Man in the middle-Write-up

Vulnerabilitatea sistemului consta in faptul ca aplicatia client nu sesizeza modificarea unui bit din exponentul public. Aceasta vulnerabilitate permite lansarea unui atac specific pt. RSA: Common Modulus. Atacatorul are acces la 2 criptogame ale aceluiasi plaintext, obtinute prin criptarea cu acelasi modul, dar cu exponenti publici diferiti :

$C_1 = E_{(n,e1)}(M)$

$C_2 = E_{(n,e2)}(M)$

Metoda de atac:

Conform teoremei Bezout, si daca gcd(e1,e2) = 1, exista intregii x,y a.i. :

$$xe_1 + ye_2 = 1$$

Coeficientii se pot determina dolosind algoritmul extins al lui Euclid.

Acest rezultat ne permite determinarea plaintext-ului dupa o relatie matematica simpla, folosind ridicarea la putere si inmultirea modulo:

$$\begin{aligned} C_1^x * C_2^y &= (M^{e_1})^x * (M^{e_2})^y \\ &= M^{e_1 x} * M^{e_2 y} \\ &= M^{e_1 x + e_2 y} \\ &= M^1 \\ &= M \end{aligned}$$

Algoritmul lui Euclid furnizeaza un coeficient negativ: sa pp. *y = -a.* In acest caz:

$$\begin{aligned} C_2^y &= C_2^{-a} \\ &= (C_2^{-1})^a \\ &= \left(\frac{1}{C_2}\right)^{-y} \end{aligned}$$

Se observa necesitatea ca C_2 sa fie inversabil modulo *n,* deci se impune ca $\gcd(C_2,n) = 1$.

Pt. a implementare acestui scenariu am folosit libraria OpenSSL 1.1.1a x64, lucrul cu big-number. Cheia publica a fost extrasa in prealabil, folosind utilitarul OpenSSL(in linie de comanda). Ciphertext-urile sunt citite din fisierele c1.in si c2.in, modulul din fisierul modul.in, iar rezultatul este salvat in format hexazecimal in fisierul fout.out.