CSE331 Computer Security Fundamentals

9/26/2017

Symmetric Key Cryptography

Michalis Polychronakis

Stony Brook University

Cryptography

Goals

Confidentiality

Keep content secret from all but authorized entities

Integrity

Protect content from unauthorized alteration

Authentication

Confirm the identity of communicating entities or data

Non-repudiation

Prevent entities from denying previous commitments or actions

Basic Terminology

Plaintext: top secret message

Ciphertext: eza dpncpe xpddlrp

Cipher: algorithm for transforming plaintext to ciphertext *(encryption)* and back *(decryption)*

Key: (usually secret) information used in a cipher, known to sender, receiver, or both

Cryptanalysis (codebreaking): the study of methods of deciphering ciphertext without knowing the key

Cryptology: the broader field of "information hiding" cryptography, cryptanalysis, steganography, …

Cryptosystem

A suite of cryptographic algorithms that take a key and convert between plaintext and ciphertext

Main components

Plaintext space: set $\boldsymbol{P}$ of possible plaintexts

Ciphertext space: set $\boldsymbol{C}$ of possible ciphertexts

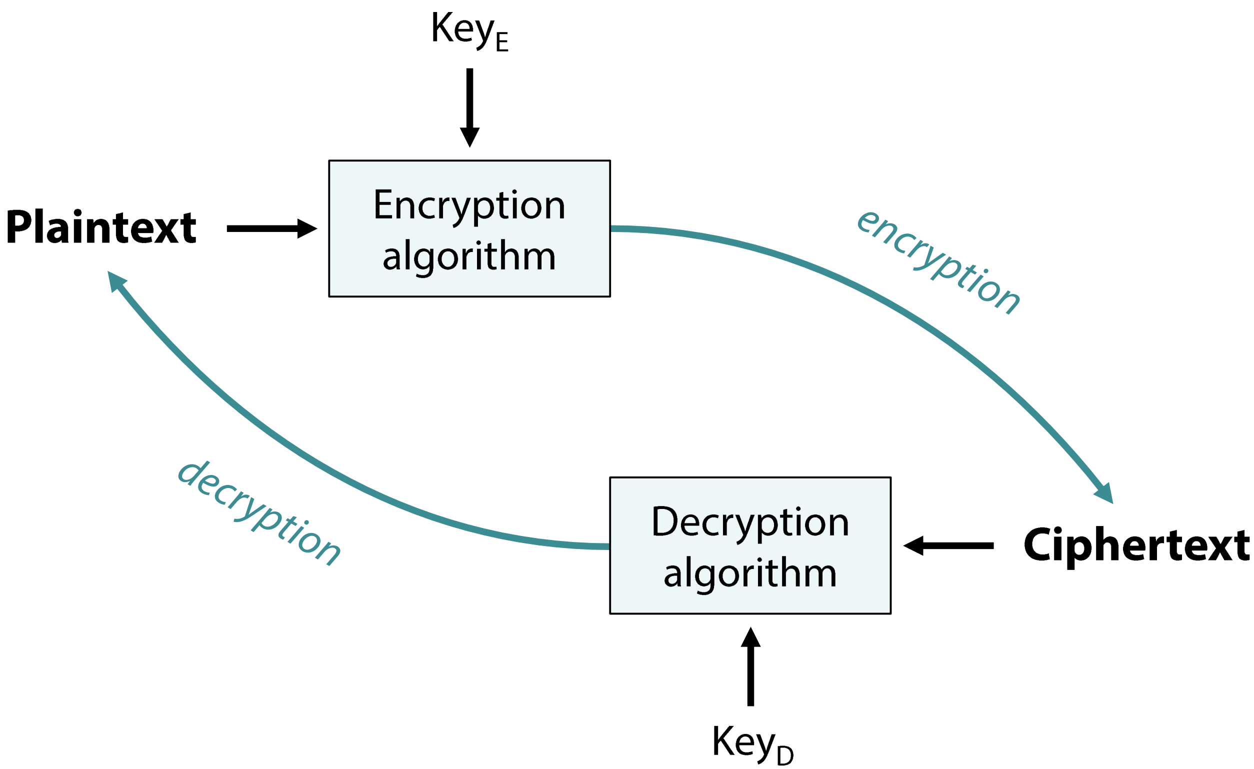
Key space: set $\boldsymbol{K}$ of encryption/decryption keys

Encryption algorithm: $\boldsymbol{E : P \times K \rightarrow C}$

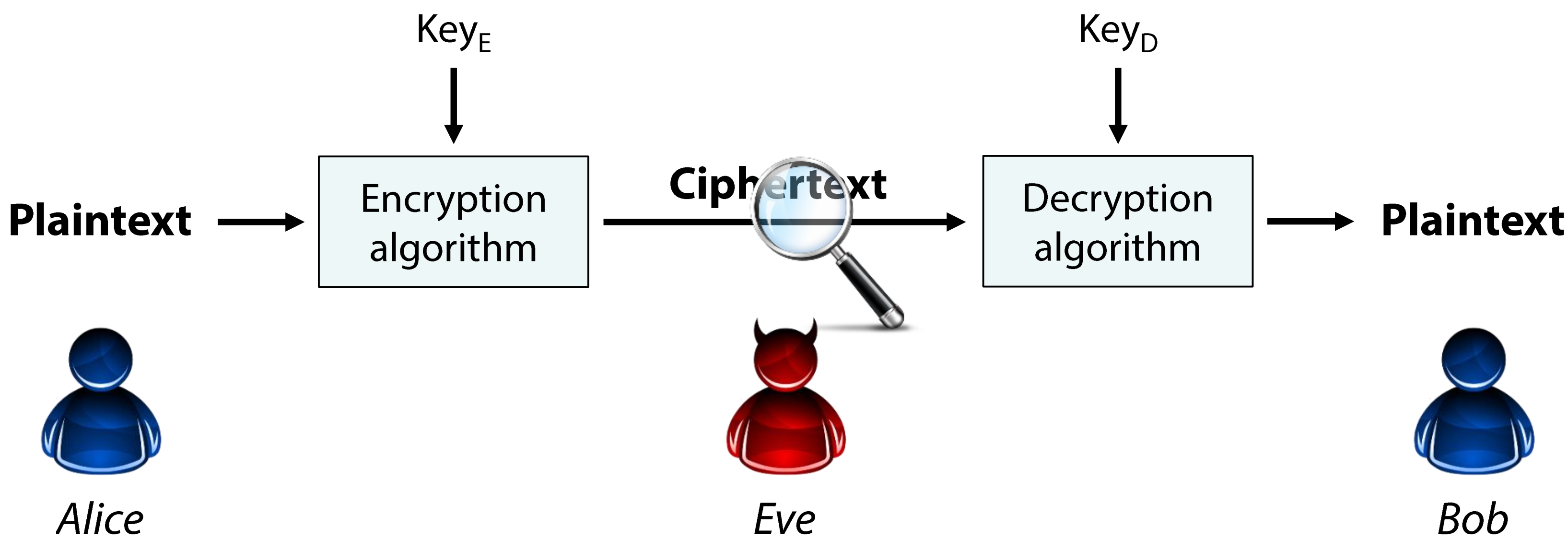
Decryption algorithm: $\boldsymbol{D : C \times K \rightarrow P}$

$$\forall p \in P, k \in K : D(E(p, k), k) = p$$

Plaintext vs. Ciphertext



Basic Scenario



Cryptographic Function Types

Hash functions: *no key*

- Input of arbitrary length is transformed to a fixed-length value
- One-way function: hard to reverse

Secret (symmetric) key functions: *one key*

- Shared secret key is used for both encryption and decryption

Public (asymmetric) key functions: *two keys*

- *Key pair:* public key is known, private key is kept secret
- Encrypt with public key and decrypt with private key
- Encrypt with private key and decrypt with public key

Kerckhoffs's Principle

A cryptosystem should be secure even if everything about the system, except the key, is public knowledge

The security of the system must rest entirely on the secrecy of the key

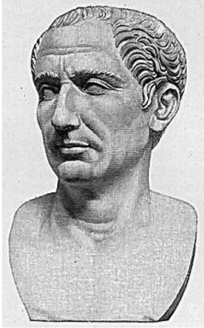
- Only brute force attacks are possible
- Otherwise the algorithm is broken

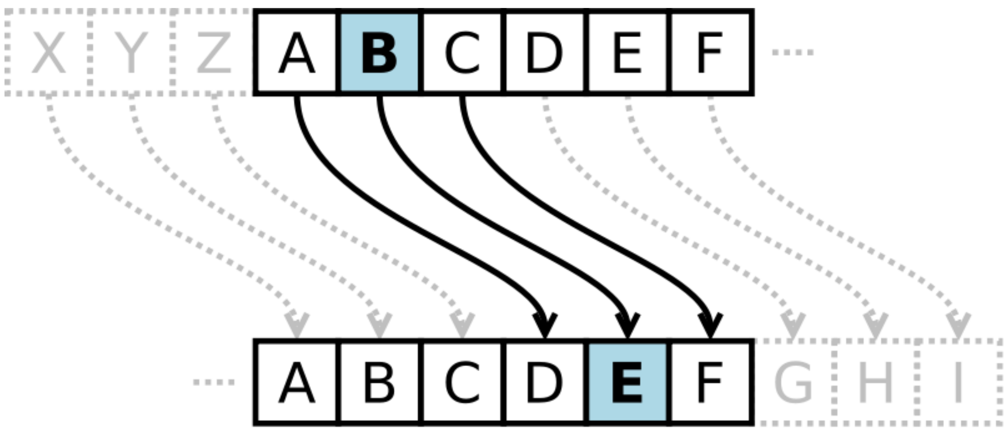
Contrast with security by obscurity: every secret creates a potential failure point

- Widely used secret algorithms will eventually be reverse engineered (or leaked, stolen, …)
- Difficult to deploy a new algorithm if an old one is compromised

A public implementation enables scrutiny by experts

Caesar Cipher






```
Ciphertext: WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ
Plaintext:  the quick brown fox jumps over the lazy dog
```

Shift by x (e.g., ROT-13)

Monoalphabetic substitution

Shift Ciphers

Plaintext space: $\boldsymbol{P} = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots, \mathbf{Z}\}$

Ciphertext space: $\boldsymbol{C} = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots, \mathbf{Z}\}$

Key space: $\boldsymbol{K} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots, \mathbf{25}\}$

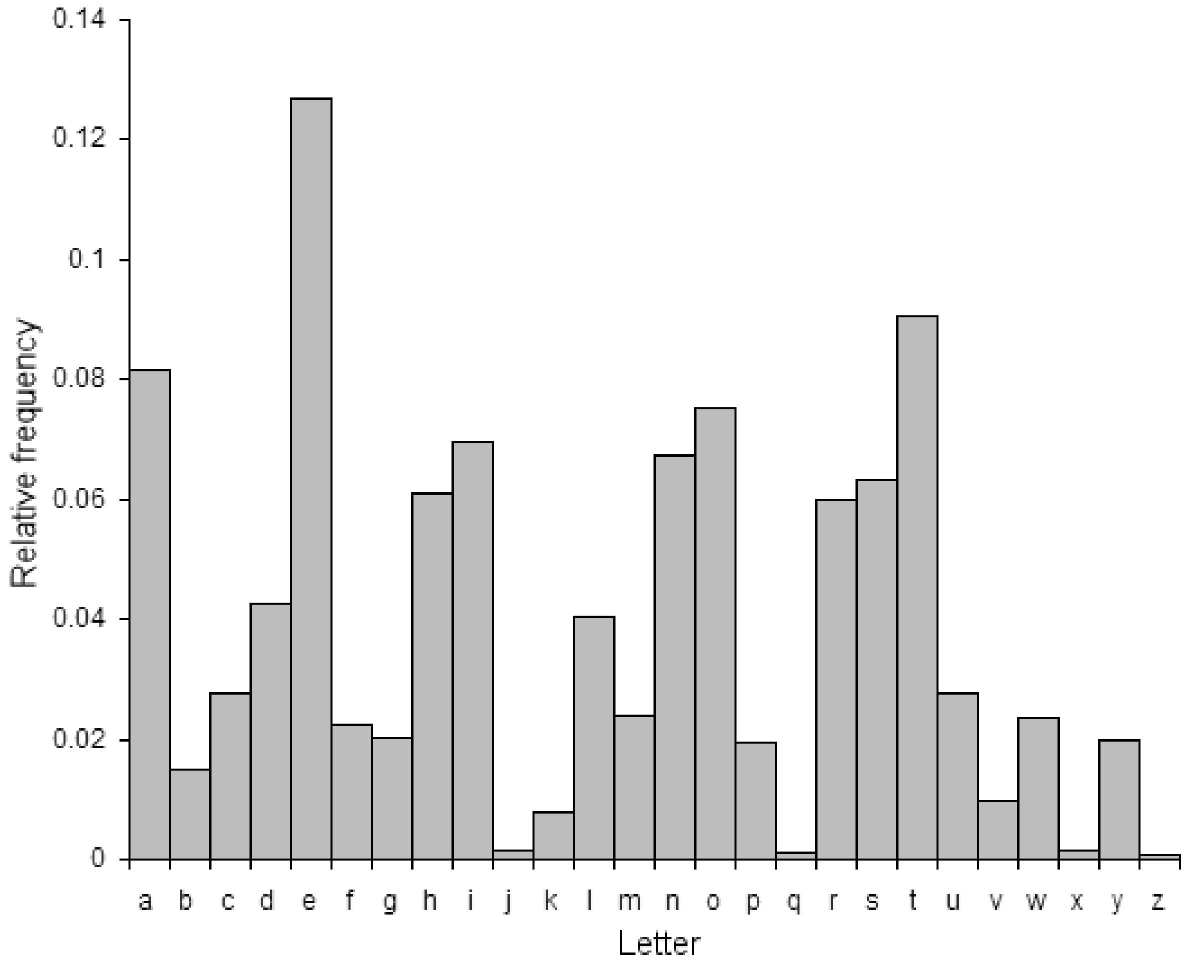
Encryption algorithm: $\boldsymbol{E(x, k) = (x + k) \bmod 26}$

Decryption algorithm: $\boldsymbol{D(x, k) = (x - k) \bmod 26}$

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Caesar Cipher: $k = 3$

Easy to break using frequency analysis



Distribution of letters in a typical sample of English language text

Vigenère Cipher

Plaintext: ATTACKATDAWN

Key: **LEMON**LEMONLE

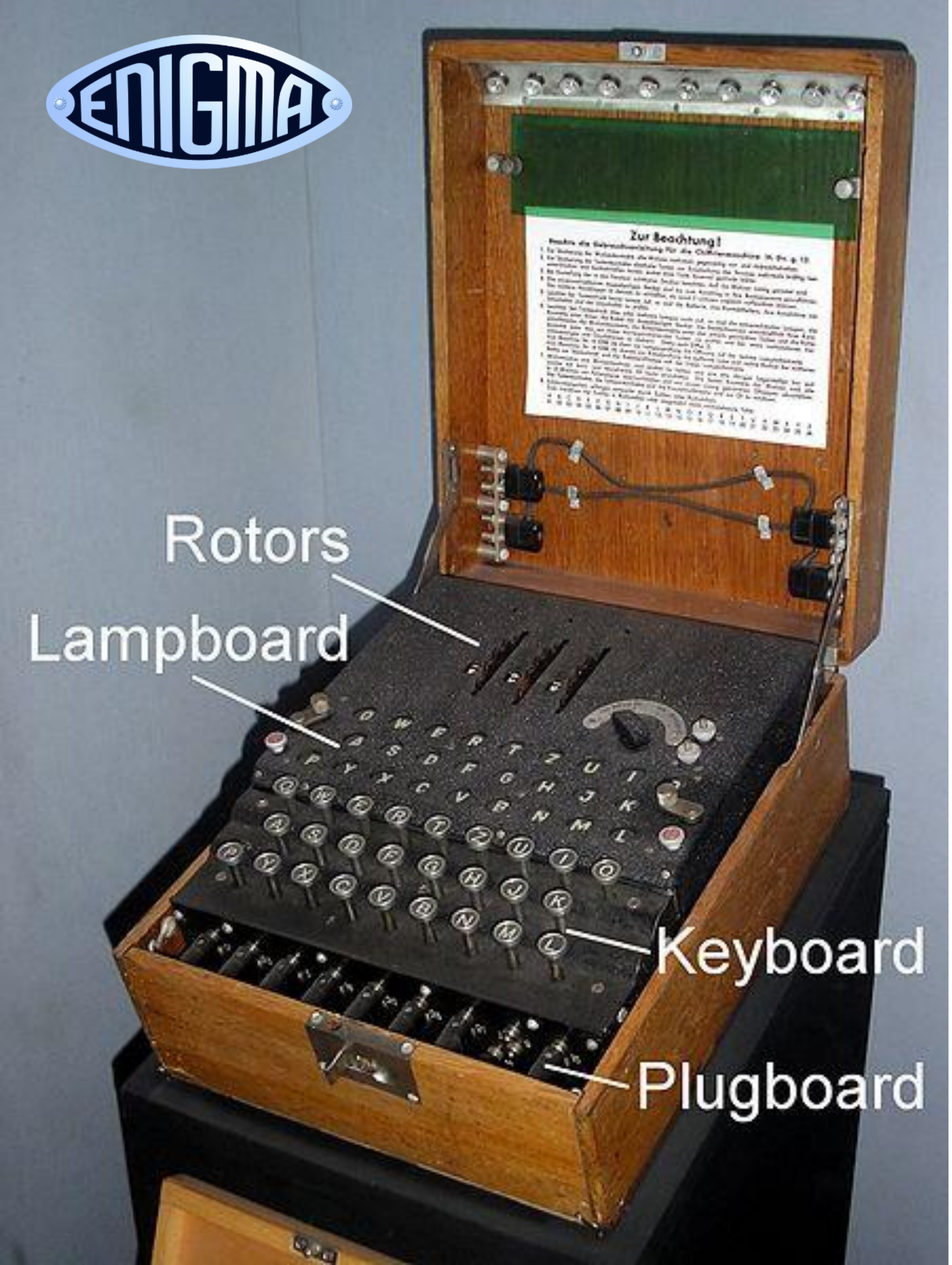
Ciphertext: LXFOPVEFRNHR

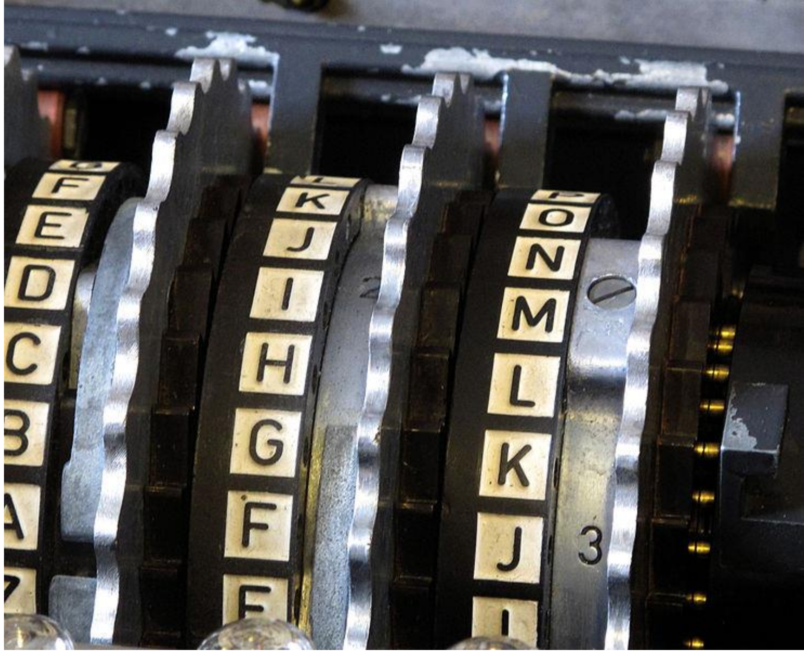
Successive Caesar ciphers with different shift values depending on a key

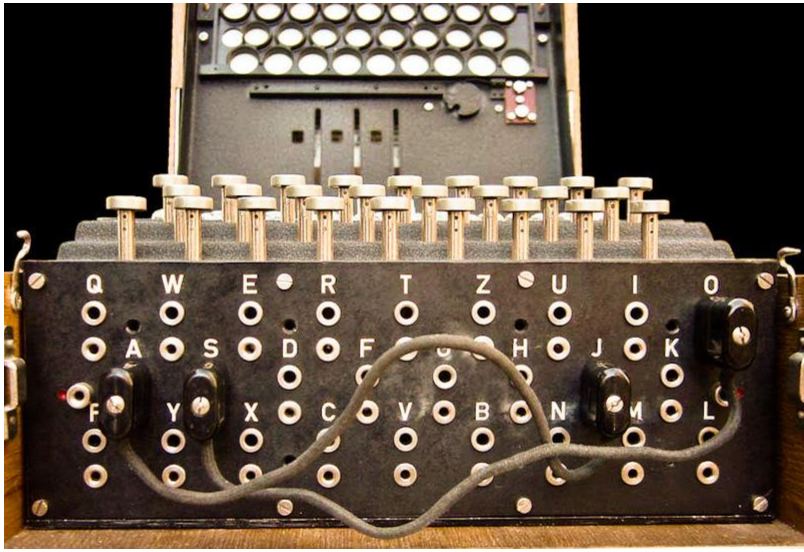
Defeats simple frequency analysis, but still breakable

Polyalphabetic substitution

	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


ENIGMA
Rotors
Lampboard
Keyboard
Plugboard





Properties of a Good Cryptosystem

Given the ciphertext, an adversary should not be able to recover the original message

- Enumerating all possible keys must be infeasible
- There should be no way to produce plaintext from ciphertext without the key

The ciphertext must be indistinguishable from true random values

- Given a ciphertext, the probability of any possible plaintext being encrypted should be the same

Cryptographic algorithms should be computationally efficient for practical use

- Fast encryption/decryption/hashing
- There are exceptions: deliberately slow password-based key derivation functions for hindering brute force/dictionary attacks

Basic Attack Models

Known Ciphertext: attacker has access to only a set of ciphertexts

- In practice some information about the plaintext might be available: language, character distribution, protocol fields, …
- Brute force frequency analysis, …

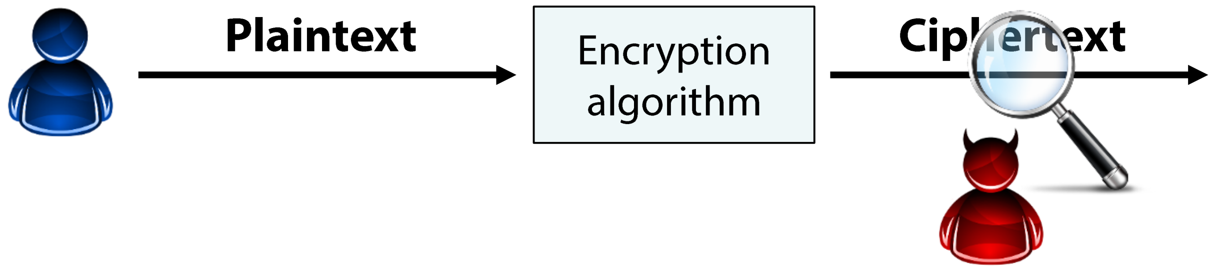
Known Plaintext: attacker has access to both the plaintext and its corresponding ciphertext

- *Passive attacker:* has at least one sample of both
- Even partial mappings can be enough

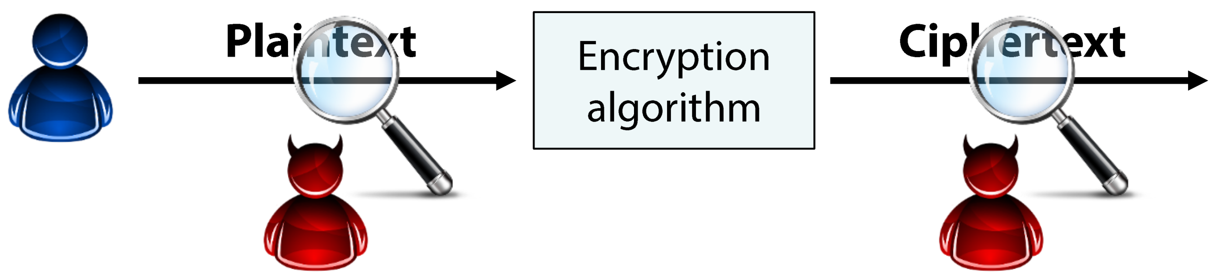
Chosen Plaintext: attacker can obtain the ciphertexts of arbitrary plaintexts

- *Active attacker:* has access to an *encryption oracle*

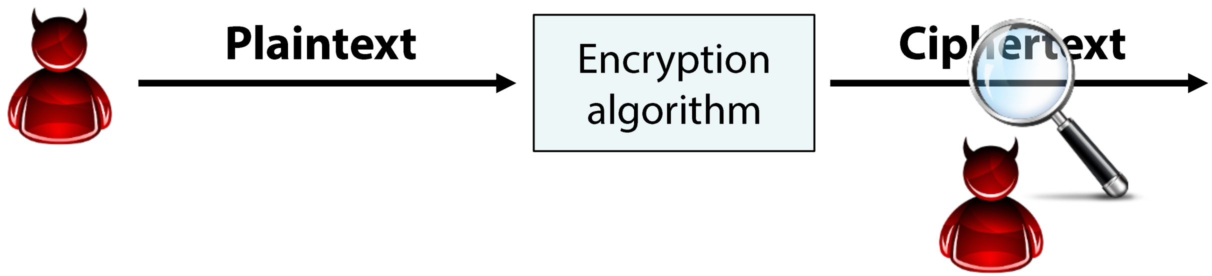
Known Ciphertext



Known Plaintext



Chosen Plaintext



Computational Difficulty

Modern cryptography: seek guarantees about the "strength" of encryption schemes

- Codes, secret writing, and other older encryption schemes were ad hoc and eventually broken

Information-theoretic security

- Cannot be broken even with unlimited computing power: *there is simply not enough information*
- Not possible if the key is shorter than the message size ➔ impractical

Computational security

- Can be broken with enough computation, but *not in a reasonable amount of time*
- Rely on *computationally hard* problems: easy to compute but hard to invert in polynomial time (integer factorization, discrete logarithm, …)
- Assume *computationally limited adversaries* ➔ frustrate exhaustive enumeration

One-time Pad

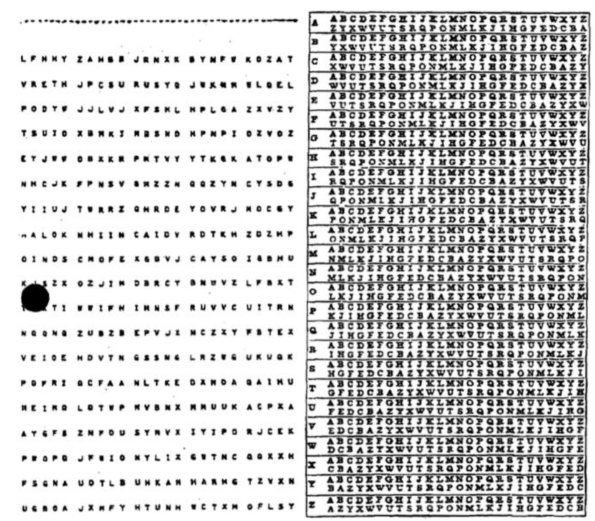
XOR plaintext with a keystream

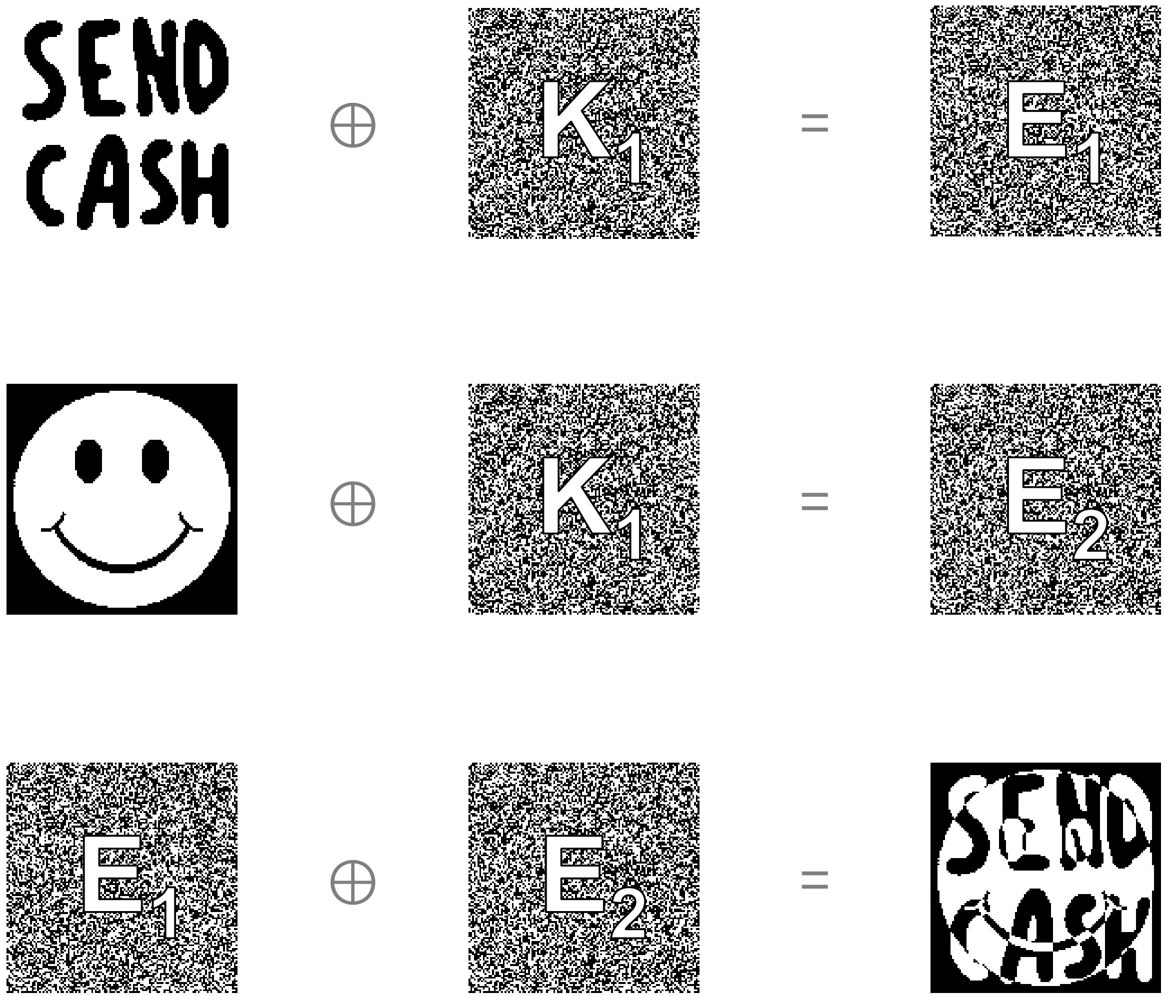
- 1882 Frank Miller [Bellovin '11]
- 1917 Vernam/Mauborgne cipher

Information-theoretically secure against ciphertext-only attacks (Shannon 1949)

The keystream must be

- Truly random
- As long as the plaintext
- Kept completely secret
- Used only once…




SEND CASH
⊕
K1
=
E1
⊕
K1
=
E2
E1
⊕
E2
=
SEND CASH

One-time Pad

Plaintext space: *all n-bit sequences*

Ciphertext space: *all n-bit sequences*

Key space: *all n-bit sequences*

Encryption algorithm: $\boldsymbol{E(x, k) = x \oplus k}$ *(bit by bit)*

Decryption algorithm: $\boldsymbol{D(x, k) = x \oplus k}$ *(bit by bit)*

Advantages

Easy to compute: simple XOR operation

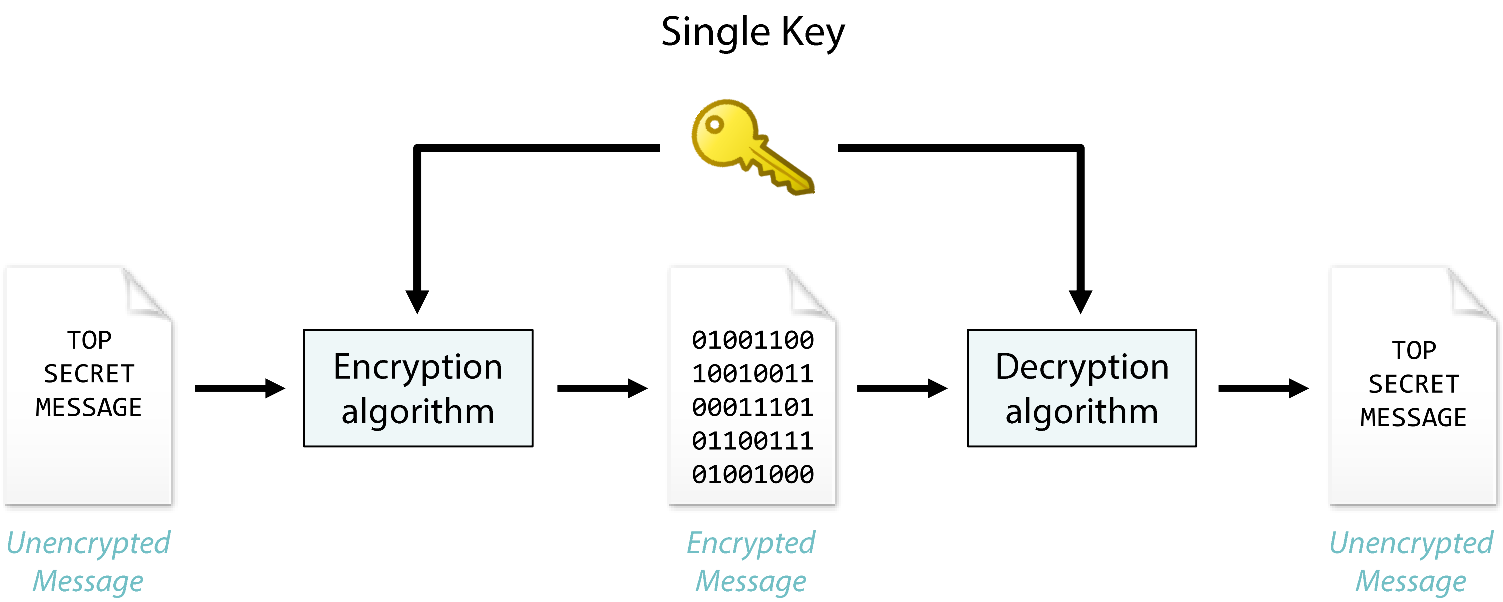
Impossible to break: information-theoretically secure

Disadvantages

Key must be as long as the plaintext

How can the sender provide the key to the receiver securely?

Symmetric Key Cryptography



Pros:

Fast

Short keys

Well known

Simple key generation

Cons:

Secrecy of keys

Number of keys

Management of keys

$n(n-1)/2$ keys needed for n parties

Block Ciphers

Process one block at a time

Substitution and transposition (permutation) techniques

Examples: *DES (Data Encryption Standard), AES (Advanced Encryption Standard) – replaced DES*

Stream Ciphers

Process one bit or byte at a time

Plaintext is combined (XOR) with a *pseudorandom* keystream *(NOT the same as one-time pad)*

Synchronous vs. asynchronous (self-synchronizing)

Examples: *RC4, any block cipher in OFB or CTR mode, …*

Block Ciphers

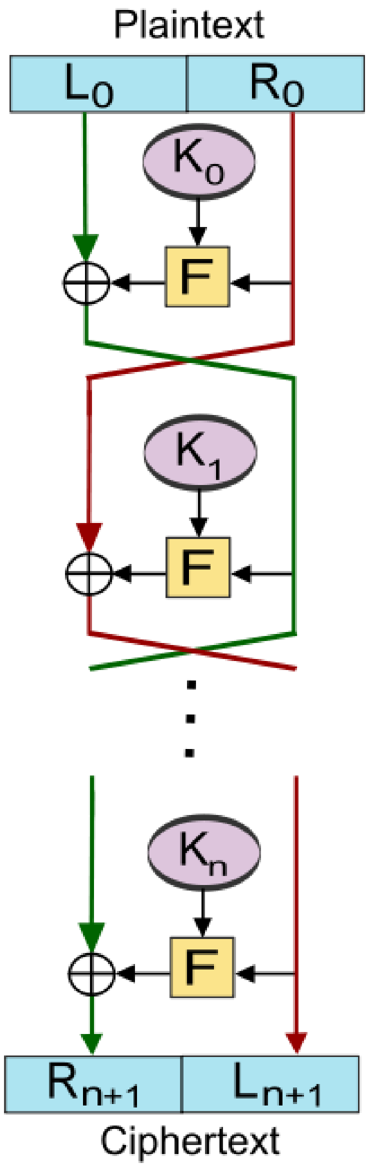
Multiple rounds of substitution, permutation, …

Confusion: each character of the ciphertext should depend on several parts of the key

Diffusion: changing a plaintext character should result in several changed ciphertext characters

	DES	AES
Key length	56 bits	128, 192, 256 bits
Block size	64 bits	128 bits
Rounds	16	10, 12, 14
Construction	Substitution, permutation	Substitution, permutation, mixing, addition
Developed	1977	1998
Status	Broken!	OK (for now)

DES rounds



Modes of Operation

Direct use of block ciphers is not very useful

- Enemy can build a "code book" of plaintext/ciphertext equivalents
- Message length should be multiple of the cipher block size

How to repeatedly apply a block cipher to securely encrypt/decrypt arbitrary inputs?

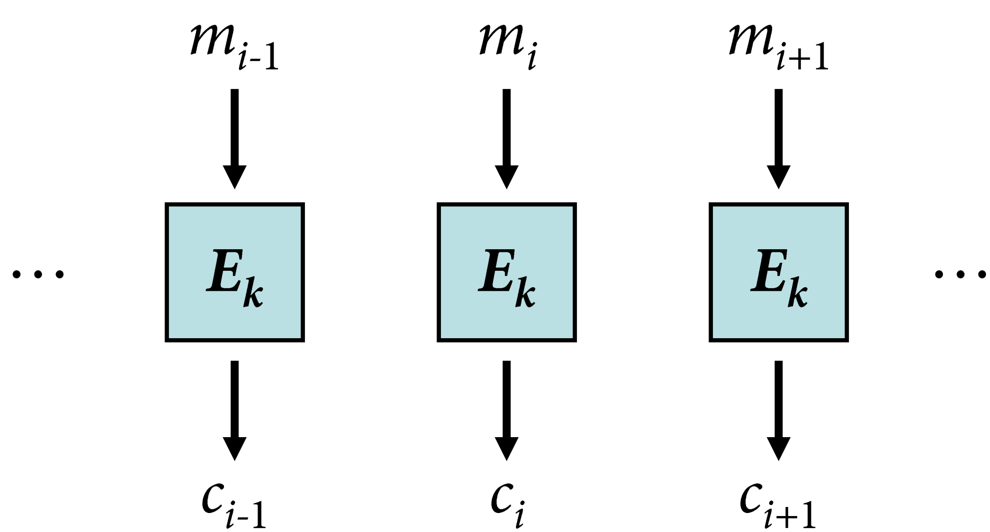
Five standard modes

- ECB: Electronic Code Book
- CBC: Cipher Block Chaining
- CFB: Cipher Feedback
- OFB: Output Feedback
- CTR: Counter

ECB: Electronic Code Book Mode

Direct use of the block cipher

- Each block is encrypted independently ➔ parallelizable
- No chaining, no error propagation



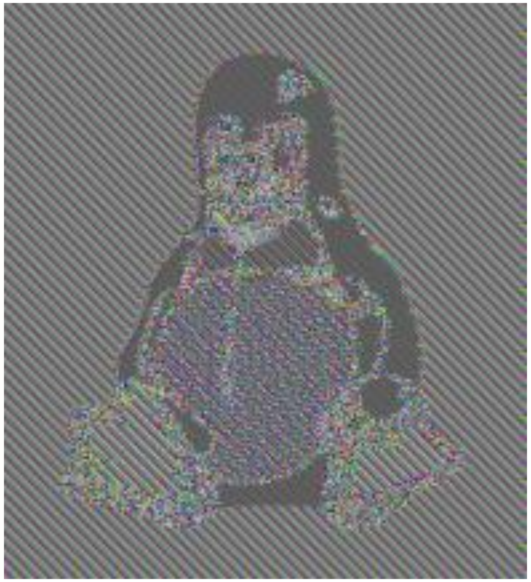
Problem: if $m_i = m_j$ then $c_i = c_j$

ECB: Electronic Code Book Mode

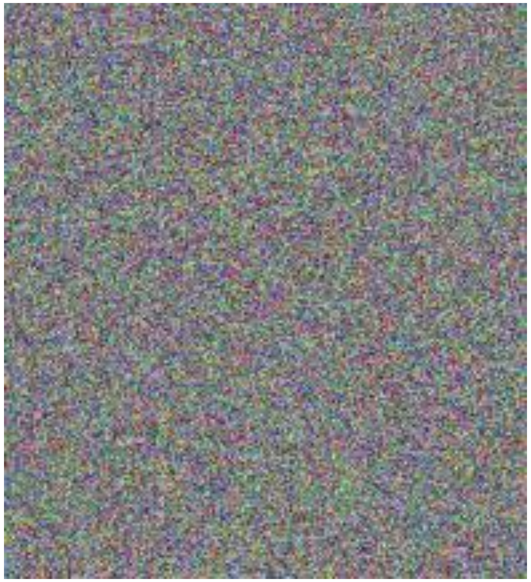
Data patterns may remain visible

Susceptible to replay attacks, block insertion/deletion

Plaintext



ECB Mode Encryption

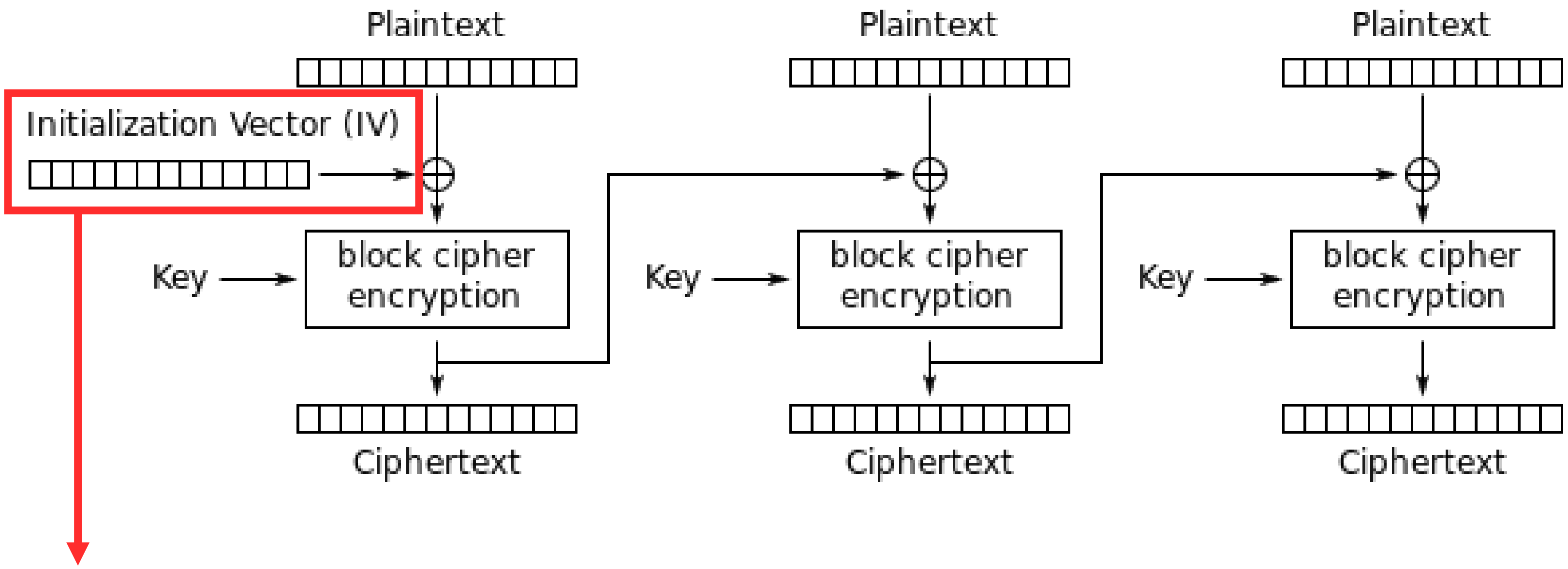


CBC/Other Modes

CBC: Cipher Block Chaining Mode

Each plaintext block is XOR'ed with the previous ciphertext block before being encrypted ➔ obscures any output patterns

Sequential process (non-parallelizable)

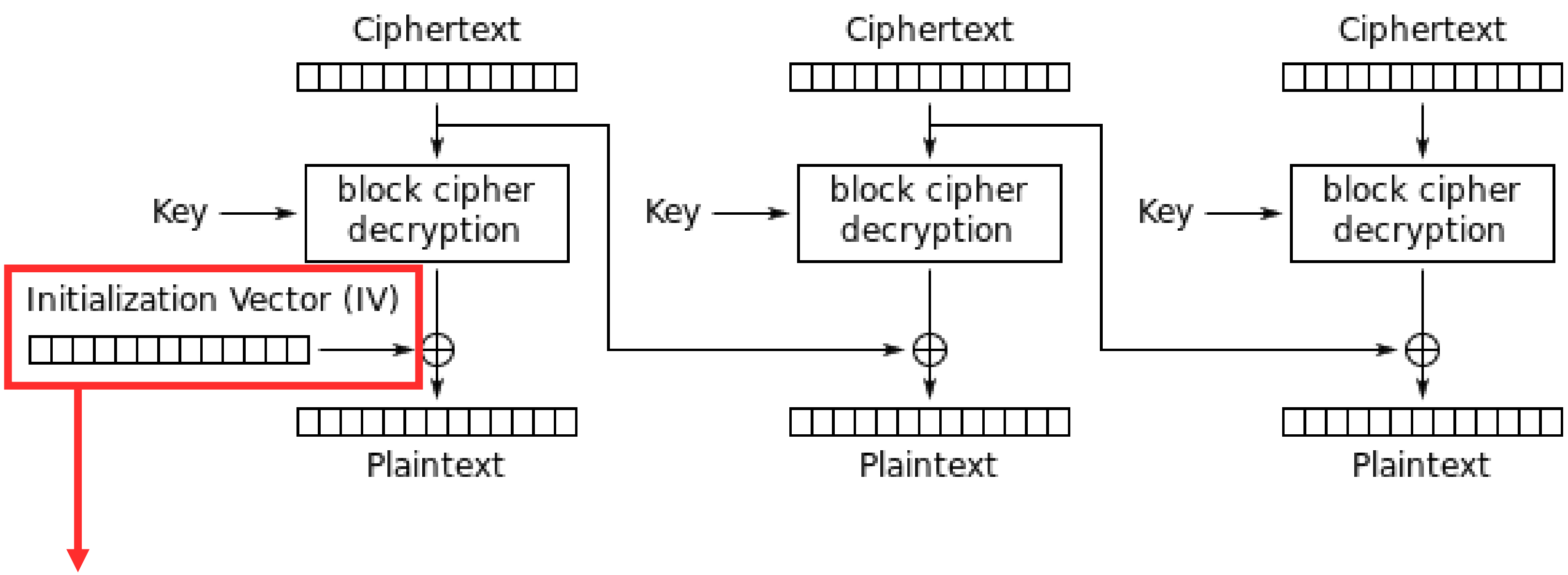


Ensures that no messages have the same beginning

Must be random! Must never be reused!

CBC: Decryption

An error in a transmitted ciphertext block also affects its following block (but not subsequent ones)

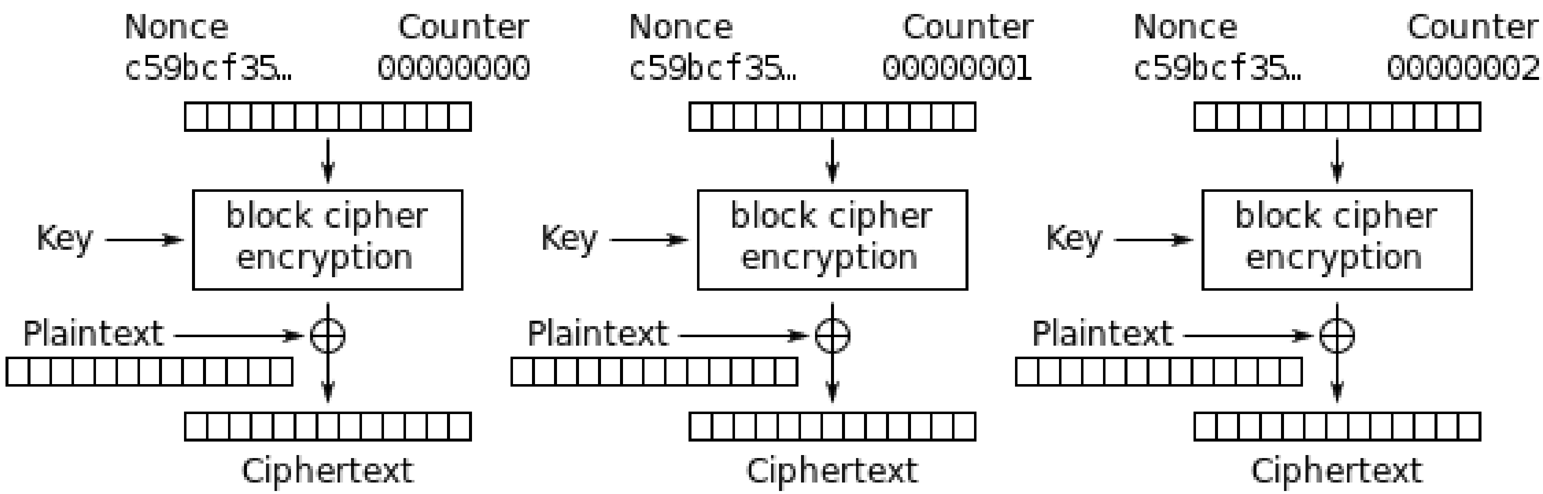


Both parties must use the same IV: can be transmitted with the message

CTR: Counter Mode

Turns a block cipher into a stream cipher

- Next keystream block is generated by encrypting successive values of a counter combined with a nonce (IV)



Counter (CTR) mode encryption