# Side channel attack against the Mbed TLS implementation of the RSA algorithm.

## Presentació

Victor Micó Biosca

Escola Politècnica Superior
Universitat de Girona

22 de Juny de 2023

Universitat de Girona

# Taula de Continguts

# Breu història de la criptografia


- 4000 AC Jeroglífics a Egipte
- 50 AC Xifra de Cèsar
- 1553 Xifra de Vigenère
- 1941 Alan Turing Desxifra la màquina enigma
- 1976 Es publica el algoritme de xifra simètric DES
- 1976 Diffie i Hellman introdueixen l'Intercanvi de claus publica i privada
- 1978 Publicació del sistema de xifrat de clau publica RSA (Rivest, Shamir i Adleman)

# RSA - Primitives criptogràfiques

- **Xifrat**: $c = m^e \mod n$ on $m$ és el missatge, $e$ és la clau pública i $c$ és el text xifrat (*ciphertext*).
- **Desxifrat**: $m = c^d \mod n$ on $c$ és el text xifrat, $d$ és la clau privada i $m$ és el missatge.
- **Firma**: En el procés de signatura, l'autor del missatge utilitza la seva clau privada per generar una firma $s = m^d \mod n$.
- **Verificació de la firma**: La firma $s$ d'un missatge $m$ es verifica computant $m' = s^e \mod n$. Si $m = m'$, llavors la firma és valida.

# RSA - Procés de generació de la clau RSA

1. Es generen dos nombres primers, $p$ i $q$ grans, distints i amb una longitud en bits similar.
2. Es calcula el mòdul $n = p \cdot q$
3. Es calcula el totient de n, i.e. $\varphi(n) = (p-1) \cdot (q-1)$
4. Es tria un nombre enter positiu que sigui coprimer amb $\varphi(n)$ i que compleixi $1 < e < \varphi(n)$. El parell $(n, e)$ serà la clau pública.
5. Es calcula l'exponent privat $d$ realitzant una operació d'aritmètica modular anomenada inversa multiplicativa. Ha de satisfer que $d \cdot e \equiv 1 \mod \varphi(n)$. L'exponent $d$ serà la clau privada.

# Dispositius criptogràfics

Els dispositius criptogràfics són capaços de rebre un missatge a través d'una interfície, xifrar el contingut del missatge i transmetre el missatge xifrat. Generalment, també són capaços de fer l'operació a la inversa: rebre un missatge xifrat, desxifrar-lo i transmetre el missatge en text pla.

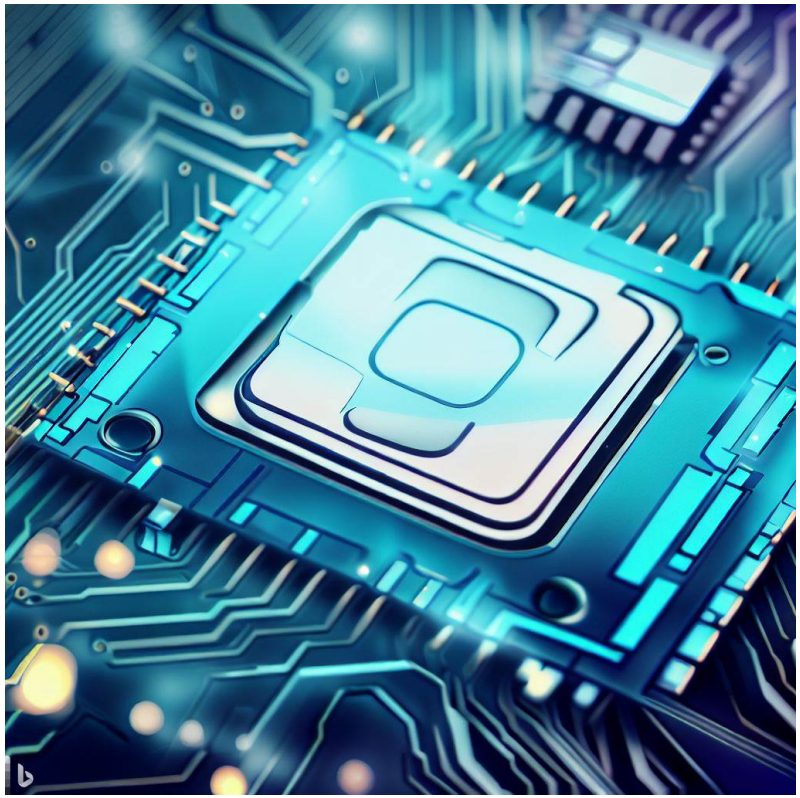

Figure: Representació d'un dispositiu criptogràfic creada amb Dall-E

# Atacs a dispositius criptogràfics

- **Atacs actius**: Un atac actiu consisteix a manipular els *inputs* o l'entorn del dispositiu amb l'objectiu que funcioni de forma errònia o diferent de les condicions normals. Amb la injecció de faltes (*fault injection*) és possible fer passar un PIN dolent per bo o extreure claus criptogràfiques, entre d'altres.
- **Atacs passius**: En un atac passiu, l'atacant extreu informació del dispositiu a través de canals laterals (*side-channel*) mentre que el dispositiu funciona en condicions normals. Aquests canals laterals poden ser el consum elèctric, la radiació electromagnètica o, fins i tot, el so o la temperatura.

# Algoritmes d'exponenciació modular

L'exponenciació modular és la operació més important del RSA.
L'algoritme més bàsic per a computar $m^e$ consisteix a multiplicar $m$ per si mateix $e$ vegades, i.e. $m \cdot m \cdot \ldots \cdot m$. Per a una clau de 1024 bits això suposaria:

$$2^{1024} > 2^{300}$$

Nombre d'operacions $>$ Nombre estimat d'àtoms a l'univers.

# Algoritmes d'exponenciació modular

**Algorithm** Left-to-right binary exponentiation

**Require:** $m$ as message
**Require:** $(e = (e_t e_{t-1} \dots e_1 e_0)_2)$ for $e_i \in (0, 1)$
**Ensure:** $m^e$

1: $A \leftarrow 1$
2: **for** $i \leftarrow t$ to 0 **do**
3: $\quad A \leftarrow A \cdot A$ {Square}
4: $\quad$ **if** $e_i = 1$ **then**
5: $\quad\quad A \leftarrow A \cdot m$ {Multiply}
6: **return** $A$

# Atacs de canal lateral

## SPA: Simple Power analysis

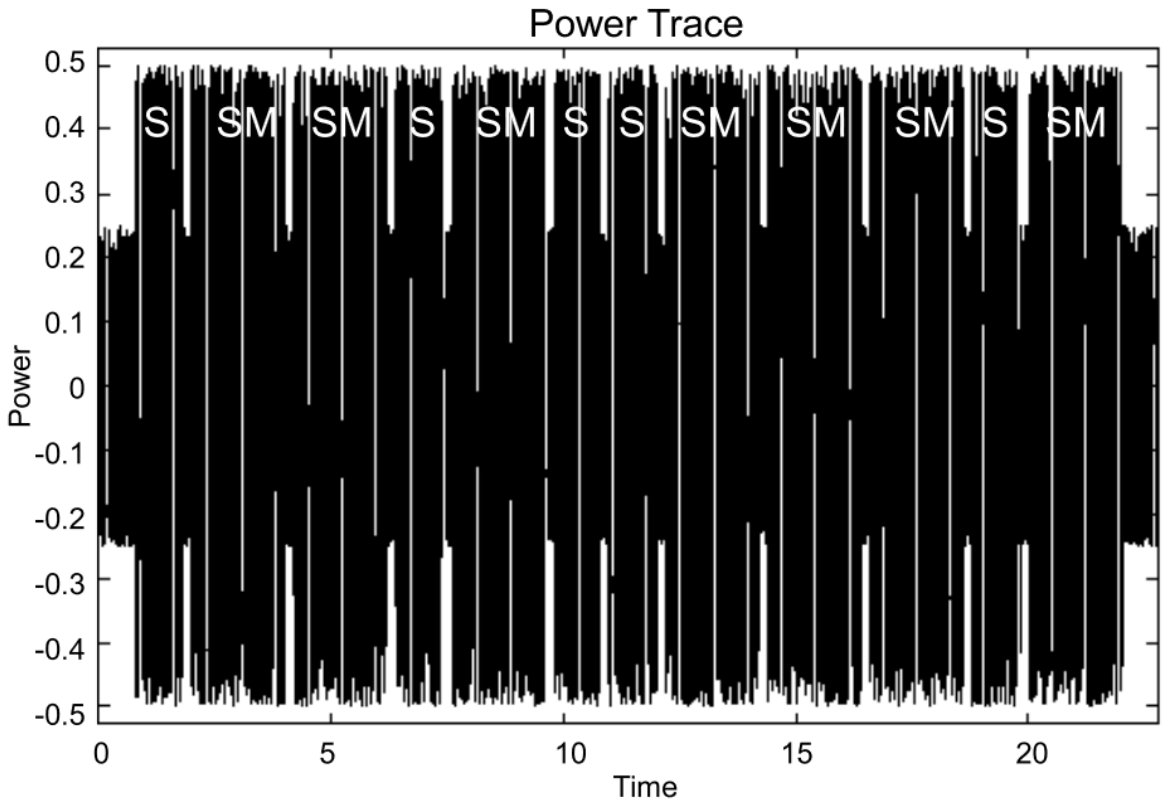


Figure: Traça de potència de RSA

# Algoritmes d'exponenciació modular

**Algorithm** Left-to-right multiply always binary exponentiation

**Require:** $m$ as message
**Require:** $(e = (e_t e_{t-1} \dots e_1 e_0)_2)$ for $e_i \in (0, 1)$
**Ensure:** $m^e$

1: $A \leftarrow 1$
2: **for** $i \leftarrow t$ to 0 **do**
3: $\quad A \leftarrow A \cdot A$ {Square}
4: $\quad$ **if** $e_i = 1$ **then**
5: $\quad\quad A \leftarrow A \cdot m$ {Multiply}
6: $\quad$ **else**
7: $\quad\quad T \leftarrow A \cdot m$ {Multiply and discard}
8: **return** $A$

# Algoritmes d'exponenciació modular

**Algorithm** Left-to-right k-ary exponentiation

**Require:** $m$ as message
**Require:** $(e = (e_t e_{t-1} ... e_1 e_0)_b)$ for $e_i$ where $b = 2^k$ for some $k > 1$
**Ensure:** $m^e$

1: $m_0 \leftarrow 1$
2: **for** $i \leftarrow 1$ to $(2^k - 1)$ **do**
3: $\quad m_i \leftarrow m_{i-1} \cdot m$ {Thus $m_i = m^i$}
4: $A \leftarrow 1$
5: **for** $i \leftarrow t$ to 0 **do**
6: $\quad A \leftarrow A^{2^k}$ {k Squares}
7: $\quad A \leftarrow A \cdot m_{e_i}$ {Multiply}
8: **return** $A$

# Algoritmes d'exponenciació modular

**Algorithm** Sliding-window exponentiation

**Require:** $m$ as message
**Require:** $(e = (e_t e_{t-1} ... e_1 e_0)_2)$ with $e_t = 1$ and integer $k \geq 1$
**Ensure:** $m^e$

1: $m_1 \leftarrow m$
2: $m_2 \leftarrow m^2$
3: **for** $i \leftarrow 1$ to $(2^{k-1} - 1)$ **do**
4: $\quad m_{2i+1} \leftarrow m_{2i-1} \cdot m_2$
5: $A \leftarrow 1$
6: $i \leftarrow t$
7: **while** $i \geq 0$ **do**
8: $\quad$ **if** $e_i = 0$ **then**
9: $\quad\quad A \leftarrow A \cdot A$ {Square}
10: $\quad\quad i \leftarrow i - 1$
11: $\quad$ **else** {Find the longest bitstring $e_i e_{i-1} \ldots e_l$ such that $i - l + 1 \geq k$ }
12: $\quad\quad A \leftarrow A^{i-l+1}$ {k Squares}
13: $\quad\quad A \leftarrow A \cdot m_{(e_i e_{i-1} \ldots e_l)2}$ {Multiply}
14: $\quad\quad i \leftarrow l - 1$
15: **return** $A$

# Atacs de canal lateral

## CPA: Correlation Power Analysis

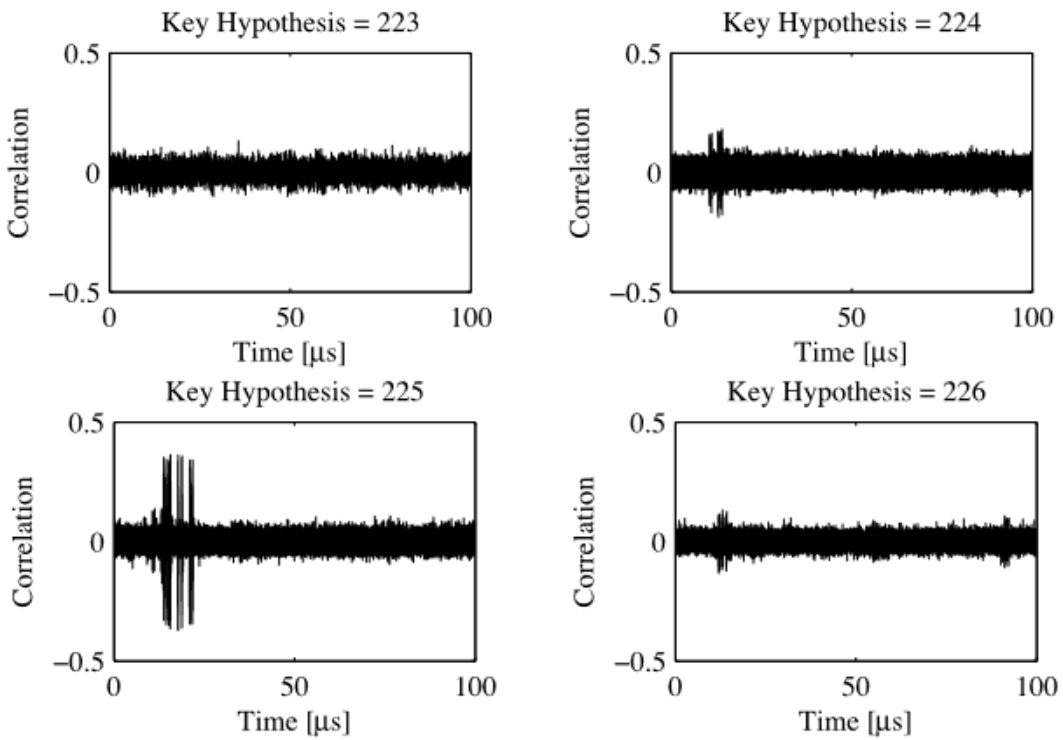


Figure: Resultat de CPA per a diferents hipòtesis de valor intermedi

# Contramesures contra Side-channel aplicades a RSA

## Ofuscació de l'exponent

Els atacs de CPA ataquen l'exponent, que és fix en múltiples traces. Per a evitar-ho, és possible ofuscar l'exponent en cada nova execució afegint-hi una màscara additiva. L'exponent secret és aleatoritzat utilitzant la següent equació:

$$d' \leftarrow d + r \cdot \phi(n)$$

On $r$ és un nombre aleatori i $\phi(n)$ és el totient d'Euler aplicat al mòdul $n$. Utilitzant l'exponent ofuscat s'obté el mateix missatge xifrat, i.e. $m^d \equiv m^{d'}$.

# Contramesures contra Side-channel aplicades a RSA

## Ofuscació del missatge

Els atacs de CPA també aprofiten que es pot controlar el missatge o bé que el missatge és conegut. Per tal que això no passi, podem ofuscar el missatge abans de la xifra. Per fer-ho, es genera un nombre aleatori $r$ i amb aquest es calculen $r_1$ i $r_2$ encarregats de fer impredictible el missatge d'entrada i de corregir el resultat final respectivament:

$$r_1 = r^e \mod n$$

$$r_2 = r^{-1} \mod n$$

Llavors durant l'operació d'RSA.

$$x' = x \cdot m_1$$

$$y' = x'^d \mod n$$

$$y = y' \cdot m_2$$

# Atacs verticals vs atacs horitzontals

Atacs Verticals

- SPA
- CPA
- Template attacks
- DL-Based attacks

Atacs Horitzontals

- Big Mac attack
- Horizontal Correlation Analysis
- Cross-correlation
- Clustering Analysis

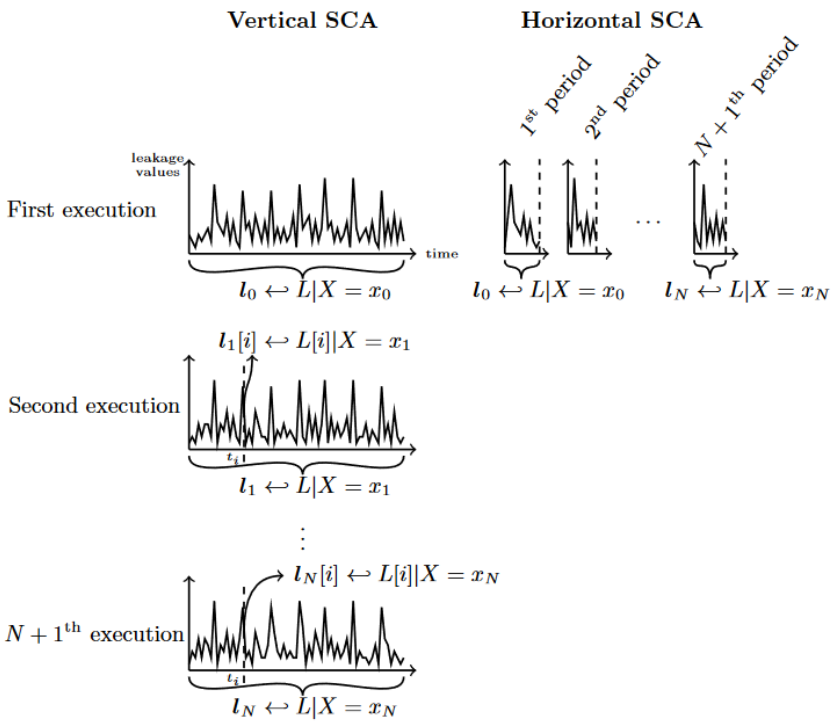


Figure: Atacs verticals i horitzontals.

# Set up

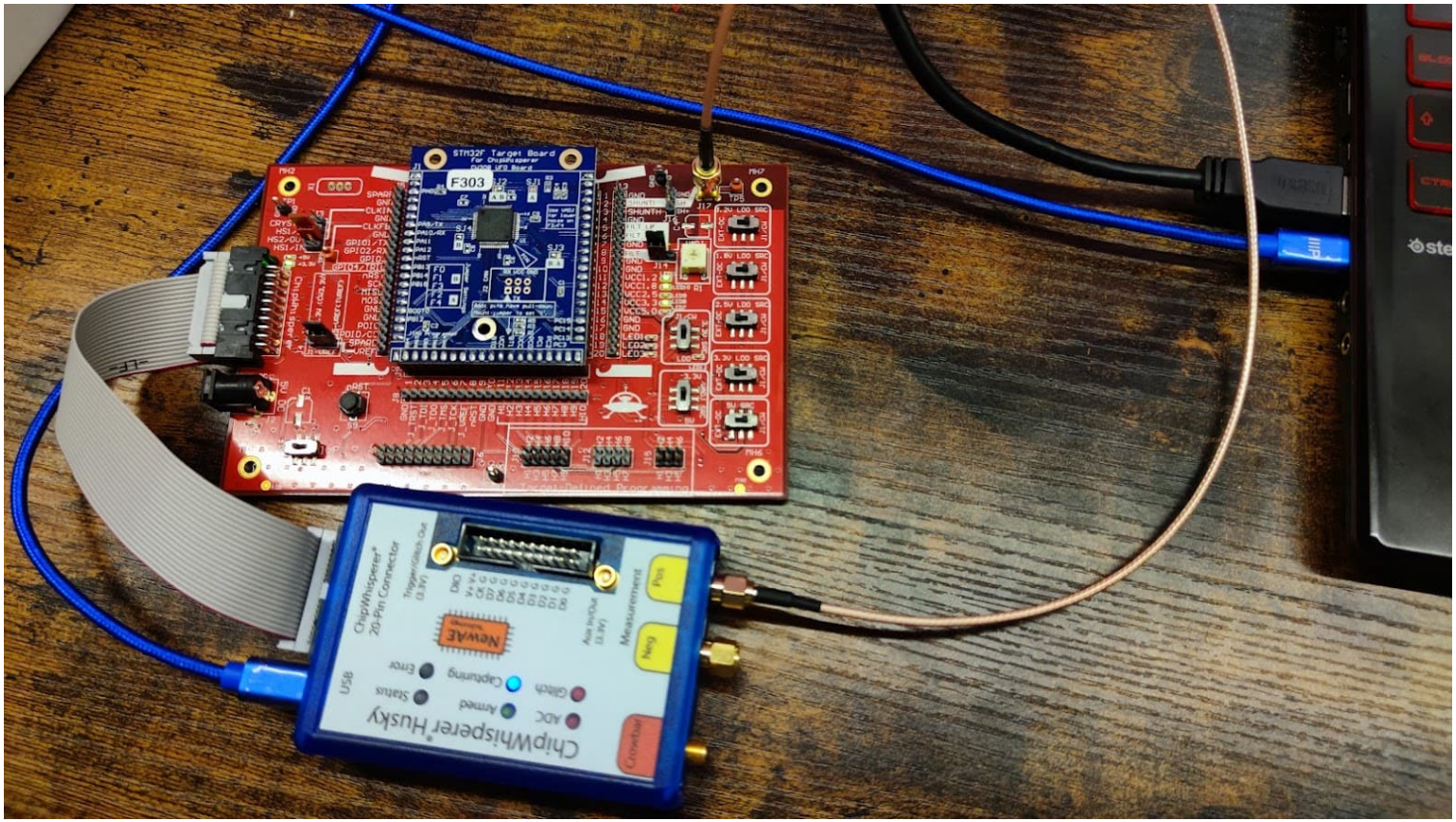


Figure: Set up

# Costos associats al projecte

| Concepte | Quantitat | Valor Unitari | Cost |
|---|---|---|---|
| Hores científic de dades | 200 | 50 €/hora | 10000 € |
| Cost d'amortització de l'ordinador | 200 | 0.11 €/hora | 22 € |
| CW308 Target base board i targets | 1 | 306 € | 306 € |
| CW Husky | 1 | 550 € | 550 € |
| Altres materials i recursos | 1 | 100 € | 100 € |
| **Total** | | | **10978 €** |

Table: Costos associats al projecte

# Mètode proposat per a l'anàlisi

1. Analitzar les diferents llibreries públiques d'RSA i seleccionar-ne una que utilitzi un algoritme d'exponenciació de finestra per a realitzar l'atac
2. Capturar una traça
3. Realitzar una anàlisi visual de la traça de potència, identificar les regions de l'RSA i les operacions modulars
4. Desenvolupar un mètode per a distingir quadrats de multiplicacions
5. Desenvolupar un mètode per a distingir els diferents valors precalculats
6. En cas d'obtenir resultats satisfactoris, informar als desenvolupadors de la llibreria de la vulnerabilitat.

# SPA

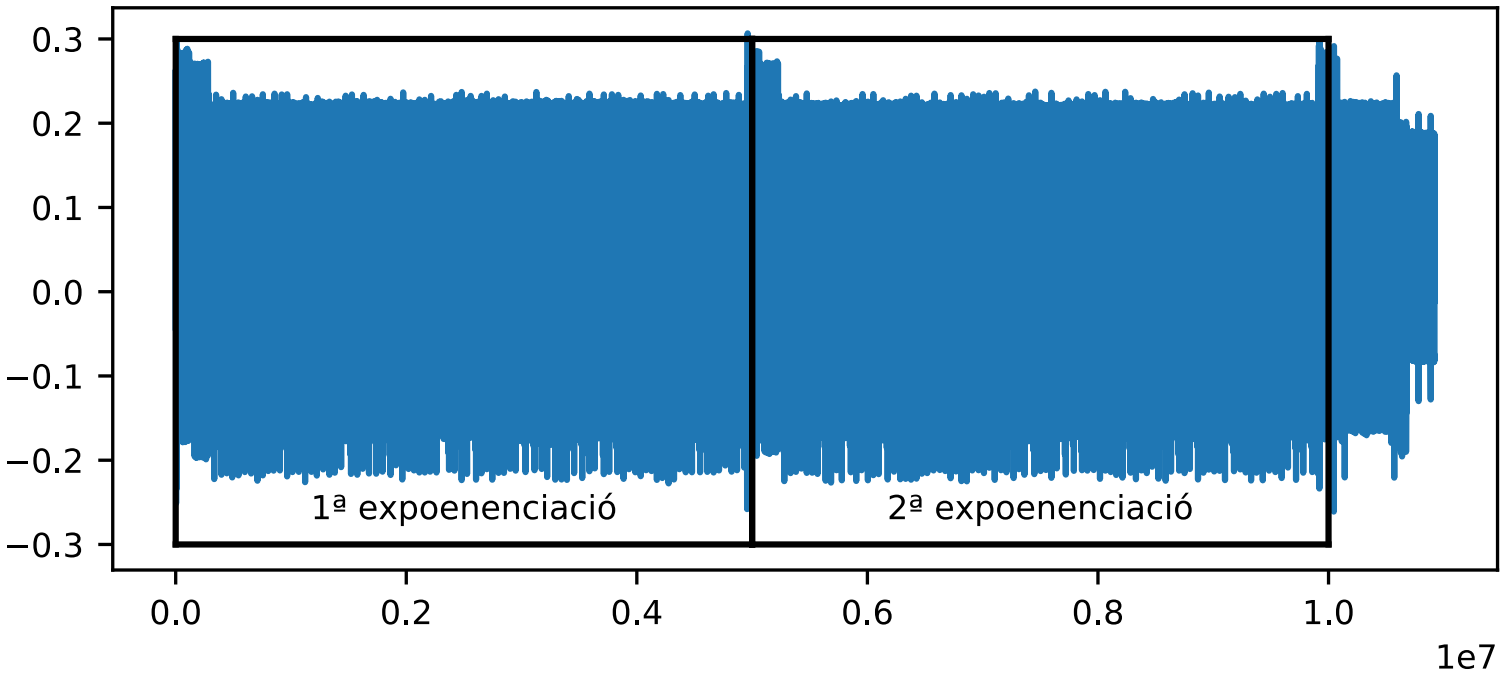


Figure: Traça RSA completa

# SPA

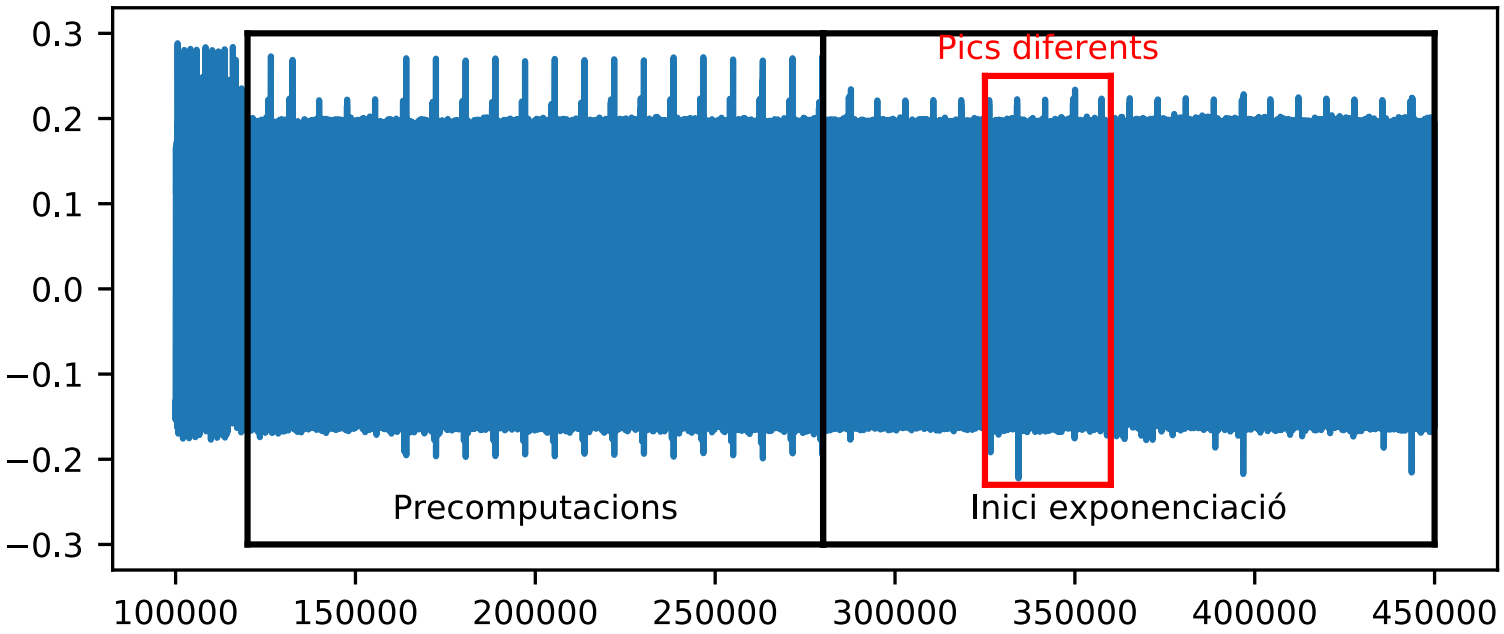


Figure: Precomputacions i inici de l'exponenciació

# SPA

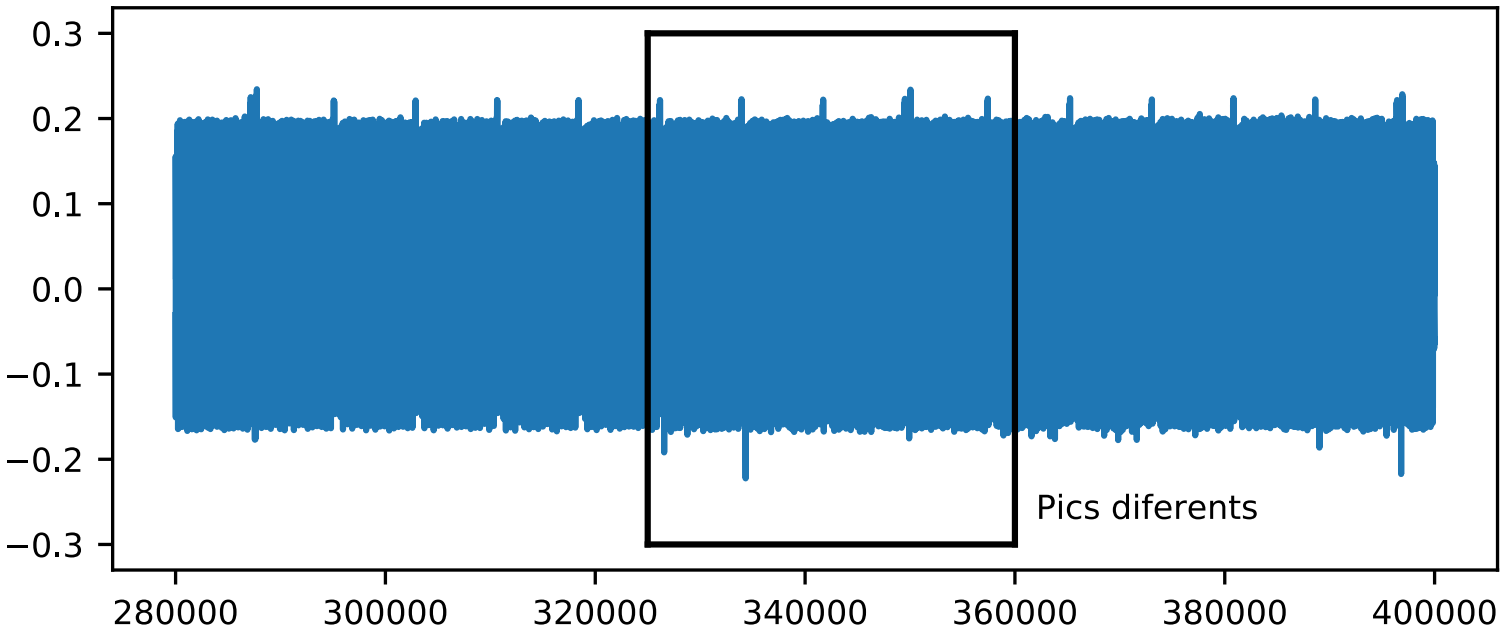


Figure: Pics diferents entre les operacions modulars

# SPA

**Algorithm** Lowpass filter

**Require:** $t$ as Trace to filter
**Require:** *weight* as Weight of the lowpass filter
**Ensure:** *result* Trace filtered

$weight_1 \leftarrow weight + 1$
$N \leftarrow length(trace)$
**for** $i \leftarrow 1 to N$ **do**
  $result[i] \leftarrow (result[i] + weight * result[i-1])/weight_1$
$i \leftarrow N - 2$
**while** $i \geq 0$ **do**
  $result[i] \leftarrow (result[i] + weight * result[i+1])/weight_1$
**return** *result*

# SPA

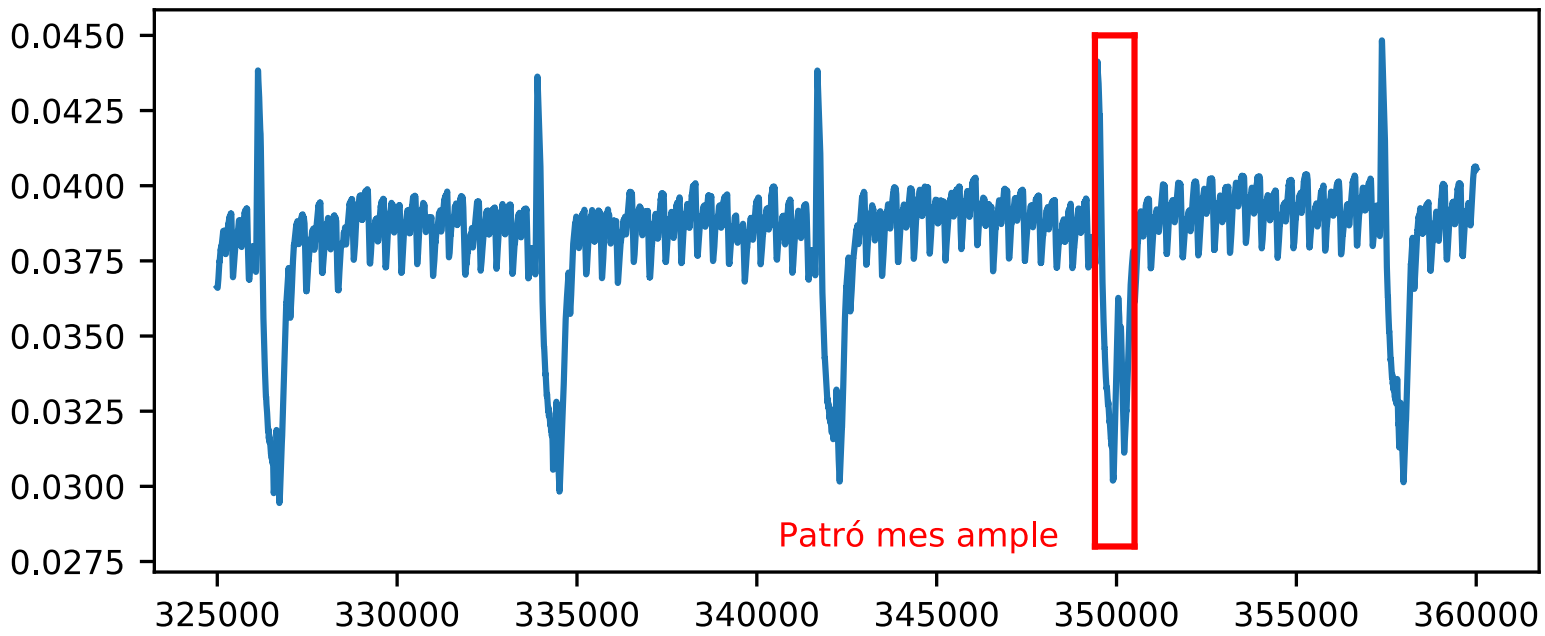


Figure: Traça filtrada amb lowpass

# Correspondència de patrons

**Algorithm** Pattern match

**Require:** $t$ as trace
**Require:** $ref$ as Reference pattern
**Ensure:** $scores$
  $N \leftarrow length(trace)$
  $n \leftarrow length(ref)$
  **for** $i \leftarrow 1 to N$ **do**
    $score[i] \leftarrow corr(ref, trace(i, i+n)$
  **return** $score$

# Correspondència de patrons: Identificació de quadrats i multiplicacions

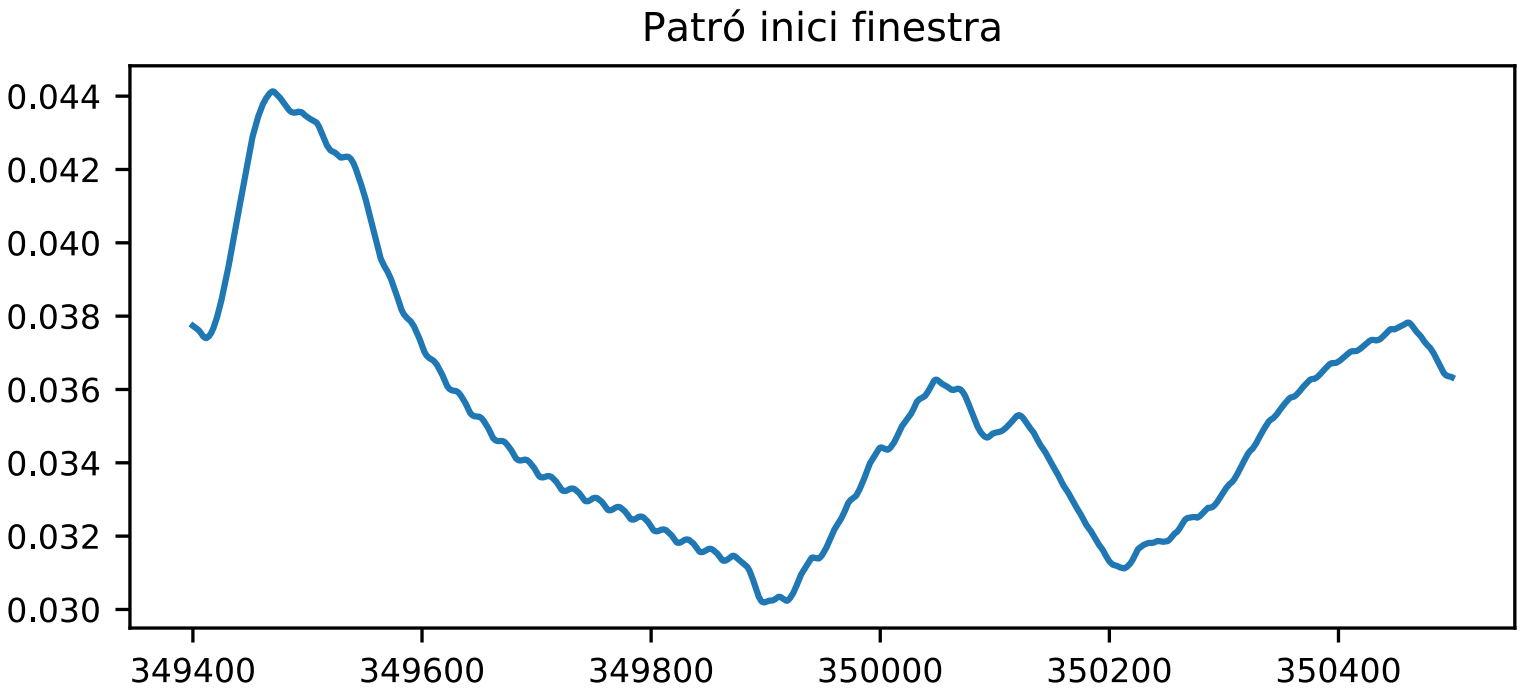


Figure: Patró inci de finestra

# Correspondència de patrons: Identificació de quadrats i multiplicacions

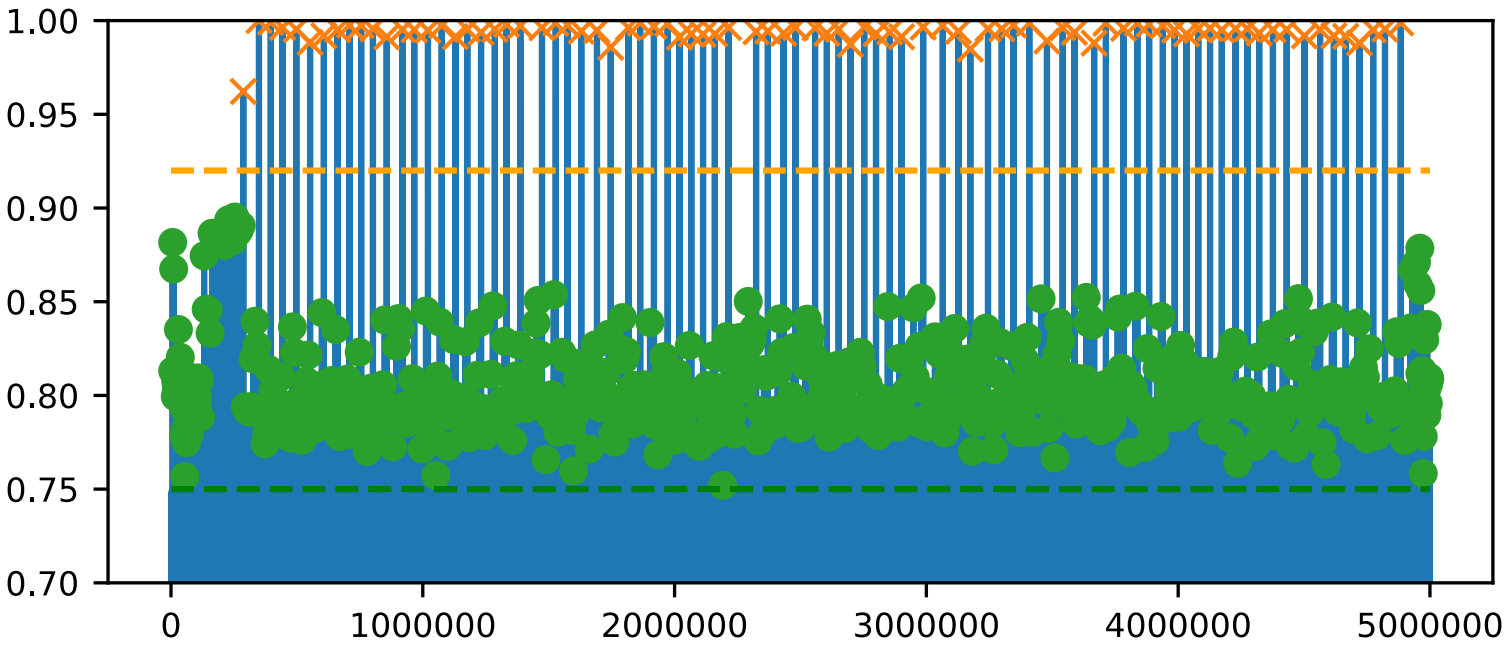


Figure: Resultat de la correspondència de patrons

# Correspondència de patrons: Identificació de quadrats i multiplicacions

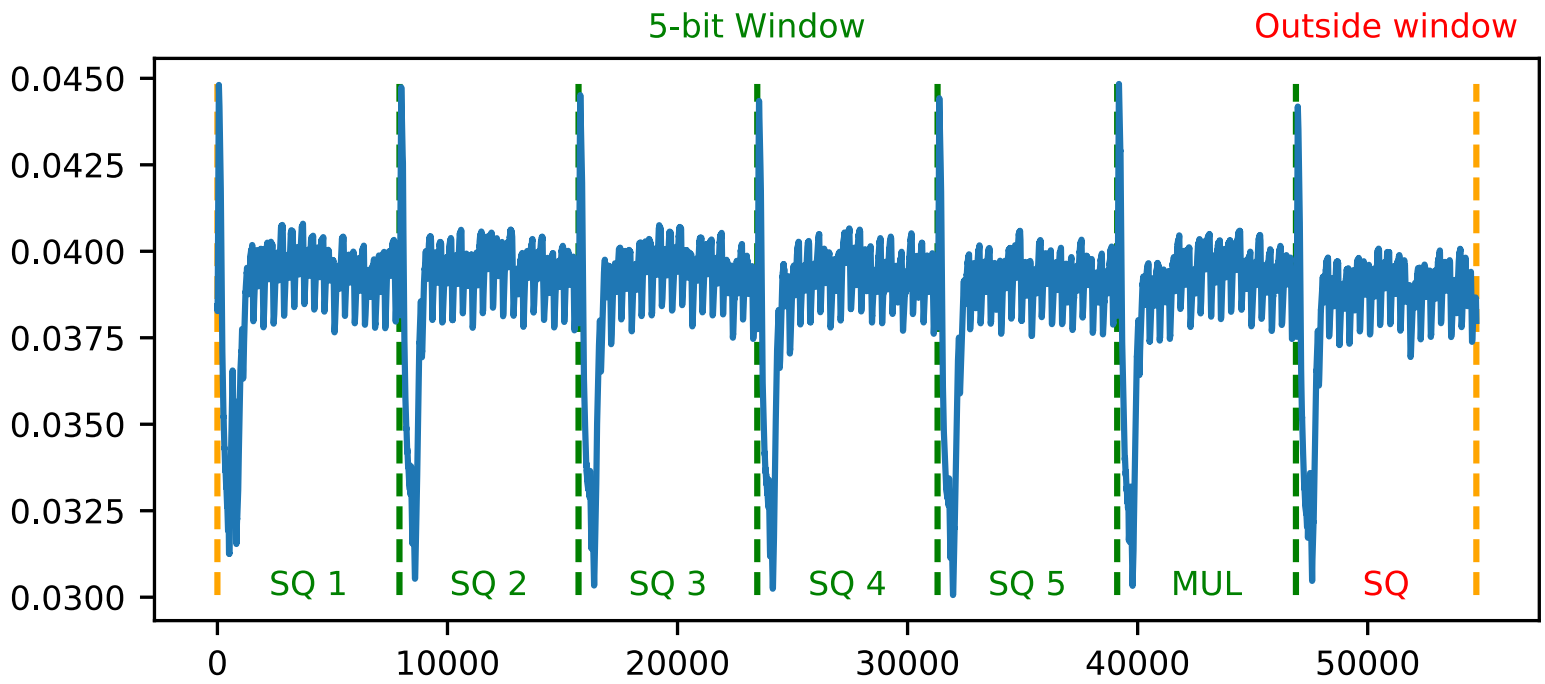


Figure: Identificació d'operacions modulars

# Correspondència de patrons: Identificació de quadrats i multiplicacions

## Bits obtinguts de la primera exponenciació

*1xxxx001xxxx1xxxx1xxxx01xxxx01xxxx01xxxx01xxxx1xxxx1xxxx1xxxx01x*
*xxx001xxxx1xxxx01xxxx01xxxx01xxxx1xxxx01xxxx01xxxx1xxxx01xxxx000*
*001xxxx01xxxx1xxxx01xxxx001xxxx01xxxx0001xxxx1xxxx01xxxx01xxxx1x*
*xxx1xxxx1xxxx1xxxx01xxxx00000001xxxx1xxxx001xxxx1xxxx00001xxxx1*
*xxxx1xxxx1xxxx01xxxx1xxxx01xxxx1xxxx000001xxxx00001xxxx001xxxx1x*
*xxx0001xxxx01xxxx1xxxx001xxxx0001xxxx001xxxx1xxxx00001xxxx1xxxx0*
*001xxxx01xxxx1xxxx01xxxx1xxxx1xxxx1xxxx1xxxx1xxxx01xxxx1xxxx1xxx*
*x01xxxx01xxxx0001xxxx001xxxx01xxxx1xxxx01xxxx01xxxx1xxxx00xxxxxx*

# Correspondència de patrons: Identificació de quadrats i multiplicacions

## Bits obtinguts de la segona exponenciació

*1xxxx01xxxx1xxxx1xxxx1xxxx1xxxx1xxxx0001xxxx1xxxx1xxxx1xxxx00001*
*xxxx1xxxx1xxxx01xxxx1xxxx1xxxx00001xxxx01xxxx01xxxx1xxxx1xxxx1xx*
*xx1xxxx1xxxx01xxxx01xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx01xxxx1xxx*
*x1xxxx1xxxx01xxxx1xxxx001xxxx1xxxx1xxxx000001xxxx01xxxx1xxxx1xxx*
*x1xxxx00001xxxx0001xxxx0001xxxx1xxxx01xxxx01xxxx1xxxx1xxxx1xxxx0*
*1xxxx0001xxxx1xxxx1xxxx1xxxx1xxxx01xxxx001xxxx1xxxx0001xxxx1xxxx*
*1xxxx0001xxxx1xxxx1xxxx1xxxx1xxxx1xxxx00001xxxx01xxxx1xxxx1xxxx1*
*xxxx01xxxx01xxxx01xxxx01xxxx01xxxx0001xxxx001xxxx00001xxxxxxxxx*

# Correspondència de patrons: Identificació de bits dins d'una finestra

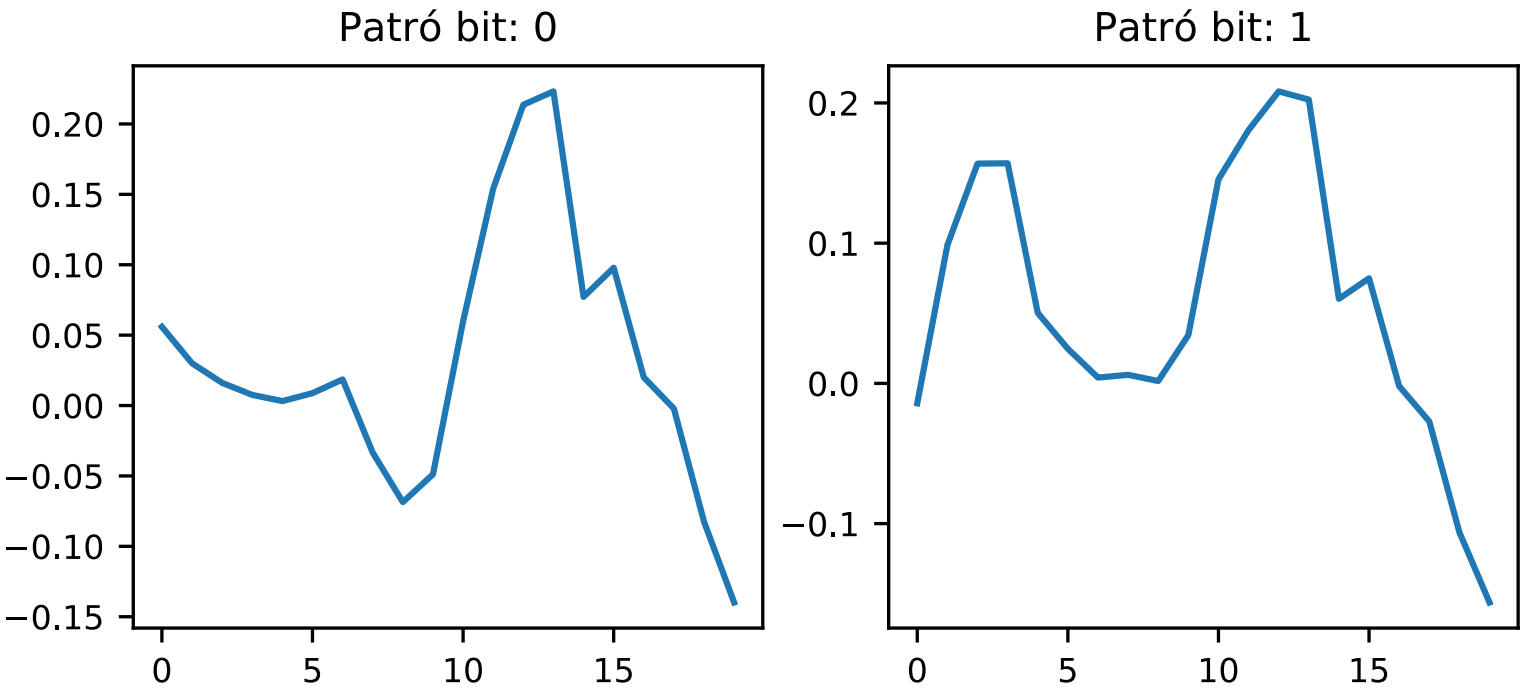


Figure: Patrons corresponents a la càrrega d'un zero i d'un u

# Correspondència de patrons: Identificació de bits dins d'una finestra

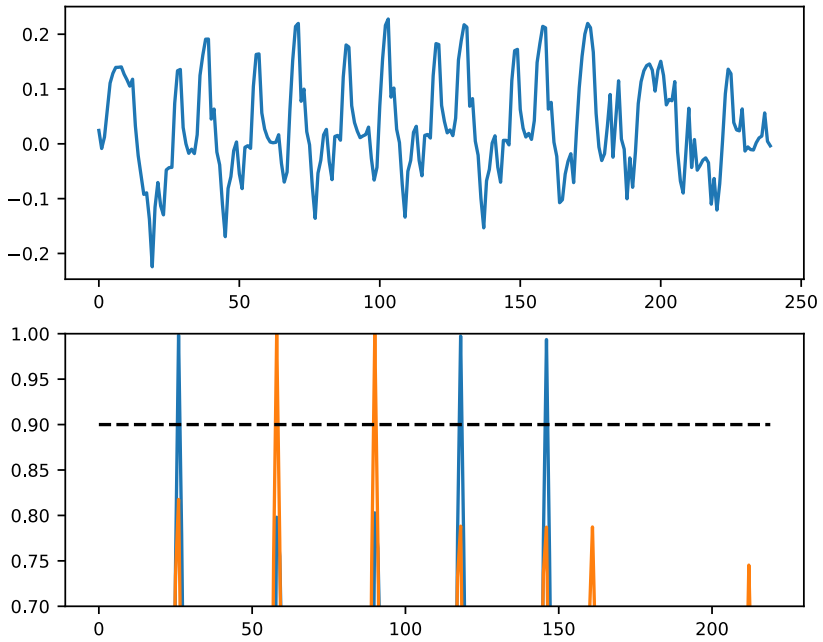


Figure: Dalt: Segment de traça corresponent a la càrrega dels bits d'una finestra Baix: Resultat de la correspondència de patrons per als bits zero i u.

# Correspondència de patrons: Identificació de quadrats i multiplicacions

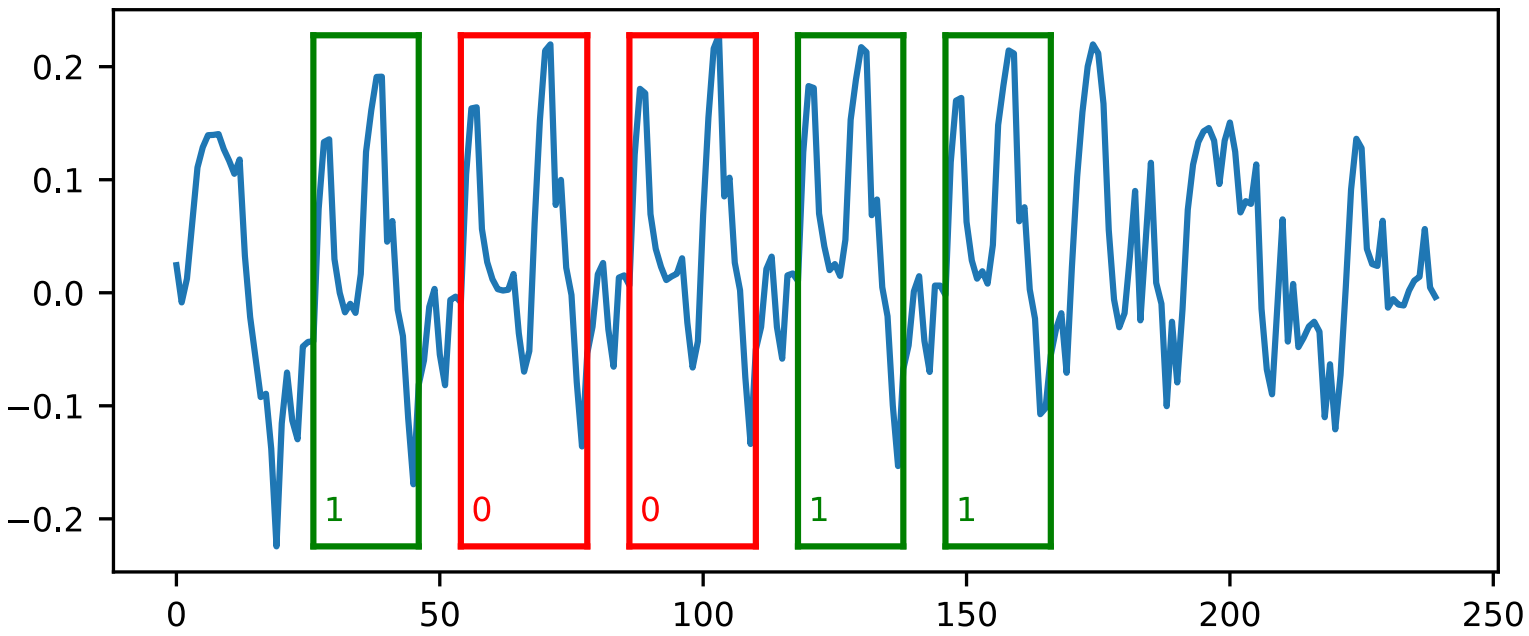


Figure: Identificació de càrrega individual de cada bit de la finestra

# Correspondència de patrons: Identificació de quadrats i multiplicacions

**Bits obtinguts de la primera exponenciació**

*1100000110101100111101010110011101010110010000100111010011111011*
*0000011110100000101110111010110101011101001001101110001010011000*
*0011110010010100110100100010001010001000101011001101011101100011*
*1111110110001110010101100000000011101101010010100111000000101101*
*1111100001000001100110001010010111110000011011000010111001110010*
*0010001101101010111001001001000011001001001111111000001111111000*
*0011110011000101010110011100110100100111101111000010110100011100*
*1010010010001000100000100010111001100001011101101010111001001 0x*

# Correspondència de patrons: Identificació de quadrats i multiplicacions

**Bits obtinguts de la segona exponenciació**

*10010010101100111001011111101101111100011111110111011010100001*
*01011101110011010010111011111000001010101010101011110011100011111*
*10110001010101011101010010101100111010010011110101000101000011000*
*01000110010011100101110010011110111110100000101100100101010111110*
*11111000001111100010011000100111100101011001001010001110001101100*
*10111000101001000111011111111100101100100101111010100010111101100*
*10000000111101000011011110101110110110000010110010010111001001111*
*10110100010110100110010101000111000001111100101100000110101010111*

# Resum de resultats

| | 1ª exponenciació | 2ª exponenciació |
|---|---|---|
| Distingir quadrats de multiplicacions | $33,98\%$ | $30,91\%$ |
| Distingir bits de cada finestra | $99,80\%$ | $100\%$ |

Table: Resum de resultats

# Conclusions

Els atacs de canal lateral son factibles, es poden realitzar amb un pressupost ajustat i amb mètodes relativament senzills de processat de senyal

Com a treball futur es proposa:

1. Actualitzar el codi de la llibreria Mbed TLS a l'última versió per comprovar si és possible explotar aquesta vulnerabilitat.
2. Provar altres dispositius *target* alternatius a l'STM32F3.
3. Utilitzar altres tècniques per extreure els valors de l'exponent, com algoritmes de *clustering*.

# MOLTES GRÀCIES!