IA012 - Tarefa 6

Rodrigo Seiji Piubeli Hirao (186837)

16 de dezembro de 2021

01) Caso a mensagem (m) a ser cifrada e a chave pública (e) sejam muito pequenas, tal que $m^e < n$, então o módulo será o próprio valor, o que implica em uma possível decriptografação sem a chave privada apenas fazendo $m^{\frac{1}{e}}$, assim pode ser adicionado bits extras ao fim da mensagem.

Uma solução para esse problema é usar um padding aleatório ao final da mensagem, mas esse processo pode ser descoberto se a mesma mensagem for enviada 2 vezes com paddings diferentes, e poderá usar o **Coppersmith's Short Pad Attack**.

Além disso há diversos outros meios de ataque a chaves pequenas, como usando o **Teorema de Coppersmith** ou o **Ataque de Broadcast de Hasta**.

02) $123^{145} \pmod{35} = 18$

Temos que

- $35 = 5 \times 7$
- $123 = 3 \times 41$
- $145 = 5 \times 29$

Logo $\phi(35) = 4 \times 6 = 24$ e 123 e 35 são primos entre si.

$$123^{145} \pmod{35} = 123^{1+6\times 26} \pmod{35} = 123 \times (123^6)^{26} \pmod{35} = 123 \times ((123^6)^{26} \pmod{35}) \pmod{35}$$

$$= 123 \times 1 \pmod{35} = 123 \pmod{35} = 18$$

03) Como o valor do expoente 2 é muito pequeno é possível uma mensagem ser elevada ao quadrado e não ser afetada pelo $modn$, o que pode fazer com que a cifra seja decifrada só com uma raiz quadrada, o que pode ser resolvido com padding, além de ocorrer os problemas listados na resposta do exercício 01.

04.01) Testando multiplicidade com todos os números primos anteriores à raiz deste.

04.02) $O(\sqrt{n})$, sendo **n** o número que deve ser testado.

04.03) Não é viável no caso da criptografia, pois ela usa da fórmula $n = 2^b$, sendo **b** o tamanho da chave. Logo, $O(\sqrt{2^b}) = O(2^{\frac{b}{2}})$, que é uma exponencial inviável, ainda mais com $b > 1024$.

05) Pois $mdc(e, \phi(n)) = mdc(e, (p-1)(q-1)) = 1$, sendo que p e q são ímpares, por serem primos muito grander, o que significa que $(p-1)(q-1)$ é um número par, logo $mdc(e, 2k) = 1$, assim e tem que ser ímpar.

06) Temos que

$$1 - (\frac{1}{4})^n \geq 99.999\%$$

$$(\frac{1}{4})^n < 0.001\% = 0.00001$$

$$2^{2n} \geq 100000$$

$$2^{20} = 1048576 \geq 100000 \rightarrow \mathbf{n} = \mathbf{10}$$

07) Como temos que a distância entre 2 números primos é $\ln N$, então, temos que a chance de encontrarmos um primo é de $\frac{1}{\ln N}$, considerando a distribuição de números escolhidos como linear, teremos que a média de números escolhidos para se fazer o teste de primalidade será de $\frac{\ln N}{2}$ (que também é a média do melhor (1) e o pior ($\ln N$) caso)

08) Temos que $4 \times a \equiv 1 \mod 11$ onde a é o inverso proporcional de 4
Multiplicando os 2 lados por 4 temos que $4^2 \times a \equiv 4 \mod 11$,
Logo, pelo teorema de Fermat podemos considerar que $4 \equiv 4^{11} \equiv 4^2 \times a \mod 11$
Assim $a = 4^9 = \mathbf{262144}$

09 Como pode ser visto nos 2 exemplos a seguir com $m = 2$, houve muito menos conta no 09.02, pois nesse não há muitos bits 1, apenas 2 (o primeiro e o último), o que faz com que não haja a conta de multiplicação pela cifra original em todas as etapas, diminuindo bastante a quantidade de cálculos.

09.01) $(e, n) = (31, 35) = (b11111, 35)$

$$
\begin{aligned}
&t \leftarrow 2 \\
&i \leftarrow 3; t \leftarrow t^2 \mod 35 = 4; e_i = 1; t \leftarrow t \times c \mod 35 = 8 \\
&i \leftarrow 2; t \leftarrow t^2 \mod 35 = 29; e_i = 1; t \leftarrow t \times c \mod 35 = 23 \\
&i \leftarrow 1; t \leftarrow t^2 \mod 35 = 4; e_i = 1; t \leftarrow t \times c \mod 35 = 8 \\
&i \leftarrow 0; t \leftarrow t^2 \mod 35 = 29; e_i = 1; t \leftarrow t \times c \mod 35 = 23 \\
&\mathbf{c = 23}
\end{aligned}
$$

09.02) $(e, n) = (33, 35) = (b100001, 35)$

$$
\begin{aligned}
&t \leftarrow 2 \\
&i \leftarrow 4; t \leftarrow t^2 \mod 35 = 4; e_i = 0 \\
&i \leftarrow 3; t \leftarrow t^2 \mod 35 = 16; e_i = 0 \\
&i \leftarrow 2; t \leftarrow t^2 \mod 35 = 11; e_i = 0 \\
&i \leftarrow 1; t \leftarrow t^2 \mod 35 = 16; e_i = 0 \\
&i \leftarrow 0; t \leftarrow t^2 \mod 35 = 11; e_i = 1; t \leftarrow t \times 2 \mod 35 = 22 \\
&\mathbf{c = 22}
\end{aligned}
$$

10) Podemos ter os números rsa de

$$
\begin{aligned}
&p = 11 \\
&q = 13 \\
&n = 143 = b10001111(8bits) \\
&\phi(n) = 120 \\
&e = 17 \\
&d = 113 = b1110001(113 \times 17 \equiv 1 \mod (120))
\end{aligned}
$$

Assim temos que a cifra de uma palavra $m = 2$ é **84**
Assim, pelo CRT temos que

$$
\begin{aligned}
d_p &= d \mod (p-1) = 3 \\
d_q &= d \mod (q-1) = 5 \\
c_p &= c \mod p = 7 \\
c_q &= c \mod q = 6 \\
m_p &= c_p^{d_p} \mod p = 2 \\
m_q &= c_q^{d_q} \mod q = 2 \\
p'_q &= p^{-1} \mod q = 6 \\
q'_p &= q^{-1} \mod p = 6
\end{aligned}
$$

Logo

$$
\begin{aligned}
m &= (q \times q'_p) \times m_p + (p \times p'_q) \times m_q \mod n \\
&= (13 \times 6) \times 2 + (11 \times 6) \times 2 \mod 143 \\
&= 156 + 132 \mod 143 \\
&= \mathbf{2}
\end{aligned}
\tag{1}
$$

Enquanto pelo Square-and-Multiply temos que

$$
\begin{aligned}
&t \leftarrow 84 \\
&i \leftarrow 5; t \leftarrow t^2 \mod 143 = 49; e_i = 1; t \leftarrow t \times c \mod 143 = 112 \\
&i \leftarrow 4; t \leftarrow t^2 \mod 143 = 103; e_i = 1; t \leftarrow t \times c \mod 143 = 72 \\
&i \leftarrow 3; t \leftarrow t^2 \mod 143 = 36; e_i = 0 \\
&i \leftarrow 2; t \leftarrow t^2 \mod 143 = 9; e_i = 0 \\
&i \leftarrow 1; t \leftarrow t^2 \mod 143 = 81; e_i = 0 \\
&i \leftarrow 0; t \leftarrow t^2 \mod 143 = 126; e_i = 1; t \leftarrow t \times c \mod 143 = 2 \\
&\mathbf{c = 2}
\end{aligned}
$$

Pode ser visto que, embora o CRT seja um algoritmo com mais etapas, suas contas são muito mais simples, apenas possuindo exponenciação 2 vezes, e não b vezes (sendo b o número de bits).