Explorando Criptografias MPKC

Guilherme Cappelli Bouzon de Amorim Cruz - DRE: 121170269

1 Introdução

Em 1997, a NIST - Instituto Nacional de Padrões e Tecnologia - iniciou um processo para selecionar uma criptografia de chave simétrica a ser usada na proteção de dados sensíveis do governo americano que estavam sob a responsabilidade da NIST. O algoritmo selecionado foi o *Rijndael*, elaborado por dois pesquisadores belgas, que foi então nomeado para ser o *Advanced Encryption Standard* - AES - que conhecemos hoje por estar presente de forma 'built-in' nos nossos web browsers.

Com o advento da Computação Quântica e o Algoritmo de Shor, muitos sistemas de criptografias, que antes eram visto como seguros (se implementados e utilizados corretamente), podem ser 'bypassados' em tempo polinomial nos computadores quânticos, como ECC. Isso levou a NIST a criar um novo processo seletivo, agora em busca de algoritmos resistentes à ataques na computação quântica e um dos candidatos é a MPKC.

2 Criptografia de Chave Pública Multivariável

Criptografias de Chave Pública Multivariável (MPKC) são sistemas que adotam mapeamentos de polinômios em várias variáveis sobre corpos finitos como chave pública. Vamos começar definindo as estruturas algébricas necessárias para entender a relação de bases de Gröbner com MPKC.

Definição 1.1

Um corpo finito $\mathbb{F}$ é um corpo cuja ordem $q \in \mathbb{N}$ é finita, para q um número primo. Se a ordem for q^m, para $m > 1$, dizemos que $\mathbb{F}$ é uma extensão de corpo.

A segurança desses esquemas é baseada na dificuldade de encontrar soluções de um sistema de equações de polinômios em várias variáveis, conhecido como *Multivariate Polynomial Problem*. Se esse sistema é composto por polinômios de grau 2, podemos enunciar o *MQ Problem*.

Definição 1.2

Sejam $\mathbb{F}[x_1, \ldots, x_n]$ um anel de polinômios sobre um corpo finito $\mathbb{F}$ e os polinômios de grau 2 $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$. O sistema de equações dos polinômios $f_1, \ldots, f_m$ que constrói o problema MQ é:

$$P = \begin{cases} f_1(x_1, \ldots, x_n) = \sum_{1 \leq i \leq j \leq n} {a_{ij}}^{(1)} \cdot x_i \cdot x_j + \sum_{1 \leq i \leq n} {b_i}^{(1)} \cdot x_i + c^{(1)} = d_1 \\ f_2(x_1, \ldots, x_n) = \sum_{1 \leq i \leq j \leq n} {a_{ij}}^{(2)} \cdot x_i \cdot x_j + \sum_{1 \leq i \leq n} {b_i}^{(2)} \cdot x_i + c^{(2)} = d_2 \\ \quad \vdots \\ f_m(x_1, \ldots, x_n) = \sum_{1 \leq i \leq j \leq n} {a_{ij}}^{(m)} \cdot x_i \cdot x_j + \sum_{1 \leq i \leq n} {b_i}^{(m)} \cdot x_i + c^{(m)} = d_m \end{cases}$$

onde todos $a_{ij}, b_{ij}, c \in \mathbb{F}$ e P é uma coleção de m polinômios quadráticos $(f_1, \ldots, f_m)$ em n variáveis.

Em outras palavras, precisamos encontrar $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}^n$ que zeram $f_i - d_i, \forall 1 \leq i \leq m$, que é a variedade afim $\mathbb{V}(\langle f_1 - d_1, \ldots, f_m - d_m \rangle)$.

Como estamos à procura de uma *trapdoor* para nossa MPKC, este problema pode se tornar bastante interessante, porque é NP-difícil e não conhecemos nenhum algoritmo que é capaz de solucioná-lo em tempo polinomial, independente se o corpo finito sob os quais os polinômios estão for GF(2).

Para fins didáticos, segue a definição de mapeamento afim abaixo, que nos será muito importante para a descrição da Criptografia de Chen et. al.

Definição 1.3

Um mapeamento afim $S := (S_{matriz}, s_{vetor}) : \mathbb{F}^m \to \mathbb{F}^m$ é definida como uma transformação linear S_{matrix} seguida de uma translação s_{vetor}.

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = S_{matriz} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} + s_{vetor}$$

3 Proteções Contra Ataques de Linearização

Jacques Patarin propôs uma alteração no sistema criptográfico HFE, que outros criptólogos concordaram em ser mais que apenas um 'patch' para ataque de linearização. Assim, sistemas de criptografia MPKC costumam adotar esses modificadores para adicionar tal camada de segurança.

3.1 Modificador 'Menos'

Neste modificação alguns dos polinômios de P passam a ser escondidos na chave pública de maneira a não alterar a capacidade de encriptar e decriptar mensagens. Apesar de acrescentar uma camada de segurança fundamental contra ataques como o *ataque de Patarin*, o algoritmo se torna mais custoso.

No artigo original, não é explicitado um valor ótimo para a, mas é intuitivo que para não afetar a resolução do sistema, a não pode ser arbitrariamente grande comparado com n, porque para estrair x de y, é necessário computar q^a possibilidades para y. Então, a pode chegar a valores próximos de $\lfloor \log_2(n) \rfloor$.

3.2 Modificador 'Mais'

Já no modificador 'Mais', serão acrescentados polinômios sobre várias variáveis de grau 2 de maneira aleatória à coleção P'. Vale destacar, que o único objetivo desse modificador é que, sem ele, o mapeamento central $F : \mathbb{F}^n \to \mathbb{F}^m$, que veremos na seção seguinte, não é injetivo. Isso ocorre, porque quando $m < n$, existem múltiplos vetores $\alpha_1, \alpha_2 \in \mathbb{F}^n$ tais que $F(\alpha_1) = F(\alpha_2)$, isto é, existem variáveis livres nesse sistema.

Nesse sentido, é necessário entender qual o limite de polinômios quadráticos que podemos inserir em F sem ocasionar problemas. No trabalho de Chen et. al, foi detalhado que existe um *blow-up factor* para erros na decriptação de uma cifra.

$$\textit{blow-up factor} = \frac{m}{n} = \frac{n + 1 - a + s}{n} = 1 + \frac{s + 1 - a}{n} \tag{1}$$

Se fixarmos um limite razoável para a probabilidade de falha da decriptação 2^B, podemos explicitar o valor de s para o qual atingimos esse limite como $s = \frac{B}{2\log(q)+a-1}$. Então,

$$\textit{blow-up factor} = \frac{B}{2\log(q) + n} \tag{2}$$

4 Criptografia de Chen et. al

Nesta seção, vamos fixar a notação $\mathbb{F}[x_1, \ldots, x_n]$ para um anel de polinômios sobre um corpo finito de ordem q. Suponha que a característica de $\mathbb{F}$ não seja 2.

Sejam $C \in \mathbb{F}^{(n+1)\times n}$ uma matriz e os polinômios $f_1, \ldots, f_{n+1} \in \mathbb{F}[x_1, \ldots, x_n]$, onde $n \in \mathbb{Z}^+$. Estruturamos esse sistema de forma semelhante à definição (1.2).

$$\begin{cases} f_1 = (x_1 - c_{1,1})^2 + \cdots + (x_n - c_{1,n})^2, \\ f_2 = (x_1 - c_{2,1})^2 + \cdots + (x_n - c_{2,n})^2 \\ \quad \vdots \\ f_{n+1} = (x_n - c_{n+1,1})^2 + \cdots + (x_n - c_{n+1,n})^2 \end{cases} \tag{3}$$

Queremos solucionar esse sistema para $y = (y_1, \ldots, y_{n+1}) \in \mathbb{F}^{n+1}$, então se tomarmos a diferença entre equações de f_{i+1} e f_i, para todo $1 \leq i \leq n$, obtemos algo na forma de

$$\begin{cases} f_2 - f_1 = y_2 - y_1, \\ \quad \vdots \\ f_{i+1} - f_i = y_{i+1} - y_i \\ \quad \vdots \\ f_{n+1} - f_n = y_{n+1} - y_n \end{cases} \tag{4}$$

Podemos perceber o que acontece nesse sistema se tomarmos dois elementos arbitrários f_{i+1} e f_i e expandirmos o produto notável. Como $f_i = (x_1^2 - 2 \cdot x_1 \cdot c_{i,1} + c_{i,1}^2) + \cdots + (x_n^2 - 2 \cdot x_n \cdot c_{i,n} + c_{i,n}^2)$, podemos agrupar esses termos pelo grau total.

$$f_i = (\sum_{i=0}^{n} x_i^2) - 2 \cdot (\sum_{j=0}^{n} x_j \cdot c_{i,j}) + (\sum_{k=0}^{n} c_{i,k}^2)$$

O que nos leva a

$$\begin{aligned} f_{i+1} - f_i &= (\sum_{i=0}^{n} x_i^2 - 2 \cdot \sum_{j=0}^{n} x_j \cdot c_{i+1,j} + \sum_{k=0}^{n} c_{i+1,k}^2) - (\sum_{i=0}^{n} x_i^2 - 2 \cdot \sum_{j=0}^{n} x_j \cdot c_{i,j} + \sum_{k=0}^{n} c_{i,k}^2) \\ &= (\sum_{i=0}^{n} x_i^2 - \sum_{i=0}^{n} x_i^2) + (2 \cdot \sum_{j=0}^{n} x_j \cdot c_{i,j} - 2 \cdot \sum_{j=0}^{n} x_j \cdot c_{i+1,j}) + (\sum_{k=0}^{n} c_{i+1,k}^2 - c_{i,k}^2) \\ &= 2 \cdot (\sum_{j=0}^{n} x_j \cdot (c_{i,j} - c_{i+1,j})) + (\sum_{k=0}^{n} c_{i+1,k}^2 - c_{i,k}^2) \end{aligned} \tag{5}$$

Contudo, sabemos do sistema inicial que $f_{i+1} - f_i = y_{i+1} - y_i$. Então, se definirmos a matriz $C' := (2(c_{i,j} - c_{i+1,j}))_{i,j} \in \mathbb{F}^{n\times n}$, obtemos o seguinte sistema linear

$$\begin{bmatrix} y_2 - y_1 \\ y_3 - y_2 \\ \vdots \\ y_{n+1} - y_n \end{bmatrix} = C' \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n} (c_{2,j}^2 - c_{1,j}^2) \\ \sum_{j=1}^{n} (c_{3,j}^2 - c_{2,j}^2) \\ \vdots \\ \sum_{j=1}^{n} (c_{n+1,j}^2 - c_{n,j}^2) \end{bmatrix} \tag{6}$$

Se existe uma inversa para C', isso significa que existe solução para o sistema via Eliminação Gaussiana, que é uma característica bastante desejável.

Para a implementação dos modificadores da seção 3, escolha (i) os primeiros $(n+1-a)$ polinômios de $f_1, \ldots, f_{n+1}$, chamaremo-os F'_α e (ii) $g_1, \ldots, g_s \in \mathbb{F}[x_1, \ldots, x_n]$ polinômios de grau 2 de forma aleatória,

denote o conjunto desses como G, e (iii) faça a união $F'_\alpha \cup G$ no mapeamento central F, que será a componente mais importante da criptografia de Chen et. al, para o método 'Menos' e 'Mais', respectivamente, onde $a, s \in \mathbb{Z}^+$ e como é fácil perceber $m = n + 1 - a + s$.

$$F = (f_1, f_2, \ldots, f_{n+1-a}, g_1, \ldots, g_s) \in \mathbb{F}^m \tag{7}$$

Considere dois mapeamentos afins invertíveis $S : \mathbb{F}^n \to \mathbb{F}^n, T : \mathbb{F}^m \to \mathbb{F}^m$ e denote a *chave pública* do criptosistema como $P : \mathbb{F}^n \to \mathbb{F}^m$ e calcule-a com $P = S \circ F \circ T$. Enquanto isso, a *chave privada* será a quádrupla (C, G, T, S). Vale destacar, que a composição da chave pública se dá pelo algoritmo abaixo:

Algoritmo 1

```
# Computando a Chave Publica P
Entrada:
    S,T -> Mapeamentos afins invertiveis
    F -> Funcao Regular
    x -> vetor das n variaveis
Saida:
    P -> Chave Publica de m polinomios
1. v = S * x + s # Calcule a transformacao linear de 'S' com o vetor 'x' e some o
   vetor de translacao 's'

2. v1 = F(v) # Tome o resultado anterior 'v' e substitua-o na funcao F

3. P = T * v1 + t # Por fim, realize a transformacao como foi feito no passo 1.
```

Devido à adoção dos métodos 'Mais' e 'Menos' na criptografia, o custo para decriptar um 'ciphertext' aumenta, embora encriptar seja extremamente barato. Seja pt uma mensagem a ser encriptada, esta deve ser convertida para uma cadeia numérica, seja transformando string para bytes ou com o auxílio de algum padding, denote o resultado dessa transformação por $\alpha \in \mathbb{F}^n$.

O processo de encriptar é feito facilmente através de $y = P(\alpha) \in \mathbb{F}^m$. Já decriptar um *ciphertext* é um pouco mais trabalhoso, visto que exige realizarmos as operações inversas recursivamente de $\alpha = T^{-1} \circ F^{-1} \circ S^{-1}$.

Seja $P(\alpha) = y \in \mathbb{F}^m$ o *ciphertext* a ser decriptado. Por construção, os dois mapeamentos afins S e T são invertíveis, logo $\beta = (\beta_1, \ldots, \beta_m) := T^{-1} \circ y$. Por conta dos modificadores, é necessário iterar sobre elementos de $\mathbb{F}[x_1, \ldots, x_n]^a$. Denote $\beta' \in \mathbb{F}[x_1, \ldots, x_n]^a$ para o laço atual.

$$\begin{cases} f_1(x_1, \ldots, x_n) = \beta_1 \\ \vdots \\ f_{n+1-a}(x_1, \ldots, x_n) = \beta_{n+1-a} \\ f_{n+1-a+1}(x_1, \ldots, x_n) = \beta'_1 \\ \vdots \\ f_{n+1}(x_1, \ldots, x_n) = \beta'_a \end{cases} \tag{8}$$

Tome $\gamma = (\gamma_1, \ldots, \gamma_n)$ como a solução do sistema de polinômios quadráticos descrito e verifique se $g_1(\gamma) = \beta_{n+1-a+1}, \ldots, g_s = \beta_m$. Se não o for, escolha outro $\beta' \in \mathbb{F}[x_1, \ldots, x_n]^a$, caso contrário, damos prosseguimento à decriptação e calculamos $\alpha = S^{-1} \circ \gamma$, que é a resposta que procuramos para $P(\alpha) = y$.

5 Bases de Gröbner: Um Ataque Algébrico

Para começarmos essa seção, vamos responder como determinamos se um elemento pertence a um ideal $I \in \mathbb{F}[x_1, \ldots, x_n]$. Sabemos que, se um elemento pode ser escrito como combinação linear dos geradores do ideal I, então ele pertence ao ideal. Em outras palavras, se o resto da divisão de $f \in \mathbb{F}[x_1, \ldots, x_n]$ pelos geradores de I for zero, então ele pertence, mas isso pode não ser verdade em alguns casos quando estamos lidando com várias variáveis.

Isso acontece, porque ao mudar a ordem dos divisores, o resultado pode ser diferente, e portanto, o resto pode não resultar zero. Uma maneira de contornar isso é encontrar uma base nova G que gere I cujo resto da divisão é único.

Definição 5.1

Sejam G um subconjunto do ideal I de $\mathbb{F}[x_1, \ldots, x_n]$ onde $\text{in}(I)$ é gerado pelos $\text{in}(g)$, para cada $g \in G$. Esse conjunto de geradores $G = \{g_1, \ldots, g_m\}$ será chamado de Base de Grobner.

Proposição 5.2

Seja I um ideal de $\mathbb{F}[x_1, \ldots, x_n]$ e $G \subseteq I$ uma base de Grobner. Então $f \in I$ se e somente se $R_G(f) = 0$.

Demonstração: ($\Longleftarrow$) Por definição, se o resto da divisão de f pelos geradores de G for zero, então ele pertence ao ideal I.($\Longrightarrow$) Suponha, por absurdo, que $f \in I$, mas que $R_G(f) \neq 0$. Portanto, posso escrever f como combinação linear dos geradores de G com o resto r, isto é, $f = q_1 \cdot g_1 + \cdots + q_s \cdot g_s + r$. Observa-se que $r \in I$, porque $f - r \in I$, mas isso quer dizer, por construção das bases de Grobner, que $\text{in}(g_i)$ divide $\text{in}(r)$, para $g_i \in G$. Encontramos uma contradição, visto que o resto não é reduzido, e portanto, o resto é único. $\square$

Além de conseguirmos solucionar o 'Problema da Pertinência', existem outras aplicações da Base de Grobner, que, em particular, são muito interessantes para MPKC, como a solução de sistemas de equações polinomiais.

5.1 Solução de Equações Polinomiais com Bases de Grobner

Quando lidamos com polinômios lineares, computar a base de Grobner reduzida é tão fácil quanto realizar Eliminação Gaussiana na matriz dos coeficientes dos geradores em várias variáveis. Contudo, quando o grau desses polinômios for estritamente maior que um, precisamos recorrer a um algoritmo para computar a base de Grobner reduzida.

Esse algoritmo, resumidamente, verifica se o inicial de um dos geradores g divide algum outro gerador k, se for o caso, $G = G \backslash \{k\}$.

Lema 5.3

Se $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$ é um ideal radical de dimensão zero, tal que todas soluções sobre um corpo algebricamente fechado tem coordenadas n distintas, então sob a ordem monomial lexicográfica $>_{lex}$ a base de Grobner reduzida de I tem forma

$$\begin{cases} x_1 - p_1(x_n) = 0 \\ x_2 - p_2(x_n) = 0 \\ \vdots \\ x_{n-1} - p_{n-1}(x_n) = 0 \\ p_n(x_n) = 0 \end{cases}$$

tal que $\text{grau}(p_i) < \text{grau}(p_n)$, para $i < n$.

Essa característica é fundamental para calcularmos eficientemente a solução do sistema de dimensão zero, isto se deve a garantia da unicidade de $g = G \cap \mathbb{F}[x_i]$ para cada $1 \leq i \leq n$.

Lema 5.4

Seja G uma base de Grobner para o ideal I de $\mathbb{F}[x_1, \ldots, x_n]$. Um sistema de equações polinomiais $g_1(x_1, \ldots, x_n) = \cdots = g_k(x_1, \ldots, x_n) = 0$ possui soluções finitas, isto é, tem dimensão zero se, e somente se, a variedade afim associada a ele $\mathbb{V}(\langle g_1, \ldots, g_k \rangle)$ for um conjunto finito.

5.2 Obtendo o plaintext com Bases de Grobner

Dados uma chave pública $P \in \mathbb{F}[x_1, \ldots, x_n]^m$ e o ciphertext $y = (y_1, \ldots, y_m) \in \mathbb{F}^m$ para a criptografia de Chen et. al. Considere $h_i = f_{i+1} - f_i$, sabemos por (3) que os monômios de grau 2 são removidos da equação, então $grau(h_i) = 1$, para todo $1 \leq i \leq n - a$.

Proposição 5.5

Se uma matriz quadrada $C' \in \mathbb{F}^{n \times n}$ é invertível então suas colunas são linearmente independentes.

Demonstração: Como C' é invertível, então existe uma matriz quadrada $B \in \mathbb{F}^{n \times n}$ tal que $C' \times B = B \times C' = I$. Suponha, por absurdo, que as colunas de C' são linearmente dependentes. Então, por haver uma redundância nesses vetores, deve existir $\vec{v} \neq \overline{0}$ tal que $C' \cdot \vec{v} = \overline{0}$. Contudo, isso implica em $B \cdot C' \cdot \vec{v} = \overline{0} \Longrightarrow I \cdot \vec{v} = \overline{0} \Longrightarrow \vec{v} = \overline{0}$, que é uma contradição. Portanto, C' é linearmente independente. □

Seja $\mathbf{span}(P) = \{p_1, \ldots, p_m\}$. Como construímos C para que C' seja invertível, o conjunto $\{h_1, \ldots, h_{n-a}\}$ é linearmente independente e, portanto, $\mathbf{span}(F) = \{h_1, \ldots, h_{n-a}, f_1, g_1, \ldots, g_s\}$. Segue que, por construção de P como composição, $\mathbf{span}(P) = S \circ \mathbf{span}(F) = \{S \circ h_1, \ldots, S \circ h_{n-a}, S \circ f_1, S \circ g_1, \ldots, S \circ g_s\}$. Podemos concluir que $\mathbf{span}(P)$ é linearmente independente, porque S e $span(F)$ também o são.

Definição 5.6

Seja $P \in \mathbb{F}^m$ um vetor com polinômios de grau 2 sobre as variáveis $x_1, \ldots, x_n$, a função Quad : $\mathbb{F}^m[x_1, \ldots, x_n] \to \mathbb{F}^{m \times n}$ armazena os coeficientes dos termos quadráticos dos polinômios em uma matriz de dimensão $m \times n$.

Para ilustrar o comportamento desta função, segue o exemplo. Sejam $P = [x_1^2 + 2x_2^5, x_1 + x_2^2, 5x_1^2 + 1] \in \mathbb{F}^3$ para $n = 2$.

$$\text{Quad}(P) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 0 \end{bmatrix}$$

Para construir o ataque, precisamos introduzir novas variáveis ao nosso anel de polinômios, que se torna $\mathbb{F}[x_1, \ldots, x_n, z_1, \ldots, z_m]$. Com essa função em mãos, montamos um sistema linear com a restrição $\sum_{i=1}^{m} z_i \cdot \text{Quad}(p_i) = 0$, onde $p_i \in P$, isto é, queremos anular toda a parte quadrática do polinômio para buscarmos apenas soluções lineares, visto que se $\sum_{i=1}^{m} z_i \cdot \text{Quad}(p_i) = 0$, então grau$(\sum_{i=1}^{m} z_i \cdot p_i) = 1$ e vice-versa.

Por construção de F, existem apenas $(n-a)$ polinômios de grau 1 na função regular, logo o subespaço da solução de $\sum_{i=1}^{m} z_i \cdot \text{Quad}(p_i) = 0$ também terá dimensão $(n-a)$. Nesse sentido, sejam os vetores linearmente independentes $Z = \{z_1^{(i)}, \ldots, z_m^{(i)}\} \in \mathbb{F}^m$, para $1 \leq i \leq (n-a)$ que formam uma base para o espaço de soluções para o núcleo de $M := [\text{Quad}(p_1), \text{Quad}(p_2), \ldots, \text{Quad}(p_m)]$, temos o sistema abaixo, que é essencialmente $Z \cdot P$

$$\begin{cases} k_1(x_1, \ldots, x_n) = z_1^{(1)} \cdot p_1 + \cdots + z_m^{(1)} \cdot p_m \\ \vdots \\ k_{n-a}(x_1, \ldots, x_n) = z_1^{(n-a)} \cdot p_1 + \cdots + z_m^{(n-a)} \cdot p_m \end{cases} \tag{9}$$

Sendo assim, o Span(P) é gerado por $k_1, \ldots, k_{n-a}$ polinômios lineares e outros $(s+1)$ polinômios quadráticos, reduzindo o problema MQ originalmente um sistema de m equações quadráticas em n variáveis, para outro com $m - (n-a) = (s+1)$ equações quadráticas em $n - (n-a) = a$ variáveis.

Por fim, ao construir o sistema linear que será usado para computar Grobner, estabelecemos que $y_i = p_i$, onde $y_i \in y$ e $p_i \in P$, para todo $1 \leq i \leq m$ com as restrições de (6), cujos z's impedem que existam coeficientes não nulos para polinômios quadráticos de P.

$$\begin{cases} p_1 - y_1 = 0 \\ \quad \vdots \\ p_m - y_m = 0 \\ \sum_{i=1}^{m} z_i^{(1)} \cdot p_i - \sum_{i=1}^{m} z_i^{(1)} \cdot y_i = 0 \\ \quad \vdots \\ \sum_{i=1}^{m} z_i^{(n-a)} \cdot p_i - \sum_{i=1}^{m} z_i^{(n-a)} \cdot y_i = 0 \end{cases} \tag{10}$$

Computando a base de Grobner reduzida do ideal $I = \langle p_1 - y_1, \ldots, p_m - y_m, \sum_{i=1}^{m} z_i^{(1)} \cdot p_i - \sum_{i=1}^{m} z_i^{(1)} \cdot y_i, \ldots, \sum_{i=1}^{m} z_i^{(n-a)} \cdot p_i - \sum_{i=1}^{m} z_i^{(n-a)} \cdot y_i \rangle$ obtemos polinômios na forma $x_i - t_i, \forall 1 \leq i \leq n$, onde $t_i \in \mathbb{F}$ é um dígito da mensagem codificada. Sendo assim, basta calcular $(x_i - (x_i - t_i))$ em $\mathbb{F}$ e decodificar a mensagem para recuperar o texto original.