GET Criptografia 1 - Material Complementar

Guilherme Cappelli

1 Introdução

Nesse material abordaremos uma breve introdução à Criptografia, explorando um pouco dos conceitos fundamentais e alguns exemplos de Criptografia. Para isso, veremos um pouco de Teoria dos Números que será essencial para a compreensão dos esquemas apresentados.

2 Cifras de Substituição

Suponha que você tenha em mãos uma mensagem "Pevcgbtensvn ru Yrtny", você deve pensar: "Esse cara arremessou o teclado na parede e deu nisso". Na verdade o que está acontecendo nessa mensagem é que as letras do alfabeto foram embaralhadas, isto é, se eu digitar a letra "C" ela pode sair uma outra letra, como "P".

Pense nesse mecanismo como uma caixinha mágica com fios emaranhados ligando uma letra do alfabeto em outra, quando você pressiona uma letra no teclado, ela é imediatamente transformada em outra do outro lado. Dessa forma, cada letra é mapeada para outra no alfabeto, então sabemos que na mensagem do exemplo a letra "C" sempre será "P" quando passar pela caixinha mágica.

Uma das maneiras de elaborar uma cifra de substituição é fazer um deslocamento das letras do alfabeto para a esquerda ou direita. Nesse sentido, se realizamos um 'shift' em 13 posições para a direita, a primeira letra do alfabeto original "A" se torna a 13º letra dele, que é o "N" e assim por diante, como podemos ver na imagem abaixo.

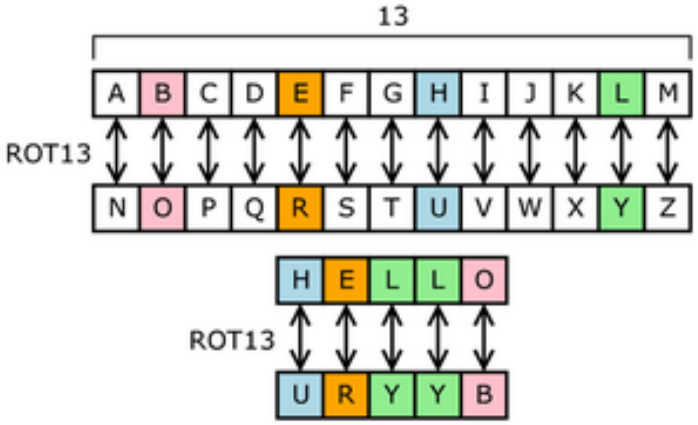

Figure 1: Rot13 - Cifra de Substituição

Para retornar a mensagem original basta realizar o mesmo processo com a cifra, no caso de ROT-13, basta realizarmos ROT-$(26 - 13) = 13$ isso é o que chamamos de 'trapdoor' ou 'alçapão' de uma Criptografia. O que quero dizer com isso é que embaralhar as letras e obter uma cifra é fácil, mas obter a mensagem original é necessário descobrir em quantas posições o alfabeto foi deslocado, que, atualmente em questão de segundos com o poder computacional de um PC da Xuxa descobrimos isso. Na época que ela foi inventada deu uma certa dor de cabeça pra entender o que estava sendo transmitido.

3 Criptografia Simétrica

Vimos na seção anterior que para reverter uma cifra, ou seja, decriptar uma mensagem bastava realizar o mesmo procedimento que havia sido utilizado para encriptar. Isto é o que caracteriza uma Criptografia Simétrica, onde o mesmo mecanismo que encripta uma mensagem também a decripta, observe outro exemplo para ficar mais claro.

3.1 Portas Lógicas

Para os que já fizeram Circuitos Digitais e não aguentam mais ver isso, parabéns você pode pular para o tópico 3.2. Em uma porta lógica são realizadas operações lógicas booleanas dadas duas ou mais entradas binárias, gerando um saída baseada em regras predefinidas.

Cada porta lógica possui uma tabela verdade que indica quando a saída será 0 ou 1, isso se dá de forma determinística dependendo exclusivamente das entradas da porta. No caso da porta XOR de duas entradas, que é a operação de 'OU' Exclusivo, é retornado 1 (verdadeiro) apenas quando uma das entradas é 1 (verdadeira), mas não ambas. Já quando ambas as entradas são iguais, ou seja, ambas as entradas são 0 ou 1, a saída é 0 (falso).

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Figure 2: Tabela Verdade XOR

Em geral, as linguagens de programação já possuem o operador bitwise XOR, que é o que nos interessa no momento. Assim, podemos tomar duas sequências binárias como entrada da porta XOR e retornamos a saída para cada dígito, observe:

$$0011 \oplus 1100 = 1111$$

$$1010 \oplus 1010 = 0000$$

No exemplo acima notamos que quando uma sequência é a negação da outra o resultado é uma sequência de 1s, já quando são iguais temos uma sequência resultante de zeros. Se quisessemos operar com mais do que duas entradas poderíamos aplicar a propriedade associativa do XOR.

$$(A \oplus B) \oplus C = A \oplus (B \oplus C)$$

Propriedades do XOR

- Associatividade: $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
- Comutatividade: $A \oplus B = B \oplus A$
- Elemento neutro: $A \oplus 0 \ldots 0 = A$
- Auto-inversão: $A \oplus A = 0$

3.2 Cifra de XOR

Temos aqui outra cifra simétrica, sendo esta baseada em portas lógicas para esconder as mensagens enviadas. A ideia central da cifra é aplicar a porta XOR bitwise com a mensagem e outras chaves numéricas. Talvez você esteja se perguntando como que vamos fazer um XOR com texto, a resposta é simples: não vamos =).

Primeiro precisamos codificar uma mensagem em números, ou seja, atribuir a cada letra um valor numérico, isso pode ser feito (i) pela tabela ASCII, (ii) criando seu próprio mapeamento bijetivo ou (iii) usando a biblioteca pycryptodome que já possui os métodos 'bytes_to_long' e 'long_to_bytes' que convertem sua string para um long int e um long int para a string de volta, respectivamente.

A ideia central dessa cifra é esconder a mensagem codificada aplicando XOR com chaves à escolha do programador, isto é, dada uma mensagem codificada m vamos aplicar a operação bitwise XOR com as chaves $k_1, \ldots, k_n$.

$$\text{ciphertext} = m \oplus \bigoplus_{i=1}^{n} k_i$$

Perceba que a ordem em que inserimos as chaves não importa por causa da comutatividade do XOR. Para recuperar a mensagem original usaremos as propriedades da porta lógica anteriormente listadas, repetindo os passos da cifragem com as chaves fornecidas.

Nesse sentido, se aplicarmos XOR de k_j para algum $1 \leq j \leq n$ em ciphertext, podemos rearranjar as chaves para aplicarmos associatividade na operação XOR de k_j com ele mesmo, para fins práticos suponha $j = 1$

$$\text{ciphertext} \oplus k_1 = m \oplus (k_1 \oplus k_1) \oplus \bigoplus_{i=2}^{n} k_i$$

$$\text{ciphertext} \oplus k_1 = m \oplus 0 \oplus \bigoplus_{i=2}^{n} k_i$$

Contudo, sabemos que 0 é elemento neutro para a porta XOR, então $m \oplus 0 = m$. Repetindo esse processo até se esgotarem as chaves obtemos a mensagem original m.

4 Aritmética Modular

A arimética é a área da matemática que estuda operações como adição e multiplicação em conjuntos definidos, por exemplo: o conjunto dos naturais $\mathbb{N} = \{1, 2, 3, \ldots\}$ (sou da religião de que 0 não é natural) e o conjunto dos inteiros $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$. Nesse contexto, a aritmética modular é um sistema de aritmética para inteiros módulo $p \in \mathbb{Z}$.

Suponha que você acordou cedo pra ter uma GET de Criptografia às 9 horas da manhã no Fundão, mas ninguém te avisou que teria matemática, então agora você só quer que dê 13 horas pra poder voltar pra casa. Na parede da sala existe um relógio de ponteiro e você sabe que o ponteiro das horas precisa avançar 4 posições para a direita.

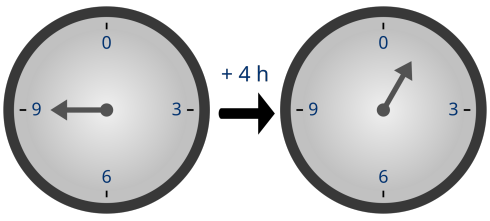

Figure 3: Relógio da Sala

Contudo, o relógio possui apenas 12 posições para o ponteiro das horas e $9 + 4 = 13 > 12$. Essa situação pode ser contornada se reiniciarmos a contagem das horas a cada volta no relógio, isto significa que a hora que você pode se levantar da cadeira quando o ponteiro das horas bater no 1, pois $13 = \underbrace{12 \cdot 1}_{\text{uma volta}} + 1$.

Definição 4.1

Sejam $a, b, k, p \in \mathbb{Z}$, dizemos que a é congruente a b módulo p ou na notação $a \equiv b \mod p$, quando $a = k \cdot p + b$.

Sendo assim, na analogia do relógio o módulo é 12, porque são necessários 12 passos até a contagem da hora ser reiniciada, mas atente-se de que se a analogia fosse sobre o ponteiro dos **minutos** seriam necessários 60 passos até voltarmos ao minuto zero. Segue abaixo alguns outros exemplos:

$$3 \equiv 1 \mod 2, \text{ porque } 3 = 2 \cdot 1 + 1$$

$$9 \equiv 0 \mod 3, \text{ porque } 9 = 3 \cdot 3 + 0$$

$$37 \equiv 5 \mod 8, \text{ porque } 37 = 8 \cdot 4 + 5$$

Agora é fácil perceber que encontrar a congruência módulo p de um inteiro a é aplicar o Algoritmo de Divisão, onde o divisor é o módulo em questão, o quociente é quantas 'voltas' no relógio o ponteiro deu e o resto é o número procurado. Assim, quando $a \equiv b \mod p$, o que queremos dizer com isso é que a diferença $(a - b)$ é um número múltiplo de p.

Lema 4.2

Se $a \equiv b \mod m$, então para todo então, para todo $p \mid m$, $a \equiv b \mod p$.

Demonstração: Por definição, tem-se que $m \mid (a - b)$, ou seja, $a - b = km$, para algum $k \in \mathbb{Z}$. Como $p \mid m$, existe $t \in \mathbb{Z}$ tal que $m = tp$, e portanto, $a - b = k(tp) = (kt)p$. Segue disso, que $p \mid (a - b) \implies a \equiv b \mod p$.

Em outras palavras, a congruência se mantém para todos os divisores do módulo m. Segue o exemplo abaixo:

Exemplo Prático 1

Sejam $a = 38$, $b = 14$ e $m = 12$. Temos $38 \equiv 14 \mod 12$, porque $38 - 14 = 24 = 2 \cdot 12$. Tome p como algum dos divisores de 12

$$p \in \{1, 2, 3, 4, 6, 12\}$$

Se $p = 6$:

$$38 \equiv x \mod 6 \overset{\text{Divisão}}{\implies} 38 = y + x \cdot 6 \implies 38 = 14 + 4 \cdot 6 \implies 38 \equiv 14 \mod 6$$

Se $p = 4$:

$$38 - 14 = 24 = 6 \cdot 4 \implies 38 \equiv 14 \mod 4$$

Se $p = 3$:

$$38 - 14 = 24 = 4 \cdot 3 \implies 38 \equiv 14 \mod 3$$

Definição 4.3

Dados $p_1, \ldots, p_n \in \mathbb{Z}$, o **máximo divisor comum** (mdc) desses números é o maior inteiro que divide todos simultaneamente. Além disso, se $\text{mdc}(p_1, \ldots, p_n) = 1$, então esses números são ditos coprimos.

Lema 4.4

Se a e b são inteiros positivos, então existem inteiros n e m de forma que $\text{mdc}(a, b) = na + mb$.

Demonstração: Seja $\mathbb{T} = \{n \cdot a + m \cdot b : n, m \in \mathbb{Z}\}$. Pelo Princípio da Boa-Ordem, podemos rearranjar os elementos de $\mathbb{T}$ em ordem crescente e assim existe $c = min\{\mathbb{T} \cap \mathbb{N}\}$. Suponha, por absurdo, que c não divide a, então pelo algoritmo de divisão $a = q \cdot c + r$, onde o resto r não pode ser zero e é estritamente menor que c. Entretanto, $r = a - q \cdot c$ e se denotarmos c por $n_0 \cdot a + m_0 \cdot b$, obtemos $r = a(1 - qn_0) + (-m_0 q)b$, que claramente divide a e b, e portanto pertence a $\mathbb{T} \cap \mathbb{N}$. Ora, se $r \in \mathbb{T} \cap \mathbb{N}$ e $r < c$, temos um absurdo, pois supomos que c é o mínimo do conjunto. Daí, c divide a e de forma análoga provamos que c divide b. Seja $d = \text{mdc}(a, b)$. Como d divide a e b por definição, devem existir $s, t \in \mathbb{Z}$ tais que $a = sd$ e $b = td$. Substituindo, $c = n_0 sd + m_0 td = d(n_0 s + m_0 t)$. Portanto, d divide c, que implica $c \geq d$. Como d é o maior divisor comum de a e b, temos que $d \geq c$. Segue disso, que $c = d = min\{\mathbb{T} \cap \mathbb{N}\} = n_0 a + m_0 b$. □

Teorema 4.5

Sejam $a, p > 1 \in \mathbb{N}$. Se $\text{mdc}(a, p) = 1$, então existe a^{-1} tal que $a \cdot a^{-1} \equiv 1 \mod p$.

Demonstração: Pelo **Lema 4.4**, existem $n, m \in \mathbb{Z}$ tais que $n \cdot a + m \cdot p = 1$ e naturalmente vemos que $n \cdot a + m \cdot p \equiv 1 \mod p$. Como o resto da divisão de $m \cdot p$ por p é zero, segue que $m \cdot p \equiv 0 \mod p$, e por consequência, $n \cdot a \equiv 1 \mod p$. Logo, $n = a^{-1} \longrightarrow a \cdot a^{-1} \equiv 1 \mod p$. □

Pequeno Teorema de Fermat

Sejam $a \in \mathbb{Z}$ e p um número primo tais que a não é divisível por p, então $a^{p-1} \equiv 1 \mod p$.

Demonstração: Considere $S := \{0, 1, \ldots, (p-1)\}$ e o conjunto $aS := \{a, 2a, \ldots, a(p-1), ap\}$. Afirmamos que os elementos de aS são incongruentes entre si sob módulo p, isto é, os restos de aS mod p é S. Suponha, por absurdo, que dados $a \cdot n, a \cdot m \in aS$, onde $1 \leq n \neq m \leq p$, temos que $an \equiv am \mod p$. Por definição, $an - am = kp$, para um $k \in \mathbb{Z}$, que nos diz que p divide $(na - ma) = a(n - m)$, ou seja, p divide a ou p divide $(n - m)$.

Entretanto, por hipótese temos que $\text{mdc}(a, p) = 1$, então p não divide a, e p não pode dividir $(n-m)$ também, porque $0 \neq (n - m) < p$ e p é primo. Isso significa que se tomarmos o produto dos elementos de $aS \setminus \{pa\}$ sob módulo p teremos o mesmo resultado do produto dos elementos de $S \setminus \{0\}$. Então, $a \cdot 2a \cdot \ldots \cdot (p-1)a = a^{p-1}(p-1)!$, que nos dá $a^{p-1}(p-1)! \equiv (p-1)! \mod p$. Como $\text{mdc}((p-1)!, p) = 1$, pelo **Teorema 4.5**, existe inverso multiplicativo módulo p de $(p-1)!$, segue disso que $a^{p-1} \equiv 1 \mod p$. □

Corolário 4.6

Sejam $a \in \mathbb{Z}$ e p um número primo tais que a não é divisível por a, então $a^p \equiv p \mod p$.

Com efeito, se multiplicarmos $a^{p-1} \equiv 1 \mod p$ por a de ambos os lados, obtemos a igualdade do colorário.

Definição 4.7

A *função totiente de Euler* dada por $\phi(n)$ é a cardinalidade do conjunto $\mathbb{Z}_n$ definida como

$$\phi(n) = \#\{1 \leq x \leq n \mid \text{mdc}(x, n) = 1\}$$

isto é, se $n = p_1^{k_1} \dots p_r^{k_r}$ é um número composto, onde p_i são primos distintos, então $\phi(n) = (p_1 - 1) \cdot p_1^{k_1 - 1} \dots (p_r - 1) \cdot p_r^{k_r - 1}$.

Em particular, se $n = p \cdot q$, então $\phi(n) = (p-1) \cdot p^0 \cdot (q-1) \cdot q^0 = (p-1) \cdot (q-1)$.

Lema 4.8

Se $\text{mdc}(p, q) = 1$, $a \equiv b \mod p$ e $a \equiv b \mod q$, então $a \equiv b \mod pq$.

Demonstração: Por hipótese existem inteiros k, l tais que $a - b = kp$ e $a - b = lp$. Defina os conjuntos $P := \{x \in \mathbb{Z} : x = mp,\ m \in \mathbb{Z}\}$ dos múltiplos de p e $Q := \{x \in \mathbb{Z} : x = nq,\ n \in \mathbb{Z}\}$ dos múltiplos de q. Temos que $a - b$ pertence à P e Q simultaneamente, portanto $a - b$ também pertence ao conjunto dos múltiplos do **mínimo múltiplo comum** $MMC := \{y \in \mathbb{Z} : y = k \cdot mmc(p, q),\ k \in \mathbb{Z}\}$. Como $\text{mdc}(p, q) = 1$ e $\text{mmc}(p, q) = \frac{p \cdot q}{mdc(p,q)}$, então $\text{mmc}(p, q) = pq$, segue disso que qualquer número $y \in MMC$ deve ser múltiplo de pq. Logo, $a - b = n \cdot (pq)$ para algum $n \in \mathbb{Z}$ e por definição $a \equiv b \mod pq$. □

Teorema de Euler

Se n é um inteiro positivo tal que $\text{mdc}(b, n) = 1$, então $b^{\phi(n)} \equiv 1 \mod n$

Demonstração: Se n for primo, então a prova segue do **Pequeno Teorema de Fermat**. Suponha que n é um número composto, e para que a seção seguinte ser mais intuitiva, tome $n = pq$, para p, q primos. Sabemos que $b^{p-1} \equiv 1 \mod p$ e que $b^{q-1} \equiv 1 \mod q$, então $(b^{p-1})^{(q-1)} \equiv 1^{(q-1)} \mod p \implies b^{(p-1)\cdot(q-1)} \equiv 1 \mod p$ e de maneira análoga obtemos que $b^{(q-1)\cdot(p-1)} \equiv 1 \mod q$. Por hipótese, $\text{mdc}(b, pq) = 1$, logo pelo **Lema 4.8**, temos que $b^{(p-1)\cdot(q-1)} \equiv 1 \mod pq$. Pela **Definição 4.7** sabemos que $\phi(pq) = (p-1)(q-1)$, então, $b^{\phi(n)} \equiv 1 \mod n$. □

5 Criptografia RSA

Nessa última seção, veremos uma criptografia assimétrica cuja trapdoor se baseia em um problema de fatoração, o RSA. Essa é chamada assim por conta das inicias de cada um dos 3 pesquisadores que a criaram.

Queremos escolher n, e, d de forma que $b^{e \cdot d} \equiv b \mod n$. Para isso, vamos tomar $n = p \cdot q$, com $p \neq q$ números primos. Repare que se $e \cdot d \equiv 1 \mod (p-1) \cdot (q-1)$, isto é,$e \cdot d = 1 + \phi(n) \cdot k$ onde $\text{mdc}(e, \phi(n)) = 1$, podemos reescrever

$$b^{ed} \equiv b^{1 + \phi(n) \cdot k} \equiv b^{\phi(n)^k} b \mod n$$

Pelo **Teorema de Euler** temos $b^{\phi(n)} \equiv 1 \mod n$. Daí

$$b^{ed} \equiv b \mod n$$

que é o que buscávamos. Considere uma mensagem m codificada a ser enviada por um canal inseguro, facilmente encriptamos-na a partir de

$$c \equiv m^e \mod n$$

Como $e \cdot d \equiv 1 \mod \phi(n)$, então

$$m \equiv c^d \mod n$$

recupera a mensagem original, restando apenas decodificar a mensagem para texto. Em resumo, a chave pública, cuja qualquer pessoa pode ter acesso é a tupla (n, e), isto é, as informações necessárias para encriptar qualquer mensagem, e a chave privada é (p, q, d), que é capaz de decriptar as mensagens cifradas com (n, e).

6 Vulnerabilidades do RSA

6.1 Ataque de Texto Cifrado Escolhido

O Chosen Ciphertext Attack vai explorar a propriedade homomórfica do RSA nas operações de encriptar e decriptar. Sejam m_1, m_2 mensagens em texto plano e $c_1 \equiv m_1^e \mod n$.

Definição 6.1

Um *Oráculo Criptográfico* é um modelo de serviço que recebe uma entrada e responde com uma saída, mas qualquer pessoa de fora não sabe exatamente como o cálculo é feito internamente, como uma caixa preta.

Suponha que eu tenho um Oráculo Criptográfico RSA que é capaz de cifrar e decifrar mensagens, isso é, opera com entradas de plaintext e ciphertext, mas por natureza não decifra mensagens que ele mesmo cifrou.

Nesse sentido, é possível explorar a capacidade homomórfica das operações $E(x) = x^e \mod n$ e $D(c) = c^d \mod n$, isso é,

$$E(m_1 \cdot m_2) \equiv (m_1 m_2)^e \equiv m_1^e \cdot m_2^e \equiv E(m_1) \cdot E(m_2) \mod n =$$

$$D(c_1 \cdot c_2) \equiv (c_1 c_2)^d \equiv c_1^d \cdot c_2^d \equiv D(c_1) \cdot D(c_2) \mod n =$$

porque se temos em mãos c_1, podemos encriptar um valor pequeno como 2 com o Oráculo, obtendo $c_2 = 2^e \mod n$ e multiplicar $c_1 \cdot c_2$.

Se passarmos como entrada da operação de decriptação para o Oráculo esse produto $c_1 c_2$, ele não será capaz de bloquear nossa operação. Portanto, temos

$$D(c_1 c_2) = D(c_1) \cdot D(2^e) = m_1 \cdot 2 \mod n$$

Sendo assim, basta dividir o resultado dessa operação pelo número $m_2 = 2$ e obtemos a mensagem codificada m_1.

6.2 Ataque Broadcast de Hastad

Teorema 6.2

Suponha $n_1, \ldots, n_k \in \mathbb{N}$ tal que $\text{mdc}(n_i, n_j) = 1$ e $a_1, \ldots, a_k \in \mathbb{Z}$. Então o sistema

$$\begin{cases} X \equiv a_1 \mod n_1 \\ \vdots \\ X \equiv a_k \mod n_k \end{cases}$$

possui uma solução única sob módulo $N = n_1 \times n_2 \times \cdots \times n_k$.

Demonstração: Seja $N = n_1 \dots n_k$ e $N_i = \frac{N}{n_i}$. Note que $\text{mdc}(n_i, N_i) = 1$, então pelo **Lema 4.4** existem $x_i, y_i \in \mathbb{Z}$ tais que $N_i \cdot x_i + n_i \cdot y_i = 1$. Segue disso, que $N_i \cdot x_i \equiv 1 \mod n_i \implies (N_i \cdot a_i)x_i \equiv a_i \mod n_i$. Perceba que se $i \neq j$, então $N_i \equiv 0 \mod n_j$, visto que N_i é o produto de todos n_k exceto n_i, então se $j \neq i$ temos que $N_i = (y) \cdot n_j$ é um múltiplo de n_j e por definição é congruente a zero. Denote $X = N_1 \cdot a_1 \cdot x_1 + \dots + N_k \cdot a_k \cdot x_k$ e tome $j \in \{1, \dots, k\}$, obtemos $X \equiv N_j \cdot a_j \cdot x_j \mod n_j$ pela observação anterior. Como $(N_j \cdot a_j) \cdot x_j \equiv a_j \mod n_j$, concluimos que $X \equiv a_j \mod n_j$.

Suponha, por absurdo, que existam duas soluções distintas X e Y para esse sistema, onde $X \equiv a_i \mod n_i$ e $Y \equiv a_i \mod n_i$, $\forall\ 1 \leq i, j \leq k$, isto é,

$$\begin{array}{rcll} X & \equiv & a_1 & \mod n_1 \\ \vdots & & & \\ X & \equiv & a_k & \mod n_k \end{array} \qquad \begin{array}{rcll} Y & \equiv & a_1 & \mod n_1 \\ \vdots & & & \\ Y & \equiv & a_k & \mod n_k \end{array}$$

Tome a diferença $X - Y$ em cada linha, obtendo o sistema de congruências

$$\begin{cases} X - Y \equiv a_1 - a_1 \equiv 0 \mod n_1 \\ X - Y \equiv 0 \mod n_2 \\ \vdots \\ X - Y \equiv 0 \mod n_k \end{cases}$$

Isto significa que $X - Y$ divide cada n_i, ou seja, divide o $\textbf{mmc}(n_1, \dots, n_k)$. Por hipótese, $\textbf{mdc}(n_1, \dots, n_k) = 1$, portanto o **mínimo múltiplo comum** entre os n_i é o produto $N = n_1 \times \dots \times n_k$, daí $X - Y$ divide N que implica $X - Y \equiv 0 \mod N \implies X \equiv Y \mod N$, absurdo. Portanto, existe solução única para o sistema. □

Exemplo Prático 2

Seja o sistema de congruências

$$\begin{cases} 4X \equiv 5 \mod 9 \quad (i) \\ 2X \equiv 6 \mod 20 \quad (ii) \end{cases}$$

Tome a congruência (i) e note que $\text{mdc}(4, 9) = 1$, logo pelo **Teorema 4.5** existe $4 \cdot 4^{-1} \equiv 1 \mod 9$. Nesse sentido, temos $X \equiv 5 \cdot 4^{-1} \mod 9$. Mas pelo Algoritmo Estendido de Euclides, que será omitido nessa aula por falta de tempo, descobrimos que $5 \cdot 4^{-1} \equiv 5 \cdot 7 \equiv 8 \mod 9$.
Já a segunda linha do sistema (ii) diz que $2X \equiv 6 \mod 20$, onde $\text{mdc}(2, 20) \neq 1$, porém ambos os lados da equivalência são inteiros pares. Sendo assim, $2X \equiv 2 \cdot 3 \mod 2 \cdot 10 \implies X \equiv 3 \mod 10$, porque $2X - 2 \cdot 3 = 2 \cdot 10k \implies_{\div 2} X - 3 = 1 \cdot 10k \implies x \equiv 3 \mod 10$.
Portanto, temos um novo sistema reduzido onde o **máximo divisor comum** dos módulos são coprimos, isto é, $\text{mdc}(9, 10) = 1$.

$$\begin{cases} X \equiv 8 \mod 9 \\ X \equiv 3 \mod 10 \end{cases}$$

Na demonstração do **Teorema 6.1** vimos que $N_i \cdot x_i \equiv 1 \mod n_i$, faremos isso então. Sejam $n_1 = 9$, $n_2 = 10$, $N_1 = \frac{9 \cdot 10}{9} = 10$ e $N_2 = \frac{10 \cdot 9}{10} = 9$, temos $N_1 \cdot x_1 \equiv 1 \mod n_1$ e $N_2 \cdot x_2 \equiv 1 \mod n_2$, isto é, $10 \cdot x_1 \equiv 1 \mod 9$ e $9 \cdot x_2 \equiv 1 \mod 10$.
Facilmente encontramos $x_1 \equiv 1 \mod n_1$, pois $10 \equiv 1 \mod 9$ e $x_2 \equiv 9 \mod 10$, porque $81 \equiv 1 \mod 9$. Por fim, precisamos calcular $X \equiv \sum N_i x_i a_i \mod N$. Assim, $X \equiv 10 \cdot 1 \cdot 8 + 9 \cdot 9 \cdot 3 \equiv 323 \mod N$. Como $N = n_1 \times n_2 = 10 \cdot 9 = 90$, segue que $X \equiv 53 \mod 90$.

Sejam as configurações de sistemas RSA (n_i, e_i, d_i), para $i \geq k \in \mathbb{N}$. Suponha que a mensagem m foi encriptada $k = 3$ vezes e você tem acesso às chaves públicas e aos ciphertexts. Para fins de sanidade

mental, defina $e_i := 3$ e monte o seguinte sistema de congruências

$$\begin{cases} X \equiv m^3 \mod n_1 \\ X \equiv m^3 \mod n_2 \\ X \equiv m^3 \mod n_3 \end{cases}$$

Pelo **Teorema 6.1**, encontramos uma solução única para o sistema da forma $X \equiv \sum N_i x_i a_i \mod N$, logo como $X, m^3 < n_1 \times n_2 \times n_3$ já está em sua forma reduzida, daí basta tirar a raiz cúbica de X e encontramos a mensagem original.

Exemplo Prático 3

Seja $m = 20$ e os sistemas RSA de formato (n, e, d): $(77, 3, d_1), (221, 3, d_2), (437, 3, d_3)$. A mensagem foi encriptada das seguintes formas:

$$\begin{cases} X \equiv 20^3 \equiv 69 \mod 77 \\ X \equiv 20^3 \equiv 44 \mod 221 \\ X \equiv 20^3 \equiv 134 \mod 437 \end{cases}$$

Pelo **Teorema 6.1** temos que $N = 77 \times 221 \times 437 = 7436429$ e $N_1 = 96577, N_2 = 33649$ e $N_3 = 17017$. Calculemos (isso não é um sistema, apenas fica mais didático desse jeito - Viva Paulo Freire).

$$\begin{cases} N_1 \cdot x_1 \equiv 96577 \cdot x_1 \equiv 1 \mod 77 \\ N_2 \cdot x_2 \equiv 33649 \cdot x_2 \equiv 1 \mod 221 \\ N_3 \cdot x_3 \equiv 17017 \cdot x_3 \equiv 1 \mod 437 \end{cases}$$

com resultado $x_1 \equiv 73 \mod 77$, $x_2 \equiv 190 \mod 221$ e $x_3 \equiv 84 \mod 437$. Por fim, $X \equiv \sum N_i x_i a_i \mod N$, logo

$$X \equiv 96577 \cdot 73 \cdot 69 + 33649 \cdot 190 \cdot 44 + 17017 \cdot 84 \cdot 134 \mod 7436429$$

$$X \equiv 8000 \mod 7436429$$

Assim, basta extrair a raiz e-ésima de X e obtemos a mensagem original. Como $e = 3$, temos $X^{(1/3)} = 20$.