

Computational Security

It focuses on efficient adversaries (attacker is a PPT Turing machine)

We also admit a small chance of success, where "small" = negligible.

We would like to parametrize everything by a security parameter.

DEF A function $\varepsilon: \mathbb{N} \rightarrow [0, 1]$ is negligible if

$$\forall p(\lambda) = \text{poly}(\lambda), \exists \lambda_0 \in \mathbb{N} \mid \forall \lambda \geq \lambda_0 \varepsilon(\lambda) \leq \frac{1}{p(\lambda)}$$

Equivalent: $\varepsilon(\lambda) = O(1/p(\lambda))$
 $\forall p(\lambda) = \text{poly}(\lambda)$

for each polynomial, ε is smaller than its own inverse.
It is an arbitrary way to say that something is small

Intuition: Think of some algorithm for solving some problem.
Assume algorithm successful w.p. p (i.e. fails w/ prob. $1-p$).

Say $p = 1/2$.

$$\Pr[\text{FAIL after } k \text{ times}] \leq \frac{1}{2^k}$$

Assume instead that $p = 1/\lambda$ but we don't know

if the algorithm is successful

Ex. Let $p(\lambda), p'(\lambda) = \text{poly}(\lambda)$

$\varepsilon(\lambda), \varepsilon'(\lambda) = \text{negl}(\lambda)$

$C \in \mathbb{N}$

Then:

i) $p(\lambda) \cdot p'(\lambda) = \text{poly}(\lambda)$

ii) $p'(p(\lambda)) = \text{poly}(\lambda)$

iii) $\varepsilon(\lambda) + \varepsilon'(\lambda) = \text{negl}(\lambda)$

iv) $p(\lambda) \cdot \varepsilon(\lambda) = \text{negl}(\lambda)$

v) $\varepsilon(\sqrt[C]{\lambda}) = \text{negl}(\lambda)$

Idea: Exploit computational hardness.

(There are some tasks that don't run in poly time, and are really hard for Turing machines)

In particular, the fact that $P \neq NP$

For example, factoring $n = p \cdot q$ where p, q are primes with λ bits.

(it's impossible to solve polynomially)

DEF: ONE-WAY FUNCTION. A deterministic function

$f: \{0,1\}^* \rightarrow \{0,1\}^*$ is a OWF if f can be computed in poly-time, and

\forall PPT attackers $A \exists \epsilon(\lambda) = \text{negl}(\lambda)$ s.t.

$$\Pr[f(x') = f(x) : x \leftarrow \$ \{0,1\}^* \Rightarrow \$ = \text{random uniform} \\ x' \leftarrow \$ A(1^\lambda, f(x))] \leq \epsilon(\lambda)$$

Equivalent: Consider

$$\text{GAME}_{A,f}^{\text{owf}}(\lambda)$$

$$A(1^\lambda) \xleftarrow{y} C(1^\lambda)$$

$$x \leftarrow \$ \{0,1\}^*; y = f(x)$$

$$\xrightarrow{x'} \text{OUTPUT } 1 \text{ if and only if } f(x') = y$$

\forall PPT $A \exists \epsilon(\lambda) = \text{negl}(\lambda)$ s.t.

$$\Pr[\text{GAME}_{A,f}^{\text{owf}}(\lambda) = 1] \leq \epsilon(\lambda)$$

But why 1^λ ? Take $f(x) = |x|$

$$\forall x \in \{0,1\}^\lambda$$

So $|x| = \lambda$, but $|f(x)| = \log \lambda$

This function is not a OWF.

If A takes λ as input, it runs in time $\text{poly}(\log \lambda)$

EX. Let $f: \{0,1\}^{n(\lambda)} \rightarrow \{0,1\}^{n(\lambda)}$

Show that there exists

① INEFFICIENT A breaking f w.p. $\epsilon = 1$

② POLY-TIME A breaking f w.p. $2^{-n(\lambda)} = \text{negl}(\lambda)$

\Rightarrow we need $n(\lambda) = \omega(\log \lambda)$

(SUPER LOGARITHMIC)

otherwise $n(\lambda) = O(\log \lambda)$ and 2^{-n} would not be negligible

Q: Is $P \neq NP$ equivalent to assuming OWFs?

A: We don't know, but for sure if OWFs exist, then $P \neq NP$

We know that OWFs are equivalent to

ONE-WAY PUZZLES (PGen, PVer)

PGen: Outputs a solved instance of PUZZLE

$y = \text{PUZZLE}$, $x = \text{SOLUTION}$

PRV: Verifies if x is solution for y .

OWFs \Leftrightarrow ONE-WAY PUZZLE

The computational worlds (Russel Impagliazzo):

① ALGORITHICA: $P = NP$

② HEURISTICA: $P \neq NP$

but no average-hard puzzles (PGen outputs y but not x)

③ PESSILAND: $P \neq NP$

average-hard puzzles but no OWFs.

④ MINICRYPT: $P \neq NP$, OWFs.

← we assume we're at least here

⑤ CRYPTOMANIA: $P \neq NP$, OWFs but no public-key CRYPTOGRAPHY

GOAL: OWFs imply SECURE SKC beating Shannon

GOAL: OWFs imply COMP. SECURE MACs beating INF. THEORETIC lower bounds

GOAL: PUBLIC-KEY CRYPTOGRAPHY (would require more than OWFs)

PSEUDORANDOM GENERATOR (PRG)

A PRG $G: \{0,1\}^\lambda \rightarrow \{0,1\}^{\lambda+\ell}$ with $\ell \geq 1$
↳ STRETCH

→ Efficiently computable

→ Secure: no PPT attacker can distinguish $G(s)$ with $s \leftarrow \$ \{0,1\}^\lambda$ from $z \leftarrow \$ \{0,1\}^{\lambda+l}$

To formalize security:

COMPUTATIONAL INDISTINGUISHABILITY

DEF: Let $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$, $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$

↳ a set of random variables

We say that $X \approx_c Y$ if

↳ distinguisher ↳ computationally close

\forall PPT \mathcal{D} , $\exists \epsilon(\lambda) = \text{neg}(\lambda)$ s.t.

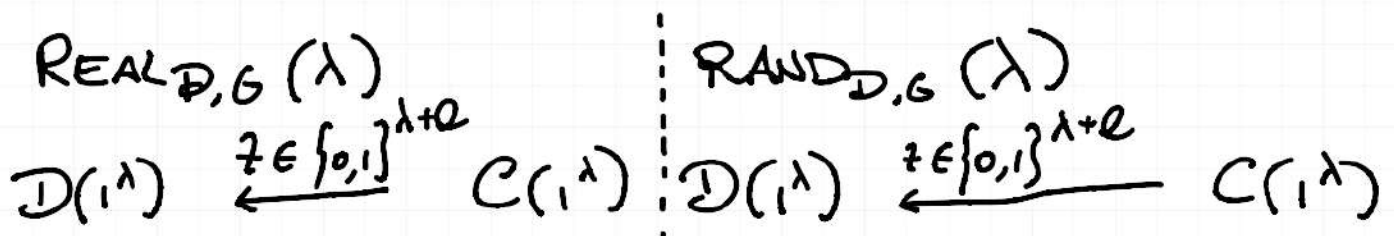
$$|\Pr[\mathcal{D}(X_\lambda) = 1] - \Pr[\mathcal{D}(Y_\lambda) = 1]| \leq \epsilon(\lambda)$$

For sufficiently large λ , the probability that the distinguisher can distinguish X from Y is negligible

The distinguisher gets a variable z picked from X or Y , and says if he thinks that $z \in X$ or $z \in Y$

For the PRG: $X_\lambda \equiv \text{REAL}_{\mathcal{D},G}(\lambda)$

$Y_\lambda \equiv \text{RAND}_{\mathcal{D},G}(\lambda)$



$$0/1 \xrightarrow{\quad} \begin{array}{l} s \leftarrow \{0,1\}^\lambda \\ z = G(s) \end{array} \quad \text{---} \quad 0/1 \xrightarrow{\quad} z \leftarrow \{0,1\}^{\lambda+e}$$

DEF: A PRG G is secure :f $\forall \text{PPT } \mathcal{D}$

$$\left\{ \text{REAL}_{\mathcal{D}, G}(\lambda) \right\}_{\lambda \in \mathbb{N}} \approx_c \left\{ \text{RAND}_{\mathcal{D}}(\lambda) \right\}_{\lambda \in \mathbb{N}}$$

if you can't distinguish G from real random source,
it's secure

PLAN: ① $\text{ONF}_S \Rightarrow \text{PRGs}$
② $\text{PRGs} \Rightarrow \text{ONE-TIME COMP. SECURE SKE}$

RECALL: Perfect secrecy has two limitations:

① $|K| \geq |M|$

② ONE-TIME SECURITY

(such as the one time pad $C = m \oplus k$)

How to REMOVE ① with a PRG?

Short random secret key $k \in \{0,1\}^\lambda$

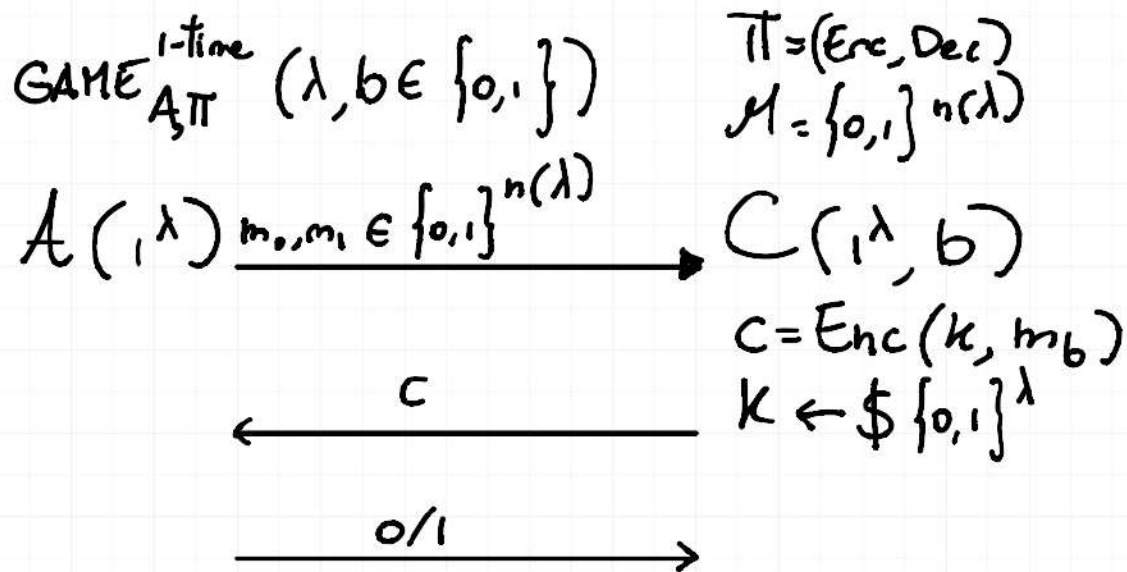
Let $G: \{0,1\}^\lambda \rightarrow \{0,1\}^{n(\lambda)}$ → $n(\lambda) \gg \lambda$

$$\text{Enc}(k, m \in \{0,1\}^n) = G(k) \oplus m = c$$

$$\text{Dec}(k, c) = G(k) \oplus c$$

What security? This k satisfies a meaningful notion of security

This security is **ONE-TIME COMPUTATIONAL SECURITY**



DEF: A SKC Π is ONE-TIME COMP. SECURE

if $\forall \text{ PPT } A, \exists \epsilon(\lambda) = \text{negl}(\lambda)$ s.t.

$$\left\{ \text{GAME}_{A, \Pi}^{\text{1-time}}(\lambda, 0) \right\} \approx_c \left\{ \text{GAME}_{A, \Pi}^{\text{1-time}}(\lambda, 1) \right\}$$

The adversary can't know if m_0 or m_1 is encrypted!