CriptoTema5.pdf

onafolch

Criptografia i Seguretat

3º Grado en Ingeniería de Datos

Escuela de Ingeniería
Universidad Autónoma de Barcelona

CRIPTOGRAFIA

SISTEMES DE CLAU PÚBLICA

1. PROBLEMA AMB LA CLAU SIMÈTRICA

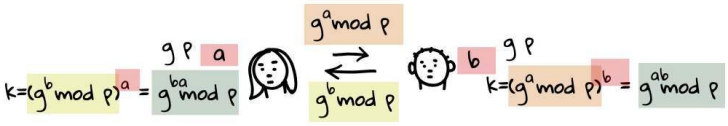
La criptografia de clau simètrica fa servir una mateixa clau tant per xifrar com per a desxifrar. Tant l'emissor d'un missatge (que el xifrarà abans d'enviar-lo), com el receptor (que l'haurà de desxifrar), comparteixen una única clau. Però presenta algunes limitacions:

1. **Canal segur**: La distribució de claus s'ha de realitzar sobre un canal segur, ja que sinó un atacant que estigués escoltant el canal podria capturar-la i utilitzar-la.
2. **Escalabilitat**: La gestió de claus es complica quan el número d'usuaris creix. Si hi ha n usuaris i cada parell d'usuaris necessita compartir una clau, caldrà gestionar n(n − 1)/2 claus.
3. **Sense propietat de no-repudi**: Com que més d'un usuari comparteixen una mateixa clau, no es pot garantir que un usuari en concret ha realitzat una acció donada, ja que sempre hi haurà algun altre usuari que podria haver-la realitzat.

La criptografia de clau pública permet superar aquestes limitacions. Proporciona mètodes per aconseguir que dos usuaris que es comuniquen per un canal insegur puguin crear claus que els permetin comunicar-se de manera segura. Permet que un conjunt d'usuaris es comuniquin dos a dos de manera segura fent servir únicament un parell de claus per cada usuari, i ens ofereix la propietat de no-repudi.

2. DIFFIE-HELLMAN

És un algorisme d'intercanvi de claus i permet que dos usuaris que es comuniquen per un canal insegur tinguin una clau compartida de manera segura. Encara que un atacant estigui escoltant el canal, aquest no pot aconseguir la clau. Tot i així, l'esquema no és segur davant d'atacants que puguin modificar la informació que viatja pel canal.



1. Es tria un nombre primer **p** i un enter **g** i es fan públics.
2. Alice escull un valor aleatori ***a*** i calcula $\boldsymbol{A} = g^a \bmod p$
3. Bob escull un valor aleatori ***b*** i calcula $\boldsymbol{B} = g^b \bmod p$
4. Alice envia A a Bob. Bob envia B a Alice
5. Alice calcula $\boldsymbol{k} = B^a \bmod p = (g^b)^a \bmod p = g^{b*a} \bmod p$
6. Bob calcula $\boldsymbol{k} = A^b \bmod p = (g^a)^b \bmod p = g^{a*b} \bmod p$

3. ESQUEMA CLAU ASIMÈTRICA

La encriptació asimètrica es basa en que tenim dues claus per cada usuari, una pública i una privada. Aquestes claus tenen la propietat que el que s'encripta amb una, es desencripta amb l'altra, ja que són complementaries.

Satisfà la propietat de **confidencialitat** si Alice envia un missatge a Bob encriptat amb la pública de Bob, ja que només ell el podrà desencriptar amb la seva clau privada.

$$Alice\colon [E_{k\,pubB}(m)\,]$$

Ona Folch

També satisfà la propietat de **autenticació i integritat** si firmo el missatge. Alice envia el missatge i el hash del missatge, encriptat amb la seva privada. Quan Bob ho rebi, desencriptarà el hash del missatge amb la pública de Alice, i si el hash del missatge que ha rebut dona el mateix que això, significarà que ha sigut Alice qui ho ha enviat.

$$Alice: [m, E_{k\ privA}(h(m))\ ]$$

Si vull que satisfaci tant la propietat de confidencialitat com de autenticació combinaré les dues. Alice enviarà el missatge encriptat amb la pública del Bob i el hash del missatge encriptat amb la seva privada

$$Alice: [\ E_{k\ pubB}(m), E_{k\ privA}(h(m))\ ]$$

4. RSA (Riverst-Shamir-Adleman)

És un criptosistema de clau pública que ens permet generar parells de claus públiques i privades.

Passos a seguir per tal de generar un parell de claus RSA:

1. Es trien dos nombres primers aleatoris **p** i **q**. (p = 5 i q = 11)
2. Es calcula $\boldsymbol{n} = p * q = 5 * 11 = 55$
3. Es calcula $\boldsymbol{\phi(n)} = (p-1)(q-1) = 4 * 10 = 40$.
4. Se selecciona un exponent públic **e** tal que $1 < e < \phi$ i $mcd(e, \phi) = 1$, per exemple **e = 7**.
5. Es calcula l'exponent privat **d** tal que $1 < d < \phi$ i $ed \equiv 1\ mod(\phi)$. Per tant **d = 23**.
6. La clau pública és **(n,e)**, per tant (55,7). La clau privada és **d = 23**. Els valors p,q i ϕ són valors secrets.

La seguretat es basa en la dificultat de factorització.

Encriptació i desencriptació: B encripta el missatge m per a A, el qual A desencripta.

1. Encriptació: B obté la clau pública de A (n,e), i representa el missatge m en un interval de [0,n-1]. Calcula $c = m^e\ mod\ n$ i envia c a A.
2. Desencriptació: A rep el missatge i utilitza la seva clau privada per calcular $m = c^d\ mod\ n$

Funció d'Euler

La funció d'Euler $\phi(n)$ ens dirà el nombre d'elements del CRR (conjunt reduït de restos), que són els restos primers amb el cos n.

- **Si n és un nombre primer** $\boxed{\phi(n) = n - 1}$ **Exemple:** CRR(7) = $\phi(7) = 6$
- **Si n es $\boldsymbol{n = p^k}$, on p és primer i k un enter** $\boxed{\phi(n) = p^{k-1}(p-1)}$ **Exemple:** CRR(16) = $\phi(16) = 8$
- **Si n és $\boldsymbol{n = p * q}$, on p i q són primer** $\boxed{\phi(n) = (p-1)(q-1)}$ **Exemple:** CRR(15) = $\phi(15) = 8$
- **Si n és un nombre qualsevol (forma genèrica)** $\boxed{\phi(n) = \prod_{i=1}^{t} p_i^{ei-1}(p_i - 1)}$

  Exemple: CRR(20) = $\phi(20) = \phi(2^2 * 5) = 2^{2-1}(2-1) * 5^{1-1}(5-1) = 2 * 1 * 1 * 4 = 8$

El teorema d'Euler ens permet calcular l'invers d'un nombre. Si mcd(a,n) = 1, sabem que $a^{\phi(n)} mod\ n = 1$. Si igualem les dues funcions, tenim que $x = a^{\phi(n)-1} mod\ n$, on x és l'invers d'a en el cos n.

Exemple: Invers de 4 en el cos 9 (inv(4,9)). Com que mcd(4,9) = 1, sabem que té invers. Calculem $\phi(9) = 6$. Utilitzant la formula del teorema d'Euler, $x = 4^{6-1} mod\ 9 = 7$. L'invers de 4 en el cos 9 és 7. inv(4,9) = 7 i inv(7,9) = 4.

Ona Folch

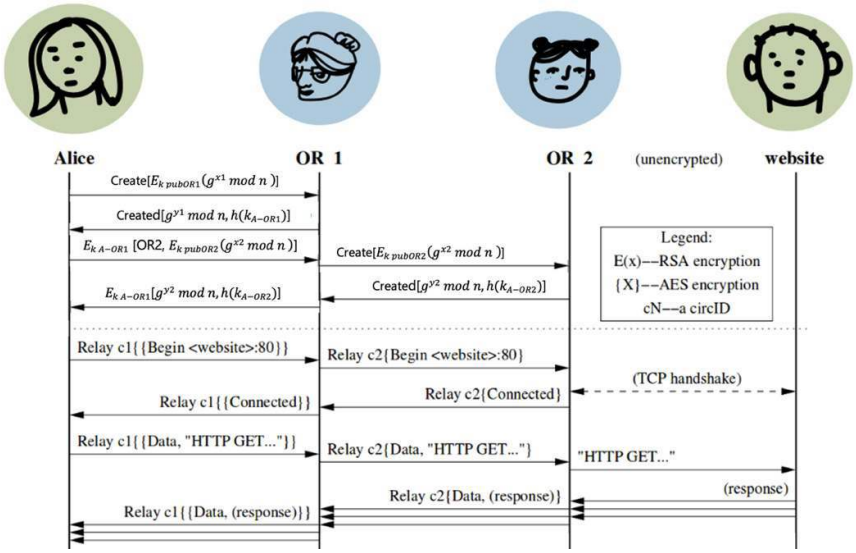
5. ANONYMITY (TOR circuit)

A vol enviar un missatge a B, sense que B sàpiga que qui li està enviant és A. A només parla amb OR1, i OR1 només parla amb OR2, el qual no sap res de A. Tots els OR tindran clau privada i clau pública.

A té una clau simètrica amb OR1 i OR2. Es fa Diffie-Hellman entre A i OR1; A li envia a OR1 $g^{x1} \ mod \ n$ encriptat amb la pública de OR1, de tal manera que només OR1 ho podrà desxifrar (A li envia $[E_{k\ pubOR1}(g^{x1} \ mod \ n\ )]$ a OR1). OR1 li contesta a A amb $g^{y1} \ mod \ n$ i el hash de la clau creada, per tal de verificar que és OR1 (OR1 li envia $[g^{y1} \ mod \ n, h(k_{A-OR1})]$ a A).

Ara A vol crear una clau simètrica amb OR2. Li envia a OR1 $g^{x1} \ mod \ n$ encriptat amb la pública de OR2, i indicant-li que vol que ho envii a OR2, i tot això encriptat amb la clau simètrica que han creat anteriorment A i OR1 (A envia $E_{k\ A-OR1}$ [OR2, $E_{k\ pubOR2}(g^{x2} \ mod \ n\ )$] a OR1). Ara OR1 desxifrarà el missatge amb la clau simètrica i veurà que ha d'enviar a OR2 el missatge encriptat (OR1 envia $[E_{k\ pubOR2}(g^{x2} \ mod \ n\ )]$ a OR2). OR2 respon a OR1 amb el hash de la clau creada i $g^{y2} \ mod \ n$ (OR2 li envia $[g^{y2} \ mod \ n, h(k_{A-OR2})]$ a OR1). I ara OR1 li envia aquest missatge a A encriptat amb la clau simètrica que té amb A(OR1 envia $E_{k\ A-OR1}$ $[g^{y2} \ mod \ n, h(k_{A-OR2})]$.

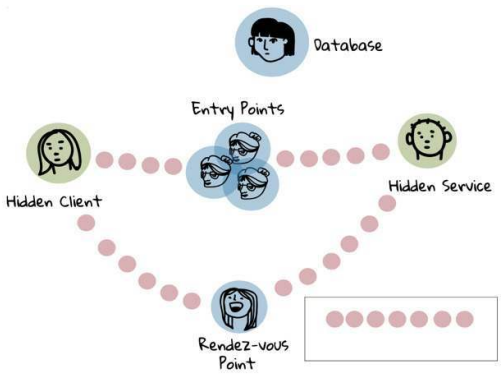
A partir d'aquí, A pot enviar missatges a B sense que B sàpiga qui és A. Si A vol enviar un missatge a B, encriptarà el missatge amb la clau simètrica amb OR2, i aquest missatge encriptat l'encriptarà amb la clau simètrica que té amb OR1.



Però A sap qui és B. Si ara volem que A no sàpiga qui és B, i que B no sàpiga qui és A, tindrem diversos canals amb el circuit de TOR.

B (Hidden Service) escull 3 nodes aleatòriament, i a la base de dades indica que si algú li vol enviar alguna cosa que ho fagin a través d'aquests 3 nodes. Si A (Hidden Client) vol contactar amb B, la primera cosa que farà serà contactar amb la base de dades de manera anònima, i la base de dades li retornarà els nodes amb qui ha de contactar.

A escull un punt de trobada (RV) i estableix un canal segur i anònim. A partir d'aquí, A li diu a B a través d'un dels 3 nodes que es veuran en el punt de trobada RV, i finalment, B crea un canal segur i anònim de B a RV.



Ona Folch