Tehnici de CRIPTARE

Claudiu Soroiu

În acest ultim episod al serialului dedicat tehnicilor de criptare vom prezenta cel mai cunoscut sistem de codificare cu cheie publică a mesajelor, și anume RSA; acesta este utilizat cu succes de foarte mulți ani.

Introducere

Algoritmii de criptare a mesajelor utilizând *chei publice* se folosesc atunci când un număr mare *n* de utilizatori trebuie să comunice între ei (*fiecare cu fiecare*). În cazul în care aceștia ar folosi tehnici tradiționale de criptare, atunci ar avea nevoie de $n \cdot (n - 1) / 2$ chei, spre deosebire de algoritmii de criptare cu *chei publice* care folosesc doar *n* chei.

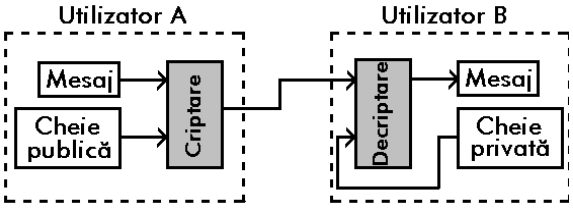
În sistemele de criptare cu *cheie publică*, dacă un utilizator dorește să comunice cu alții, trebuie să își construiască două chei:

- o *cheie publică* pe care o va transmite oricărui alt utilizator care dorește să-i trimită un mesaj criptat;
- o *cheie privată* care este secretă și pe care o va folosi la decriptarea unui mesaj criptat cu *cheia publică*.

Cele două chei se presupun a fi unice pentru fiecare utilizator.

Secretul sistemelelor de criptare cu *cheie publică* este acela că dacă un utilizator deține *cheia publică* a unui alt utilizator ale cărui mesaje dorește să le intercepteze și decripteze, de obicei, este imposibil să construiască *cheia privată* și să o folosească pentru decriptarea mesajelor. În cazul în care *cheia privată* se poate calcula pe baza *cheii publice,* atunci timpul necesar unui sistem de calcul este suficient de mare și, după decriptarea mesajului acesta nu ar mai fi de actualitate.

Dacă avem doi utilizatori **A** și **B**, și utilizatorul **A** dorește să-i trimită utilizatorului **B** un mesaj criptat, procesul de comunicare are loc așa cum indică figura următoare.



Pentru a cripta mesajul, utilizatorul **A** are nevoie de *cheia publică* a utilizatorului **B**. După criptare, mesajul este transmis utilizatorului **B** și acesta, pentru a-l putea accesa, îl decriptează folosind propria *cheie privată*.

Presupunând că avem un grup de *n* utilizatori care au nevoie să comunice între ei în siguranță și toți folosesc o aceeași tehnică de criptare cu *cheie publică*, atunci, pentru comunicare, fiecare utilizator *i* din cei *n* are nevoie de:

- *cheile publice* ale celorlalți *n* - 1 utilizatori; o *cheie publică* a unui utilizator *j* este folosită în procesul de criptare a unui mesaj de către utilizatorul *i*, urmând ca acesta să fie trimis utilizatorului *j*;
- o *cheie privată* care este folosită de către utilizatorul *i* pentru a decripta orice mesaj care sosește de la unul dintre ceilalți și a fost criptat folosindu-se *cheia publică* a utilizatorului *i*.

RSA

RSA este un sistem de criptare cu *cheie publică* dezvoltat în anul 1977 de către profesorii de la ***MIT*** (***M**assachusetts **I**nstitute of **T**echnology*) *Ronald L. Rivest*, *Adi Shamir* și *Leonard M. Adleman*, cu scopul de a asigura securitatea datelor pe *Internet*.

Generarea cheilor

Algoritmul *RSA* are la bază elemente de teoria numerelor și teoria grupurilor.

Primul pas în folosirea algoritmilor de criptare și decriptare pentru tehnica *RSA* constă în construirea celor două chei.

Pe baza unei parole (șir de caractere) introdusă de utilizator și care trebuie să aibă o lungime suficient de mare, se generează două numere prime mari (cu o lungime mai mare de 64 de biți) *p* și *q*.

După ce s-au generat cele două numere prime se construiește un număr *n* ca fiind produsul dintre *p* și *q*.

Fie $\phi = (p - 1) \cdot (q - 1)$.

La pasul următor se caută un număr *e* care trebuie să fie relativ prim cu ϕ, adică cel mai mare divizor comun al numerelor *e* și ϕ este 1.

În continuare se construiește numărul *d* ca fiind inversul numărului *e* modulo ϕ:

$$d = \texttt{rest}([e^{-1}/\phi])$$

Numerele *n* și *e* formează *cheia privată*, în timp ce *cheia publică* este dată de numerele *n* și *d*.

Pentru ca generarea cheilor să fie optimă în timp, la calcularea numărului *d* se folosește forma extinsă a algoritmului lui *Euclid*.

Algoritmul extins al lui Euclid

După cum se cunoaște, algorimul lui *Euclid* calculează cel mai mare divizor comun a două numere naturale astfel:

```
algoritm Euclid(a, b)
dacă b = 0
  atunci returnează a
  altfel returnează Euclid(b, rest([a/b]))
```

Forma extinsă a algoritmului lui *Euclid*, pe lângă cel mai mare divizor comun *d* a două numere *a* și *b*, determină două numere întregi *x* și *y* cu proprietatea $d = a \cdot x + b \cdot y$. Se poate demonstra matematic că perechea de numere *x* și *y* este unică.

Forma extinsă a algoritmului lui *Euclid* este următoarea:

```
algoritm EuclidExtins(a, b)
dacă b = 0
  atunci returnează (a, 1, 0)
  altfel
    (d', x', y') ← EuclidExtins(b, rest([a/b]))
    returnează (d', y', x' - [a/b] · y')
```

Revenind la determinarea inversului numărului *e*, se poate demonstra că pentru orice număr pozitiv ϕ relativ prim cu *e*, ecuația `rest([(e·x)/ϕ]) = 1` are soluție unică și nu are soluție în cazul în care nu sunt respectate condițiile. În acest caz avem:

`EuclidExtins(`*e*, ϕ`) = (1,` *x*, *y*`)`

În cadrul clasei de resturi modulo ϕ, avem $\phi \cdot y = 0$, de unde rezultă că *x* este inversul numărului *e*.

Criptarea și decriptarea

Fie *m* mesajul (un număr) care trebuie să fie criptat cu lungimea mai mică decât *n*.

Relația prin care se obține mesajul criptat *c*, din mesajul *m* este:

$$c = \texttt{rest}([m^e/n])$$

După cum am afirmat anterior, la decriptarea mesajului *c* se vor folosi numerele *d* și *n* astfel:

$$m = \texttt{rest}([c^d/n])$$

Folosind aceste relații se poate demonstra matematic corectitudinea algoritmului *RSA*.

Pentru ca procesele de criptare și decriptare să fie optime în timp, se folosesc diverși algoritmi performanți pentru calcularea restului.

Unul dintre cei mai utilizați astfel de algoritmi se bazează pe *teorema chineză a restului*. Mai multe detalii despre această teoremă puteți găsi în cartea ***Introducere în algoritmi***, scrisă de *Thomas H. Cormen*, *Charles E. Leiserson* și *Ronald R. Rivest*, a cărei traducere în limba română a apărut la editura *Computer Libris Agora* din *Cluj-Napoca*.

Securitatea RSA

Securitatea sistemului este asigurată de faptul că este foarte dificilă factorizarea numerelor întregi mari. În cazul în care cineva reușește să factorizeze numărul *n*, atunci poate obține *cheia privată* din *cheia publică*.

În cazul în care factorizarea numerelor mari ar fi o operație simplă, un sistem de securitate bazat pe algoritmul *RSA* ar fi foarte ușor de evitat.

Pentru a asigura o securitate ridicată a datelor utilizând *RSA*, este indicat să se folosească numere care au suficient de multe cifre cât să îngreuneze foarte mult factorizarea dar criptarea mesajelor să fie optimă în timp.

În unele cazuri, pentru o securitate mai mare, se folosește o criptare hibridă a mesajelor cu ajutorul algoritmului *RSA* și a unui alt algoritm rapid cu cheie nepublică, iar în alte cazuri se folosește o criptare dublă *RSA* și, în consecință, fiecare dintre cele două chei este formată din patru elemente: (n_1, n_2, e_1, e_2) și (n_1, n_2, d_1, d_2).

Semnături digitale

Cu toate că părerile multora sunt contrare, *semnătura publică* (digitală) reprezintă cea mai importantă utilizare a sistemului *RSA*.

O *semnătură cu cheie publică* îi permite destinatarului să verifice autenticitatea unui mesaj. O astfel de semnătură trebuie să aibă următoarele proprietăți:

- semnătura nu trebuie să poată fi falsificată;
- semnătura nu trebuie să fie reutilizabilă; semnătura reprezintă o funcție a documentului și nu poate fi transferată la un alt document.

Sistemul *RSA* de *semnare digitală a mesajelor* are aceste proprietăți.

Algoritmul *RSA* de semnare digitală a mesajelor este aproape identic cu cel folosit pentru criptarea cu *cheie publică*, singura diferență fiind aceea că rolul cheilor se schimbă, adică perechea de numere *n* și *d* constituie *cheia privată*, iar perechea de numere *n* și *e* constituie *cheia publică*.

Fie *m* un mesaj care urmează să fie autentificat.

Dacă un utilizator **A** dorește să trimită mesajul *m* autentificat unui utilizator **B**, atunci **A** păstrează cheia *privată* (*n*, *d*) și trimite cheia publică (*n*, *e*) și mesajul autentificat *c*, unde:

$$c = \texttt{rest}([m^d/n])$$

Pentru a valida semnătura, utilizatorul **B** verifică dacă:

$$m = \texttt{rest}([c^e/n])$$

În continuare, mesajul original poate fi preluat doar dacă expeditorul deține *cheia privată*. În consecință, semnătura este autentică.

Claudiu Soroiu este redactor al GInfo. Poate fi contactat prin e-mail la adresa **csoroiu@yahoo.com**.