1 Public key encryption

Umožnilo revoluci v kryptografii tím, že odstranilo problém private key encryption – potřebu domluvit se na klíči předem. To vycházelo z historického náhledu na šifrování, kdy schopnost šifrovat $\Leftrightarrow$ schopnost dešifrovat $\Leftrightarrow$ znalost klíče

1.1 Diffie + Hellman (1976)

- můžou být dva klíče
- 1. popis kryptografie s veřejným klíčem (před tím vynalezeno britskou tajnou službou)
- public key slouží pro šifrování
- kdokoliv může šifrovat
- veřejná databáze veřejných klíčů
- soukromý klíč umožňuje dešifrování
- soukromý klíč nesmí být možné odvodit od veřejného klíče

Předpoklad existence databáze veřejných klíčů je netriviální – dosud neexistuje důvěryhodné řešení $\rightarrow$ samostatná přednáška o spravování (někde v přednáškách o aplikované kryptografii)

1.2 Definice Public key encryption scheme

- trojice algoritmů (G, E, D):
- G - generuje dvojice klíčů $(pK, sK) \leftarrow G(^)$
- E - pravděpodobnostní algoritmus $c \leftarrow E_{pK}(m)$
- D - deterministický algoritmus $m = D_{sK}(c)$

Korektnost: $\forall(m, (pK, sK) \leftarrow G(1^n))D_{sK}(E_{pK}(m)) = m$

1.3 Definice Indistinguishability ciphertextů

Nechť (G, E, D) je PKE na prostoru všech správ $\mathcal{M} = \bigcup_{i \in \mathbb{N}} \mathcal{M}_i$, $\forall PPTA$ existuje negligible $\epsilon()$,

$$\forall m_0, m_1 \in \mathcal{M}_n | \Pr[A(1^n, pK, E_{pK}(m_0)) = 1] - Pr[A(1^n, pK, E_{pK}(m_1)) = 1]| \leq \epsilon(n) \forall n$$

kde pravděpodobnost je přes $(pK, sK) \leftarrow G(1^n)$ a náhodné mince A a E.

$\Rightarrow$ pro PKE máme ekvivalenci ind. ciphertextů a semantic security

Pro PKE potřebujeme kvalitativně silnější předpoklady nestačí jen existence jednosměrných funkcí potřebujeme jednosměrné permutace se zadními vrátky (trapdoor permutace) one way permutation + trapdoor

1.4 Definice Kolekce trapdoor permutací

Kolekce funkcí $F = f_k \times D_k \to D_k$ je kolekcí trapdoor permutací $\iff$

- pro všechny k je f_k permutací
- existuje generátor (k, t) kde t je trapdoor
- znalost k umožňuje efektivně vybrat efektivně z rovnoměrného rozdělení na D_k
- znalost k umožňuje efektivně vyhodnotit $f_k(x)$ pro všechny $x \in D_k$
- $\forall PPTA \exists negl.\epsilon() \Pr[A(1^n, K, f_K(X)) \in f_K^{-1}(f_K(n))] \leq \epsilon(n) \forall n$ pro $(K, T) \leftarrow G(1^n), X \leftarrow D_k$ a pro náhodné mince A
- znalost T umožňuje efektivně nalézt $f_k^{-1}(y)$ pro všechna $(k, t) \leftarrow G(1^n)$

Pro PKE potřebujeme schopnost vygenerovat "těžký problém a jeho řešení" (nemůžeme si vybrat ten problém) Pro trapdoor funkce máme výrazně méně kandidátů než pro jednosměrné funkce

1.5 Příklad: RSA

- kolekce $f_{N,e} : Z_N* \to Z_N*$
- $N = pq$
- $e \leftarrow Z * \phi(N)$
- $f_{N,e}(x) = x^e \mod N$
- $t = d; ed = 1 \mod \phi(N)$
- inverzní zobrazení: $f_{N,e}^{-1}(y, d) = y^d \mod N$
- $y = x^e \mod N => f(y, d)^{-1} = y^d \mod N = x^e d \mod N = x \mod N$
- $pK = (e, N)$
- $sK = (d)$

Jak generátor spočítá k, t?

- $k = e, N$
- $d = e^{-1}$ v $Z_{\phi N}*$ (pomocí rozšířeného Eukleidova algoritmu)

1.6 Rabinova kolekce TDPs

- parametry $p \equiv q \equiv 3(\mod 4)$, $N = pq$
- trapdoor (p, q)
- $fN : QR_N \rightarrow QR_N fN(x) = x^2 modN$

inverze: pro a z QR hledáme z takové, aby $a \equiv z^2 modN$
$Z_{pq}* \approx Z_p * xZ_q*$
pro prvočísla se druhá odmocnina hledá velmi snadno
$z_p \equiv a^{p+1/4} modp$ $z_q \equiv a^{q+1/4} modp$
stačí se podívat na $\pm\alpha z_p \pm \beta z_q$ $\alpha \equiv 1 \mod p$ $\alpha \equiv 0 \mod q$ $\beta \equiv 0 \mod p$ $\beta \equiv 1 \mod q$
dostáváme 4 druhé odmocniny a v Z_N* právě jedna bude v QR_N
Konstrukce PKE z TDPs 1) vyber $(k, t) \leftarrow G(1^n)$ 2) $pK = k$ $sK = t$ $E_{pK}(x) = f_k(x)$ $D_{sK}(y) = f_k^{-1}(y)$
INSECURE!!! - Pro RSA nefunguje například pro malé e
těch problémů je tam víc všechny RSA standardy mluví o paddingu je potřeba ty zprávy nějak náhodně rozdistribuovat
$OWF \not\rightarrow OWP$
$OWF \not\rightarrow PKE$
Otevřený problém: konstrukce PKE z OWF (když ale nebudu OWF používat jako blackbox)?
Tím, že máme málo kandidátů, hrozí jejich vyřešení (mimo jiné kvantovým počítačem)
konstrukce z hardcore bitů
$M = 0, 1$
$(k, t) \leftarrow G(1^n)$
$pK = k$
$sK = t$
$E_{pK}(m) : m \leftarrow Dk$ vrať $fk(x), bk(x) \oplus m$
$D_{sK}(c) : c = c_1, c_2$ $f_k^{-1}(c_1) = \tilde{x}$ vrať $b_k(X) \oplus c_2$
Tvrzení: pokud $\{f_k\}$ je kolekce TDPs s hardcore bity $\{b_k\}$ pak předešlé schéma splňuje ind. ciphertextů
adversary nemůže rozpoznat b_k od náhodných
Myšlenka důkazu:

$$(pK, E_{pK}(0)) \equiv (k, (f_k(x), b_k(x))) \equiv_c (k, (f_k(x), R)) \equiv (k, (f_k(x), !R)) \equiv_c (k, (f_k(x), !b_k(x))) \equiv (pK, E_{pK}(1))$$

Tohle způsobuje dvojnásobnou délku ciphertextu Efektivnější varianta: $M = 0, 1^l$ $E_{pK}(m)$: $x \leftarrow Dk$ $r = G_l(x) = b_k(x)||b_k(f_k(x))||b_k(f_k^2(x))||...||b_k(f_k^{l-1}(x))$ vrať $(f_k^l(x), r \oplus m)$ Důkaz jako pro $PRG1 \rightarrow PRGl$
-¿ Blum + Goldwasser staví na Rabinově kolekci TDPs $p \equiv q \equiv 3 \mod 4$ $N = pq$ $pK = N$ $sK = (p, q)$
$E_{pK}(m)$:

1. $x \leftarrow QR_N$

2. $r = lsb(x)||lsb(x^2 \mod N)||lsb(x^4 \mod N)||...||lsb(x^{2^(l-1)} \mod N)$ – Blum Blum Shub generátor

3. vrať $c = (x^{2^l} \mod N, r \oplus m)$

$$x_p \equiv y^{((p+1)/4)^l} \mod p$$

$$x_q \equiv y^{((q+1)/4)^l} \mod q$$

$$X \equiv q(q^{-1} \mod p)x_p + p(p^{-1} \mod q)x_q(\mod N)$$

1.7 Diffie-Hellman key exchange

mějme cyklickou grupu G řádu q a generátor g

Alice:

- zvolí náhodně G, q, g
- vybere $x \leftarrow Z_q$
- $h_A = g^x$
- pošle (G, q, g, h_A) Bobovi

Bob:

- vybere $y \leftarrow Z_q$
- $h_B = g^y$
- pošle h_B Alici

Oba:

$$k_A = h_B^x = g^{xy} = h_A^y = k_B$$

vyžaduje to aby diskrétní logaritmus byl těžký
Computational Diffie-Hellman assumption:

$$G, q, q, g^x, g^y \not\to g^x y$$

Decision Diffie-Hellman assumption: (g^z = náhodný prvek G)

$$G, q, g, g^x, g^y, g^x y \equiv_c G, q, g, g^x, g^y, g^z$$

$\to$ El Gamal (příště (za dva týdny))