Criptografia cu cheie publică. Protocolul Diffie-Hellman

Se poate genera o cheie comună secretă?

Presupunem din nou că Alice şi Bob vor să schimbe mesaje printr-un canal nesigur (de exemplu o linie telefonică), care este ascultat de Oscar. De aceea, se hotărăsc să folosească un criptosistem simetric adică cheia de decriptare este aceeaşi cu cea de criptare sau se poate calcula foarte uşor din aceasta. Problema este cum trebuie să se decidă asupra cheii dacă mijlocul de comunicaţie este canalul nesigur ales? Ar putea să trimită şi cheia ca un mesaj criptat tot cu un cifru simetric, dar apare din nou aceeaşi problemă pentru noua cheie, adică şi aceasta ar trebui trimisă ca un mesaj şi aşa mai departe. Această situaţie paradoxală poartă numele de schimbul cheii secrete. Pentru a înţelege mai bine problema, am putea să ne imaginăm următorul scenariu. Să presupunem că Alice doreşte să trimită un mesaj lui Bob printr-un curier. Pentru a nu putea fi citit de către "curiosul" curier, ea pune plicul cu mesajul într-o cutie pe care o închide cu un lacăt şi o dă curierului care trebuie să o ducă lui Bob. Din păcate Bob nu are cheia lacătului (aceasta a rămas evident la Alice). Alice ar putea să pună cheia într-o altă cutie pe care o închide cu un alt lacăt şi aşa mai departe.

Se părea că criptografia se afla într-un impas counfruntându-se cu o problemă fără soluţie. Aceasta a fost însă rezolvată în 1976 de Whitfield Diffie şi Martin Hellman în articolul "New directions in criptography".

Pentru a vedea cum au gândit cei doi soluţia problemei să reluăm scenariul cu lacătele. Să presupunem că Alice poate ataşa cutiei două lacăte unul lângă altul. Dacă Alice se hotărăşte să trimită plicul folosind cutia, închide cutia cu un lacăt, păstrează cheia şi trimite cutia prin curier lui Bob. Acesta, ca mai sus nu o va putea deschide, dar la rândul lui, va mai adăuga încă un lacăt, păstrând cheia acestuia, după care trimite cutia înapoi lui Alice. Aceasta, când primeşte cutia, va scoate lacătul pus de ea, şi retrimite cutia lui Bob. Aceasta este închisă în acest moment doar cu lacătul lui Bob. Când Bob va primi cutia, o va putea deschide şi citi mesajul.

Ar putea fi oare aplicată această idee simplă pentru rezolvarea problemei? La prima vedere s-ar părea că da pentru că nu s-a schimbat vreo cheie. Alice ar putea cripta mesajul cu cheia ei şi-l trimite lui Bob. Acesta ar putea să continue să cripteze mesajul cu cheia lui şi-l retrimite lui Alice. Aceasta decriptează mesajul cu cheia ei şi-l va retrimite înapoi lui Bob care decriptează mesajul cu cheia lui.

În practică însă procedura nu funcţionează deoarece ordinea cifrărilor este importantă. Cifrurile cu care se lucra în acea perioada nu erau deloc triviale (precum cifrul Caesar la care procedura ar putea merge).

În realitate scenariul cu lacătele ar suna substanţial diferit. E ca şi cum Alice ar pune mesajul într-o cutie pe care o închide cu un lacăt, o trimite lui Bob, acesta pune la rândul său cutia într-o cutie mai mare pe care o închide cu lacătul său şi o trimite lui Alice. Dar acum

Alice nu mai are acces la cutia ei pentru a o deschide. Cu alte cuvinte, în general, criptările nu comută. Ne putem gândi că mai întâi ne punem ciorapii şi apoi pantofii, urmând ca apoi să ne scoatem mai întâi pantofii şi după aceea ciorapii. Deci ordinea normală ar fi: Alice criptează, Bob criptează, Bob decriptează, Alice decriptează, dar atunci ne-am întors de unde am plecat.

Deşi scenariile de mai sus nu ofereau un răspuns la problema schimbului de chei, au dat totuşi unele idei celor doi (de fapt trei pentru că li s-a alăturat şi Ralph Merkle) asupra unor posibile soluţii.

Să considerăm acum următorul scenariu (naiv), care ar putea să ne dea o schemă logică pentru generarea unei chei comune secrete, prin care practic nu este transferată în clar nici o cheie.

1. Alice şi Bob se gândesc fiecare separat la câte un număr pe care îl păstrează secret. De exemplu Alice alege $a = 7$ şi Bob alege $b = 39$. Cele două numere vor fi folosite pentru a construi o cheie comună secretă.
2. Ei cad de acord asupra unui al treilea număr nesecret, pe care evident îl ştie şi Oscar. De exemplu $c = 27$.
3. Alice calculează suma $a + c = 34$ şi trimite rezultatul lui Bob. Oscar ştie această sumă!
4. Bob calculează suma $b + c = 66$ şi trimite rezultatul lui Alice. Oscar ştie această sumă!
5. Bob adună la suma primită de la Alice numărul său secret şi obţine $a + c + b = 73$.
6. Alice face acelaşi lucru cu suma primită de la Bob şi obţine evident acelaşi număr $b+c+a = 73$.

Iată cum cei doi au reuşit să construiască o cheie "secretă" fără ca această valoare să fi apărut în corespondenţa lor în clar. Operaţiile în sine sunt evident ridicole pentru că Oscar poate afla imediat cheia comună ţinând cond de datele pe care le are. Acest lucru este posibil deoarece operaţia de adunare este "reversibilă", adică foarte uşor de făcut dar şi foarte uşor de inversat. Ideea însă, se pare că este bună. Trebuia să se găsească nişte operaţii matematice "într-o singură direcţie", adică uşor de făcut dar foarte greu de inversat. În final cei trei au luat o astfel de funcţie din aritmetica modulară.

Logaritmul discet

Fie p un număr prim (foarte mare) şi g o rădăcină primitivă modulo p, adică ordinul său, ca element al grupului multiplicativ $(\mathbb{Z}/p\mathbb{Z})^*$, este $p-1$ ($g^{p-1} \equiv 1 \pmod p$ şi nu există d, $1 \leq d < p-1$ pentru care $g^d \equiv 1 \pmod p$). Sau, altfel spus, g este rădăcină primitivă modulo p dacă g este un generator al grupului multiplicativ $(\mathbb{Z}/p\mathbb{Z})^*$, adică mulţimile $\{1, g, g^2, \ldots, g^{p-1}\}$ şi $\{1, 2, \ldots, p-1\}$ sunt egale.

Observaţia 1. *Există exact $\varphi(p-1) = \varphi(\varphi(p))$ rădăcini primitive modulo p.*

Observaţia 2. *Cum se găsesc rădăcinile primitive? Iată un exemplu simplu: Dacă numărul p satisface condiţia $p-1 = 2 \cdot q$, cu q număr prim, atunci $\varphi(2q) = q-1$ şi deci, aproape jumătate din elementele lui $(\mathbb{Z}/p\mathbb{Z})^*$ sunt rădăcini primitive modulo p. Astfel, dacă avem nevoie de o rădăcină primitivă, alegem la întâmplare un număr g, $1 \leq g \leq p-1$ (probabilitatea este $1/2$ ca*

să fie un generator) şi apoi verificăm dacă $g^2 \equiv 1 \pmod p$ *şi* $g^q \equiv 1 \pmod p$*. Dacă nici una din congruenţele precedente nu este satisfăcută, înseamnă că* g *este rădăcină primitivă modulo* p*. Am folosit faptul că într-un grup finit, ordinul unui element al său divide ordinul grupului (Teorema lui Lagrange).*

Exemplul 1. *În grupul multimplicativ* $(\mathbb{Z}/11\mathbb{Z})^*$ *există exact* $\varphi(10) = 4$ *rădăcini primitive. Acestea sunt* $\{2, 6, 7, 8\}$*. De exemplu* $g = 6$ *este o rădăcină primitivă modulo* 11 *deoarece* $6^2 \equiv 3 \not\equiv 1 \pmod{11}$ *şi* $6^5 \equiv 10 \not\equiv 1 \pmod{11}$*. Puterile sale sunt calculate în tabelul de mai jos:*

a	1	2	3	4	5	6	7	8	9	10
$6^a \pmod{11}$	6	3	7	9	10	5	8	4	2	1

Pentru celelalte elemente din $(\mathbb{Z}/11\mathbb{Z})^*$ *avem*

$$3^5 \equiv 4^5 \equiv 5^5 \equiv 9^5 \equiv 10^5 \equiv 1 \pmod{11},$$

Să vedem ce operaţie au propus Diffie şi Hellman pentru a-şi pune în practică protocolul.

Definiţia 1. *Fie* p *un număr prim şi* g *o rădăcină primitivă modulo* p*. Funcţia*

$$f : \{0, 1, 2, \ldots, p-2\} \to \{1, 2, \ldots, p-1\},$$

definită prin:

$$f(a) = g^a \pmod p,$$

se numeşte exponenţiala discretă (modulo p*).*

Deoarece g *este rădăcină primitivă, funcţia exponenţială este bijectivă. Inversa sa se numeşte logaritmul discret.*

Observaţia 3. *Logaritmul dscret are multe proprietăţi asemănătoare cu logaritmul clasic din analiză, însă există o diferenţă fundamentală: valorile logaritmului discret sunt împrăştiate total neregulat spre deosebire de logaritmul din analiza reală. De aceea exponenţiala discretă este un exemplu de funcţie "într-o direcţie" (one way function), adică valorile ei sunt foarte uşor de calculat (în timp polinomial dacă folosim exponenţierea rapidă), dar foarte greu de inversat.*

Definiţia 2. *Fie* g *o rădăcină primitivă modulo* p *şi fie* h *un element nenul din* $\mathbb{Z}/p\mathbb{Z}$*. Problema găsirii exponentului* x*,* $0 \leq x \leq p-2$*, astfel încât* $g^x \equiv h \pmod p$ *se numeşte Problema Logaritmului Discret (PLD). Uneori* x *se notează* $d\log_g h$ *sau simplu* $\log_g h$*.*

Rezolvarea PLD este condiderată a fi o problemă foarte dificilă.
Există mai multe metode de a calcula logaritmul discret

1. Folosind "forţa brută", adică prin calcularea tuturor puterilor g^a. Consumul este $\mathcal{O}(p)$ paşi, unde fiecare pas constă în multiplicarea cu g.

2. Algoritmul lui Shanks calculează logarirmul discret în $\mathcal{O}(\sqrt{p}\log p)$ paşi, adică mult mai rapid decat forţa brută, dar în continuare este tot exponeţial. în plus apar şi probleme de stocare.

3. Algoritmul Polard-ρ o face în $\mathcal{O}(\sqrt{p})$ paşi şi sunt rezolvate şi problemele de stocare.

4. Algoritmul PohligHellman rezolvă PLD în cel mia ră caz în $\mathcal{O}(\sqrt{p})$ paşi. Dacă numărul $p-1$ este neted (adică factorii primi p_i din descompunerea sa $p-1=\prod_{i=1}^{k} p_i^{e_i}$ sunt mici) atunci complexitatea algoritmului este $\mathcal{O}(\sum_i e_i(\log p+\sqrt{p_i}))$.

5. Algoritm de calcul al indexului este subexponenţial.

Observaţia 4. *Logaritmul discret poate fi definit în orice grup ciclic având un generator g.*

Dacă vom considera problema logaritmului discret în grupul aditiv $\mathbb{Z}/p\mathbb{Z}$, o soluţie se va găsi rezolvând congruenţa $x \cdot g \pmod p$. Acest lucru se poate face în $\mathcal{O}(\log p)$ paşi, adică în timp liniar şi deci acest grup nu este un candidat foarte bun pentru o funcţie într-o direcţie. Remarcăm astfel, că dificultatea rezolvării problemei logaritmului discret depinde foarte mult de grupul cu care lucrăm. Astfel, dacă se lucrează în grupul $(\mathbb{Z}/p\mathbb{Z})^*$ cel mai bun algoritm lucrează în timp subexponenţial. Se poate însă considera problema în nişte grupuri mai "exotice" (grupul curbelor eliptice) în care rezolvarea PLD se face doar în timp exponenţial.

Protocolul Diffie-Hellman

Alice şi Bob vor să genereze o cheie comună secretă care va fi folosită pentru criptarea şi decriptarea într-un criptosistem simetric. Ei pot comunica doar printr-un canal nesigur. Orice informaţie care va fi circula între ei prin acest canal va putea fi intreceptată de Oscar. Cum va putea fi generată cheia fără ca aceasta să fie "văzută" de Oscar?

Whitfield Diffie şi Martin E. Hellman au rezolvat această problemă în 1976 îarticolul "New Directions in Cryptography" IEEE Transactions on Information Theory. 22 (6): 644?654. Descriem acum protocolul:

1. Alice şi Bob se înţeleg asupra unui număr prim p foarte mare (p are aproximativ 1000 biţi, adică $p \equiv 2^{1000}$) şi asupra unei rădăcini primitive modulo p (g este de ordinul lui $p/2$). p şi g sunt publice, adică le ştie şi Oscar. De exemplu $p=11$ şi $g=6$.

2. - Alice alege un întreg foarte mare a, $0 \leq a \leq p-2$ (valoarea lui a este secretă şi va fi ştiută doa de Alice) şi calculează $u \equiv g^a \pmod p$. De exemplu alege $a=3$ şi calculează $u \equiv 6^3 \equiv 7 \pmod{11}$.
   - Bob alege un întreg foarte mare b, $0 \leq b \leq p-2$ (valoarea lui b este secretă şi va fi ştiută doa de Bob) şi calculează $v \equiv g^b \pmod p$.De exemplu alege $b=7$ şi calculează $v \equiv 6^7 \equiv 8 \pmod{11}$.

3. - Alice trimite valoarea u lui Bob (Oscar ştie această valoare)
   - Bob trimite valoarea v lui Alice (Oscar ştie această valoare)

4. - Alice calculează $k \equiv v^a (\equiv g^{ab}) \pmod p$, adică cheia comună. În cazul exemplului nostru avem $k \equiv 8^3 \equiv 6$.
   - Bob calculează $k \equiv u^b (\equiv g^{ab}) \pmod p$, adică cheia comună. În cazul exemplului nostru avem $k \equiv 7^7 \equiv 6$.

În general, dilema lui Oscar este următoarea. El ştie valorile lui u şi v (adică ştie valorile g^a şi g^b). El ştie şi valorile lui g şi p şi deci, dacă ar putea rezolva problema logaritmului discret, atunci ar putea găsi valorile a şi b, după care ar fi foarte uşor să calculeze cheia secretă. Se pare

că Alice şi Bob pot sta liniştiţi atâta timp cât Oscar nu poate rezolva problema logaritmului discret, dar acest lucru nu este chiar corect. Este clar că o metodă de a afla cheia secretă este rezolvarea problemei logaritmului discret, dar de fapt nu aceasta este problema pe care Oscar trebuie să o rezolve. Securitatea cheii secrete obţinute de Alice şi Bob stă de fapt în dificultatea următoarei probleme (potenţial mai uşoară):

Definiţia 3. *Fie p un număr prim şi g o rădăcină primitivă modulo p. Problema Diffie-Hellman (PDH) este problema calculării valorii g^{ab} (mod p), folosind valorile cunoscute g^a (mod p) şi g^b (mod p).*

Este clar că PDH nu este mai grea decât PLD. Dacă Oscar poate rezolva PLD, atunci poate calcula exponenţii secreţi a şi b din valorile interceptate $u = g^a \pmod p$ şi $v = g^b \pmod p$, după care calculează uşor g^{ab}. De fapt Oscar are nevoie doar de a sau b. Reciproca însă nu este deloc clară. Presupunem că Oscar deţine un algoritm care poate rezolva PDH. Poate el folosi în mod eficient algoritmul pentru a rezolva PLD? Răspunsul nu este ştiut.

Algoritmul lui Shanks

Propoziţia 1. *(Algoritmul lui Shanks: Paşi de Pitic-Paşi de Uriaş) Fie p un număr prim şi g o rădăcină primitivă mod p. Fie $h \in \mathbb{Z}_p$. Următorul algoritm rezolvă problema logaritmului discret $g^x = h$ în $\mathcal{O}(\sqrt{p}\log p)$ paşi. Notăm cu $N = p - 1$ ordinul lui g.*

1. *Fie $n = 1 + \lfloor\sqrt{N}\rfloor$, şi deci, în particular $n > \sqrt{N}$.*
2. *Se generează două liste*
   (a) *$g^0 = 1, g^1, g^2, \ldots, g^n$, (înmulţirea cu g este un pas de copil) şi*
   (b) *$h \cdot g^0 = h, h \cdot g^{-n}, h \cdot g^{-2n}, h \cdot g^{-3n}, \ldots, h \cdot g^{-n^2}$ (înmulţirea cu g^{-n} este un pas de uriaş).*
3. *Căutăm potriviri (coliziuni) între elementele celor două liste, să zicem $g^i = hg^{-jn}$.*
4. *În final $x = i + jn$ este soluţie a ecuaţiei $g^x = h$.*

Demonstraţie: Începem cu câteva observaţii:

• când se creează lista (b), se începe cu calculul valorii $u = g^{-n}$ şi apoi se completează lista prin calculul valorilor $h, h \cdot u, h \cdot u^2, \ldots, h \cdot u^n$. Astfel, crearea celor două liste consumă aproximativ $2n$ multiplicări;

• dacă presupunem că există potriviri între cele două liste, putem găsi o astfel de coliziune într-un multiplu mic al lui $\log n$ de paşi folosind algoritmi standard de sortare şi căutare, aşa încât pasul (3) se face în $\mathcal{O}(\log n)$ paşi;

• timpul total de rulare al algoritmului este $\mathcal{O}(n \log n) = \mathcal{O}(\sqrt{p}\log p)$. Pentru acest ultim pas am folosit faptul că $n \approx \sqrt{p}$ şi deci $n \log n \approx \sqrt{p}\log\sqrt{p} = \frac{1}{2}\sqrt{p}\log p$.

Pentru a arăta că algoritmul funcţionează, trebuie demonstrat că există mereu potriviri între anumite elemente din cele două liste. Pentru aceasta, fie x soluţia ecuaţiei $g^x = h$, şi scriem pe x ca $x = nq + r$ cu $0 \le r < n$. Pe de altă parte ştim că $1 \le x < N$ $(= p - 1)$ şi deci

$$q = \tfrac{x-r}{n} < \tfrac{N}{n} < n$$

pentru că $n > \sqrt{N}$.

Putem acum rescrie ecuaţia $g^x = h$ ca $g^r = h \cdot g^{-qn}$ cu $0 \leq r < n$ şi $0 \leq q < n$. Se vede că g^r se află în lista (a) iar $h \cdot g^{-qn}$ se află în lista (b), ceea ce arată că cele două liste au un element comun.■

Exemplul 2. *Să presupunem că vrem să rezolvăm ecuaţia* $6^x \equiv 8 \pmod{11}$*. Aplicăm algoritmul de mai sus. Avem* $p = 11$ *deci* $N = 10$*, rădăcina primitivă este* $g = 6$ *(ordinul lui 6 este 10 în grupul* $(\mathbb{Z}^*, 11.\cdot)$*) şi* $h = 8$*.*

1. $n = 1 + \lfloor \sqrt{10} \rfloor = 1 + 3 = 4$.
2. $u = g^{-n} = 6^{-4} = 6^6 = 5$ *şi cele două liste sunt:*
   (a) $6^0 = 1, 6^1 = 6, 6^2 = 3, 6^3 = 7.6^4 = 9$ *şi*
   (b) $8 \cdot g^0 = 8, h \cdot 5 = 7, 8 \cdot 3 = 3, 6 \cdot 4 = 10, 8 \cdot 9 = 6$.
3. *Se observă că al patrulea termen din prima listă conincide cu al doilea din a doua listă* $6^3 = 8 \cdot 6^{-4}$*. Deci* $i = 3$ *şi* $j = 1$*.*
4. *Obţinem în final* $x = 2 + 1 \cdot 4 = 7$*.*

Observaţia 5. *Se observă că avem şi* $i = 1$ *şi* $j = 4$*, adică* $x = 1 + 4 \cdot 4 = 17$*, dar trebuie să lucrăm modulo* $10\ (= p - 1)$ *şi deci soluţia este tot* $x = 7$*.*