Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 13, No. 4, December 2025, pp. 873∼887
ISSN: 2089-3272, DOI: 10.52549/ijeei.v13i4.7366

A Message Transmission Scheme for IoT Inspired by The Shamir Scheme and Based on The Hidden Number Problem (HNP)

Mohamed El Ouafi[1], Abderrahim Aslimani[2], Kaoutar Lamrini Uahabi [1], and Abderrahim Zannou [3]
[1]Department of Mathematics, Pluridisciplinary Faculty, Mohammed First University, Laboratory of Applied Mathematics and Information Systems, Selouane, Nador, Morocco
[2]Department of Mathematics, Pluridisciplinary Faculty, Mohammed First University, Laboratory of Applied Mathematics of the Oriental, BP 300-62700, Selouane, Nador, Morocco
[3]ERCI2A, FSTH, Abdelmalek Essaadi University, Tetouan, Morocco

Article Info

Article history:

Received Oct 13, 2025
Revised Nov 2, 2025
Accepted Nov 15, 2025

Keywords:

lightweight cryptography
Internet of Things (IoT)
Hidden Number Problem
secure transmission
keyless protocol

ABSTRACT

The Internet of Things (IoT) requires cryptographic mechanisms that are both lightweight and resistant to emerging attacks. Classical public-key protocols such as RSA or ElGamal, as well as many post-quantum lattice-based schemes, consume too many resources for devices with limited memory, processing power, and energy. In this study, we propose a three-pass message transmission protocol that avoids any prior key exchange. Inspired by Shamir's keyless scheme and relying on the hardness of the Hidden Number Problem (HNP) and the Decisional Diffie–Hellman (DDH) assumption, the protocol operates in a finite field with safe primes and refreshes random masks at each execution, providing strong resistance to brute-force and small-subgroup attacks. We formally prove IND-CPA security and implement the HNP 3-pass scheme, showing that each pass executes in 1.4–1.8 ms on a workstation, with 132-byte public keys and 192–256-byte secret keys. Estimated energy consumption per iteration is 101.562 mJ. Comparative simulations on a workstation and embedded platforms (Arduino Uno and Raspberry Pi) against RSA-512, ECC (secp256r1/secp521r1), and post-quantum Kyber-512 show that our scheme achieves execution times comparable to ECC and Shamir's 3-pass protocol, is significantly faster than RSA, and consumes less energy than Kyber-512. This combination of low latency, moderate key sizes, and energy efficiency highlights the practicality of the HNP 3-pass protocol for resource-constrained IoT environments.

Copyright © 2025 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Mohamed El Ouafi
Multidisciplinary Faculty of Nador, Mohammed First University ,Oujda, Morocco
Nador, Morocco
Email: mohamed.elouafi10@ump.ac.ma

1. INTRODUCTION

The large-scale growth of the Internet of Things in areas such as healthcare, industrial and civil smart cities, transportation, and many other fields makes the protection of information, as well as its continuous transmission and reception, extremely important. This becomes even more critical

because of IoT resource limitations, such as low energy, weak computing power, and sometimes very small memory size [1, 2]. Unencrypted communications, or encrypted ones using weak protocols and algorithms, can play a major role in privacy violations and threats to public safety [3, 4]. In fact, adapting and simplifying traditional cryptographic protocols to fit IoT constraints usually reduces their robustness. Saving computing resources, memory, and energy often comes at the expense of strong encryption and overall security.

To ensure secure and efficient communication between IoT components, cryptographic solutions must use minimal computation and storage, while keeping a reasonable security level and low energy usage. This is key for efficient and sustainable deployment in constrained IoT environments .Therefore, using traditional protocols such as RSA and ElGamal [5, 6] or symmetric algorithms like AES [7] on IoT devices, which often rely on limited microcontrollers, is not the best choice. These protocols face issues with key sizes and storage requirements, and they require heavy computations, which leads to high energy consumption and slow communication [3, 4]. However, reducing key sizes and simplifying computations in classical protocols to fit IoT makes them weaker and more vulnerable to attacks. Moreover, the possibility of stable, commercial quantum computers is becoming more realistic, especially after Microsoft announced its Majorana chip [8]. This chip is presented as having quantum computing properties, which makes classical cryptography ineffective against powerful quantum algorithms such as Shor's [9] and Grover's [10], which can solve the discrete logarithm problem (DLP) and integer factorization in polynomial time.

Since this quantum challenge has been a known issue for a long time, many post-quantum cryptographic proposals have already been made, and some were selected by NIST [11]. Examples include CRYSTALS-Kyber [12], LightSaber [12], NTRUPrime [13], and BIKE [14]. Most of these are based on the Learning With Errors (LWE) problem [15–17], with variants such as Ring-LWE or Module-LWE. Others rely on error-correcting codes [18, 19], or isogenies on supersingular elliptic curves [20]. Unfortunately, most of these schemes either require large secret and public keys, or lack flexibility for all IoT use cases. This motivates the design of lightweight and flexible schemes that do not need permanent key storage, while still providing good security.

In cases where authentication is not necessary for encrypted communication, or when a secondary authentication system can be added, encryption schemes like *Shamir's*, the *Three-Pass Protocol* [21, 22], or the *Omura-Massey* cryptosystem [23, 24] are good options, since they do not require prior key exchange.These methods use passes [25] to exchange a message between two parties.This three-pass transmission method is even used in secure image exchange: for example, in [26], the authors applied *multi-parameter fractional Fourier transforms*. According to Shamir's keyless protocol, if we rely on the difficulty of the discrete logarithm problem, Alice can send a message to Bob as follows:

pass 1: Alice chooses a random element $r \in \mathbb{F}_p^*$, then computes m^r and sends it to Bob.

pass 2: Bob also chooses a random element $s \in \mathbb{F}_p^*$, computes $(m^r)^s = m^{rs}$, and sends it back to Alice.

pass 3: Alice computes

$$(m^{rs})^{r^{-1}} = m^s \tag{1}$$

and sends it to Bob.

Finally, Bob computes

$$(m^s)^{s^{-1}} = m \tag{2}$$

and recovers the original message m.

The protocol's security relies on the hardness of retrieving m from m^r, m^{rs}, or m^s without knowing the secret exponents r and s.

In this work, we take inspiration from Shamir's scheme and propose a new approach. Our scheme is based on the *Hidden Number Problem* [27] and uses random elements from $\mathbb{F}_p$, refreshed in each run, to ensure strong resistance against brute-force attacks. The Hidden Number Problem is defined as follows: in a cyclic group G with generator g, given g and $\alpha \cdot g^x$ for $x, \alpha \in G$, find x.

In this paper, related works is discussed in Section 2. Then, the proposed scheme and its details are described in Section 3. A security proof is given in Section 4, and we conclude the paper and present the future work in Section 5.

2. RELATED WORKS

This section reviews the main asymmetric message encryption schemes, starting with classical protocols, followed by post-quantum schemes. For each family, we describe the basic principles, advantages and limitations, and provide illustrative comparison tables.

2.1. Classical asymmetric encryption schemes

Classical schemes rely on the hardness of well-studied arithmetic problems, such as integer factorization or the discrete logarithm problem. The most common ones are: RSA, ElGamal, and elliptic curve schemes (ECC).

The RSA scheme [28] is based on the difficulty of factoring a large integer $N = pq$. The public key is (N, e), while the private key is $d = e^{-1} \bmod \varphi(N)$. Encryption is defined as:

$$\mathrm{ct} = m^e \bmod N, \tag{3}$$

and decryption as:

$$m = \mathrm{ct}^d \bmod N. \tag{4}$$

RSA-2048 [29] is still widely used, providing a security level considered equivalent to 112 bits.

The ElGamal encryption scheme [6] uses the multiplicative group of a finite field. The public key is $(p, g, h = g^x)$ and the private key is x. Encryption of a message m is done as follows:

$$r\mathbb{Z}_{p-1}, \quad (c_1, c_2) = (g^r, m \cdot h^r) \bmod p. \tag{5}$$

Decryption uses $m = c_2/c_1^x \bmod p$. ElGamal is probabilistic, providing IND-CPA semantic security.

Elliptic curve schemes [30,31] reduce key sizes while providing the same level of security. For example, ECC-256 offers security comparable to RSA-3072.

Table 1 summarizes the main differences between classical encryption schemes. It highlights that elliptic curve cryptography provides security equivalent to RSA and ElGamal while using shorter keys and offering greater efficiency.

Table 1. Comparison of classical encryption schemes

Scheme	Public Key Size	Private Key Size	Ciphertext Size
RSA-2048	2048 bits	2048 bits	2048 bits
ElGamal-2048	2048 bits	2048 bits	4096 bits
ECC-256	512 bits	256 bits	512 bits

2.2. Post-quantum asymmetric encryption schemes

With the advent of quantum computers, classical schemes are under threat. Post-quantum schemes (PQC) rely on problems that are resistant to quantum attacks: lattices, error-correcting codes, multivariate systems, etc. The NIST has recently standardized several KEM.

Lattice-based schemes, especially Module-LWE [32], Ring-LWE [15] and LWR [33], provide a good trade-off between performance and security.

Classical McEliece schemes [34], based on Goppa codes, provide strong resistance but require very large public keys.

Isogeny-based schemes (e.g. SIKE) have been broken [35], and multivariate schemes [36] are still less common for encryption.

Table 2 summarizes the main characteristics of post-quantum encryption schemes. It shows that lattice-based approaches, such as Kyber or Saber, offer a good balance between efficiency and security, with relatively small key sizes. In contrast, code-based schemes like McEliece provide strong resistance but require very large public keys. This comparison highlights the diversity of post-quantum solutions and the trade-offs between performance and security.

Table 2. Comparison of post-quantum schemes (KEM)

Scheme	Family	Public Key	Ciphertext	Secret (plaintext)	NIST Status
Kyber-512	Lattice (Module-LWE)	800 B	768 B	32 B	Standardized
Kyber-768	Lattice (Module-LWE)	1.2 KB	1.1 KB	32 B	Standardized
Kyber-1024	Lattice (Module-LWE)	1.5 KB	1.5 KB	32 B	Standardized
Classic McEliece	Code-based	260–1024 KB	128 B	32 B	In progress
FrodoKEM-640	Lattice (LWE)	9 KB	9.5 KB	16 B	Finalist
NTRU-HRSS	Lattice (Ring-LWE)	930 B	1138 B	32 B	Under evaluation
Saber (LightSaber)	Lattice (LWR)	672 B	736 B	32 B	Finalist

3. CONTRIBUTION

In this section, we propose a three-pass message transmission scheme, based on the HNP, that allows secure communication without any prior sharing of secret keys.

1 The system chooses a large prime number p such that $p-1=2q$ with q also prime, and all computations are done modulo p.

2 **Procedure**

1: **Alice**: Chooses secret keys $m \in \mathbb{F}_p$, $a \in \mathbb{F}_p$, and $\alpha, \gamma \in \mathbb{F}_p$.

2: **Alice**: Computes $(m^a \cdot \alpha)$ and sends $(m^a \cdot \alpha, \gamma^a)$.

3: **Bob**: Chooses secret keys $b \in \mathbb{F}_p$ and $\beta \in \mathbb{F}_p$.

4: **Bob**: Computes and sends $\left(\beta \cdot \left(m^{ab} \cdot \alpha^b\right) \quad , \quad (\gamma^a \cdot \beta)^{-b^{-1}}\right)$.

5: **Alice**: Computes and sends

$$\left(m^2 \cdot \gamma \cdot (m^b \cdot \alpha^{a^{-1}b}) \cdot \beta^{a^{-1}} \quad , \quad \alpha^{-a^{-1}} \cdot \gamma^{-b^{-1}} \cdot \beta^{-b^{-1}a^{-1}}\right).$$

6: **Bob**: Computes and recovers m as

$$m = \left[\left(m^2 \cdot \gamma \cdot (m^b \cdot \alpha^{a^{-1}b}) \cdot \beta^{a^{-1}}\right)^{b^{-1}} \cdot \alpha^{-a^{-1}} \cdot \gamma^{-b^{-1}} \cdot \beta^{-b^{-1}a^{-1}}\right]^{(2b^{-1}+1)^{-1}}.$$

Algorithm 1: Steps for transmitting the message m

In the algorithms 1 and 2 we work in the finite field $\mathbb{F}_p$, where p is a *safe prime* such that $p = 2q + 1$ with q prime. Such a number p is called a safe prime, and q is called a Sophie Germain prime. Its advantage is to minimize the number of subgroups in $\mathbb{F}_p$, preventing attacks on small subgroups such as the Pohlig-Hellman attack [37].

Initialization (Alice). To send the message m to Bob, Alice:

chooses $a, \alpha, \gamma \in \mathbb{F}_p$ as secrets, where a is **odd** to ensure the existence of the inverse modulo $p-1$;

computes and sends to Bob:

$$A_1 = m^a \cdot \alpha, \quad A_2 = \gamma^a. \tag{6}$$

Response (Bob).

chooses $b, \beta \in \mathbb{F}_p$ as secrets, where b is **odd** to ensure the existence of the inverse modulo $p-1$, then checks whether $2b^{-1}+1$ is invertible modulo $p-1$;

computes and sends to Alice:

$$B_1 = \beta\left(m^{ab}\,\alpha^b\right), \quad B_2 = \left(\gamma^a\,\beta\right)^{-b^{-1}}. \tag{7}$$

Finalization (Alice).

computes and sends to Bob:

$$C_1 = m^2 \cdot \gamma \cdot (B_1)^{a^{-1}} = \gamma\, m^{2b+1}\, \alpha^{a^{-1}b}\, \beta^{a^{-1}}, \quad C_2 = \alpha^{a^{-1}} \cdot (B_2)^{a^{-1}} = \alpha^{-a^{-1}}\, \gamma^{-b^{-1}}\, \beta^{-b^{-1}a^{-1}}. \tag{8}$$

Recovery of m (Bob). Bob applies b^{-1} and simplifies:

$$m^{2b^{-1}+1} \;=\; \left(C_1\right)^{b^{-1}} \;\cdot\; C_2 \pmod p. \tag{9}$$

Therefore:

$$m = \left(m^{2b^{-1}+1}\right)^{(2b^{-1}+1)^{-1}}. \tag{10}$$

1 **Alice** **Bob**

1: $\alpha, a, \gamma \xleftarrow{U} \mathbb{F}^*{}_p$, $\xrightarrow{\text{send}}$ $\left(A_1, A_2\right)$

2: $\left(B_1, B_2\right)$ $\xleftarrow{\text{send}}$ $\beta, b \xleftarrow{U} \mathbb{F}^*{}_p$

3: Alice computes:

$$\left(C_1 = m^2 \cdot \gamma \cdot (B_1)^{a^{-1}},\ C_2 = \alpha^{a^{-1}} \cdot (B_2)^{a^{-1}}\right) \xrightarrow{\text{send}} (C_1, C_2)$$

4: **Bob recovers** m :

$$m = \left[C_1^{b^{-1}} \cdot C_2\right]^{(2b^{-1}+1)^{-1}}$$

Algorithm 2: Illustration of the different exchanges between Alice and Bob to transmit message m.

4. SECURITY PROOF AND EXPERIMENTAL EVALUATION

4.1. Semantic Security Proof (IND-CPA via Game Sequence)

First, we define the **hard Hidden Number Problem (HNP)**: Given a prime p, and access to an oracle that returns $\langle g, g^x \cdot \alpha\rangle_p$ for a secret x and random inputs α, it is hard to recover x. We note that our variant of the *Hidden Number Problem* is strictly harder than those studied in the cryptographic literature, as there are no constraints on the mask α or the exponent a. In other words, an adversary must recover the exact values α and a from x and $\alpha \cdot x^a$. For every candidate a', there may exist a corresponding α' satisfying the equality, yielding **multiple compatible pairs** (α', a'). Consequently, it is difficult to identify the correct pair. This ambiguity provides our scheme with the flexibility to **reduce the group order** $\mathbb{F}_p$ (and thus accelerate computations) in constrained environments while maintaining a reasonable security level.

We also recall the **Decisional Diffie-Hellman (DDH)** assumption: In a cyclic group of prime order p generated by g, distinguishing the tuple (g^a, g^b, g^{ab}) from (g^a, g^b, g^c) where c is random, is hard.

Under the assumption that the *Hidden Number Problem* is hard and DDH holds in the considered group, our scheme is IND-CPA secure.

We use a standard sequence-of-games argument.

Game 0 (Real IND-CPA attack). The adversary chooses two messages m_0, m_1. The challenger samples $x \leftarrow \{0, 1\}$, executes the protocol with m_x, and sends the transcript:

$$\begin{pmatrix} A_1 = m_x^a\, \alpha, \\ A_2 = \gamma^a, \\ B_1 = \beta \cdot (m_x^{ab}\, \alpha^b), \\ B_2 = (\gamma^a\, \beta)^{-b^{-1}}, \\ C_1 = m_x^2\, \gamma\, B_1^{a^{-1}}, \\ C_2 = \alpha^{-a^{-1}}\, B_2^{a^{-1}} \end{pmatrix} \tag{11}$$

The adversary outputs a guess x'.

Game 1. We replace A_1, B_1 by two independent uniform values $\widetilde{A_1}, \widetilde{B_1} \leftarrow \mathbb{Z}_p^*$. By the DDH and HNP assumptions:

$$|\Pr[\text{Succ}_0] - \Pr[\text{Succ}_1]| \leq \mathsf{Adv}_{\text{DDH}} + \mathsf{Adv}_{\text{HNP}}.$$

Game 2. We also replace C_1, C_2 with random values $\widetilde{C_1}, \widetilde{C_2} \leftarrow \mathbb{Z}_p^*$. At this point, the adversary's view is independent of m_x, so

$$\Pr[\text{Succ}_2] = \tfrac{1}{2}.$$

Combining everything:

$$|\Pr[\text{Succ}_0] - \tfrac{1}{2}| \leq |\Pr[\text{Succ}_0] - \Pr[\text{Succ}_1]| + |\Pr[\text{Succ}_1] - \Pr[\text{Succ}_2]| \leq \mathsf{Adv}_{\text{DDH}} + \mathsf{Adv}_{\text{HNP}}. \quad (12)$$

Thus, the protocol achieves IND-CPA security under DDH and HNP.

4.2. Experimental Evaluation by Simulation

4.2.1. Implementation

We present a Python implementation of the transmission algorithm. The demonstration presents the four functions that make up the protocol.ons.

The protocol can be summarized as follows:

Alice (Initialization): chooses secret values (a, α, γ), computes A_1 and A_2, and sends them to Bob.

Bob (Response): chooses (b, β), performs checks, and computes B_1, B_2, which are sent back to Alice.

Alice (Finalization): computes (C_1, C_2) using her secrets and Bob's response, then sends them to Bob.

Bob (Recovery of m): reconstructs the original message m by combining (C_1, C_2) with his secret information.

To evaluate the performance of the proposed algorithm, we conducted the experiments directly on our workstation, an **HP EliteBook 830 G5** (processor **Intel© Core™ i5-7300U CPU @ 2.60 GHz**, **x64** architecture, operating system **Microsoft Windows 10 Professional**, BIOS HP Q78 Ver. 01.29.01, **UEFI** mode).

The measurements focus on both the **average execution time** and the **estimated energy consumption** for different key sizes. They were obtained using the Python script , which runs the complete protocol multiple times for each configuration and then calculates the averages and standard deviations.

We show in listing1 the execution results (console output) for different key sizes. It can be observed that the execution time sometimes decreases even when the key size increases. This is due to the randomness in the choice of the masks a, b, α, β, and γ, which directly influences the effective computational complexity.

Listing 1. Console output of the script

```
1
2
3
4  Estimating energy per configuration (Windows, software estimation).
5  Using CPU power estimate = 65.0 W.
6  ######## for 150 bits #########
7  Success rate: 100.0%
8  Duration mean = 1.285 ms
9  Energy mean = 0.835 mJ
10
11 ######## for 200 bits #########
12 Success rate: 100.0%
13 Duration mean = 1.577 ms
14 Energy mean = 1.025 mJ
15
```

```
####### for 250 bits #########
Success rate: 100.0%
Duration mean = 2.181 ms
Energy mean = 18.079 mJ

####### for 300 bits #########
Success rate: 100.0%
Duration mean = 3.520 ms
Energy mean = 63.091 mJ
```

The numerical results are summarized in the following table 3 .

Table 3 summarizes the numerical results obtained for different key sizes. It shows that as the key size increases, both the execution time and the energy consumption grow proportionally. This highlights the trade-off between higher security and computational efficiency.

Table 3. Estimated execution time and energy consumption for different key sizes.

Key Size (bits)	Average Time (ms)	Average Energy (mJ)
150	1.285	0.835
200	1.577	1.025
250	2.181	18.079
300	3.520	63.091

The evolution of time and energy as a function of key size is illustrated in Figure 1. The plotted curves show a clear trend of increasing execution time and energy consumption with larger keys. The time curve (in milliseconds) and the energy curve (in millijoules) follow similar growth patterns, highlighting the direct relationship between execution duration and energy usage. Overall, the figure provides a visual confirmation of the scalability of the algorithm and indicates that, for small to medium key sizes, the computational cost remains reasonable

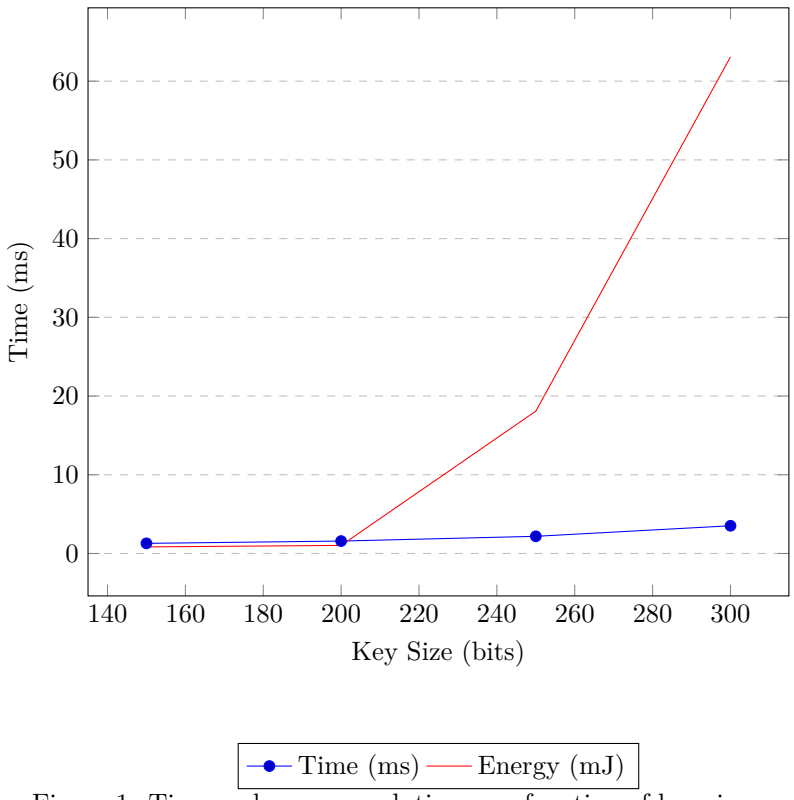

Figure 1. Time and energy evolution as a function of key size.

4.2.2. Comparative simulation

Before presenting the simulations on Arduino and Raspberry Pi, we begin with a comparison on a workstation of several cryptographic schemes. The data comes from the actual execution of the commands using the Python script on our workstation

As shown in Listing 2, the benchmark results provide a comprehensive comparison of Kyber-512, RSA-512, ECC (secp256r1), and our HNP 3-pass scheme. The HNP 3-pass protocol executes each pass for Alice and Bob in approximately 1.4 to 1.8 milliseconds, which is significantly faster than RSA key generation and competitive with ECC operations, while maintaining relatively small public and secret key sizes (PK 132 bytes, SK 192–256 bytes), making it suitable for memory-constrained devices such as IoT nodes. Although Kyber-512 exhibits slightly lower latency for individual operations, its larger key sizes (800 bytes for public keys and 1632 bytes for secret keys) make it less favorable for low-memory environments

Regarding energy consumption, it is estimated at 101.562 mJ, which is lower than that of Kyber-512 and RSA, and competitive with ECC-521. Moreover, the overall speed of RSA and Kyber-512 is slower than that of our scheme, which confirms its practical efficiency.

Listing 2. Console output of Kyber, RSA, ECC, and HNP 3-pass benchmarks

```
--- Benchmark ECC curve secp521r1 ---
KeyGen      : AVG 3.423 ms | MIN 2.532 ms | MAX 8.381 ms | AVG cycles
    10,269,059
Sign        : AVG 3.996 ms | MIN 2.641 ms | MAX 14.177 ms | AVG cycles
    11,989,185
Verify      : AVG 2.741 ms | MIN 2.225 ms | MAX 5.114 ms | AVG cycles
    8,223,104
Sizes (bytes) pub/priv/msg/sig mean: 158/241/32/139
Energy per iteration: 126.953 mJ
=== ML-KEM-512 (Kyber-512) ===
KeyGen     : Avg 2.955 ms | Min 2.438 ms | Max 4.881 ms | Avg cycles
    8,864,910
Encaps     : Avg 4.467 ms | Min 3.567 ms | Max 7.163 ms | Avg cycles
    13,401,036
Decaps     : Avg 6.322 ms | Min 5.109 ms | Max 10.191 ms | Avg cycles
    18,965,436
Sizes      : ek(public) 800 B | dk(secret) 1632 B | ct 768 B
Estimated energy: 253.906 mJ per iteration

--- RSA 512-bit ---
KeyGen      : Avg 24.977 ms | Min 3.421 ms | Max 64.675 ms | Avg cycles
    74,929,784
Encrypt     : Avg 0.020 ms | Min 0.018 ms | Max 0.024 ms | Avg cycles 60,300
Decrypt     : Avg 0.651 ms | Min 0.588 ms | Max 0.753 ms | Avg cycles
    1,952,400
Sizes       : pub/priv/msg/cipher = 71/132/64/64
Energy      : 253.906 mJ

--- ECC secp256r1 ---
KeyGen      : Avg 0.228 ms | Min 0.037 ms | Max 3.502 ms | Avg cycles 683,744
Sign        : Avg 3.287 ms | Min 0.063 ms | Max 63.984 ms | Avg cycles
    9,860,340
Verify      : Avg 0.160 ms | Min 0.102 ms | Max 0.240 ms | Avg cycles 481,484
Sizes       : pub/priv/msg/sig = 91/138/32/71
Energy      : 0.016 mJ

--- HNP 3-pass (our scheme) ---
Alice pass1       : Avg 1.420 ms | Min 1.329 ms | Max 2.275 ms | Avg cycles
    4,260,466
Bob response      : Avg 1.432 ms | Min 1.341 ms | Max 2.431 ms | Avg cycles
```

```
    4,296,829
Alice finalize     : Avg 1.817 ms | Min 1.720 ms | Max 2.711 ms | Avg cycles
    5,450,946
Bob recover        : Avg 1.400 ms | Min 1.320 ms | Max 2.145 ms | Avg cycles
    4,200,902
Sizes (B) PK Alice: 132/130/132
Sizes (B) SK Alice: 256/254/256
Sizes (B) PK Bob  : 132/131/132
Sizes (B) SK Bob  : 192/190/192
Passes 1/2/3 average: 132/132/132
Estimated energy: 101.562 mJ
```

The Table 4 shows the results of our practical experiment. It presents the actual time for different steps of the protocol for Kyber, RSA, ECC, and our scheme. From Figure 2, we can see that our scheme has an advantage in speed. It is close to ECC and faster than RSA and Kyber during key generation, encryption, and decryption. Although RSA is fast in encryption and decryption, slow key generation is a problem in limited-resource environments. Therefore, our scheme provides a good balance between speed and security.

Table 4. Average latencies per operation.

Scheme	KeyGen	Encaps / Pass	Decaps / Recover	Remarks
Kyber-512	2.955	4.467	6.322	—
RSA-512	24.977	0.020	0.651	—
ECC-secp256r1	0.228	3.287	0.160	—
ECC-secp521r1	2.741	3.144	2.169	—
HNP 3-pass	—	1.420 / 1.432 / 1.817	1.400	3 total passes

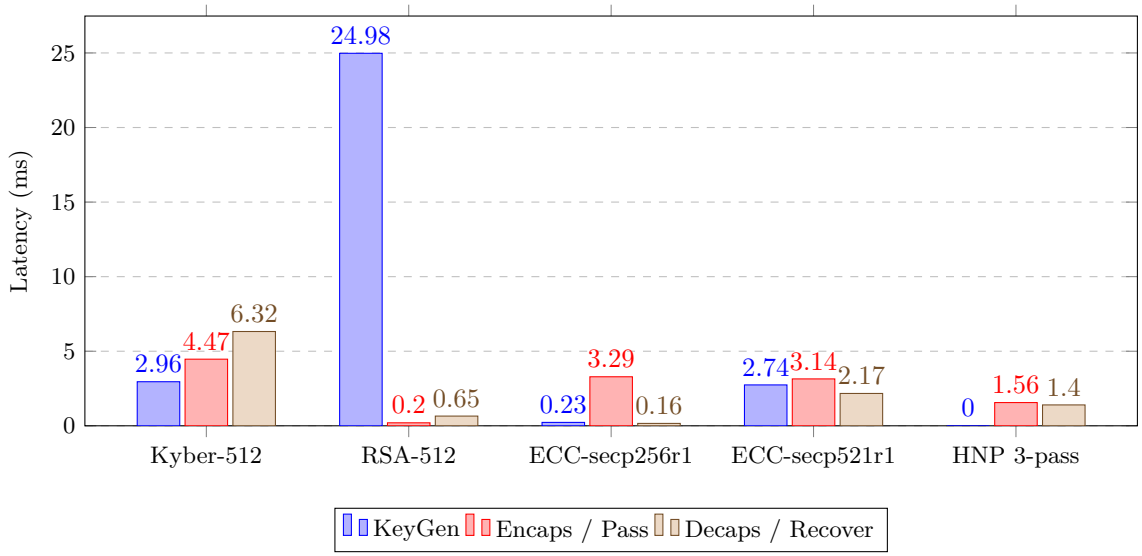

Figure 2. Average latencies per operation for different cryptographic schemes.

Table 5 presents the energy consumption and key sizes for the different schemes. We can see that Kyber-512 uses relatively large public and secret keys and has a high energy consumption (253.906 mJ), which can be problematic in memory-constrained environments. For RSA-512, the keys are more compact, but the energy consumption remains similar, while ECC-secp256r1 and ECC-secp521r1 offer very efficient trade-offs with small key sizes and low energy consumption.

Table 5. Key sizes (bytes) and estimated energy per iteration.

Scheme	PK / ek size	SK / dk size	Energy (mJ)
Kyber-512	800	1632	253.906
RSA-512	71	132	253.906
ECC-secp256r1	91	138	0.016
ECC-secp521r1	158	241	126.953
HNP 3-pass	132	192–256	101.562

As illustrated in Figure 3, the public and private key sizes vary significantly across the cryptographic schemes, with Kyber-512 having the largest keys. Moreover, the energy consumption per iteration also differs, highlighting the efficiency trade-offs between the schemes.

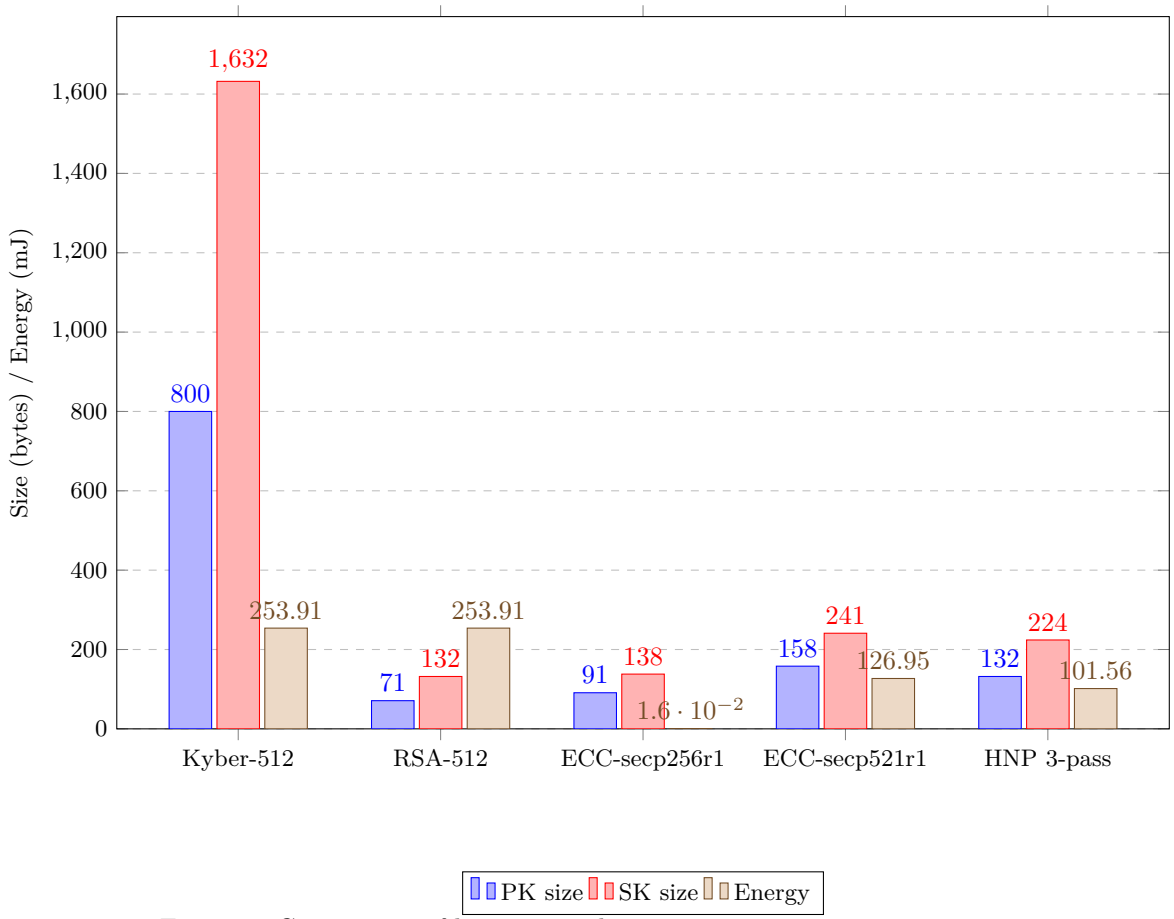

Figure 3. Comparison of key sizes and energy consumption per iteration.

We now present a comparative simulation of three protocols (*Classical RSA*, *Shamir's 3-pass scheme*, and *Our Scheme*) on two embedded platforms: **Arduino Uno** and **Raspberry Pi**. The results show average execution times (in milliseconds) for different key sizes.

Table 6 shows the average values obtained by the simulation. Actual times may differ significantly depending on the model and characteristics of the hardware (for example, Raspberry Pi 3 vs Raspberry Pi 4, Arduino Uno vs Arduino Nano/Pro Mini), processor frequency, system load, software version (interpreter, math libraries), timer resolution, and the availability of entropy for random draws. For robust conclusions, direct and repeated measurements should be performed on the target platform.

Table 6. Simulated execution times (ms) according to key size and platform

Key Size (bits)	Platform	Protocol	Time (ms)
8	Raspberry Pi	Classical RSA	0.0338
8	Arduino Uno	Classical RSA	0.4905
8	Raspberry Pi	Shamir	0.0292
8	Arduino Uno	Shamir	0.4753
8	Raspberry Pi	Our Scheme	0.0335
8	Arduino Uno	Our Scheme	0.4821
12	Raspberry Pi	Classical RSA	0.0596
12	Arduino Uno	Classical RSA	0.8789
12	Raspberry Pi	Shamir	0.0453
12	Arduino Uno	Shamir	0.7104
12	Raspberry Pi	Our Scheme	0.0487
12	Arduino Uno	Our Scheme	0.7476
16	Raspberry Pi	Classical RSA	0.0815
16	Arduino Uno	Classical RSA	1.2219
16	Raspberry Pi	Shamir	0.0667
16	Arduino Uno	Shamir	0.9866
16	Raspberry Pi	Our Scheme	0.0731
16	Arduino Uno	Our Scheme	1.0934
20	Raspberry Pi	Classical RSA	0.0973
20	Arduino Uno	Classical RSA	1.4608
20	Raspberry Pi	Shamir	0.0759
20	Arduino Uno	Shamir	1.1136
20	Raspberry Pi	Our Scheme	0.0841
20	Arduino Uno	Our Scheme	1.2409
24	Raspberry Pi	Classical RSA	0.1239
24	Arduino Uno	Classical RSA	1.8198
24	Raspberry Pi	Shamir	0.0940
24	Arduino Uno	Shamir	1.3745
24	Raspberry Pi	Our Scheme	0.1041
24	Arduino Uno	Our Scheme	1.5290

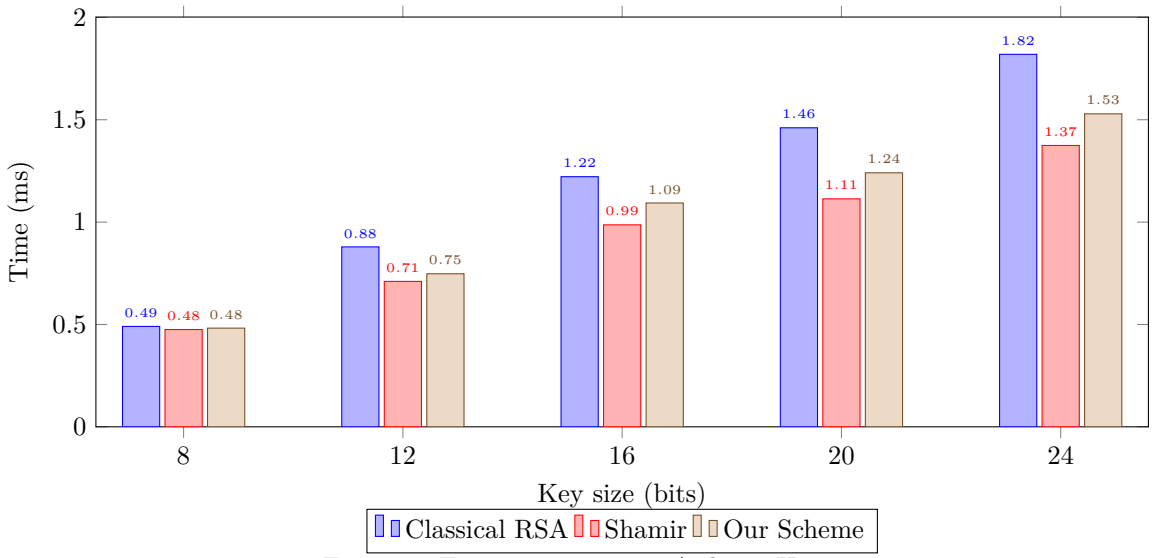

Figure 4. Execution times on Arduino Uno

Figures 4 and 5 illustrate these results graphically for the Arduino Uno and Raspberry Pi platforms, respectively. The curves and bars show that our scheme achieves execution times very close to Shamir's protocol and slightly better than classical RSA on both devices. This confirms that the proposed method provides a balanced trade-off between performance and security.

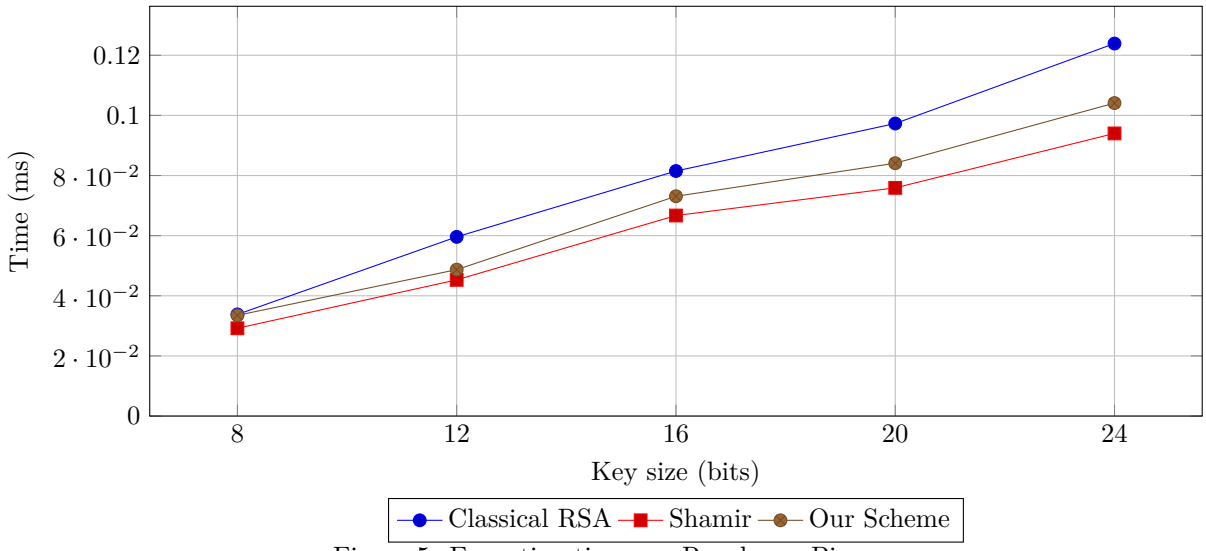

Figure 5. Execution times on Raspberry Pi

Overall, these results demonstrate that our scheme is suitable for low-power or resource-limited environments such as embedded systems, where maintaining both computational efficiency and cryptographic robustness is crucial.

5. CONCLUSION AND FUTURE WORK

In this work, we proposed and simulated a lightweight and secure transmission scheme that can be adapted to resource-constrained platforms such as microcontrollers and embedded boards (Arduino, Raspberry Pi). The simulation results show that our approach achieves execution times comparable to Shamir's protocol and slightly lower than classical RSA, thereby confirming its feasibility and efficiency.

The main advantage of our scheme lies in its combination of speed and security. By using randomly generated keys and masks, it becomes difficult for an attacker to perform brute-force attacks even in groups of small order, since multiple values are admissible without any straightforward way to verify the correct one. Our approach therefore enables secure transmissions while preserving performance, which is particularly important for embedded applications and the Internet of Things (IoT).

As future work, it would be relevant to conduct real experiments on different Raspberry Pi models and Arduino boards in order to obtain experimental performance measurements rather than simulated ones. Furthermore, a natural extension of this work would be to analyze the robustness of the scheme against other types of attacks (side-channel attacks, power analysis, etc.) and to explore its integration into concrete IoT applications requiring a balance between efficiency and security.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013, publisher: Elsevier.

[2] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128613000054

[3] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer, 2004.

[4] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Algorithmic Number Theory*, J. P. Buhler, Ed. Berlin, Heidelberg: Springer, 1998, pp. 267–288.

[5] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978. [Online]. Available: https://dl.acm.org/doi/10.1145/359340.359342

[6] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1057074

[7] W. Stallings, "The advanced encryption standard," *Cryptologia*, vol. 26, no. 3, pp. 165–188, 2002, publisher: Taylor & Francis.

[8] D. Aasen, M. Aghaee, Alam *et al.*, "Roadmap to fault tolerant quantum computation using topological qubit arrays," Jul. 2025, arXiv:2502.12252 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2502.12252

[9] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997. [Online]. Available: https://doi.org/10.1137/S0097539795293172

[10] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[11] "NIST Releases First 3 Finalized Post-Quantum Encryption Standards," *NIST*, Aug. 2024, last Modified: 2025-08-29T11:34-04:00. [Online]. Available: https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards

[12] R. Avanzi, "CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation," Jan. 2017. [Online]. Available: https://www.academia.edu/72492383/CRYSTALS_Kyber_Algorithm_Specifications_And_Supporting_Documentation

[13] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal, "NTRU Prime: Reducing Attack Surface at Low Cost," in *Selected Areas in Cryptography – SAC 2017*, C. Adams and J. Camenisch, Eds. Cham: Springer International Publishing, 2018, pp. 235–260.

[14] N. Aragon, P. S. L. M. Barreto, S. Bettaieb, and others, "BIKE: Bit Flipping Key Encapsulation," 2023. [Online]. Available: https://bikesuite.org/

[15] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Journal of the ACM (JACM)*, vol. 60, no. 6, pp. 1–35, 2013, publisher: ACM New York, NY, USA.

[16] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 34:1–34:40, Sep. 2009. [Online]. Available: https://doi.org/10.1145/1568318.1568324

[17] M. R. Albrecht and A. Deo, "Large Modulus Ring-LWE $\geq$ Module-LWE," 2017, publication info: A major revision of an IACR publication in ASIACRYPT 2017. [Online]. Available: https://eprint.iacr.org/2017/612

[18] N. Sendrier, "McEliece Public Key Cryptosystem," in *Encyclopedia of Cryptography and Security*. Springer, Boston, MA, 2011, pp. 767–768. [Online]. Available: https://link.springer.com/rwe/10.1007/978-1-4419-5906-5_384

[19] C. Aguilar Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, and G. Zémor, "Hamming Quasi-Cyclic (HQC)," Nov. 2017, pages: https://pqc-hqc.org/. [Online]. Available: https://hal.science/hal-01946880

[20] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4*. Springer, 2011, pp. 19–34.

[21] D. Boneh and R. Venkatesan, "Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and Related Schemes," in *Advances in Cryptology — CRYPTO '96*, N. Koblitz, Ed. Berlin, Heidelberg: Springer, 1996, pp. 129–142.

[22] D. M. Nguyen and S. Kim, "A quantum three pass protocol with phase estimation for many bits transfer," in *2019 International Conference on Advanced Technologies for Communications (ATC)*. IEEE, 2019, pp. 129–132.

[23] A. J. Menezes, P. C. v. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton: CRC Press, Dec. 2018.

[24] J. L. Massey and J. K. Omura, "Method and apparatus for maintaining the privacy of digital

messages conveyed by public transmission," Jan. 1986.
[25] E. Faure, A. Shcherba, A. Lavdanskyi, M. Makhynko, and M. Khizirova, "Three-pass protocol on permutations: Implementation example and security." *CPITS*, pp. 110–124, 2024.
[26] J. Lang, "A no-key-exchange secure image sharing scheme based on Shamir's three-pass cryptography protocol and the multiple-parameter fractional Fourier transform," *Optics Express*, vol. 20, no. 3, pp. 2386–2398, 2012, publisher: Optical Society of America.
[27] D. Boneh and R. Venkatesan, "Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and Related Schemes," in *Advances in Cryptology — CRYPTO '96*, N. Koblitz, Ed. Berlin, Heidelberg: Springer, 1996, pp. 129–142.
[28] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978. [Online]. Available: https://dl.acm.org/doi/10.1145/359340.359342
[29] M. A. Sadikin and R. W. Wardhani, "Implementation of RSA 2048-bit and AES 256-bit with digital signature for secure electronic health record application," in *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. IEEE, 2016, pp. 387–392.
[30] S. R. Singh, A. K. Khan, and T. S. Singh, "A critical review on elliptic curve cryptography," in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*. IEEE, 2016, pp. 13–18.
[31] O. İşler, "Implementation and performance evaluation of elliptic curve cryptography over SECP256R1 on STM32 microprocessor," Cryptology ePrint Archive, Paper 2024/1121, 2024. [Online]. Available: https://eprint.iacr.org/2024/1121
[32] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, vol. 75, no. 3, pp. 565–599, 2015, publisher: Springer.
[33] J.-P. D'Anvers, A. Karmakar, S. Sinha Roy, and F. Vercauteren, "Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM," in *International Conference on Cryptology in Africa*. Springer, 2018, pp. 282–305.
[34] R. Overbeck and N. Sendrier, "Code-based cryptography," in *Post-Quantum Cryptography*, D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds. Berlin, Heidelberg: Springer, 2009, pp. 95–145. [Online]. Available: https://doi.org/10.1007/978-3-540-88702-7_4
[35] R. Elkhatib, B. Koziel, and R. Azarderakhsh, "Faster Isogenies for Post-quantum Cryptography: SIKE," in *Topics in Cryptology – CT-RSA 2022*, S. D. Galbraith, Ed. Cham: Springer International Publishing, 2022, pp. 49–72.
[36] T. Yasuda, X. Dahan, Y.-J. Huang, T. Takagi, and K. Sakurai, "A multivariate quadratic challenge toward post-quantum generation cryptography," *ACM Commun. Comput. Algebra*, vol. 49, no. 3, pp. 105–107, Nov. 2015. [Online]. Available: https://doi.org/10.1145/2850449.2850462
[37] R. Rahim, "APPLIED POHLIG-HELLMAN ALGORITHM IN THREE-PASS PROTOCOL COMMUNICATION," *Journal of Applied Engineering Science*, vol. 16, no. 3, Sep. 2018. [Online]. Available: https://aseestant.ceon.rs/index.php/jaes/article/view/16557

ACKNOWLEDGEMENT

The author would like to express sincere gratitude to his teammates for their valuable collaboration and contributions throughout this work.

Mohamed El Ouafi
is a researcher at the *Multidisciplinary Faculty of Nador*, *Mohammed First University*, Morocco. He received his Master's degree in *Modeling and Scientific Computing* from the same university. His research interests include cryptography, information security, embedded systems, and the Internet of Things (IoT).
He can be contacted at email: *mohamed.elouafi10@ump.ac.ma*.

Abderrahim Aslimani
is a Professor in the *Department of Mathematics*, *Multidisciplinary Faculty of Nador*, *Mohammed I University*, Morocco. His research interests include stochastic analysis, probability theory, applied mathematics, partial differential equations, functional analysis, and measure theory.
He can be contacted at email: *a.slimani@ump.ac.ma*.

Kaoutar Lamrini Uahabi
is a Professor in the *Multidisciplinary Faculty of Nador*, *Mohammed First University*, Morocco. She is a member of the *Laboratory of Applied Mathematics and Information Systems,*. Her research interests include nonlinear analysis, fractal geometry, and their applications in medical imaging and complex systems. She is also interested in chaotic dynamical systems and the mathematical modeling of random phenomena.
She can be contacted at email: *ka.lamrini@ump.ac.ma*.

Abderrahim Zannou
is a Professor at the *Faculty of Sciences and Techniques of Al Hoceima*, *Abdelmalek Essaadi University*, Morocco. His research interests include the Internet of Things (IoT), and blockchain technologies.
He can be contacted at email: *a.zannou@uae.ac.ma*.