# DME: A FULL ENCRYPTION, SIGNATURE AND KEM MULTIVARIATE PUBLIC KEY CRYPTOSYSTEM

IGNACIO LUENGO AND MARTÍN AVENDAÑO

Abstract. DME is a multivariate public key cryptosystem based on the composition of linear and exponential maps that allow the polynomials of the public key to be of a very high degree. A previous version of DME was presented to the NIST call (in the KEM category). This new version of DME adds one or two extra rounds of exponentials to the original two rounds. With this setting the composition gives a deterministic trapdoor one way permutation and allows to use as random padding OAEP for KEM and PSS00 for signature. In the preprint we give the SUPERCOP timing of DME-OAEP and DME-PSS00 for three and four exponentials and compare them with NIST finalists. For NIST security level 5 the size of ciphertext and signature is only 64 bytes.

## 1. Introduction

The main components of the DME cryptosystem are exponential maps $E_A : K^n \to K^n$ associated to matrices $A = (a_{ij}) \in \mathcal{M}_{n\times n}(\mathbb{Z})$, where $K$ is a finite field, whose precise definition is given by the following formula:

$$E_A(x_1, \dots, x_n) = (x_1^{a_{11}} \cdot \ldots \cdot x_n^{a_{1n}}, \dots, x_1^{a_{n1}} \cdot \ldots \cdot x_n^{a_{nn}}). \tag{1}$$

For simplicity, we will use the notation $E_A(x_1, \dots, x_n) = (x_1, \dots, x_n)^A$. The following two facts are extremely useful and also easy to verify:

a) If $A, B \in \mathcal{M}_{n\times n}(\mathbb{Z})$ and $C = B \cdot A$, then $F_C = F_B \circ F_A$.
b) If $\det(A) = \pm 1$, then the inverse matrix $A^{-1}$ has integer entries, $F_A$ is invertible on $(\mathbb{F}_q \setminus \{0\})^n$, and its inverse is given by $F_{A^{-1}}$.

The of monomial maps that are extensively used in Algebraic Geometry and produce birrational maps. In Projective Geometry they are also called Cremona transformations. In [2] these transformations are used to produce a multivariate public key cryptosystem.

If $\det(A) \neq \pm 1$, the monomial map is not birrational. In fact one has:

**Proposition 1.1.** *Let $F_A : K^n \to K^n$ be a monomial map as (1), where $K$ is an algebraically closed field of characteristic zero. Then the monomial map $F_A$ has geometric degree $d := |\det(A)|$ on $(K \setminus \{0\})^n$, that is, for a generic $x \in (K \setminus \{0\})^n$, the fiber $F_A^{-1}(x)$ has $d$ preimages.*

Let $q = p^e$ be a prime power and $\mathbb{F}_q$ denote a finite field of $q$ elements. It is not necessary to consider exponents greater than $q-2$ since $x^{q-1} = 1$ for all $x \in \mathbb{F}_q \setminus \{0\}$. We take $A \in \mathcal{M}_{n\times n}(\mathbb{Z}_{q-1})$ and then we have:

**Theorem 1.2.** *Let $A \in \mathcal{M}_{n\times n}(\mathbb{Z}_{q-1})$ and $G_A : \mathbb{F}_q^n \to \mathbb{F}_q^n$ be the corresponding monomial map. If $\gcd(\det(A), q-1) = 1$, and we set $b := \det(A)^{-1} \in \mathbb{Z}_{q-1}$ and $B := b\mathrm{Adj}(A)$, then $A^{-1} = B \in \mathcal{M}_{n\times n}(\mathbb{Z}_{q-1})$ and $F_A : (\mathbb{F}_q \setminus \{0\})^n \to (\mathbb{F}_q \setminus \{0\})^n$ is bijective with inverse $F_{A^{-1}}$.*

*Proof.* The proof is immediate since, as we mentioned above, we can reduce the exponents modulo $q-1$. By construction, we have $b\det(A) = 1 + \lambda(q-1)$, so $AB = b\det(A)I_n \equiv I_n \pmod{q-1}$ and $\underline{x}^{AB} = \underline{x}^{I_n} = \underline{x}$. □

*Date*: November 14, 2022.

The exponential maps $F_A$ can be used to build a multivariate PKC in the standard way by putting powers of $q$ in the non-zero entries of the matrix $A$. For instance, if each row has 2 non zero entries $q^{a_{ij}}$, then after composition with two linear maps at both ends, one gets a quadratic public key, if we allow 3 non zero entries, we get cubic polynomials, and so on. We made extensive computer tests leading to the conclusion that those systems are not safe against Gröbner basis attack for reasonable key size.

In order to make an scheme stronger against algebraic cryptanalysis we take $q = 2^e$ and allow the non-zero entries of $A$ to be powers of 2 that are not powers of $q$. This choice produces final polynomials with degree up to $q - 1$ in each variable. The kernel of our scheme is a composition of $r$ exponentials with $n$ variables and $n + 1$ linear maps, that we denote by DME-$(r, n, 2^e)$. We have successfully built DME-$(r, n, 2^e)$ schemes with $n = 6, 8$ and $3 \leq r \leq 6$. For simplicity, we take $r = 4$ and $n = 8$ in the following description of the DME.

## 2. Mathematical description of DME-$(4, 8, 2^e)$

The DME-$(4, 8, 2^e)$ cryptosystem works with plain texts and cypher texts in $\mathbb{F}_q^8$ with $q = 2^e$. Let $u^2+au+b \in \mathbb{F}_q[u]$ be an irreducible polynomial, consider the field extension $\mathbb{F}_{q^2} = \mathbb{F}_q[u]/\langle u^2+au+b\rangle$ of degree two over $\mathbb{F}_q$. Let $\phi : \mathbb{F}_q^2 \to \mathbb{F}_{q^2}$ be the bijection defined by $(x, y) \mapsto x + y\bar{u}$ and let $\bar{\phi} : \mathbb{F}_q^8 \to (\mathbb{F}_{q^2})^4$ be the map $(x_1, \ldots, x_8) \mapsto (\phi(x_1, x_2), \phi(x_3, x_4), \phi(x_5, x_6), \phi(x_7, x_8))$. The values of $e, a, b$ are fixed during the setup of the system.

The DME-$(4, 8, 2^e)$ cryptosystem combines 5 linear+affine maps $L_0, \ldots, L_4 : \mathbb{F}_q^8 \to \mathbb{F}_q^8$ with 4 exponential maps $E_1, \ldots, E_4 : (\mathbb{F}_{q^2})^4 \to (\mathbb{F}_{q^2})^4$. More precisely, the encryption map

$$F = \Psi(L_0, \ldots, L_r, E_1, \ldots, E_r) : \mathbb{F}_q^8 \to \mathbb{F}_q^8$$

is given by the composition

$$\begin{aligned}
&\mathbb{F}_q^8 \xrightarrow{L_0} \mathbb{F}_q^8 \xrightarrow{\bar{\phi}} (\mathbb{F}_{q^2})^4 \xrightarrow{E_1} (\mathbb{F}_{q^2})^4 \\
&\xrightarrow{\bar{\phi}^{-1}} \mathbb{F}_q^8 \xrightarrow{L_1} \mathbb{F}_q^8 \xrightarrow{\bar{\phi}} (\mathbb{F}_{q^2})^4 \xrightarrow{E_2} (\mathbb{F}_{q^2})^4 \\
&\xrightarrow{\bar{\phi}^{-1}} \mathbb{F}_q^8 \xrightarrow{L_2} \mathbb{F}_q^8 \xrightarrow{\bar{\phi}} (\mathbb{F}_{q^2})^4 \xrightarrow{E_3} (\mathbb{F}_{q^2})^4 \\
&\xrightarrow{\bar{\phi}^{-1}} \mathbb{F}_q^8 \xrightarrow{L_3} \mathbb{F}_q^8 \xrightarrow{\bar{\phi}} (\mathbb{F}_{q^2})^4 \xrightarrow{E_4} (\mathbb{F}_{q^2})^4 \\
&\xrightarrow{\bar{\phi}^{-1}} \mathbb{F}_q^8 \xrightarrow{L_4} \mathbb{F}_q^8
\end{aligned}$$

of the linear+affine and exponential maps interleaved with the bijections $\bar{\phi}$ and $\bar{\phi}^{-1}$.

Each linear+affine map $L_i$ is made up of four linear maps $L_{i1}, \ldots, L_{i4} : \mathbb{F}_q^2 \to \mathbb{F}_q^2$ and four translation vectors $a_{i1}, \ldots, a_{i4} \in \mathbb{F}_q^2$, so that

$$L_i(x_1, \ldots, x_8) = (L_{i1}(x_1, x_2) + a_{i1}, L_{i2}(x_3, x_4) + a_{i2}, L_{i3}(x_5, x_6) + a_{i3}, L_{i4}(x_7, x_8) + a_{i4}).$$

The matrices of the blocks $L_{i1}, \ldots, L_{i4}$ are $A_{i1}, \ldots, A_{i4} \in \mathbb{F}_q^{2\times 2}$, respectively.

In order to avoid failures of decryption (more technical details about this will be given below) we use translations in only one intermediate step $1 \leq i_0 < 4$ and set $a_{ij} = 0$ for all $i \neq i_0$.

The exponential maps $E_i : (\mathbb{F}_{q^2})^4 \to (\mathbb{F}_{q^2})^4$ are defined by

$$(y_1, y_2, y_3, y_4) \mapsto F_{E_i}(y_1, y_2, y_3, y_4) = (y_1, y_2, y_3, y_4)^{E_i}$$

where $E_i = (\alpha_{i,k})_{1 \le k \le 8}$ is a $4 \times 4$ matrix with coefficients in $[0, q^2 - 1]$. It is not necessary to consider exponents greater than $q^2 - 1$ since $x^{q^2} = x$ for all $x \in \mathbb{F}_{q^2}$.

The linear+affine maps $L_i : \mathbb{F}_q^8 \to \mathbb{F}_q^8$ are invertible if and only if each of the $2 \times 2$ blocks $L_{i1}, L_{i2}, L_{i3}, L_{i4}$ have non-zero determinant. In this case, the inverse of $L_i$ is

$$L_i^{-1}(x_1, \ldots, x_8) = (L_{i1}^{-1}(x_1, x_2) - L_{i1}^{-1} a_{i1}, \ldots, L_{i4}^{-1}(x_7, x_8) - L_{i4}^{-1} a_{i4}),$$

i.e. $L_i^{-1}$ is also a linear+affine map.

The exponential maps $E_i : (\mathbb{F}_{q^2})^4 \to (\mathbb{F}_{q^2})^4$ are not invertible in general. However, their restrictions to the torus $\widehat{E}_i : (\mathbb{F}_{q^2}^*)^4 \to (\mathbb{F}_{q^2}^*)^4$ are invertible if and only if

$$\gcd(\det(E_i), q^2 - 1) = 1.$$

The inverse of $\widehat{E}_i$ is also an exponential map $\widehat{E}_i^{-1} : (\mathbb{F}_{q^2}^*)^4 \to (\mathbb{F}_{q^2}^*)^4$, given by the inverse of the matrix $E_i$ modulo $q^2 - 1$. This matrix has coefficients in $[0, q^2 - 2]$. Using the same matrix, we extend $\widehat{E}_i^{-1}$ to an exponential map $E_i^{-1} : (\mathbb{F}_{q^2})^4 \to (\mathbb{F}_{q^2})^4$.

The private key consists of the coefficients of the linear+affine maps $L_0, \ldots, L_4$ and exponential maps $E_1, \ldots, E_4$. That information is enough to apply all those maps in reverse, that is, to being able to decrypt. The public key is the polynomial representation of the composition of the maps,

$$F(x_1, \ldots, x_8) = (F_{4,1}, F_{4,2}, F_{4,3}, F_{4,4}, F_{4,5}, F_{4,6}, F_{4,7}, F_{4,8})$$

.

## 3. Computing the monomials of $F$

If $\underline{x} = (x_1, \ldots, x_8) \in \mathbb{F}_q^8$ are the initial coordinates, then the composition of all the maps allow us to compute the components of $F(\underline{x})$ as polynomials $F_{4,j} \in \mathbb{F}_q[x_1, \ldots, x_8]$. In order to keep the number of monomials small, we choose the matrices $E_i$ with the following properties:

(1) The entries of $E_i$ are powers of 2.
(2) Each row of $E_i$ has one or two non zero entries.
(3) If we define $d_i = \frac{1}{\det(E_i)} \mod q^2 - 1$, then we have that $d_i$ has a big binary weight for some $1 < i \le 4$.

The inverse map $F^{-1}$ is also composition of 4 exponentials so if the number of monomials of $F^{-1}$ is not very big, one can get the polynomial components of $F^{-1}$ by interpolation, provided enough number of pairs $(\underline{x}, F(\underline{x}))$. To avoid this attack we take some $i$ such that $d_i$ has a big binary weight to ensure that the inverse $E_i^{-1}$ has entries with big binary weight that will produce a big number of monomial of the inverse $F^{-1}$ above a given security level.

It is possible to get the monomials of the $F_i$ without computing the composition of all the maps. It is easy to verify that after exponential $E_i$ plus $\bar{\phi}^{-1}$ the 8 resulting polynomials

$$F_{i,1}, F_{i,2}, F_{i,3}, F_{i,4}, F_{i,5}, F_{i,6}, F_{i,7}, F_{i,8}$$

verify that $F_{i,2k-1}$, $F_{i,2k}$ and $F_{i,2k-1} + \bar{u}.F_{i,2k}$ share the same monomials $M_{ik}$ unless some coefficient vanish and also the same happens after we apply $L_i$.

Let $M = [m_1, \dots m_s]$ a list of monomials and $\alpha$ a power of 2, we define $M^\alpha = [m_1^\alpha, \dots, m_s^\alpha]$. If $M = [m_1, \dots, m_s]$ and $N = [n_1, \dots, n_t]$ are lists of monomials, we define

$$M^\alpha \otimes N^\beta = [m_i^\alpha \otimes n_j^\beta, 1 \le i \le s, 1 \le j \le t],$$

that is, $M^\alpha \otimes N^\beta$ is the Kronecker tensor product of $M^\alpha$ and $N^\beta$ as row matrices.

It is easy to verify that $M_{ij}^\alpha \otimes M_{ik}^\beta$ is the list of monomials of the polynomial

$$(F_{i,2j-1} + \bar{u}.F_{i,2j})^\alpha \cdot (F_{i,2k-1} + \bar{u}.F_{i,2k})^\beta$$

since the exponents $\alpha$ and $\beta$ are powers of 2.

Let $M_{01} = [x_1, x_2], M_{02} = [x_3, x_4], M_{03} = [x_5, x_6], M_{04} = [x_7, x_8]$. We use the following notation for the entries of each matrix $E_i$: we call $\alpha_{i,2k-1}$ the first non zero entry of the row $k$ and $\alpha_{i,2k}$ the second non zero entry. If there is only one non zero entry, we just set $\alpha_{i,2k} = 0$.

We reduce the list of monomials when some of them are repeated. Let us define an operation $Rm(M)$ on a list of monomials $M$ that removes all duplicates, keeping only the first appearance of each monomial in the list and erasing the rest. The following algorithm, called MON, shows how to compute the lists of monomials of the $F_{rj}$.

**Algorithm 3.1** MON, compute the monomials in the public-key polynomials.

**Input:** $(E_1, \dots, E_r)$
**Output:** $(M_{r1}, M_{r2}, M_{r3}, M_{r4})$

**for** $i = 0$ **to** $r - 1$ **do**
  **for** $k = 1$ **to** $4$ **do**
    $M_{(i+1)k} = M_{ik_1}^{\alpha_{i,2k-1}} \otimes M_{ik_2}^{\alpha_{i,2k}}$, where $M_{ik_2} = [1]$ if $\alpha_{i,2k} = 0$
    $M_{(i+1)k} = Rm(M_{(i+1)k})$
    **if** $a_{(i+1)k} \neq 0$ **then**
      append 1 to the list $M_{(i+1)k}$
    **end if**
  **end for**
**end for**

The size of the lists $M_{ri}$ can be up to double exponential on the number of exponentials $r$ for instance if all the rows of the $E_i$ have two non zero entries then $card(M_{ri}) = 2^{2^r}$. We can reduce the size of the list of monomials by imposing some linear condition on the exponents $e_{i,j}$ of $\alpha_{i,j}$ ($\alpha_{i,j} = 2^{e_{i,j}}$), in such a way that some of the monomials become equal and the coefficient of the repeated monomial is a sum of several terms, which will give us a defense against the structural cryptanalysis. In fact, we need to take care of the following:

After the last exponential the final polynomials are obtained by computing $F_{(r-1)k_1}^{\alpha_{i,2k-1}} \cdot F_{(r-1)k_2}^{\alpha_{i,2k}}$. Let $F_{(r-1)k_1}^{\alpha_{i,2k-1}} = \sum B_i m_i$ and $F_{(r-1)k_2}^{\alpha_{i,2k}} = \sum C_j n_j$. Then,

$$F_{(r-1)k_1}^{\alpha_{i,2k-1}} \cdot F_{(r-1)k_2}^{\alpha_{i,2k}} = \left(\sum B_i m_i\right) \cdot \left(\sum C_j n_j\right) = \sum B_i C_j m_i n_j = \sum H_{ij} m_i n_j.$$

Thus, we have $H_{ij} = B_i C_j$, and it is clear now that the coefficients $H_{ij} \in \mathbb{F}_{q^2}$ satisfy $H_{ij}H_{kl} = H_{il}H_{kj}$, which will be called quadratic relations (QR) from now on. Since the coefficients of final polynomials $F_1, \dots F_8$ are obtained applying $\bar{\phi}^{-1}$ and $L_r$, we can use the QR to compute equations for the coefficients of the components of inverse of $L_r^{-1}$. Given that the QR are homogeneous (of degree two), one can solve those equations to find $L_r^{-1}$ and $L_r$ up to a constant.

In order to eliminate the QR among the $H_{ij}$, the strategy is to force many coincidences among the final monomials, that is, if $H_{ij}$ is a sum $= \sum B_k C_l$ it will by more difficult to get the quadratic relations or any polynomial relations among the $H_{ij}$. The implicit equations on the $H_{ij}$ are obtained

by computing the equations of the image of the map $Q = (Q_{ij})$, defined by $H_{ij} = Q_{ij}(B, C) = \sum B_k C_l$, that is by eliminating the $B_1$ and $C_j$ from the system $\langle H_{ij} - \sum B_k C_l \rangle$

$$Q : \mathbb{F}_{q^2}[B_k, C_l] \longrightarrow \mathbb{F}_{q^2}[H_{ij}]$$

For instance, for the second component of example 1 there are no QR, the source has 24 variables and the target 48.

Assume that we are at the step $i$ of the algorithm MON and we are computing the list $M_{(i+1)k}$. We can force a reduction of the monomials only if there are two non zero entries $2^{e_{i,2k-1}}$ and $2^{e_{i,2k}}$ in the corresponding row of the matrix $E_i$, so we'll have to compute $M_{(i+1)k} = M_{ik_1}^{\alpha_{i,2k-1}} \otimes M_{ik_2}^{\alpha_{i,2k}}$. Now, we take a variable that is in both lists with exponent a power of 2, which for simplicity we'll assume it is $x_1$. More precisely, the monomial $x_1^{2^{l_1}} \cdot m_1$, where $l_1 = l_1(e_{j,l} \,:\, 1 \leq j \leq i-1)$ is a linear form and $m_1$ is a monomial in the other variables would appear in $M_{ik_1}$, and $x_1^{2^{l_2}} \cdot m_2$ in the list $M_{ik_2}$. By the method that the lists are constructed ($x_1$ and $x_2$ play exactly the same role), we would also have the monomials $x_2^{2^{l_1}} \cdot m_1$ and $x_2^{2^{l_2}} \cdot m_2$ in the lists $M_{ik_1}$ and $M_{ik_2}$, respectively.

Now, when we compute $M_{ik_1}^{\alpha_{i,2k-1}}$, the exponent of $x_1$ in the first monomial is $2^{l_1+e_{i,2k-1}}$ and in the other list is $2^{l_2+e_{i,2k}}$. We can forze that $2^{l_1+e_{i,2k-1}} = 2^{l_2+e_{i,2k}}$ if we substitute $e_{i,2k}$ by $e_{i,2k-1} + l_1 - l_2$ and then the monomials in both lists became

$$x_1^{2^{l_1+e_{i,2k-1}}} \cdot m_1^{2^{e_{i,2k-1}}},\ x_2^{2^{l_1+e_{i,2k-1}}} \cdot m_1^{2^{e_{i,2k-1}}}$$

in the first list, and

$$x_1^{2^{l_1+e_{i,2k-1}}} \cdot m_2^{2^{e_{i,2k-1}+l_1-l_2}},\ x_2^{2^{l_1+e_{i,2k-1}}} \cdot m_2^{2^{e_{i,2k-1}+l_1-l_2}}$$

in the second.

When the tensor product of both lists is computed, we get that two of the four monomials are equal:

$$\begin{aligned} x_1^{2^{l_1+e_{i,j2k-1}}} \cdot m_1^{2^{e_{i,j2k-1}}} \cdot x_2^{2^{e_{i,j2k-1}+l_1-l_2}} \cdot m_2^{2^{e_{i,j2k-1}+l_1-l_2}} \\ = x_2^{2^{l_1+e_{i,j2k-1}}} \cdot m_1^{2^{e_{i,2k-1j}}} \cdot x_1^{2^{l_1+e_{i,j2k-1}}} \cdot m_2^{2^{e_{i,j2k-1}+l_1-l_2}}. \end{aligned}$$

If there are other variables repeated in both lists that have different exponents after the change $e_{i,2k} = e_{i,2k-1} + l_1 - l_2$, we can repeat the same procedure of imposing a linear condition, but in this case the linear equations involves terms $e_{jk}$ with $j \leq i-1$. In general, each linear condition will produce the reduction of many monomials, but the actual number depends of the structure of the matrices $E_i$ and it is not possible to give a formula for the final number of monomials of $F$. we call this algorithm RED , the input is the set $\{E_i\}$. Next, we present an example of the procedure.

**Example 1:** For this example, we take $q = 2^e$, $n = 6$ and following matrices over $\mathbb{Z}_{q^2-1}$:

$$E_1 = \begin{pmatrix} \alpha_{1,1} & 0 & \alpha_{1,2} \\ \alpha_{1,3} & \alpha_{1,4} & 0 \\ 0 & 0 & \alpha_{1,5} \end{pmatrix}, \quad E_2 = \begin{pmatrix} \alpha_{2,1} & \alpha_{2,2} & 0 \\ 0 & \alpha_{2,3} & \alpha_{2,4} \\ \alpha_{2,5} & 0 & \alpha_{2,6} \end{pmatrix}, \quad E_3 = \begin{pmatrix} \alpha_{3,1} & 0 & \alpha_{3,2} \\ \alpha_{3,3} & \alpha_{3,4} & 0 \\ 0 & \alpha_{3,5} & \alpha_{3,6} \end{pmatrix}.$$

As usual, $\alpha_{i,j} = 2^{e_{i,j}}$ and $e_{i,j} \leq e-1$. If the $e_{i,j}$ are generic, the lists of monomials after the first exponential $(M_{11}, M_{12}, M_{13})$ have size $(2^2, 2^2, 2)$, after the second exponential the lists $(M_{21}, M_{22}, M_{23})$ have size $(2^4, 2^3, 2^3)$, and after the third one the final lists $(M_{31}, M_{32}, M_{33})$ have size $(2^7, 2^7, 2^6)$. We can apply the method in this section and find 7 independent linear conditions on the $e_{i,j}$ as follows: after $E_1$, the lists $(M_{11}, M_{12}, M_{13})$ have size $(2^2, 2^2, 2)$, after $E_2$, we observe that the list $M_{21}$ comes from tensoring $M_{11}$ and $M_{13}$, which have $x_1$ and $x_6$ in common, so the linear condition $e_{2,2} = e_{1,1} + e_{2,1} - e_{1,3}$ reduces the number of monomials to 12. For $M_{21}$ there are no common variables and for $M_{23}$ we get the condition $e_{2,4} = -e_{2,5} + e_{2,6} - e_{1,1} + e_{1,3} + e_{2,3}$, that

gives $(12, 2^3, 6)$ monomials. Finally, after $E_3$, the lists have size $(72, 96, 48)$. For the list $M_{31}$ we get the condition $e_{3,2} = e_{3,1} + e_{2,1} - e_{2,5}$ that reduces the size of $M_{31}$ to 32. For the list $M_{32}$ we get the condition $e_{3,4} = e_{3,3} + e_{1,1} + e_{2,1} - e_{1,3} + e_{2,3}$ that reduces the size of $M_{32}$ to 38. There is another independent linear equation $-e_{1,2} + e_{1,5} - e_{1,3} - e_{2,3} + e_{2,4}$ that reduce the size of $M_{32}$ to 36. For the list $M_{33}$ we get the condition $e_{3,6} = e_{3,5} - e_{1,1} + e_{1,3} - e_{2,5} + e_{2,3}$ that reduce the size of $M_{33}$ to 24.

By making the above linear changes in the exponents of the $E_i$, new matrices $E'_i$ and lists that have $(32, 36, 24)$ monomials appear, where one can verify that there are no quadratic relations among the coefficients $H_{ij}$. using a CAS system one can compute binomial relations of the type $\prod(H_{ij}) - \prod(H_{kl})$ up to some degree. In this example we check with Maple that there are no binomial relations up to degree 10 .

This scheme can not be used for the kernel of the DME as is, because the three determinants $det(E'_i)$ are a power of 2 and then the inverse $F^{-1}$ will have a small number of monomials. If we do not use the last linear relation we get $M_{33}$ with 48 monomials, and there is no reduction of monomials there are many QR which eventually will allow us to compute the last component the matrix $L_3$. A priori, this is not a problem for the security of the schema because the other two components of $L_3$ can not be obtained.

By checking the final lists of monomials, we can observe and interesting structure:
if we make the changes of variables in $S_1$, $S_2$ and $S_3$:

$$S_1 = \begin{bmatrix} x_1^{2^{e_{1,1}+e_{1,1}+e_{2,1}}} = y_{11}, x_2^{2^{e_{1,1}+e_{1,1}+e_{2,1}}} = y_{12}, x_3^{2^{e_{1,4}+e_{1,1}+e_{2,1}-e_{1,3}+e_{3,1}}} = y_{13}, \\ x_4^{2^{e_{1,4}+e_{1,1}+e_{2,1}-e_{1,3}+e_{3,1}}} = y_{14}, x_5^{2^{e_{1,2}+e_{2,1}+e_{3,1}}} = y_{15}, x_6^{2^{e_{1,2}+e_{2,1}+e_{3,1}}} = y_{16} \end{bmatrix}$$

$$S_2 = \begin{bmatrix} x_1^{2^{e_{1,1}+e_{2,1}+e_{3,3}}} = y_{21}, x_2^{2^{e_{e_{1,1}}+e_{2,1}+e_{3,3}}} = y_{22}, x_3^{2^{e_{1,4}+e_{1,1}+e_{2,1}-e_{1,3}+e_{3,3}}} = y_{23}, \\ x_4^{2^{e_{1,4}+e_{1,1}+e_{2,1}-e_{1,3}+e_{3,3}}} = y_{24}, x_5^{2^{e_{1,2}+e_{2,1}+e_{3,3}}} = y_{24}, x_6^{2^{e_{1,2}+e_{2,1}+e_{3,3}}} = y_{26} \end{bmatrix}$$

$$S_3 = \begin{bmatrix} x_1^{2^{e_{1,3}+e_{2,3}+e_{3,5}}} = y_{31}, x_2^{2^{e_{1,3}+e_{2,3}+e_{3,5}}} = y_{32}, x_3^{2^{e_{1,4}+e_{2,3}+e_{3,5}}} = y_{33}, \\ x_4^{2^{e_{1,4}+e_{2,3}+e_{3,5}}} = y_{34}, x_5^{2^{e_{1,2}-e_{1,1}+e_{1,3}+e_{2,3}+e_{3,5}}} = y_{35}, x_6^{2^{e_{1,2}-e_{1,1}+e_{1,3}+e_{2,3}+e_{3,5}}} = y_{36} \end{bmatrix}$$

we get polynomials $\overline{F_i} = F_i(y) \in \mathbb{F}_q[y_{11}, \dots y_{36}]$ of low degree 6 or 7. Therefore, using $S_1, S_2, S_3$ and $\overline{F_i}(y)$ instead of $F_i(x)$ as public key will make faster encryption for DME-KEM and faster signature verification for DME-SIGN.

## 4. Computing the coefficients of the public key $F$

Once the list of monomials of the $F_{r,j}$ is obtained, one gets the coefficient of each group of polynomials by evaluating the polynomials $F_{r,1}, ..., F_{r,8}$. The set of pairs $(\underline{c}, F_{r,j}(\underline{c}))$ should be big enough to guarantee that the corresponding linear equations are independent. That is, if $Q_k = [q_1 ... q_d]$ and $F_{r,j} = \sum_{i=1}^{d} f_{rji} q_i(x)$, we take vectors $\underline{c}_1, \dots, \underline{c}_R$ such that the linear equations on the coefficients $f_{rij}$ in $F_k(c_e) = \sum f_{rji} q_i(c_e)$ are independent and can be solved to get the coefficients of the polynomials $F_{r,1}, \dots, F_{r,8}$.

To compute the polynomials $F_{r,k}$ faster we can use the same idea used to compute the lists of monomials of the polynomial $(F_{i,2j-1} + \bar{u}F_{i,2j})^{\alpha}(F_{i,2k-1} + \bar{u}F_{i,2k})^{\beta}$, i.e. $M_{ij}^{\alpha} \otimes M_{ik}^{\beta}$. Let $s_{ij}$ be the size of the list $M_{ij}$. Now, regard $M_{ij}$ as a $1 \times s_{ij}$ matrix, which by abuse of notation, we will still write it as $M_{ij}$. We denote by $C_{ij}$ the $s_{ij} \times 2$ matrix of the coefficients of the polynomials $F_{i,2j-1}$

and $F_{i,2j}$ on the monomials of $M_{ij}$, as shown in the following formula:

$$C_{ij} = \begin{bmatrix} c_{11}^{ij} & c_{12}^{ij} \\ c_{21}^{ij} & c_{22}^{ij} \\ \vdots & \vdots \\ c_{s_{ij}1}^{ij} & c_{s_{ij}2}^{ij} \end{bmatrix}$$

Now we have that $F_{i,2j-1} + \bar{u}F_{i,2j} = M_{ij} \cdot C_{ij} \cdot (1, \bar{u})^t$.

If $\alpha = 2^b$, then $(F_{i,2j-1} + \bar{u}F_{i,2j})^\alpha = M_{ij}^\alpha \cdot C_{ij}^\alpha \cdot (1, \bar{u}^\alpha)^t$.

Applying the mixed-product property of the Kronecker product we get:

$$\begin{aligned}(F_{i,2j-1} + \bar{u}F_{i,2j})^\alpha \cdot (F_{i,2k-1} + \bar{u}F_{i,2k})^\beta \\ &= (M_{ij}^\alpha \cdot C_{ij}^\alpha \cdot (1, \bar{u}^\alpha)^t) \otimes (M_{ik}^\beta \cdot C_{ik}^\beta \cdot (1, \bar{u}^\beta)^t) \\ &= (M_{ij}^\alpha \otimes M_{ik}^\beta) \cdot (C_{ij}^\alpha \otimes C_{ik}^\beta) \cdot (1, \bar{u}^\beta, \bar{u}^\alpha, \bar{u}^{\alpha+\beta})^t\end{aligned}$$

Let's call $U_{\alpha\beta}$ the $4 \times 2$ matrix defined by

$$(1, \bar{u}^\beta, \bar{u}^\alpha, \bar{u}^{\alpha+\beta})^t = U_{\alpha\beta} \cdot (1, \bar{u})^t.$$

Then, we have the following result:

**Lemma 4.1.** *The matrix of coefficients of* $(F_{i,2j-1} + \bar{u}F_{i,2j})^\alpha \cdot (F_{i,2k-1} + \bar{u}F_{i,2k})^\beta$ *with respect of the monomials* $M_{ij}^\alpha \otimes M_{ik}^\beta$ *is* $(C_{ij}^\alpha \otimes C_{ik}^\beta) \cdot U_{\alpha\beta}$

Now, we can compute the coefficients of the $F_{r,j}$ with algorithms similar to Rm and MON. Given the matrices of coefficients $(M, C)$ of a component we define Rc(C) the matrix coefficient obtained by adding of the coefficient of a the same monomial in the case that is repeated in the monomial list $M$.

**Algorithm 4.1** COE, compute the coefficients of the public-key polynomials.

**Input:** $(E_1, \ldots, E_r, L_0 \ldots L_r)$
**Output:** $(C_{r1}, C_{r2}, C_{r3}, C_{r4})$

$M_{01} \leftarrow [x_1, x_2]$, $M_{02} \leftarrow [x_3, x_4]$, $M_{03} \leftarrow [x_5, x_6]$, $M_{04} \leftarrow [x_7, x_8]$
$C_{01} \leftarrow A_{01}, \ldots, C_{04} \leftarrow A_{04}$
**for** $i = 0$ **to** $r - 1$ **do**
  **for** $k = 1$ **to** $4$ **do**
    **if** $\alpha_{i,2k} \neq 0$ **then**
      $C_{(i+1)k} = \left(C_{ik_1}^{\alpha_{i,2k-1}} \otimes C_{ik_2}^{\alpha_{i,2k}}\right) \cdot U_{\alpha_{i,2k-1},\alpha_{i,2k}}$
    **else**
      $C_{(i+1)k} = C_{ik_1}^{\alpha_{i,2k-1}} \cdot (1, \bar{u}^\alpha)$
    **end if**
    $C_{(i+1)k} = Rc(C_{(i+1)k})$
    **if** $a_{(i+1)k} \neq 0$ **then**
      add $a_{(i+1),k}$ to the coefficient matrix $C_{(i+1)k}$
    **end if**
  **end for**
**end for**

## 5. DME as a trapdoor one way permutation

Let's assume that the public key is

$$F = \Psi(L_0, \ldots, L_r, E_1, \ldots, E_r) : \mathbb{F}_q^8 \to \mathbb{F}_q^8.$$

By construction, $F$ is a composition of bijections of $(\mathbb{F}_{q^2} \setminus \{0\})^4$ if there is no affine translations $a_{i,j} = 0$ for all $i$, that is:

**Remark 5.1.** *Let $\mathbb{U} = \bar{\phi}^{-1}((\mathbb{F}_{q^2} \setminus \{0\})^4) \subset \mathbb{F}_q^8$ then $F : \mathbb{U} \to \mathbb{U}$ is a bijection.*

If we add an affine translation in only the step $i_0$, then given $\underline{x}_0 \in \mathbb{U}$ the translation $a_{i_0,j}$ can produce a 0 in the step $i_0$, which in turn will give $F(\underline{x}) \notin \mathbb{U}$, so $F$ can not be inverted at $F(\underline{x})$. On the other hand, if $F(\underline{x}) \in \mathbb{U}$, then $F$ is invertible at $F(\underline{x})$, that is, we have:

**Theorem 5.2.** *Let $F$ be a public key map such that there is only one step $1 \leq i_0 < r$ with non-zero affine components then $F$ is invertible at $F(\underline{x})$ if $F(\underline{x}) \in \mathbb{U}$. In other words,*

$$F : \mathbb{U} \cap F^{-1}(\mathbb{U}) \to \mathbb{U} \cap F(\mathbb{U})$$

*is a bijection.*

*Proof.* Let $\underline{x}$ and $\underline{y}^0 = (y_1^0, y_2^0, y_3^0, y_4^0) = \bar{\phi}(L_0(\underline{x})) \in (\mathbb{F}_{q^2} \setminus \{0\})^4$.

By construction, all the successive maps of which $F$ is made up are bijections in $(\mathbb{F}_{q^2} \setminus \{0\})^4$ or $\mathbb{U}$ until we get the linear map $L_{i_0}$. If we have that $L_{i_0}(\underline{x}^{i_0}) \in \mathbb{U}$, this property is preserved by the rest of the maps, so $F(\underline{x}) \in \mathbb{U}$.

On the contrary, if $L_{i_0}(\underline{x}^{i_0}) \notin \mathbb{U}$, then there exist a $k$ such that $L_{i_0 k}(x_{i_0,2k-1}, x_{i_0,2k}) + a_{i_0,k} = (0,0)$. As $det(E_i) \neq 0$, there is a non-zero entry $\alpha_{i_0 k}$ in the $k$-th column, $\underline{y}^{\alpha_{ik}} = 0$ and $F_{i_0+1}(x) \notin \mathbb{U}$ this property is preserved by the rest of the maps because there are no more translations, hence $F(\underline{x}) \notin \mathbb{U}$.

In this case, it is clear that there are some $\underline{x}$ such that $F(\underline{x}) \notin \mathbb{U}$ and therefore $F(\mathbb{U}) \not\subset \mathbb{U}$. This means that there will be messages that, after padding, $\underline{x} \in \mathbb{U}$ but can not be signed. By the same argument above, those messages can be detected because $F^{-1}(\underline{x}) \notin \mathbb{U}$ and the message can be signed by changing the padding. □

In the case that there are affine translations in more than one step then can be failure of decryption even if $F(\underline{x}) \in \mathbb{U}$. In example 1, if we take $a_{11} \neq 0, a_{21} \neq 0, a_{22} \neq 0$ and the rest of the $a_{ij}$ are zero, after $L_1$ we may have $(x_1^1, x_2^1) = (0,0)$ and $E_1(y^0)$ can not be inverted but as $a_{21} \neq 0$ and $a_{22} \neq 0$ then we may have $\underline{x}^2 \in \mathbb{U}$ and $F(\underline{x}) \in \mathbb{U}$, but clearly $F$ is not invertible at $F(\underline{x})$. One can check that if we take $a_{13} \neq 0$ and $a_{21} \neq 0$ then $F$ has the property that if $F(\underline{x}) \in \mathbb{U}$ then $F^{-1}(F(\underline{x})) = \underline{x}$, but the converse of this statement is not true because the matrices $E_i^{-1}$ have all the entries different from zero.

From now on, we will assume that there are non zero affine translations only at one level $1 < i_0 \leq r-1$. Then, as shown above, $F : \mathbb{U} \cap F^{-1}(\mathbb{U}) \to \mathbb{U} \cap F(\mathbb{U})$ is a bijection. We can consider $F$ as trapdoor one way permutation. Given a message $m$, we add some padding to get $\underline{x} \in \mathbb{U}$. Now we can get $F(\underline{x}) \notin \mathbb{U}$ with probability $1/q^2$, then we change the padding and try again. For the signature of a message we can do the same. In this work, we are mainly interested into find out the performance and security of the DME and we will not elaborate more about padding . For padding, we use the standards OAEP for PKE and KEM and PSS00 for signature, and we will denote by DME-OAEP and DME-PSS the corresponding cryptosystem.

## 6. Security of the DME

The security of the DME depends on the chosen settings and parameters. We will describe first the setting of the the scheme $DME(r, n, 2^e)$:

6.1. **The configuration of matrices.** We define a **Configuration of Matrices** ( $\mathcal{CM}$) as a list of $r$ matrices for the exponentials where the non zero entries are substituted by 1. We denote such matrices by $E_i^*$. Let $\mathcal{CM} = [E_r^*, \dots E_1^*]$ be a configuration. Then, it is easy to get the number of monomials of the each component of $F$ from $\mathcal{CM}$ if there are no repeated monomials, just compute $E^* = E_r^* \cdots E_1^*$ and let $t_k$ be the sum of the entries in the $k-th$ row of $E^*$, in which case the number of monomials of the components $F_{2k-1}, F_{2k}$ is $2^{t_k}$. In the example 1 we have

$$E^* = E_3^* \cdot E_2^* \cdot E_1^* = \begin{pmatrix} 3 & 1 & 3 \\ 3 & 2 & 2 \\ 2 & 1 & 3 \end{pmatrix}$$

and the corresponding number of monomials is $(2^7, 2^7, 2^6)$. The algorithm RED reduce number of monomials to $(32, 36, 24)$. Please notice that the output of algorithm RED depend inly in the configuration $\mathcal{CM}$, we will denote it by RED($\mathcal{CM}$).

Another example is the configuration $\mathcal{CM}_2$ that we study in the next section and we implemented with $q = 2^{64}$. The matrices of $\mathcal{CM}_2$ are:

$$E_1^* = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, E_2^* = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, E_3^* = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, E_4^* = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$$E^* = E_4^* \cdot E_3^* \cdot E_2^* \cdot E_1^* = \begin{pmatrix} 5 & 2 & 1 & 1 \\ 4 & 2 & 1 & 2 \\ 5 & 2 & 0 & 1 \\ 3 & 2 & 1 & 3 \end{pmatrix}$$

and the number of monomials is $(2^9, 2^9, 2^8, 2^9)$. After imposing 8 linear conditions we reduce the number of monomials to $(72, 90, 36, 96)$ and there are no QR.

When we consider in 6.3 a possible attack of the DME by Weil descent, the $t_k$ give also the degree of the components $F_{2k-1}, F_{2k}$ when we express them as polynomials over $\mathbb{F}_2$. In fact one of the main reason to use 4 or more exponentials is to increase the values in the list $(t_1, t_2, t_3, t_4)$.

6.2. **Reduction of the number of monomials.** Given a $\mathcal{CM}$, it is straightforward to use the algorithm RED($\mathcal{CM}$) in section 3 to reduce drastically the number of monomials of the $F_i$, in fact the linear relations depends only on $\mathcal{CM}$ and they are easy to compute. It is more complicate to get the maximal reduction and simultaneously to produce a determinant $det(E_i)$ with a big binary weight so that $E_i^{-1}$ yields a large number of monomials. Nevertheless, it is possible to get a few $\mathcal{CM}$ that verify that condition. Remember that the algorithm produce some linear condition on the exponents of the matrices that allow us to eliminate some parameters and find new matrices with exponents in the remainder parameters.

An important point for the security of the DME is that the final monomials depend on fewer parameters than the final matrices, this fact implies that given the monomials the public key $F$,we can deduce the parameters involved in the public key and the rest of free parameters will produce a big list of matrices with the same exponents as $F$. In example 1, there are initially 17 parameters that reduce to 12 after the reduction of monomials and examining the lists of exponents that appear in $(F, S1, S2, S3)$ we can verify that the exponents of $F$ depend only on 6 parameters as follows:

We make a list $EX$ with the second exponent of the monomials in $x_i$ in the lists $\{S_1, S_2, S_3\}$,

$$[e_{1,1} + e_{2,1} + e_{3,1}, e_{1,4} + e_{1,1} + e_{2,1} - e_{1,3} + e_{3,1}, e_{1,2} + e_{2,1} + e_{3,1}, e_{1,1} + e_{2,1} + e_{3,3},$$
$$e_{1,4} + e_{1,1} + e_{2,1} - e_{1,3} + e_{3,3}, e_{1,2} + e_{2,1} + e_{3,3}, e_{3,3} + e_{2,3} + e_{3,5},$$
$$e_{1,4} + e_{2,3} + e_{3,5}, e_{1,2} - e_{1,1} + e_{1,3} + e_{2,3} + e_{3,5}]$$

The 9 linear forms of $EX$ define a linear map $H : \mathbb{Z}_{q^2-1}^{12} \to \mathbb{Z}_{q^2-1}^{9}$ that has rank 6.. That is given the Public Key there are $12 - 6 = 6$ free parameter in the matrices of $\mathcal{CM}$ that produce the given PK because fixing $(F, S1, S2, S3)$ we fix a vector $h_0$ of the image of $H$ and its anti image $H^{-1}(h_0)$ is an affine space of dimension 6. As the 6 remaining parameters verify that $1 < e_{ij} \leq q^2 - 1$, given the exponents of the public key and the base field $\mathbb{F}_{2^e}$, there are $2^{6(\log_2(e)+1)}$ matrices that can produce those exponents.

It is interesting to remark that the reduction algorithm RED($\mathcal{CM}$) do not change the number of free parameters , for instance in the example before RED there are 17 parameters and the rank of $H$ is 11 that give the same number of free parameters 6.

For the configuration $\mathcal{CM}_1$ in section 7 , we have initially 23 parameters and the reduction of exponent left 16 free parameters. Now the final exponents depend only on 8 parameters, that means that given the monomials of the public key there are $2^{8(\log_2(e)+1)}$ sets of matrices that produce the same monomials. This means that for $q = 2^{64}$, there are $2^{56}$ sets of matrices for a given public key.

For the configuration $\mathcal{CM}_2$ in section 7 we start with 28 parameters $e_{i,j}$ corresponding to the non zero entries in $\mathcal{CM}_2$ and we impose 8 linear conditions to get $(72, 90, 36, 96)$ monomials. We fix $F$, in order to determine the monomials in $F$ we need to give values to 8 of the remaining parameters that is after we fix $F$ there are 12 free parameters. For instance if $q = 2^{64}$ the given $F$ there are $2^{12(\log_2(e)+1)} = 2^{84}$ set of matrices from $\mathcal{CM}$ that give the same exponents as $F$.

Given two sets of matrices $MA_1$ and $MA_2$ with the same final exponents we can ask if both sets can give the same coefficients for $F$, that is $MA_1$ and $MA_2$ can produce equivalent private keys. Counting the parameters on the coefficients of $F$ one can see that it is very unlikely that two sets matrices $MA_1$ and $MA_2$ gives the same $F$. For instance in example 1 the coefficients of $F$ are polynomials on the entries of the matrices of the linear isomorphisms $L_i$, that is 48 variables and $(32 + 36 + 48)2 = 232$ coefficients in $F$. The sets $MA_1$ and $MA_2$ will give two different solutions to this system of 232 equations with 48 variables.

6.3. **Weil descent.** Taking a base of $\mathbb{F}_q$ over $\mathbb{F}_2$, namely $B = \{v_1, \ldots, v_e\}$, we can express the polynomial of $F$ as polynomials $\tilde{F}$ in $ne$ variables over $\mathbb{F}_2$. It is easy to verify that before the reduction of monomials, the degrees of the components of $\tilde{F}$ are $(t_1 \ldots t_{n/2})$. In fact the raise of the binary degree of the public key was one of the reasons to use more than two exponentials on the DME ([4] )

The reduction of monomials can produce also a reduction of the degrees of $\tilde{F}$ and it is not possible to determine apriori the degrees of the $\tilde{F}$. One has to examine the list of monomials after the reduction and compute the degrees. For instance, in example 1 the degrees reduced from $(7, 7, 6)$ to $(5, 6, 6)$ . For $\mathcal{CM}_2$ in section 7 the degrees of $\tilde{F}$ are reduced from $(9, 9, 8, 9)$ to $(7, 8, 6, 7)$.

6.4. **Gröbner basis.** To determine the resistance of a $\mathcal{CM}$ to the Gröbner basis attack, we have to estimate the complexity of computing the Gröbner basis of the ideal

$$I = \langle f_1(\underline{x}) - y_1, \ldots, f_n(\underline{x}) - y_n, x_1^{2^e} - x_1, \ldots, x_n^{2^e} - x_n \rangle$$

where $F(\underline{x}) = \underline{y}$. Let $sd(I)$ be the **solving degree of** $I$, i.e. the the highest degree of polynomials involved in the computation of the Gröbner basis. The complexity of computing the Gröbner basis using a algorithm like F4/F5 is bounded from above by

$$O\left(\binom{n + sd(I)}{n}^{\omega}\right) \tag{2}$$

where $\omega$ is the exponent in the complexity of matrix multiplication. It is easy to see that this upper bound is well above $O(2^{256})$, since $sd(I)$ is bounded below be degree of the initial basis $I$ , $x_n^{2^e} - x_n \in I$ and a typical monomial of $F$ has from 4 to 8 variables we can force the degree of $I$ to

be bounded below by $2^e$. Now if we take a $\mathcal{CM}$ with 8 variables (2) is bounded below by $2^{16e}$. If we use $q = 2^{64}$ then the complexity is bounded by $O(2^{1024})$.

We can safely assume that $2^e \leq sd(I)$, the problem is that we do not know if the bound (2) is accurate or not for the Gröbner basis computation of this kind of ideals. In order to make an experimental testing of the above bound, we used Magma in a cluster with several fat nodes with 512 Gb of RAM each. After an extensive series of computations, Magma can find the Gröbner basis only for $q = 2^3$ and or $q = 2^4$. For $q = 2^5$ Magma exhausted the RAM before the end of the computation. Here are the conclusions that we get from our experiments.

- Given a $\mathcal{CM}$, the time of computing the Gröbner basis depends mainly on the exponents of $F$, but not of the actual matrices that give $F$.
- The initial basis $I$ can be considered sparse because it has a low number of monomials by rapport to the degree but the intermediate computations of Magma show that the number of monomials can be very big.
- The upper bound (2) seems to be accurate, but further research is needed to confirm this fact.

Of course those conclusions can not be extrapolated for higher $q$. If any one can try to verify those conclusion for $e \geq 5$ we can provide them the basis for different $\mathcal{CM}$.

We can use the special form of the monomials that allow to substitute $F(\underline{x})$ by $F(y_{11}, \dots)$ as described in example 1, but this will give a greater complexity because we will have much more variables but the degree will not decrease much. Let's explain this in the example 1. We have now that $\bar{F}$ has 18 variables $\{y_{11}, \dots, y_{36}\}$. If we examine the relations among the $x_i$ and the $y_{jk}$ given by the lists $S_1, S_2, S_3$ we find, for instance, $x_1^{2^{e_{3,1}+e_{1,1}+e_{2,1}}} = y_{21}, x_1^{2^{e_{1,3}+e_{2,3}+e_{3,3}}} = y_{31}$, so we would get a relation $y_{31} = y_{21}^{2^a}$ for some $a \leq q$ and we would end with a basis $\bar{I}$ such that $sd(\bar{I}) \geq 2^e$ as before.

6.5. **Estimation of the number of monomials of the inverse.** As we mentioned earlier we set that $1/det(A_i)$ has a big binary weight to get a number of monomials of the inverse big enough. Next we will do a more precise estimation of this number of monomial. Lets denote by $C_i$ the matrix obtained from $A_i^{-1}$ changing the non zero entries by 1. If the entries of $A_i^{-1}$ were powers of two, then the number of monomials of $F^{-1}$ is $(s_1, s_2, s_3, s_4)$ where $s_i$ is the sum of entries of de row $i$ of $C^* = C_1^* \cdots C_r^*$ but now each entry in $A_i^{-1}$ is multiplied by $1/det(A_i)$ or $-1/det(A_i)$. Let $b_i$ the binary weight of $1/det(A_i)$ mod $q^2 - 1$ then $128 - b_i$ is the binary weight of $-1/det(A_i)$ mod $q^2 - 1$. We can impose that $b_i \leq 128 - b_i$ and then $b_i$ is a lower bound for the binary weight of each entry of $A_i^{-1}$. It is easy to verify that the number of monomial is bounded below be $(s_1 b_i, s_2 b_i, s_3 b_i, s_4 b_i)$. If the $\mathcal{CM}$ has two matrices $A_i, A_j$ with determinant not a power of 2 then one get the bound $(s_1 b_i b_j, s_2 b_i b_j, s_3 b_i b_j, s_4 b_i b_j)$.

For the configuration $\mathcal{CM}_1$ with 3 exponentials only $A_3$ has determinant not a power of 2. The computation of $C^*$ gives

$$C^* = C_1^* \cdot C_2^* \cdot C_3^* = \begin{pmatrix} 3 & 3 & 3 & 3 \\ 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 \\ 6 & 6 & 6 & 6 \end{pmatrix}$$

This means that each polynomial has at least $2^{12 b_3}$ monomials. If we take $b_3 = 10$ they have at least $2^{120}$ monomials that gives a complexity of $120w \geq 256$ bits .

For the configuration $\mathcal{CM}_2$ with 4 exponentials $A_3$ has determinant not a power of 2. The computation of $C^*$ gives

$$C^* = C_1^* \cdot C_2^* \cdot C_3^* \cdot C_4^* = \begin{pmatrix} 1 & 4 & 4 & 4 \\ 3 & 9 & 9 & 9 \\ 4 & 12 & 12 & 12 \\ 9 & 9 & 9 & 9 \end{pmatrix}$$

This means that each polynomial has at least $2^{13b_3}$ monomials. If we take $b_3 = 9$ they have at least $2^{117}$ monomials that gives a complexity of $117w \geq 256$ bits .

6.6. **Structural Cryptanalysis.** In the structural cryptoanalysis we can start by considering that given the monomials of the public key $F$ and $\mathcal{CM}$,we get after the reduction of monomials $2^{l(\log_2(e)+1)}$ matrices that get the same monomials for $F$, and different sets of matrices will give different coefficients of the public key with hight probability. We can try to invert $F$ directly by starting with the inverse of the last linear map $L_r$. As we explained in section 3, for each linear map $L_{rk}$ we can use the the relations $H_{ij} = Q_{ij}(B, C) = \sum B_k C_l$, to get homogeneous implicit equations for the $H_{ij}$ by eliminating $B_k$ and $C_l$ from those equations. It is not clear what is the complexity of the Gröbner basis computation for eliminate the $B_k, C_l$ from the equations $H_{ij} = Q_{ij}(B, C)$ but the number of variables is high, for instance for the last component of $\mathcal{CM}_2$ we have $96 + 36 = 132$ variables.

We can give an upper bound of the cost using linear algebra as follows. Let $n_1$ the the number of variables $H_{ij}$, that is, the size of the corresponding list $M_{rk}$ and let $n_2$ be the number of variables $B_k, C_l$. Let $P_d(H_{ij})$ a homogeneous polynomial of degree $d$, by making the substitution $Q_{2d}(B, C) = P_d(Q_{ij})$ we get a polynomial of degree $2d$ , $Q_{2d}(B, C)$ in the variables $B_k$ and $C_l$ Taking the coefficients of $P_2(H_{ij}) = \sum c_{ij} H_{ij}$ as variables , the coefficients of $Q_{2d}(B, C)$ are linear forms on the $c_{ij}$ and we can impose the condition $Q_{2d}(B, C) = 0$ by solving the corresponding linear equations. Let $HM(n, d) = \binom{n+d-1}{d}$ be the number of monomials of degree $d$ in $n$ variables. In our situation $HM(n_1, d) < HM(n_2, 2d)$ for small $d$, but we can get $HM(n_1, d) > HM(n_2, 2d)$ taking $d$ big enought and the implicit equations on $H_{ij}$ can be obtained by solving those linear equations. For instance, in for the second list of example 1, we have $n_1 = 36$, $n2 = 12 + 8$, and for $25 \leq d$ one has $HM(36, d) > HM(20, 2d)$. For $d = 25$, $HM(36, d) \approx 2^{55}$ and then solving the linear equations take $O(2^{55\omega}) = O(2^{131})$.

It is possible to get higher cost for solving the linear equations. For instance, we have a $\mathcal{CM}$ with $n_1 = 120$, $n_2 = 56$, $d = 45$, $HM(120, 45) \approx 2^{137}$ and solving the linear system requires at least $O(2^{317})$ operations.

This means that for each $k \leq n/2$ we would have a solution for the matrix of $L_{rk}$ that is defined up to a multiplicative constant $\lambda_k$, and given $(\lambda_1, \ldots, \lambda_{n/2}) \in \mathbb{F}_q \setminus \{0\}$ we can find the inverse of the $L_{rk}$. For instance, if we have $n = 8$ and $q = 2^{64}$, we have a choice of $O(2^{256})$ vectors $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ and the translations at round $i_0$ will prevent to transfer the $\lambda_k$ to the next rounds . For each vector we have $2^{7l}$ possible matrices, where $l$ is the number of free parameters in the exponents.

## 7. Implementation and timings

For test the timing we implemented two configuration of matrices $\mathcal{CM}_1$ for DME-$(3, 8, 2^{64})$ and $\mathcal{CM}_2$ for DME-$(4, 8, 2^{64})$ for KEM (with OAEP as padding) and for SIGN (with PSS as padding) For the implementation, we used the special instructions that modern Intel processors have to perform arithmetic in finite fields, which gives th algorithm an impressive boost in performance. In all the cases, when a hash function was needed, we used the NIST approved standard SHA-2 function.

The matrices for $\mathcal{CM}_1$ with DME-$(3, 8, 2^{64})$ are the following:

$$E_1^* = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} , E_2^* = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} , E_3^* = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$E^* = E_3^* \cdot E_2^* \cdot E_1^* = \begin{pmatrix} 4 & 1 & 1 & 1 \\ 4 & 2 & 1 & 0 \\ 4 & 2 & 0 & 1 \\ 4 & 1 & 0 & 2 \end{pmatrix}$$

The number of monomials of $F$ before reduction is $(2^7, 2^7, 2^7, 2^7)$ and the binary degree $(7, 7, 7, 7)$. As we explain in 6.1 and in example 1 in more detail, given the $\mathcal{CM}$, the linear equations that reduce the monomials are determined by matrices of the $\mathcal{CM}$. The only alternative is if we use all the linear equations for the reduction or we use all but one in order to ensure that $det(E_i)$ is not 0 or a power of 2. For this $\mathcal{CM}$ in particular, we get 8 linear equations, one for each row from the $E_2$ and $E_3$. By substituting the 8 equations one get $(40, 30, 30, 30)$, but then $det(E_3) = 0$ which is not valid. For this reason we drop the linear equation coming from the first row of $E_3$ and we get $(72, 30, 30, 30)$. The equations that we obtain with the algorithm RED are

$$\begin{aligned} e_{2,2} &= e_{2,1}, \\ e_{2,4} &= e_{1,1} + e_{2,3} - e_{1,5}, \\ e_{2,6} &= e_{1,1} + e_{2,5} - e_{1,7}, \\ e_{2,8} &= e_{1,5} + e_{2,7} - e_{1,7}, \\ e_{3,4} &= e_{2,1} + e_{3,3} - e_{2,5}, \\ e_{3,6} &= e_{2,3} + e_{3,5} - e_{2,5}, \\ e_{3,8} &= e_{1,1} - e_{1,5} + e_{2,3} + e_{3,7} - e_{2,7} \end{aligned}$$

The equation that we do not use is $e_{3,2} = e_{1,1} - e_{1,5} + e_{2,1} + e_{3,1} - e_{2,7}$. As in example 1, we can make a change to the variables $y_{ij}$, which in this case are given by:

$$S_1 = \begin{bmatrix} x_1^{2^{e_{3,1}+e_{1,1}+e_{2,1}}} = y_{11}, x_1^{2^{e_{3,2}+e_{1,5}+e_{2,7}}} = y_{12}, x_2^{2^{e_{3,1}+e_{1,1}+e_{2,1}}} = y_{13}, x_2^{2^{e_{3,2}+e_{1,5}+e_{2,7}}} = y_{14}, \\ x_3^{2^{e_{1,2}+e_{2,1}+e_{3,1}}} = y_{15}, x_4^{2^{e_{1,2}+e_{2,1}1+e_{3,1}}} = y_{16}, x_5^{2^{e_{1,4}+e_{2,1}+e_{3,1}}} = y_{17}, x_6^{2^{e_{1,4}+e_{2,1}+e_{3,1}}} = y_{18}, \\ x_7^{2^{e_{1,6}+e_{2,7}+e_{3,2}}} = y_{19}, x_8^{2^{e_{1,6}+e_{2,7}+e_{3,2}}} = y_{1,10} \end{bmatrix}$$

$$S_2 = \begin{bmatrix} x_1^{2^{e_{1,1}+e_{2,1}+e_{3,3}}} = y_{21}, x_2^{2^{e_{1,1}+e_{2,1}+e_{3,3}}} = y_{22}, x_3^{2^{e_{1,2}+e_{2,1}+e_{3,3}}} = y_{23}, \\ x_3^{2^{e_{1,2}+e_{2,1}+e_{3,3}}} = y_{24}, x_5^{2^{e_{1,4}+e_{2,1}1+e_{3,3}}} = y_{25}, x_6^{2^{e_{1,4}+e_{2,1}1+e_{3,3}}} = y_{26} \end{bmatrix}$$

$$S_3 = \begin{bmatrix} x_1^{2^{e_{1,1}+e_{2,1}+e_{3,3}}} = y_{31}, x_2^{2^{e_{1,1}+e_{2,1}+e_{3,3}}} = y_{32}, x_3^{2^{e_{1,2}+e_{2,1}+e_{3,3}}} = y_{33}, \\ x_4^{2^{e_{1,2}+e_{2,1}+e_{3,3}}} = y_{34}, x_5^{2^{e_{1,4}+e_{2,1}+e_{3,3}}} = y_{35}, x_6^{2^{e_{1,4}+e_{2,1}+e_{3,3}}} = y_{36} \end{bmatrix}$$

$$S_4 = \begin{bmatrix} x_1^{2^{e_{1,1}+e_{2,3}+e_{3,7}}} = y_{41}, x_2^{2^{e_{1,1}+e_{2,3}+e_{3,7}}} = y_{42}, x_3^{2^{e_{1,2}+e_{2,3}+e_{3,7}}} = y_{43}, \\ x_4^{2^{e_{1,2}+e_{2,3}+e_{3,7}}} = y_{44}, x_7^{2^{e_{1,6}+e_{1,1}+e_{2,3}-e_{1,5}+e_{3,7}}} = y_{45}, x_8^{2^{e_{1,6}+e,11+e_{2,3}-e_{1,5}+e_{3,7}}} = y_{46} \end{bmatrix}$$

With this changes the degrees of $\bar{F}$ aree $(5, 8, 7, 7)$ and the binary degrees of $\tilde{F}$ after Weil descent are $(7, 6, 6, 6)$. The translations are in the first component of the third linear map, yielding $(78, 36, 30, 30)$ monomials. The length of the secret key is 542 bytes, the length of the public key is 2739 bytes, and a ciphertext takes 64 bytes.

We take $n = 8$ and $q = 2^{64}$, we have a choice of $O(2^{256})$ vectors $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$,and for each vector $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ we get (by 6.5) $l = 8$ free parameters for the matrices of this $\mathcal{CM}$, i.e. $2^{7l} = 2^{56}$ possible matrices, so this gives a total of $O(2^{312})$ operations that are enough for the NIST level 5.

The matrices for $\mathcal{CM}_2$ with DME-$(4, 8, 2^{64})$ are the following:

$$E_1^* = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, E_2^* = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, E_3^* = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, E_4^* = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

,

$$E^* = E_4^* \cdot E_3^* \cdot E_2^* \cdot E_1^* = \begin{pmatrix} 5 & 2 & 1 & 1 \\ 4 & 2 & 1 & 2 \\ 5 & 2 & 0 & 1 \\ 3 & 2 & 1 & 3 \end{pmatrix}$$

The number of monomial of $F$ before reductions is $(2^9, 2^9, 2^8, 2^9)$. After imposing 8 linear conditions we reduce the number of monomials to $(72, 90, 36, 96)$ and there are no QR. The equations that we obtain by method in section 3 are:

$$\begin{aligned}
e_{2,2} &= e_{1,1} + e_{2,1} - e_{1,3}, \\
e_{3,4} &= e_{1,1} - e_{1,3} + e_{2,1} - e_{2,3} + e_{3,3}, \\
e_{3,6} &= e_{1,7} - e_{1,6} + e_{2,6} - e_{2,7} + e_{3,5}, \\
e_{3,8} &= e_{1,3} - e_{1,1} + e_{2,3} - e_{2,5} + e_{3,7}, \\
e_{4,2} &= e_{3,1} - e_{3,4} + e_{4,1}, \\
e_{4,4} &= e_{2,1} - e_{2,5} + e_{3,3} - e_{3,5} + e_{4,3}, \\
e_{4,6} &= e_{1,1} - e_{1,3} + e_{2,1} - e_{2,3} + e_{3,3} - e_{3,7} + e_{4,5}, \\
e_{4,8} &= e_{1,1} - e_{1,3} + e_{2,5} - e_{2,5} + e_{3,5} - e_{3,7} + e_{4,7}
\end{aligned}$$

As in the previous 3 round case one can make changes the to get the polynomials $\bar{F}(y_{ij})$ and after those changes the degrees of $\bar{F}$ are $(9, 9, 8, 9)$ and the binary degrees of $\tilde{F}$ after Weil descent are $(7, 8, 6, 7)$.

If we take the first component $F_1$ of $F$ we have $n_1 = 72$ monomials and $n2 = 32$ variables $B_i, C_j$ and for $22 \leq d$ we have $HM(72, d) > HM(32, 2d)$. For $d = 22$, $HM(72, 22) \approx 2^{70}$ and finding the solutions would have complexity at least $O(2^{164})$ to find the matrix of the last linear $L_{41}$ up to a constant $\lambda_1$. Now we have a choice of $O(2^{256})$ vectors $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$,and for each vector $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ we get (by 6.5) $l = 12$ free parameters for the matrices of this $\mathcal{CM}$, i.e. $2^{7l} = 2^{84}$ possible matrices, so this gives a total complexity of $O(2^{340})$ operations that are enough for the NIST level 5.

The translations are in the first component of the fourth linear map, yielding $(80, 90, 36, 96)$ monomials. The length of the secret key is 675 bytes, the length of the public key is 4843 bytes, and a ciphertext takes 64 bytes.

The timings of DME-KEM have also been measured with SuperCop, to allow a fair comparison with other schemes. The DME implementation has been optimised for processors with the special `clmul` operation (carry-less multiplication) that gives a considerable speed-up in the arithmetic over finite fields of characteristic two. Currently, we are using a naive algorithm for computing inverses in $\mathbb{F}_q$ based on the binary exponentiation algorithm and the relation $a^{-1} = a^{q-2}$. Any optimization here would translate in further improvements in the timings.

| DME-$(3, 8, 2^{64})$-SIGN-PSS | |
|---|---|
| KeyGen | $380\mu s$ |
| Sign | $37\mu s$ |
| Verify | $7\mu s$ |
| Public Key | 2793 bytes |
| Secret Key | 542 bytes |
| Signature | 64 bytes |

| DME-$(3, 8, 2^{64})$-KEM-OAEP | |
|---|---|
| KeyGen | $384\mu s$ |
| Decrypt | $36\mu s$ |
| Encrypt | $7\mu s$ |
| Public Key | 2793 bytes |
| Secret Key | 542 bytes |
| Shared Secret | 32 bytes |
| Ciphertext | 64 bytes |

FIGURE 1. Timings for DME-SIGN (100 byte messages) and DME-KEM

| DME-$(4, 8, 2^{64})$-SIGN-PSS | |
|---|---|
| KeyGen | $931\mu s$ |
| Sign | $44\mu s$ |
| Verify | $9\mu s$ |
| Public Key | 4843 bytes |
| Secret Key | 675 bytes |
| Signature | 64 bytes |

| DME-$(4, 8, 2^{64})$-KEM-OAEP | |
|---|---|
| KeyGen | $929\mu s$ |
| Decrypt | $43\mu s$ |
| Encrypt | $9\mu s$ |
| Public Key | 4843 bytes |
| Secret Key | 675 bytes |
| Shared Secret | 32 bytes |
| Ciphertext | 64 bytes |

FIGURE 2. Timings for DME-SIGN (100 byte messages) and DME-KEM

| | NSL | KeyGen | Enc | Dec | PKey | SKey | SS | CText |
|---|---|---|---|---|---|---|---|---|
| dme-4r-8v-64b-oaep | 5 | 3468004 | 80097 | 216374 | 4843 | 675 | 32 | 64 |
| dme-3r-8v-64b-oaep | 5 | 1510324 | 46929 | 224497 | 2793 | 542 | 32 | 64 |
| kyber1024 | 5 | 137520 | 147921 | 112820 | 1568 | 3168 | 32 | 1568 |
| ntrukem743 | 5 | 2002204 | 426806 | 610610 | 1023 | 1173 | 48 | 1023 |
| mcelice348864 | 1 | 120686580 | 75741 | 278278 | 261120 | 6492 | 32 | 96 |
| sikep751 | 5 | 23110975 | 37404352 | 40155660 | 564 | 644 | 32 | 596 |
| bikel3 | 3 | 2729828 | 385831 | 8923861 | 3083 | 10105 | 32 | 3115 |

FIGURE 3. Average CPU cycles for KEM as measured by SuperCop on an Intel(R) Core(TM) i7-1165G7 @ 2.80GHz

| | NSL | KeyGen | Sign | Verify | PKey | Skey | Signature |
|---|---|---|---|---|---|---|---|
| dme-4r-8v-64b-pss | 5 | 4609827 | 222307 | 55484 | 4843 | 675 | 64 |
| dme-3r-8v-64b-pss | 5 | 1953078 | 182009 | 40197 | 2793 | 542 | 64 |
| dilithium2 | 2 | 169935 | 238597 | 147235 | 1312 | 2544 | 2420 |
| dilithium5 | 5 | 319828 | 617804 | 337222 | 2492 | 4880 | 4595 |
| falcon1024dyn | 5 | 78644060 | 2080846 | 310257 | 1793 | 2305 | 1330 |
| sphincsf256shake256robust | 5 | 23130618 | 530274683 | 25373313 | 64 | 128 | 49216 |

FIGURE 4. Average CPU cycles for SIGN as measured by SuperCop on an Intel(R) Core(TM) i7-1165G7 @ 2.80GHz (message length = 93 bytes)

## REFERENCES

[1] J. Ding, D.r Schmidt: Solving degree and degree of regularity for polynomial systems over finite fields. Number theory and cryptography, pp. 34–49, Lecture Notes in Comput. Sci., 8260, Springer, Heidelberg, 2013.

[2] J. Ding, C. Wolf, B. Yang: l-Invertible Cycles for Multivariate Quadratic (MQ) Public Key Cryptography.

[3] I. Luengo: DME a public key, signature and KEM system based on double exponentiation with matrix exponents. Preprint 2017. https://csrc.nist.gov/CSRC/media/Presentations/DME/images-media/dme-April2018.pdf
[4] J.C. Faugère, L. Perret.:An efficient algorithm for decomposing multivariate polynomials and its applications to cryptography. Journal of Symbolic Computation 44 (2009) 1676–1689

Departamento de Álgebra, Geometría y Topología, Facultad de Matemáticas, Universidad Complutense de Madrid. Plaza de Ciencias 3, 28040 Madrid, Spain.
*Email address*: iluengo@ucm.es