CY 2550 Foundations of Cybersecurity

Cryptography Part 2

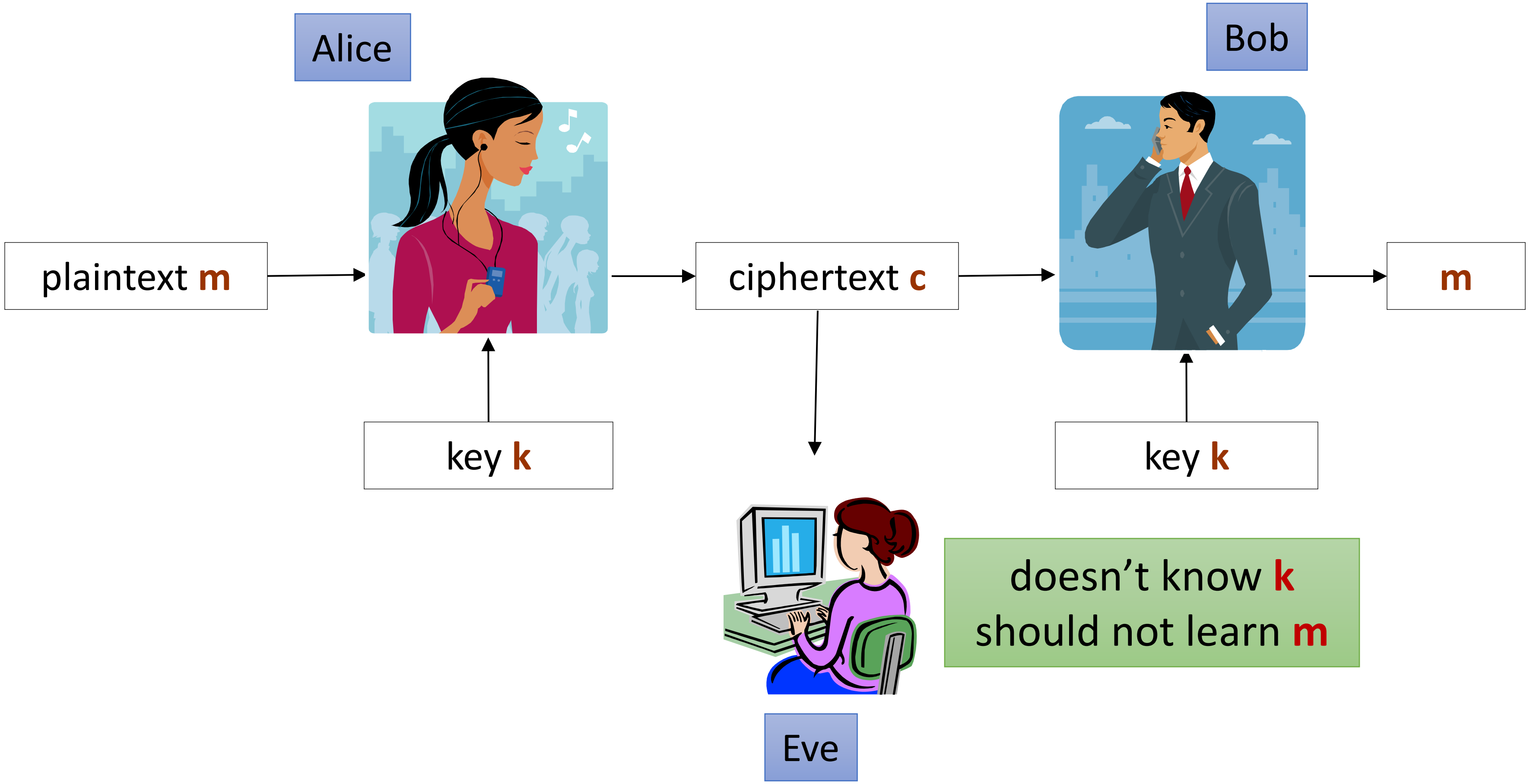
January 23

Alina Oprea

Associate Professor, Khoury College

Northeastern University

Encryption Terminology



Encryption scheme = encryption & decryption procedures

One Time Pad (1920s)

- Fix the vulnerability of the Vigenère cipher by using very long keys
- Key is a random string that is at least as long as the plaintext
- Similar encryption as with Vigenère (different shift per letter)

One-time pad

ℓ – a parameter

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

component-wise **xor**

Vernam's cipher:

$$\mathrm{Enc}_k(m) = k \oplus m$$
$$\mathrm{Dec}_k(c) = k \oplus c$$

Gilbert Vernam

(1890 –1960)

Correctness:

$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) \;=\; k \oplus (k \oplus m)$$
$$m$$

"The adversary should not learn any information about **m**."

An encryption scheme is **perfectly secret** if

for every distribution of **M**

and every **m** $\in \mathcal{M}$ and **c** $\in \mathcal{C}$

$$\Pr[\, M = m \,] = \Pr[\, M = m \mid C = c \,]$$

Ciphertext-only attack
(passive)

In English

- The adversary believes the probability that the plaintext is *m* is *Pr(M=m)* **before seeing the ciphertext**
  - Maybe they are very sure, or maybe they have no idea
- The adversary believes the probability that the plaintext is *m* is *Pr(M=m | C=c)* **after seeing that the ciphertext is *c***
- *Pr(M=m | C=c) = P(M= m)* means that after knowing that the ciphertext is *c*, the adversary's belief does not change
  - Intuitively, the adversary learned **nothing** from the ciphertext

Put Another Way

- Imagine you have a ciphertext *c* where the length $|c| = 1000$
- I can give you a key k_i with $|k_i| = 1000$ such that:
  - The decrypted message m_i is the first 1000 characters of Hamlet
- Or, I can give you a key k_j with $|k_j| = 1000$ such that:
  - The decrypted message m_j is the first 1000 characters of the US Constitution

- If an algorithm offers perfect secrecy then:
  - For a given ciphertext of length *n*
  - All possible corresponding plaintexts of length *n* are possible decryptions

Is Shift Cipher Perfectly Secure?

An encryption scheme is **perfectly secret** if

for every distribution of **M**

and every **m** $\in \mathcal{M}$ and **c** $\in \mathcal{C}$

$$\Pr[\, M = m \,] = \Pr[\, M = m \mid C = c \,]$$

- Perfectly secure for 1 letter message:
  - $\Pr[M= m] = 1/26$
  - $\Pr[M= M|C=c]= \Pr[K=c-m \bmod 26] = 1/26$

- Counterexample (2-letter message):
  - M_1=AB; M_2=AZ; c=BC
  - $\Pr[M= M_1|C=c]= \Pr[k=1] = 1/26$
  - $\Pr[M= M_2|C=c]= 0$

Cryptanalysis of OTP

- Intuitively, the key is random, so ciphertext is also random (because of properties of XOR)
- OTP achieves Perfect Secrecy
  - Shannon or Information Theoretic Security
  - Basic idea: ciphertext reveals no “information” about plaintext
- Caveats
  - If the length of the OTP key is less than the length of the message…
    - It’s not a OTP anymore, not perfectly secret!
  - If you reuse the OTP key…
    - It’s not a OTP anymore, not perfectly secret!
- Major issue with OTP in practice?
  - How to securely distribute the key books to both parties

Why the one-time pad is not practical?

1. **The key is as long as the message.**
2. **The key cannot be reused.**
3. **Alice and Bob must share a new key every time they communicate**

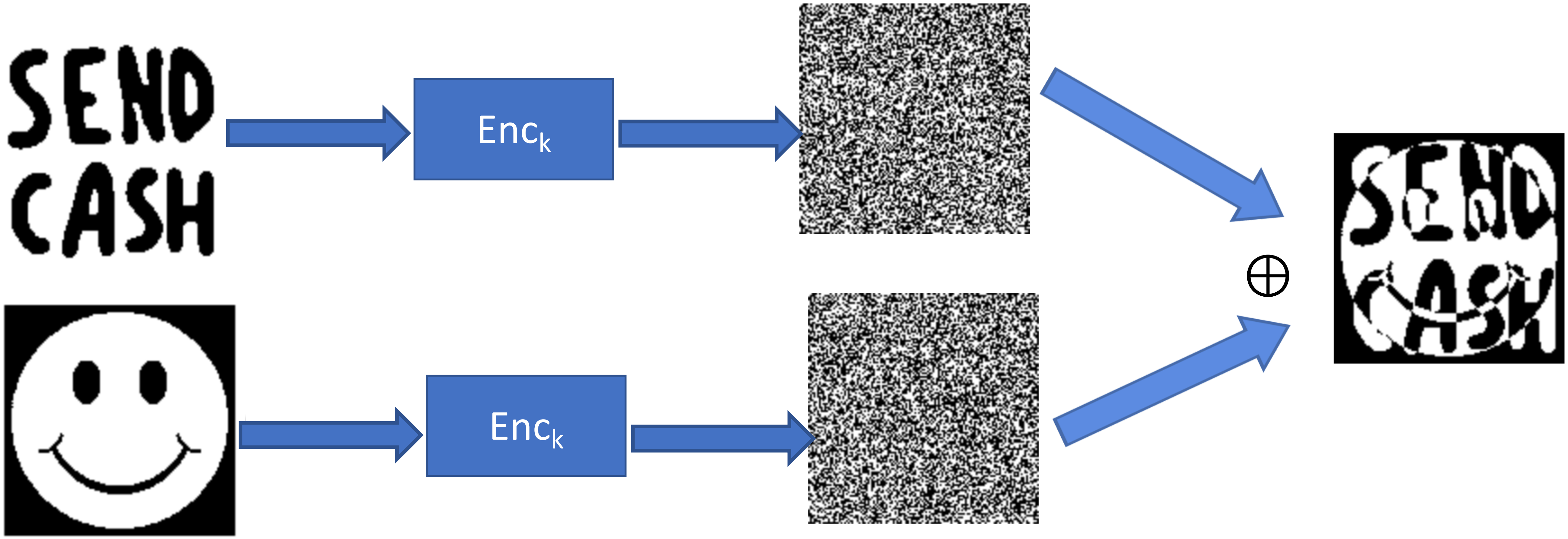
All three are necessary for perfect secrecy!

This is because:

$$\mathrm{Enc}_k(m_1) \text{ xor } \mathrm{Enc}_k(m_2) = (k \text{ xor } m_1) \text{ xor } (k \text{ xor } m_2)$$

$$= m_1 \text{ xor } m_2$$

Example: key reuse



Venona project (1946 – 1980)

Ethel and Julius Rosenberg

American **National Security Agency** decrypted **Soviet** messages that were transmitted in the 1940s.

That was possible because the Soviets reused the keys in the one-time pad scheme.

Key takeaways

- Historical methods for encryption are not secure
  - Shift cipher, mono-alphabetic substitution cipher, Vigenere
  - Attacks: Brute force (small key space), frequency analysis
- Defining security for encryption is difficult
  - Perfect secrecy is one of the first rigorous notion of security
- One-time pad is perfectly secure
  - But many practical drawbacks
  - Still has been used in critical military applications
- Modern cryptography relies on computational assumptions to become practical
  - E.g., it is computationally hard to factor large numbers; adversary has limited computational resources

Computational Security

“Real” cryptography starts here!

Restriction: **Eve is computationally-bounded**

We will construct schemes that in **principle can be broken** if the adversary has a huge computing power or is extremely lucky.

- E.g., break the scheme by enumerating all possible secret keys. (“**brute force attack**”)
- E.g., break the scheme by guessing the secret key.

Goal: cannot be broken with reasonable computing power with reasonable probability.

Towards Computational Security

- Perfect secrecy is too difficult to achieve in practice
  - Imagine trying to do OTP encryption with every website that uses HTTPS
- Computational security uses two relaxations:
  1. Security is preserved only against computationally bounded adversaries
     - Limits on computational power and storage
     - Polynomial-time adversaries
  2. Adversaries may successfully crack encryption with a very small probability
     - So small that (we hope) it becomes negligible
     - Example negligible probability: $\frac{1}{2^{128}}$
- Computational assumptions are part of the threat model

Eavesdropping security

- Ciphertext INDistinguishability under an EAVesdropping attacker (IND-EAV)

Charlie (Challenger)

Adv

Round 1: Charlie chooses k and encryption algo k, Enc_k

Round 2: Adv chooses two plaintext messages $m_0, m_1 \in \boldsymbol{M}$

Round 3: Charlie chooses a random binary number $b \leftarrow_R \{0, 1\}$

Round 4: Charlie encrypts the corresponding message $c = Enc_k(m_b)$

Round 5: Adv guesses a the value of *b* $b' \in \{0, 1\}$

Adversary wins if $b = b'$

Examples

- If E is a perfectly secure algorithm (e.g., OTP), what is the probability that $b = b'$?

$P(Adv\ wins) = ½$ SECURE

- If E is a Caesar shift, what is the probability that $b = b'$**?**

$P(Adv\ wins) = 1$ NOT SECURE

Charlie (Challenger)

k, Enc_k

Adv

$m_0, m_1 \in \boldsymbol{M}$

$b \leftarrow_R \{0, 1\}$

$c = Enc_k(m_b)$

$b' \in \{0, 1\}$

Adversary wins if $b = b'$

Computational secure IND-EAV

- If *Enc* is computationally secure algorithm, what is the probability that $b = b'$?

$P(Adv\ wins) = ½ + negligible(|k|)$

How to achieve this?

Charlie (Challenger)

$k,\ E_k$

Adv

$m_0, m_1 \in \boldsymbol{M}$

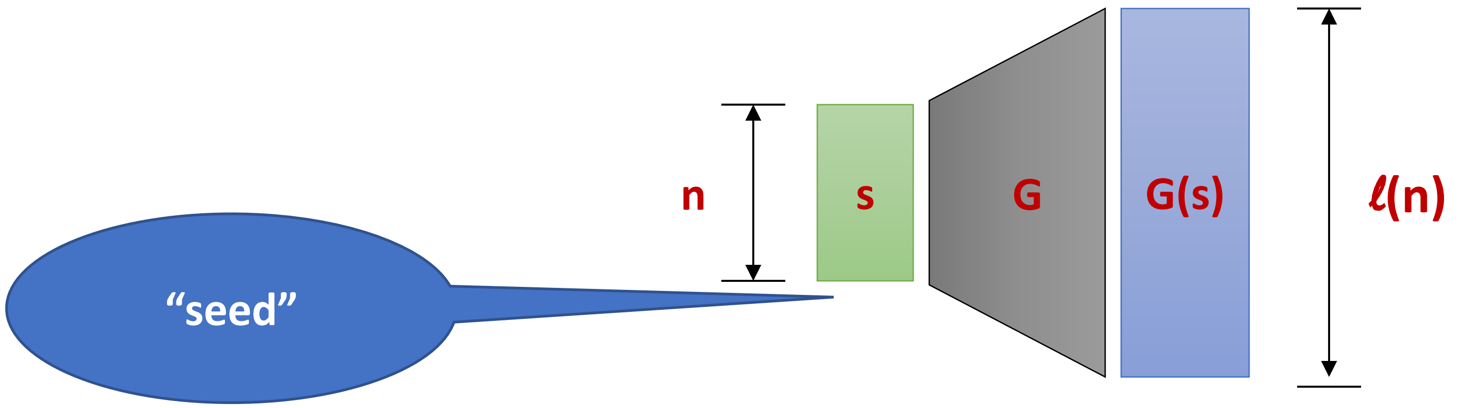
$b \leftarrow_R \{0, 1\}$

$c = Enc_k(m_b)$

$b' \in \{0, 1\}$

Adversary wins if $b = b'$

Pseudorandom generators (PRG)



A pseudorandom generator is a deterministic algorithm
$G : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$.

- **Output length:** $\ell(n)$ for all s with $|s| = n$ we have $|G(s)| = \ell(n)$.
- **Stretch:** $\ell(n) - n$

Goal (imprecise): If s chosen randomly from $\{0,1\}^n$,
then $G(s)$ "looks" like it was chosen randomly from $\{0,1\}^{\ell(n)}$.

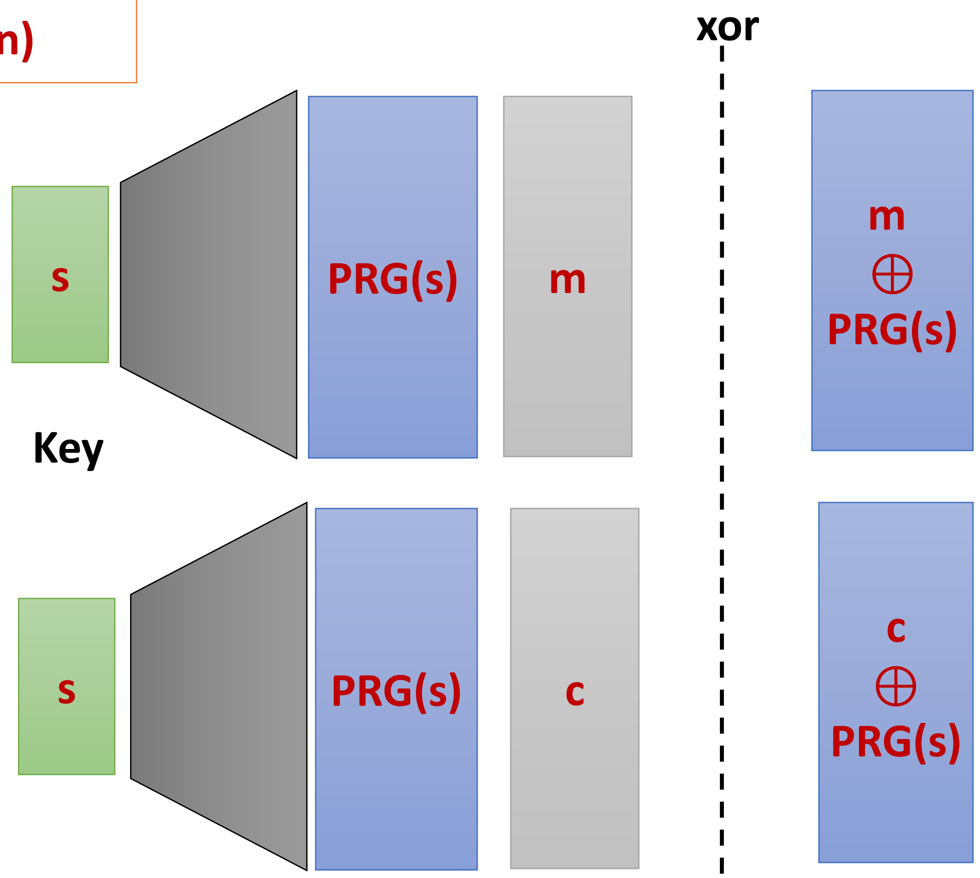
Using a PRG to build efficient OTP

Use PRGs to “shorten” the key in the one time pad

Key: random string of length **n**
Plaintexts: strings of length **ℓ(n)**

Enc(s,m)

Dec(s,m)



STREAM CIPHER
Examples:
RC4, Salsa20

IND-EAV secure one-time pad

Adversarial capability

- Ciphertext-only attack: Perfect security, IND-EAV
  - Adversary observes ciphertext(s)
  - Infer information about plaintext
- Chosen-plaintext attack: IND-CPA
  - Adversary can encrypt messages of his choice
- Chosen-ciphertext attack: IND-CCA
  - Adversary can decrypt ciphertexts of its choice
  - Learn plaintext information on other ciphertext

IND-CPA security

- Ciphertext Indistinguishability under a Chosen-Plaintext Attack (CPA)

Charlie (Challenger)

Adv

Query: Encrypt m

Reply: Ciphertext c

Round 1: Charlie chooses k and encryption algo k, Enc_k

Round 2: Adv chooses two plaintext messages $m_0, m_1 \in \boldsymbol{M}$

Round 3: Charlie chooses a random binary number $b \leftarrow_R \{0, 1\}$

Round 4: Charlie encrypts the corresponding message $c = Enc_k(m_b)$

Query: Encrypt m

Reply: Ciphertext c

Round 5: Adv guesses a the value of *b* $b' \in \{0, 1\}$

Adversary wins if $b = b'$