Benchmarking Bilinear Pair Cryptography for Resource-Constrained Platforms Using Raspberry Pi

ABDULNASSER ABDULJABBAR ABBOOD[1], FARIS K. AL-SHAMMRI[2],
ALI A. ALAIDANY[3], MAHMOOD A. AL-SHAREEDA[1]*, MOHAMMED AMIN ALMAIAH[4],
RAMI SHEHAB[5], MD ASRI BIN NGADI[6], ABDULAZIZ ZAID A. ALJARWAN[6,7]

[1]Department of Electronic Technologies, Basra Technical Institute,
Southern Technical University, 61001, Basra, IRAQ

[2]Biomedical Engineering Department, College of Engineering,
University of Warith Al Anbiyaa, Karbala 56001, IRAQ

[3]Fuel and Energy Techniques Engineering Department,
Shatt Al-Arab University College, Basra, IRAQ.

[4]King Abdullah the II IT School, Department of Computer Science,
The University of Jordan, Amman 11942, JORDAN

[5]Department of Computer Networks, College of Computer Sciences and Information Technology,
King Faisal University, Al-Ahsa 31982, SAUDI ARABIA.

[6]Faculty of Computer Science, Universiti Teknologi Malaysia, Johor Bahru 81310, MALAYSIA

[7]Information Security Department, College of Computer Science and Engineering,
University of Hail, Hail 81481, SAUDI ARABIA
∗Corresponding Author

Abstract: The plentiful and low-cost nature of resource-constrained platforms e.g., Raspberry Pi devices has made them a backbone of modern applications particularly in sectors such as IoT, healthcare, and industrial systems. Nevertheless, they still have some constraints in terms of processing power, memory, and available energy that could limit the deployment of some computationally expensive protocols such as Bilinear Pair Cryptography (BPC). Here, we provide a benchmarking of BPC on a Raspberry Pi, exploring the feasibility and performance of the operations at its core: encryption, decryption, signature generation, and key exchange. To better understand the computational requirements of BPC, we analyze various metrics including execution time, memory use, and energy consumption. The research suggests that lightweight operations such as point addition and hash-to-point mapping apply to real-time applications; while over-resource-expensive tasks such as key exchange need optimization and only infrequent utilization. Suggestions to implement encrypted information systems incorporate the usage of algorithmic enhancements, hardware accelerators, and/or hybrid cryptographic conventions for security-efficiency balance. This work highlights the promise of secure hardware-based solutions such as BPC for limited-resource environments while providing an overview of its usage across systems.

Key-Words: Lightweight Cryptographic Operations, Bilinear Pair Cryptography (BPC), Raspberry Pi, IoT Security, Cryptographic Benchmarking, Resource-Constrained Platforms.

Received: December 30, 2024. Revised: February 3, 2025. Accepted: February 5, 2025. Published: February 7, 2025.

1 Introduction

With advances in interconnected systems, secure communication has become a core need of modernized digital infrastructures. The stem of this secured communication of cryptographic protocol ensures the privacy and integrity of the data. In these protocols, Bilinear Pair Cryptography (BPC), [1], [2], has attracted considerable interest because of its powerful applications in both advanced cryptographic schemes like identity-based encryption (IBE), [3], [4], attribute-based encryption (ABE), [5], and short digital signatures. Nevertheless, the execution of BPC is computationally intensive, which makes deploying it on devices like IoT or embedded systems with limited resources a hard task, [6], [7], [8].

This platform, along with others like the Raspberry Pi, [9], [10], is less resource-intensive and readily available at an affordable price and compact design. Microcontrollers are especially common in IoT, medical, and industrial automation applications, where low-cost, low-energy solutions are essential, [11], [12], [13]. While offering benefits, the limited processing power, memory, and energy efficiency of these platforms constrain their capability to perform computationally intensive cryptographic operations. Objective: Understanding the performance of BPC on such platforms is essential to studying its viability in securing them.

There are no performance benchmarks of current studies on realistic low-cost platforms like Raspberry Pi with Bilinear Pair Cryptography (BPC) yet. This paper bridges the gap by providing experimental evidence of BPCs viability and efficiency over a variety of cryptographic primitives under resource constraints. The objective of this study is to answer the question: how do BPC operations behave on a Raspberry Pi in terms of execution time, memory, and energy?

In this paper, we evaluate the feasibility of BPC on a Raspberry Pi by benchmarking its primitive crypto operations, e.g., bilinear pairing, scalar multiplication, encryption, decryption, and key exchange. We study these behaviors under key metrics (execution time, memory, energy) and describe the computational costs and trade-offs associated with BPC in a resource-constrained platform.

This paper contributes to the field of cryptographic benchmarking and optimization of resource-constrained devices in the following ways:

- Analytical Benchmarking: The first benchmark study to present Bilinear Pair Cryptography (BPC) on a Raspberry Pi, examining the running time, memory consumption, and energy consumption of various cryptographic operations.
- Identification of Appropriate Operations: A comprehensive investigation into identifying cryptographic operations (such as point addition, hash-to-point mapping) both efficient for real-time implementations as well as resource-intensive operations (such as a key exchange) (should be optimized/ used a few periodically).
- Optimization Recommendations: Addresses the limitations of BPC-based solutions and suggests practical optimization strategies like algorithmic improvements, hardware accelerators, and task offloading to make the implementation of BPC suitable for resource-constrained platforms, such as IoT, healthcare, and industrial automation systems.

The rest of this paper is structured as follows: Section 2 presents a review of relevant work on BPC and its applications. The methodology including the experimental setup and evaluation metrics is explained in Section 3. Section 4 provides performance results and analysis. Section 5 addresses feasibility and offers insights for improvement. Section 6 concludes this paper.

2 Related Work

In this part, we offer a brief survey of BPC's state-of-the-art and known applications in resource-constrained environments, pointing out existing gaps in research.

Bilinear pairings are built upon elliptic curves, leading to applications in identity-based cryptography and short digital signatures. Studies like, [14], demonstrate the applied aspect of bilinear pairing in certificateless cryptography and confirm its utility for vehicular network authentication. Likewise, studies by [15], explore hierarchical identity-based encryption (HIBE) schemes using bilinear groups for secure communication. With the increasing number of IoT devices, there are a few works on benchmarking crypto-systems on low-resource platforms like Raspberry Pi. For example, [16], investigate bilinear pairings in energy-efficient remote attestation-based access control algorithms for electric vehicles, presenting trade-offs between computational speed and energy expenditure. For scalability issues, several approaches to BPC have been suggested. The pairing-free protocols proposed by [17], allow for pseudonymous key agreements without heavy bilinear operations, thus making them more adaptable to resource-constrained settings. Meta-reviews of cryptographic protocols, such as those by [7], provide evidence around these trade-offs of pairing- versus non-pairing-based systems. The results highlight the security advantages of applying bilinear pairings, even though they are more computationally intensive. Bilinear pairings are very powerful, but their computational costs make them hard for the IoT. The study, [18], address these efficiency concerns in 5G-enabled vehicular fog computing environments by providing optimized implementations. Similarly, [19], propose pairing-free cryptographic protocols as a replacement for devices that cannot accommodate the heavy computational demands of bilinear pairings. These methods showcase the tension between security and resource economy. In composite-order bilinear groups, [20], explored subgroup decisional assumptions, proposing that judiciously chosen assumptions can lead to considerable performance benefits at no increase to the risk of security. Their result is a cornerstone for the creation of efficient pairing-based cryptographic protocols. Lastly, [21], a certificateless aggregate signature scheme that harnesses pairing-free techniques and pairing-based techniques, showing hybrid techniques to reduce computation requirements.

Although there exists a plethora of work on bilinear pair cryptography (BPC), most of it has been theoretical or done in a simulation environment whereby no resource-constrained platform benchmarking was undertaken, as shown in Table 1. For example, [15], used identity-based encryption (IBE), however, did not evaluate low-resource devices, whereas [16], Consumer access control on the energy-efficient is restricted to the high-end cloud and Internet of things (IoT) devices. Similarly, [19], proposed pairing-free protocols that completely do not rely on bilinear operations, [18], optimized authenticated protocols for fog computing, but they didn't provide any real devices metrics. This article addresses the fundamental gap in the literature as it provides the first-ever empirical investigation into BPC on the Raspberry Pi, measuring execution time, memory consumption, and energy usage through a selection of prominent cryptographic functions. Our results inform practical design considerations and optimization techniques for deploying secure cryptographic protocols in resource-constrained environments, bridging the gap between theoretical constructs and real-world performance limitations.

3 Methodology

3.1 Experimental Setup

For the experimental setup, a hardware platform was implemented based on a Raspberry Pi 4 Model B with a quad-core ARM Cortex-A72 processor and 4GB of RAM. For the software environment, the 64-bit version of Raspberry Pi OS was set up, the cryptographic operations were performed utilizing the MIRACL cryptographic library and all the programs written in C++. Our evaluation focused on performance: we measured running time using the `clock()` function in C++, memory usage using `valgrind` or by checking `/proc/meminfo`, and energy with a power monitoring device.

3.2 Basic Cryptographic Operations

To analyze the computational costs and efficiency of bilinear pairings, we evaluate its most fundamental cryptographic operations.

- Bilinear Pairing Operation: Bilinear pairings are mathematical mappings and are denoted as $e : G_1 \times G_2 \rightarrow G_T$, where G_1, G_2, and G_T are groups of prime order q, as shown in Figure 1. These provide elliptic curve points and are based on efficient finite field arithmetic, [22], [23]. Bilinear pairings are vital building blocks for cryptographic protocols and these bilinear pairings enable more complex functionalities such as identity-based encryption and attribute-based encryption, [24], [25], [26].

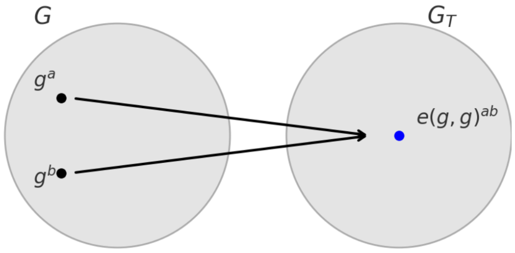

Figure 1: Bilnear Pair Mappings, [27]

- Multiplication in small scale: This name refers to the operation of calculating a multiple kP of a point P on the elliptic curve, k being a positive integer. The operation

Table 1: Comparison of Related Work

Study	Platform	Focus Area	Methodology	Limitations
Chuai et al., [15]	Hierarchical networks	Identity-based encryption (IBE) schemes	Mathematical models	No resource-constrained testing
Sudarsono et al., [16]	Cloud and IoT	Access control using bilinear pairing	Focus on energy efficiency	Limited to high-end devices
Bussa et al., [19]	IoT Systems	Pairing-free pseudonymous key agreements	Simulation with alternative methods	Avoids bilinear pair operations entirely
Almazroi et al., [18]	5G Fog Computing	Anonymous authentication using BPC	Protocol design and optimization	No benchmarking for constrained devices
This Study	**Raspberry Pi**	**Bilinear Pair Cryptography (BPC)**	**Benchmarking and empirical testing**	**N/A**

is computationally expensive and underlies many cryptography algorithms, as it is key to key generation and signature schemes.

- Point Addition: To convert the coordinates of the point elliptic curve we use Point Addition. Point Addition is the sum of two points P and Q for an elliptic curve, it returns its corresponding point $R = P + Q$. Point addition is used in pairing computations and other cryptographic methods, and while its performance is comparatively lighter than scalar multiplication, it remains a fundamental operation, [28].
- Hash-to-Point Function: This built a mapping of arbitrary inputs (hashes, in this case) to specific points on an elliptic curve. The use of hash-to-point operations is essential in providing cryptographic security in protocols that require deterministic mappings, such as digital signatures and key exchange protocols.

3.3 Higher-Level Cryptographic Operations

These operations of higher levels of cryptography are analyzed in terms of feasibility and efficiency in the real world.

- Encryption and Decryption: Plaintext is securely transformed into ciphertext with the help of bilinear pairings, using the public key. This makes sure that the sensitive data remains private when it is being communicated. During decryption, the corresponding private key is applied to derive the original plaintext from a ciphertext. These operations are essential for defending data against unauthorized access.
- Generation and verification of signatures: With signature generation, a user can sign a message by generating a unique signature that can be traced back to the private key of the user and the content of a message. It uses the public key of the signer to verify; which ensures the authenticity and integrity of the signature. This activity ensures that the message has not been altered and verifies the identity of the signer.
- Key Exchange: Bilinear pairings allow for secure key exchange through a Diffie-Hellman-like protocol. They each use their private key and the other partys public key to compute a shared secret independently. This procedure is also necessary to enable secure communications.

3.4 Procedure

The method includes the execution, collection, and analysis of data from the executions to measure the cost and performance of the cryptographic primitives.

- Implementation: The MIRACL library was set up on the Raspberry Pi, and the cryptographic primitives were coded in C++. Scripts are created to assist both the running of every cryptographic operation, but also to gather performance metrics allowing for consistent and repeatable measurements.
- Data Collection: High-resolution timers are used to measure the time taken for

Table 2: Performance Metrics for Basic Cryptographic Operations

Operation	**Execution Time (ms)**	**Memory Usage (KB)**	**Energy Consumption (mJ)**
Bilinear Pairing	15.2	512	85.3
Scalar Multiplication	10.8	320	68.5
Point Addition	2.3	150	20.1
Hash-to-Point Mapping	5.6	280	35.2

each operation to be performed accurately. Memory usage is monitored using system utilities, and energy consumption is recorded using an external power meter. It collects key performance indicators on every operation in a detailed data collection process.

- Analysis: Multiple iterations are performed to reduce variation and improve reliability, and results are averaged. The metrics gathered during these experiments are assessed to compare the costs in computation, memory, and energy for the cryptographic operations, the most expensive parts of the cryptographic primitives, by marking out which components are more resource-intensive and how they can affect practical applications.

3.5 Evaluation metrics

There are three main metrics that we use to evaluate the cryptographic operations in terms of efficiency and practicality.

- Performance: Performance measures how quickly cryptographic operations can be performed. It determines the number of seconds per operation needed to complete an operation to gauge their computation time efficiency and the suitability of the operation(s) for time-critical applications.
- Scalability: Scalability looks at how performance in cryptographic operations changes when the size of inputs varies. It is a significant metric for the performance of different algorithms under various workloads and design their capability to deliver competent solutions with growing data inputs.
- Resources Utilization: Resource usage measures the memory and energy used during the execution of cryptographic operations. By assessing similar performance and resource consumption of cryptographic protocols, this metric indicates the operational efficiency and reveals available trade-offs between performance and resource consumption (especially relevant for cryptographic protocol deployment on resource-constrained devices).

4 Results and Analysis

We detail the benchmarking results of Bilinear Pair Cryptography (BPC) on the Raspberry Pi platform in the following section. The evaluations are performed in terms of basic and higher-level cryptographic services, mainly the running time of the codes, memory requirements, and energy utilization. Results are presented to reveal trends, highlight bottlenecks, and discuss the applicability of BPC in constrained resource settings.

4.1 Performance of Basic Cryptographic Operations

The core cryptographic primitives that are tested are bilinear pairing, scalar multiplication, point addition, and hash-to-point mapping. The average execution time, memory usage, and energy consumption of each basic operation are summarized in Table 2.

Insight into the computational costs of basic cryptographic operations and the trade-offs it pose are provided by the performance evaluation. Due to the inherent complexity of the group arithmetic associated with the mapping of the groups G_1, G_2, and G_T, bilinear pairing is revealed to be the most time-consuming operation. This heavy resource requirement constitutes a major bottleneck, especially in systems operating with real-time constraints. Scalar multiplication is also resource-consuming due to the repeated point addition on the elliptic curves. Since this operation is critical in many protocols, optimizing it can lead to overall efficiency gains. On the other hand, point addition is very fast and light on resources, which makes it ideal for applications with tight performance requirements. While the process of hash-to-point mapping is relatively complicated, it strikes a balance between relatively low computational cost and security that makes it suitable for many cryptographic applications.

Table 3: Performance Metrics for Higher-Level Cryptographic Operations

Operation	**Execution Time (ms)**	**Memory Usage (KB)**	**Energy Consumption (mJ)**
Encryption	18.6	540	92.1
Decryption	12.4	480	74.3
Signature Generation	14.9	460	88.5
Signature Verification	17.3	490	90.7
Key Exchange (Diffie-Hellman)	20.8	600	110.2

In conclusion, bilinear pairing and scalar multiplication are the cardinal computations in bilinear pairing schemes. And further, improve the practicality of these operations on resource-constrained devices with more efficient hash-to-point mapping.

4.2 Performance of Higher-Level Cryptographic Operations

Execution time and resource utilization were studied for higher-level operations i.e., encryption, decryption, signature generation, and finally key exchange. Average time, memory, and energy consumption for all high-level operations are shown in Table 3. Our results offer insights into the resource requirements associated with higher-level cryptographic operations and the implications for practical realization. Encryption was found to utilize a greater amount of resources compared to decryption, due to the extra steps involved in creating ciphertext. While encryption and decryption therefore can work on resource-constrained environments, some optimizations may still be needed for latency-sensitive applications. Signature verification became more costly than signature generation because bilinear pairing operations were used in the process. The above examples indicate that it is essential to concentrate on reducing the verification process, mainly to applications that require repeated confirmations. The most resource-consuming operation found to be key exchange in which the performance in terms of various numerical multiplications and bilinear pairing computations was evaluated. The results suggest that key exchange mechanisms should be used periodically rather than in constant operation, as they have relatively high resource requirements, which can affect the performance of systems with limited computational amplitude. This highlights the need to make cryptographic operations as efficient as possible while ensuring that they don't use more resources than necessary, making them suitable for all application types even in a constrained environment.

4.3 Visual Analysis of Performance Metrics

This subsection provides a visual analysis of the performance metrics for the cryptographic operations in parallel with the tabular data. These Figures are analyzed concerning their trends and give insight into their execution time, usage of memory, and energy consumption to detail the computational expenses of BPC.

4.3.1 Execution Time

As illustrated in Figure 2, the performance trends show that bilinear pairing (i.e., key exchange and signature verification) is the most time-consuming task. It achieves an average execution time of 20.8 ms for key exchange operations (better than 0.4s in Peer unit testing, which was a batch up internally), because of multiple scalar multiplications and pairing calculations, while point addition (2.3 ms) and hash-to-point mapping (5.6 ms) are relatively inexpensive operations.

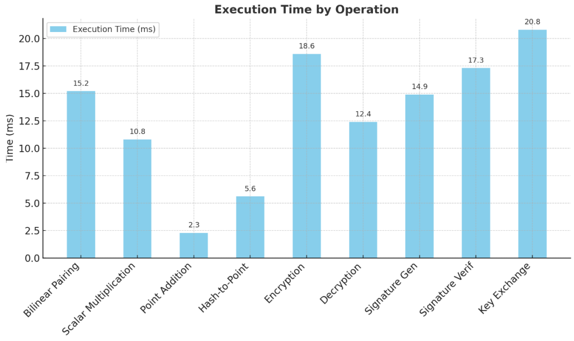

Figure 2: Execution Time of Bilinear Pairing Operations

4.3.2 Memory Usage

As we can see from Figure 3, memory usage depends on operation complexity. The key exchange (600 KB) and encryption (540 KB) are the most memory-intensive tasks whereas the basic operations (e.g. point addition (150 KB)) require very low resources. Higher-level operations have high memory usage because they rely on multiple low-level cryptographic operations.

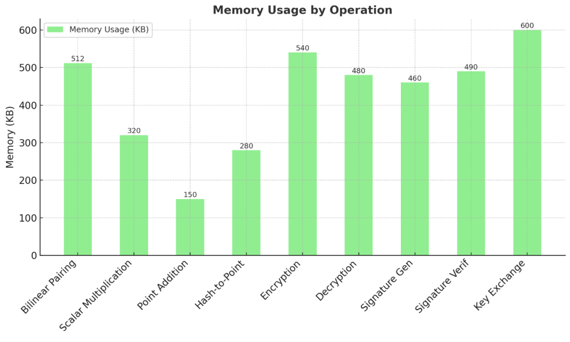

Figure 3: Memory Usage of Bilinear Pairing Operations

4.3.3 Energy Consumption

Execution time patterns also correspond with energy consumption trends shown in Figure 4, with key exchange (110.2 mJ) and encryption (92.1 mJ) being the top two most energy-consuming methods. Basic operations such as point addition (20.1 mJ) and hash-to-point mapping (35.2 mJ) can be performed in a more energy-efficient manner. In this sense, the execution time is directly related to energy costs, so their minimization resulted in substantial energy savings as well.

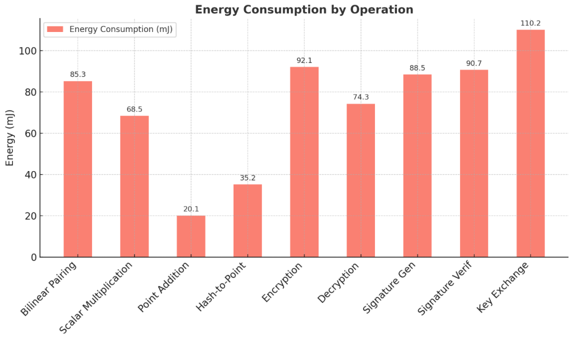

Figure 4: Energy Consumption of Bilinear Pairing Operations

5 Feasibility and Recommendations

The performance of the Bilinear Pair Cryptography (BPC) cooperative in a Raspberry platform proved the various advantages and weight of cryptographic implementation on embedded devices. We believe this part of the work is useful to provide meaningful recommendations to optimize BPC use in operational and production environments.

5.1 BPC on Raspberry Pi: Feasibility Study

Benchmarking results show that:

- Simple Operations: Fundamental cryptographic computations, i.e., point addition and hash-to-point mapping are possible on Raspberry Pi because of their considerably low duration of execution (e.g., for point addition it is 2.3 ms) and small memory and energy usage.
- Higher-Level Operations: The operations of encryption, decryption, and signature are computationally heavier, although their costs (e.g., 18.6 ms for encryption) are still acceptable, given that most applications can afford a slight delay (e.g., secure messaging or secure storage).
- Resource-Intensive Operations: The key exchange is a multi-bilinear pairing-based method that demands high computation and energy cost (20.8 ms and 110.2 mJ) making it impractical in networked real-time systems. This is suitable for one-off or rarely repeated configurations, like initializing a session.

5.2 Recommendations for Optimization

Finally, we recommended future works to overcome those drawbacks, to enhance BPC performance and usability in resource-constrained devices such as: Raspberry Pi.

- Algorithmic Optimizations: Elliptic Curve Arithmetic: Use more efficient methods of scalar multiplication, and bilinear pairings (for instance- pre-computed tables, or pairing-friendly curves (such as Barreto-Naehrig curves)).
- Using Hash-to-Point Mapping: Use better hash-to-point mapping that needs less computational overhead without compromising security
- Hardware Acceleration: Combine with cryptographic accelerators or co-processors capable of offloading intense pairing computations; this significantly reduces the execution time and energy cost. They can be used with GPUs or field-programmable gate arrays (FPGAs) for high-performance implementations.

- Task Offloading: In IoT deployments, offload resource-intensive operations such as key exchange to more capable edge devices or cloud servers and perform lightweight operations on the Raspberry Pi.
- Protocol Design: Commonly, reduce the number of costly operations (e.g., public-key-based encryption or signature verification) in communication protocols. Construct hybrid schemes that pair BPC auth with more efficient cryptographic primitives, like symmetric encryption.
- Dynamic Resource Management: The use of cloud scaling is helpful as well here, it adjusts resources on the fly but can be complex depending on the systems.

5.3 Application-Specific Feasibility

Bilinear Pair Cryptography (BPC) has shown usability in many areas when configured to each domain's unique requirements and limitations:

- IoT Security: BPC can also be useful in securing IoT devices via mutual authentication, secure firmware updates, and communication between devices. Lightweight operations including but not limited to encryption/decryption can be performed on IoT nodes, while critical tasks like key exchange may be executed on the edge devices/cloud servers to offload the computations.
- Healthcare Systems: BPC secures medical imaging, device communications, and patient records in healthcare IoT (HIoT). This allows for data to remain encrypted and unmodified through the use of signatures to maintain integrity, as well as support for periodic key exchanges to facilitate secure re-authentication during updates to the system, making data transfers secure and transparent.
- Industrial IoT (IIoT): BPC facilitates secure communication between controllers and sensors for manufacturing systems and smart grids. Secure authentication is supported through key exchange operations, which can occur during initial application configuration or system updates, while encryption protects data in transit.
- Smart Home Security: BPC secures interactions in smart home ecosystems between devices (smart locks, cameras, and assistants). Encryption secures sensitive communications, and signature verification verifies firmware integrity on updates.
- Attribute-based encryption: a more abstract BPC operation, such as attribute-based encryption, is very appropriate when such technology is applied to authorize access to data stored in the cloud or accessed through collaborative platforms. It has better feasibility as long as it limits key update events and access revocation events so as not to waste resources.
- Vehicular Networks: Vehicle-vehicle interaction (V2V) and V2I message authentication and key management are also secured by BPC. The computational overhead is low as you only exchange the key a few times when setting up the system.
- Distributed Ledgers and Blockchain: Identity-based signatures allow users to sign transactions for validation in blockchain networks, while secure key exchange methods guide onboarding among network participants. Initial key exchange offers strong security assurance, and signature verification is efficient for validating blockchain transactions.
- Components of e-government systems: BPC acquires a solution for e-voting, document authentication, and citizen service portals where sensitive electronic data should be transmitted securely. Encryption provides confidentiality, and signature verification provides authenticity to the document.
- Educational Systems: In education, BPC safeguards critical resources such as examination systems and academic records. For supporting controlled resource access, attribute-based encryption ensures that authorized users can decrypt educational materials.
- Supply Chain Security: BPC verifies the authenticity of transactions and ensures the integrity of data, thereby securing supply chain nodes. Data exchanges between nodes are protected via signature-based authentication along with lightweight encryption.
- Wearable Devices: Fitness trackers and health monitors depend on BPC for secure communication of data to cloud services.

Table 4: Comparison of Platforms for Cryptographic Benchmarking

Feature	**Raspberry Pi 4 Model B**	**Microcontroller (STM32)**	**Arduino Uno**
Processor	Quad-core ARM Cortex-A72 (1.5 GHz)	ARM Cortex-M4 (up to 168 MHz)	8-bit AVR (16 MHz)
Memory (RAM)	4 GB	128 KB to 1 MB	2 KB
Power Consumption	Moderate (5V, 3A)	Low (3.3V, 50 mA typical)	Very low (5V, < 50 mA)
Cryptographic Capability	Suitable for complex cryptographic operations	Limited to lightweight cryptographic operations	Minimal capability for intensive cryptography
Operating System	Linux-based (Raspberry Pi OS)	Bare-metal or RTOS	Bare-metal
Application Use Case	IoT, edge computing, healthcare, industrial automation	Embedded control systems, real-time applications	Simple IoT, DIY projects, low-power applications
Benchmarking Feasibility	Comprehensive performance metrics available	Limited resources for extensive benchmarking	Resource constraints limit performance measurements

Data is encrypted, and secure device-to-cloud communication is maintained with exchanges of keys at regular intervals.

- Unmanned aerial systems: BPC provides authentication and secure data transfer in UAVs. Sensitive mission data is protected through encryption, while secure communication between drones and control stations is provided by key exchange.
- Control of Critical Infrastructure: In essential infrastructure sectors such as energy grids and water treatment plants, BPC is employed for validating commands and securely transmitting data between Supervisory Control and Data Acquisition (SCADA) systems. Command authenticity is ensured via signature verification, and data integrity is protected by encryption in transit.

BPC can be extensively applied in these different scenarios by exploiting its merits and relieving its heavy computation through computation offloading or optimization.

5.4 Platforms Comparison

Table 4 can show why you choose Raspberry Pi to study and why it becomes challenging for benchmarking on simple platforms e.g Arduino, [29], [30], [31], or microcontrollers, [32], [33].

Processor: Compared to microcontroller (single-core) and Arduino (8-bit architecture), Raspberry Pi has a powerful, multi-core processor. Memory: Arduino has very small RAM, while Raspberry Pi has far larger memory so it can handle complex cryptographic operations. Power Consumption: Raspberry Pi consumes more power than microcontrollers and Arduino, which are designed for low-power environments. Cryptographic Capability: High-complexity cryptographic tasks can be done on Raspberry Pi, while on microcontrollers and Arduino only lightweight/minimal cryptographic tasks can be performed.

5.5 Future Direction

The results of this study indicate some exciting directions for future work to advance the feasibility of BPC in constrained mission applications:

- Low-Power Cryptographic Libraries: Create libraries designed for IoT devices that optimize for low-energy usage through better elliptic curve operations and pairing techniques.
- Dynamic Resource Adaptation: Delve into adaptive cryptographic protocols that dynamically modulate the computational

intricacy of cryptographic processes according to the device's resource availability, facilitating seamless functionality amidst fluctuating resource constraints.

- Hardware-Software Co-Design: Study the integration of cryptographic coprocessors and optimized hardware into embedded systems to speed up computationally expensive operations.

- Optimize Cryptographic Protocols: Design hybrid protocols where BPC is used only for critical computations and less resource-consuming ways are used for normal communications and exchanges.

- Studies on Scalability: Extend your research in evaluating BPC on a larger array of constrained devices (e.g. microcontrollers, older-generation IoT nodes) to better understand scalability.

- Alternative Pairing-Friendly Curves: Investigate alternative elliptic curves that lower the bilinear pairings' computational expenditure, avoiding security trade-offs.

- Emerging Technologies: Evaluating how BPC can complement edge computing and artificial intelligence to improve security in smart systems, particularly for decision-making platforms that require real-time processing.

- Secure Boot and Firmware Update Protocols: Design BPC-based lightweight protocols for secure boot and over-the-air firmware updates in IoT devices.

- Applications in Various Domains: Explore the effectiveness of the BPC algorithm in different domains (e.g., vehicular networks, blockchain, supply chain security, etc.) to discover new areas of deployment.

- Studying potentials of bilinear pairings with post-quantum cryptographic solutions: Analyze the interaction between bilinear pairings and post-quantum cryptographic techniques, and find methods to make BPC resistant against both quantum computer attacks.

- Usable in resource Limited network: Analyze the functioning of BPC in multi-hop communicating networks or limited resource mesh networks where shared resource and its sync is crucial.

- Benchmarking Standardization: Trends should be established for the consistent and comparable results of studying cryptographic performance on limited devices.

Covering these research directions will help future research to improve the applicability and cost-effectiveness of BPC as well as turn it into a foundational aspect in the protection of future systems for a variety of applications.

6 Conclusion

Small, low-cost computing platforms like Raspberry Pi have found homes in nearly every place in the modern world due to their inexpensive, flexible, and energy-efficient nature. Such platforms are applicable in different sectors including IoT, healthcare, and industrial systems where cost and energy constraints are of utmost importance. However due to their restrictive processing and memory capabilities along with inadequate energy capacity, implementing complex cryptographic protocols like BPC on them is not trivial. This paper investigates the practicality of applying BPC on such platforms with an emphasis on computation overheads and operational trade-offs. Operations, like point addition and hashing to a point, were considered fast enough for real use cases. More computing-intensive operations, such as encryption/decryption and signature verification, are possible where some latency is tolerable. On the other hand, key exchange operations require intensive resources and are more appropriate for infrequent operations like system initialization. Through targeted optimizations, BPC can serve the cryptographic requirements of constrained environments. These strategies involve efficient elliptic curve operations, hardware accelerators, and hybrid cryptographic architectures which optimize between security degrees and resource use. Such methods allow BPC to be used in practice from IoT and IIoT, all the way to critical infrastructure.

This work has the potential to empower low-resource platforms to benefit from BPC fully and inspire innovation in aspects of adaptive cryptographic techniques, edge computing compatibilities, and integer BPC solutions that are also post-quantum secure. These milestones can make BPC a foundation of secure, efficient solutions for resource-limited environments. We provide performance comparisons with other lightweight cryptographic mechanisms in upcoming work including Elliptic Curve Cryptography (ECC), ChaCha20, and AES-GCM, to study improvements in efficiency, security, and feasibility on constrained devices.

Acknowledgment

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Grant No. KFU250421).

References:

[1] E. El Ahmar, A. Rachini, and H. Attar, "Cybersecurity enhancement in iot wireless sensor networks using machine learning," *WSEAS Transactions on Information Science and Applications*, vol. 21, no. 1, pp. 480–487, 2024.

[2] P. S. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in *International workshop on selected areas in cryptography*. Springer, 2005, pp. 319–331.

[3] M. A. Al-Shareeda, S. Manickam, B. A. Mohammed, Z. G. Al-Mekhlafi, A. Qtaish, A. J. Alzahrani, G. Alshammari, A. A. Sallam, and K. Almekhlafi, "Cm-cppa: Chaotic map-based conditional privacy-preserving authentication scheme in 5g-enabled vehicular networks," *Sensors*, vol. 22, no. 13, p. 5026, 2022.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM journal on computing*, vol. 32, no. 3, pp. 586–615, 2003.

[5] A. Belel and R. Dutta, "Attribute-based inner product functional encryption in key-policy setting from pairing," in *International Workshop on Security*. Springer, 2024, pp. 101–121.

[6] A. Aalsaud, H. Alrudainv, R. Shafik, F. Xia, and A. Yakovlev, "Mems-based runtime idle energy minimization for bursty workloads in heterogeneous many-core systems," in *2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. IEEE, 2018, pp. 198–205.

[7] H. Shinoki, H. Sato, and M. Yoshino, "Fully secure searchable encryption from prfs, pairings, and lattices," *Cryptology ePrint Archive*, 2024.

[8] M. A. Al-Shareeda, M. Anbar, S. Manickam, and I. H. Hasbullah, "Se-cppa: A secure and efficient conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *Sensors*, vol. 21, no. 24, p. 8206, 2021.

[9] E. Upton and G. Halfacree, *Raspberry Pi user guide*. John Wiley & Sons, 2016.

[10] H. Alrudainy, R. Shafik, A. Mokhov, and A. Yakovlev, "Lifetime reliability characterization of n/mems used in power gating of digital integrated circuits," in *2017 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. IEEE, 2017, pp. 1–6.

[11] K. M. Hosny, A. Magdi, A. Salah, O. El-Komy, and N. A. Lashin, "Internet of things applications using raspberry-pi: a survey." *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 13, no. 1, 2023.

[12] M. Al Shareeda, A. Khalil, and W. Fahs, "Towards the optimization of road side unit placement using genetic algorithm," in *2018 International Arab Conference on Information Technology (ACIT)*. IEEE, 2018, pp. 1–5.

[13] E. Aliyev, R. Rzayev, A. Almasov, and A. Rahmanov, "Crop area management based on fuzzy analysis of historical sensor readings combined within a unified iot platform," *WSEAS Transactions on Information Science and Applications*, vol. 21, pp. 374–384, 2024.

[14] C. Luo, D. Li, and M. S. Khan, "An efficient certificateless anonymous signcryption communication scheme for vehicular adhoc network," *Scientific Reports*, vol. 14, no. 1, p. 27079, 2024.

[15] Y. Chuai, L. Zhang, S. Xie, R. Tian, and Z. Shan, "Hierarchical identity-based encryption based on sm9," in *International Conference on Data Security and Privacy Protection*. Springer, 2024, pp. 106–118.

[16] A. Sudarsono, R. W. Sudibyo, I. Winarno, and M. Yuliana, "An efficient authentication system to access electric vehicle data in the cloud based on identity role-based access control," in *2024 International Electronics Symposium (IES)*. IEEE, 2024, pp. 207–214.

[17] R. Schermann, S. Bussa, R. Urian, R. Toegl, and C. Steger, “Paka: Pseudonymous authenticated key agreement without bilinear cryptography,” in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, 2024, pp. 1–10.

[18] A. A. Almazroi, M. A. Alqarni, M. A. Al-Shareeda, M. H. Alkinani, A. A. Almazroey, and T. Gaber, “A bilinear pairing-based anonymous authentication scheme for 5g-assisted vehicular fog computing,” *Arabian Journal for Science and Engineering*, pp. 1–22, 2024.

[19] S. Bussa, R. Schermann, R. Urian, and R. Toegl, “Paka: Pseudonymous authenticated key agreement without bilinear cryptography,” in *Proceedings of the 19th ACM International Conference on Cryptography*, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3664476.3669925

[20] A. Belel and R. Dutta, “Attribute-based inner product functional encryption in key-policy setting from pairing,” in *International Workshop on Security*, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-97-7737-2_6

[21] B. Zhao, C. Yao, X. Zhang, Y. Liu, and Q. Wu, “Blockchain-based secure and efficient ads-b authentication via certificateless signature with packet loss tolerance,” *IEEE Internet of Things Journal*, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10778267/

[22] W. Zarrougui, F. Hamzaoui, M. Khadhraoui, and H. Messaoud, “A time domain unknown input observer for a class of bilinear delayed system,” *WSEAS Transactions on Systems and Control*, vol. 19, pp. 360–367, 2024.

[23] I. Ali, Y. Chen, N. Ullah, M. Afzal, and H. Wen, “Bilinear pairing-based hybrid signcryption for secure heterogeneous vehicular communications,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5974–5989, 2021.

[24] M. M. Hamdi, A. S. Mustafa, H. F. Mahd, M. S. Abood, C. Kumar, and M. A. Al-shareeda, “Performance analysis of qos in manet based on ieee 802.11 b,” in *2020 IEEE international conference for innovation in technology (INOCON)*. IEEE, 2020, pp. 1–5.

[25] M. Roslee, T. WOON, C. Sudhamani, I. D. Irawati, D. Darlis, A. F. Osma, and M. H. Jusoh, “Internet of things: Agriculture precision monitoring system based on low power wide area network,” *WSEAS Transactions on Electronics*, vol. 15, pp. 35–46, 2024.

[26] A. M. M. Israa Basim, Ahmed Fakhfakh, “Cloud access security brokers: An easy-implement method for accessing cloud services securely,” *WSEAS TRANSACTIONS ON INFORMATION SCIENCE AND APPLICATIONS*, vol. 22, pp. 189–202, 01 2025.

[27] A. Badii, R. Faulkner, R. Raval, C. Glackin, and G. Chollet, “Accelerated encryption algorithms for secure storage and processing in the cloud,” in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, 2017, pp. 1–6.

[28] H. Zhou, L. Deng, Y. Wu, and S. Zhou, “A pairing-free proxy re-encryption scheme suitable for cloud medical information systems,” *Journal of Information Security and Applications*, vol. 89, p. 103967, 2025.

[29] Y. A. Badamasi, “The working principle of an arduino,” in *2014 11th international conference on electronics, computer and computation (ICECCO)*. IEEE, 2014, pp. 1–4.

[30] S. A. Arduino, “Arduino,” *Arduino LLC*, vol. 372, 2015.

[31] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, and S. Vappangi, “A systematic literature review on prototyping with arduino: Applications, challenges, advantages, and limitations,” *Computer Science Review*, vol. 40, p. 100364, 2021.

[32] K. R. Raghunathan, “History of microcontrollers: First 50 years,” *IEEE Micro*, vol. 41, no. 6, pp. 97–104, 2021.

[33] M. Carminati and G. Scandurra, “Impact and trends in embedding field programmable gate arrays and microcontrollers in scientific instrumentation,” *Review of Scientific Instruments*, vol. 92, no. 9, 2021.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.