DME: A PUBLIC KEY, SIGNATURE AND KEM SYSTEM BASED ON DOUBLE EXPONENTIATION WITH MATRIX EXPONENTS

I. LUENGO[1]

The system presented here for the NIST call is a multivariate public key cryptosystem based on a new construction of the central maps, that allow the polynomials of the public key to be of an arbitrary degree. In order to get a reasonable size for the public key one has to use a small number of variables and special non-dense linear maps at both ends of the composition.

We will present the algorithms and construction of the system in general, but for the implementation we will choose parameters that give polynomials with 6 to 12 variables. We will build the central map using a vectorial exponentiation with matrix exponents as follows:

Let us take a finite field $\mathbb{F}_q$, $q = p^e$, and a matrix $A = (a_{ij}) \in \mathscr{M}_{n\times n}(\mathbb{Z}_{q-1})$ one can define a kind of exponentiation of vectors by using a monomial map G_A associated to the matrix A as follows:

$$G_A : \mathbb{F}_q^n \to \mathbb{F}_q^n : \quad G_A(x_1, \ldots, x_n) = (x_1^{a_{11}} \cdot \ldots \cdot x_n^{a_{1n}}, \ldots, x_1^{a_{n1}} \cdot \ldots \cdot x_n^{a_{nn}}). \tag{1}$$

The following two facts are easy to verify:

a) If $A, B \in \mathscr{M}_{n\times n}(\mathbb{Z}_{q-1})$ and $C = BA$, then the composition $G_C = G_B \circ G_A$.
b) If $\det(A) = \pm 1$ and the inverse matrix $A^{-1} \in \mathscr{M}_{n\times n}(\mathbb{Z})$, then G_A is invertible on $(\mathbb{F}_q \setminus \{0\})^n$ and the inverse is given by $G_{A^{-1}}$.

Note that if the matrix has r entries different from zero in a column k, then the product of r copies of $\mathbb{F}_q$ is mapped to $O = (0, \ldots, 0)$ so G_A as a map in $\mathbb{F}_q^n$ is a univariate polynomial of degree at least q^r.

This kind of maps are extensively used in Algebraic Geometry, they produce birational maps. In Projective Geometry they are also called Cremona transformations. In [2] these Cremona transformations are used to produce multivariate public key cryptosystem.

If $\det(A) \neq \pm 1$, then the monomial map is not birational, in fact one has [5] :

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

Now if we take $A \in \mathscr{M}_{n\times n}(\mathbb{Z}_{q-1})$ then:

Theorem 2. *Let $A \in \mathscr{M}_{n\times n}(\mathbb{Z}_{q-1})$ 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 $b := \det(A)^{-1} \in \mathbb{Z}_{q-1}$, $B := bAd(A)$ then $A^{-1} = B \in \mathscr{M}_{n\times n}(\mathbb{Z}_{q-1})$.*

This is easy to verify, since $b.\det(A) = 1 + \lambda(q-1)$ and if I_n is the identity matrix, then $AB = b.\det(A)I_n \mod (q-1)$.

We can use this fact to build a multivariate PKC in the standard way by entering powers of q in the matrix A. If each row has 2 entries $q^{a_{ij}}$, then after composition with two linear maps at both ends one gets a quadratic public key (see [3]). In our case we made extensive computer tests arriving to the conclusion that those systems are not safe against Gröbner basis attack for a reasonable key size, which is what happens with most multivariate PKC's.

In order to make an stronger system against algebraic cryptanalysis we will produce a system with the following design options:

- We allow the entries of the matrix A to be of the form p^a instead of $q^a (q = p^a)$, this will make the final polynomials with arbitrary degree up to q,
- the determinant $d = \det(A)$ has an expansion in base p with many non-zero digits.

These two conditions make the resulting system safe against Gröbner basis attacks, but in order to make it safe against structural attacks we propose to use two exponentials in two different intermediate fields as central maps, $\mathbb{F}_q^n$ and $\mathbb{F}_q^m$ and the resultant public key will be polynomials in $n.m$ variables with degree up to q in each variable. In the system we implemented we use the parameters $m = 3, n = 2$ and the public key F has 6 polynomials with 64 monomials each.

For convenience we denote the coordinates in $(\mathbb{F}_q)^{nm}$ as

$$\underline{x} = (x_{11}, \ldots, x_{1n}, \ldots, x_{n1}, \ldots, x_{nm}).$$

We will use a padding $H : \mathbb{F}_p^N \to \mathbb{F}_q^{nm}$ by adding $S \geq m$ random elements of $\mathbb{F}_p$ in such a way that the coordinates $x_{1n}, x_{2n}, \ldots, x_{nm}$ of $H(u) = (x_{11}, \ldots, x_{nm})$ are different from zero. The padding can be chosen in several different ways. For instance, one can add only one bit in each x_{in} and the encryption is deterministic or one can add random bits to each component x_{ij} in order to address the IND-CPA security.

The public key is $K_P = (h, \pi_0, F)$, where $F : \mathbb{F}_q^{nm} \to \mathbb{F}_q^{nm}$ is a map obtained as composition of five maps, $F = L_3 \circ G_2 \circ L_2 \circ G_1 \circ L_1$, according to the diagram:

$$\mathbb{F}_q^{nm} \xrightarrow{L_1} (\mathbb{F}_{q^n})^m \xrightarrow{G_1} (\mathbb{F}_{q^n})^m \xrightarrow{L_2} (\mathbb{F}_{q^m})^n \xrightarrow{G_2} (\mathbb{F}_{q^m})^n \xrightarrow{L_3} \mathbb{F}_q^{mn}$$
$$\mathbb{F}_q^{nm} \xrightarrow{\qquad\qquad F \qquad\qquad} \mathbb{F}_q^{mn}$$

The maps L_1, L_2 and L_3 are $\mathbb{F}_q$-linear isomorphism and L_1 satisfies that for every $x \in H(\mathbb{F}_p^N)$, $L_1(x) \in (\mathbb{F}_q^n \setminus \{0\})^m$. The map L_2 is designed to verify the condition:

$$\forall y \in (\mathbb{F}_q^n \setminus \{0\})^m,\ L_2(y) \in (\mathbb{F}_q^m \setminus \{0\})^n.$$

The maps G_1 and G_2 are monomial maps with the invertible determinant and entries powers of p. With all the above conditions it is clear that F is injective in $H(\mathbb{F}_p^N)$ and the components of F and F^{-1} are given by polynomials in $\mathbb{F}_q[x_1, \ldots, x_{mn}]$. The maps G_1 and G_2 are chosen in such a way that the polynomial F have few monomials and the polynomial F^{-1} has a huge number of monomials. For instance, for the parameters that we choose for the implementation, $(M = 3, n = 2, s = 2, t = 2)$, we get that each component of F has 64 monomials and that each component of F^{-1} has at least 2^{100} monomials.

Let's describe the five maps in detail.

The map $L_1 = \tilde{\pi_1} \circ \tilde{L_1} \circ \tilde{l}$ is obtained as composition of three linear $\mathbb{F}_q$-isomorphism according to the diagram (2).

$$\mathbb{F}_q^{nm} \xrightarrow[\sim]{} (\mathbb{F}_q^n)^m \xrightarrow{\tilde{L_1}} (\mathbb{F}_q^n)^m \xrightarrow[\sim]{\tilde{\pi}_1} (\mathbb{F}_{q^n})^m \tag{2}$$
$$\mathbb{F}_q^{nm} \xrightarrow{\qquad L_1 \qquad} (\mathbb{F}_{q^n})^m$$

The map $\tilde{\pi_1} = (\pi_1, \ldots, \pi_1)$ is defined by using an $\mathbb{F}_q$ linear isomorphism $\pi_1 : \mathbb{F}_q^n \to \mathbb{F}_{q^n}, \pi_1(v_1, \ldots, v_n) = \alpha_1 v_1 + \ldots + \alpha_n v_m$, where $\{\alpha_1, \ldots, \alpha_n\}$ is a fixed $\mathbb{F}_q$ basis of $\mathbb{F}_{q^n}$.

The isomorphism $\tilde{L_1} = (L_{11}, \ldots, L_{1m})$ is defined by its components $L_{1i} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ given by $L_{1i}(\underline{x}_i) = \underline{x}_i A_{1i}$, where $A_{1i} \in GL_n(\mathbb{F}_q)$.

The isomorphism $\tilde{l}$ is obtained by grouping the components of x in m vectors according to its index $h(x_1, \ldots, x_{nm}) = (\underline{x}_1, \ldots, \underline{x}_m))$, where $\underline{x}_i = (x_{i1}, \ldots, x_{in})$.

The $\mathbb{F}_q$-linear isomorphism $L_2 = {\tilde{\pi_1}}^{-1} \circ M \circ \tilde{L_2} \circ \tilde{\pi_2}$ is a composition according to the diagram (3).

$$(\mathbb{F}_{q^n})^m \xrightarrow[\sim]{\tilde{\pi}_1^{-1}} (\mathbb{F}_q^n)^m \xrightarrow{M} (\mathbb{F}_q^m)^n \xrightarrow{\tilde{L_2}} (\mathbb{F}_q^m)^n \xrightarrow[\sim]{\tilde{\pi}_2} (\mathbb{F}_{q^m})^n \tag{3}$$
$$(\mathbb{F}_{q^n})^m \xrightarrow{\qquad\qquad L_2 \qquad\qquad} (\mathbb{F}_{q^m})^n$$

The *mixing* isomorphism M transforms the m vectors of $\mathbb{F}_q^n$ in n vectors of $\mathbb{F}_q^m$ in such a way that the components of $\underline{x}_1, \ldots, \underline{x}_n$ are placed in the first n components of $\underline{x}'_1, \ldots, \underline{x}'_n$ and the components of $\underline{x}_{m-n+1}, \ldots, \underline{x}_m$ are placed in the last $m-n$ components of $\underline{x}'_1, \ldots, \underline{x}'_n$. For instance a way to produce such mixing is the following composition of maps

$$(\mathbb{F}_{q^m})^n \xrightarrow[\sim]{\tilde{\pi}_2} (\mathbb{F}_q^m)^n \xrightarrow{\tilde{L}_3} (\mathbb{F}_q^m)^n \xrightarrow[\sim]{} \mathbb{F}_q^{mn} \qquad (L_3) \tag{4}$$

That is, if we write the first matrix in two blocks $\binom{M_1}{M_2}$, where M_1 is given by the first n rows and M_2 is given by the mixing M the last rows the mixing map send $\binom{M_1}{M_2}$ to $(M_1, M_2^t) = (M_1, M'_2)$ Any bijective map that send the $(m-n)\times n$ entries of M_2 in the $n \times (m-n)$ entries of M'_2 will be also valid, but the final number of monomial depends on the mixing.

For instance taking $m = 4$ and $n = 2$ one can have the next two mixing:

$$\begin{pmatrix} x_{11} & x_{12} & x_{31} & x_{41} \\ x_{21} & x_{22} & x_{32} & x_{42} \end{pmatrix} \tag{5}$$

or

$$\begin{pmatrix} x_{11} & x_{12} & x_{31} & x_{32} \\ x_{21} & x_{22} & x_{41} & x_{42} \end{pmatrix} \tag{6}$$

It will be explained below how to calculate the monomials of F_i, and will be noted that (5) produces more mixing and 144 monomials whereas (6) produces less mixing but 64 monomials in each component.

This construction of the mixing map M guarantees that if $x \in (\mathbb{F}_{q^n} \setminus \{0\})^m$ then $x' \in (\mathbb{F}_{q^m} \setminus \{0\})^n$ but there is no corollation in the other direction, that is $x' \in (\mathbb{F}_{q^m} \setminus \{0\})^n$ does not imply $x \in (\mathbb{F}_{q^n} \setminus \{0\})^m$. This fact means that one can always encrypt and decrypt a message but there are messages that can not be signed.

The $\mathbb{F}_q$-linear isomorphism $L_3 = e \circ \tilde{L}_3 \circ \pi_2^{-1}$ is defined as

$$(\mathbb{F}_{q^n})^m \xrightarrow[\sim]{\pi_2^{-1}} (\mathbb{F}_q^n)^m \xrightarrow[\tilde{L}_3]{} (\mathbb{F}_q^n)^m \xrightarrow[\ell^{-1}]{} \mathbb{F}_q^{nm} \qquad (L_3) \tag{7}$$

The morphism $\tilde{L}_3$ is defined as $\tilde{L}_3 = (L_{31}, \ldots, L_{3n})$ where $L_{3j}(x'_j) = x'_j A_{3j}$, $A_{3j} \in \mathscr{M}_{n\times n}(\mathbb{F}_q)$ and $\det(A_{3j}) \neq 0$.

The main part of the design of the system is given by the two exponential maps G_1 and G_2 built with monomial maps as follows:

$$G_1(x_1,\ldots,x_m)=(x_1^{a_{11}}\cdot\ldots\cdot x_m^{a_{1m}},\ldots,x_1^{a_{m1}}\cdot\ldots\cdot x_m^{a_{mm}}),\qquad G_1:(\mathbb{F}_{q^n})^m\to(\mathbb{F}_{q^n})^m$$

where $A_1=(a_{ij})\in\mathscr{M}_{m\times m}(\mathbb{Z}_{q^n-1})$ such that $d_1'=\det(A_1)$ is prime with q^n-1 and

$$G_2(x_1',\ldots,x_n')=({x_1'}^{b_{11}}\cdot\ldots\cdot {x_n'}^{b_{1n}},\ldots,{x_1'}^{b_{n1}}\cdot\ldots\cdot {x_n'}^{b_{nn}}),\qquad G_1:(\mathbb{F}_{q^m})^n\to(\mathbb{F}_{q^m})^n$$

where $B_2=(b_{ij})\in\mathscr{M}_{n\times n}(\mathbb{Z}_{q^m-1})$ such that $d_2'=\det(B_2)$ is prime with q^m-1.

If $\underline{x}=(x_{11},\ldots,x_{nm})\in\mathbb{F}_q^{nm}$ are the initial coordinates, then the composition of the five maps L_1,G_1,L_2,G_2, and G_3 allow us to compute the components of $F(\underline{x})$ as polynomials $F_i\in\mathbb{F}_q[x_{11},\ldots,x_{nm}]$. In order to keep the number of monomials small, we choose the matrices A_1 and B_2 with the following properties:

(1) The entries of A_1 and B_2 are of the form p^a.
(2) For any two integers s and t such that the rows of A_1 have at most s non-zero entries and the rows of B_2 have at most t non-zero entries, the monomials in the F_i can be computed with the algorithm described below, resulting that the total number of monomials is $MON=(b\cdot n^s)^t$, where b depends on the mixing map M.
(3) The inverse maps G_1^{-1} and G_2^{-1} can be computed the same way from the inverse matrix of A_1 and B_2 respectively and F_1^{-1} is also a polynomial.

If the number of monomials in F^{-1} is not very big, one can get the coefficient of the polynomial by computing enough number of pairs $(x,F(x))$. To avoid this attack we take A_1 such that $d_1=\frac{1}{\det(A_1)} \mod q^n-1$ has a expansion in base p with $d_1=[K_0,\ldots,K_e]$ with at least s_1 non-vanishing digits and the same with B_2 and $d_2=\frac{1}{\det(B_2)}$ (with at least t_1 non-vanishing digits). The details of values of t_1,s_1 will be given when discussing the security of the system.

The public key of the system is $K_P=(h,\pi_0,F)$ and the private key is given by h, π_0 and the five maps $L_1,\ldots,L_3$ and their inverses that can be used to encrypt and decrypt. Given an encrypted message $z=F(\underline{x})=DM(\overline{x})$, one computes $\underline{x}=F^{-1}(z)$ and discards the random entries with the use of h.

It is possible to get the monomials of the F_i without computing the composition of the five maps as follows: we start with m lists that contain the coordinates of the $\underline{x}_i, M_{11}=[x_{11},\ldots,x_{1n}],\ldots,M_{mn}=[x_{m1},\ldots,x_{mn}]$, and we define the operations on lists: multiplication and exponentiation. If $S=[s_1,\ldots,s_m],T=[t_1,\ldots,t_m]$ then $S\cdot T=[s_i\cdot t_j]$ and $S^a=[s_i^a]$.

With these notations, one can see that the exponential G_1 produces, on each component, polynomials whose list of monomials is $N_{0k}=M_{01}^{a_{k1}}\cdot\ldots\cdot M_{0n}^{a_{kn}}$.

The mixing map M determines that in the list of monomials of each x'_k appears the list N_{0k}, joint with the list N_{0j} of the vectors that are placed at the $m-n$ last entries of x'_k. If b_k is the number of vectors adjoined to x'_k and by P_{0k} $(k = 1, \ldots, n)$ such a list, then the final list of monomials of each component after G_2 to each monomial ${x'_1}^{b_{k1}} \cdot \ldots \cdot {x'_n}^{b_{kn}}$ gives $Q_{0k} = P_{01}^{b_{k1}} \cdot \ldots \cdot P_{0n}^{b_{kn}}$.

Note that when we apply the final $\mathbb{F}_q$-linear bijection $\tilde{L}_3$, each component still has the same monomial, that means that there are n groups of m polynomials $F_{k1}, \ldots, F_{km}$ such that they have the same monomials, namely the list Q_{0k}.

It is clear that the number of monomials of Q_{0k} is at most $((1+b_k)\cdot n^s)^t$. So there are at most $(b_{\max} \cdot n^s)^t$ monomials on each component where $b_{\max} = \max_k(1 + b_k)$.

Once the list of monomials of the F_i is obtained, one gets the coefficient of each group of polynomials by evaluating the polynomials $F_{k1}, ...F_{km}$. The set of pairs $(\underline{c}, F_{ki}(\underline{c}))$ should be big enough to guarantee that the corresponding linear equations are independent. That is if $Q_k = [q_1...q_d]$ and $F_{kj} = \sum_{i=1}^{d} f_{ji} q_i(x)$ we take vectors $\underline{c}_1, \ldots, \underline{c}_R$ such that the linear equations (on the f_{ij}) $F_k(c_e) = \sum f_{ji} q_i(c_e)$ are independent and can be solved to get the coefficients of the polynomials $F_{k1}, \ldots, F_{km}$. This algorithm is implemented in the system to get the public key from the private key.

It is also possible to use this algorithm to get a fast evaluation of the $F_{ij}(\underline{c})$ to encrypt a message. If one starts with the list of the coordinates of $\underline{c}$ instead of the list of variables in the algorithm one finally obtains a list of the evaluated monomials $[q_j(\underline{c})]$. In order to evaluate the polynomials $F_{kj}(\underline{c}) = \sum_{i=1}^{d} f_{ji} q_i(\underline{c})$ one only needs to write their coefficients f_{ij} in a matrix $MF_k = (f_{ji})$ and compute a matrix multiplication $b_i(x) \cdot MF_k$.

Summary of the DME system

Fix parameters (m, n, s, t, N, S), a field $\mathbb{F}_q$ with $q = p^e$ and an $\mathbb{F}_q$-isomorphism $\pi_0 : \mathbb{F}_p^e \to \mathbb{F}_{p^e}$. The public key is $K_P = (h, \pi_0, F)$ or $K_P = (h, \pi_0, F, A_1, B_2)$ if we allow to use the fast evaluation algorithm. The private key is given by the maps L_1, G_1, L_2, G_2, L_3 defined by the matrices A_{1i}, A_{2j}, A_{3j}, the exponent matrices A_1 and B_2 and the mixing map M. The $\mathbb{F}_q$-linear isomorphisms $\pi_1 : \mathbb{F}_q^n \to \mathbb{F}_{q^n}$ and $\pi_2 : \mathbb{F}_q^m \to \mathbb{F}_{q^m}$ are not needed for encryption and can be chosen once for all users of the system or individually for each user and form part of the private key.

The exponent matrices A_1 and B_2 can be deduced from the exponents of the monomials in F_i so there is no need to hide them and can be made public in order to use them for the fast method to evaluate the polynomials of the public key.

Digital signature and KEM with the system

The system can be used to sign a message in $(\mathbb{F}_q^m \setminus \{0\})^n$ by computing $F^{-1}(z)$. Since F is not surjective onto $(\mathbb{F}_q^m \setminus \{0\})^n$ there are messages that can not be signed. One needs to add some randomness to the message. Given $z \in (\mathbb{F}_q^m \setminus \{0\})^n$ there exists $x \in F^{-1}(z)$ if $(L_3 \circ F \circ L_2)^{-1}(z) \in (\mathbb{F}_q^n \setminus \{0\})^m$, so the probability for $z \notin \mathrm{Im}(F)$ is of order $\frac{1}{q^n}$.

One can sign a message v in $(F_p)^{N_1}$, $N_1 < enm$, by padding it in a similar way that we do to encrypt a message. We need to choose a map $h_1 : \{1, \ldots, N_1\} \to \{1, \ldots, e \cdot n \cdot m\}$ and fill the entries not in $\mathrm{Im}(h_1)$. There is a difference with the encryption which is the fact that N_1 need not be fixed a priori. The signature of a message $z_0 \in (\mathbb{F}_p)^{N_1}$ is $\mathrm{sig}(z_0) = (x, z_0, h_1)$ such that there exists $x = F^{-1}(z)$. If it does not exist we padd again z_0 to get a different z. For the verification of the signature one computes $F(x) = z$ and throws away the random digits to get z_0.

Given two parties A and B, say A wants to send an encrypted message x to B, A encrypts x with the public key of B obtaining $z \in (\mathbb{F}_q)^{nm}$ that can not be padded because $N_1 = e \cdot n \cdot m$. If it is not possible to get the signature $y = (F_A)^{-1}(z)$ one can encrypt x again (because the system is not deterministic) up until one gets a message that can be signed.

The system can be used for KEM in a standard way but for KEM there is no need to use the padding. If two parties want to share a key for a symmetric system like AES they pick up a hash function and one of them A chooses a random $x \in (\mathbb{F}_q)^{nm}$ with $x_i \neq 0$ and sends $z = F_B(x)$ to B who decrypts z and both parties compute the common hash.

The setting of the system DME that is implemented in the proposal

We take $m = 3$, $n = 2$, $s = t = 2$ and $q = 2^e$. The number of monomials of each component is $(2 \cdot n^s)^t = 64$. The polynomial map of the public key is $F = (F_1, \ldots, F_6) : (\mathbb{F}_2^e)^6 \to (\mathbb{F}_2^e)^6$ where F_1, F_2, F_3 shares 64 monomials and F_4, F_5, F_6 shares another 64 monomials. For 128 bit security we propose $q = 2^{24}$ that is the message space is $(\mathbb{F}_2)^{144}$. We will justify this choice when we discuss the security in the corresponding paragraph. For the padding we can add from 3 to 16 bits. For instance if we add only 3 bits, one $'1'$ in each coordinate x_{12}, x_{22} and x_{32}; one gets a deterministic public key system. We choose to add 12 random bits, 4 bits in each coordinate, so the encryption maps are $DM : (\mathbb{F}_2)^{132} \to (\mathbb{F}_2)^{144}$

$$\mathbb{F}_2^{132} \xrightarrow{H} \mathbb{F}_{2^{24}}^6 \xrightarrow{F} \mathbb{F}_{2^{24}}^6$$

for 128 bit.

For 256 bit security we propose $q = 2^{48}$, that is, the message space is $(\mathbb{F}_2)^{288}$ with 24 random bits, that is, the encryption map is $DM : (\mathbb{F}_2)^{132} \to (\mathbb{F}_2)^{144}$.

$$\mathbb{F}_2^{264} \xrightarrow{H} \mathbb{F}_{2^{48}}^6 \xrightarrow{F} \mathbb{F}_{2^{48}}^6$$

The two implementations we present correspond to DME-$(3, 2, 48)$ (288 bits) and DME-$(3, 2, 48)$. They are optimized for any processor and we do not include the padding map, since it is not an essential part of the system from the point of view of the cost and speed of the system. The especifications of the implemetations are explained in the readme of the DME-$(3, 2, 48)$ so this can be considered as the reference implementation.

The different sizes of DME-$(3, 2, 48)$ system are: Public key 2304 bytes. Secret key 288 bytes. Plain text an encrypted message size 36 bytes (288 bits). If ones use the padding it is necessary to substract between 3 and 32 bits to the plain text size.

The different sizes of DME-$(3, 2, 24)$ system are: Public key 1152 bytes. Secret key 144 bytes. Plain text and encrypted message size 18 bytes (144 bits). If one use the padding it is necessary to substract between 3 and 16 bits to the plain text size.

The timing for this implementation can be read at the end of the file `kat.txt` created after the compilation. For that give the average of the timings for 1000 tests. We include a file in the documentation with the timings for an Intel i7.

The API consists of the follwing four functions (see `api.h`):

```
int dme_skey_to_pkey(char *pkey, const char *skey);
int dme_encrypt(char *ct, const char *pt, const char *pkey);
int dme_decrypt(char *pt, const char *ct, const char *skey);
int dme_encrypt_with_skey(char *ct, const char *pt, const char *skey);
```

`skey_to_pkey` This function computes the public key from the secret key that consists of a list of the coefficients of the matrix that define the three linear isomorphisms. They are randomly choosen and the program check if the determinat are non zero.

`encrypt_with_skey` The function `encrypt_with_skey(ct, pt, skey)` does the same as encrypt, but using a secret key skey instead of a public key pkey, that is the maps L_1, G_1, L_2, G_2, and L_3. It is used to generate enough pairs $(x, F(x))$ to compute the coefficient of the monomials of F and also for testing purposes.

Security of proposed setting

We will discuss here the security of DME-$(3, 2, 48)$ that produces a 288 bits message. We claim that it reaches the level 5 of security and that it is as difficult

to break as AES-256 by classical attacks. The same arguments apply to DME-$(3, 2, 24)$ (128 bits) or DME-$(3, 2, 36)$ (192 bits). We have estimated the security against Gröbner basis attackd and the other standard attacks against multivariate systems and some standard structural attacks. Since the resulting polynomials are very structured it is reasonable to imagine that there are other structural attacks, but we were unable to study this question due to the lack of time. In fact the system DME is very recent, it was finished in November, 2017 and needs further studies, but since the NIST rules do not allow to send proposals after the deadline, we will pursue the study of structural attacks along with the rest of the PQCrypto community.

Algebraic attacks. We have made many computer experiments using MAGMA and its implementation of the Faugere algorithm F4 with the public key polynomials DME-$(3, 2, e)$ for $2 < e < 9$. Our estimations are partial because F4 can find the Gröbner only until $e = 5$ (30 bits). For $e = 6$ or higher F4 can not find the Gröbner basis because it exhausts the available RAM memory (512GB). Our conclusions are:

- The highest degree of the steps in the algorithm is at least q. If one computes the number of operations for the Gröbner Basis algorithm at this degree it is much bigger than q^6, but of course the system it is not generic.
- The number of monomials involved in the computations is at least q^4. One can also estimate the size of the matrix involved in the algorithm. Also note that given the matrix of exponents, the number of solutions of the system over the algebraic closure of F_q can be made at least q^3 and this affects the size of the intermediate computations.
- It is safe to estimate that the number of binary operations in the Gröbner basis algorithm for DME-$(3, 2, 2^e)$ is at least 2^e.

One can fix r of the variables and solve the system for the remaining $6 - r$ variables, but the bound on the number of operations is basically the same. Let us assume that $q = 2^{48}$, if we fix 5 variables we get 6 polynomials in one variable with degree that we may assume is at least 2^{24}. We can use the Euclid's algorithm but since we have to try $2^{48\cdot 5} = 2^{240}$ times, it is clear that the number of binary operations at least 2^{256}. For $r = 4, 3, 2, 1$ one has to solve 2^{48r} systems in r variables and the estimates are similar as above.

Another possible attack is to represent the map F as a polynomial P in one variable over $\mathbb{F}_{q^6}$. As F sends the coordinate 2-planes to 0 the degree of P is at least q^2 and it will cost more than q^6 bit operations to get P.

Another standard attack is to represent F as a polynomial Q map over $\mathbb{F}_2$. Each of the monomials of F involves 4 variables with exponents 2^a. The polynomials in Q will have degree up to 4. The total number of monomials of degree up to 4 in 288 variables is of the order of 2^{28}. There are no sharp estimates of the Gröbner basis complexity for quartic polynomials, this has to be closely investigated but I believe that the hypothesis that the complexity of solving the resulting system of 288 equations is at least 2^{256} is realistic. In fact this attack and the next one are the main reason to take two exponentiations. Let us remark that there is a straightforward modification of the parameter choice that makes the system more secure to this attack. If we allow two of the rows of the matrix A_1 to have 3 non-zero entries, then the total number of final monomials is 144 with 6 variables each.

As we have mentioned earlier the inverse of F is also polynomial but if we take the matrices A_1 and B_2 such that the inverse of their determinants has a big binary weight, then the number of the monomials of F^{-1} will be exponentially high. For instance for $q = 2^{48}$ we may assume that the inverse of the $\det(B_2)$ mod (q^3-1) has at least binary weight 40. By applying the algorithm to compute the monomials to the list of $64 = 2^6$ monomials, we get a list of $(2^6)^{40}$. As the total number of monomials on 6 variables of degree at most q is around q^6 many of the monomials of the list will be equal, but we can assume that for a generic matrix the total number will be more than 2^{40}. After that we have to apply G_{A_1} and we can estimate that the final list will have at least 2^{100}. This is one of the strongest properties of the system that produce invertible maps with 64 monomials but the inverse can have more than 2^{100}.

Probably the most serious attack to this kind of system is a structural attack. For instance we can set the entries of the matrices of L_1, L_2, L_3 as variables and compute the coefficients of the monomials. If we want to solve the resulting equations we will get $6 \cdot 64$ equations in 48 variables of degree up to q. This seems a hopeless task but as the resulting coefficients are very structured, may be there is some feasible attack.

Advantages and disadvantage of the DME system. The main disadvantage of the system is that is very new, probably secure against standard attacks, but its resistance against possible new attacks is unclear.

Advantages.

- The parameters are very easy to adapt with modest decrease of the performance of the system. For instance we can increase e, let say $q = 2^{60}$ or to take $m = 4, 5, 6$ or increase the number of non-zero entries in A_1.

- The system is mathematically very simple. Only a precise number of multiplication and exponentiation are used. For this reason it is easy to protect the system against timing side channel attacks.
- The system can encrypt and decrypt without failures.
- Digital signatures can fail. The probability of failure is $1/q^2$, but this can be fixed with a small padding.
- KEM is straightforward and do not need padding.
- Encryption and signature verification can be sped up using a logarithm table, if e is not too big, that makes storage and looking at the table impractical. This property can be very useful for servers that have to verify thousands of signatures per second.

References

[1] Jintai Ding, Dieter 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] Ding, C. Wolf, B. *Yangl-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.

[4] J.C. Faugère, A. Joux. *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Groebner Bases.* CRYPTO 2003, LNCS vol. 2729, pp. 44-60. Springer, 2003.

[5] Tadao Oda,Convex Bodies and Algebraic Geometry Ergebnisse der Mathematik 3. Springer 1988

Departament of Algebra, Geometry and Topology, Complutense University of Madrid, Plaza de las Ciencias s/n, Ciudad Universitaria, 28040 Madrid, SPAIN

E-mail address: iluengo@ucm.es