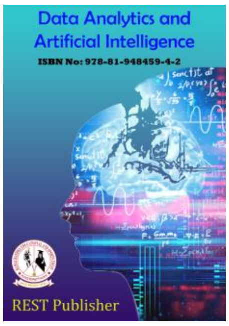
Data Analytics and Artificial Intelligence

Vol: 3(2), 2023

REST Publisher; ISBN: 978-81-948459-4-2

Website: http://restpublisher.com/book-series/daai/

DOI: https://doi.org/10.46632/daai/3/2/30


Data Analytics and Artificial Intelligence
ISBN No: 978-81-948459-4-2
REST Publisher

Energy Consumption Analysis in Post Quantum Cryptography Using Multivariate Signature Algorithms

***Priyavani, N.Kowsalya**

Sri Vijay Vidyalaya College of Arts and Science, Dharmapuri, Tamilnadu, India.

*Corresponding Author Email: pspriyasuresh@gmail.com

Abstract. *Classical cryptographic schemes in use today are based on the difficulty of certain number theoretic problems. Security is guaranteed by the fact that the computational work required to break the core mechanisms of these schemes on a conventional computer is infeasible; however, the difficulty of these problems would not withstand the computational power of a large-scale quantum computer. To this end, the post-quantum cryptography (PQC) standardization process initiated by the National Institute of Standards and Technology (NIST) is well underway. In this paper, the energy consumption of PQC measurements are categorized based on their proposed cryptographic functionality. The results are used in order to identify the most energy-efficient schemes.*

Keywords: *Post Quantum Cryptography, Digital Signature, Rainbow, GeMSS, MQDSS, Multivariate cryptography*

1. INTRODUCTION

Mobile Ad hoc Network (MANET for short) is a kind of communication network which is different from the traditional wireless network due to its characteristic of noncentral administration. It is a self-configurable and autonomous network, and consists of several independent nodes. Those nodes participate with each other to share the responsibilities of network. It can achieve the communications among the nodes, and doesn't need any fixed network device or the support of the Centralized management [1]. Today's world revolves around communication. Modern society depends on Internet as a fundamental building block for any interaction between two parties. There comes a need to protect and maintain the privacy of data being transmitted. Cryptography is a field devoted to the sole purpose of data security, where countless researchers work to ensure that the privacy and integrity of data are maintained through the implementation of various security algorithms. Post Quantum Cryptography is the study of new cryptosystems which cannot be cracked by both quantum and classical computers. The cryptosystems are divided into several families based on the underlying problem upon which the security is established. These underlying problems are believed to be unsolvable by both classical and quantum computers. The major families are lattice-based cryptography, isogeny-based cryptography, non-commutative cryptography, code-based cryptography, hash-based digital signatures, and multivariate cryptography [2].[3]. PQC is the building of cryptosystems that can secure both classical computers and quantum computers should incase an intruder possess it. NIST initiated a process for standardizing post-quantum algorithms. Post Quantum Cryptography is an answer to a threat coming from a full-scale quantum computer able to execute Shor's algorithm. With this algorithm implemented on a quantum computer, currently used public key schemes, such as RSA and elliptic curve cryptosystems, are no longer secure. The U.S. NIST made a step toward mitigating the risk of quantum attacks by announcing the PQC standardization process for new public key algorithms. In March 2019 NIST published a list of candidates qualified to the second round of the standardization process. The cryptosystems are designed for tasks of information exchange and digital signatures. In the case of digital signatures preliminary analysis indicates some advantages of algorithms based on quadratic public rules of Multivariate Cryptography. These systems provide the smallest sizes of the used digital signatures.

2. MULTIVARIATE-BASED CRYPTOSYSTEM

Multivariate-based signature schemes are known as an un balanced Oil-Vinegar (UOV)system which is the process of hiding quadratic equations in n unknowns or Oil and v = n unknowns called "vinegar" in a finite field k [4], and it is based on solving quadratic equations over finite fields, making it an NP-hard problem. The security of the signature scheme is based on the number of variables and the field size, which leads to large key sizes[5]. This family is regarded as one of the key public key cryptography (PKC) [6][7] families capable of withstanding even the most powerful quantum computers in the future. The public is the set of quadratic polynomials:

$$P = (p1(w1,...,wn),..., p_m(w1,...,w_n))$$
$$z_k = p_k(w) := \sum_i p_{ik} w_i + \sum_i Q_{ik} w_i^{21} + \sum_{i>j} R_{ijk} w_i w_j$$

At any given value, the evaluation of these polynomials corresponds to either the encryption or verification procedure.

3. MULTIVARIATE ALGORITHMS

While a number of multivariate encryption schemes have been proposed to NIST's post-quantum standardization program, multivariate cryptography has historically been more successful in signature schemes. Out of the 19 submitted signature schemes, multivariate algorithms take up the biggest part with several multivariate signature schemes.[8].In this paper, the multivariate quadratic (MQ) signature schemes are introduced. The general structure of a MQ-signature (multivariate quadratic) scheme over 𝔽q is as follows. Let us define a system $\mathcal{P}$=(P(1),…, P(m)) of multivariate quadratic polynomials of m equations and n variables by

$$p^{(k)}(x1,...,x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} p_i^{(k)} x_i + p_0^{(k)}$$

For k=1,…, and $p_{ij}^{(k)}$, $p_i^{(k)}$, $p_0^{(k)} \in R$𝔽q .

The main idea for the construction of MQ-signature scheme is to choose a central map $\mathcal{F}$=($\mathcal{F}$(1),…, $\mathcal{F}$(m)):𝔽qn→𝔽qm of multivariate quadratic polynomials, which can be easily inverted. After that two affine or linear invertible maps S:𝔽qm→𝔽qm and T:𝔽qn→𝔽qn are chosen, in order to hide the structure of the central map in a public key.
A public key is the composed quadratic map $\mathcal{P}$=S∘$\mathcal{F}$∘T which is supposedly hardly distinguishable from a random system and therefore difficult to invert. A secret key consists of (S,$\mathcal{F}$, T) which allows to invert $\mathcal{P}$.[4]

Signature Generation

$$\begin{tikzcd}
w \in f^m \arrow[r, "s^{-1}"] & x \in f^m \arrow[r, "f^{-1}"] & y \in f^n \arrow[r, "T^{-1}"] & z \in f^n \arrow[lll, bend left, "P"]
\end{tikzcd}$$

Signature Verification

Figure1: Process of generating and verifying a signature using the MQ-signature scheme.

4. MULTIVARIATE SIGNATURE ALGORITHM

In this section a basic version of the signature algorithm is explained [9]. The algorithm consists of private and public key generating algorithms, signature generating and verifying algorithms. They all are presented in this section along with the signature verification proof. The entries of e represented as integers are bounded in absolute values. A detailed description of the multivariate signature algorithm is in this section below.

Rainbow: The public key of Rainbow is a set of non-linear equations; the verification process involves evaluating the public-key polynomials; and the signing process involves solving a system of linear equations Ding and Schmidt proposed a signature scheme called Rainbow, which is a multilayer variant of Unbalanced Oil and Vinegar [10][11][12].First, we define parameters that determine the layer structure of Rainbow. Let t be the number oflayers in Rainbow. Let v1,……,vt+1 be a sequence of t +1 positive integers such that 0 < v1 < v2 <…… <vt< vt+1. For i = 1; : : : ;

t, the set of indices of the i-th layer in Rainbow is defined by all integers from vi to vi+1, namely Oi = {vi+1;vi+2,….,vi+1 -1;vi+1} The number of indices for the i-th layer, O_i is then vi+1 -vi, and this is denoted by o_i = vi+1 -vi. Note that the smallest integer in O1 is v1+1. Upon defining V1 = {1, 2…., v1}, and for i = 2, 3…., t +1, we have

Vi =V1 $\cup$ O1 $\cup$ O2 $\cup$ ….. $\cup$ Oi-1 = {1,2,….,v_i}

The number of elements in Vi is exactly vi for i = 1, 2…., t +1. The sets O_i and Vi are used for the respective indices of the Oil and Vinegar variables in Rainbow. We define n = vt+1 as the maximum number of variables used in Rainbow.

Next, let K be a finite field of order q. Rainbow consists of t layers of n variables polynomials. For h = 1, 2…, t, the h-th layer of Rainbow contains the following system of o_h multivariate polynomials: Fork $\in O_h$,

$$\sum_{i \in o_h, j \in v_h} \alpha_{i,j}^{(k)} x_i x_j + \sum_{i,j \in v_{h,} i \le j} \beta_{i,j}^{(k)} x_i x_j + \sum_{i \in V_{h+1}} \gamma_i^{(k)} x_i + n^{(k)},$$

$\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, n^{(k)} \in k$ We call the variables xi (i$\in$Oh) and xj (i$\in$Vj) Oil and Vinegar variables, respectively. The central map of Rainbow is constructed according to G = (gv1+1,..,gn) : $k^n \rightarrow k^{n-v1}$. Note that a system of oh equations, gk(b1,…,bv_h ,xv_{h+1},…,xvh+1) = ak (k $\in$ Oh)

Becomes oh linear equations in oh variables for any (av_{h+1},…,av_{h+1}) $\in K^{oh}$ and (b1,…,bv_h)$\in$Kvh . Therefore, once we know the values of the Oil variables in the h-th layer, we can then compute the values of the Vinegar variables in the (h+1) th layer.

```
Algorithm 1 $G^{-1}$(y)
Input: y = (y1; : : : ; ym) $\in Lm$.
Output: x D (x1; : : : ; xn) 2 Ln.
1: Randomly choose q1,…..,qv1 $\in L$ and let i= 1.
2: Substitute (x1,….., xv1)=(q1,…….., qv1 ) into $Gv_{i+1}$,……………,$g_{vi+oi}$ to get a system of linear equations Lx = u in oi variables (If the system is not regular, go back to line 1).
3: Solve the system using Gauss Elimination and obtain a solution ($x_{vi+1}$ …$xvi_{+oi}$) = ($q_{vi+1}$... qvi+oi).
4: Let i=i+ 1. If i<t + 1, go back to line 2.
5: return (x1,…., xn).
```

1. Public Key

For a Rainbow signature scheme, the public key consists of the $n-v1$ polynomial components of $\overline{F}$ and the field structure of *k*.

2. Private Key

The private key consists of the maps *L*1, *L*2 and *F*.

1) Key generation.

The private key consists of two randomly chosen invertible affine linear maps, *L*1 on$k^{n-v1}$and the map $F = (f_{v1+1}(x),...., f_n(x))$. The number of polynomial components of *F*is *m* = *n* – *v*1.The public key is the composed map $\overline{F}(x) = L_1 \circ F \circ L_2$

3. Signing a Document

To sign a document, which is an element Z '= (Z1'. . . Z'n-v1) in $k^{n-v1,}$ one needs tofind a solution of the equation

*L*1 ◦ *F* ◦ *L*2(*x*1, . . .,*xn*) = *F* (*x*1, . . .,*xn*) = *Z'*

We can apply the inverse of *L*1 first, then we have

F ◦ *L*2(*x*1, . . .,*xn*) = $L_1^{-1}Y'=\overline{Y}$

Next, we need to invert *F*. In this case, we need to solve the equation

$$F(x1,...., xn) = \overline{Z} = (\overline{Z_1'},....., \overline{Z_{n-v1}'})$$

We first randomly choose the values of *x*1, . . . , *xv*1 and plug them into the first layer of *o*1 equations given by

$$\overline{F} = (\overline{z}_1',......., \overline{z}_{01}')$$

This produces a set of *o*1 linear equations with *o*1 variables, *xo*1+1, . . . , *xv*2, which we solve to find the values of *xo*1+1, . . . , *xv*2. Then we have all the values of *xii* $\in S2$. Then we plug these values into the second layer of polynomials, which will again produce *o*2 number of linear equations, which then gives us the values of all *xi*, $i \in S3$. We repeat the procedure until we find a solution. If at any time, a set of linear equations does not have a solution, we will start from the beginning again by choosing another set of values for *x*1, . . ., *xv*1. We will continue until we find a solution. We know from [Pat96], that with a very high probability we can expect to succeed if the number of layers is not too large.

Then we apply the inverse of *L*2, which gives us a signature of *Y'* which wewill denote by *X'* = $(x_1^{'},..........,x_n^{'})$

4. Verifying the Signature

To verify a signature, one only needs to check if indeed

$$\overline{F}(X')=z'$$

In order to sign a large document, one can go through the same procedure for Flash as in by applying a hash function first, then sign the hash value of the document.

MQDSS (Multivariate Quadratic Signature Scheme): MQDSS is designed to have are its small key sizes, security of The Multivariate Quadratic Signature Scheme [13], simplicity, and ability to be parallelized. Some of the disadvantages described in the documentation include its large signature size and the problems associated with rewinding of the adversary and adaptively programming the random oracle. We now explicitly construct the functions key generation, Signing and Verification in this algorithm. the start by presenting the parameters of the scheme in general. Parameters in MQDSS is parameterized by a security parameter k ∈ N, and m, n ∈ N such that the security level of the MQ instance MQ (n, m, F2) ≥ k. The latter fix the description length of the equation system F, Flen= m· n·(n+1)/ 2. Cryptographic hash functions H : {0, 1} ∗ → {0, 1} k , H1 : {0, 1} 2k → F31 r , and H2 : {0, 1} 2k → {0, 1} r . – two string commitment functions Com0 : F31 n × F31 n × F31 m → {0, 1} k and Com1 : F31 n × F31 m → {0, 1} k , – pseudo-random generators GSF : {0, 1} k → F31 Flen , GSK : {0, 1} k → F31 n , and Gc : {0, 1} 2k → F31 r·(2n+m) .
Key generation: The MQDSS-q-n key generation algorithm formally samples a MQ relation. Practically, the algorithm is realized as shown in below. Given the security parameter k, we randomly sample a secret key of k bits SK ←R {0, 1} k as well as a seed SF ←R {0, 1} k.
KGen() sk ←R {0, 1} k
SF, Ss, Sρ, Srte ← PRGsk(sk)
F ← XOFF(SF)
s ← PRGs(Ss)
v ← F(s) pk := (SF, v) Return (pk,sk)
In more detail, given the security parameter k, the key generation algorithm KGen() performs the following operations:
- Randomly sample a secret key of k bits sk ←R {0, 1} k .
- Use the secret key sk as input (seed) to PRGsk to derive the following values:
  - ✓ SF, a seed of k bits from which the system parameter F is expanded;
  - ✓ Ss, a seed of k bits from which the secret input to the MQ function is generated;
  - ✓ Sρ, a seed of k bits that is used to generate random values ρ (i) 0 and ρ (i) 1,i∈ {1, . . ., r} needed for the string commitment functions. Note that this seed is not yet needed during key generation, but is required during signing.
  - ✓ Srte, a seed of k bits that is used to sample all vectors r (i) 0, t (i) 0 and e (i) 0,i∈ {1, . . ., r}. Note that this seed is not yet needed during key generation, but is required during signing.
- Expand the seed SF using XOFF to a Flen bits long string, where for q = 2, Flen = n·(n·(n−1) 2 +n) and for q > 2, Flen = n·(n·(n+1) 2 +n) dlog2 qe. Parse the pseudorandom string as an MQ system F ∈MQ (n, n, Fq).
- Use the seed Ss as input to the PRGs to obtain s, a string of length n dlog2 qe bits, that will be used as the secret input to the MQ function;
- Parse s as a vector s ∈ F n q, and evaluate the MQ system F(s) to obtain the vector v ∈ F n q.
- Set pk: = (SF, v) as the public key.
- Return the public/secret key pair (pk, sk). The obtained public key pk is of length k + n dlog2 qe bits, and the secret key sk of length k bits

Signing: The signature algorithm takes as input a message m ∈ {0, 1} ∗ and a secret key sk = (SK, SF). Similarly, as in the key generation, we derive F = GSF (SF). Then, we derive a message-dependent random value R = H (SK k m), where "k" is string concatenation. Using this random value R, we compute the randomized message digest D = H (R k m). The value R must be included in the signature, so that a verifier can derive the same randomized digest. As mentioned in algorithm the core of the derived signature scheme essentially consists of iterations of the IDS. We refer to the number of required iterations to achieve the security level k as r (note that this should not be confused with r0 and r1, which are vectors of elements of F31). Given SK and D, we now compute Gc (SK, D) to produce (r (0,0), . . .,r (0, r),t (0,0), . . . , t(0,r) , e(0,0), . . . , e(0,r)). Using these values, we compute c (0, i) and c (1, i) for each round i, as defined in the IDS. Recall that G(x, y) = F(x + y) − F(x) − F(y), and that Com0 and Com1 are string commitment functions: c(0,i) =Com0(r(0,i) , t(0,i) , e(0,i)) and c(1,i) =Com1(r(1,i) , G(t(0,i) , r(1,i)) +e(0,i))

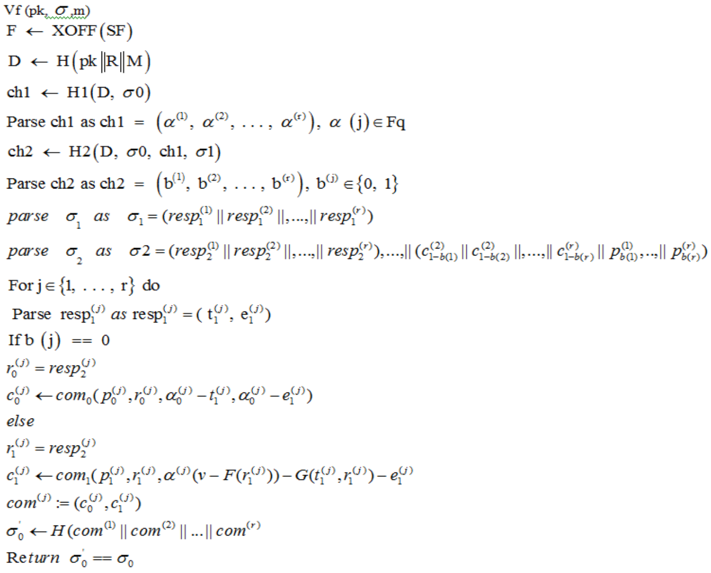
Algorithm 2 Sign (sk,msg),from [14]

```
$s_l, s_s, s_m, s_{rte} \leftarrow PRG_{sk}(sk)$
$L \leftarrow XOF_l(S_l)$
$S \leftarrow PRG_s(s_s)$
$pk := (sl, l(s))$
$R \leftarrow H(s_k \,||\, Msg)$
$D \leftarrow H(m_k \,||\, R \,||\, Msg)$
$m_0^1, \ldots, m_0^r, m_1^{(1)}, \ldots., m_0^{(r)} \leftarrow PRG_p(s_p, D)$
$r_0^{(1)}, \ldots, r_0^r, t_0^{(1)}, \ldots., t_0^{(r)}, e_0^{(1)}, \ldots, e_0^{(r)} \leftarrow PRG_{rte}(s_{rte}, D)$
$forj \in \{1, \ldots, r\}do$
$r_1^{(j)} \leftarrow s - r_0^{(j)}$
$com_0^{(j)} \leftarrow H(m_0^{(j)}, r_0^{(j)}, t_0^{(j)}, e_0^{(j)})$
$com_1^{(j)} \leftarrow H(m_1^{(j)}, r_1^{(j)}, G(t_0^{(j)}, r_1^{(j)}, e_0^{(j)})$
$endfor$
$\sigma_0 \leftarrow H(com_0^{(1)}, com_1^{(1)}, \ldots, com_0^r, \ldots, com_1^{(r)}$
$ch_1 \leftarrow H_1(D, \sigma_0)$
$parse \quad ch1as \quad ch1 = \{\alpha^{(1)}, \ldots, \alpha^{(r)}\}, \alpha^{(j)} \in F_q$
$for \quad j \in \{1, \ldots, r)do$
$t_1^{(j)} \leftarrow \alpha_1^{(j)} r_0^{(j)} - t_1^{(j)}, e_1^{(j)} \leftarrow \alpha^{(j)} F(r_1^{(j)} - e_0^{(j)}$
$rsp_1^j \leftarrow \alpha^{(j)} r_0^{(j)} - t_1^{(j)}, e_1^{(j)} \leftarrow \alpha^{(r)}\}, \alpha^{(j)} \in f_q$
$end \quad for$
$\sigma_1 \leftarrow (rsp_1^{(1)}, \ldots., rsp_1^{(r)})$
$ch_2 \leftarrow H_2(D, \sigma_0, ch1, \sigma 1)$
$parse\ ch_2\ as\ ch_2 = \{b^{(1)}, \ldots., b^{(r)}\}b^{(j)} \in f_2$
$\sigma_2 \leftarrow (r_{b(1)}^{(1)}, \ldots, r_{b(r)}^{(r)}, com_{1-b(1)}^{(1)}, \ldots, com_{1-b(r)}^{(r)}, m_{b(1)}^{(1)}, \ldots, m_{b(r)}^{(r)})$
$return\ \sigma = (R, \sigma_0, \sigma_1, \sigma_2)$
```

Verification: Upon receiving a message M, a signature σ = (R, σ0, σ1, σ2), and a public key pk = (SF, v), the verifier performs the verification routine as listed in Figure 7.3. In more detail, the main goal of the verification process is to reconstruct the missing commitments, and calculate a value σ 0 0 that will be verified against the inputted σ0. The whole procedure is as follows:

- Using the pubic key pk = (SF, v) and the value R from the signature σ, compute the system parameter F ← XOFF(SF) and the randomized message digest D ← H(pk||R||M).
- Since the signature contains σ0, compute the first challenge ch1 as ch1 ← H1(D, σ0) and parse it as ch1 = (α (1), α(2), . . . , α(r)), α (j) ∈Fq
- Next, compute the challenge ch2 ← H2(D, σ0, ch1, σ1), from the two parts σ0, σ1 of the signature and the computed ch1 in the previous step. Parse it as ch2 = (b (1), b(2), . . . , b(r)), b (j) ∈ {0, 1}.

Signature verification



```
Vf (pk, $\sigma$,m)
$F \leftarrow XOFF(SF)$
$D \leftarrow H(pk \| R \| M)$
$ch1 \leftarrow H1(D, \sigma 0)$
Parse ch1 as ch1 $= (\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(r)}), \alpha\ (j) \in Fq$
$ch2 \leftarrow H2(D, \sigma 0, ch1, \sigma 1)$
Parse ch2 as ch2 $= (b^{(1)}, b^{(2)}, \ldots, b^{(r)}), b^{(j)} \in \{0, 1\}$
$parse \quad \sigma_1 \quad as \quad \sigma_1 = (resp_1^{(1)} \,||\, resp_1^{(2)} \,||, \ldots, ||\, resp_1^{(r)})$
$parse \quad \sigma_2 \quad as \quad \sigma 2 = (resp_2^{(1)} \,||\, resp_2^{(2)} \,||, \ldots, ||\, resp_2^{(r)}), \ldots, || (c_{1-b(1)}^{(2)} \,||\, c_{1-b(2)}^{(2)} \,||, \ldots, ||\, c_{1-b(r)}^{(r)} \,||\, p_{b(1)}^{(1)}, \ldots, ||\, p_{b(r)}^{(r)})$
For $j \in \{1, \ldots, r\}$ do
 Parse $resp_1^{(j)}$ as $resp_1^{(j)} = (t_1^{(j)}, e_1^{(j)})$
If $b\ (j)\ ==\ 0$
$r_0^{(j)} = resp_2^{(j)}$
$c_0^{(j)} \leftarrow com_0(p_0^{(j)}, r_0^{(j)}, \alpha_0^{(j)} - t_1^{(j)}, \alpha_0^{(j)} - e_1^{(j)})$
else
$r_1^{(j)} = resp_2^{(j)}$
$c_1^{(j)} \leftarrow com_1(p_1^{(j)}, r_1^{(j)}, \alpha^{(j)}(v - F(r_1^{(j)})) - G(t_1^{(j)}, r_1^{(j)}) - e_1^{(j)}$
$com^{(j)} := (c_0^{(j)}, c_1^{(j)})$
$\sigma_0^{'} \leftarrow H(com^{(1)} \,||\, com^{(2)} \,||\, \ldots ||\, com^{(r)}$
Return $\sigma_0^{'} == \sigma_0$
```

Parse the two signature parts σ1 and σ2 as σ1 = ($resp_1^{(1)} \| resp_1^{(2)}$ ‖ . . . ‖ $resp_1^{(r)}$) and σ2 = (resp (1) 2 ‖ $resp_2^{(1)}$ ‖ . . . ‖ $resp_2^{(2)}$ ‖ $c_{1-b(1)}^{(2)}$ ‖ $c_{1-b(2)}^{(2)}$ ‖ . . . ‖ $c_{1-b(r)}^{(r)}$ ‖ $p_{b(1)}^{(1)}$ ‖ . . . ‖ $p_{b(r)}^{(r)}$) respectively.

Since the verifier knows the values b (j) from the previous step, he knows which of the two parts of the commitments com(j) were included in σ2, and can now proceed to recovering the other, missing part. This is done for all j ∈ {1, . . ., r} as follows:

– Parse resp (j) 1 to obtain ($t_1^{(j)}, e_1^{(j)}$), and

– if b (j) == 0 compute $c_0^{(j)}$ as $c_0^{(j)} \leftarrow com_0(p_0^{(j)}, r_0^{(j)}, \alpha_0^{(j)} - t_1^{(j)}, \alpha_0^{(j)} - e_1^{(j)})$ otherwise compute c (j) 1 as c (j) 1 ← $com_1(p_1^{(j)}, r_1^{(j)}, \alpha^{(j)}(v - F(r_1^{(j)})) - G(t_1^{(j)}, r_1^{(j)}) - e_1^{(j)}$

• Calculate σ 0 ← H(com(1)‖com(2)‖ . . . ‖com(r)) from the obtained commitments com(j)

• Return the truth value of σ 0 0 == σ0. This means that for verification to succeed, σ 0 0 = σ0 should hold.

GeMSS: GeMSS [15] is a multivariate signature scheme, based on a system of polynomial equations over the field F2. The Hidden Field Equations scheme (HFE) [16]. after that HFE [17] generalizes the central map F, substituting the monomials for polynomials. Hidden Field Equations for encryption in which significantly influences the development of multivariate public key cryptography. Let q be a power of a prime (odd or even) and K a degree n extension of Fq. HFE uses the following type of polynomials over K as the central map,

$$H(X) = \sum aij^{Xqi+q\,j} + \sum bi^{Xqi} + c$$

Where the coefficients are randomly chosen in K and the degree of H is bounded by a relatively small number D. We shall call such an F an HFE map (polynomial). The parameter D determines the efficiency and security level of HFE. H(X) = Y can be solved by Berlekamp's algorithm and the complexity is known as O (nD2 logq D + D3).

So, it can be efficient if deg(H) ≤ D is small enough. However, it is first found that D cannot be too small otherwise it can be broken. Public-key. It is given by a set of m quadratic square-free non-linear polynomials in n + v variables over F2. That is, the public key is p =(p1,...,pm) ∈ F2[x1,...,xn+v]^m. It is obtained from the secret-key by taking the first m = n − ∆ polynomials of:

$$(f1((x1,....xm)S),....., fn((x1,.....,xm)S))T,$$

And reducing it modulo the field equations, i.e. modulo hx$_1$ − x1,...,x− xn+vi. We denote these polynomials by p =(p1,...,pm) ∈ F2[x1,...,xn+v]^m.

We summarize the public-key/secret-key generation in Algorithm3. It takes the security parameter λ as input. As discussed in Section 8, the security level of GeMSS will be a function of D, n, v and m. In Section 3 and in Section 9, we specify precisely these parameters. Section 3 presents some parameters in order to achieve a security level λ ∈ {128, 192, 256}. In section 9, we specify some others possible parameters.

Algorithm of Key generation in GeMSS

```
1: procedure GeMSS.KeyGen($1^λ$)
2: Randomly sample (S, T) ∈GLn+v (F2) × GLn (F2).
3: Randomly sample F ∈ F2[X, v1..., vv] with HFEv-shape of degree D.
4: sk ← (F, S, T) ∈ F2[X, v1...,vv] × GLn+v (F2) × GLn (F2)
5: Compute f = (f1..., fn) ∈ F2[x1,...,xn+v]$^n$ such that:
$F\sum_{k=1}^{n}\theta_k x_k, v1,..., v_v = \sum_{k=1}^{n}\theta_k f_k$
6: Compute (p1,...,pn)=( (f1(x1,...,xn+v)S),...,fn (x1,...,xn+v)S))T mod $(x_1^2 - x_1,..., x_{n+v}^2 - x_{n+v}) \in f_2[x_1,...., x_{n+v}]^n$
7:  pk ← p =(p1,...,pm) ∈ F2[x1,...,xn+v]$^m$ .
Computed in step 6
8:   return (sk, pk)
9: end procedure
```

Key generation:

1. Randomly choose S, T and F choosing the coefficients of F uniformly at random in

F2n and the elements of S and T in F2.
2. The private key consists of (S,T, F).
3. Compute p = (p1…, pn)
4. The public key is P = (p1…, pm), the first m components of p.

Signing:
Given a key-pair ((S, T, F), P) and a message digest h, compute the signature performing
the following steps:
1. Set S0 = 0 $\in F_2^m$
2. Repeat for i = 1 to t the following steps:
(a) Get Di the first m bits of h and compute $D_i^{'}$ = Di $\oplus$ Si-1.
(b) Randomly choose (v1…., vv) $\in$ F2^v and r $\in$ F2^{n-m}
(c) Compute Ai = ϕ^{-1}((D_i^1 , r). T^{-1})
(d) Compute a root Z of F -Ai.
(e) Compute (Si,Xi) = (ϕ (Z),v1; : : : ; vv) . S$^{-1} \in F_2^m \times F_2^{n+v-m}$
(f) Compute h = H(h).
3. The signature is z = (St,Xt,…,X1).

Verification: To verify a signature z on a message digest h perform the following steps:
1. Repeat for i = 1 to t
(a) Get Di the first m bits of h.
(b) Compute h = H(h).
2. Repeat for i = t -1 to 0
(a) Compute Si = P(Si+1, Xi+1) $\oplus$ Di+1.
3. Check if S0 = 0.

Performance: Any digital signature scheme based on HFEv- will have a very costly step that is the inversion of the polynomial equation. In GeMSS, this step has estimated complexity of ~O (nD) operations in F2n. Given this, the most expensive part of GeMSS is the multiplication in F2n. The generation of keys requires O(n2 log (D)2+nv log (D)) multiplications and the signature process requires ~O(nD) multiplications. Given this, the most expensive part of GeMSS is the multiplication in F2n. The generation of keys requires O(n2 log (D)2+nv log (D)) multiplications and the signature process requires ~O(nD) multiplications.

5. EXPERIEMENTAL RESULTS

Each algorithm must include an optimized implementation, which is a basic portable C implementation. In order to make a reasonable comparison between each algorithm, these implementations are first profiled to give a fair, baseline comparison of all submitted schemes. This will be referred to as the optimized C implementation. In some cases, the optimized C implementation is a copy of the reference implementation. In this work, a simple C main file is written which calls the three operations required by multivariate signature algorithms were compared Rainbow,MQDSS,GeMSS. The energy and time required for an operation to complete is obtained. A minimum of 1000 iterations of each operation are executed. The energy values are measured with the results reporting the sum of the two values. Based on the expected performance of each of the submissions under consideration, it is anticipated that many of the candidate algorithms will execute much faster than the update rate of the tool (~1ms). In these cases, a loop is used to increase the number of iterations of the algorithm performed. When profiling for digital signature schemes, the results will change based on the message that is being signed. To provide consistent data between tests, a text _le containing 1000 randomly generated 3300-byte messages is created to be used by all digital signature schemes.

Signature Size: In Figure 2 we summarize the dimensions in bytes of the public keys and the correspondingDimensions of the signatures of all the schemes presented in this paper, as well as those of the three multivariate signature algorithms are Rainbow, GeMSS, and MQDSS.

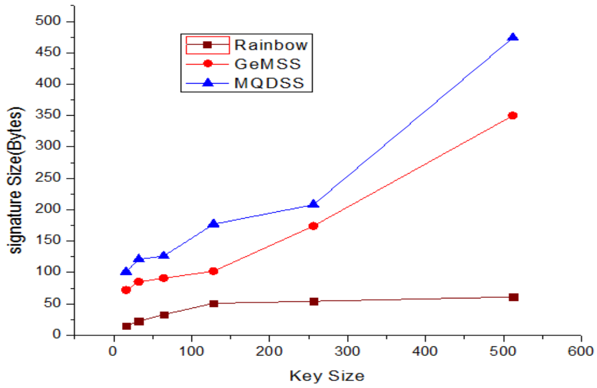


Figure 2: Signature Size

It is interesting to notice that the multivariate schemes have small signatures, but the size of their public keys is the largest among all the schemes. On the other hand, GeMSS and MQDSS have small public keys, but large signatures, while the algorithms based on multivariate have intermediate values in terms of both public keys and signatures. Finally, it is worth to point out that, among all the schemes depicted, the best compromise in terms of dimension is still obtained by the algorithm of Rainbow.

Energy Consumption: When considering the digital signature submissions, it is observed that the multivariate cryptography- based Rainbow's signature scheme is the most energy-efficient. They have the lowest median energy consumption across all security levels; Rainbow, GeMSS, MQDSS figure 1show the level of energy consumption for key generation, signing, and verification, respectively. there are several schemes which are very competitive with the Rainbow, GeMSS, MQDSS algorithms, as will be studied in proceeding sections.

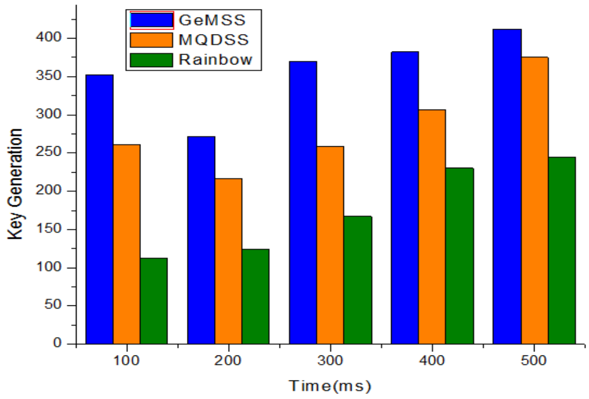


Figure 3: Key Generation

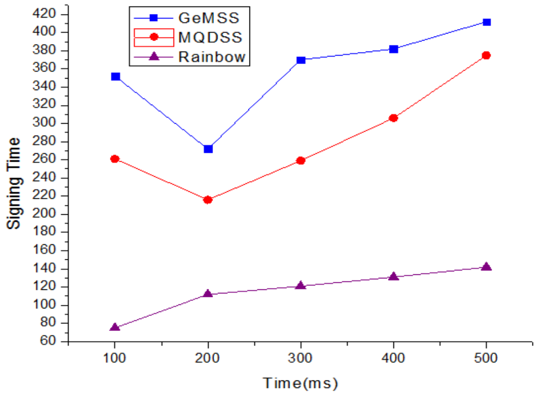


Figure4: Signing Time

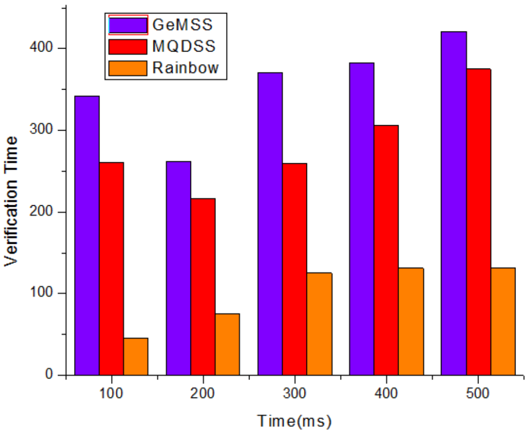


Figure 5: Verification Time

The findings explain the algorithms for Rainbow, GeMSS, MQDSS When the Manet receives a comparable number of times in energy increases dramatically, whereas in the case of rainbow, make span and energy either decreases or fluctuates. This is due to the ability of the algorithm to preserve convergence that was done by having the starting point close to the minimum. Figure 6 displays the behavior of the Rainbow algorithm in contrast to which was before schemes for an average energy consumption scenario with a variable number of times. According to the findings, the Rainbow algorithm significantly reduces network energy usage compared to other digital signature schemes. Only certain numbers of tasks are held accountable for resulting in balanced energy consumption.

6. CONCLUSION

Multivariate algorithms provide secure signature schemes. The NIST's post-quantum standardization program gives a great overview of the field and presents us a variety of options for the future crypto standards, leaving NIST with a difficult task of examining and testing all of the submissions to find the most efficient and secure algorithms. Lastly, Rainbow is used to identify the most energy-consuming submissions to highlight potential areas for multivariate signature algorithm.

REFERENCES

[1]. Zhou L and Haas Z J, "Securing ad hoc networks [J]," IEEE network, special issue on network security, November/December, 1999.

[2]. D. J. Bernstein, \Introduction to Post-Quantum Cryptography," Springer, p. pp. 1{14, 2009.

[3]. L. Chen, S. Jordan, Y.K. Liu, D. Moody, R. Peralta, R. Perlner, D. Smith-Tone. (2016) *NISTIR 8105 Report on Post-Quantum Cryptography*. National Institute of Standards and Technology.

[4]. Kipnis, A., Patarin, J., Goubin,L.: Unbalanced oil and vinegar signature schemes. In: InternationalConferenceontheTheoryandApplicationsofCryptographic Techniques.pp.206{222.Springer(1999)

[5]. Jintai Ding, Lei Hu, XuyunNie, Jianyu Li and John Wagner, \High Order Linearization Equation (HOLE) Attack on Multivariate Public Key Cryptosystems," Public Key Cryptography - PKC 2007: 10th International Conference on Practice and Theory in Public-Key Cryptography. LNCS 4450, 233-248. Springer (2007)

[6]. T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", Advances in Cryptology – Crypto' 84, LNCS Vol. 196, Springer-Verlag, 1985, pp. 10–18; Journal version in IEEE Trans. Information Theory, Vol. 31(4), 1985, pp. 469–472

[7]. Garey, M.R.; Johnson, D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness;W.H. Freeman and Company: New York, NY, USA, 1979.

[8]. Manish Kumar," Post-quantum cryptography Algorithm's standardization and performance analysis", Array 15 (2022) 100242

[9]. R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, Vol. 21(2), 1978, pp. 120–126

[10]. J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in Proc. of the 3rd International Conference on Applied Cryptography and Network Security (ACNS'05), New York, New York, USA, LNCS, vol. 3531. Springer-Verlag, June 2005, pp. 164–175.

[11]. O. Billet, H. Gilbert. Cryptanalysis of Rainbow: SCN 2006, LNCS vol. 4116, pp. 336 - 347. Springer, 2006.

[12]. J. Ding, D. Schmidt: Rainbow, a new multivariable polynomial signature scheme. ACNS 2005, LNCS vol. 3531, pp. 164 - 175. Springer, 2005.

[13]. Jelle Don 1 , Serge Fehr 1 , 2 , and Christian Majenz 1 , 3,"The Measure-and-Reprogram Technique 2.0: Multi-Round Fiat-Shamir and More", https://doi.org/10.1007/978-3-030-56877-1_21

[14]. Chen, M.S., H□ulsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: MQDSS specifications (March 2019), version 2.0, Available at mqdss.org/_les/MQDSS Ver2.Pdf, MQDSS signature scheme

[15]. Casanova, A., Faugere, J.C., Macario-Rat, G., Patarin, J., Perret, L., Ryckeghem, J., "GeMSS: a great multivariate short signature, Submission to the NIST's post-quantumcryptography standardization process", https://www-polsys.lip6.fr/Links/NIST/GeMSS_specification_round2.pdf, (2017).

[16]. L. Bettale, J. C. Faug`ere, and L. Perret. Cryptanalysis of HFE, Multi-HFE and Variants for Odd and Even Characteristic. Des. Codes Cryptography, 69(1):1–52, 2013

[17].Kipnis, A. and Shamir, A.: Cryptanalysis of the HFE public key cryptosystem. In Advances in Cryptology — CRYPTO 1999, volume 1666 of Lecture Notes in Computer Science, pages 19–30. Michael Wiener, ed., Springer (1999). http://www.minrank.org/hfesubreg.ps orhttp://citeseer.nj. nec.com/kipnis99cryptanalysis.html.