dSik - Confidentiality

Lukas Peter Jørgensen, 201206057, DA4

16. juni 2014

1 Sikkerhedsmål

CAA

CAA står for:

Confidentiality: Information skal holdes hemmelig for uvedkommende, gælder både under forsendelse, opbevaring og behandling af data

Authenticity: Informationen er autentisk, den er ikke blevet manipuleret af en uautoriseret person.

Availability: Systemer skal være tilgængelig når de skal bruges

Definition af et sikkert system

Det er typisk svært eller umuligt at bevise et system er sikkert. Tit bliver man nødt til at lave mange antagelser om angriberens muligheder for at bevise at et system er sikkert og disse antagelser er typisk forkerte.

Vi bruger udtrykket "sikret system"istedet for "sikkert system"da det blot indikerer at vi har sikret systemet mod nogle bestemte tilfælde. Disse tilfælde defineres ved en *Sikkerhedspolitik* på baggrund af en *Trusselmodel*, og implementerer herefter denne sikkerhedspolitik vha. nogle *Sikkerhedsmekanismer*. På den måde får vi, at et sikret system kan beskrives som:

$$Sikretsystem = Sikkerhedspolitik + Trusselsmodel + Sikkerhedsmekanismer$$

Kryptologi

Kryptografi: Videnskaben om at lave koder og ciffere.

Kryptoanalyse: Videnskaben om at bryde koder og ciffere.

2 Secret-key/Public-key Confidentiality

Vi må antage at vore fjender altid kender til vore krypteringsalgoritmer, så et cryptosystem må ikke afhænge af at holde algoritmen hemmelig.

Istedet skal man bruge en nøgle, som skal distribueres mellem parterne, og bruges i kombination med krypteringsalgoritmen. Der er overordnet set to måder at bruge nøgler, Secret-key og public-key systemer.

Secret-key

Man krypterer en besked m med nøglen k således:

$$c = E_k(m)$$

Dekrypterer således:

$$m = D_k(c) = D_k(E_k(m))$$

For at undgå at c er den samme for krypteringer af m tilføjer man tit et nonce.

Public-key

Man krypterer en besked under en public-key, og dekrypterer den så under en private key

$$c = E_{pk}(m)$$

$$m = D_{sk} = D_{sk}(E_{pk}(m))$$

Begge systemer er kan brydes ved *exhaustivekeysearch*, dette sørger man dog for er usandsynligt i praksis. Der findes algoritmer til at finde sk ud fra pk derfor er nøglestørrelserne for public-key kryptering langt større end for secret-key. (Typisk minimum 1024 bit. GPG anbefaler 2048 bit).

Hvorfor bruger man ikke altid public-key? Det er for langsomt.

Stream ciphers

Man udvider nøglen til en meget lang tilfældigt udseende streng (eks. bruge nøglen som seed til randomgenerator). Man bruger så denne streng som en one-time pad, således kan man kryptere en bit af gangen.

Block ciphers

Opererer i blokke af data, består typisk af 3 ting:

1. En krypteringsalgoritme (e.g. DES, 3DES eller AES)
2. Et "Mode of operation"
3. En $InitializationVector$ (IV) Note: ikke all modes of operation bruger en IV, men det gør de typisk.

Modes of operation

ECB

Electronic codebook, del koden op i blokke af n bit hvor n er blockstørrelserne for den pågældende algoritme (e.g. 64bit for DES) så man får beskeder $M_0 ... M_{t-1}$.

$$C_i = E_k(M_i)$$

Problem: Manglende randomness, replay attacks.

CBC

Igen har vi blokkene $M_0...M_t$, men denne gang er der $t+1$ bloks, den ekstra blok er IV'en som ligger ved M_0, og er blot en tilfældigt valgt blok.

$$C_i = E_k(M_i \bigoplus C_{i-1})$$

$$M_i = D_k(C_i) \bigoplus C_{i-1}$$

Problemer: Den kan ikke paralleliseres, den er sårbar overfor korruption af bits.

3 RSA

1. Vælg to store primtal p og q og sæt $n = pq$, således n bliver produktet af 2 primtal.
2. Udregn nu $t = \phi(n) = (p-1)(q-1)$, som kaldes Eulers totient af n.
3. Vælg en positiv integer e some er større end 1 og mindre end t, og som er indbyrdes primisk med t. Dette kan også skrives som at $e \in \mathbb{Z}, 1 < e < t, gcd(e,t) = 1$. En måde at gøre dette er at vælge e til at være et primtal. man vil typisk gerne have e til at være lille da det gør krypteringen nemmere og ingen sikkerhedsmæssig betydning har.
4. Udregn d, således at den opfylder kongruens relationen $ed \equiv 1(\mod t)$. Hvilket vil sige at $\mod t$ for $ed-1$ skal give 0.

Fordi e kan vælges til at være lille, kan man sørge for at krypteringen går hurtigt, dette kan man ikke gøre for d, derfor vil kryptering typisk gå meget hurtigere end dekryptering.

Nu har vi så 3 tal: n, d og e som giver nøglerne $pk = (n,e)$ og $sk = (n,d)$. Så krypterer man ved:

$$c \equiv m^e(\mod n)$$

$$m \equiv c^d \equiv m^{ed}(\mod n)$$

Euler's totient teorem siger os:

$$m^{ed} \equiv m^{de} \equiv m^{t+1} \equiv m^1 \equiv M(\mod n)$$

Angreb på RSA

Exhaustive key search

Prime factorization

Man kan faktorisere n.

Discrete Logarithms

Side-channel attack

Regnetiden for 0 er mindre den for 1, så man kan aflæse tiden eller strømforbruget for at aflæse resultatet.