Security: Cryptography

Lecture 37

Some High-Level Goals

- Confidentiality (aka Authorization)
  - Non-authorized users have limited access
- Integrity
  - Accuracy/correctness/validity of data
- Availability
  - No down-time or disruptions
- Authentication
  - Agents are who they claim to be
- Non-repudiation
  - A party to a transaction can not later deny their participation

Methods of Attack

- Target people ("social engineering")
  - Phishing: email, phone, surveys, …
  - Baiting: click & install, physical media, …
- Target software ("exploits")
  - Unpatched OS, browser, programs
  - Buffer overflow
  - Code injection and cross-site scripting
- Target channel ("man-in-the-middle")
  - Eavesdropping
  - Masquerading, tampering, replay

Cryptography

- Etymology (Greek)
  - *kryptos*: hidden or secret
  - *grapho*: write
- Basic problem:
  - 2 agents (traditionally "Alice" and "Bob")
  - A & B want to exchange private messages
  - Channel between A & B is not secure ("Eve" is eavesdropping)
- Solution has other applications too
  - Protect *stored* data (*e.g.* on disk, or in cloud)
  - Digital *signatures* for non-repudiation
  - Secure *passwords* for authentication

Core Idea: A Shared Secret

- Alice & Bob share some *secret*
  - Secret can not be the message itself
  - Secret used to protect arbitrary messages
- Crude analogy: a padlock
  - Copies of the physical key are the secret
  - Alice puts message in box and locks it
  - Bob unlocks box and reads message
- But real channels are bit streams
  - Eve can see the bits!
  - Message must be garbled in some way
  - Secret is strategy for garbling/degarbling

Protecting Messages

- Alice garbles (encrypts) the message
- Sends the encrypted cipher-text
- Bob knows how to degarble (decrypt) cipher-text back into plain-text

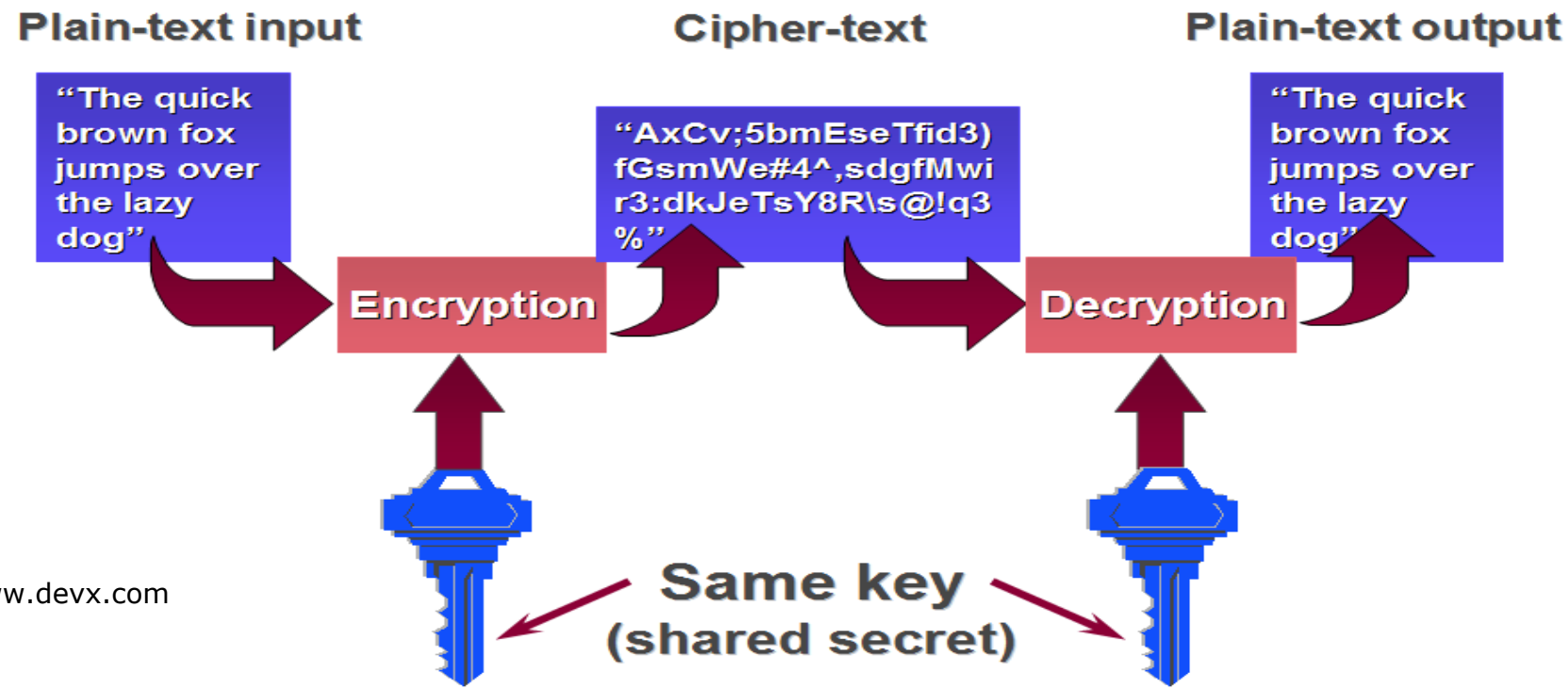
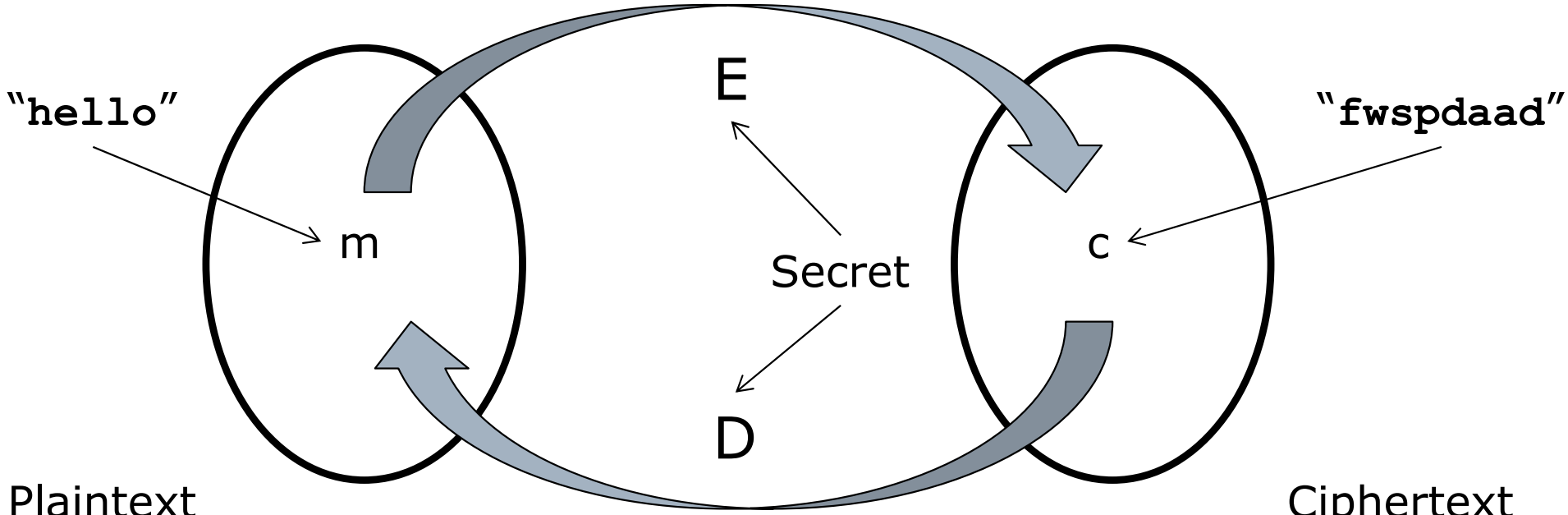


Image: www.devx.com

Encryption/Decryption Function



E: P → Q
D: Q → P

E(m) = c
D(c) = m
i.e. D = E^{-1}

Note: often P = Q
So E is a *permutation*

Families of Encryption Functions

- Each pair of agents needs their own E
  - Many E's (& corresponding D's) needed
- But good E's are hard to invent
- Solution: design one (good) E, which is *parameterized* by a number
  - That is, have a huge *family* of E's: E_0, E_1, E_2, … E_K
  - Everyone knows the family of E's
  - Secret: *which* E_i is used (*i* is the key)

Classic Example: Caesar Cipher

- Shift each letter by *x* positions in alphabet
  - Example: *x* = 3
    a → d, b → e, c → f, d → g, e → h, …
  - The key is *x*
- Encode a string character-by-character
  - For m = "**hello world**", E_3(m) = "**khoor zruog**"
- Questions:
  - What is P (set of plaintext messages)?
  - What is Q (set of ciphertext messages)?
  - How many different ciphers?
  - Is this a strong or weak cipher?

Classic Example: Caesar Cipher

- Shift each letter by *x* positions in alphabet
  - *E.g. x* = 3
    a → **d**, **b** → **e**, **c** → **f**, **d** → **g**, **e** → **h**, …
  - The key is *x*
- Encode a string character-by-character
  - For m = "**hello world**", E_3(m) = "**khoor zruog**"
- Questions:
  - What is P (set of plaintext messages)?
    - The alphabet, ie {"**a**", "**b**", "**c**", "**d**", "**e**", …}
  - What is Q (set of ciphertext messages)?
    - The alphabet, ie {"**a**", "**b**", "**c**", "**d**", "**e**", …}
  - How many different ciphers?
    - 26
  - Is this a strong or weak cipher?
    - Weak: Just try all 26 possibilities

Generalized Caesar Cipher

- Generalization: arbitrary mapping
  - Example: The qwerty shift

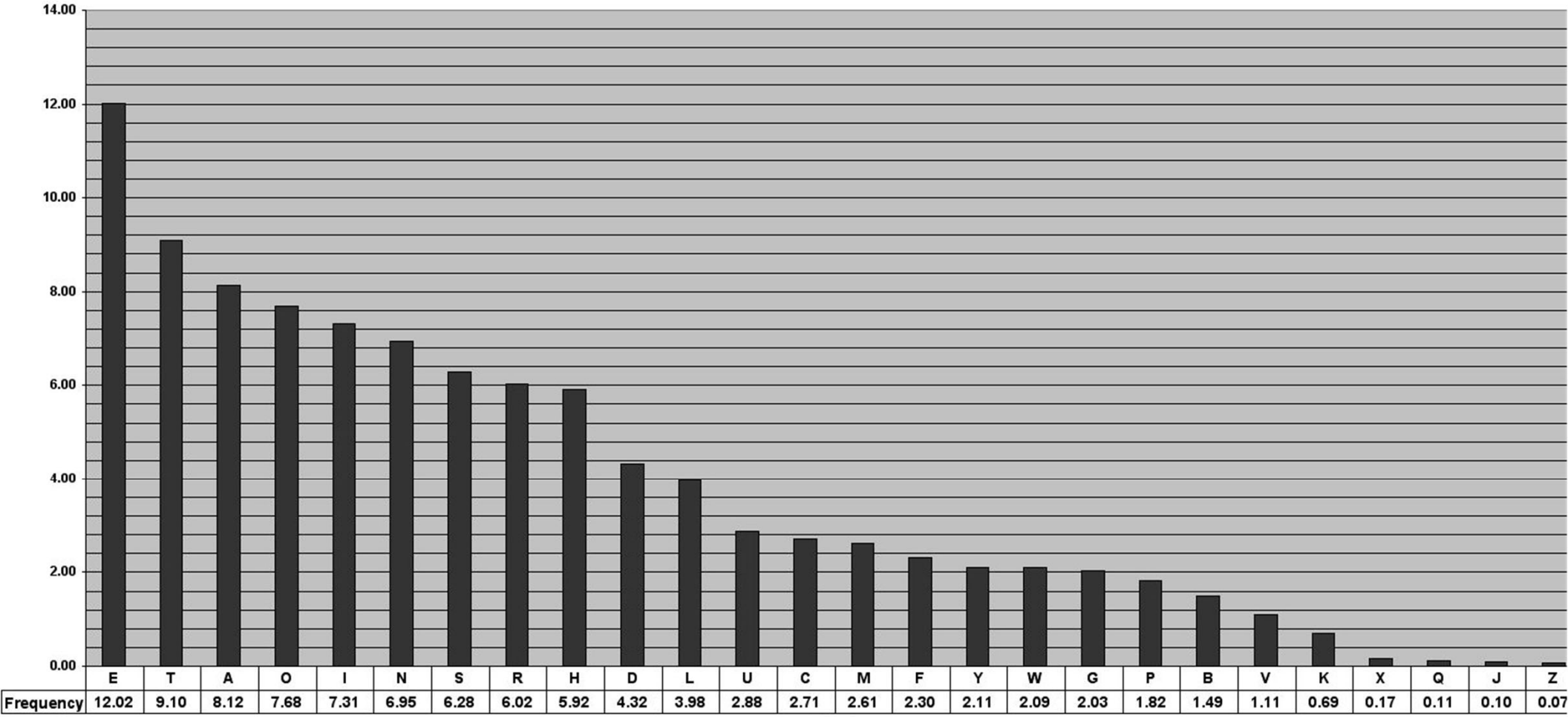
    a → s, b → n, c → v, d → f, e → r, …
  - For m = "**hello world**",
    E(m) = "**jraap eptaf**"
  - 26! possible ciphers… that's a lot!
    - Approximately 4 x 10^{26}
    - There are ~10^{18} nanoseconds/century
- Weakness?

Generalized Caesar Cipher

- Generalization: arbitrary mapping
  - Example: The qwerty shift

    a → s, b → n, c → v, d → f, e → r, …
  - For m = "**hello world**",
    E(m) = "**jraap eptaf**"
  - 26! possible ciphers… that's a lot!
    - Approximately 4 x 10^{26}
    - There are ~10^{18} nanoseconds/century
- Weakness?
  - In English text, letters appear in predictable ratios
  - From enough ciphertext, can infer E

Frequency Analysis



	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

Leon Battista Alberti

WW II: Enigma Machine

Polyalphabetic Cipher

- Alberti's idea: Use different E_i's within the same message
  - E("**hello world**") = E_a("**h**")E_b("**e**")E_c("**l**")E_d("**l**")E_e("**o**")…
- Alice & Bob agree on the *sequence* of E's to use
- Claude Shannon proved that this method is perfectly secure (1949)
  - Precise information-theoretic meaning
  - Known as a *one-time pad*

One-Time Pad

- Message is a sequence of bits

  $m_0\ m_1\ m_2\ m_3\ m_4\ m_5\ m_{6\ldots}$

- One-time pad is *random* bit sequence

  $x_0\ x_1\ x_2\ x_3\ x_4\ x_5\ x_{6\ldots}$

- E is bit-wise XOR operation, $\oplus$
- Cipher text is

  $m_0 \oplus x_0\ m_1 \oplus x_1\ m_2 \oplus x_2\ m_3 \oplus x_3\ m_4 \oplus x_4\ m_5 \oplus x_5\ m_6 \oplus x_{6\ldots}$

- Problem: Pad is long and cannot be re-used (hence cumbersome to share)
- In practice: pseudo-random sequence, generated from a seed (the key)
  - *Not* perfectly secure, in Shannon sense

Comparison: Stream *vs* Block

Stream Cipher

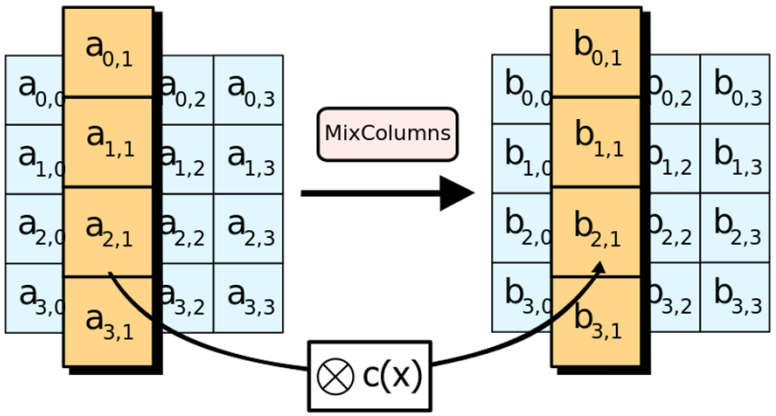
- Encrypts bit-by-bit
- $|P| = |Q| = 2$
- Few choices for E (roughly 2)
- Message can have any length

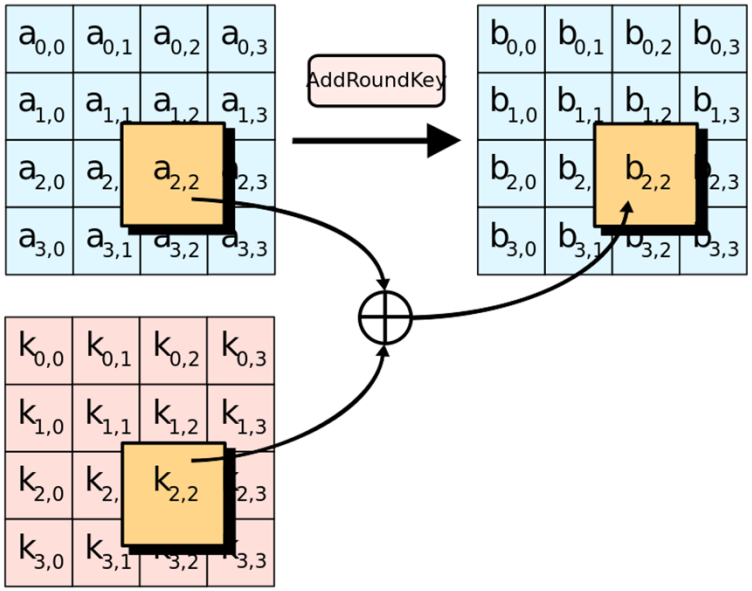
Block Cipher

- Encrypts a fixed-length (k-bit) sequence
- $|P| = |Q| = 2^k$
- Many choices for E (roughly $2^k!$)
- Padding added s.t. $|m| \bmod k = 0$

Example of Block Cipher: AES

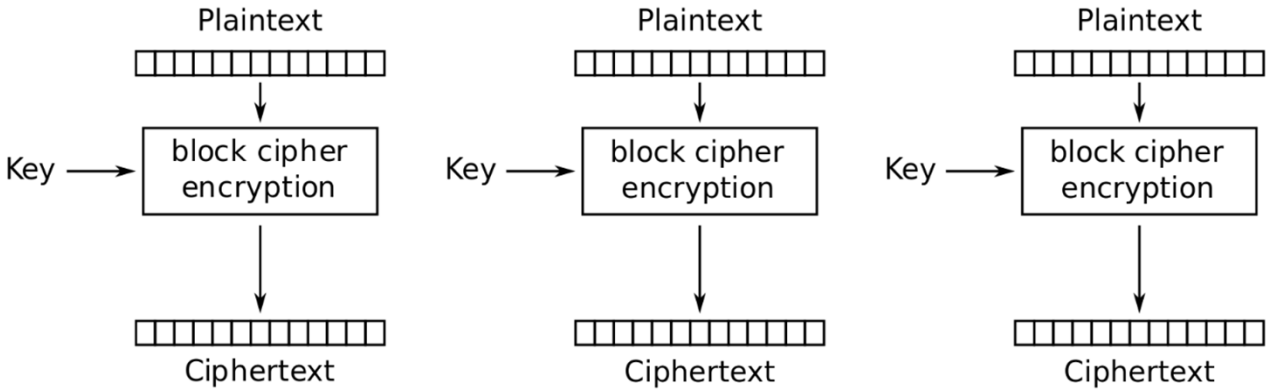
- Advanced Encryption Standard (2001)
  - Replaced DES (1977)
- Block size always 128 bits (4x4 bytes)
- Key size is 128, 192, or 256 bits
- Multi-step algorithm, many rounds



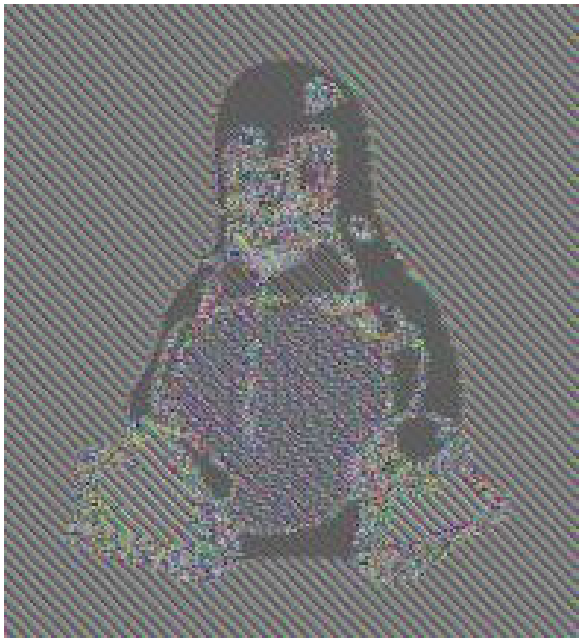


Limitation of Fixed Block Size

- Message can be longer than block size
- Reuse same E for each block?
  - Danger: Frequency analysis vulnerability
  - Don't do this (for multiblock messages)!



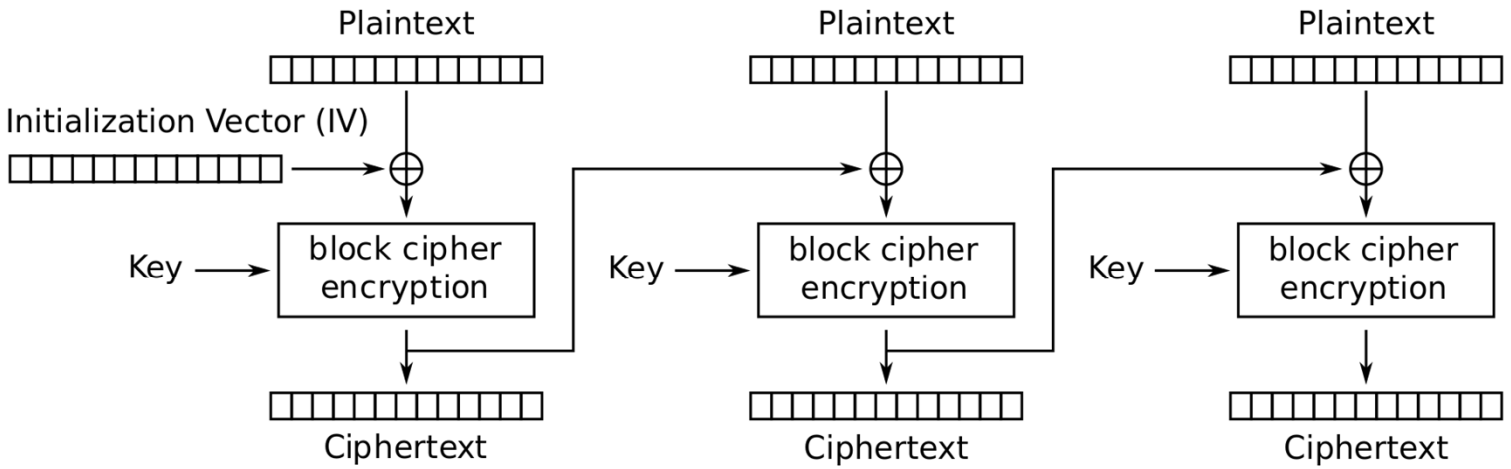
Electronic Codebook (ECB) mode encryption



https://en.wikipedia.org/wiki/Image:Tux_ecb.jpg
https://commons.wikimedia.org/wiki/File:Tux.jpg

Solution: Initialization Vector

- Add a random block to start
- Combine adjacent blocks to make ciphertext block
  - Many combination strategies (aka modes)



Cipher Block Chaining (CBC) mode encryption

Summary

- Cryptography
  - Encryption: Maps plaintext → ciphertext
  - Decryption is the inverse
- Symmetric-key encryption
  - Sender and receiver share (same) secret key
  - Stream ciphers work one bit at a time (*e.g.*, one-time pad)
  - Block ciphers work on larger blocks of bits (*e.g.*, AES)