

CS459/659

Privacy, Cryptography, Network and Data Security

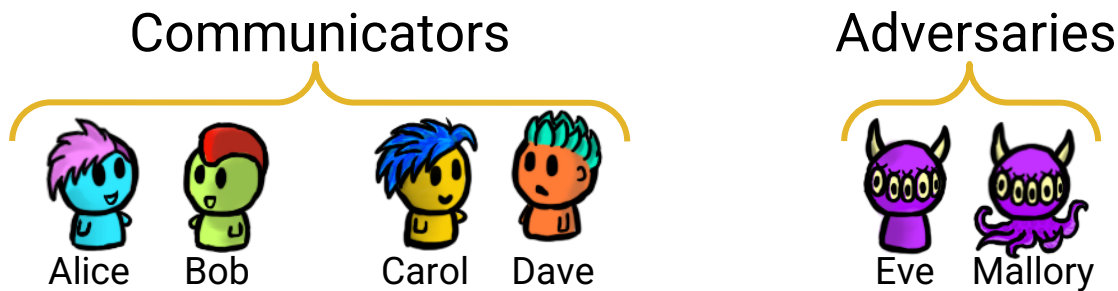
Cryptography Basics

Learning Outcomes

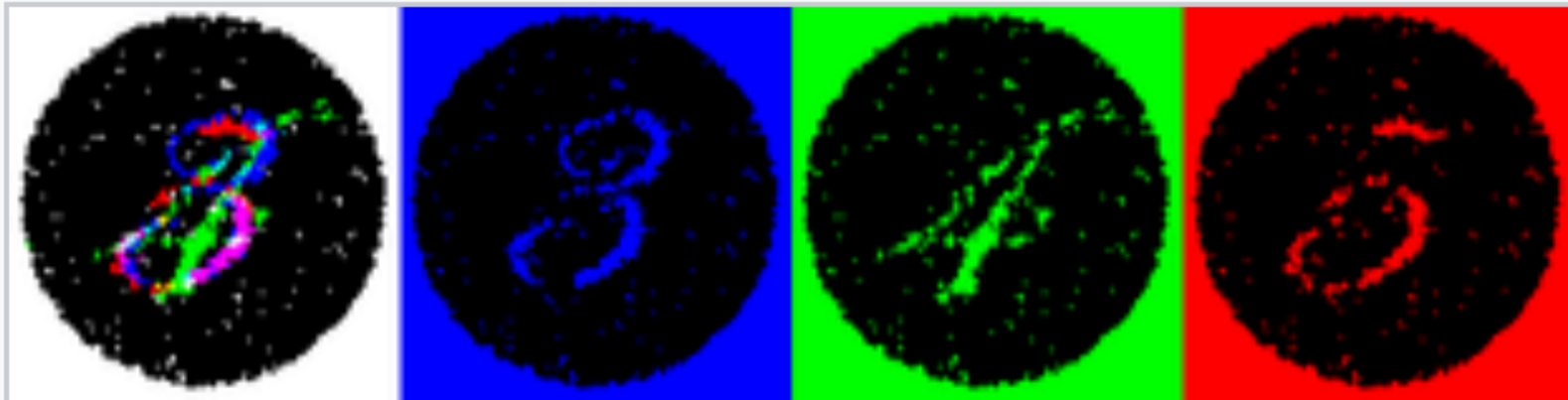
- Identify attack techniques and apply them (cryptanalysis)
- Explain building blocks of modern cryptography
- Explain how modern cryptography properties arose

Goal: Basically, know what cryptography tools exist and how to securely use them. Build a foundation of primitives for more complicated “applied cryptography” later.

Cryptography - Writing “secret” messages



Steganography- Secretly “hidden” messages

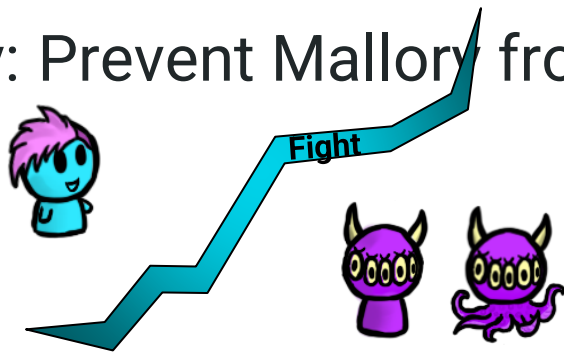


The same image viewed by white, blue, green, and red lights reveals different hidden numbers.

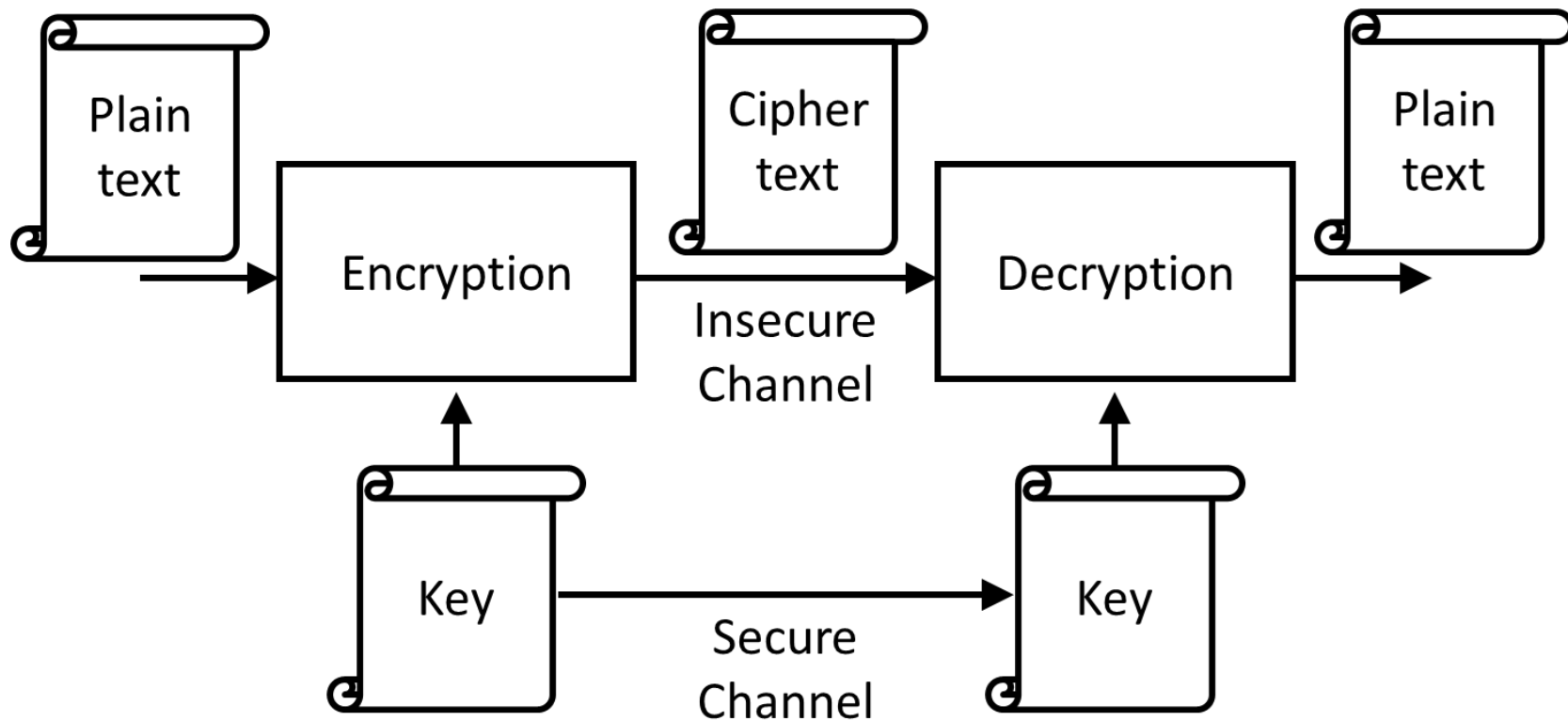


Remember **CIA**? Different **A** for Crypto Power ⚡

- **Confidentiality**: Prevent Eve **reading** Alice's messages
- **Integrity**: Prevent Mallory from **changing** Alice's messages (or at least Mallory's changes can be detected)
- **Authenticity**: Prevent Mallory from **impersonating** Alice



Cryptography - Path for Secret Messages



Historical Ciphers: Example One

FUBSWRJUDSKB

CRYPTOGRAPHY

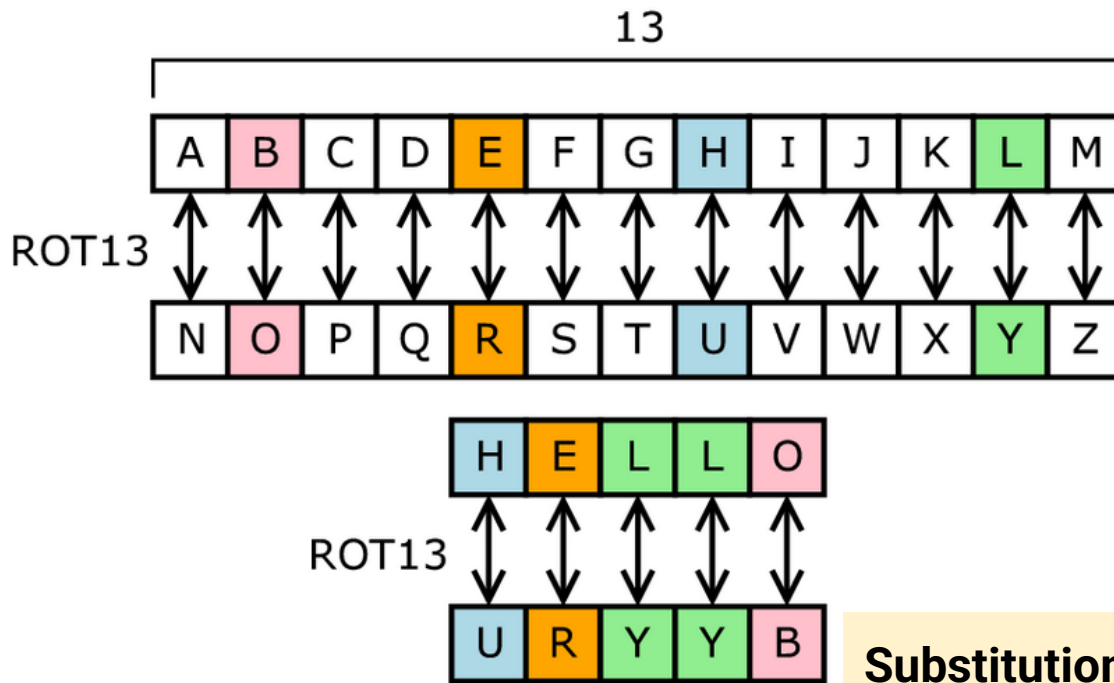
Historical Ciphers: Example One

FUBSWRJUDSKB

CRYPTOGRAPHY

**Substitution Cipher (shift by 3)
(monoalphabetic)**

Caesar Cipher



Substitution Cipher (shift by 13)

Shift and Substitution Ciphers

Replace symbols (letters) by others

- Using a shifting rule
 - e.g., $y = x + 13 \pmod{26}$
 - Key: length of shift
- Using a mapping table
 - e.g., A -> X, B -> F, C -> K,...
 - Key: table

Cryptanalysis - Analyzing “secret” messages

Mwahaha



We will learn the
secretsssss.

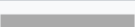
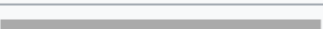
Historical Ciphers: Example Two

gsrh xlfihv rh zyfg xibkgltizksb uli gsv urihg gsivv dvvph. zmw
gsvm zkkorvw xibkgltizksb uli kirezxb zmw hvxfirgb lu wzgz.



English Frequency (first letter of an English word)

A	11.7%	
B	4.4%	
C	5.2%	
D	3.2%	
E	2.8%	
F	4%	
G	1.6%	
H	4.2%	
I	7.3%	
J	0.51%	
K	0.86%	
L	2.4%	
M	3.8%	

N	2.3%	
O	7.6%	
P	4.3%	
Q	0.22%	
R	2.8%	
S	6.7%	
T	16%	
U	1.2%	
V	0.82%	
W	5.5%	
X	0.045%	
Y	0.76%	
Z	0.045%	



Historical Ciphers: Example Two

gsrh xlfihv rh zyfg xibkgltizksb uli **gsv** urihg **gsivv** dvvph. zmw
gsvm zkkorvw xibkgltizksb uli kirezxb zmw hvxfirgb lu wzgz.





Historical Ciphers: Example Two

gsrh xlfihv rh zyfg xibkgltizksb uli **gsv** urihg **gsivv** dvvph. zmw
gsvm zkkorvw xibkgltizksb uli kirezxb zmw hvxfirgb lu wzgz.



This course is about cryptography for **the** first **three** weeks.
And **then** applied cryptography for privacy and security of
data.

Kerckhoffs's Principle

The security of a cryptosystem should solely depend on the secrecy of the key, but never on the secrecy of the algorithms.

Historical Ciphers: Example Three

LECTURE SECURITY AND CRYPTOGRAPHY I



LENGECDRCUCATRRPUIYHRTPYEYTISAO

Historical Ciphers: Example Four

LECTURES

ECURITYA

NDCRYPTO

GRAPHYI



LENGECDRCUCATRRPUIYHRTPYEYTISAO

Historical Ciphers: Example Four

LECTURES

SECURITYA

NDGCVYB

GRAP

Shannon's Maxim!!!! (design
assuming adversaries will
learn the algorithm)

VHRTPYEYTISAO

Shannon's Maxim & Kerckhoffs's Principle:

- Security shouldn't rely on the secrecy of the method
- Use public algorithms with secret “keys”
- The adversary's target is... **the key**

Idea: Easier to change a “short” key than your whole system.
(e.g., Recovery)

Unconditionally Secure: One-Time Pad

Message:

x_0	x_1	x_2
-------	-------	-------

 ...

x_n

$\oplus$

Key:

k_0	k_1	k_2
-------	-------	-------

 ...

k_n

=

Ciphertext:

y_0	y_1	y_2
-------	-------	-------

 ...

y_n

Rule: $y_i = x_i + k_i \pmod{2}$

Provable Security for One-Time Pad

<Ciphertext is uniformly distributed independent of the plaintext distribution>

$x_i = 0$ with probability p ($x_i = 1: 1-p$)

$k_i = 0$ with probability 0.5 ($k_i = 1: 0.5$)

$y_i = 0$ with probability:

$$\begin{aligned} p(y_i = 0) &= p(x_i = 0) p(k_i = 0) + p(x_i = 1) p(k_i = 1) \\ &= 0.5p + 0.5(1-p) \\ &= 0.5 \end{aligned}$$

Provable Security for One-Time Pad

Every ciphertext y can be decrypted **into every arbitrary plaintext** x using the **key** $k = x \oplus y$

Consequently the ciphertext cannot contain any information about the plaintext

Encryption is “deniable”



Well...this sucks for me...

What if it is a Many-Time Pad?

Key: K

Ciphertext₁ = message₁ $\oplus$ K = 2c1549100043130b1000290a1b

Ciphertext₂ = message₂ $\oplus$ K = 3f16421617175203114c020b1c



Hmmm... how can I relate these messages together?

What if it is a Many-Time Pad?

Key: K

$\text{Ciphertext}_1 \oplus \text{Ciphertext}_2 =$

$\text{message}_1 \oplus K \oplus \text{message}_2 \oplus K =$

$\text{message}_1 \oplus \text{message}_2 = 13030b0617544108014c2b0107$



What if it is a Many-Time Pad?

$\text{message}_1 \oplus \text{message}_2 = 13030b0617544108014c2b0107$

Suppose message_1 starts with “Alice” (414C696365)

- message₂ seems to start with readable text (“Rober”)



Is “Alice” here...?

What if it is a Many-Time Pad?

$\text{message}_1 \oplus \text{message}_2 = 13030b0617544108014c2b0107$

Suppose message_1 starts with “Alice” (416C696365)

- message₂ seems to start with readable text (“Rober”)

Suppose it starts with “Alice and Bob” (416C69636520616E6420426F62)

- message₂ is fully readable now! (“Robert feline”)



Ah!

Many-time pad? Messages Lack True Randomness



C_1



C_2



$C_1 \oplus C_2$



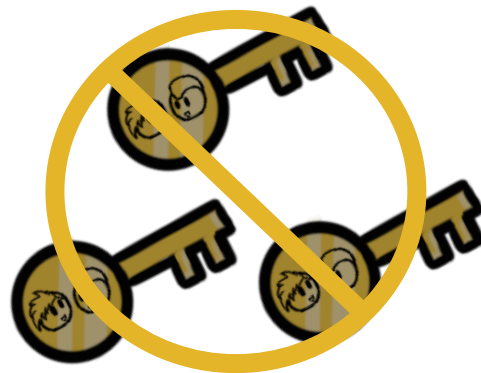
M_2



M_1

One-Time Pad - Conditions...

- Key as long as the message
- Key uniformly random
- Only used once



So...Cryptography?

- Simple substitution/transposition is computationally insecure
- One-Time Pad is inefficient over the secure channel

Goal: Securely communicate “a lot” of information on an insecure channel while requiring “limited” communication over a secure channel

Recap: A, B, C versus A and B and C

Substitution is insecure...

Transposition is insecure...

Key reuse using XOR (one-time pad) is insecure...

BUT

Repeat it often enough and it can be widely regarded as secure

Recap: A, B, C versus A and B and C

Substitution is insecure...

Transposition is insecure...

Key reuse

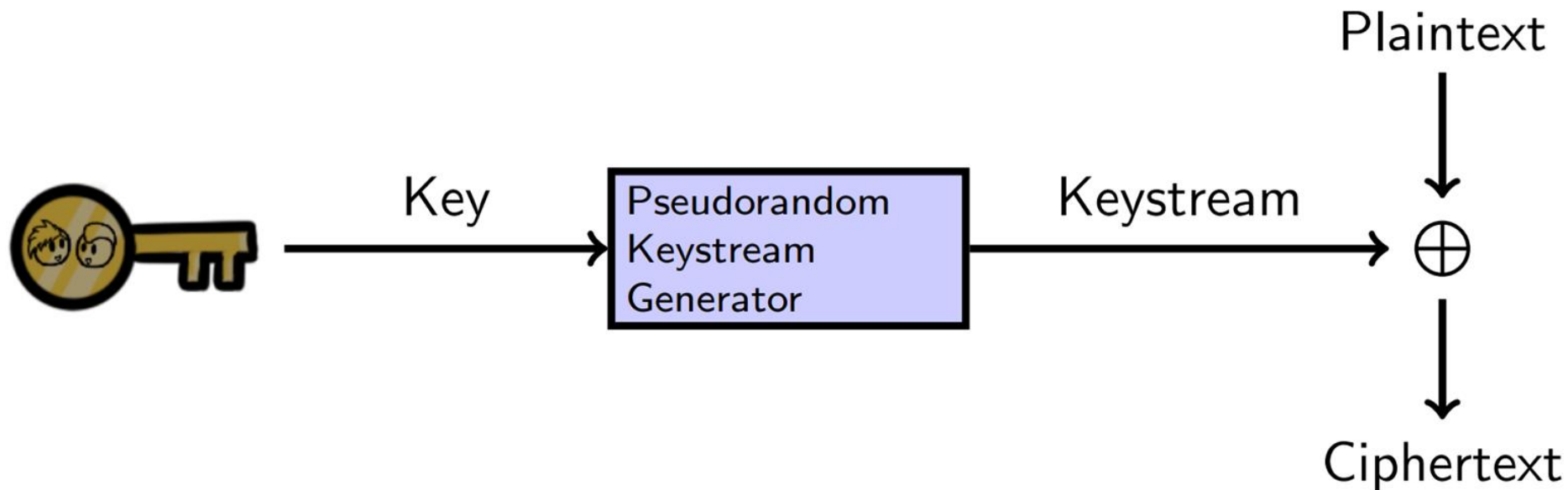
Stream Ciphers and Block
Ciphers

... is insecure...

BUT

Repeat it often enough and it can be widely regarded as
secure

Stream Cipher?

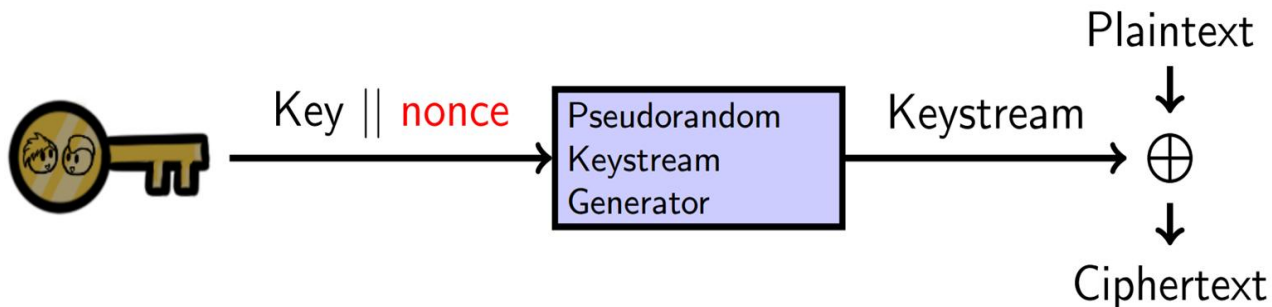


Fun(?) Facts:

- RC4 was the most common stream cipher on the Internet but deprecated.
- ChaCha increasingly popular (Chrome and Android), and SNOW3G in mobile phone networks.

Stream Ciphers Share Conditions with OTP

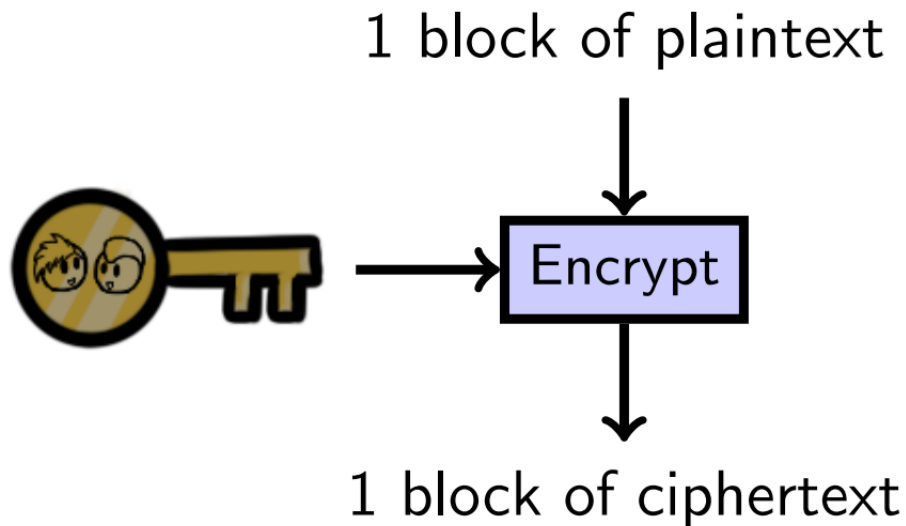
- Stream ciphers can be **very fast**
 - This is useful if you need to send a lot of data securely
- But they can be **tricky** to use correctly!
 - We saw the issues of re-using a key! (two-time pad)
 - **Solution:** concatenate key with nonce (**number used once**), which does not need to be secret



Fun(?) Facts:

- WEP, PPTP are great examples of how **not** to use stream ciphers

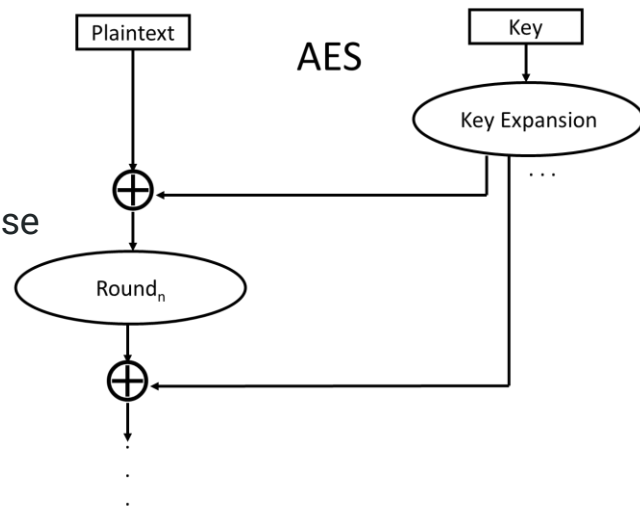
Bit by bit....do you have to?



Block ciphers!!!

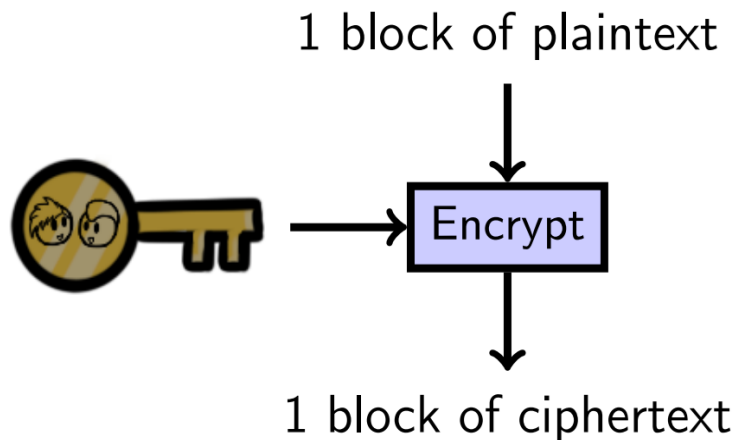
Block Ciphers

- Weakness of streams...one bit at a time?
 - What happens in a stream cipher if you change just one bit of the plaintext?
- Welcome, use of block ciphers
 - Block ciphers operate on the message one block at a time
 - Blocks are usually 64 or 128 bits long
- **AES**, the current standard
 - You better have a very...very good reason to choose otherwise

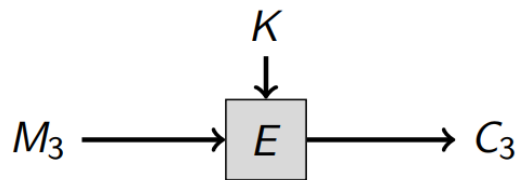
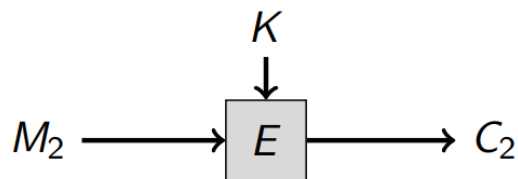
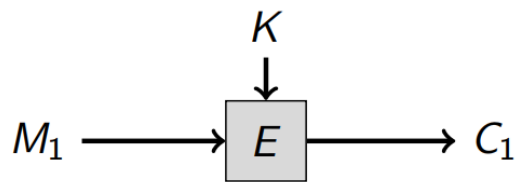


Two Catches with Block Ciphers

- Message is **shorter** than one block?
 - Requires padding
- Message is **longer** than a block?
 - Requires **modes of operation**



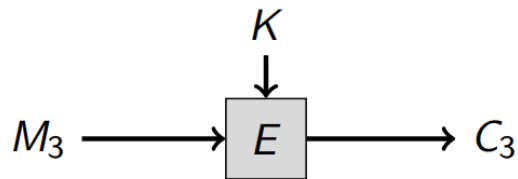
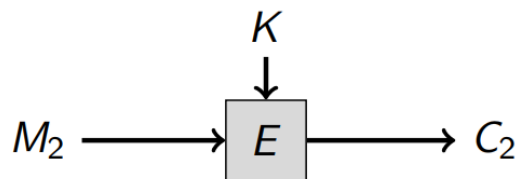
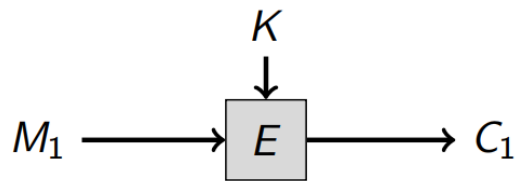
Block Ciphers and Modes of Operation: ECB Mode



$\vdots$ $\vdots$ $\vdots$

- ECB: Electronic Code Book
- Encrypts each successive block separately

Block Ciphers and Modes of Operation: ECB Mode

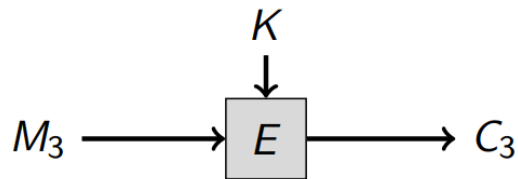
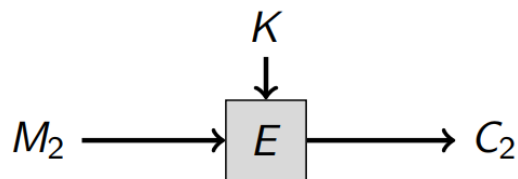
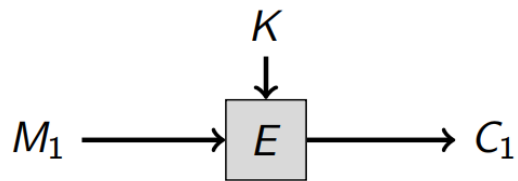


$\vdots$ $\vdots$ $\vdots$

- ECB: Electronic Code Book
- Encrypts each successive block separately

Q: What happens if the plaintext M has some blocks that are identical, $M_i = M_j$?

Block Ciphers and Modes of Operation: ECB Mode



$\vdots$ $\vdots$ $\vdots$

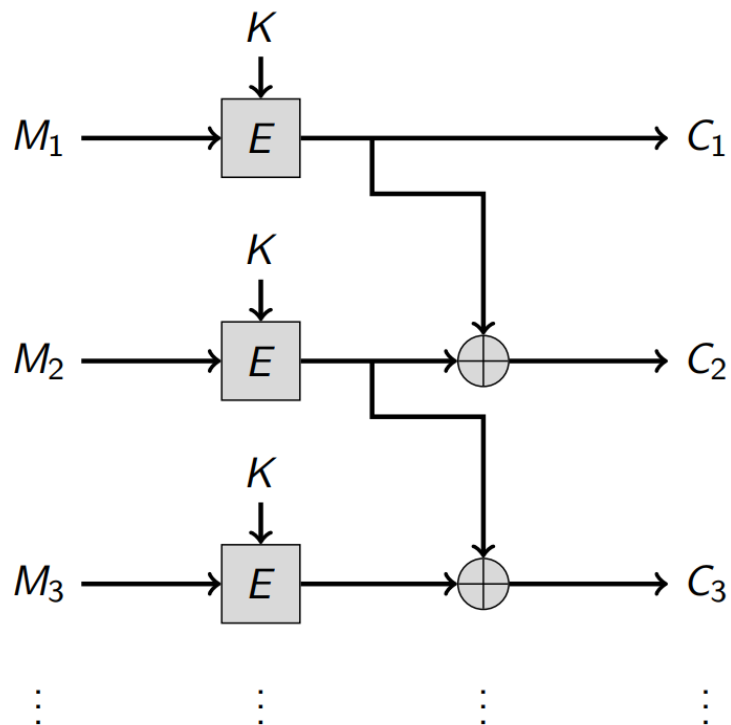
- ECB: Electronic Code Book
- Encrypts each successive block separately

Q: What happens if the plaintext M has some blocks that are identical, $M_i = M_j$?

A: $C_i = E_K(M_i), C_j = E_K(M_j) \Rightarrow C_i = C_j$



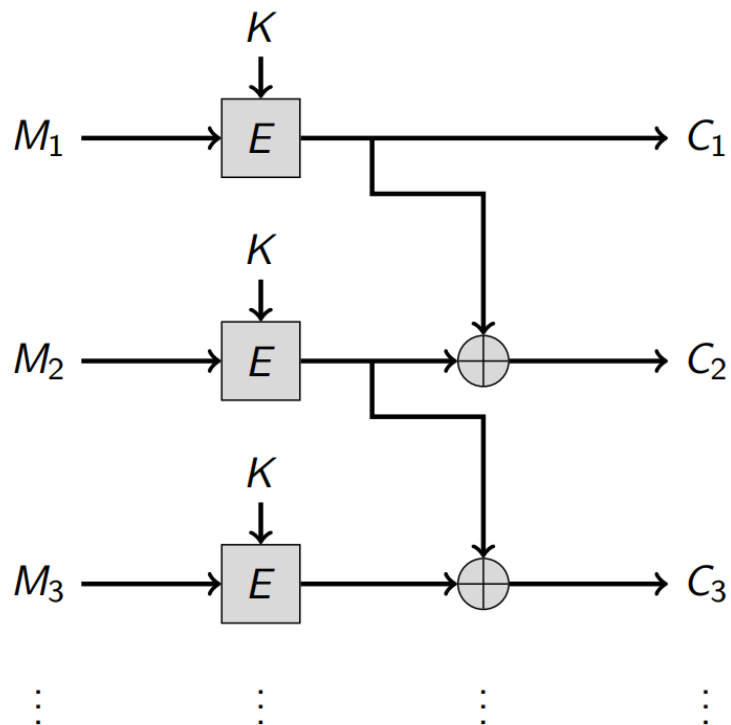
Attempt 1: Fixing ECB₁



- Provide “feedback” among different blocks, to avoid repeating patterns...

Q: Fix repeating patterns? Are there other issues?

Attempt 1: Fixing ECB₁

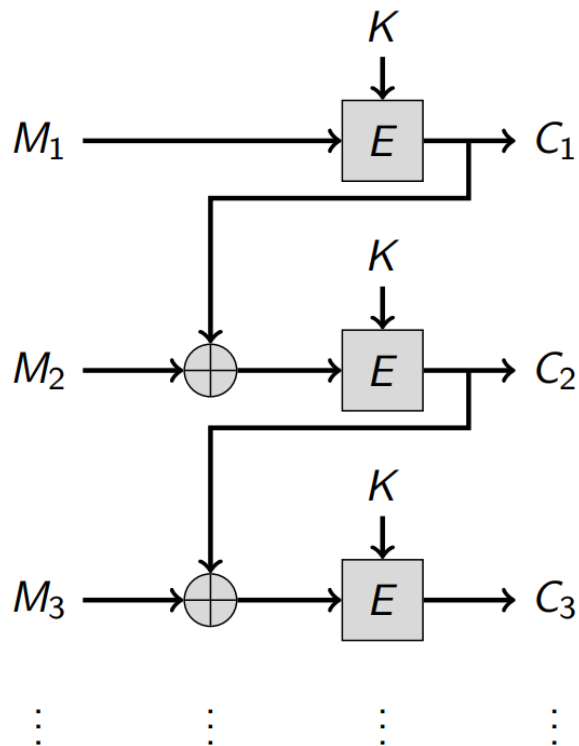


- Provide “feedback” among different blocks, to avoid repeating patterns...

Q: Fix repeating patterns? Are there other issues?

A: Yes. We can un-do the XOR if we get all the ciphertexts. This basically does not improve compared to ECB.

Attempt 2: ECB₂!!!

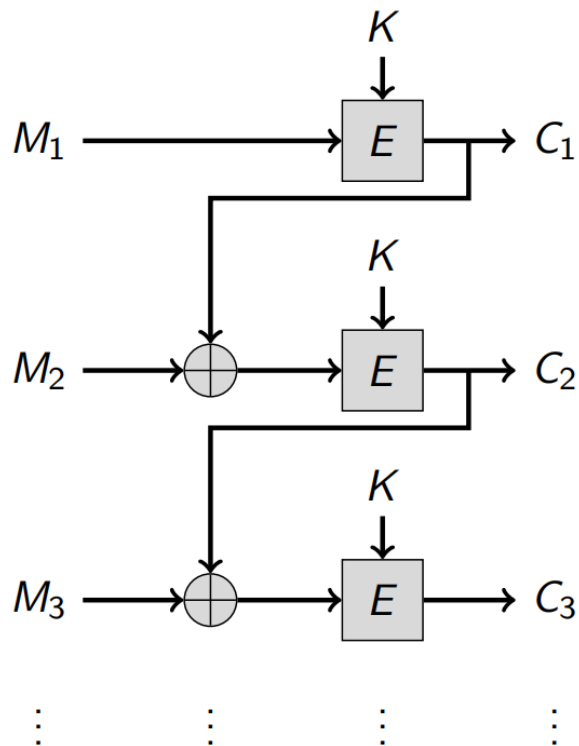


Q: Spot the difference?

Q: Is it fixed this time?

Q: Does this avoid repeating patterns among blocks?

Attempt 2: ECB₂!!!



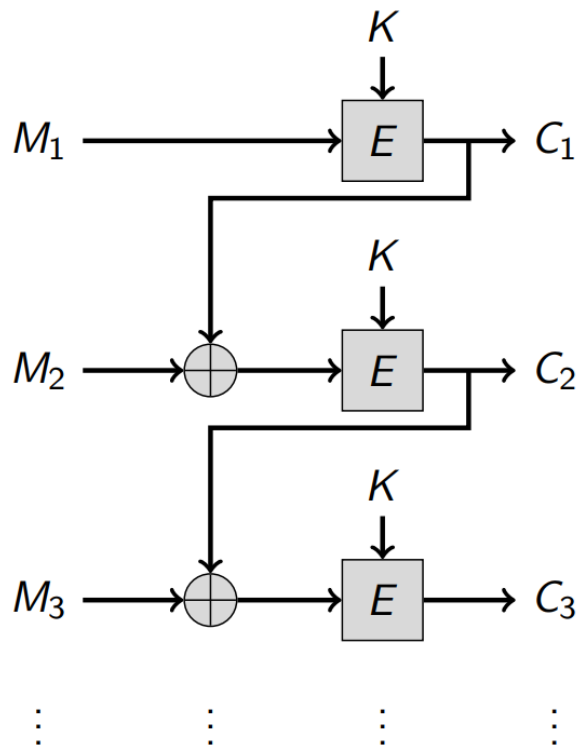
Q: Spot the difference?

Q: Is it fixed this time?

Q: Does this avoid repeating patterns among blocks?

Q: What would happen if we encrypt two equal messages with the same key?

Attempt 2: ECB₂!!!



Q: Spot the difference?

Q: Is it fixed this time?

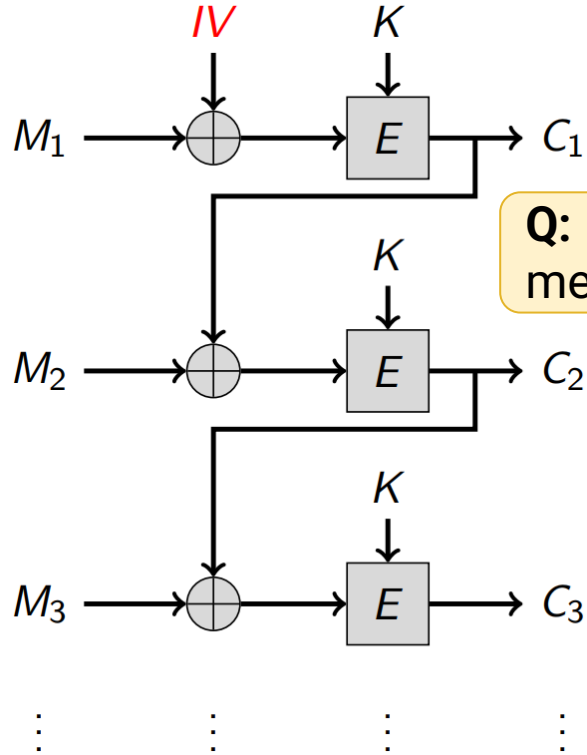
Q: Does this avoid repeating patterns among blocks?

Q: What would happen if we encrypt two equal messages with the same key?

A: For $M = N$,
 $C = E_K(M)$, $D = E_K(N) \Rightarrow C = D$



New Plan: Cipher Block Chaining (CBC) Mode

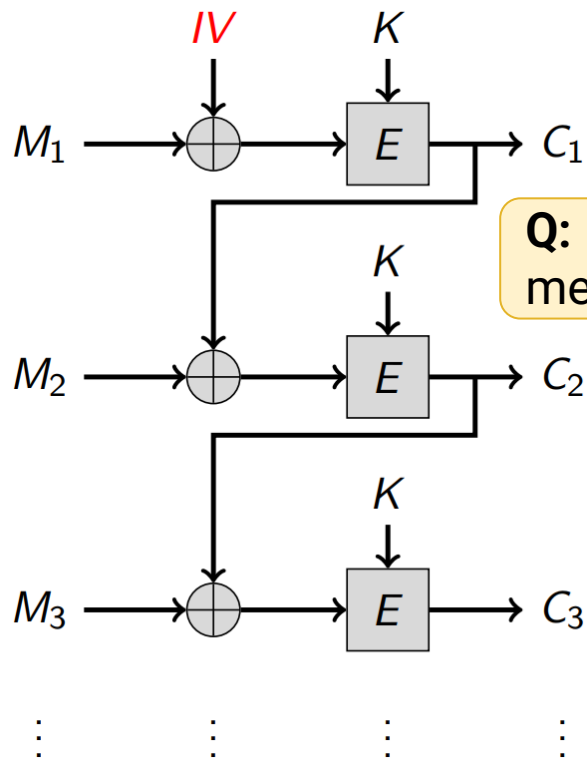


Q: Does this solve the issue of encrypting equal blocks?

Q: Does this solve the issue of encrypting equal messages/plaintexts?

IV, an initialization vector, nonce, salt (one for each message)

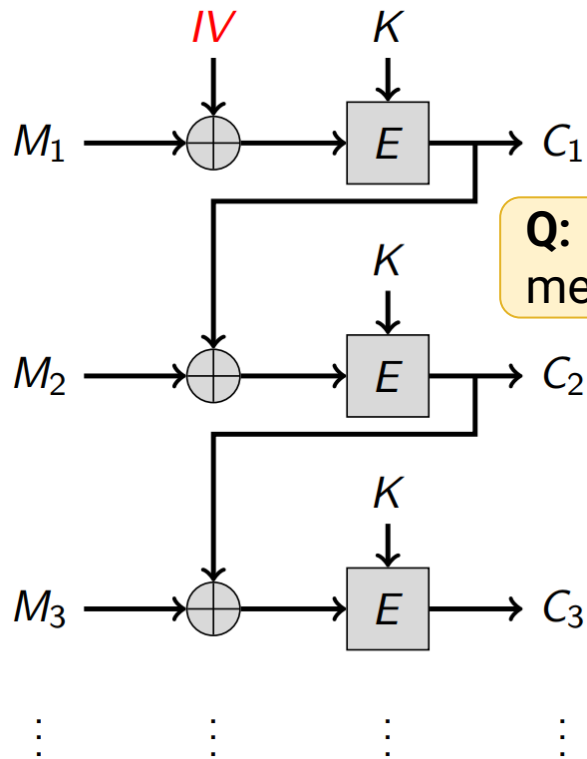
New Plan: CBC Mode



Q: Does this solve the issue of encrypting equal blocks?

Q: Does this solve the issue of encrypting equal messages/plaintexts?

New Plan: CBC Mode



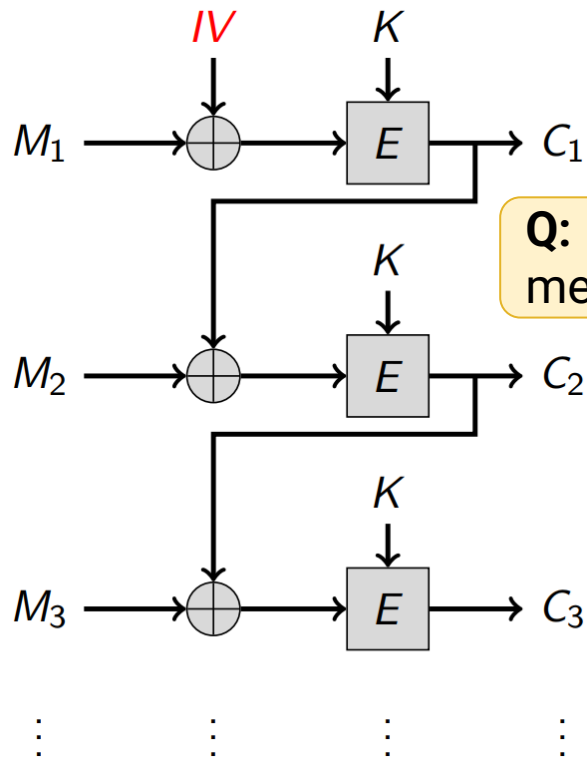
Q: Does this solve the issue of encrypting equal blocks?

Q: Does this solve the issue of encrypting equal messages/plaintexts?

A: Yes!!!



New Plan: CBC Mode



Q: Does this solve the issue of encrypting equal blocks?

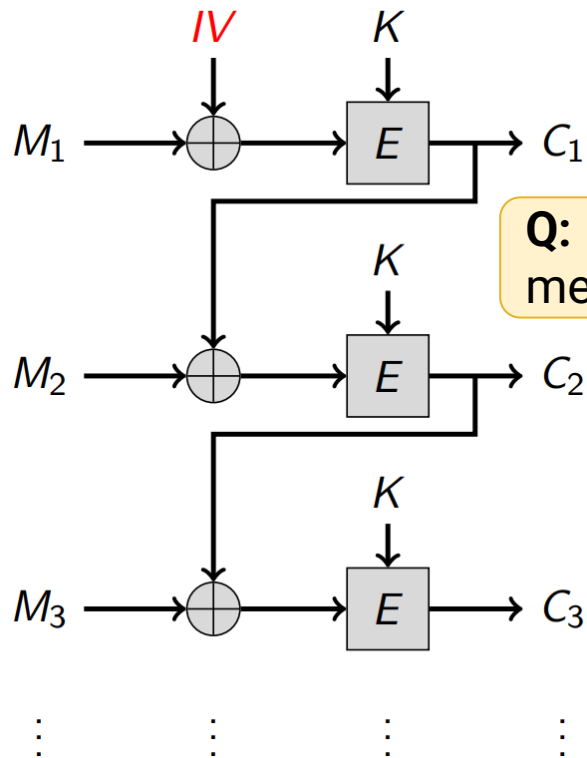
Q: Does this solve the issue of encrypting equal messages/plaintexts?

A: Yes!!!



Q: Can we share IV in the clear?

New Plan: CBC Mode



Q: Does this solve the issue of encrypting equal blocks?

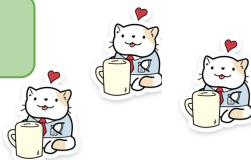
Q: Does this solve the issue of encrypting equal messages/plaintexts?

A: Yes!!!



Q: Can we share IV in the clear?

A: Yes!!!



Modes of Operation Collection

- Cipher Block Chaining (**CBC**), Counter (**CTR**), and Galois Counter (**GCM**) modes
- Patterns in the plaintext are no longer exposed because these modes involve some kind of “feedback” among different blocks.
- But you need an **IV**



So...now what?

- How do Alice and Bob share the secret key?
 - Meet in person; diplomatic courier...
- In general this is very hard

Or, we invent new technology!!

Spoiler Alert: it's already been invented...

Stay tuned!

Until next time...
