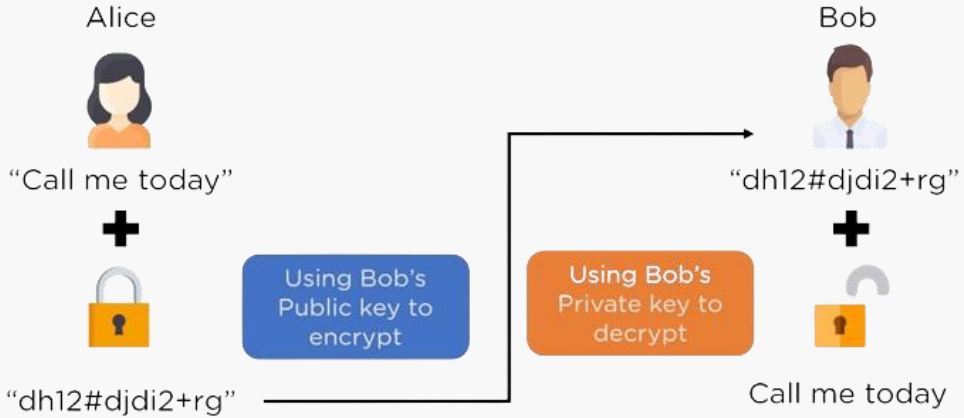
Criptografie
Algoritmul RSA

Student: Nichifor Costin

Coordonator: Lector dr. Dorin Iordache

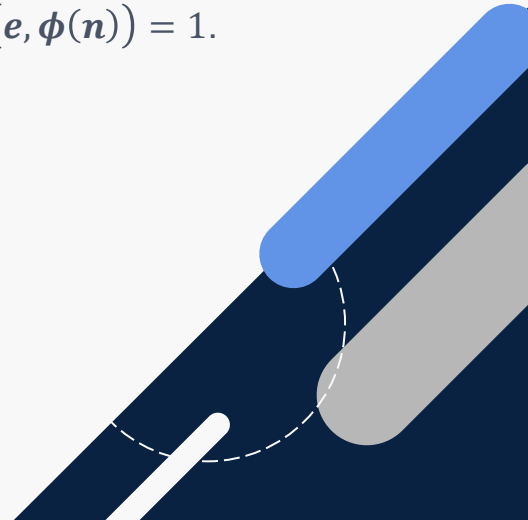
Algoritmul RSA

- Dezvoltat în anul 1977 de către **Ron Rivest**, **Adi Shamir** și **Leonard Adleman**.
- Baza fundamentală pentru criptografia cu cheie publică.
- Ideea centrală: dificultatea factorizării unui număr într-un produs de numere prime mari.



Formula pentru generarea cheilor

- Alegem 2 numere prime mari: $\boldsymbol{p}$ și $\boldsymbol{q}$.
- Calculăm $\boldsymbol{n}$ folosind formula: $\boldsymbol{n} = \boldsymbol{p} \times \boldsymbol{q}$.
- Calculăm funcția Euler: $\boldsymbol{\phi}(\boldsymbol{n}) = (\boldsymbol{p} - 1) \times (\boldsymbol{q} - 1)$.
- Alegem un exponent public astfel încât să satisfacă: $1 < \boldsymbol{e} < \boldsymbol{\phi}(\boldsymbol{n})$ și $\left(\boldsymbol{e}, \boldsymbol{\phi}(\boldsymbol{n})\right) = 1$.
- Calculăm exponentul privat: $\boldsymbol{d} \times \boldsymbol{n} \left(mod\ \boldsymbol{\phi}(\boldsymbol{n})\right) = 1$.
- Cheia publică este $(\boldsymbol{n}, \boldsymbol{e})$ și cheia privată este $(\boldsymbol{n}, \boldsymbol{d})$.



Formula pentru criptare

$$C \equiv M^e \, mod \, (n)$$

- C reprezintă mesajul criptat.
- M este mesajul de criptat.
- e este cheia publică de criptare.
- n este produsul a două numere prime p și q.

Formula pentru decriptare

$$M \equiv C^d \, mod \, (n)$$

- M reprezintă mesajul decriptat.
- C este textul criptat.
- d este cheia privată de criptare.
- n este produsul a două numere prime p și q.

Exemplu de generare

- **Generarea cheilor:**

  $p = 59$ și $q = 11$.

  $n = p \times q = 59 \times 11 = 649$.

  $\phi(n) = (p - 1) \times (q - 1) = (59 - 1) \times (11 - 1) = 580$.

  Alegem un exponent public e coprim cu 580. Să alegem $e = 17$.

  Calculăm exponentul privat d astfel încât: $d \times e \equiv 1 \left(mod\ \phi(n)\right)$. Să alegem $d = 273$.

- **Deci,**

  Cheia publică este: $(n, e) = (649,17)$.

  Cheia privată este: $(n, d) = (649,273)$.

Exemplu de criptare și decriptare

Mesaj clar: “**B**” (notat cu 66).

Bloc de mesaj: $\boldsymbol{m} = 66$.

- **Criptarea** pentru blocul $\boldsymbol{m}$ folosind cheia publică $(\boldsymbol{n}, \boldsymbol{e}) = (649, 17)$:

$$\boldsymbol{m} = 66, \ \boldsymbol{C} \equiv 66^{17} \ (mod \ 649) \approx 341$$

Astfel, mesajul "**B**" = [66] devine [341].

- **Decriptarea** pentru blocul $\boldsymbol{C}$ folosind cheia privată $(\boldsymbol{n}, \boldsymbol{d}) = (649, 273)$:

$$\boldsymbol{C} = 341, \qquad \boldsymbol{m} \equiv 341^{273} (mod \ 649) \ \approx 66$$

Prin urmare, mesajul criptat [341] este decriptat în "**B**" = [66].

Exemplu folosind Python

- Fie $\boldsymbol{p} = 277$ și $\boldsymbol{q} = 397$.
- $\boldsymbol{n} = \boldsymbol{p} \times \boldsymbol{q} = 277 \times 397 = 109969$.
- $\boldsymbol{\phi}(\boldsymbol{n}) = (\boldsymbol{p} - 1) \times (\boldsymbol{q} - 1) = (277 - 1) \times (397 - 1) = 276 \times 396 = 109296$.
- $\boldsymbol{e} = 17$ (Verifică $1 < \boldsymbol{e} < \boldsymbol{\phi}(\boldsymbol{n})$ și $\left(e,\ \phi(n)\right) = 1$.
- Cheia publică este: $(\boldsymbol{n}, \boldsymbol{e}) = (109969, 17)$.
- Cheia privată este: $(\boldsymbol{n}, \boldsymbol{d}) = (109969, 70721)$.
- Mesajul pe care vreau să îl criptez este: “comunicari matematice”.

```
Mesajul necriptat: comunicari matematice
p: 277
q: 397
n: 109969
phi(n): 109296
e: 17
Cheia publică: (109969, 17)
Cheia privată: (109969, 70721)
Mesajul separat pe litere: ['c', 'o', 'm', 'u', 'n', 'i', 'c', 'a', 'r', 'i', ' ', 'm', 'a', 't', 'e', 'm', 'a', 't', 'i', 'c', 'e']
Reprezentarea în formmat ASCII a fiecărei litere: [99, 111, 109, 117, 110, 105, 99, 97, 114, 105, 32, 109, 97, 116, 101, 109, 97, 116, 105, 99, 101]
Pașii de criptare:
   [99^17 (mod 109969) = 93859]
   [111^17 (mod 109969) = 48130]
   [109^17 (mod 109969) = 29093]
   [117^17 (mod 109969) = 103205]
   [110^17 (mod 109969) = 86012]
   [105^17 (mod 109969) = 49694]
   [99^17 (mod 109969) = 93859]
   [97^17 (mod 109969) = 57205]
   [114^17 (mod 109969) = 3777]
   [105^17 (mod 109969) = 49694]
   [32^17 (mod 109969) = 47789]
   [109^17 (mod 109969) = 29093]
   [97^17 (mod 109969) = 57205]
   [116^17 (mod 109969) = 10409]
   [101^17 (mod 109969) = 4295]
   [109^17 (mod 109969) = 29093]
   [97^17 (mod 109969) = 57205]
   [116^17 (mod 109969) = 10409]
   [105^17 (mod 109969) = 49694]
   [99^17 (mod 109969) = 93859]
   [101^17 (mod 109969) = 4295]
Mesajul criptat: [93859, 48130, 29093, 103205, 86012, 49694, 93859, 57205, 3777, 49694, 47789, 29093, 57205, 10409, 4295, 29093, 57205, 10409, 49694, 93859, 4295]
```

```
Mesajul criptat: [93859, 48130, 29093, 103205, 86012, 49694, 93859, 57205, 3777, 49694, 47789, 29093, 57205, 10409, 4295, 29093, 57205, 10409, 49694, 93859, 4295]
Pașii de decriptare:
   [93859^70721 (mod 109969) = 99]
   [48130^70721 (mod 109969) = 111]
   [29093^70721 (mod 109969) = 109]
   [103205^70721 (mod 109969) = 117]
   [86012^70721 (mod 109969) = 110]
   [49694^70721 (mod 109969) = 105]
   [93859^70721 (mod 109969) = 99]
   [57205^70721 (mod 109969) = 97]
   [3777^70721 (mod 109969) = 114]
   [49694^70721 (mod 109969) = 105]
   [47789^70721 (mod 109969) = 32]
   [29093^70721 (mod 109969) = 109]
   [57205^70721 (mod 109969) = 97]
   [10409^70721 (mod 109969) = 116]
   [4295^70721 (mod 109969) = 101]
   [29093^70721 (mod 109969) = 109]
   [57205^70721 (mod 109969) = 97]
   [10409^70721 (mod 109969) = 116]
   [49694^70721 (mod 109969) = 105]
   [93859^70721 (mod 109969) = 99]
   [4295^70721 (mod 109969) = 101]
Mesajul decriptat: comunicari matematice
```

Semnătură digitală folsind RSA

- **Semnarea cu RSA:**

  Calculăm hash-ul mesajului: $h = hash(msj)$.

  Criptăm h pentru a calcula semnătura: $s \equiv h^d (mod\ n)$.

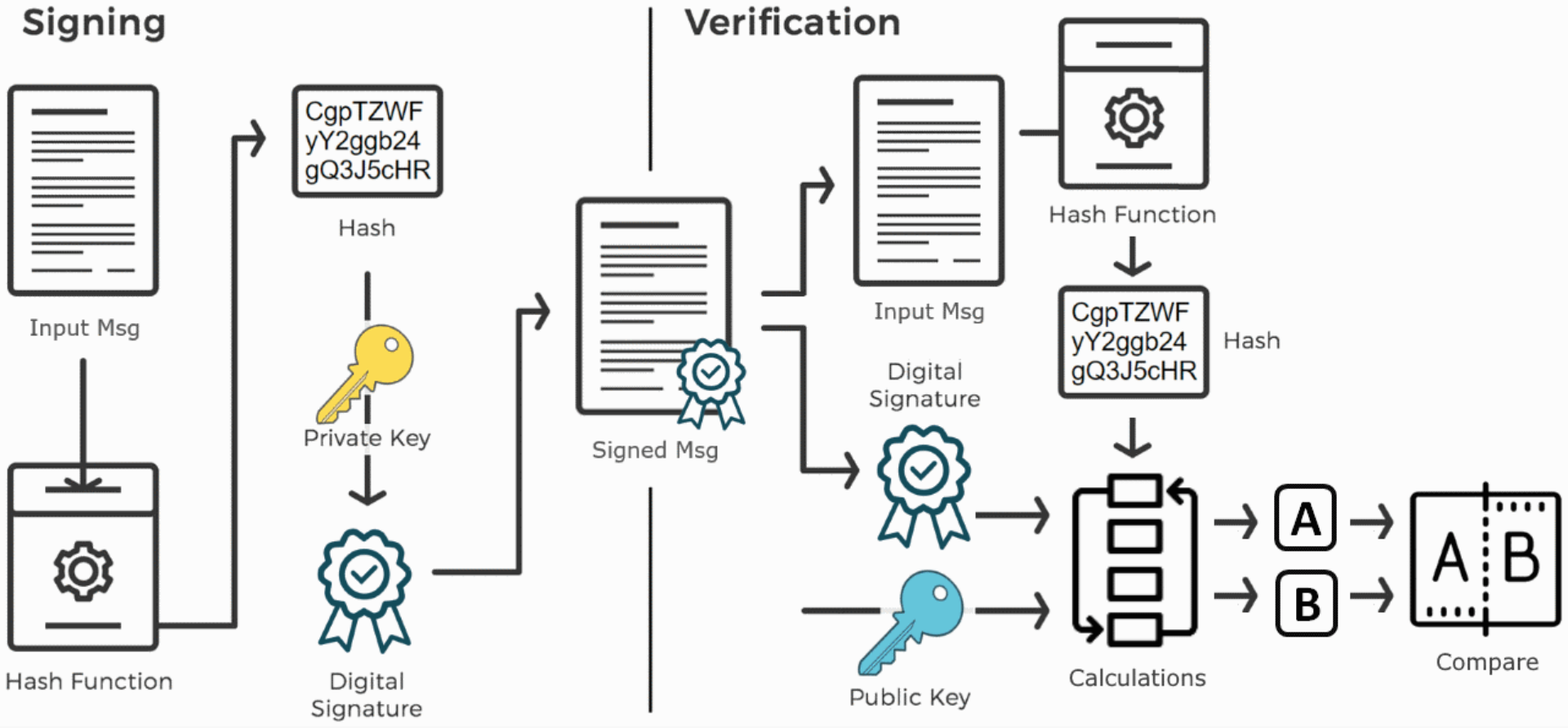
- **Verificarea semnăturii:**

  Calculăm hash-ul mesajului: $h = hash(msj)$

  Decriptăm semnătura: $h' = s^e (mod\ n)$

  Verificăm dacă $h = h'$.

  Dacă cele două hash-uri coincid, semnătura este considerată validă.


Signing
Verification
Input Msg
Hash Function
CgpTZWF
yY2ggb24
gQ3J5cHR
Hash
Private Key
Digital
Signature
Signed Msg
Input Msg
Hash Function
Digital
Signature
CgpTZWF
yY2ggb24
gQ3J5cHR
Hash
Public Key
Calculations
A
B
A B
Compare

```
Cheia publică:
  n=
   b029b1c3659151b436659a660e62f64c548eaca6dc6d5fa5ff369893377f0963
   6f1425726e09f5f670e83e17105342bd1d76273a2769815c538d132f89bf7dc7
   50f542ed268532262b08d4542cf4345c2d490238443fcdad2c1c876530276d51
   c0ee0959991b171d58fea9aacb90064c76699476a585d9daaf0c5381bb075749
  e=10001

Cheia privată:
  n=
   b029b1c3659151b436659a660e62f64c548eaca6dc6d5fa5ff369893377f0963
   6f1425726e09f5f670e83e17105342bd1d76273a2769815c538d132f89bf7dc7
   50f542ed268532262b08d4542cf4345c2d490238443fcdad2c1c876530276d51
   c0ee0959991b171d58fea9aacb90064c76699476a585d9daaf0c5381bb075749
  d=
    e6b313ab729de6c3fce0fe5c5cc6d90a8949f76ce60b0e51d3726a499149fabf
    8f3e70f3f48b72e3746590ba9f948a38241711b424dd149e16850ff5178e0166
    12265c90ab2461d401a483e7a703d0f3a2ab48344b6b9838f6a174852fddcbaf
    7dceb86a21c3b16704bb0a390fdf3c163b3fabca6cd42dc9609a6465c0e79c1

Mesajul hash-uit:
0fab5559eead530b47301772c4df4bfdc7e44e8a70e2af431e08f7623e9a2878

Semnătura digitală este:

6758294fc9b4d9e6e182023900a4da8611be3e3acc18f6b2c26bf450814cb32f
f80677ed188fb5999cb9278ff4dc4b7d18d6360ae902abc555ec8561ac94d9db
d2eebaf9f797829350038b2c5232244c0bae5f043a08bc085826b67973e142d0
632f20a87e64ed0f7b2acfe5c80e7a6f9c22c03be78e2e8f154859bfa1d59ec2

Semnătura este corectă pentru mesajul: b'Sesiunea de comunicari matematice 2023'

Semnătura nu este corectă pentru mesajul: b'Sesiunea de comunicari matematice 2022'
```

Slăbiciuni ale algoritmului RSA

- Dacă p si q sunt apropiate unul de celălalt atunci putem folosi teorema lui Fermat pentru a îl descompune pe n.
- $n = (a + b) \times (a - b)$.
- Dacă p si q sunt apropiate atunci valoarea lui a va fi apropiată de $\sqrt{n}$.
- Astfel, incrementând pe a cu 1 putem afla valoarea lui b folosind formula $b^2 = a^2 - n$.
- Dacă b este pătrat perfect atunci: $p = a + b$ și $q = a - b$.

SageMath 9.3 Console

```
sage: n = 52619338446501009084300300833980988386880181471495295334654447193855668646057815764873053567170748825058827015852977657893237262583560356927698974206208587747636941176344
....: 08028918270394852404169072671551096321238430993811080749636806153881798472848720411673994908247486124703888115308603904735959457057925225503197625820670522050494196703154086 3
....: 16062123787934777520599894745147260327060174336101658295022275013051816321617046927321006322752178354002696596328204277122466231388232487691224076847557856202947748540263791 7
....: 67128195927179588238799470987669558119422552470505956858217654904628177286026365989987106877656917
sage: n.nbits()
2046
sage: a = isqrt(n) +  1
sage: a
72539188337409048434517657668785982436503618029818802387833126880251213106684983301847459281756173872849655980341983435213476251581941251979385718844779811793903505447789359427083285619120435708725262938153683654066403635954292883105608903936926001552609343022442053757361595080026446437841528409159732216537
sage: while True:
....:     b2 = a^2 - n
....:     if is_square(b2):
....:         b = sqrt(b2)
....:         break
....:     a = a + 1
....:
sage: b
43307383642926417598009375629968592082792556530851401797151778925058527859538856484810160284683714885685721518431228682976939244025192110722719675022535522
sage: p = a + b
sage: q = a - b
sage: p
72539188337409048434517657668785982436503618029818802387833126880251213106684983301847459281756173872849655980341983435213476251581941251979385718844779855101287148374206957436458915587712518501281793789555480805845328694482152421962093714097210685267495028743960484986044572019270471629952251128834754752071
sage: q
72539188337409048434517657668785982436503618029818802387833126880251213106684983301847459281756173872849655980341983435213476251581941251979385718844779768486519862521371761417707655650528352916168732086751886502287478577426433344249124093776641317837723657300923622528678618140782421245730805689484709681027
sage: p.is_prime()
True
sage: q.is_prime()
True
sage: n == p * q
True
sage: e = 65537
sage: phi_n = (p-1)*(q-1)
sage: d = inverse_mod(e, phi_n)
sage: d
17800185137539517698383167821068381411067543879381886530803806922620507101074229450649794735556639782894459541044913784205762700634413127135619363995172870139998946944162049834615263843916840933875692858660823940066700152838887872193178137460082557354094147194274355977140345941913177555971130235006633535597630270496599409229331971771651884528189042351791560236165094346800978984741521873968515101369848343304881978375861938273241248940334851915021105752020805341847108491130526886356953308720044791902966786160476361483730986540024432735291754733367533545745958771610232827728458139639945653948706933236494949596 73
sage:
```

Dezvoltarea Calculatoarelor Cuantice:

- **Procesare paralelă exponențială**: Calculatoarele cuantice utilizează qubiți pentru a efectua operații în mod paralel, oferind o creștere exponențială a capacității de procesare.
- **Factorizare rapidă**: Algoritmii cuantici, precum Shor's Algorithm, pot factoriza numere mari într-un timp mult mai scurt decât algoritmii clasici.

RSA number	Qubits	Kn-depth	2DSL-depth	LNN-depth
RSA-128	37	113	121	150
RSA-256	64	194	204	258
RSA-512	114	344	357	458
RSA-1024	205	617	633	822
RSA-2048	372	1118	1139	1490

Bibliografie

www.asecuritysite.com

www.rowikipedia.org

https://cryptobook.nakov.com/digital-signatures/rsa-signatures

https://mathstats.uncg.edu/sites/pauli/112/HTML/secelgamal.html

https://arxiv.org/abs/2212.12372

https://www.sohamkamani.com/nodejs/rsa-encryption/

https://arxiv.org/abs/2212.12372