that the rate of the throughput per second was close to half between SHA3-256/512, this can be explained by the output of the hash function. The 256 outputs a 256-bit hash, while the 512 outputs a 512-bit hash. This is why the throughput rate drops by close to half between the 2 figures.

# Conclusion

In summary, we benchmarked of cryptographic algorithms SHA-2 256/512 and SHA-3 256/515 hash functions when applied to softcore processors. SHA is a type of hashing algorithm used in computer security to perform tasks such as data verification creating digital certificates, and act as a password storage. We used OpenSSL library for benchmarking processors' performance in implementing SHA2 and SHA3 hash functions. Two benchmarks were conducted for each hash function, measuring the number of blocks processed in three seconds and the throughput in 1000 bytes per second for both 256-bit and 512-bit hash outputs. The general-purpose cryptography library within OpenSSL was employed for these assessments. We examined this benchmarking study of five different processors. They are Raspberry Pi 4B Raspberry Pi 4B single processor board, Desktop 1 (Roberto’s Hardware), Desktop 2 (Alec’s Hardware), Desktop 3 (Avesta’s Hardware), Desktop 4 (Jasmine’s Hardware) and a Laptop (Roberto’s Hardware). For both SHA2 256/512 and SHA3 256/512, the analysis of the provided graphs reveals consistent patterns in processor performance. The Intel i9-12900K processor (Avesta’s Hardware), coupled with high-speed DDR5 RAM, consistently emerges as the top-performing processor, attributed to its superior clock speed, L2 Cache size, and RAM specifications. Conversely, the ARM Cortex-A72 on the Raspberry Pi 4 Model B (Roberto’s Hardware) consistently performs at the lowest level due to its low clock speed and slower RAM.