Lecture 23 - Cryptography
[Hidden writing]

[Game do a no-\#-theory version]

[Script - Pass - Shелат]

Alice – Eve \[\rightarrow\] Bob

\[m, \text{Enc}(m) \rightarrow \text{Dec}(\cdot)\]

\[\text{msg}\]

\[\text{say gossip}\]

\[\text{"plaintext"}, \text{\"ciphertext\"}\]

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

\[\{\text{but she could just run Dec(\cdot)!!}\}

\[\text{Need sthg to distinguish Bob \& Eve.}\]

\[\text{Make Dec() a secret?} \times \]

\[\text{\'probably leaked\'; can't eval security if don't know it}\]

\[\text{tenet: All algorithms should be public}\]

\[\text{\# Have secret inputs, }\checkmark\]

Symmetric/shared key model:

\[\text{SK } \coloneqq \text{Gen()} \]

\[\text{\textcolor{red}{randomized}}\]

\[\text{\textcolor{red}{Not less usable in practice, we'll talk about getting past!}}\]

\[\text{Alice} \xrightarrow{\text{c} \coloneqq \text{Enc}(m, \text{SK})} \text{Bob} \]

\[\text{m} \coloneqq \text{Dec}(\text{c}, \text{SK})\]

\[\text{What key does \text{Gen()} have? Randomized Else Eve can run it!}\]

\[\text{Security: If Eve can't learn SK? Eve can't output m?}\]

\[c \text{ looks random to Eve?}\]

\[\text{Simulatability: Eve can generate sthg indistinguishable from c herself. Seeing c, have no useful}\]

def: An \text{SKE}ncryption scheme \text{(Gen, Enc, Dec)} is \underline{perfectly secure} if \forall \text{msg s} M_0, M_1 \text{ and } \forall \text{strings c,}\]

\[\text{Pr}[\text{Enc}(\text{Gen}(\text{SK})) = c] = \text{Pr}[\text{Enc}(\text{SK}) = c] = \text{negl}()\]

\[\text{SK \& Gen()}\]

\[\text{=} \text{Shannon - secure, as invented by Shannon in '49}\]
In crypto, def's are very important, esp. if they have evocative names. If you don't feel this is a good def, fine, but we will prove this involving 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,

\[ \text{Gen}() \rightarrow \text{uniform and } SK \sim \{0,1\}^n \]

\[ \text{Enc}(m, SK) := m \oplus SK \]

\[ \text{Dec}(c, SK) := c \oplus SK \]

\[ \text{Dec}(\text{Enc}(m, SK), SK) \]

Thm: It's perfectly secure.

Pr: \[ \forall m, c, \text{ Pr} [\text{Enc}(m, SK) = c] = 2^{-n} \]

[\text{uniform rand}]  

[Great! We done? (SK > m) is necessary for perf sec.]  

Hard to keep around. [Should we use even if only once, Rags 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 SK's." So the key idea is to assume computational hardness.

Main crypto assumption: Eve/adversary is PPT \( \leq \) probabilistic poly time.

(in the "security param" \( n \))

- Hosts parties should be PPT too.

\[ \text{Rem} . \text{ Advs. allowed to be nonuniform; i.e., circuits} \]

[I.e., we allow them to precompute any poly amount of info based only on \( n \).]
Prob. adversaries could always theoretically guess your SK's (with any low prob).

Have to allow possibility of failure, want it smaller than any poly. -

\[
def: \negl(n) := \frac{1}{n^{\omega(1)}} \quad \text{i.e.} \quad \forall \epsilon > 0 \exists n \in \mathbb{N} : \negl(n) < \epsilon
\]

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

\[\text{PRG: } \{0,1\}^n \rightarrow \{0,1\}^{m(n)} \quad \text{(looks random/indistinguishable from uniform \{0,1\}^{m(n)})}\]

"Could we do this with much shorter key?"

\[\text{def: Let } \{X_n\}, \{Y_n\} \text{ be (sets of) } \{0,1\}^{m(n)} - \text{valued rvs, } \text{"ensembles"}
\]

\[m(n) \leq \text{poly}(n), \quad \text{Computationally indistinguishable}\]

\[\text{means } \forall [\text{non-unif. PPT } A], \quad \left| \Pr[A(X_n)=1] - \Pr[A(Y_n)=1] \right| \leq \negl(n)
\]

In fact... In fact...

"Hybrid Lemma": Let \( (X_n^1), ..., (X_n^T) \) be ensembles, \( T(n) \leq \text{poly}(n) \)

s.t. \( X_n^i \subseteq X_n \quad \forall i \). Then \( X_n^i \approx X_n \)

pf: Let \( A \) be PPT.

\[\left| \Pr[A(X_n^1)=1] - \Pr[A(X_n^T)=1] \right|
\]

\[= \sum_{i=1}^{T(n)} \left| \Pr[A(X_n^i)=1] - \Pr[A(X_n^{i+1})=1] \right|
\]

\[\leq T(n) \cdot \negl(n) = \negl(n) \quad : T(n) \leq \text{poly}(n) \]

Fact: If \( X_n \approx Y_n \) and \( B \) is a PPT alg. then \( B(X_n) \approx B(Y_n) \)

pf: 
def: A cryptographic PRG is a deterministic poly(n)-time computable
\( G: \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)} \) s.t.
\( G(U_n) \approx U_{\ell(n)}. \)

Assumption/Conj.: Crypto PRGs with \( \ell(n) = n+1 \) exist. [We'll return to this.]

Rem: \( \text{NP} \neq \text{P} \) implies \( \text{NP} \neq \text{BPP} \) easy, at least.

Thm: Shaltiel, Goldreich-Levin [FPRR]
Existence of FPRG with \( \ell(n) = n+1 \) \( \Rightarrow \) Existence of FPRG with \( \ell(n) = n^c \)

pf: Omitted (not helpful); construction:

[Diagram of a construction process]

Blum-Blum-Shub:
- \( G \): 2n-bit input as two #s \( 1 \leq X \leq N \)
- For \( i = 1, \ldots, d \)
  \( X \leftarrow X^2 \mod N \)
- Output LSB(X)

Conj: For any \( \ell(n) = \text{poly}(n) \), this is a PRG.

"True": True if \( \exists \) poly-time alg. factoring \( pq \) w/ more than negligible
(implies stated) probability, where \( pq \) are \( n \)-bit primes \( \equiv 3 \text{ \( \mod 4 \)} \)

Return to SKC...

def: \((\text{Gen, Enc, Dec})\) is single-msg comp. secure if, for \( SK \leq \text{Gen} \)
\( (\text{SK} \leq \text{Gen}, \text{Enc}(M, SK)) \approx (\text{SK} \leq \text{Dec} \circ \text{Enc}(M, SK)) \)
Let $G : 30, B^n \rightarrow \mathcal{E}(n)$ be a (crypto) PKE. The following is single-msg secure:

$$SK \leftarrow \text{Gen}(1)$$

for $n$-bit msgs $A$:

$$\text{Enc}(m, SK) = \text{msg} \oplus G(SK)$$

$$\text{Dec}(c; SK) = c \oplus G(SK)$$

Proof: For $SK \leftarrow \text{Gen}(1)$, any $m$, $\{\text{Enc}(m, SK)\} = \{\text{msg} \oplus G(SK)\}$

$\oplus \{\text{msg} \oplus U(n)\}$

So all msgs indist. from random.

If $S_{x, y} \neq \text{distinguishing msg} \oplus G(SK)$

from $\text{msg} \oplus U(n)$ Then

$B(x) = A(x \oplus n)$ 

Can do lazy msg of $\text{len} \rightarrow \text{key}$

Still bad to use more than once. A fresh key w/ every msg? Makes things stateful:

Better see notion: "IND-CPA" — poly many msgs

Better security notion: "IND-CPA" — poly many msg pairs

Thm: IND-CPA also achievable ft. PRGs, easy upgrading PRGs to

PKE via generators

PRG $\rightarrow$ PRF $\rightarrow$ SKE

OWFs $\rightarrow$ One-way i.e.: $f: 30, B^* \rightarrow \mathcal{E}(n)$

uniform, deter polynomial computable

"hard to invert" $\cdot$ A PPTA, Pr $[A(f(x), 1^n) \text{ outputs } y \text{ s.t. } f(y) = f(x)] \leq \text{negl}(n)$

$P: |\delta| \leq 1 - \text{negl}(n)$

Given oracle $f$, $\delta(x_0, \ldots, x_{\ell-1}) = (f(x_0) \ldots f(x_{\ell-1}))$. $m = 2^{\ell/6}$
Candidate OVF \( f(a_1, \ldots, a_n) : (a_1, \ldots, a_n) \in \mathbb{Z}_{q_i}^n \rightarrow \mathbb{Z}_{q_i} \), encrypted by \( \text{Krepsack/Subset Sum} \).

It's mod \( 2^n \) and \( n \) bits.

\[ \text{Impagliazzo's Worlds} \]

\[ \text{Minnecrypt} \]

\[ \text{Crypto Mania} \]

\[ \text{Public Key Encr?} \]

**PKE** (Gen, Enc, Dec):

- **A**
- **B**

\[ (\mathbb{PK}, \mathbb{SK}) \leftarrow \text{Gen}(t) \]

\[ \mathbb{PK} \]

\[ \mathbb{SK} \]

\[ \mathbb{SK} \]

1-bit org

**Security**:

For \( (\mathbb{PK}, \mathbb{SK}) \leftarrow \text{Gen} \),

\[ \langle \mathbb{PK}, \text{Enc}(0, \mathbb{PK}) \rangle \neq \langle \mathbb{PK}, \text{Enc}(1, \mathbb{PK}) \rangle \]

Not hard then: Given 1-bit sec. scheme, can construct IND-CPA scheme.

**Thm**: "trapdoor OWP \( \Rightarrow \) PKE.” Few explicit believable eg's of trapdoor OWP?

**RSA**

**Diffie-Hellman**

Conjectures about a problem being very hard on average have been difficult to prove. Almost everything is based on hardness assumptions.

\[ \text{LWE} \Rightarrow \text{PKE} \]

**Thm**: Hard on average assuming the "Gap SVP_{ns}" problem on lattices is worst-case hard. *WOW!!!*

Not known in P. Best alg. \( Z_4/\# \)

Good complexity evidence it's not NP-hard.
LWE Assumption: Given $n$, Fix $q = \text{poly}(n)$, (typically prime $\approx n^2$) $X$ an "error distrib": $z \sim N(0, \frac{q^2}{2})$

Say a "secret" $s \sim \mathbb{Z}_q^n$ chosen. An alg can ask for "noisy linear eqs" abt. $s$:

- get $a_1s_1 + \ldots + a_ns_n \approx 6$
- where $a_i \ldots a_n \sim \mathbb{Z}_q$ unif, $6 : q, s_1, \ldots, s_n + \epsilon$, where $\epsilon \sim X$.

Assump: no PPT $A$ can output $> \frac{1}{q}$ why.

(Rem: Regev de-quantified the worst-case to avg-case reduction, but w/ exponential in $n$, which is okay, but makes crypto avg not very practical.)

Regev's PKE: Gen(1$^n$):

- $SK : s \sim \mathbb{Z}_q^n$:
- $PK : m$ eqns drawn as above: $(m = \mathbb{Z}_q^n)$

Enc(0, PK):

- choose $\Gamma \subseteq [m] \sim \text{random}$
- ciphertext $= (\sum a_i^{(0)}, \sum b_i^{(0)})$

Enc(1, PK):

- same, but
- ciphertext $= (\sum a_i^{(1)}, \sum b_i^{(1)} + \frac{q}{2})$

Dec $(a \oplus b, SK)$: if $a \cdot s - 6 \text{ closer to 0}$ then $\frac{a}{2}$, output 0
- else output 1
Correctness: If no "error", Dec is always correct.
Wrong Dec occurs only if sum of m errors > 1/4

\[ x \text{ normal with stddev } \sqrt{m/\theta} \leq \sqrt{\Theta(n)} \text{ for } n \in \mathbb{Z} \]

\[ = \exp(-\log^2(n)) = \text{negl}(n) \]

Security proof not too bad.

Advantages of Lattice-based crypto:
- Based on worst-case hardness assumption
- Supports several crypto prims (e.g., "Fully Homomorphic Enc")
- Not known using any other assumptions
- Not broken by Shor's alg

Disadvantages: somewhat less efficient. None (?)