Evaluating Cryptographic Performance of Raspberry Pi Clusters

Daniel Hawthorne
Department of Electrical Engineering & Computer Science
United States Military Academy
West Point, NY
Daniel.Hawthorne@westpoint.edu

Michael Kapralos
Department of Electrical Engineering & Computer Science
United States Military Academy
West Point, NY
Michael.Kapralos@westpoint.edu

Raymond W. Blaine
Department of Electrical Engineering & Computer Science
United States Military Academy
West Point, NY
Raymond.Blaine@westpoint.edu

Suzanne J. Matthews
Department of Electrical Engineering & Computer Science
United States Military Academy
West Point, NY
Suzanne.Matthews@westpoint.edu

***Abstract*—ARM-based single board computers (SBCs) such as the Raspberry Pi capture the imaginations of hobbyists and scientists due to their low cost and versatility. With the deluge of data produced in edge environments, SBCs and SBC clusters have emerged as low-cost platform for data collection and analysis. Simultaneously, security is a growing concern as new regulations require secure communication for data collected from the edge. In this paper, we compare the performance of a Raspberry Pi cluster to a power-efficient next unit of computing (NUC) and a mid-range desktop (MRD) on three leading cryptographic algorithms (AES, Twofish, and Serpent) and assess the general-purpose performance of the three systems using the HPL benchmark. Our results suggest that hardware-level instruction sets for all three cryptographic algorithms should be implemented on single board computers to aid with secure data transfer on the edge.**

***Index Terms*—Raspberry Pi, CryptSetup, edge computing**

I. Introduction

Single board computers (SBCs) enjoy widespread popularity due to their low cost and applicability to a wide range of applications. While early models of SBCs contained "weak" ARM processors, newer versions boast more powerful ARM processors that have similar computational performance to older Intel Pentium processors [1]. The energy efficiency and inexpensiveness of SBCs and SBC clusters make them an attractive option for data analysis and collection in edge environments [2]–[7]. While edge computing can be built with a wide variety of systems, common solutions for smaller laboratories include workstations for their raw processing power [8], [9], Next Unit Computers (NUCs) for their efficiency [10]–[12], or clusters of single board computers (SBCs) [10], [13]–[15]. SBC clusters are especially attractive due to their extensibility, allowing researchers to build and modify a custom cluster that best fits their needs.

As single board computers become increasingly prevalent in edge environments, security becomes a key concern. New regulations [16]–[18] require secure end-to-end communication and security for data-at-rest for devices operating at the edge. Scientists using commodity systems desire fast encrypt/decrypt operations to enable secure transfer of data to other systems in their workflows. As the popularity of SBCs continues to increase for edge analysis, it is critical to evaluate the cryptographic performance of SBCs and SBCs clusters, especially as they compare to other alternatives that scientists may choose for localized data analysis.

In this paper, we evaluate the cryptographic performance of three popular symmetric-key block-text ciphers (AES, Twofish and Serpent) on a Raspberry Pi cluster, a power-efficient next-unit of computing (NUC) and a mid-range desktop (MRD) system. AES was chosen over Twofish and Serpent in the AES selection competition in 2001 primarily due to its performance on Intel machines; one of our research questions is if AES outperforms Twofish and Serpent on ARM-based SBCs. We focus specifically on the Raspberry Pi 3B+ SBC due to its extreme popularity, low cost, and wide community support. We also compare the performance of the three systems using the High Performance LINPACK (HPL) [19] benchmark, and measure power consumption. Lastly, we provide instructions to enable researchers to reproduce our cluster and results [20].

Our results show that Twofish yields the best encryption throughput and similar decryption throughput to AES on the Raspberry Pi clusters at higher key sizes. Our results also demonstrate that a 6-node Raspberry Pi cluster (24 cores, $320.00, 37.6 watts) outperforms the NUC and MRD for certain encryption tasks while achieving 15.72 GFLOPS on the HPL benchmark. While the NUC and MRD systems achieve higher encrypt/decrypt throughput than the Pi cluster on AES, we note that the Raspberry Pi does not yet implement AES instructions in hardware. Our results support the notion that hardware implementations of common cryptographic ciphers should be implemented on ARM processors used for SBCs, and the viability of algorithms like Twofish for software-level encryption/decryption operations on SBC clusters should be explored more deeply.

II. Background & Related Work

This work focuses specifically on the Raspberry Pi 3B+ single board computer due to its popularity, wide community support, and inexpensiveness ($35.00 for the board, $\approx$ $50.00 total for all needed components). We note however, that our results are applicable to any ARM-based single board computer (SBC) architecture. The Raspberry Pi was also chosen for the availability of the Ubuntu Core distributions and its existing support of the cryptographic benchmark suite used in this study (see Section III-C for details). We investigated other SBCs for the same performance metrics and found them similarly lacking in AES NI hardware instructions (see Section V for details).

The historically low price-point of the Raspberry Pi is due to the relative "weakness" of the on-board System-on-a-Chip (SoC); the original Raspberry Pi released in 2012 featured an ARM 700 MHz single-core processor and 512 MB of RAM. Researchers consequently began to assemble Raspberry Pi Beowulf *clusters* in the hopes that many Pis in aggregate would yield non-trivial computational performance. Iridis-Pi [21] is the most noteworthy early example, and consisted of 64 Raspberry Pi Model 1 nodes. Other notable early clusters include the Glasgow Cloud [22], the Bolzano Raspberry Pi Cluster [23] and the BSU Pi Cluster [24].

While early Pi clusters performed modestly well on computational tasks, the motivations for constructing early clusters were largely educational. However, the demand in the mobile computing community for power-efficient ARM SoCs led to the development of more sophisticated chipsets, enabling the release of iteratively more powerful Raspberry Pi SBCs while maintaining the the $35.00 price-point. The Raspberry Pi 3B+ is the focus of this study, and features 1 GB of RAM and a 1.4 GHz quad-core ARM Cortex A53 processor.

Many researchers [6], [21], [25] choose to employ the High Performance LINPACK (HPL) [19] benchmark, a standard employed by the HPC community to estimate the real-world compute performance of supercomputers. For example, the 16-node Raspberry Pi 3B+ cluster described in [6] achieved a peak performance of $\approx$ 40 GFLOPS. The researchers also note that the LAN7515 chip used in the Raspberry Pi 3B+ severely limits the Ethernet (measured at 328 Mbps [6]), and therefore the overall performance of the system.

A key novelty of our work is evaluating the *cryptographic performance* of the Raspberry Pi and Raspberry Pi clusters. Cryptographic performance generally includes performance of cryptographic algorithms and primitives that may include encryption, decryption, and hashing. We focus specifically on the encryption and decryption throughput of some of the world's leading symmetric key algorithms: Rijndael (now AES), Serpent, and Twofish. Rijndael was selected as the AES standard to replace DES in 2001 for its strong performance on a variety of platforms and its ease of implementation in hardware [26]. The other two algorithms, Serpent and Twofish were finalist candidates for AES. All three algorithms are commonly included in Linux distributions and are still widely implemented in many tools and included in many operating systems. We note that when AES was selected, x86 was the primary architecture being evaluated, though performance on RISC-based and embedded systems also played a factor. Thus, one of our research goals is to determine if AES outperforms Twofish and Serpent on ARM-based single board computers. CryptSetup, a standard Linux package, provided an ideal benchmark platform for cryptographic performance. This package allows users to encrypt their storage, but the benchmark component runs in memory to allow users to determine the encrypt and decrypt potential of their CPU. Relating general purpose performance to cryptographic performance assists with assessing when a given amount of processing power should be considered a threat [27]. Our paper extends this work by using similar instrumentation on a Raspberry Pi cluster to make general statements about cryptographic performance of SBCs. Lastly, while the more powerful Raspberry Pi 4 was recently released, it was not originally used to due to its incompatibility with commercial clustering solutions and unavailability of the requisite encryption modules. Recently, Ubuntu Core was released for the Raspberry Pi 4. A brief discussion is included in Section V.

Existing evaluation of SBCs and SBC clusters for cryptographic applications is relatively scant. A performance study [28] on password cracking using John the Ripper showed that Parallella and Raspberry Pi clusters outperformed a high-end laptop when running a dictionary-style attack against "weak" passwords encrypted with bcrypt. A separate study [29] showed that the Raspberry Pi 3 outperforms the Intel Atom processor (Intel's embedded offering) on brute-forcing SHA-1 hashed passwords; we note that SHA-1 has recently been shown to be unsafe [30]. A separate study compares the performance of AES, Twofish and Serpent on various Android OS ARM mobile platforms in the context of a chat program [31]. Other papers [32]–[35] also compared the performance of cryptographic algorithms on various ARM processors. To the best of our knowledge, our study is the first to compare the cryptographic performance of ARM-based SBC clusters to desktop-grade Intel and AMD systems.

Today, hardware instructions for AES key expansion, encrypt and decrypt exist for x86_64 platforms. These new AES instructions (AES NI) provide a five-fold improvement of the encrypt and decrypt performance of the AES algorithm [36]. We note that while AES hardware acceleration is an option for several classes of ARM processors, it is not enabled by any known operating system available for SBCs. Therefore scientists leveraging SBCs cannot leverage hardware acceleration for their encrypt and decrypt tasks. A final goal was to study how the lack of this feature affects throughput.

III. Methods

Table I outlines the three architectures under study. These include an Intel-based highly power-efficient next unit of computing (NUC) and a mid-range desktop (MRD) that contains a workstation-grade AMD processor. The Intel and AMD systems were chosen to mimic the types of systems

TABLE I: Overview of Architectures under Study

Description	CPU	Memory	AES NI?	No. Cores	Est. Cost
Power-Efficient NUC	Intel Core i5-8500T	8 GB DDR4	TRUE	6	$700.00
Mid-Range Desktop	AMD Ryzen 7 2700	16 GB DDR4	TRUE	8 (16 HT)	$912.00
Raspberry Pi 3 B+	Broadcom BCM2837	1 GB DDR4	FALSE	4	$50.00
Raspberry Pi 3 B+ Cluster	Broadcom BCM2837	20 GB DDR4	FALSE	80	$1,800.00

that smaller research labs may have access to. A benefit of an SBC cluster to a commodity system is its *extensibility*; researchers can add nodes or shrink their cluster to match their computational needs or energy requirements. In several of our experiments, we compare the performance of the Intel and AMD systems to various subsets of our cluster. We also use a single Raspberry Pi as a base-line in some experiments to make additional generalizations about theoretical performance.

A. Cluster Setup

The Raspberry Pi cluster used in this work (Figure 1) is built using a BitScope Blade Rack 20 [37] and 20 Raspberry Pi Model B+ boards. While custom rack constructions [6] are more economical, the BitScope Blade is an elegant commercial solution that was chosen to aid others in reproducing our setup and prepare the cluster for long-term laboratory use. The BitScope rack simplifies cable management and provides voltage protections for the individual Pi units. The rack itself is a 19-inch 5RU platform composed of 10 Duo Pi boards. Each Duo Pi board powers two Raspberry Pis while providing board-level protection via a built-in 5 Volt switch-mode power supply and 3 Amp current capability. The 10 Duo boards are powered from a single input that can accept a voltage range of 9 to 48 Volts DC.

Each Raspberry Pi runs the Ubuntu Core 18.04-3 Server ARM64 operating system. Ubuntu is not the default Raspberry Pi operating system; however, it was chosen to simplify the inclusion of the cryptographic benchmarks. Raspbian, the more common Raspberry Pi operating system, does not include the kernel modules for Serpent or Twofish, which are required for CryptSetup to execute those benchmarks. We also note that none of the Raspberry Pi operating systems provide hardware-level support for AES, Twofish or Serpent.

The cluster is networked via on-board Ethernet and a rack-mounted Netgear 24 port ProSAFE Gigabit switch. A separate Raspberry Pi acts as a dynamic host configuration protocol (DHCP) server, domain name server (DNS), and a gateway providing network address translation (NAT) services.

A Corsair 750 watt ATX power supply provides centralized power to the cluster. We note that for smaller numbers of Raspberry Pis, utilizing individual power supplies is far more energy-efficient. However, the ATX power supply is more efficient at higher numbers of nodes and is capable of supporting over 80 Raspberry Pis. It also provides granular DC power utilization, which will be used for future projects involving the cluster. We also note that the Corsair power supply at $160.00 costs the same as 20 individual Raspberry Pi power supplies at $8 each.

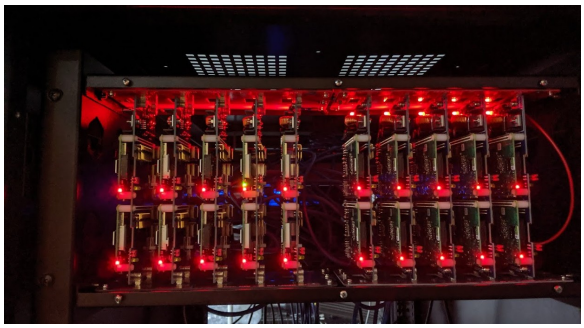


Fig. 1: Raspberry Pi 3B+ Cluster

The BitScope Blade is expensive ($695.00); replicating our exact configuration costs approximately $1800.00. Smaller configurations of the BitScope Blade cluster can be built at roughly $80.00 per Pi. It is also possible to build a less elegant version of the full cluster by using individual power supplies and an unmanaged network switch. With each Pi unit estimated at $50.00 (Pi, power supply, and 32 GB SD card) and a 24-port switch estimated at $100.00, a low-cost 20-node cluster can be built for as little as $1100.00. However, the voltage protections provided by BitScope Blade's built-in regulators increase the resiliency of the system to power failures.

B. HPL Setup

The portable implementation of High-Performance LINPACK (HPL) version 2.2 [19] was used to benchmark general compute performance. While the Intel LINPACK benchmark suite is available for Intel systems, we use HPL on all three systems to ensure consistency. Both the NUC and MRD employ MPI over Chameleon (MPICH) version 3.3 and the Basic Linear Algebra Subprograms (BLAS) library version 3.8. Based on the recommendations of prior work [6], the Raspberry Pi cluster utilizes version 3.10.3 of Automatically Tuned Linear Algebra Software (ATLAS) [38].

The HPL parameters of most interest are the size of the grid (P and Q), the block size (NB), and the problem size (N). P and Q were chosen to represent the total number of cores, and in accordance with the literature [6], [19] to be almost square with Q and slightly larger than P respectively and 80% of total available memory. HPL was tuned on the Raspberry Pi cluster parameters $P = 10$ and $Q = 8$ with an effort to match the parameters and performance of [6].

Using the above configuration, a single Pi 3B+ is able to produce approximately 3.4 GFLOPS of performance. While prior work [6] achieved approximately 4.4 GFLOPS on a single Raspberry Pi 3B+, we believe OS choice is a major

driver in this difference; prior work used Raspbian Stretch Lite [6], which was not an option due to Raspbian's lack of support for the required cryptography kernel modules.

C. Cryptography Setup

The cryptography benchmarks in this work utilize CryptSetup, a common Linux tool for encrypting storage. CryptSetup has an in-memory benchmark feature that we use to measure cryptographic performance. Cryptographic performance is measured in throughput, or Mebibytes per second (MiB/s). The CryptSetup utility relies on cryptography kernel modules for encryption and decryption. Since this study compares AES, Twofish, and Serpent, we chose Ubuntu 18.04 core as the operating system for our Pi cluster, as it included all three of those modules. As previously mentioned, Raspbian only includes the module for AES. The NUC and MRD use a similarly minimized Arch Linux OS for benchmarking based on prior work [27], [39]. Details of our experimental setup are available in GitHub [20].

The command `cryptsetup benchmark` benchmarks the encrypt and decrypt for all available algorithms. It gives results for 128-bit and 256-bit key sizes; we present results for both lengths for this paper. If present, CryptSetup utilizes hardware cryptographic instructions, such as the AES NI present in the MRD and NUC. We note that hardware cryptographic instructions are not available on the Raspberry Pi, or any SBC considered for this study (see Section V).

The process for benchmarking the NUC and MRD is straight forward, but the cluster required starting the benchmarks simultaneously and collecting the results. We use the DMUX utility [40] to distribute the benchmark command using SSH keys and aggregate the results. This test emulates each device individually encrypting and/or decrypting data as part of a cluster where the workload is compartmentalized or distributed to each node and requires atomic security. This data parallel approach to analysis is quite common to local data summarization applications; to encrypt the transmission of scientific data, it is therefore advantageous for each individual node to encrypt separately and in parallel. Our experimental design simulates this application.

CryptSetup also provides results for multiple cryptographic modes. This paper uses the results from the common cipher block chain (CBC) mode. CBC is capable of random access decryption, making decryption fully parallelizable, but encrypts sequentially. This mode and its constraints have a significant impact on the performance, favoring decryption.

The encryption, being sequential, runs serially on each device. Each node in the cluster is able to encrypt one block at a time. The NUC and MRD are also single nodes in this sense, each only able to encrypt one block at a time. For a workload where the nodes were virtualized on a single MRD or even NUC host, the aggregate encryption performance on those platforms may improve; however, this use case falls outside of our stated application.

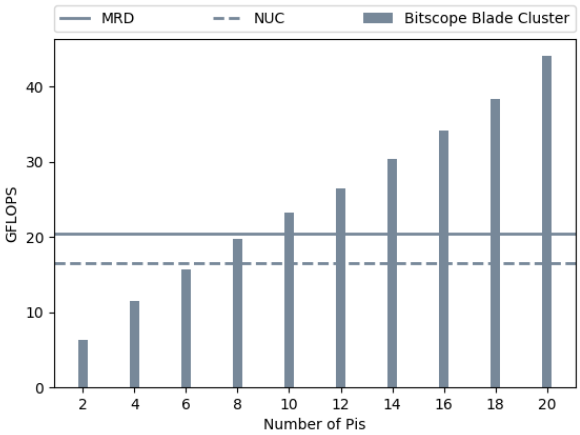


Fig. 2: HPL Results

IV. Results

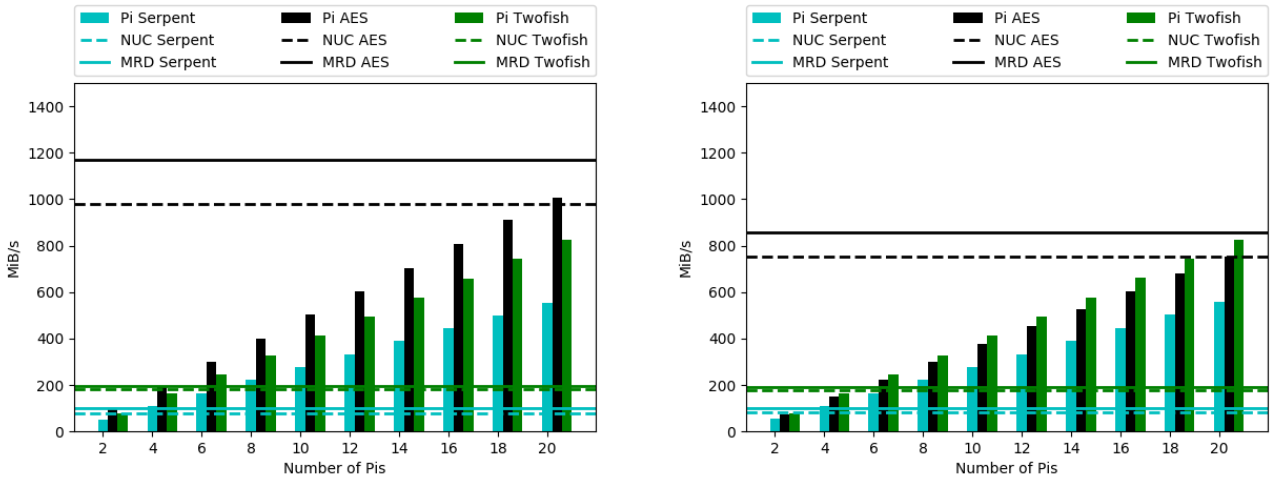
Figure 2 compares the HPL benchmark results of the MRD and NUC to different sizes of the Raspberry Pi cluster. The x-axis represents the number of Pis utilized, while the y-axis represents the GFLOPS achieved. We delineate the performance of the MRD and NUC with a solid line (20.48 GFLOPS) and a dashed line (16.51 GFLOPS), respectively.

The 20-node Raspberry Pi cluster achieves 44.1 GFLOPS on the HPL benchmark, which is consistent with prior work [6]. With a single Pi measuring at 3.4 GFLOPS, 20 Pis should ideally achieve a performance of 68 GFLOPS; however, communication overhead over the Pi's Ethernet interface serves as a bottleneck that limits theoretical scalability. Prior work [6] mentions this limitation, noting that the scalability of their 16-node Raspberry Pi 3B+ cluster leveled out around 40 GFLOPS, as the Raspberry Pi 3B+ lacks a true Gigabit connection. We note that this bottleneck does not exist on the Raspberry Pi 4.

Observe that six Raspberry Pis perform similarly to the NUC on the HPL benchmark, while eight Raspberry Pis perform similarly to the MRD on the HPL benchmark. At a cost of roughly $\$50.00 - \$80.00$ per Pi (depending on whether the BitScope is used), it may be less expensive for certain workflows to assemble a small Raspberry Pi cluster than buying a MRD or NUC. The results also suggest that 24 ARM A53 cores yield similar compute performance to 6 Intel i5 Cores, and that 32 ARM A53 cores yield similar performance to 8 Ryzen 7 cores.

A. Cryptographic Benchmarks

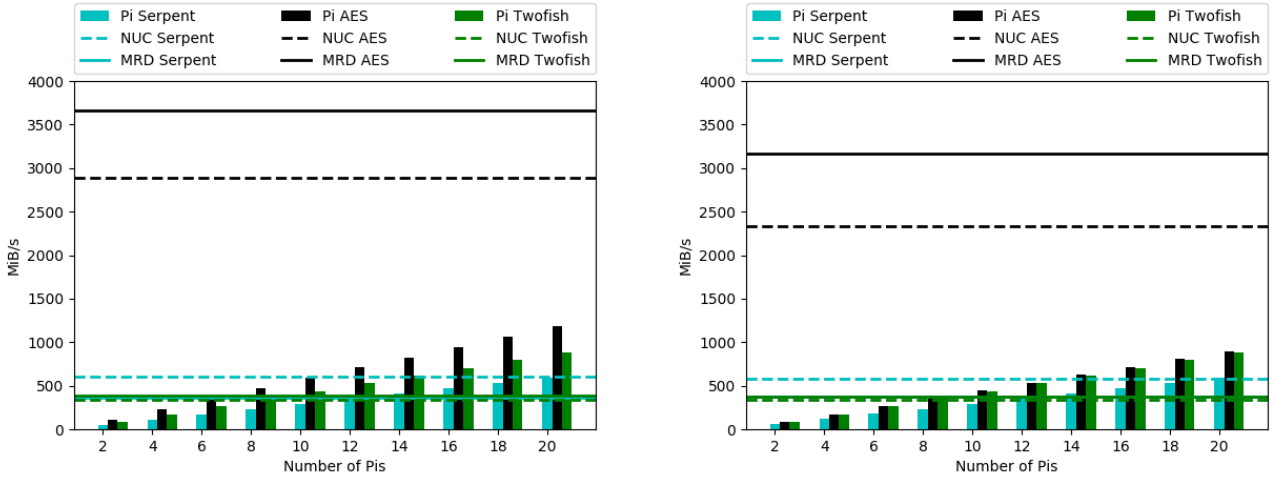
The next set of experiments compare the performance of the Pi cluster, NUC and MRD on various cryptographic benchmarks. The results for encryption and decryption are shown in Figure 3 and Figure 4 respectively. The colors (blue, black and green) represent the Serpent, AES and Twofish benchmarks.



(a) 128 Bit Encryption Results

(b) 256 Bit Encryption Results

Fig. 3: Cryptographic Benchmarks for Encrypt



(a) 128 Bit Decryption Results

(b) 256 Bit Decryption Results

Fig. 4: Cryptographic Benchmarks for Decrypt

A solid line is used to denote the performance of the MRD, while a dashed line denotes the performance of the NUC. Bars depict the the Raspberry Pi cluster's performance on different node configurations. The x-axis on the plot denotes the number of Raspberry Pi nodes, and the y-axis denotes the throughput (MiB/s). Each quantity is the average of 10 runs of a particular benchmark.

1) Encryption: Figure 3 shows the results of the benchmarks on encryption. On both 128-bit and 256-bit encryption tasks, the Raspberry Pi cluster running Serpent yields similar performance to the NUC and MRD at 4 nodes. The 4-node cluster achieves 108.9 and 109.49 MiB/s for 128 and 256-bit key size respectively, while the NUC achieves 79.2 MiB/S and 83.4 MiB/s. Lastly, the MRD achieves 100.8 and 102.5 MiB/s on 128-bit and 256-bit encryption tasks respectively.

Similarly, 6 nodes of the Raspberry Pi cluster outperforms the NUC and MRD on the Twofish benchmark. The 6-node cluster achieves 245.24 and 246.2 MiB/s for 128 and 256-bit encryption compared to 182.3 and 175.7 for the NUC. Finally, the MRD achieves 194.8 and 193.1 MiB/s for the 128-bit and 256-bit Twofish encryption tasks.

Unsurprisingly, the NUC and MRD outperform the Raspberry Pi cluster on AES encryption tasks, due to the presence of AES hardware acceleration on those platforms. Again, hardware instructions for AES are not currently available on SBCs. On 128-bit encryption tasks, the NUC and MRD achieve 977 MiB/s and 1167.2 MiB/s respectively, compared to 1007 MiB/s on the 20-node Pi cluster. However, a drawback of AES

is that it is slower on larger key sizes. On 256-bit encryption tasks, The NUC and MRD achieve 750 MiB/s and 854.7 MiB/s, while the full cluster achieves 753.6 MiB/s. Perhaps more interestingly, Twofish achieves the highest encryption throughput on all configurations of the Raspberry Pi cluster for 256-bit encryption tasks, suggesting that it may be a more preferable approach for encryption tasks on ARM-based SBCs where maximizing performance and a larger key size is a priority.

2) Decryption: Figure 4 shows the results of the cryptographic benchmarks on decryption. Throughput is higher for decryption, owing to the fully parallel nature of CBC for decryption [26]. On 128-bit decryption tasks, 8 Raspberry Pis achieve similar throughput on Twofish decryption to the NUC (350.58 vs 341.1 MiB/s), and 10 Raspberry Pis exceeds the Twofish decryption throughput of the MRD (439.68 vs 386.6 MiB/s). A similar trend is observed for the larger 256-bit key size. The 8-node Raspberry Pi cluster again achieves similar throughput to the NUC on Twofish decryption (351.52 vs 340.4 MiB/s), and the 10-node Raspberry Pi cluster outperforms the MRD (440.33 MiB/s vs 378.7 MiB/s).

Interestingly, the NUC is able to perform Serpent decryption tasks faster than either the MRD or the 20-node Raspberry Pi cluster. For 128-bit and 256-bit decryption tasks, the NUC achieves a throughput of 610.5 MiB/s and 581.1 MiB/s respectively. In contrast, the MRD achieves a throughput of 360.6 MiB/s and 374.3 MiB/s on 128-bit and 256-bit Serpent decryption tasks. Lastly, the 20-node Raspberry Pi cluster achieves a Serpent decryption throughput of 594.79 MiB/s and 596.07 MiB/s on 128-bit and 256-bit key sizes, respectively.

Due to hardware acceleration, the NUC and MRD running 128-bit AES are 2.44 times and 3.1 times faster than the Raspberry Pi cluster. For 256-bit key sizes, the NUC and MRD are respectively 2.6 and 3.53 times faster than the Pi cluster. We also note that AES and Twofish decryption throughput is very similar on the larger 256-bit key size for the Pi cluster.

Some literature [36] suggests that AES hardware instructions yields a 5-fold performance improvement on AES on Intel Architectures; we hypothesize that if the Raspberry Pi supported AES hardware instructions, smaller configurations of the cluster will outperform the NUC and MRD. For example if AES hardware instructions yielded a similar 5-fold performance improvement on the Pi, it is estimated that the cluster would outperform the NUC on 12 nodes and would be nearly identical in performance to the MRD on 14 nodes for 256 bit decryption tasks.

Our results suggest that if hardware instructions were available for Twofish and AES on SBCs like the Raspberry Pi, the cluster approach would be especially useful for decryption workloads such as high read-to-write ratio cloud storage where decryption is far more important than encryption. Fully parallel decryption modes like CBC benefit encrypted storage-at-rest that is accessed more than it is modified. Parallel decrypt modes allows personal computing devices that access remote resources to require less computational power than the servers that provide those resources. The popular hardware instructions for AES further the trend of light weight end-user devices. Decryption is also the primitive involved in brute force and many other attacks, which (while far from currently feasible) eventually needs to be considered.

TABLE II: Power Profile (W)

Description	Idle	CryptSetup	HPL
NUC	7.2	42	61.3
MRD	33.5	73.5	135
Pi 3 B+ Cluster (20)	72	160	188

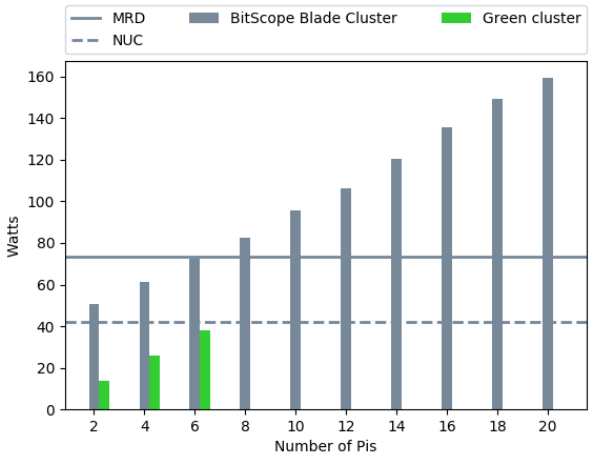


Fig. 5: Peak Power During Cryptographic Benchmarks

B. Power Benchmarks

We measure the amount of power that each system consumes during idle and peak usages for each of our benchmarks using a KillAWatt [41] power monitoring tool. Results are shown in Table II, with power consumption measured in watts.

Of the three systems, the NUC had the lowest peak power consumption during the cryptographic and HPL benchmarks. All three systems used less power during the cryptographic benchmarks compared to the HPL benchmarks; however peak CPU utilization was still measured close to 100% during these experiments.

Unlike the NUC and MRD systems, the Raspberry Pi cluster is extensible and may be constructed with as few as 2 nodes. Figure 5 compares the peak power consumption of the Raspberry Pi cluster during the cryptographic benchmarks to the NUC and MRD during the same benchmark on different node configurations. The MRD is depicted as a solid line, the NUC a dashed line, and the Raspberry Pi cluster as bars for corresponding numbers of nodes. We note that 6 Raspberry Pis have similar power consumption to the MRD system (72 and 73.5 watts respectively).

We also note that the BitScope Blade setup used for this work is extremely inefficient at low numbers of Pis. Each of the 10 Duo Pi boards in the BitScope Blade has its own voltage regulation, and while this protective feature is useful for certain applications, it introduces 10 new sources of power

TABLE III: Other SBC Performance

Description	Key Size Bits	AES (Twofish) Encrypt MiB/s	AES (Twofish) Decrypt MiB/s	Core Speed GHz
Raspberry Pi 3 B+	128 256	47.3 (38.5) 37.2 (40.5)	57.6 (43.2) 43.9 (43.3)	1.4
Raspberry Pi 4 B	128 256	23.8 (54.9) 17.4 (56.9)	78.0 (59.0) 59.1 (59.0)	1.5
BeagleBone Black	128 256	24.5 (20.8) 27.0 (22.8)	28.0 (24.1) 26.9 (24.2)	1.0
TinkerBoard	128 256	54.7 (n/a) 41.6 (n/a)	57.0 (n/a) 43.7 (n/a)	1.8
ODROID C2	128 256	45.6 (46.8) 38.1 (47.7)	48.4 (51.3) 39.3 (51.3)	1.5
ODROID XU4	128 256	74.6 (59.4) 59.8 (59.4)	70.4 (62.2) 56.6 (62.3)	2.0

inefficiency that inflates the cluster's power consumption. Lastly, the power supply for the cluster (which supports upwards of 80 Pis) and the network switch dominate power consumption at lower numbers of Pis.

Power-efficient configurations are possible at lower numbers of nodes by setting up a cluster using individual power supplies and a power-efficient switches. For example, a 6-node power-efficient Raspberry Pi cluster consumes just 37.6 watts of power when performing the cryptographic benchmarks, corresponding to a reduction of approximately 24 watts. This cluster utilizes a smaller 8-port power-efficient Gigabit switch that costs $19.00 and consumes roughly 2 watts of power. The total estimated cost of the 6-node cluster is $320.00.

V. Other Platforms

While Raspberry Pis are the most ubiquitous SBCs on the market, we provide a brief comparison of the cryptographic performance of the Raspberry Pi 3B+ to some other well-known SBCs. Table III summarizes some of these results.

Interestingly, while the Pi 4 has higher encryption throughput than the Pi 3B+ for Twofish, it has a lower throughput for AES. While the reason for this result is unknown, it does support the argument that encryption methods such as Twofish require a closer look for implementation on the Raspberry Pi.

In terms of AES, the Raspberry Pi 3B+ [42] has similar or better performance to many of the other SBCs on the market, with the exception of the ODROID XU4 [43] and the TinkerBoard [44]. We note that while the TinkerBoard has higher throughput for AES, it lacks the necessary kernel modules for Twofish or Serpent. Lastly, we note the higher power consumption of the Odroid XU4 and Raspberry Pi 4 SBCs make these boards incompatible with the BitScope Blade. Scientists looking to adapt these boards into clusters will need to build their own custom racks. As commercial clustering solutions for the Raspberry Pi 4 and Odroid XU4 become available, we anticipate wider use of these systems for data summarization tasks in the scientific community.

VI. Conclusions

As scientific computing continues to grow in edge environments, security becomes an increasing concern. Specifically, new regulations require security for data-at-rest in edge environments and secure end-to-end communication for data travelling from the edge. Scientists must therefore examine the utility of their systems from a security perspective. Popular commodity systems used by scientists for data summarization tasks include mid-range desktop (MRD), next-unit of computing systems (NUCs), and ARM-based single board computer (SBC) clusters. The popularity of SBC clusters have grown in recent years due to their inexpensiveness and extensibility.

This paper evaluates the cryptographic performance of a Raspberry Pi cluster to an Intel-based NUC and MRD system representing commodity options typically used by scientists performing local data analysis close to the edge. The authors were specifically interested in evaluating the performance of the systems on storage encryption tasks. To this end, we used CryptSetup, a standard package for storage encryption that includes a benchmark utility. AES, Twofish and Serpent, three cryptographic algorithms that are widely implemented in Linux, were used for the study.

Our results show that on larger key sizes, Twofish has the highest encryption throughput on the Raspberry Pi cluster, and similar decryption throughput to AES. We also note that a 6-node Raspberry Pi cluster (24 ARM cores) yields similar Twofish encryption performance to the NUC and MRD systems and at lower power consumption. These results suggest that Twofish should be examined more closely for data encryption/decryption tasks on SBCs such as the Raspberry Pi. Our results also support prior work in demonstrating the extensibility of SBC clusters. For scientists operating on smaller budgets, it is often more cost-effective to add or remove nodes as needed for analysis.

Both the NUC and MRD systems include AES hardware instructions. Consequently, the NUC and MRD outperform the cluster on AES encryption and decryption tasks. We note that while the AES instruction set is supported by ARMv8, it is only listed as optional [45], and not implemented in any known SBCs. Recently, ARM opened up its instruction set to pave the way for workload-specific compute [46]; our results support the notion that hardware instructions for AES should be implemented on ARM-based SBCs to maximize their ability to perform encryption tasks in edge environments. Additionally, our results suggest that the creation of hardware instructions for Twofish will increase the performance of cryptographic applications on Raspberry Pis and other SBCs.

Future work will concentrate on expanding our evaluation strategy to include the performance of asymmetric approaches. We anticipate our results will be useful for SBC developers and scientists alike.

Acknowledgment

Funding for this project is provided the U.S. Army Futures Command, CCDC Armaments and the DOD High Performance Computing Modernization Program (HPCMP). The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Army, Department of Defense or the U.S. Government.

REFERENCES

[1] R. Longbottom, "Linpack benchmark results on PCs," Internet Website, Last accessed 11/13/2019, 2017, http://www.roylongbottom.org.uk/linpack%20results.htm.

[2] C. Pahl, S. Helmer, L. Miori, J. Sanin, and B. Lee, "A container-based edge cloud PaaS architecture based on raspberry pi clusters," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, Aug 2016, pp. 117–124.

[3] L. Miori, J. Sanin, and S. Helmer, "A platform for edge computing based on raspberry pi clusters," in *Data Analytics*, A. Calì, P. Wood, N. Martin, and A. Poulovassilis, Eds. Cham: Springer International Publishing, 2017, pp. 153–159.

[4] D. Fernández-Cerero, J. Y. Fernández-Rodríguez, J. A. Álvarez-García, L. M. Soria-Morillo, and A. Fernández-Montes, "Single-board-computer clusters for cloudlet computing in internet of things," *Sensors*, vol. 19, no. 13, p. 3026, 2019.

[5] Y. Jiang, Y. Chen, S. Yang, and C. Wu, "Energy-efficient task offloading for time-sensitive applications in fog computing," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2930–2941, Sep. 2019.

[6] P. J. Basford, S. J. Johnston, C. S. Perkins, T. Garnock-Jones, F. P. Tso, D. Pezaros, R. D. Mullins, E. Yoneki, J. Singer, and S. J. Cox, "Performance analysis of single board computer clusters," *Future Generation Computer Systems*, vol. 102, pp. 278 – 291, 2020.

[7] S. J. Matthews, "Harnessing single board computers for military data analytics," in *Military Applications of Data Analytics*. Auerbach Publications, 2018, pp. 63–77.

[8] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, "A scalable and privacy-aware IoT service for live video analytics," in *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 2017, pp. 38–49.

[9] C. Bachhuber, A. S. Martinez, R. Pries, S. Eger, and E. Steinbach, "Edge cloud-based augmented reality," in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2019, pp. 1–6.

[10] R. Morabito, "Inspecting the performance of low-power nodes during the execution of edge computing tasks," in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2017, pp. 148–153.

[11] A. R. Elias, N. Golubovic, C. Krintz, and R. Wolski, "Where's the bear?-automating wildlife image processing using IoT and edge cloud systems," in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2017, pp. 247–258.

[12] V. Cozzolino, A. Y. Ding, J. Ott, and D. Kutscher, "Enabling fine-grained edge offloading for IoT," in *Proceedings of the SIGCOMM Posters and Demos*. ACM, 2017, pp. 124–126.

[13] R. Morabito, "Virtualization on internet of things edge devices with container technologies: a performance evaluation," *IEEE Access*, vol. 5, pp. 8835–8850, 2017.

[14] R. Dautov, S. Distefano, D. Bruneo, F. Longo, G. Merlino, A. Puliafito, and R. Buyya, "Metropolitan intelligent surveillance systems for urban areas by harnessing IoT and edge computing paradigms," *Software: Practice and Experience*, vol. 48, no. 8, pp. 1475–1492, 2018.

[15] B. Qureshi, K. Kawlaq, A. Koubaa, B. Sultan, and M. Younis, "A commodity SBC-edge cluster for smart cities," *arXiv preprint arXiv:1902.06661*, 2019.

[16] European Commission, "2018 reform of EU data protection rules," Online at https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

[17] California Legislative Information, "The California Consumer Privacy Act of 2018 (CCPA)," Online at https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375, 2018.

[18] Centers for Medicare & Medicaid Services, "The Health Insurance Portability and Accountability Act of 1996 (HIPAA)," Online at http://www.cms.hhs.gov/hipaa/, 1996.

[19] A. Petitet, C. Whaley, J. Dongarra, A. Cleary, and P. Luszczek, "HPL - a portable implementation of the high-performance linpack benchmark for distributed-memory computer," Internet Website, Last accessed 10/10/2019, 2012, http://www.netlib.org/benchmark/hpl/.

[20] D. Hawthorne, M. Kapralos, R. W. Blaine, and S. J. Matthews, "CryptoPi - raspberry pi cryptographic testbed," Internet Website, Last accessed 11/13/2019, 2019, https://github.com/dshawth/cryptopi.

[21] S. J. Cox, J. T. Cox, R. P. Boardman, S. J. Johnston, M. Scott, and N. S. O'Brien, "Iridis-pi: a low-cost, compact demonstration cluster," *Cluster Computing*, vol. 17, no. 2, pp. 349–358, Jun 2014. [Online]. Available: https://doi.org/10.1007/s10586-013-0282-7

[22] F. P. Tso, D. R. White, S. Jouet, J. Singer, and D. P. Pezaros, "The glasgow raspberry pi cloud: A scale model for cloud computing infrastructures," in *2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops*. Philadelphia, PA: IEEE, July 2013, pp. 108–112.

[23] P. Abrahamsson, S. Helmer, N. Phaphoom, L. Nicolodi, N. Preda, L. Miori, M. Angriman, J. Rikkilä, X. Wang, K. Hamily *et al.*, "Affordable and energy-efficient cloud computing clusters: The bolzano raspberry pi cloud cluster experiment," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, vol. 2. IEEE, 2013, pp. 170–175.

[24] J. Kiepert, "Creating a raspberry pi-based beowulf cluster," Boise State University, Tech. Rep., 2013.

[25] D. Papakyriakou, D. Kottou, and I. Kostouros, "Benchmarking raspberry pi 2 beowulf cluster," *International Journal of Computer Applications*, vol. 179, no. 32, pp. 21–27, 2018.

[26] W. E. Burr, "Selecting the advanced encryption standard," *IEEE Security Privacy*, vol. 1, no. 2, pp. 43–52, March 2003.

[27] D. Hawthorne, "A quantitative study of advanced encryption standard performance as it relates to cryptographic attack feasibility," Ph.D. dissertation, Colorado Technical University, 2018.

[28] S. J. Matthews, R. W. Blaine, and A. F. Brantly, "Evaluating single board computer clusters for cyber operations," in *2016 International Conference on Cyber Conflict (CyCon U.S.)*. Washington, DC: IEEE, Oct 2016, pp. 1–8.

[29] P. Augustynowicz and A. Buraczyńska, "Comparison between experimental, analytical and simulation model of distributed computation on ARM processors in high-performance computing," *Computer Science and Mathematical Modelling*, vol. Well. 5, pp. 5–10, 2017.

[30] G. Leurent and T. Peyrin, "SHA-1 is a shambles," Online at https://shambles.github.io/, 2020.

[31] N. Rachmat *et al.*, "Performance analysis of 256-bit AES encryption algorithm on android smartphone," *Journal of Physics: Conference Series*, vol. 1196, no. Conference 1, 2019.

[32] A. Biryukov and J. Großschädl, "Cryptanalysis of the full AES using GPU-like special-purpose hardware," *Fundamenta Informaticae*, vol. 114, no. 3-4, pp. 221–237, 2012.

[33] J. Grossschadl, S. Tillich, C. Rechberger, M. Hofmann, and M. Medwed, "Energy evaluation of software implementations of block ciphers under memory constraints," in *2007 Design, Automation Test in Europe Conference Exhibition*, April 2007, pp. 1–6.

[34] S. Derhami, "Software performance analysis for ARM architectures," Department of Computer Science, Linnaeus University, Tech. Rep., 2015.

[35] C. Strydis, D. Zhu, and G. N. Gaydadjiev, "Profiling of symmetric-encryption algorithms for a novel biomedical-implant architecture," in *Proceedings of the 5th Conference on Computing Frontiers*, ser. CF '08. New York, NY, USA: ACM, 2008, pp. 231–240. [Online]. Available: http://doi.acm.org/10.1145/1366230.1366272

[36] S. Gueron, "Intel® advanced encryption standard (AES) new instructions set," Intel Corporation, Tech. Rep. 323641-001 Revision 3, 2010.

[37] Bitscope Blade Solutions, "Bitscope br20a," Internet Website, Last accessed 09/27/2019, 2019, http://my.bitscope.com/store/?p=view&i=product+BR20A.

[38] R. C. Whaley and A. Petitet, "Minimizing development and maintenance costs in supporting persistently optimized BLAS," *Software: Practice and Experience*, vol. 35, no. 2, pp. 101–121, February 2005, http://math-atlas.sourceforge.net/.

[39] D. Hawthorne, "DCS files," Internet Website, Last accessed 11/13/2019, 2019, https://github.com/dshawth/DCS.

[40] ——, "DMUX files," Internet Website, Last accessed 11/13/2019, 2019, https://github.com/dshawth/DMUX.

[41] P3 International, "Killawatt," Internet Website, last accessed 11/12/2019, 2018, http://www.p3international.com/products/p4400.html.

[42] Raspberry Pi Foundation. (2019) Raspberry pi 3 model B+. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

[43] R. Roy and V. Bommakanti, "Odroid XU4 user manual," Online at https://magazine.odroid.com/wp-content/uploads/odroid-xu4-user-manual.pdf, 2020.

[44] ASUSTeK Computer Inc., "Tinkerboard," Online at https://www.asus.com/us/Single-Board-Computer/Tinker-Board/, 2020.
[45] Arm Limited, *Arm Architecture Reference Manual Armv8, for Armv8-A architecture profile*, https://static.docs.arm.com/ddi0487/ea/DDI0487E_a_ armv8_arm.pdf.
[46] Arm Developer, "Arm custom instructions," Internet Website, Last accessed 11/15/2019, 2019, https://developer.arm.com/architectures/instruction-sets/custom-instructions.