28 April 2025

Cryptography

Plan

* Motivation
* Announcements
* Shannon Secrecy vs. Semantic Security.

4820 Recap

* Algorithms
  - Greedy
  - Dynamic Programming
  - Flow
  - Mathematical / Divide & Recurse

* Complexity
  - P vs. NP
  - NP-Hard Problems
  - Even harder! (Undecidable)

4820 Recap

* Algorithms
  - Greedy
  - Dynamic Programming
  - Flow
  - Mathematical / Divide & Recurse

* Complexity
  - P vs. NP
  - NP-Hard Problems
  - Even harder! (Undecidable)

Algorithms + Complexity = Cryptography!

Cryptography

* Classic Definition: Practice & Study of secure communication in the presence of adversaries

Cryptography

* Classic Definition: Practice & Study of secure communication in the presence of adversaries

* Modern Take: Practice & study to enable mutually-distrustful parties to cooperate reliably

Cryptography

* Classic Definition: Practice & Study of secure communication in the presence of adversaries

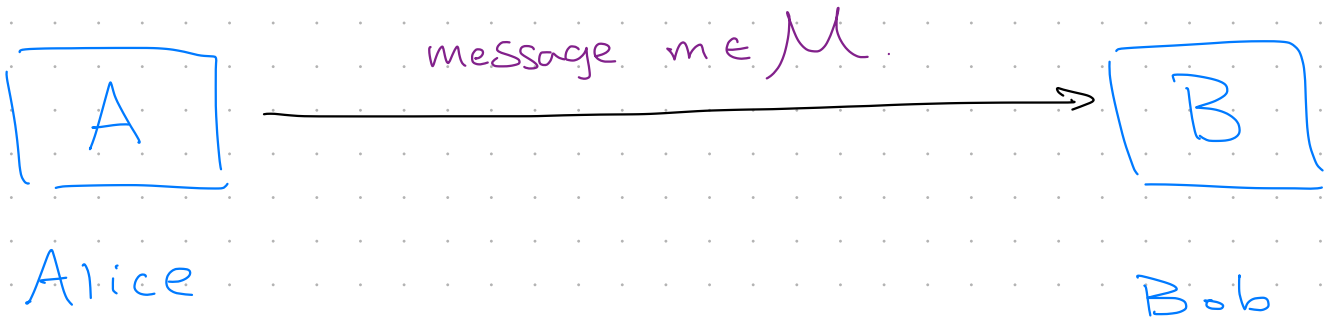
* Modern Take: Practice & study to enable mutually-distrustful parties to cooperate reliably

Cryptographic Functionalities

- Encryption
- Digital Signatures
- Secure Multi-Party Computation
- Zero-Knowledge Proof Systems
- Verifiable Machine Learning .....

Secure Communication

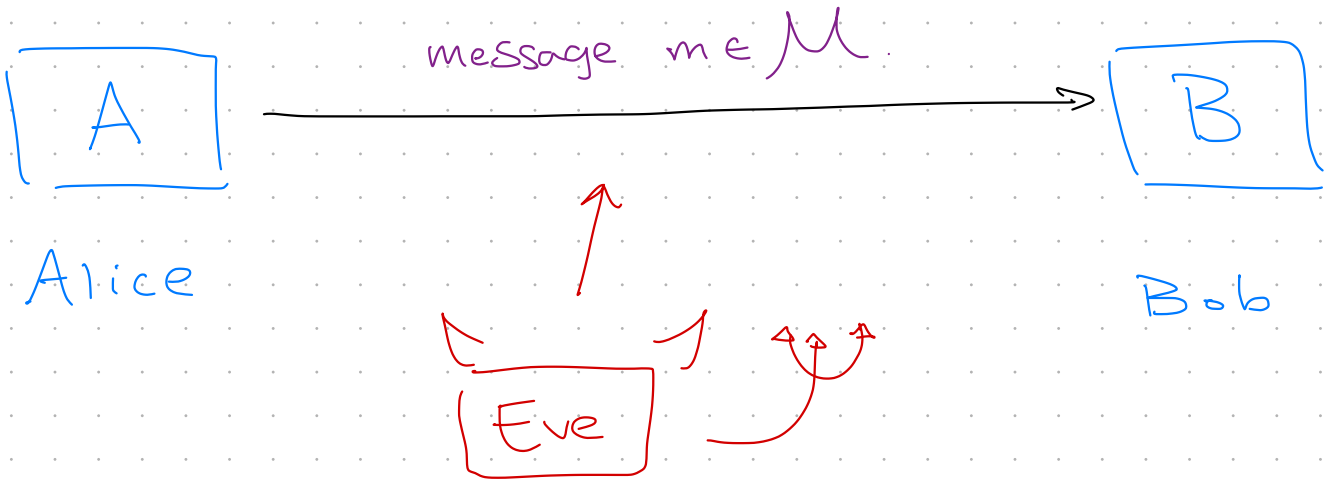
* Alice sends message to Bob



* Goal: maintain secrecy of message

Secure Communication

* Alice sends message to Bob



* Goal: maintain secrecy of message

↳ from whom?

passive Eavesdropper

(postcard model)

Encryption Scheme consists of 3 algorithms

* $k \leftarrow Gen()$ // generates key k
* $c \leftarrow Enc(m; k)$ // encrypts message m w/ key k to ciphertext c
* $m \leftarrow Dec(c; k)$ // decrypts ciphertext c w/ key k to message m

Encryption Scheme consists of 3 algorithms

* $k \leftarrow Gen()$ // generates key k
* $c \leftarrow Enc(m; k)$ // encrypts message m w/ key k to ciphertext c
* $m \leftarrow Dec(c; k)$ // decrypts ciphertext c w/ key k to message m

Functionality

$\forall k \in \mathcal{K}, \forall m \in \mathcal{M}$

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

Encryption Scheme consists of 3 algorithms

* $k \leftarrow Gen()$ // generates key k
* $c \leftarrow Enc(m; k)$ // encrypts message m w/ key k to ciphertext c
* $m \leftarrow Dec(c; k)$ // decrypts ciphertext c w/ key k to message m

Functionality

$\forall k \in \mathcal{K}, \forall m \in \mathcal{M}$

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

What about secrecy?

Announcements

* HW 8 Ongoing
* HW 9 Optional (Required for CS 5820)

* Final: 13 May 2025
  7pm
  Barton 100 WEST
  } No alternate options

Encryption Scheme

* $k \leftarrow Gen()$
* $c \leftarrow Enc(m; k)$
* $m \leftarrow Dec(c; k)$

Functionality: Decryption inverts Encryption

$$\forall\, k \in \mathcal{K},\ \forall\, m \in \mathcal{M}$$

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

Secrecy: Eve learns nothing about message m from ciphertext c

Secrecy: Eve learns nothing about message m from ciphertext c

Secrecy: Eve learns nothing about message m from ciphertext c

Perfect (Shannon) Secrecy

$\forall\ m_0, m_1 \in \mathcal{M}. \quad \forall\ c \in \mathcal{C}$

$$\Pr_{k \leftarrow Gen()}\left[Enc(m_0, k) = c\right] = \Pr_{k \leftarrow Gen()}\left[Enc(m_1, k) = c\right]$$

Secrecy: Eve learns nothing about message m from ciphertext c

Perfect (Shannon) Secrecy

$\forall\ m_0, m_1 \in \mathcal{M}. \quad \forall\ c \in \mathcal{C}$

$$\Pr_{k \leftarrow Gen()}\left[Enc(m_0, k) = c\right] = \Pr_{k \leftarrow Gen()}\left[Enc(m_1, k) = c\right]$$

Given a random key,

the distribution of ciphertexts is identical

for all plaintext messages.

Secrecy: Eve learns nothing about message m from ciphertext c

c does not change Eve's posterior over m

Perfect (Shannon) Secrecy

$\forall\ m_0, m_1 \in \mathcal{M}.\quad \forall\ c \in \mathcal{C}$

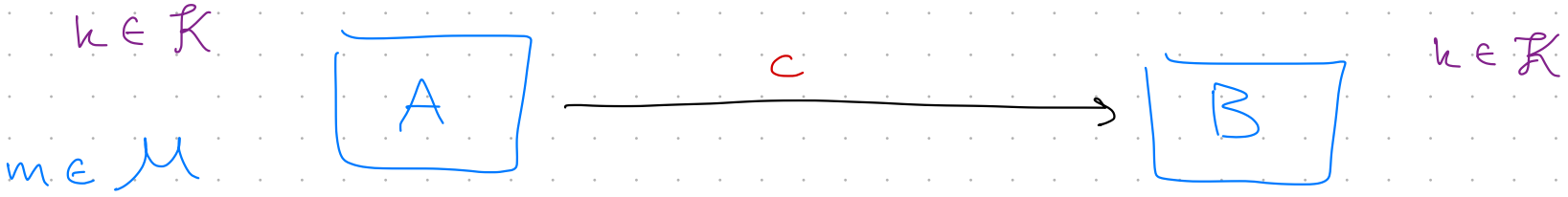
$$\Pr_{k \leftarrow Gen()}\left[Enc(m_0, k) = c\right] = \Pr_{k \leftarrow Gen()}\left[Enc(m_1, k) = c\right]$$

Given a random key,

the distribution of ciphertexts is identical

for all plaintext messages.

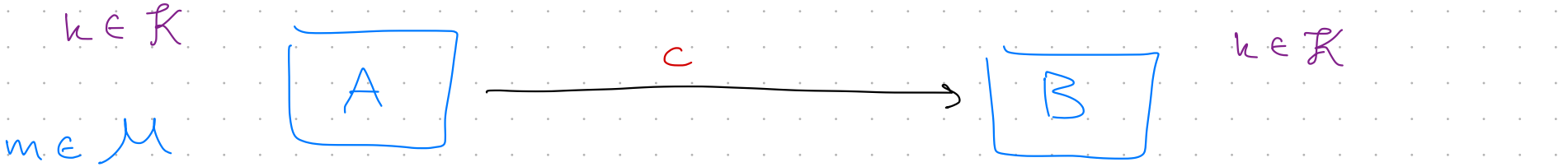
Secure Communication from Perfect Secrecy Enc.



Alice computes

$c = Enc(m; k)$

Secure Communication from Perfect Secrecy Enc.



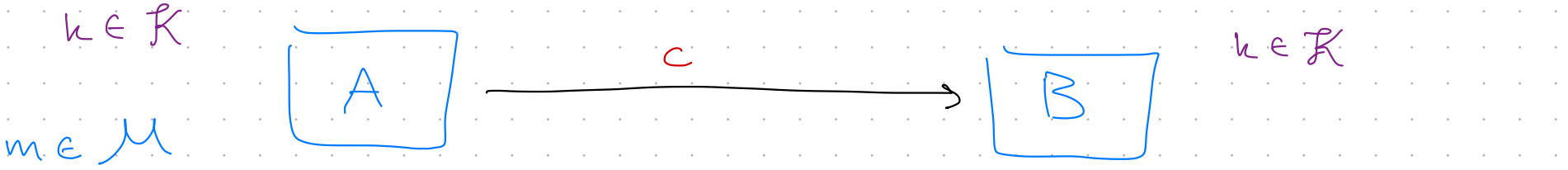
Alice computes

$c = Enc(m; k)$

Bob computes

$m = Dec(c; k)$

Secure Communication from Perfect Secrecy Enc.



Alice computes

$c = Enc(m; k)$

Bob computes

$m = Dec(c; k)$

only sees c

$\Downarrow$ perfect secrecy

learns nothing about m

Perfect Secrecy Encryption: One-Time Pad

$$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$$

OTP $\equiv$ XOR w/ Random Key

Perfect Secrecy Encryption: One-Time Pad

$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$

Gen() = sample $k \leftarrow \mathcal{K}$ uniformly at Random

$Enc(m; k) = m \oplus k$

$Dec(c; k) = c \oplus k$

$m = 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1$

$\oplus$

$k = 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1$

$c = 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0$

Perfect Secrecy Encryption: One-Time Pad

$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$

$\text{Gen}()$ = sample $k \leftarrow \mathcal{K}$ uniformly at Random

$\text{Enc}(m; k) = m \oplus k$

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

Functionality $\quad \text{Dec}(\text{Enc}(m; k); k) = (m \oplus k) \oplus k$

$= m$ ✓

Perfect Secrecy Encryption: One-Time Pad

$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$

$Gen()$ = sample $k \leftarrow \mathcal{K}$ uniformly at Random

$Enc(m; k) = m \oplus k$

$Dec(c; k) = c \oplus k$

Functionality $Dec(Enc(m; k); k) = (m \oplus k) \oplus k$

$= m$ ✓

Secrecy Fix m. Fix c.

$$\Pr_{k \leftarrow \{0,1\}^n}[Enc(m; k) = c]$$

Perfect Secrecy Encryption: One-Time Pad

$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$

$Gen()$ = sample $k \leftarrow \mathcal{K}$ uniformly at Random

$Enc(m; k) = m \oplus k$

$Dec(c; k) = c \oplus k$

Functionality $Dec(Enc(m; k); k) = (m \oplus k) \oplus k$

$= m$ ✓

Secrecy Fix m. Fix c.

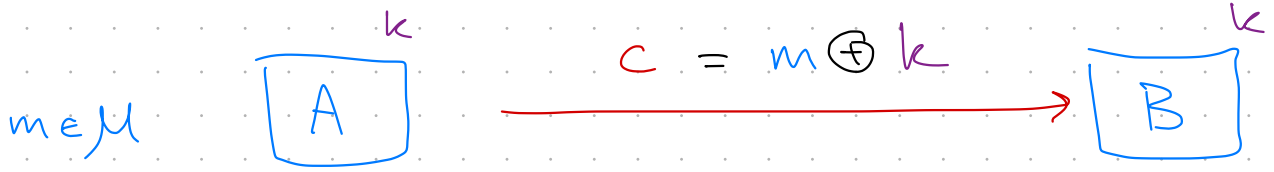
$$\Pr_{k \leftarrow \{0,1\}^n}[Enc(m; k) = c]$$

$$= \Pr_{k \leftarrow \{0,1\}^n}[m \oplus k = c] = 2^{-n}$$ ✓

$\leftarrow$ independent of m, c

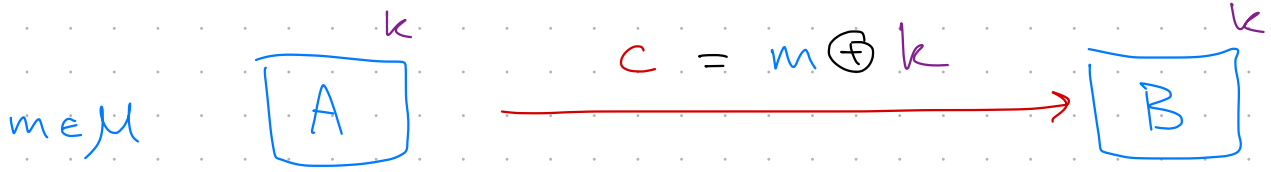
Are we done?

* One-time pad achieves perfect secrecy!



Are we done?

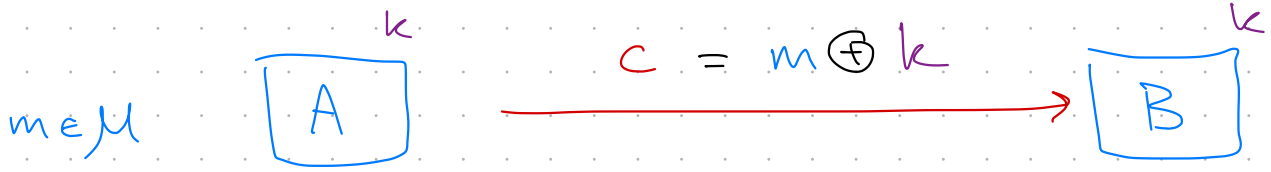
* One-time pad achieves perfect secrecy!



Problem. One-time pad only secures one message!

Are we done?

* One-time pad achieves perfect secrecy!



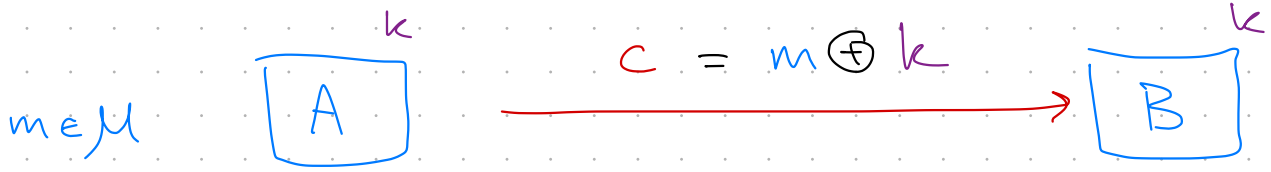
Problem. One-time pad only secures one message!

Reusing One-Time Pad is NOT Secure!

$$\left.\begin{aligned} c_0 &= m_0 \oplus k \\ c_1 &= m_1 \oplus k \end{aligned}\right\} \quad c_0 \oplus c_1 = (m_0 \oplus k) \oplus (m_1 \oplus k)$$

Are we done?

* One-time pad achieves perfect secrecy!



Problem. One-time pad only secures one message!

Reusing One-Time Pad is NOT Secure!

$$\left.\begin{aligned} c_0 &= m_0 \oplus k \\ c_1 &= m_1 \oplus k \end{aligned}\right\} \quad c_0 \oplus c_1 = (m_0 \oplus k) \oplus (m_1 \oplus k)$$

$$= m_0 \oplus m_1$$

Significant leak!

New key for each message?

Key is as long as the message / ciphertext

- Alice & Bob must agree on key

Theorem. Every Perfect Secrecy Encryption Scheme Requires $|K| \geq |M|$.

$\Rightarrow$ prohibitive for practical applications

Enter Complexity Theory

* Perfect Secrecy requires No information leakage

* But some information may be hard to extract.

Enter Complexity Theory

* Perfect Secrecy requires No information leakage

* But some information may be hard to extract.

e.g. Consider some CNF ϕ.

↳ Let my message = lexicographically minimal satisfying assignment to ϕ (and $\perp$ if UNSAT)

ϕ defines m.

↳ But what can we know about m in polynomial time?

Semantic Security (Goldwasser & Micali)

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

Semantic Security (Goldwasser & Micali)

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

Practical Cryptography requires Computational Hardness!