CS 121 Lecture 24 Nov 21, 2019

## Modern Cryptography

- DH / RSA are simpler than Enigma, and allow public key

## Players

Algos: $E, D$

$x = D_k(y)$

**Transmitter** $y = E_k(D)$
- plaintext $x \in \{0,1\}^*$
- secret key $k \in \{0,1\}^*$

**Receiver**
- secret key $k \in \{0,1\}^*$

**3rd party**
- unbounded computational power

## Encryption Definition

For $k \in \{0,1\}^n$, $E_k : \{0,1\}^{L(n)} \to \{0,1\}^{C(n)}$, $D_k : \{0,1\}^{C(n)} \to \{0,1\}^{L(n)}$

- Validity: $(E, D)$ is valid if $\forall k \in \{0,1\}^n \ \forall x \in \{0,1\}^{L(n)}$
  $D_k(E_k(x)) = x$
- Security: defined for every message but for random key

"no secrecy w/o randomness"

- $(E, D)$ is secure if Adversary cannot learn anything about the plaintext
- Shannon: $(E, D)$ is perfectly secret if for every $x, x' \in \{0,1\}^{L(n)}$, $\{E_k(x)\}_{k \sim \{0,1\}^n}$, $\{E_k(x')\}_{k \sim \{0,1\}^n}$ are identical distributions

↳ corollary:

$$\Pr\left[\begin{array}{l}\text{Adversary guesses whether} \\ y = E_k(x) \text{ or } y = E_k(x')\end{array}\right] \le \frac{1}{2}$$

## Unbreakable Encryption

Thm: $\exists$ perfectly secret encryption

Proof:

ex one-bit perfectly secret encryption

| $x \backslash k$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

observing cypher text has <u>no</u> impact on your knowledge of the plaintext

Encrypt $\ell$ bits: repeat $\ell$ ind. keys

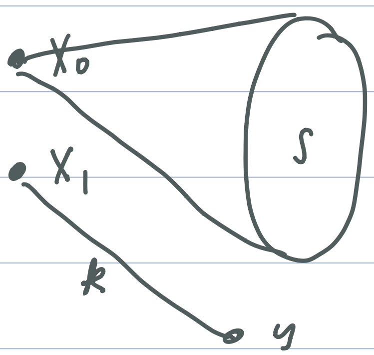
one-time pad

$x \in \{0,1\}^\ell$, $k \in \{0,1\}^\ell$, $E_k(x) = x \oplus k$

## Limitation of Perfect Secrecy

Thm 20.5: If $(E,D)$ is perfectly secret, then

|keysize| $\ge$ |message size|




$2^n$ size

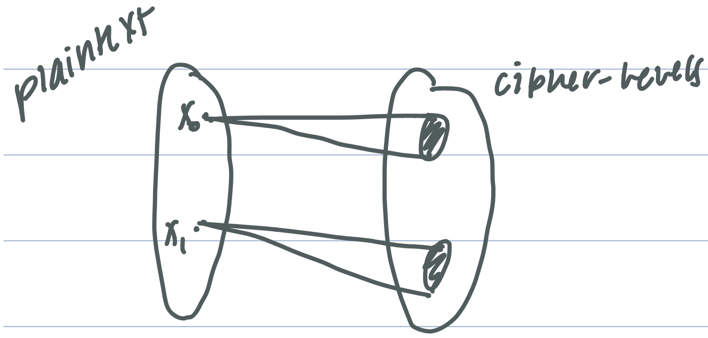
$y \notin S$

observing $y \rightarrow$ wasn't encryption $x_0$ by trying all possible keys ◻

## Computational Secrecy

· adversary runs in polynomial time

· can always break encryption schemes w/ |key| << |message| with unbounded computational time




$y \in \{0,1\}^m \quad m > 2n$

BREAK$(y)$:

if $\exists\, k \in \{0,1\}^n$ s.t. $y = E_k(x_0)$:

output "$x_0$"

else

output "$x_1$"

## Public key

public key: $e \in \{0,1\}^*$

Transmitter

public key: $e \in \{0,1\}^*$

cyphertext: $y = E_e(x)$

Reciever

$x = D_d(y)$

generate key pair

$r \sim \{0,1\}^n$

$(e, d) = G(r)$

Adversary

public key: $e \in \{0,1\}^*$

Impossible to achieve if $P = NP$

· Diffie-Hellman: discrete logarithm

· Rivest Shamir Adelman: factoring

RSA

## Fully Homomorphic Encryption

Thm. Exists secure encryption where

$\boxed{x} \quad \boxed{x'} \rightarrow \boxed{NAND(x,x')}$

Algorithm EVAL$(c, c')$:

EVAL doesn't get key

EVAL doesn't get key!

$$D_k(EVAL(E_k(X), E_k(X'))) = NAND(X, X')$$

## Applications

ex: secret data to store on cloud

Solution 1: Encrypt information, store encrypted data on kiwi.com

Problem: Can ask for total sales in july?

```
for i in range(n):
    if(X[i].month == "July"):
        total += X[i].sale
return total
```

→ unroll loop into NAND-CIRC

CLIENT

key

$C : \{0,1\}^n \to \{0,1\}^l$

SERVER

$E_k(X_0) \cdots E_k(X_{n-1})$

$E_k(X_i), E_k(X_j) \to E_k(NAND(X_i, X_j))$