# Cryptography I

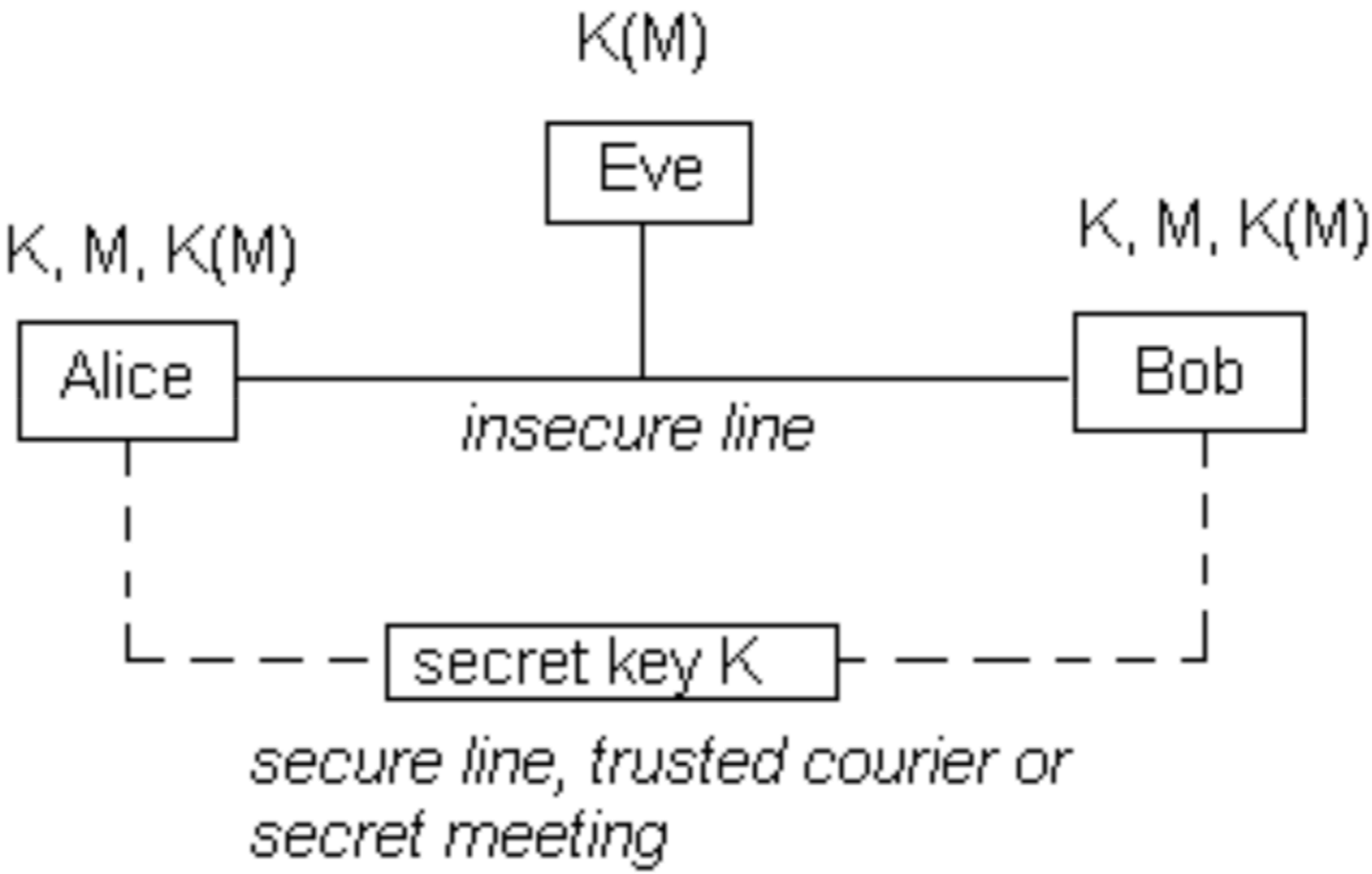
Instructor: Paruj Ratanaworabhan

Credits:
D. Wagner, UC Berkeley
M. Stamp, San Jose State U.

# Definition

- Cryptography is about secure communication over insecure media
- Security properties enabled by cryptography
  - Confidentiality - preventing eavesdroppers from accessing content intended for communicating parties
  - Integrity - anyone without permission cannot modify the message content
  - Authenticity - content originated from verifiable sources

# Traditional (symmetric-key) cryptosystem



# Kerckhoff's Principle

- *Crypto systems should remain secure even when the attacker knows all internal details of the system*
- *The key should be the only thing that must be kept secret, and the system should be designed to make it easy to change keys that are leaked*
- *If your secrets are leaked, it is usually a lot easier to change the key than to replace every instance of the running software*

# Kerckhoff's principle:
# no security by obscurity

# Symmetric-key encryption: from shift cipher to Vigenere cipher

# Shift Cipher (Caesar Cipher)

- Key space:
  - [0 .. 25]
- Encryption with key K:
  - Every character in a plain text P is substituted with the character K positions to the right of this character
- Decryption with key K:
  - Replace a character in the cipher text with character K positions to its left

# Shift Cipher Analysis

- How can an attacker discover the key K?
  - Brute force search; only 26 keys for English letters
  - Key space is very small in shift cipher
- Lesson learned
  - Key space must be large to prevent brute force search

# Mono-Alphabetic Substitution Cipher

- Key space: permutation of $\Sigma = \{A,B,C,\ldots,Z\}$
- Encryption with key $\pi$:
  - The letter X in a plain text P is substituted with $\pi(X)$
- Decryption with key $\pi$:
  - The letter Y in a cipher text P is substituted with $\pi^{-1}(Y)$

**Example:**

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\pi$= | B | A | D | C | Z | H | W | Y | G | O | Q | X | S | V | T | R | N | M | L | K | J | I | P | F | E | U |

BECAUSE $\rightarrow$ AZDBJSZ

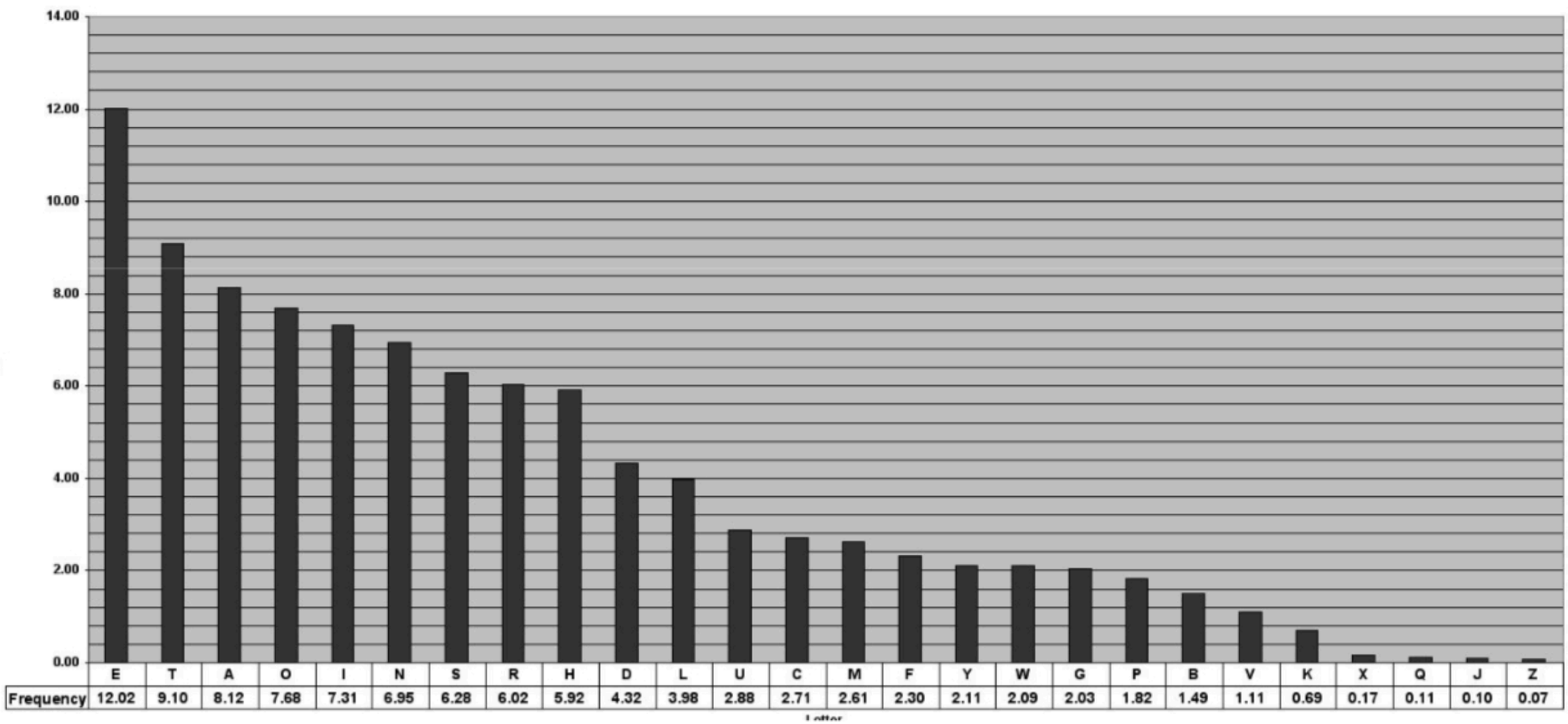
# Analysis of Mono-Alphabetic Substitution Cipher

- Extremely hard to brute force search for the key
  - Size of key space is $26! \approx 4 \times 10^{26} \approx 2^{88}$
- Nevertheless, this scheme is vulnerable to an attack that is relatively easy to mount on

# Frequency Analysis

- Every language has some unique structures and features; some letters are used more than others

- Substitution cipher does not change these structures and features, hence, it is vulnerable to a frequency analysis attack




| | E | T | A | O | I | N | S | R | H | D | L | U | C | M | F | Y | W | G | P | B | V | K | X | Q | J | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Frequency | 12.02 | 9.10 | 8.12 | 7.68 | 7.31 | 6.95 | 6.28 | 6.02 | 5.92 | 4.32 | 3.98 | 2.88 | 2.71 | 2.61 | 2.30 | 2.11 | 2.09 | 2.03 | 1.82 | 1.49 | 1.11 | 0.69 | 0.17 | 0.11 | 0.10 | 0.07 |

**Frequency of each English letter**

# Bigrams, Trigrams

http://mathcenter.oxford.emory.edu/site/math125/englishLetterFreqs

**Most common bigrams (in order)**
th, he, in, en, nt, re, er, an, ti, es, on, at, se, nd, or, ar, al, te, co, de, to, ra, et, ed, it, sa, em, ro.

**Most common trigrams (in order)**
the, and, tha, ent, ing, ion, tio, for, nde, has, nce, edt, tis, oft, sth, men

# Safety of Substitution Cipher

- Can be attacked using frequency analysis
- Lesson learned: increasing key space does not strengthen a cryptographic scheme

# Vigenere Cipher

- Improvement over vanilla substitution ciphers to make it more difficult to perform frequency analysis
  - Widely used for over 300 years; no one seems to have broken it
  - Shift cipher where the shift value, K, changes with the key
- Encryption and decryption follow the equations below:

$$C_i = E_K(M_i) = (M_i + K_{(i \bmod m)}) \bmod \ell,$$
$$M_i = D_K(C_i) = (C_i - K_{(i \bmod m)}) \bmod \ell.$$

# Encryption with Vigenere Cipher

- Plaintext:

  ATTACKATDAWN

- Key:

  LEMON

- Keystream:

  LEMONLEMONLE

- Ciphertext:

  LXFOPVEFRNHR

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| I | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| K | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| L | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| M | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| N | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| O | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| P | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| Q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| R | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| T | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| U | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| V | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| W | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| X | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| Z | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

# Vigenere Cipher Safety

- Much more difficult to perform frequency analysis
- Nevertheless, sophisticated frequency analysis is possible; see the method used by Friedrich Kasiski in 1863 from Wikipedia

  (en.wikipedia.org/wiki/Vigen%C3%A8re_cipher#Cryptanalysis)
- The main idea of the Kasiski method is this: the same plain text that occurs multiple times and separated by the distance of the key length times a constant yields the same cipher text

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Key | K | I | N | G | K | I | N | G | K | I | N | G | K | I | N | G | K | I | N | G | K | I | N | G |
| Plain | t | h | e | s | u | n | a | n | d | t | h | e | m | a | n | i | n | t | h | e | m | o | o | n |
| Cipher | D | P | R | Y | E | V | N | T | N | B | U | K | W | I | A | O | X | B | U | K | W | W | B | T |

# One-Time Pad

- Is it possible to design a cryptographic scheme that is completely immune to frequency analysis
- Yes, that scheme is called one-time pad and the key idea is: **make the key length in Vigenere cipher equal to the size of the plain text message to be encrypted**

# Bit-Level Version for One-Time Pad

- Plain text space = cipher text space = key space = $\{0,1\}^n$
- Key must be chosen randomly for every new encryption
- cipher text = plain text XOR key

  Plain text: 1 1 0 1 1 0 1 1

  Key: 0 1 1 0 1 0 0 1

  Cipher text: 1 0 1 1 0 0 1 0

# Safety of One-Time Pad

- If key is never reused and is chosen truly randomly, perfect secrecy can be achieved
  - Perfect secrecy = the output cipher text reveals no information about input the plain text
    - Probability of guessing for the plain text does not improve with the presence of the output cipher text
  - Quantum computers do not do better that classical computers when it comes to one-time pad encryption

# Key Reuse in One-Time Pad

- XORing of two cipher texts can reveal information about the corresponding plain texts
- Let C, C’ and M, M’ be a pair of cipher texts and  plain texts, respectively
- If C = M xor K and C’= M’ xor K
- Then, C xor C’= M xor M’
- Use a technique known as crib dragging to retrieve  M or M’
  - www.cribdrag.com/about.html

# Steganography

- Covered writing instead of secret writing
- Hiding secret in plain sight
- Secret hiding using steganography attracts less attention than writing secret with cryptography; the latter invites people to take a crack at it

# Steganography VS Cryptography

- In practice, steganography is not widely used because it offers lower security and lesser efficiency than does cryptography
- If an attacker learns the technique of secret hiding in steganography, no more secret can be hidden with that technique
  - Steganography relies on security by obscurity which is against the Kerckhoff we discussed earlier on
- Steganography incurs lots of overhead, for example, in hiding messages inside an image, the number of bits used for encoding the messages is usually much lower than that for representing the image
  - Each 8-bit pixel may reserve 1 bit for message encoding; so we send out 8-bit of data to get only a single bit of information

# Present-Day Symmetric-Key Cryptography

- Two main categories:
  - Block cipher
  - Stream cipher

# Block Cipher

- Dividing the input data to encrypt/decrypt into blocks and each block contains 64 or 128 bits of data
- Before encrypting/decrypting, must first produce a block by collecting all the required data to meet the block size requirement
- 3 modes to consider:
  - Electronic Codebook (ECB)
  - Cipher Block Chaining (CBC)
  - Counter Mode (CTR)

# ECB

- Dividing the data into blocks and encrypt the individual block

**Encrypt**

$C_0 = E(P_0, K)$,

$C_1 = E(P_1, K)$,

$C_2 = E(P_2, K), \ldots$

**Input**
Plaintext: $P_0$ $P_1$ … $P_m$ …
Key: K

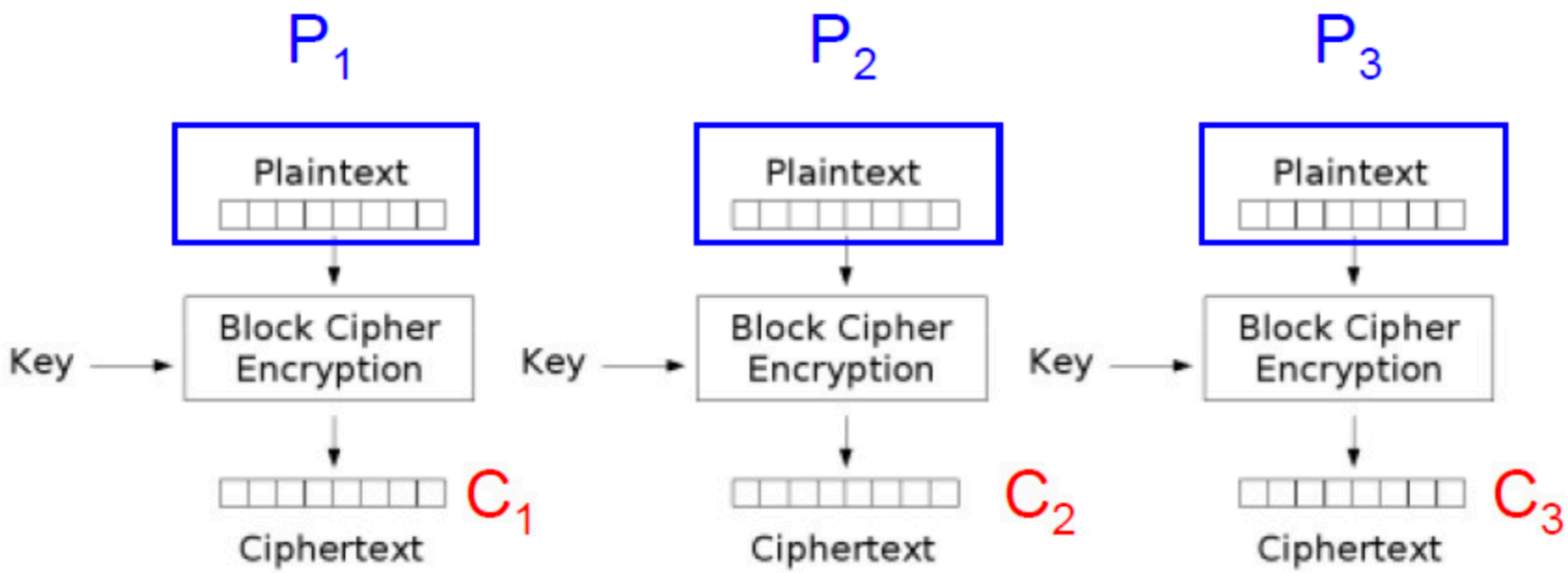
**Decrypt**

$P_0 = D(C_0, K)$,

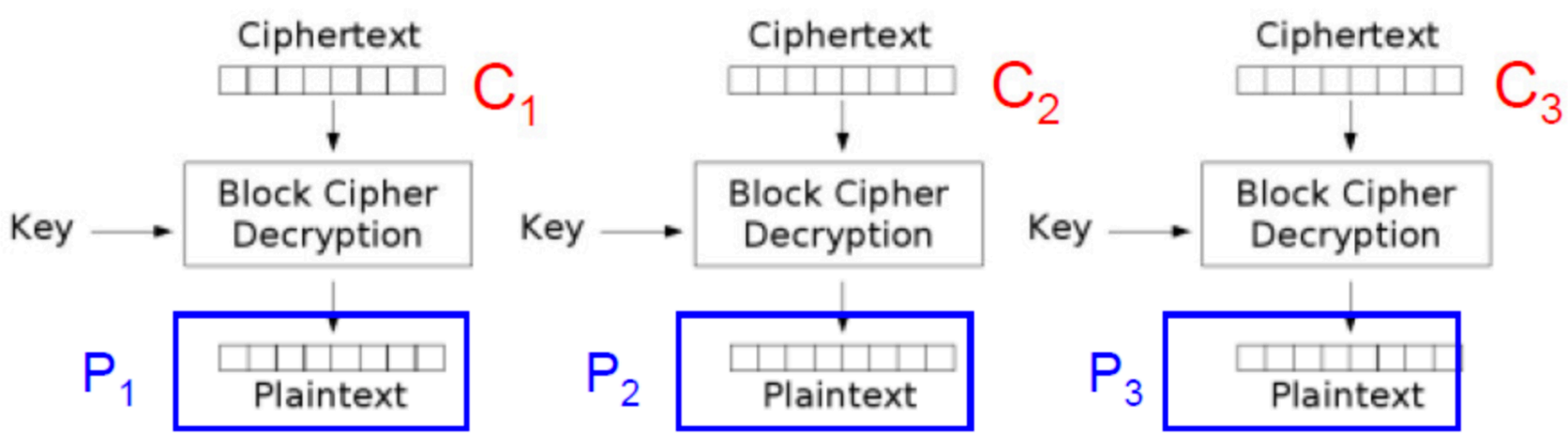
$P_1 = D(C_1, K)$,

$P_2 = D(C_2, K), \ldots$

**Input**
Ciphertext: $C_0$ $C_1$ … $C_m$ …
Key: K




Electronic Codebook (ECB) mode encryption

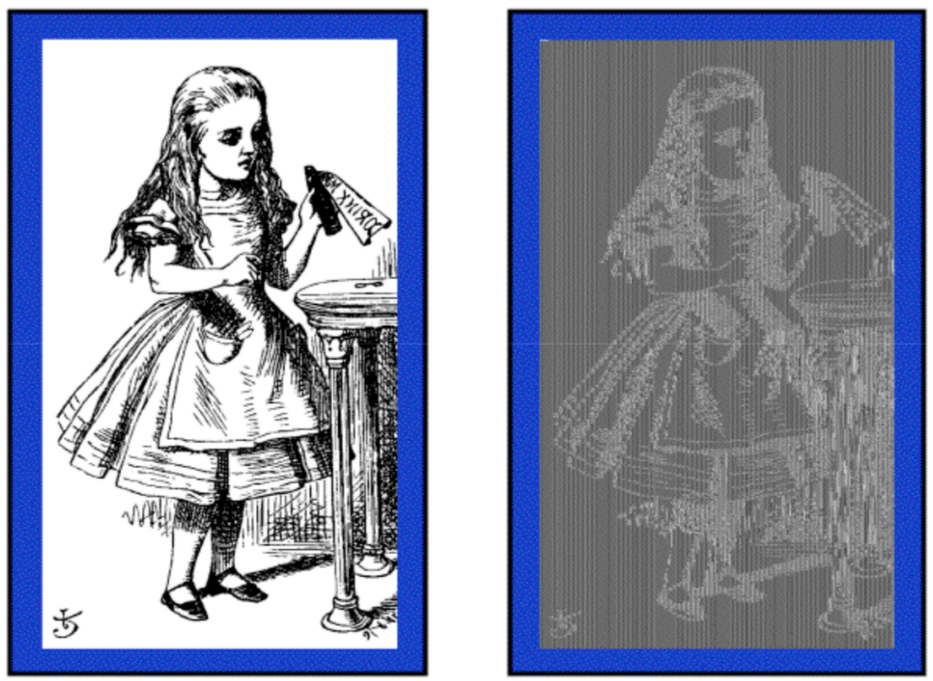


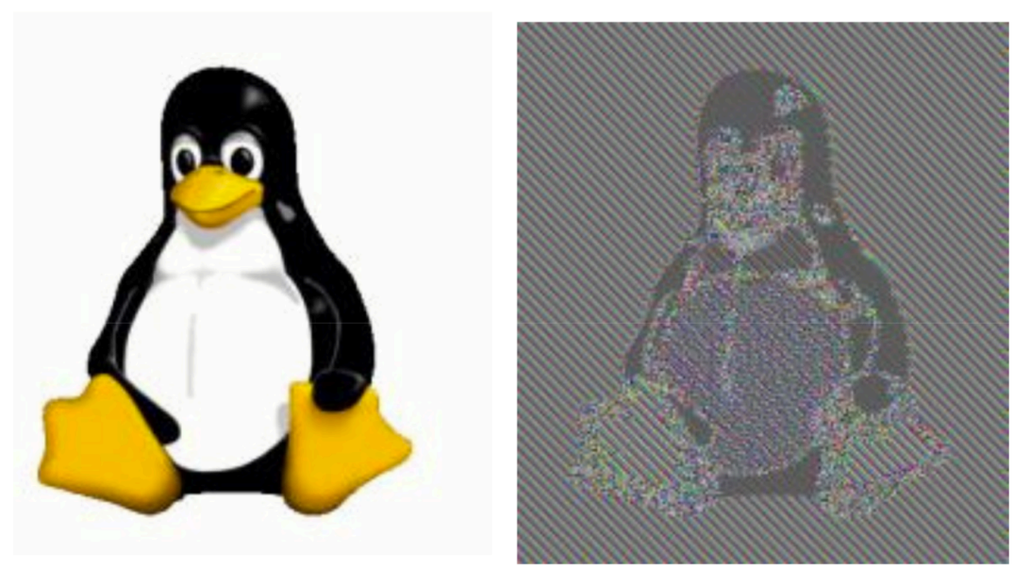

Electronic Codebook (ECB) mode decryption

# Leakage of Input Data to the Cipher Text

- Same plain text in different blocks outputs the same cipher text

Alice's uncompressed image, Alice ECB encrypted





# CBC

- Chain each block together with a non-secret starting random initialization vector (IV) injected into the first block

**Encryption**

$C_0 = E(IV \oplus P_0, K)$,

$C_1 = E(C_0 \oplus P_1, K)$,

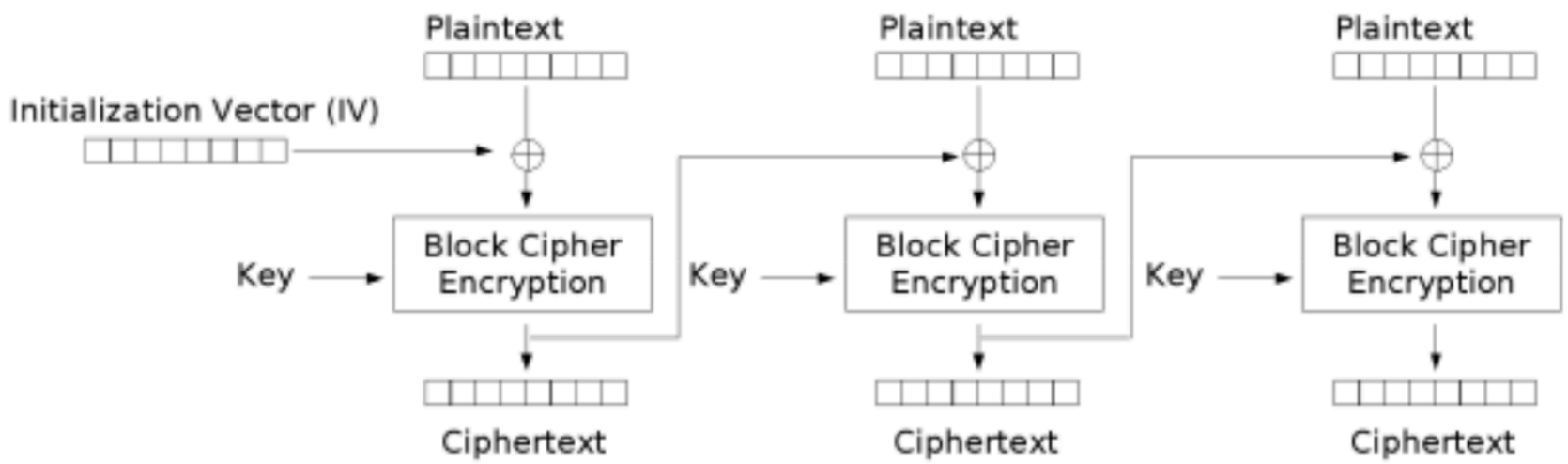
$C_2 = E(C_1 \oplus P_2, K)$,…

**Decryption**

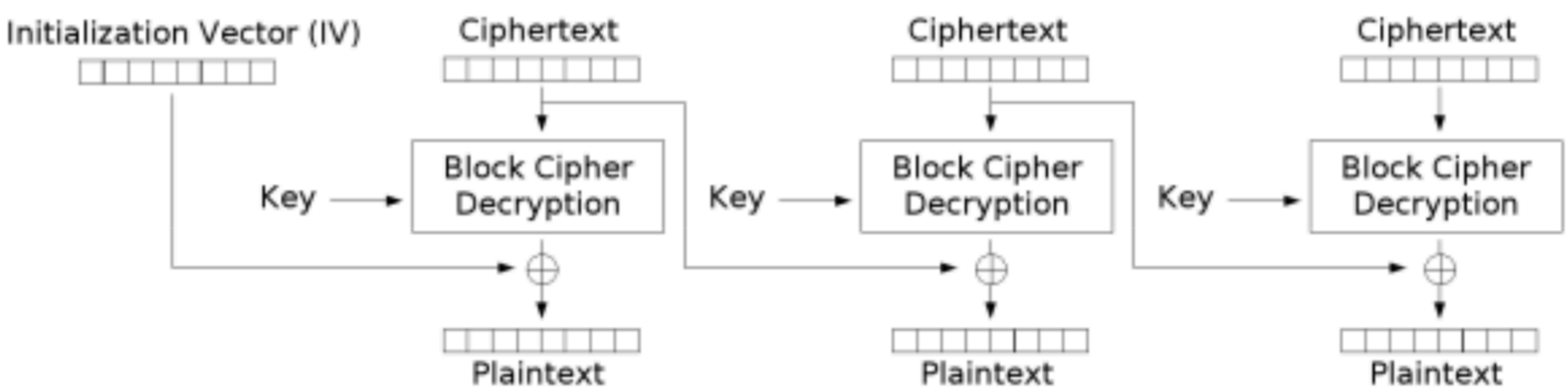
$P_0 = IV \oplus D(C_0, K)$,

$P_1 = C_0 \oplus D(C_1, K)$,

$P_2 = C_1 \oplus D(C_2, K)$,…



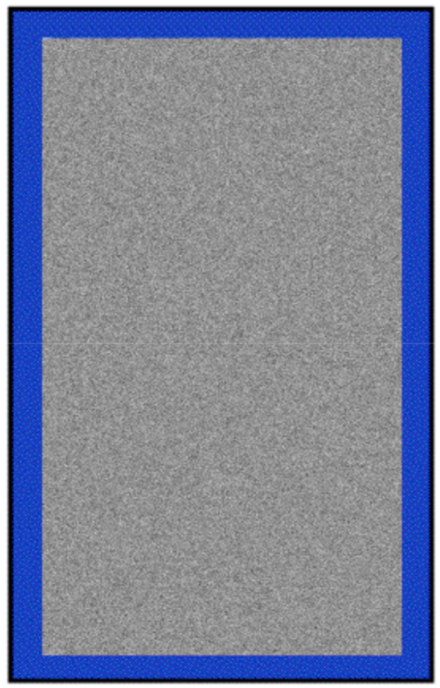

Cipher Block Chaining (CBC) mode encryption

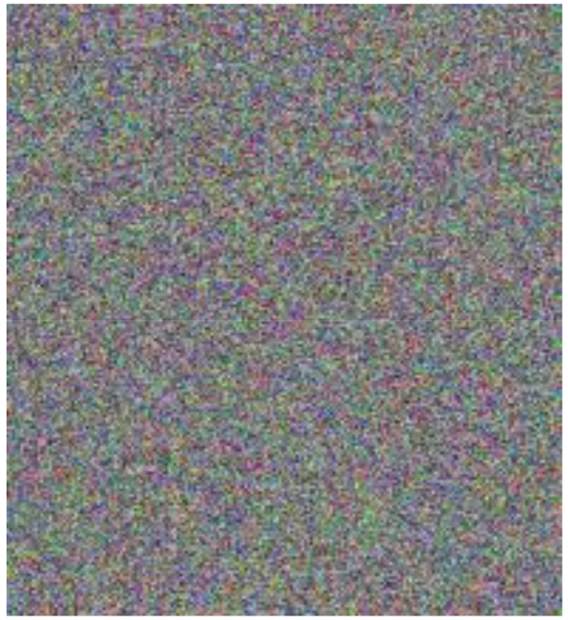



Cipher Block Chaining (CBC) mode decryption

# Negligible Information Leakage to the Cipher Text

Alice's uncompressed image, Alice CBC encrypted





# CTR

- Improving the CBC performance with parallel processing

**Encryption**

$C_0 = P_0 \oplus E(IV, K)$,

$C_1 = P_1 \oplus E(IV+1, K)$,

$C_2 = P_2 \oplus E(IV+2, K)$,…

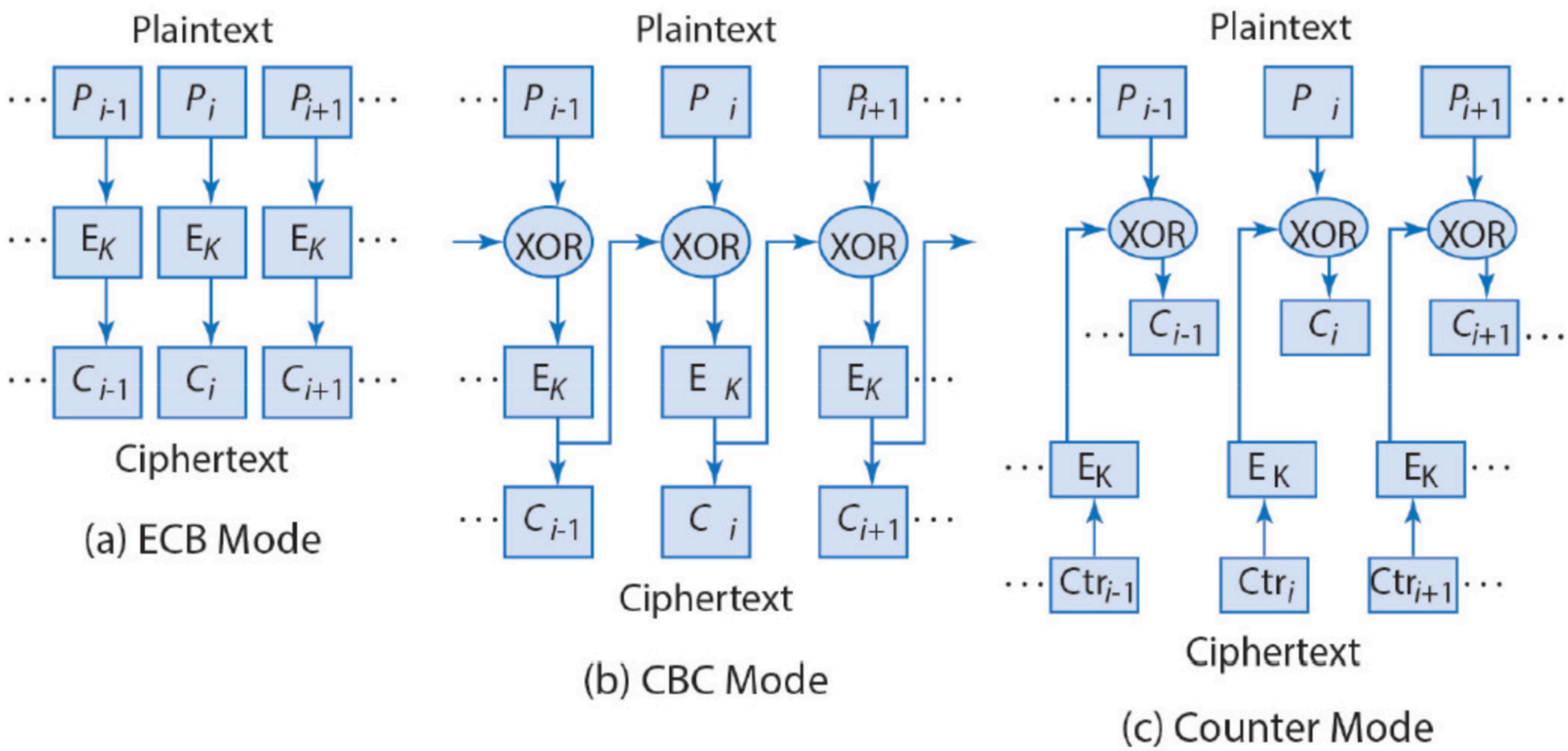
**Decryption**

$P_0 = C_0 \oplus E(IV, K)$,

$P_1 = C_1 \oplus E(IV+1, K)$,

$P_2 = C_2 \oplus E(IV+2, K)$,…

# Comparing the 3 Modes of Block Cipher



# Present-Day Block Cipher

- DES, TripleDES, and AES
- All of these are open-source algorithms
  - Key is the only thing that is kept secret, following Kerckhoff's principle to let there be no security by obscurity
- Not shooting for one-time pad style perfect security, but more like computation security (brute force key search will take forever and cipher text analysis not as straightforward as, say, frequency analysis)
- Use substitution–permutation network or Feistel network that contain these primitives operations, shift, xor, permute, rotate, substitute, transpose, etc
  - https://en.wikipedia.org/wiki/Substitution%E2%80%93permutation_network
  - https://en.wikipedia.org/wiki/Feistel_cipher

# Stream Cipher

- Encrypt each byte of plain text in real time as it arrives
  - Not each block of 4, 16, or 32 bytes
- Much more efficient that block cipher and geared towards real-time communication
  - SNOW is a stream cipher used in 5G networks (https://en.wikipedia.org/wiki/SNOW)
- Early day cryptographic schemes like shift, substitution, or Vigenere cipher, are actually stream cipher

# Stream Cipher

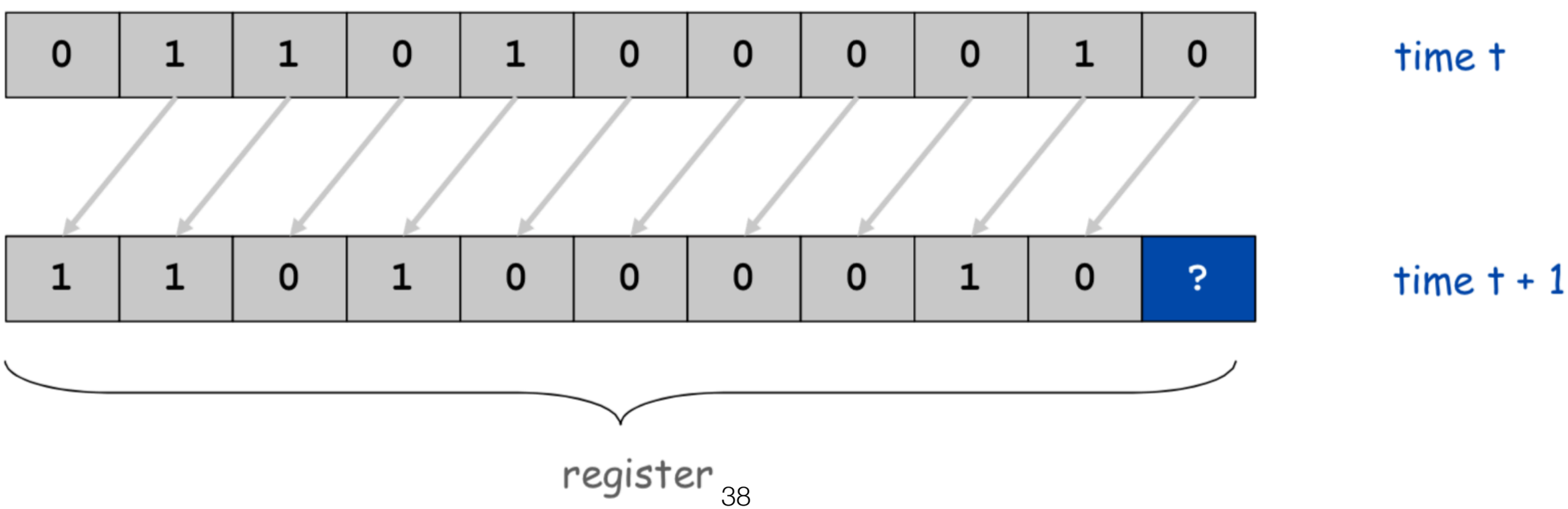
- Present-day stream cipher tries to mimic the one-time pad scheme
  - Use infinite key sequence of infinite random bits to XOR with the input plain text
  - No key reuse
- Rely on a process of generating sequence of infinite random bits to use as the encryption key
- Is there such a process?

# A generator for approximate infinite random bit sequence: LFSR

# Shift Register

Shift register terminology.

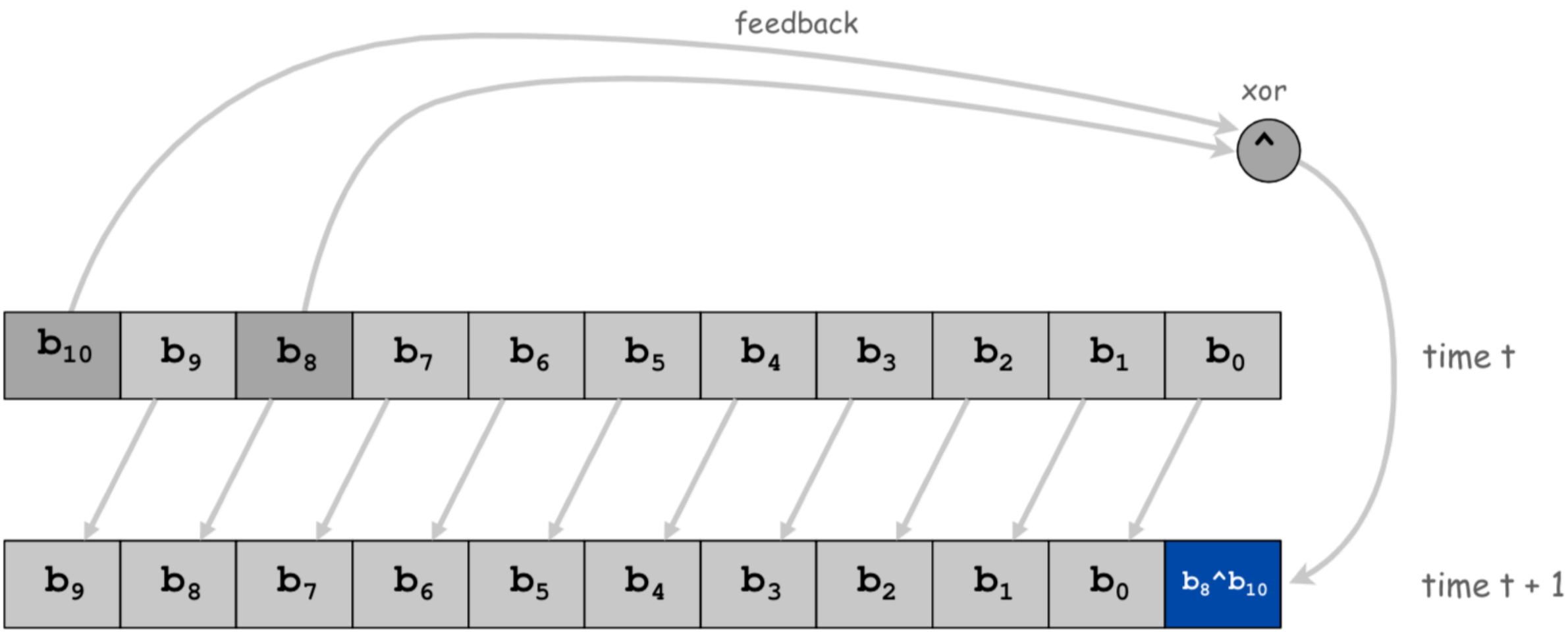
- Bit: 0 or 1.
- Cell: storage element that holds one bit.
- Register: sequence of cells.
- Seed: initial sequence of bits.
  - Shift register: when clock ticks, bits propagate one position to left.



# Linear Feedback Shift Register (LFSR)

{8, 10} linear feedback shift register.

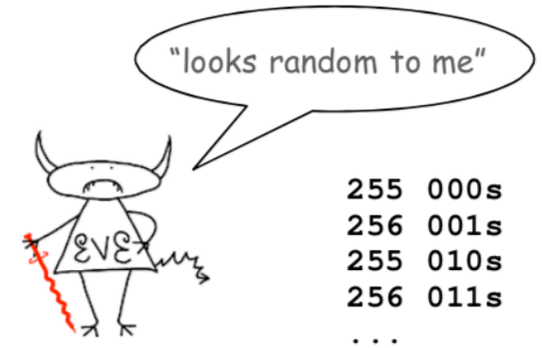
- Shift register with 11 cells.
  - Bit $b_0$ is is XOR of previous bits $b_8$ and $b_{10}$.
  - Pseudo-random bit = $b_0$.



# Linear Feedback Shift Register Demo

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | Time 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | Time 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Time 2 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | Time 3 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | Time 4 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | Time 5 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | Time 6 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | Time 7 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | Time 8 |

# Random Numbers




Q. Are these 2000 numbers random?

If not, what is the pattern?

```
1100100100111101101110010110101110011000101111110100100001001101001011110011001001111111011100
0001010110001000011101010011010000111100100110011101111111010100000100001000101001010100011000
0010111100010010011010110111100011010011011100111101011110010001001110101011101000001010010001
0001101010101110000000101100000100111000101110110100101011001100001111111001100000111111000110
0001101111001110100111101001110010011101110111010101010100000000001000000001010000001000100001
0101010010000000110100000111001000110111010111010100010100001010001001000101011010100001100001
0011110010111001110010111101110010010101110110000101011100100001011101001001010011011000111101
1101100101010111100000010011000010111110010010001110110101101011000110001110111110110101001011 0
0001100111001111110111100001010011001000111111010110000100011100101011011100001101011001110001
1111011011000101101110100110101001111000011100110011011111111110100000001001000001011010001001 3
0010101111110000100001100101001111100011100011011011011101101101010110110000011011100011101011
0110100011011001011101111001010100111000001110110001101011101110001010101101000000110010000111
1101001100010011111010111000100010110101010011000000111110000011000110011110111111001010000111 0
0010011011010111110110001001011101011001010001111000101100110100111111001110000111101100110010 1
1111111100100000011101000011010010011100110111011111010101000100000010101000010000010010100010 5
1000101001110100011101001011010011001100111111111100000000011000000011110000011001100011111111
1011000000101110000100101100101100111100111110011110001110011011001111101111100010100011101000
1011100101001011100011001011011111001101000111110010110001110011101101111010110100100011001101
0111111100010000011010100011100001011011001001101111011110100101001001100011011111011101000101
0100101000001100010001111010101100100000111101000110010010111110110010001011110101001001000011
0110100111011001110101111101000100010010101010110000000011100000011011000011101110011010101111
1000001000110001010111101 0
```

A. No. This is output of {8, 10} LFSR with seed 01101000010!

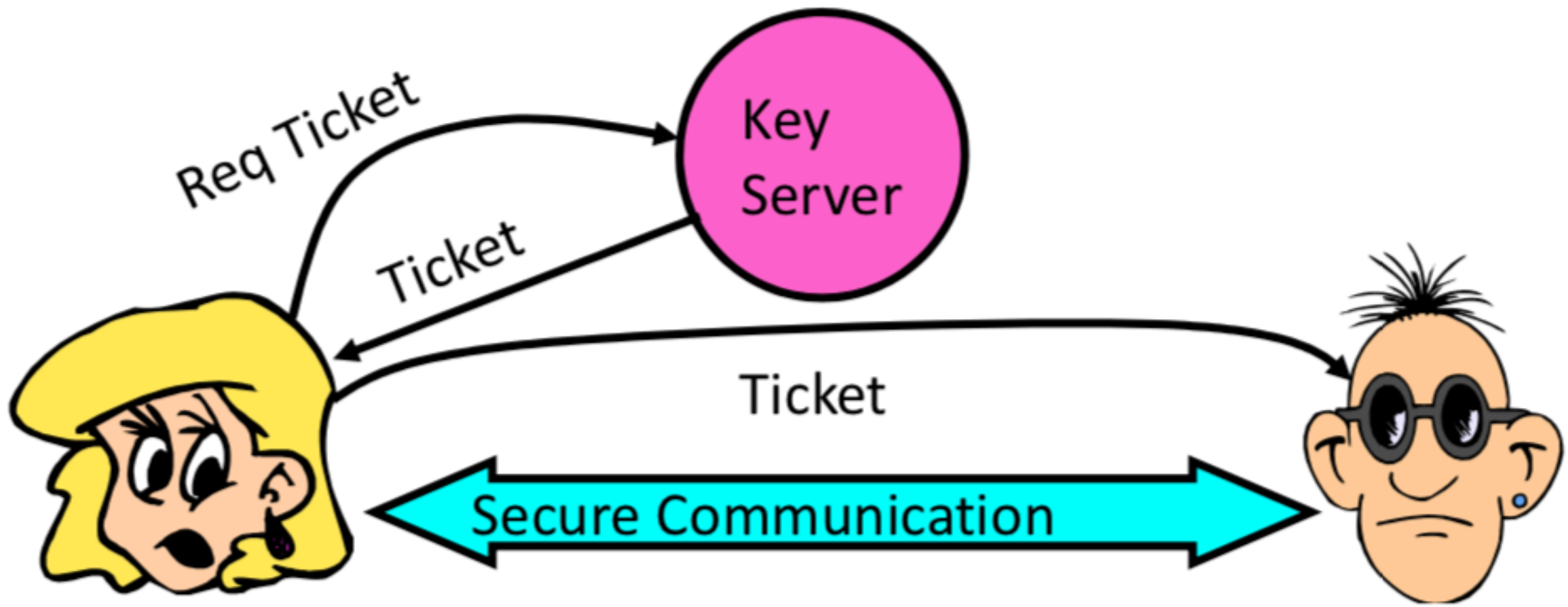
**LFSR vulnerability: even though the sequence looks random, it does repeat itself after an extended period**

Generating truly random key aside, another problem with symmetric-key encryption is key distribution: how can communication parties come to possess the shared key in the first place

# Third-Party Server

- Act as an authentication server that serves as a key distribution center for all its members, for example, the Kerberos system
- Notation:
  - $K_{xy}$ is the shared key for communication between x and y
  - $(...)^K$ is the encrypted message of (…) using key K
  - Client: members of the key distribution system, e.g., A (Alice) or B (Bob)
  - Authentication server: S

# Kerberos Authentication Server



# Kerberos Session

- A contacts the server for a key to communicate with B
  - A→S: [Hi! I'd like a key for talking between A and B]
  - The message may be sent unencrypted; others may learn that A wants to talk to B
- The server responds back with a session key
  - S→A: Message [ Use $K_{ab}$ (This is A! Use $K_{ab}$) $^{K}{}_{sb}$ ] $^{K}{}_{sa}$
- When A is ready to talk to B
  - A→B: A sends the ticket "(This is A! Use $K_{ab}$) $^{K}{}_{sb}$" to B
  - B is confident that $K_{ab}$ can be trusted as it came from S

# What We Have Learned (1)

- Symmetric-key cryptography and its evolution
  - Shift cipher
  - Substitution cipher
  - Vigenere cipher
  - One-time pad and perfect secrecy
- Steganography vs cryptography
- Block cipher vs stream cipher
- Kerberos

Is it possible to safely distribute a shared key without relying on an authentication server?

# Diffie-Hellman Key Agreement

Setup: p prime and g generator; p and g public.

$g^a \bmod p$ →

← $g^b \bmod p$

Pick random, secret a

Compute and send $g^a \bmod p$

$K = (g^b \bmod p)^a = g^{ab} \bmod p$

Pick random, secret b

Compute and send $g^b \bmod p$

$K = (g^a \bmod p)^b = g^{ab} \bmod p$

www.khanacademy.org/computing/computer-science/cryptography/modern-crypt/v/diffie-hellman-key-exchange-part-1