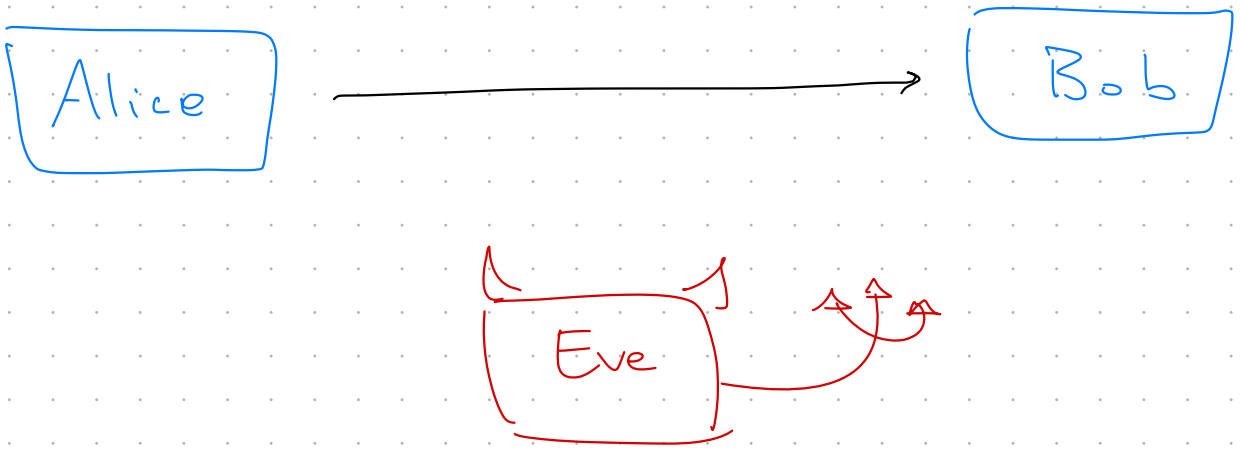
30 April 2025

One-Way Functions

Plan

* Cryptography from Computational Hardness
* Announcements
* One-Way Functions

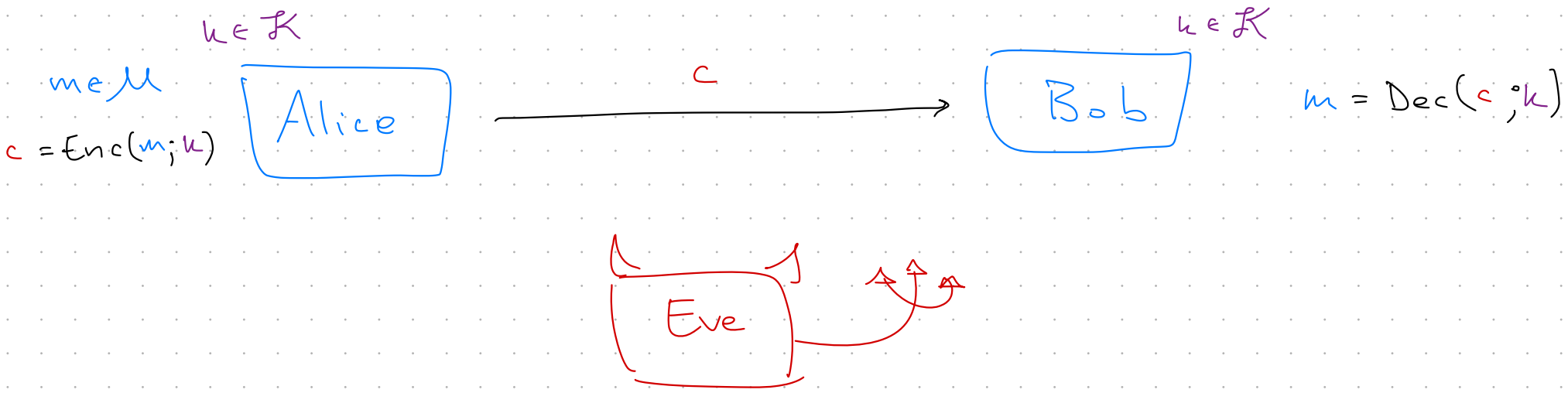
Motivation: Secure Communication



Goal:

* Alice sends message to Bob
* Eve observes transmission, but should learn nothing.

Motivation: Secure Communication



Encryption Scheme

$k \leftarrow Gen()$

$c \leftarrow Enc(m; k)$

$m \leftarrow Dec(c; k)$

① **Functionality**

$$Dec(Enc(m; k); k) = m$$

② **Secrecy**

Eve shouldn't learn about m given c

Information-Theoretic Solution: One-Time Pad

$$c = m \oplus k$$

$\Rightarrow$ c is uniformly random bit string (for random k)

$\Rightarrow$ leaks no information about m

Information-Theoretic Solution: One-Time Pad

$$c = m \oplus k$$

$\Rightarrow$ c is uniformly random bit string (for random k)

$\Rightarrow$ leaks no information about m

Problem: Infeasible!

↳ Requires a new key for every message.

Proposed Solution: Computationally - Bounded Adversaries

$\in$ polynomial time

Proposed Solution: Computationally-Bounded Adversaries

$\in$ polynomial time

Semantic Security (Goldwasser & Micali)

An Encryption Scheme is semantically secure if every efficient eavesdropper learns (essentially) nothing about message m given $c = Enc(m; k)$.

Computational Security via Reduction

* Suppose problem $\Pi \notin P$.

* Build Encryption Scheme.

Computational Security via Reduction

* Suppose problem $\Pi \notin P$.
* Build Encryption Scheme.
* Show Reduction (Security Proof)

Π reduces to (in poly-time) Breaking Scheme

$\Rightarrow$ Scheme is secure against poly-time adversaries!

(else $\Pi \in P$)

Announcements

* HW8 due
* HW9 optional for 4820 (graded on completion)
  required for 5820

* Final May 13
  ↳ info about exam & review materials will be posted to Ed

Computational Security via Reduction

* Suppose problem $\Pi \notin P$.
* Build Encryption Scheme.
* Show Reduction (Security Proof)

Π reduces to (in poly-time) Breaking Scheme

$\Rightarrow$ Scheme is secure against poly-time adversaries!

(else $\Pi \in P$)

Computational Security via Reduction

* Suppose problem $\Pi \notin P$.
* Build Encryption Scheme.
* Show Reduction (Security Proof)

Π reduces to (in poly-time) Breaking Scheme

$\Rightarrow$ Scheme is secure against poly-time adversaries!

(else $\Pi \in P$)

Computational Hardness for Cryptography

* Need a problem that is Hard.

$$\Pi \neq P$$

* But NP-Hardness may not work...

Computational Hardness for Cryptography

* Need a problem that is Hard.

$$\Pi \neq P$$

* But NP-Hardness may not work...

↳ Π must be hard on typical instances

(3SAT hard in worst-case)

Computational Hardness for Cryptography

* Need a problem that is Hard.

$$\Pi \notin P$$

* But NP-Hardness may not work...

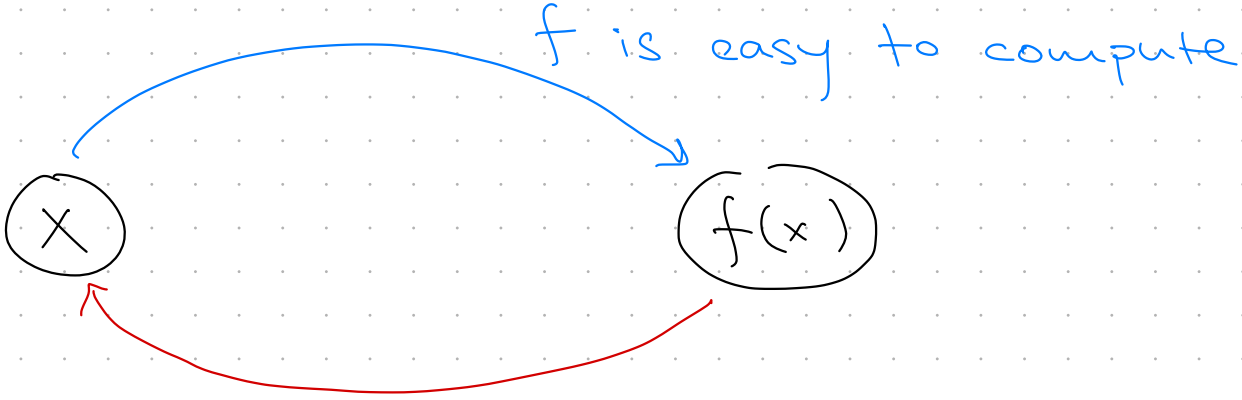
↳ Π must be hard on typical instances

(3SAT hard in worst-case)

↳ Need to be able to generate solved instances of Π easily.

(Given a SAT formula, how do we know if it's SAT or UNSAT ??)

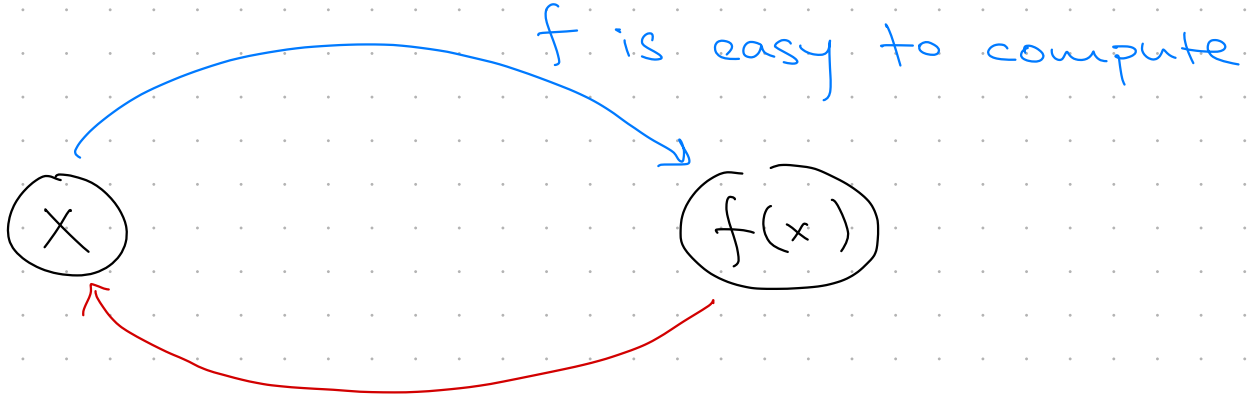
One-Way Functions



// Given f(x), hard to find x

(even for RANDOM x)

One-Way Functions



f is HARD to invert

Candidate OWFs

Multiplication vs. Factoring

$p \cdot q = N$

Modular Exponentiation vs. Discrete Log

$g^x \bmod N = c$

Def. A **one-way function** $f: \{0,1\}^* \longrightarrow \{0,1\}^*$ satisfies

* **Computation EASY**: there is an algorithm that given x, returns $f(x)$ in $\text{poly}(|x|)$ time.

Def. A **one-way function** $f: \{0,1\}^* \to \{0,1\}^*$ satisfies

* **Computation EASY**: there is an algorithm that given x, returns $f(x)$ in $\text{poly}(|x|)$ time.

* **Inversion HARD**: For **all** polynomial-time algorithms A

$$\Pr_{x \sim \{0,1\}^n}\left[A(1^n; f(x)) = x' \quad \wedge \quad f(x') = f(x)\right] \leq 1/n^c$$

for all $c \in \mathbb{N}$ and sufficiently large n.

↳ Inversion succeeds w/ "Negligible" probability.

P vs. NP & One-Way Functions

Fact. If one-way functions exist,

$$P \neq NP.$$

P vs. NP & One-Way Functions

Fact. If one-way functions exist,

$$P \neq NP.$$

So. $P = NP \Rightarrow$ No Cryptography!

P vs. NP & One-Way Functions

Fact. If one-way functions exist,

$$P \neq NP.$$

So. $P = NP \Rightarrow$ No Cryptography!

Open Question. Does $P \neq NP$ imply the existence of OWFs?

(See recent works of Prof. Rafael Pass & collaborators)

Candidate OWF: Multiplication

$$f_{mult} : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^{2n}$$

$$f_{mult}(p, q) = p \cdot q = N$$

Factoring

Given $N \in \{0,1\}^{2n}$, find factors s.t. $p \cdot q = N$

Candidate OWF: Multiplication

$$f_{mult} : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^{2n}$$

$$f_{mult}(p, q) = p \cdot q = N$$

Factoring

Given $N \in \{0,1\}^{2n}$, find factors s.t. $p \cdot q = N$

Conjecture: Factoring $\notin$ P.

Current Best Classical Attack: $2^{\tilde{O}(n^{1/3})}$

Candidate OWF: Multiplication

$$f_{mult} : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^{2n}$$

$$f_{mult}(p, q) = p \cdot q = N$$

Factoring

Given $N \in \{0,1\}^{2n}$, find factors s.t. $p \cdot q = N$

Conjecture: FACTORING $\notin$ P.

Current Best Classical Attack: $2^{\tilde{O}(n^{1/3})}$

Shor's Quantum algorithm: poly-time!

FACTORING $\in$ BQP.

Is Multiplication a OWF?

$\forall$ efficient algorithms A

$$\Pr_{p,q \text{ random}}\left[A(1^n; p\cdot q) = p'\cdot q' \wedge p'\cdot q' = p\cdot q\right] \leq \text{negligible}$$

Is Multiplication a OWF?

$\forall$ efficient algorithms A

$$\Pr_{p,q \leftarrow \text{Primes}_n}\left[A(1^n; p \cdot q) = p' \cdot q' \wedge p' \cdot q' = p \cdot q \right] \leq \text{negligible}$$

↑ with probability at least 3/4 p or q is <u>even</u>.

Is Multiplication a OWF?

$\forall$ efficient algorithms A

$$\Pr_{p,q \leftarrow \text{Primes}_n}\left[A(1^n, p \cdot q) = p' \cdot q' \wedge p' \cdot q' = p \cdot q \right] \leq \text{negligible}$$

↑ with probability at least 3/4 p or q is even.

$\Rightarrow$

```
A_even(N):
    Compute N/2.
    if integer: return (2, N/2)
    Return ⊥.
```

succeeds w.p. 3/4.

Takeaway

* Distribution of instances is critical!

Problem $p, q \sim \{0,1\}^n$ has too many factors!

EASY

Takeaway

* Distribution of instances is critical!

Problem $p, q \sim \{0,1\}^n$ has too many factors!

EASY

Fix. Sample $p, q \sim$ Random n-bit primes

$\Rightarrow$ only prime factors of $N = p \cdot q$

are p & q.

Takeaway

* Distribution of instances is critical!

Problem $p, q \sim \{0,1\}^n$ has too many factors!

EASY

Fix. Sample $p, q \sim$ Random n-bit primes

$\Rightarrow$ only prime factors of $N = p \cdot q$

are p & q.

Assumption. Factoring the product of two n-bit primes is infeasible.

$\Rightarrow$ OWF.