Lecture 23 – Cryptography

[[Hidden writing]]

[[Gonna do a no-#-theory version]]

[[Scribe – Pass-Shelat]]

Alice —Eve [[is dropped]]→ Bob

m. c := Enc(m) Dec(m) = c

msg

[[A wants to send secret to Bob, but ∄ Eve]]

[[Modeling: parties are algs.]]

[[message, say bitstring "plaintext"]] [["ciphertext"]]

"Security": "Eve should not get any info abt. m from c."

[[But she could just run Dec()!]]

[[Need sthg to distinguish Bob & Eve.]]

- Make Dec() a secret? × [[· probably leaked & can't eval security if don't know it]]

tenet: All algorithms should be public.

- Have secret inputs. ✓

Symmetric/shared key model: SK :=$ Gen()

[[a bit less usable in practice; we'll talk about getting around it w/ PKE]]

[[string]]

Alice: c :=$ Enc(m, SK) —Eve→ Bob: m = Dec(c, SK)

(randomized)

[[What key fpty must Gen() have? Randomized. Else Eve can run it!]]

Security? [[Eve can't learn SK? Eve can't output m? c looks random to Eve?]]

→ [[Simulatability: Eve can generate sthg indistinguishable from c herself. Seeing c, having A,B useless]]

def: An SKEncryption scheme (Gen, Enc, Dec) is perfectly secure if ∀ msgs m_0, m_1 and ∀ strings c, $\Pr_{SK \leftarrow Gen()}[Enc(m_0, SK) = c] = \Pr_{SK \leftarrow Gen()}[Enc(SK, m_1) = c]$

[[= "Shannon-secure", as invented by Shannon in '49]]

[In crypto, defs are very important, esp. if they have evocative names. If you don't feel this is a good def, fine. But we will prove thms invdlvg it. Just remember "perfectly secure" is a technical term.]

[Well known, ancient sol'n?]

def: One-time pad: Given n, M, C, SK all n-bit strings

$Gen() \to$ unif rand $SK \sim \{0,1\}^n$

$Enc(m, SK) := m \oplus SK$

$Dec(c, SK) := c \oplus SK$

$Dec(Enc(m, SK), SK)$? ✓

thm: It's perfectly secure.

pf: ✓ $\forall m, c, \Pr_{SK}[Enc(m, SK) = c] = 2^{-n}$ [unif rand]

[Great! We done? $|SK| \geq |M|$ ∴ (necessary for perf sec.) Hard to keep around. Terrible to use (easy more than once. Regs secret agreement ... Probably the main prob is that we want a short secret key that lets us encrypt lots & lots of msgs. There's no way to stop the attack of "try all SKs". So the key idea is to assume computational boldness.]

Main crypto assumption: Eve/adversary is PPT = probabilistic poly time (in the "security param" n)

- Honest parties should be PPT too

rem: • Advs allowed to be nonuniform; ie., circuits

[I.e. we allow them to precompute any poly amt of info based only on "n".]

[[Prob. adversaries could always theoretically guess your SK's (with very low prob) Have to allow possibility of failure, Want it smaller than any poly...]]

def: $\underline{\text{negl}(n)} := \frac{1}{\omega(n)}$; i.e., $f(n) \le \text{negl}(n)$ means $f(n) n^c \to 0$ $\forall c$. $n \to \infty$

[[Can weaken perf. security to say that the (randomized) encryption of any two distinct msgs "looks the same" to any PPT observer.]]

? PRG: $\{0,1\}^n \to \{0,1\}^{\ell(n)}$, $\ell \le \text{poly}(n)$ "looks rand / indistng from unif on $\{0,1\}^{\ell(n)}$" [[Could use to make OTPs w/ much shorter key]]

def: Let $\{X_n\}$, $\{Y_n\}$ be (seqs of) $\{0,1\}^{m(n)}$-valued r.v.s, ("ensembles") $m(n) \le \text{poly}(n)$. Computationally indistinguishable $X_n \overset{c}{\approx} Y_n$ means $\forall$ [[non-unif]] PPT A, $\underbrace{|\Pr[A(X_n)=1] - \Pr[A(Y_n)=1]|}_{\text{"Adv}_A(X_n, Y_n)\text{"}} \le \text{negl}(n)$

facts: $\overset{c}{\approx}$ is an eq. rel^n; in partic., $X_n \overset{c}{\approx} Y_n$, $Y_n \overset{c}{\approx} Z_n \Rightarrow X_n \overset{c}{\approx} Z_n$

[[In fact...]]

"Hybrid Lemma" / argument: Let $(X_n^1), \ldots, (X_n^T)$ be ensembles, $T \le \text{poly}(n)$, s.t. $X_n^i \overset{c}{\approx} X_n^{i+1}$ $\forall i$. Then $X_n^1 \overset{c}{\approx} X_n^T$.

pf: Let A be PPT. $|\Pr[A(X_n^1)=1] - \Pr[A(X_n^T)=1]|$

Δ-ineq. $\le \sum_{i=1}^{T-1} |\Pr[A(X_n^i)=1] - \Pr[A(X_n^{i+1})=1]|$.

$\le T \cdot \text{negl}(n) \le \text{negl}(n)$ $\because T(n) \le \text{poly}(n)$ $\square$

fact: If $X_n \overset{c}{\approx} Y_n$ and B is a PPT alg then $B(X_n) \overset{c}{\approx} B(Y_n)$

pf: _____

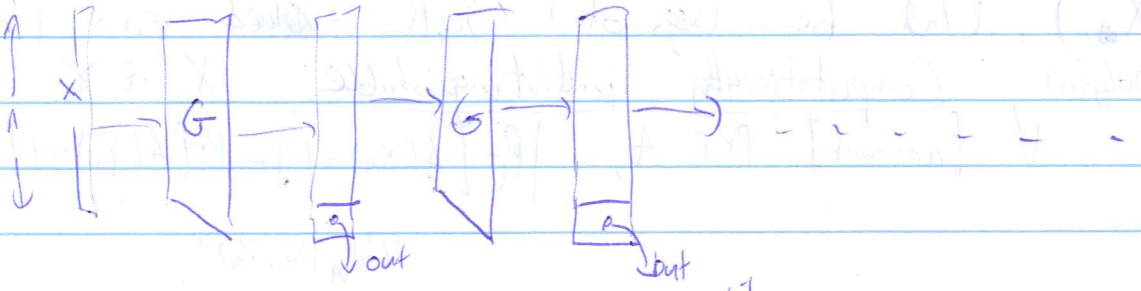
not^n: (U_n) denotes unif. random n-bit string.
[much stronger than the kinds considered in Lec 9]

def: A cryptographic PRG is a deterministic [unif] poly(n)-time computable $G: \{0,1\}^n \to \{0,1\}^{\ell(n)}$ ($\ell(n) > n$ $\forall n$) s.t. $G(U_n) \approx_c U_{\ell(n)}$.

Assumption/Conj.: Crypto PRGs with $\ell(n) = n+1$ exist. [We'll return to this.]
Rem: ~~Crypto~~ $\Rightarrow NP \neq P$ (implication $NP \not\subseteq BPP$ easy, at least)

thm [Blum, Goldreich-Levin (Boolean Fourier analysis!)] $\exists$ PRG with $\ell(n) = n+1$ $\Rightarrow$ $\exists$ PRG with $\ell(n) = n^c$ $\forall$ const. c

pf: Omitted [not too hard]; construction:



Conj: [Blum-Blum-Shub]: G: interp. n-bit input as two #s $1 \le X \le N$.
for $i = 1 \ldots \ell$
$X \leftarrow X^2 \bmod N$
Output $LSB(X)$.

Conj: For any $\ell(n) \le \text{poly}(n)$, this is a PRG.
"Thm": True if $\nexists$ poly-time alg. factoring $p \cdot q$ w/ more than $\text{negl}(n)$ [imprec. stated] prob. where p, q are rand n-bit primes $\equiv 3 \bmod 4$.

[Return to SKE...]

def: (Gen, Enc, Dec) is single-msg comp. secure if, for $SK \leftarrow Gen$ and n-bit msgs M_0, M_1 &
$(Enc(M_0, SK)) \approx_c (Enc(M_1, SK))$.

potentially n^{100}

Thm: Let $G: \{0,1\}^n \to \{0,1\}^{\ell(n)}$ be a (crypto) PRG. The following is single-msg secure: for $\ell(n)$-bit msgs

$SK \leftarrow Gen()$ (A / B)

$Enc(m; SK) = m \oplus G(SK)$ $\quad Dec(c; SK) = c \oplus G(SK)$

pf: For $SK \leftarrow Gen()$, any m, $\{Enc(m, SK)\} = \{m \oplus G(SK)\}$

$\approx_c \{m \oplus U(\ell(n))\}$ ← why?

$= \{U(\ell(n))\}$.

[So all msgs comp indist from random.]

[Sps $\exists$ A distinguishing $m \oplus G(SK)$ from $m \oplus U(\ell(n))$. Then $B(x) := A(x \oplus m)$ dists G from U] □

☺ Can do long msgs of len >> key.

☹ Still bad to use more than once. [Append a fresh key w/ every msg? Makes things stateful — what if you miss a msg?]

new: Better sec notion: "IND-CPA" → poly many msgs

ising. hosen plaintext → attacker can aff. encr'd plaintexts

Thm: IND-CPA also achievable fr. PRGs, reqs upgrading PRGs to PRF(unction) generators.

[very tough] HILL'88

PRG ⇒ PRF ⇒ SKE

OWFs ← One-way fcns: $f: \{0,1\}^* \to \{0,1\}^*$

- unif. det polytime computable
- $\forall$ PPTA, $\Pr_{x \sim U_n}[A(f(x), 1^n) \text{ outputs } y \text{ s.t. } f(y) = f(x)] \leq negl(n)$

OWF Existence ⇒ NP ≠ P. Most basic assump. in crypto.

weak-OWFs ⇒ "hard to invert"

$\Pr[\ ] \leq 1 - \delta$, $\delta = \frac{1}{poly(n)}$. Given weak f, $f'(x^{(1)}, \ldots, x^{(m)}) := (f(x^{(1)}), \ldots f(x^{(m)}))$, $m = 2n/\delta$.

S

Candidate OWF "Knapsack / Subset Sum"

$f(a_1, \ldots, a_n, S \subseteq [n]) := (a_1, \ldots, a_n, \sum_{i \in S} a_i \bmod N)$.

$\#s \bmod 2^n$ $S \subseteq [n]$, n bits

(Weak)-OWFs $\Rightarrow$ ---------- many things

$\stackrel{?}{\Rightarrow}$ Public Key Encr ?

Impagliazzo's Worlds
"Minicrypt"
"Cryptomania".

PKE (Gen, Enc, Dec): sec parm n.

A B

$(PK_B, SK_B) \leftarrow Gen(1^n)$

PK_B (public)

m: $c = Enc(m; PK_B)$ $\longrightarrow$ $Dec(c, SK_B) = m$!

1-bit msg

Security: For $(PK, SK) \leftarrow Gen$,

$\langle PK, Enc(0; PK) \rangle \stackrel{c}{\approx} \langle PK, Enc(1; PK) \rangle$

not hard thm: Given 1-bit sec. scheme, can construct IND-CPA scheme.

thm: "trapdoor $OWP_{permutation}$" $\Rightarrow$ PKE.

RSA
Diffie-Hellman $\Rightarrow$

↑ Conjectures about a problem being very hard-on-average
(all but neg! frac. of inputs comp. hard.

[Few explicit believable eg's of Trapdoor OWP]

(Regev '05) LWE $\Rightarrow$ PKE

thm: Hard-on-average assuming the "Gap $SVP_{n^{1.5}}$" problem on lattices is worst-case hard* [*Quantum!!!]

not known in P. best alg. $2^{n/\log n}$

∃ good complexity evidence it's NOT NP-hard.

[Best known LWE runtime is $2^{O(n)}$]

LWE (learning with errors) Assumption: Given n, fix $q = \text{poly}(n)$ (typically prime $\approx n^2$), $\alpha = \frac{1}{\text{poly}(n)}$ (typically $\approx \frac{1}{n \log^2 n}$), χ an "error distrib": $z \sim N(0, \alpha^2 q^2)$ [Gaussian w/ stddev αq]

- output $\lfloor z \rceil \bmod q$.

Say a "secret" $s \sim \mathbb{Z}_q^n$ chosen.
An alg can ask for "noisy linear eq's abt s":
→ gets "$a_1 s_1 + \dots + a_n s_n \approx b$"
where $a_1, \dots, a_n \sim \mathbb{Z}_q$ unif, $b := a_1 s_1 + \dots + a_n s_n + e$, where $e \sim \chi$.

Assump: no PPT A can output s whp.

[Rem: Roughly, Peikert de-quantified the worst-case to avg-case reduction, but w/ q exponential in n which is okay, but makes crypto apps not very practical.]

Regev's PKE: $\text{Gen}(1^n)$: $SK := s \sim \mathbb{Z}_q^n$ $(a^{(i)}, a^{(i)} \cdot s \approx b^{(i)})$
$PK :=$ m eqns drawn as above $(m = O(n \log q))$

$\text{Enc}(0, PK)$: · choose $S \subseteq [m]$ @ random
· ciphertext $= \left(\sum_{i \in S} a^{(i)}, \sum_{i \in S} b^{(i)} \right)$

$\text{Enc}(1, PK)$: same, but
$c = \left(\sum_{i \in S} a^{(i)}, \sum_{i \in S} b^{(i)} + \lfloor \frac{q}{2} \rfloor \right)$

$\text{Dec}((a, b), SK)$: if $a \cdot s - b$ closer to 0 than $\frac{q}{2}$, output 0
else output 1.

Correctness: If no "error", Dec() is always correct.
Wrong Dec occurs only if sum of m errors $\geq q/4$

x normal with stddev $\sqrt{m}\,\alpha q \approx \sqrt{\Theta(n \log n)}\,\frac{1}{\sqrt{n}\log^2 n}\cdot q$

$= \exp(-\log^{1.5} n) = \text{negl}(n)$.

Security proof not too bad.

Advantages of Lattice-based crypto:
- based on worst-case hardness assumption
- supports several crypto prims (eg "Fully Homomorphic Enc")
  not known using any other assumpts
- not broken by Shor's alg

Disadvs: ~~somewhat less efficient.~~ (NONE (?))

Lyubashevsky, Peikert, Regev...