Algoritmo LUOV

26 de febrero de 2020

María Victoria Granados Pozo

1. Cuerpos finitos

Se trabajará con el cuerpo finito de 128 elementos, GF(2^7), que es un cuerpo extendido de GF(2) que corresponde con el cociente

$$\mathrm{GF}(128) = \frac{\mathrm{GF}(2)[x]}{\langle x^7 + x + 1 \rangle} \tag{1}$$

Además el orden del cuerpo de las unidades es 127, que es primo entonces todo elemento del cuerpo distinto de 1 es un elemento primitivo, es decir, un generador.

La tabla 1 muestra una representación de los elementos no nulos del cuerpo. En la implementación se ha utilizado la representación como cadena de bits, puesto que a la hora de trabajar es más fácil con una cadeba de bits que con los polinomios.

Tabla 1: Representación de los elementos no nulos del cuerpo finito de 2^7 elementos

Polinomio	**Bits**	$\log_a$
1	[0, 0, 0, 0, 0, 0, 1]	0
a	[0, 0, 0, 0, 0, 1, 0]	1
a^2	[0, 0, 0, 0, 1, 0, 0]	2
$\vdots$	$\vdots$	$\vdots$
$a^6+a^5+a^4+1$	[1, 1, 1, 0, 0, 0, 1]	124
a^6+a^5+1	[1, 1, 0, 0, 0, 0, 1]	125
a^6+1	[1, 0, 0, 0, 0, 0, 1]	126

La implementación del cuerpo finito de 2^7 elementos no se ha realizado de forma genérica sino para que sea específica para el algoritmo *LUOV*, de esta forma es mucho más sencillo implementar la aritmética del cuerpo. Para la suma en $\mathbb{Z}_2$ sólo tenemos que fijarnos que es lo mismo que el operador lógico *XOR*, mientras que para el producto, al encontrarmos en un cuerpo como un orden pequeño, se usarán unas tablas que contienen las correspondencias entre los elementos no nulos del cuerpo y sus logaritmos en base a, por lo que el producto se convierte en una suma módulo 127.

2. Parámetros y fórmula

Para empezar indicamos los parámetros que serán de utilidad para entender el algoritmo.

- r: Grado del cuerpo extendido, $\mathbb{F}_2 \subset \mathbb{F}_{2^r}$.
- x: Vector de n componentes, denominando a las primeras v componentes $x_1, \cdots, x_v$ vinagre y al resto aceites.

- m: Tamaño de la clave pública, además del número de variables de aceite.
- v: Número de variables vinagre.

$\mathcal{P} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^m$, esta función se puede descomponer como $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$, donde $\mathcal{T} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^n$ es invertible, y $\mathcal{F} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^m$ siendo sus m componentes de la forma:

$$f_k(x) = \sum_{i=1}^{v} \sum_{j=i}^{n} \alpha_{i,j,k} x_i x_j + \sum_{i=1}^{n} \beta_{i,k} x_i \tag{2}$$

donde $\alpha_{i,j,k}$ y $\beta_{i,k}$ se toman aleatoriamente en $\mathbb{F}_2$ siendo α una matriz triangular superior. De esta manera será más eficiente y no afectará a la seguridad del algoritmo.

3. Generación de la clave privada

La clave privada está formada por $\alpha_{i,j,k}$ y $\beta_{i,k}$ que son valores del cuerpo $\mathbb{F}_2$, tomados de forma aleatoria.

4. Generación de la clave pública

Generaremos una clave pública partiendo de la clave privada $\alpha_{i,j,k}$ y $\beta_{i,k}$.
Para entenderlo mejor ponemos las m ecuaciones (2) en forma matricial,

$$f_k(x) = x^v \ [\alpha_{i,j,k}] \ (x^v, x^m)' + [\beta_{i,k}] \ (x^v, x^m)' \tag{3}$$

siendo $[\alpha_{i,j,k}]$ y $[\beta_{i,k}]$ son las representaciones matriciales de $\alpha_{i,j,k}$ y $\beta_{i,k}$, x^v los vinagres y x^m los aceites, así x se puede expresar como $(x^v, x^m)'$.
Conociendo los valores de la clave privada α y β, tomando de forma aleatoria los del vinagre x^v, los cuales pasaremos a denominarlos como a^v, y tomando los m primeros bits del hash del mensaje h_k podemos generar la clave pública.
Hacemos el cambio de notación $A_k = a^v \ [\alpha_{i,j,k}] = (A_k^v, A_k^m)$, lo sustituimos en la ecuación (3) y despejamos los aceites.

$$h_k = A_k^v \ (a^v)' + A_k^m \ (x^m)' + \beta_k^v \ (a^v)' + \beta_k^m \ (x^m)' + \gamma_k \tag{4}$$

$$(A_k^m + \beta_k^m)(x^m)' = h_k - (A_k^v + \beta_k^v)(a^v)' - \gamma_k \tag{5}$$

$$(x^m)' = (A_k^m + \beta_k^m)^{-1}(h_k - (A_k^v + \beta_k^v)(a^v)' - \gamma_k) \tag{6}$$

Si $(A_k^m + \beta_k^m)$ fuese una matriz singular, entonces se tomarían otros valores de vinagres.

Para generar la clave pública necesitamos incluir una nueva matriz T, donde $T \cdot s' = x'$. Incluimos esta matriz T para aumentar la seguridad del algoritmo y así sea más complejo calcular la función inversa $\mathcal{P}$

$$T = \left[\begin{array}{c|c} I_v & T_{vxm} \\ \hline 0 & I_m \end{array}\right] \tag{7}$$

Despejando x, obtenemos:

$$x = s \cdot T' = s \left[\begin{array}{c|c} I_v & 0 \\ \hline T'_{vxm} & I_m \end{array}\right] = [s^v, s^m] \left[\begin{array}{c|c} I_v & 0 \\ \hline T'_{vxm} & I_m \end{array}\right] = (s^v + s^m T'_{vxm}, s^m) \tag{8}$$

Sustituimos en (3):

$$f_k(x) = s \left[\frac{I_v}{T'_{vxm}}\right] [\alpha_{i,j,k}]_{\substack{1<i<v \\ i<j<n}} \; T \; s' + [\beta_{j,k}]_{1<j<n} \; T \; s' \tag{9}$$

donde $k \in \{1, ..., m\}$

Así obtenemos las claves públicas definidas para cada k

- $\alpha_{pub_k} = \left[\frac{I_v}{T'_{vxm}}\right] [\alpha_{i,j,k}]_{\substack{1<i<v \\ i<j<n}} \; T$
- $\beta_{pub_k} = [\beta_{j,k}]_{1<j<n} \; T$

5. Algoritmo de firma

Por la definición de x obtenemos la firma s como

$$s = x \cdot T'^{-1} \tag{10}$$

donde $x = (x^v, x^m)$ con x^v son los vinagres aleatorios y x^m los aceites que hemos calculado en la ecuación (6).

6. Algoritmo de verificación

Para comprobar que el mensaje es correcto y que no ha sufrido ninguna transformación durante el envío del mismo, se tiene que cumplir la igualdad (11).

$$h_k = \; \alpha_{pub_k} \; s' + \beta_{pub_k} \; s' \tag{11}$$