Once one gets the list of monomials of the F_i one gets the coefficient of each group of polynomials. $F_{K1} \cdots F_{Km}$ by evaluating on a set of pairs $(x, F_{Ki}(x))$ ~~for~~ big enough ~~for the c~~ to garantee that the corresponding linear equations are independents. That is if $Q_{oK} = [q_1 \cdots q_d]$ and

$$F_{Kj} = \sum_{i=1}^{d} f_{ji}\, q_i(x).$$

we take vector $c_1 \cdots c_R$ such that the linear equation ~~on the f_{ij}~~ $F_K(c_e) = \sum f_{ji}\, q_i(c_e)$ are independent., and can be resolved to get the polynomials $F_{K1} \cdots F_{Km}$.

This algorithm is ~~the one~~ implemented in the system to get the public key from the private key.

It is possible to use this algorithm to ~~evaluate very fast the~~ get a fast evaluation of $F_{ij}(c)$ ~~for~~ a to get the encripted message. ~~that,~~

If we start with the list of coordinates of $\underline{c}$ instead of the list of ~~indeterm~~ variables in the algorithm (*), we get at ~~the~~ the end the list of evaluated monomials $[q_i(\underline{c})]$. In order to get the evaluated polynomials $F_{kj}(\underline{c}) = \sum_{i=1}^{d} f_{ji} q_i(\underline{c})$ one need only to write the f_{ji} in a matrix $MF_k = (f_{ji})$ and compute a matrix multiplication $b_k(X) \cdot MF_k$.

~~De~~Resumée of the system.

Fix parameters (m, n, t_1, t_2, N, s), a field $q = p^e$, and an ~~isom~~ $\mathbb{F}_p$ isomorphism $\pi_0 : \mathbb{F}_p^e \to \mathbb{F}_{p^e}$.

The public key is $K_p \in (h, \pi_0, F)$ and the private key are the maps L_1, G_1, L_2, G_2, L_3 defined by the matrices

$\{A_{1i}\}, \{A_{2j}\}, \{A_{3j}\}$, the exponents matrices A_1 and B_2 and the mixing map M_0 the $\mathbb{F}_q$-linear isomorphism.

$\pi_1 : \mathbb{F}_q^n \longrightarrow \mathbb{F}_{q^n}$, $\pi_2 : \mathbb{F}_q^m \to \mathbb{F}_{q^m}$

are not needed for encryptation an can be choosen once for all users of the system or individualy for each user and foorm part of the private key.

The exponent matrices A_1 and B_2 can be deduced from the exponents of the F_i so there is no need to hide them and can be made public. If one use the fast method to ~~encript~~ evaluate the monomials ~~the public key~~ one should ad the matrix. A_1 and B_2 to the public key $k_P = (h, \pi_0, F)$

~~It~~ the sistem can be used for singuture of a message $z \in \mathbb{F}_q^{nm}$, but as $F : \mathbb{F}_q^{nm} \longrightarrow \mathbb{F}_q^{nm}$ is not surjective one need to add some randomes to the message. ~~The image $F(\mathbb{F}_q^N) \subset (\mathbb{F}_q^m - 304)^n$ but.~~

~~Given $z \in (\mathbb{F}_q^m - 304)^n$ $\exists x$ such that $F(x) = z$ if~~

~~The map inverting M^{-1}~~

~~the map M^{-1}~~

Given a $z \in (\mathbb{F}_{q^m - 104})^n$ there exist $x \in F^{-1}(z)$ if $(L_3 \cdot G \cdot L_2)^{-1} \in (\mathbb{F}_{q^m - 104})^m$, so the probability of the $\mathbb{E} \notin \mathrm{Im}(F)$ is of the order $\frac{1}{q^m}$.

One can sign a message z_1 in $\mathbb{F}_p^{N_1}$ with $N_1 < n m$ by padding it with random entries, by using a map $h_1 : \mathbb{F}_1 - N_1 \mathbb{F}_1 \to \mathbb{F}_1 - n m \mathbb{F}_1$, In this case the length of need not to be fixed. So the signature of a message $z_0 \in \mathbb{F}_p^{N_1}$ is

$$\mathrm{sig}(z_0) = (x, z_0, h_1) \quad \text{with} \quad x = F^{-1}(z).$$

the verification of the signature consist on computing $z = F(x)$ and throw away the random entries of z with the help of h_1 to get z_1.

If two parties A and B can interchange signed

If two parties A and B want to encript and signe a message $x \in \mathbb{F}_p^N$ for B, then compute $DM_B(x)$. then he has to compute $F_A^{-1}(DM_B(x))$. but in this case $DM_B(x) \in \mathbb{F}_p^{n m}$ and can not be padded. As the encriptation is not deterministic one can encript again x_1 until get a z_1 such that a signature $F^{-1}(z_1)$ exist.

The system can be used for KEM in an standard way but in this case ther is no need to use the padding. If two parties want to share a key for a symetric system like AES they choose a Hash function and A choose a random $x \in \mathbb{F}_q^{nm}$, encript. it with the map $F_B(x) = z$, the other partie B decrip z to get x and both parties Shared $H_S(x)$.

The setting that we implemented are as follow.

We take $m=3$, $n=2$, $s=t=2$. and $q=2^e$

The polinomial map $F: \mathbb{F}_q^6 \longrightarrow \mathbb{F}_q^6$.

where $F = (F_1 - - F_6)$ and F_1, F_2, F_3 share the same monomials and also F_4, F_5, F_6.

The number of monomials is at mos $(2n^s)^6 = 64$.

To get 128-bit security we set $q=2^{24}$ and for 256-bits we set $q=2^{48}$. We will justfie this choices in the security paragraph. For the padding we put 4 random bits in x_{12}, x_{22} an x_{32} for 128-bits security and 8 bits in x_{12}, x_{22} and x_{32} for 256 bits security.

So the encriptation maps are.

$$\mathbb{F}_2^{132} \xrightarrow{H} \mathbb{F}_{2^{24}}^{6} \xrightarrow{F} \mathbb{F}_{2^{24}}^{6}$$ for 128 bit.

an $\mathbb{F}_2^{264} \xrightarrow{H} \mathbb{F}_2^{144} \xrightarrow{} \mathbb{F}_{2^{48}}^{6} \xrightarrow{F} \mathbb{F}_{2^{48}}^{6}$ for 256 bit

~~In~~ fact for 128-bits security we need ~~that the~~ $nm \geq 144$ thats why we use $q = 2^{24}$, but the number of random digit s can change without affect the security. For instan we can have $s = 3$ and put 1 in the zone portion in ~~x_{21}~~ x_{12}, x_{22} and x_{32}. Then $H: \mathbb{F}_2^{141} \to \mathbb{F}_2^{144}$ will give a deterministic encriptation.