

Cryptography

INTRODUCTION

Why study cryptography?

Secure communication:

- Web
- Wireless

Encrypting files:

- File System
- Archives

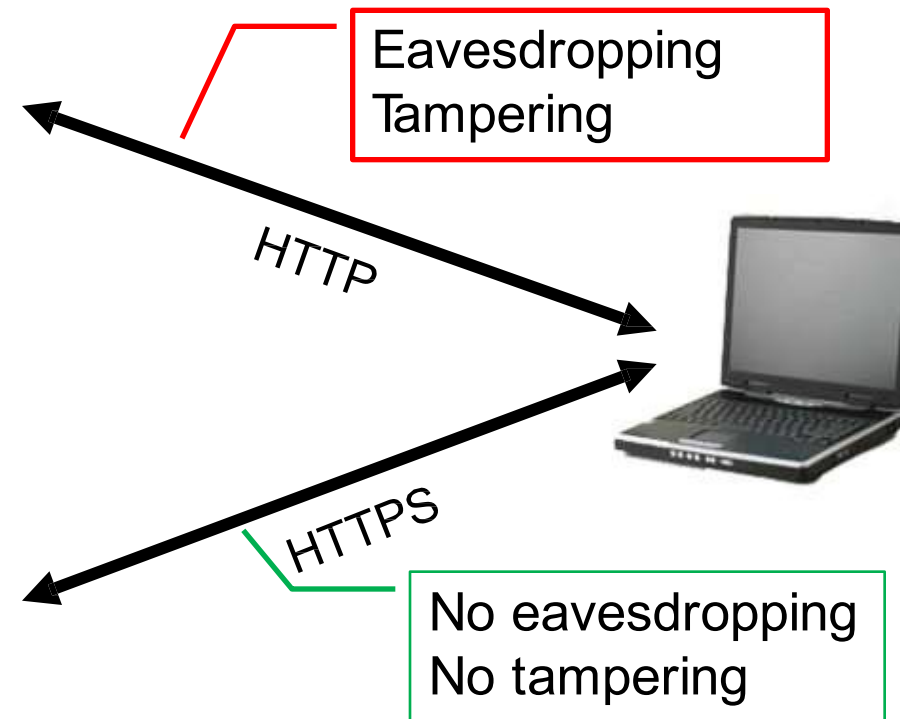
Content protection:

- CD, DVD, Blu-ray

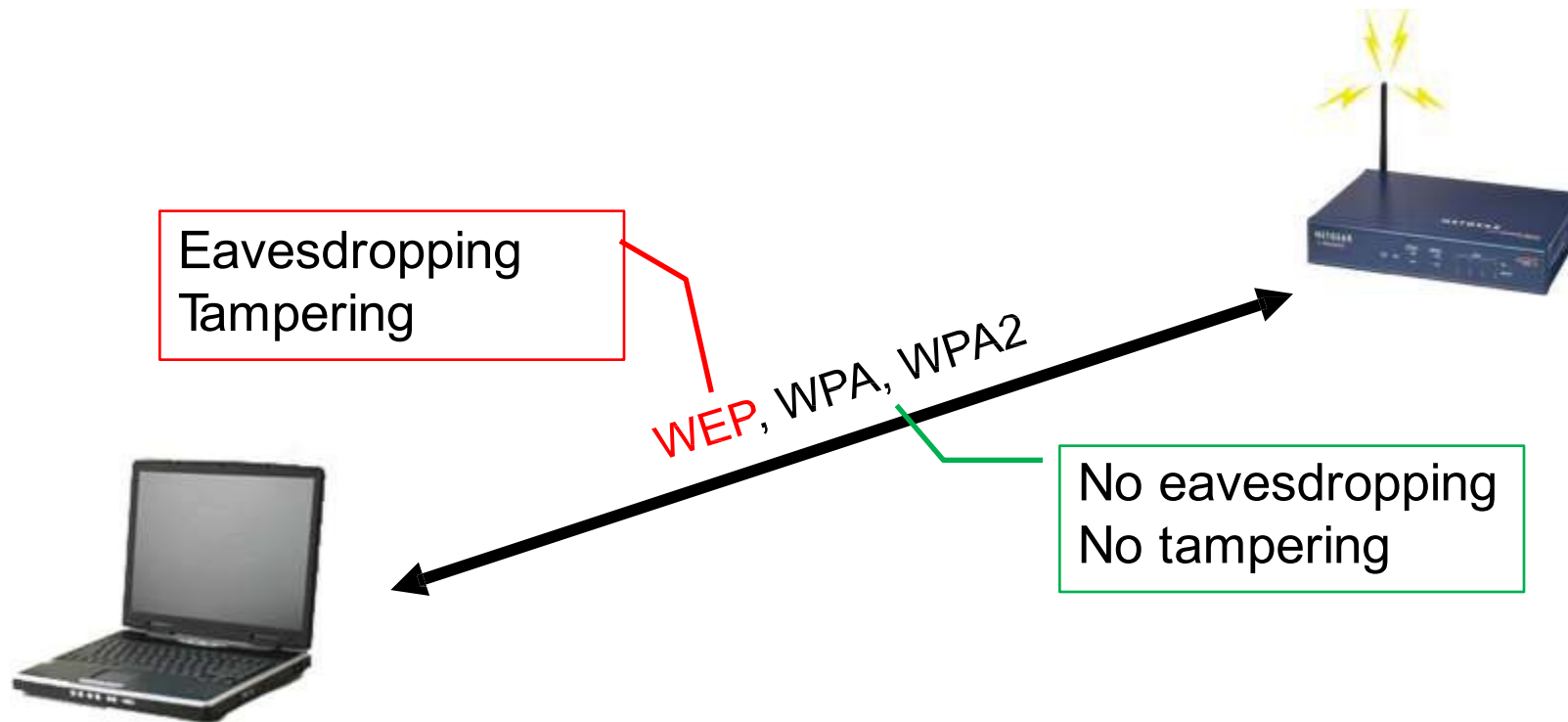
User authentication:

- Digital Identity, Login

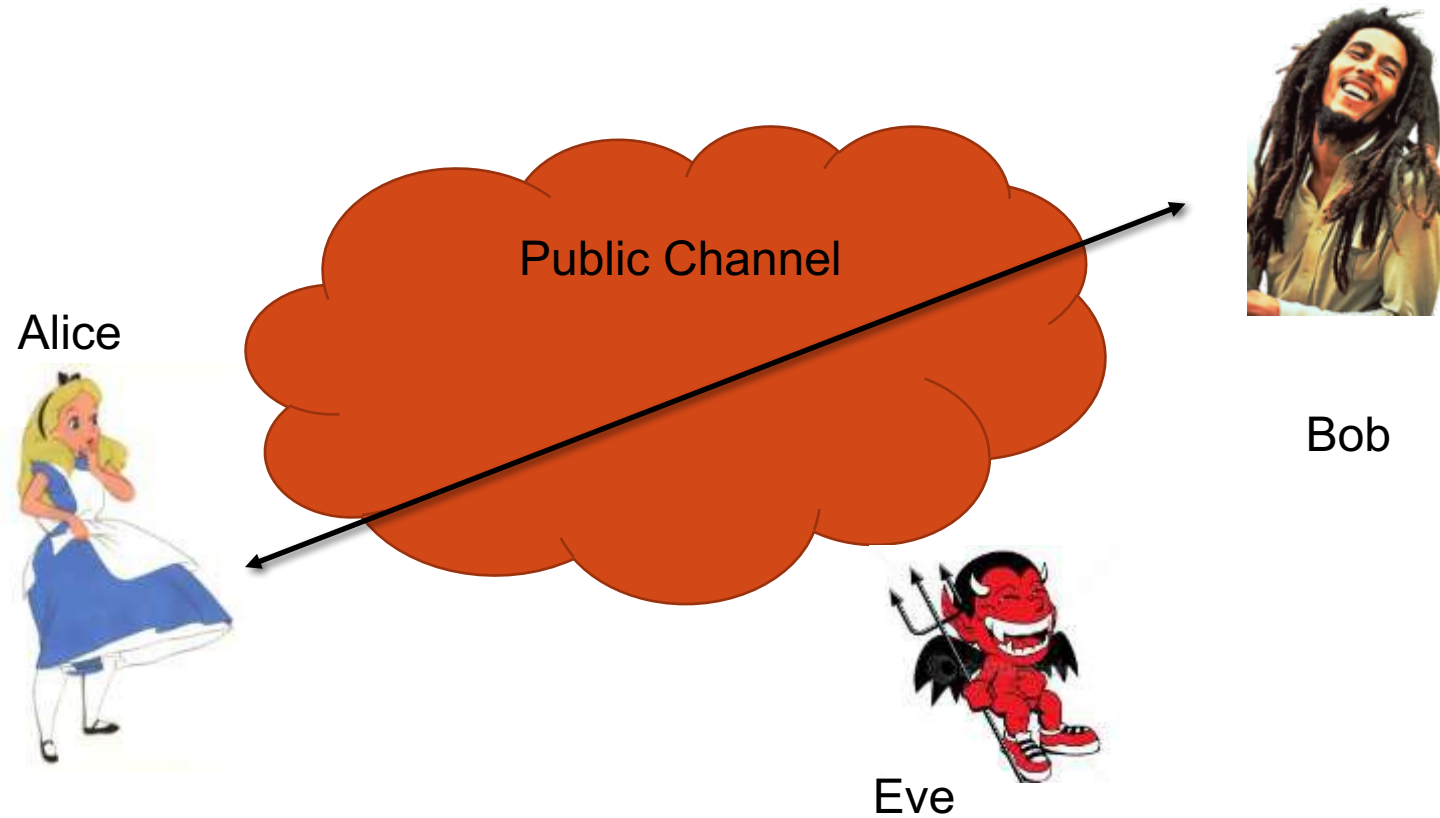
Example: Secure Web Traffic



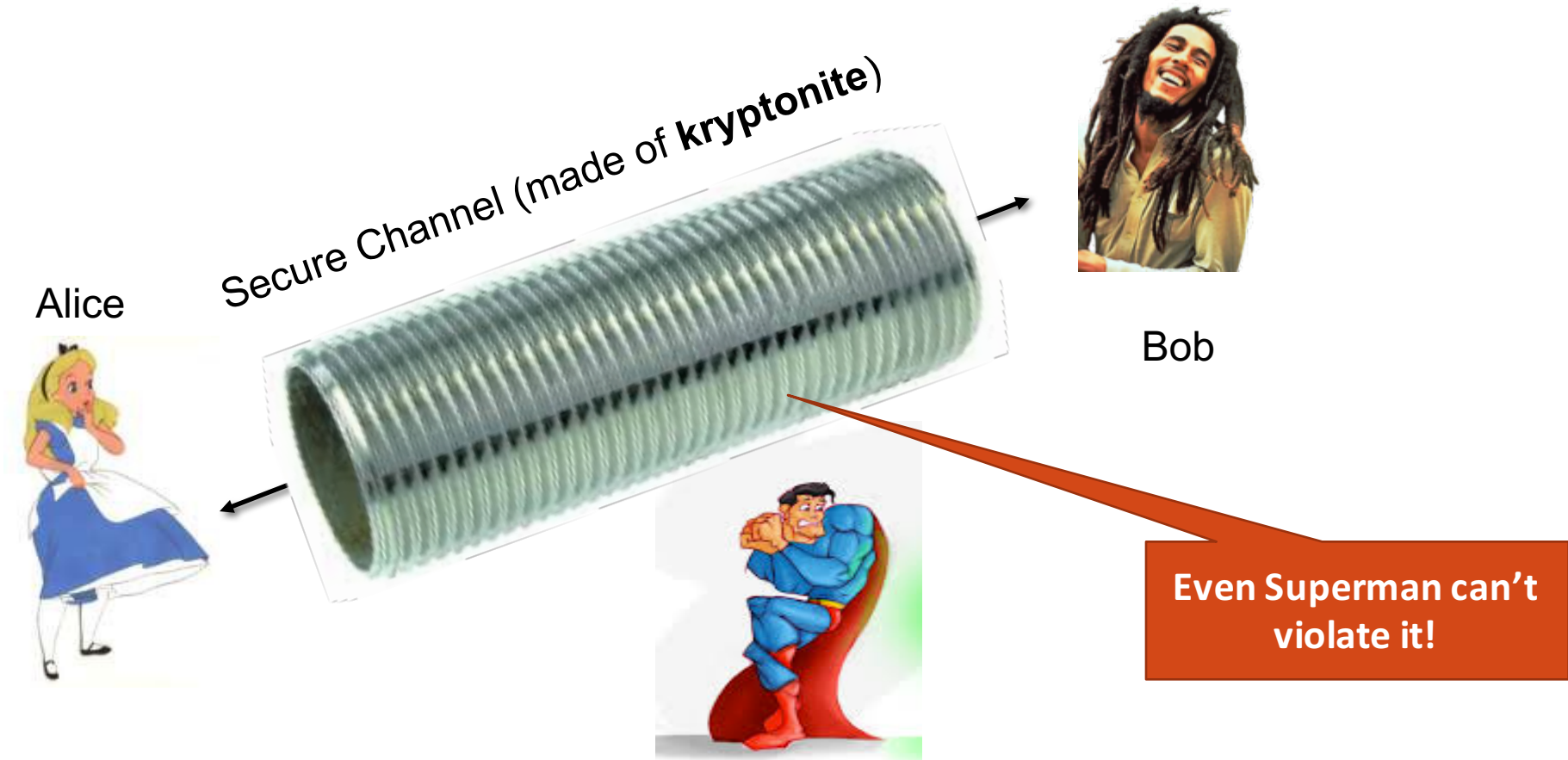
Example: Secure Wireless Traffic



Cryptography Actors



Cryptography Actors: Secure Channel



What can cryptography do?

Cryptography is an incredibly powerful tool

- Fundamental building block for security

Adding cryptography makes things secure?

- Not True!!!

NOT the solution to all security problems

- Buffer overflow, Social Engineering, Malware

NOT Something you should try to invent yourself

- Many examples of broken ad-hoc designs
- Reliable unless implemented and used properly

Cryptography

WHAT CRYPTOGRAPHY OFFERS?

Security Goals: Confidentiality

I don't want others to see my emails or chats

- Get my social-security number, credit-card number or medical records
- Know which web sites I visit, what I buy, where I travel
- Know my salary, how I vote, what movies I like, whether I sing in the shower

Real World Examples

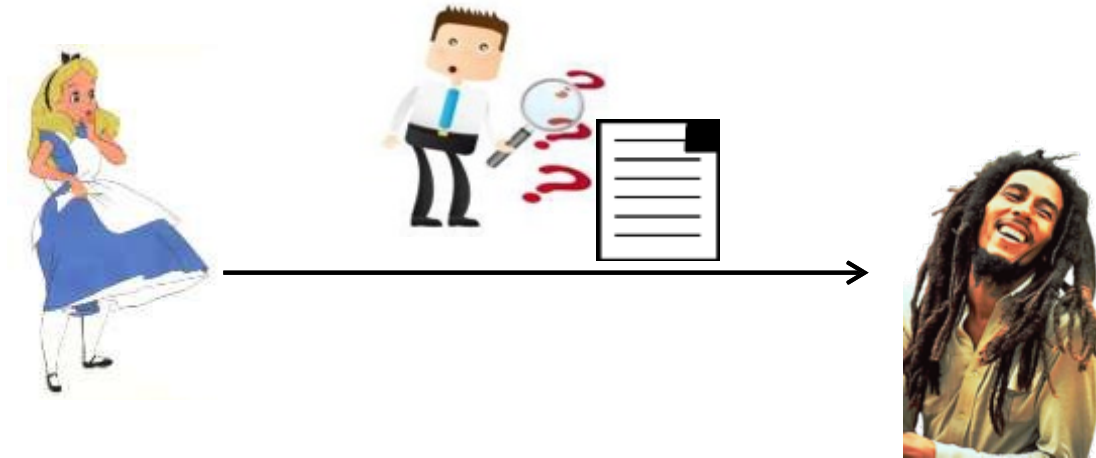
- Coca-Cola does not want its formula revealed
- Corporations want to protect their technology
- Governments want their plans kept secret
- Political dissidents want their identities to be secret
- ...



Security Goals: Confidentiality

Prevents unauthorized from accessing information

- *“Can unauthorized understand messages?”*
- Encryption removes meanings from information



Security Goals: Integrity

I don't want the emails or chats I send or receive to be modified or faked

I don't want my allergy information to be erased from my medical record.

I don't want my accounts to be broken into.

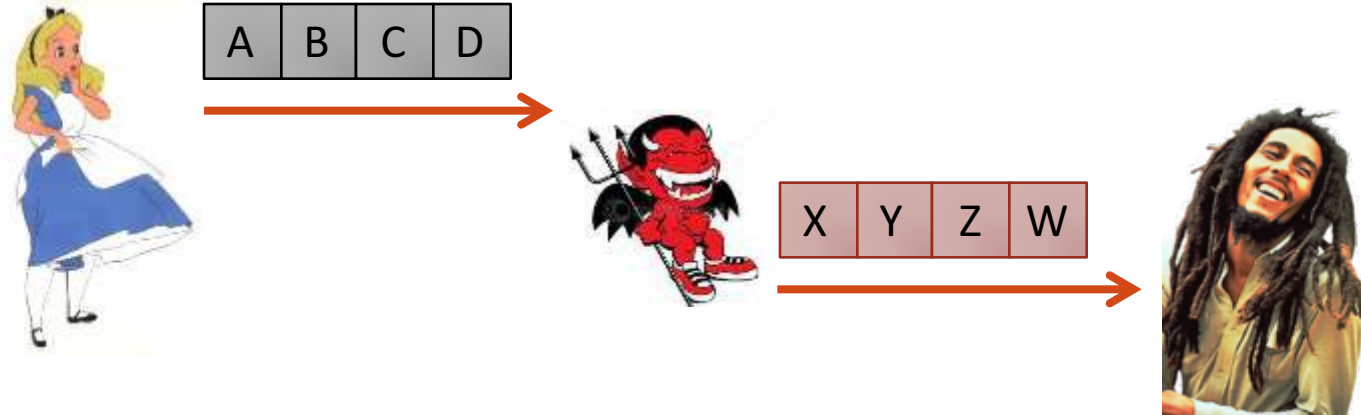
I don't want the data I communicate to my bank to be modified.

Servers don't want to be hacked into. Companies want to control access to their databases.

Security Goals: Integrity

Data cannot be modified without detection

- *“Is the received message the original sent?”*
- Hash functions make it extremely difficult to change information



Security Goals: Authentication

I want to be sure that entities I interact with are who they claim to be

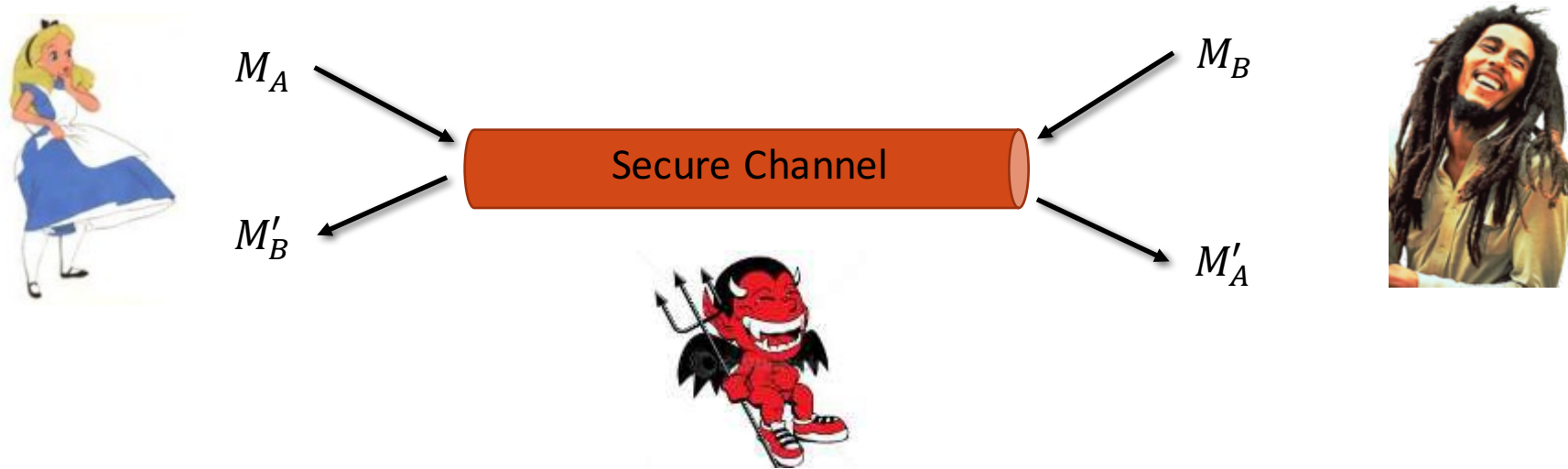
- whether it be my friend Alice
- my doctor
- google

Security Goals: Authentication

- A message can be created only by a particular party
 - “*Am I really talking to who I expect?*”
 - Key only know by *allowed* parties



Mix everything: Secure Channel



Confidentiality: Adversary does not learn anything about M_A, M_B

Integrity: $M'_A = M_A$ and $M'_B = M_B$

Mix everything: Secure Channel



Confidentiality: Adversary does not learn anything about M_A, M_B

Integrity: $M'_A = M_A$ and $M'_B = M_B$

Authentication: Alice is really "Alice" and Bob is really "Bob"



Many crypto tools...

Secure key establishment

Secure communication

Digital signatures

Anonymous communication

- Anonymous payments
- Anonymous e-voting
- Private queries

Secure computation

Much much more...

Classical Ciphers

SUBSTITUTION CIPHERS

Caesar Cipher

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	A	B	C
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

T	H	I	S		I	S		T	H	E		M	E	S	S	A	G	E
---	---	---	---	--	---	---	--	---	---	---	--	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

The key k is the offset that shifts the alphabet

- Encrypt using
 - $C_i = (M_i + K) \bmod 26$
- Decrypt using
 - $M_i = (C_i - K) \bmod 26$



Caesar Cipher

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	A	B	C
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

T	H	I	S		I	S		T	H	E		M	E	S	S	A	G	E
---	---	---	---	--	---	---	--	---	---	---	--	---	---	---	---	---	---	---

A																			
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

The key k is the offset that shifts the alphabet

- Encrypt using
 - $C_i = (M_i + K) \bmod 26$
- Decrypt using
 - $M_i = (C_i - K) \bmod 26$



Caesar Cipher

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	A	B	C
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

T	H	I	S		I	S		T	H	E		M	E	S	S	A	G	E
---	---	---	---	--	---	---	--	---	---	---	--	---	---	---	---	---	---	---

A	M																		
---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

The key k is the offset that shifts the alphabet

- Encrypt using
 - $C_i = (M_i + K) \bmod 26$
- Decrypt using
 - $M_i = (C_i - K) \bmod 26$



Caesar Cipher

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	A	B	C
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

T	H	I	S		I	S		T	H	E		M	E	S	S	A	G	E
---	---	---	---	--	---	---	--	---	---	---	--	---	---	---	---	---	---	---

A	M	N																	
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

The key k is the offset that shifts the alphabet

- Encrypt using
 - $C_i = (M_i + K) \bmod 26$
- Decrypt using
 - $M_i = (C_i - K) \bmod 26$



Caesar Cipher

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	A	B	C
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

T	H	I	S		I	S		T	H	E		M	E	S	S	A	G	E
---	---	---	---	--	---	---	--	---	---	---	--	---	---	---	---	---	---	---

A	M	N	V		N	V		A	M	H		P	H	V	V	D	L	H
---	---	---	---	--	---	---	--	---	---	---	--	---	---	---	---	---	---	---

The key k is the offset that shifts the alphabet (Caesar used $K=3$)

- Encrypt using
 - $C_i = (M_i + K) \bmod 26$
- Decrypt using
 - $M_i = (C_i - K) \bmod 26$



Substitution Cipher

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

U	C	H	G	F	P	L	M	Q	V	I	O	R	A	T	D	N	S	B	E
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

T	H	I	S		I	S		T	H	E		M	E	S	S	A	G	E
---	---	---	---	--	---	---	--	---	---	---	--	---	---	---	---	---	---	---

S	M	Q	N		Q	N		S	M	F		I	F	N	N	U	L	F
---	---	---	---	--	---	---	--	---	---	---	--	---	---	---	---	---	---	---

The key k_i is the letter in the scrambled alphabet

- Encrypt using
 - $C_i = (M_i \leftarrow K_i)$
- Decrypt using
 - $M_i = (C_i \leftarrow \mathcal{A}_i)$

Cryptanalysis of Monoalphabetic

Easily breakable by knowing the cipher algorithm

- Worked well back in the day
 - High illiteracy rate
 - Lack of algorithms knowledge

How to break it?

- Brute force
 - Try every possible key (*Always an option*)
 - For a 26-letter alphabet, only 26 possible values for k
 - That's so simple!
- Frequency analysis

Frequency Analysis

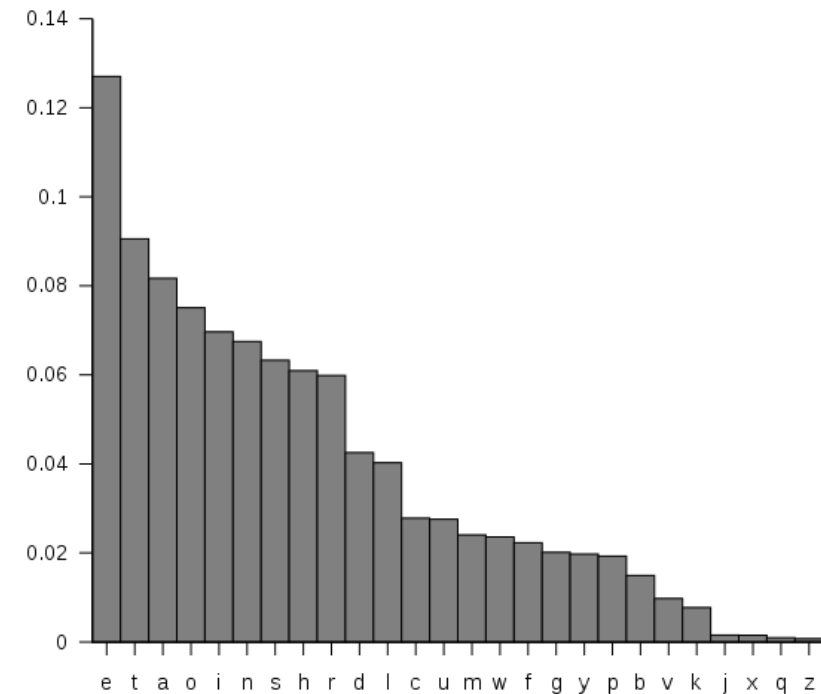
In every language

- Symbols occur with different probabilities

Frequency analysis

- Looks at how often each is seen in a sample
- Match frequency in ciphertext to frequency in plaintext
- Gives a short list of possible mappings

Italian		English	
E	11,79%	E	12,31%
A	11,74%	A	9,59%
I	11,28%	I	8,05%
O	9,83%	O	7,94%
N	6,88%	N	7,19%



Polyalphabetic Ciphers

Monoalphabetic cipher

- Applies the same key to every symbol
- So simple to break!

Polyalphabetic cipher

- Switches between a set of keys
- Harder to break...

We'll look at the Vigenère Cipher

- Symbols are changed exact same way as Caesar's cipher
- Difference is that there are multiple key symbols

Vigenère Cipher

N	E	T	S	E	C	C	L	A	S	S
L	E	A	R	N						

Polyalphabetic Cipher

- Plaintext from column
- Key from row
- Encrypt using
 - $C_i = (M_i + K_i \bmod \text{len}(K))$
- Decrypt using
 - $M_i = (C_i - K_i \bmod \text{len}(K))$

	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

Vigenère Cipher

N	E	T	S	E	C	C	L	A	S	S
L	E	A	R	N	L	E	A	R	N	L

Polyalphabetic Cipher

- Plaintext from column
- Key from row
- Encrypt using
 - $C_i = (M_i + K_i \bmod \text{len}(K))$
- Decrypt using
 - $M_i = (C_i - K_i \bmod \text{len}(K))$

	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

Vigenère Cipher

N	E	T	S	E	C	C	L	A	S	S
L	E	A	R	N	L	E	A	R	N	L
Y										

Polyalphabetic Cipher

- Plaintext from column
- Key from row
- Encrypt using
 - $C_i = (M_i + K_i \bmod \text{len}(K))$
- Decrypt using
 - $M_i = (C_i - K_i \bmod \text{len}(K))$

	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

Vigenère Cipher

N	E	T	S	E	C	C	L	A	S	S
L	E	A	R	N	L	E	A	R	N	L
Y	I									

Polyalphabetic Cipher

- Plaintext from column
- Key from row
- Encrypt using
 - $C_i = (M_i + K_i \bmod \text{len}(K))$
- Decrypt using
 - $M_i = (C_i - K_i \bmod \text{len}(K))$

	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

Vigenère Cipher

N	E	T	S	E	C	C	L	A	S	S
L	E	A	R	N	L	E	A	R	N	L
Y	I	T								

Polyalphabetic Cipher

- Plaintext from column
- Key from row
- Encrypt using
 - $C_i = (M_i + K_i \bmod \text{len}(K))$
- Decrypt using
 - $M_i = (C_i - K_i \bmod \text{len}(K))$

	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

Vigenère Cipher

N	E	T	S	E	C	C	L	A	S	S
L	E	A	R	N	L	E	A	R	N	L
Y	I	T	J	R	N	G	L	R	F	D

Polyalphabetic Cipher

- Plaintext from column
- Key from row
- Encrypt using
 - $C_i = (M_i + K_i \bmod \text{len}(K))$
- Decrypt using
 - $M_i = (C_i - K_i \bmod \text{len}(K))$

	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

Vigenère Cipher

Can condense symbols into a simple expression

- $C[i] = P[i] + K[i \bmod \text{len}(k)]$
- The second part selects the correct key symbol to use
 - If you do this, watch out for the spaces
 - If they aren't in the alphabet, they should be ignored

Spaces may preserve plaintext word length

- or may occur at fixed intervals to obscure word length

Cryptanalysis of Vigenère Cipher

Figure out the key length, n

- Look for patterns
 - Common words are likely to be encrypted multiple times if the text is long enough
 - “the” is very common in English
 - If there are at least n occurrences of “the” in the plaintext, we can expect at least 2 to have identical ciphertext
- When you find two words of ciphertext that you believe to encrypt the same ciphertext
 - Find the difference in position, d
 - Assume that $n|d$ (n divides d)
 - Repeat and narrow in on n by looking for common factors

Cryptanalysis of Vigenère Cipher

KKALC LGQLC CREFC KVMPW BSURR ZUZMH PWZJO ZFHIF
FBMAV VFQAS COKSI IGOIB VTDSA RBOMS EHSVI UUQFF
VOWXC ESIQI KWZCK YSDIQ ZJUPP CCAHA RYQWQ ZJUPV
RBPWI EQXIO ETDSA WCDXV KVQJO KOXPC ZBEST KVQWS
KKAJC VGMT0 ZFAJG KODGF FGEHZ FJQVG KOWIH YSUVZ

1. Spaces occur at fixed intervals
2. Look for any repeated groupings

Cryptanalysis of Vigenère Cipher

Distance: 160

KKALC	LGQLC	CREFC	KVMPW	BSURR	ZUZMH	PWZJO	ZFHIF
FBMAV	VFQAS	COKSI	IGOIB	VTDSA	RBOMS	EHSVI	UUQFF
VOWXC	ESIQI	KWZCK	YSDIQ	ZJUPP	CCAHA	RYQWQ	ZJUPV
RBPWI	EQXIO	ETDSA	WCDXV	KVQJO	KOXPC	ZBEST	KVQWS
KKAJC	VGMTQ	ZFAJG	KODGF	FGEHZ	FJQVG	KOWIH	YSUVZ

1. Spaces occur at fixed intervals
2. Look for any repeated groupings
 - KKA (0,160)

Cryptanalysis of Vigenère Cipher

KKALC LGQLC CREFC KVMPW BSURR ZUZMH PWZJO ZFHIF
FBMAV VFQAS COKSI IGOIB VTDSA RBOMS EHSVI UUQFF
VOWXC ESIQI KWZCK YSDIQ ZJUPP CCAHA RYQWQ ZJUPV
RBPWI EQXIO ETDSA WCDXV KVQJO KOXPC ZBEST KVQWS
KKAJC VGMTO ZFAJG KODGF FGEHZ FJQVG KOWIH YSUVZ

1. Spaces occur at fixed intervals
2. Look for any repeated groupings
 - KKA (0,160)
 - OZF (34,169)
 - TDSA (61,131)
 - QZJUP (99,114)
 - KVQ (140,155)
 - GKO (174,189)

Cryptanalysis of Vigenère Cipher

1. Find the differences between pairs and factor
 - $160 - 0 = 160 = 2^5 * 5$
 - $169 - 34 = 135 = 3^3 * 5$
 - $131 - 61 = 70 = 2 * 5 * 7$
 - $114 - 99 = 15$ $155 - 140 = 15$ $189 - 174 = 15 = 3 * 5$
2. Identify common factors
 - They all have 5 as a factor
 - Since 5 is a prime and the key has an integer length
 - We know $n = 5$
 - If the only factor is composite, it may be the key length or a multiple of the key length
3. Split the ciphertext by key character
4. Now perform frequency analysis!

Kerckhoffs' Principle

“A cryptosystem should be secure even if the attacker knows all details about the system, with the exception of the secret key. In particular, the system should be secure when the attacker knows the encryption and decryption algorithms.”

Encryption and Decryption algorithm

- *Can be public?*

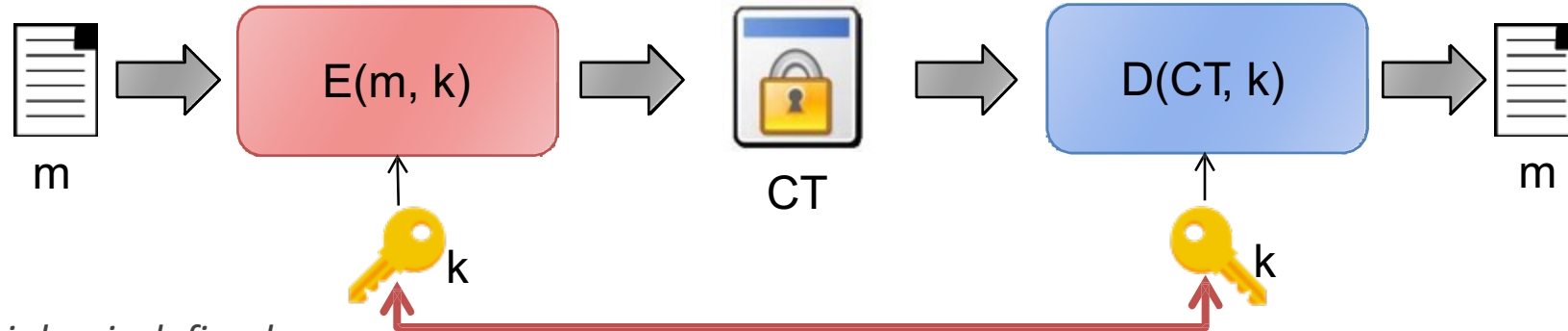
Security only depends on the secrecy of the key?

Security by obscurity cannot work!

Symmetric Cryptography

VERNAM CIPHER (ONE-TIME PAD)

Symmetric Cipher



A symmetric cipher is defined as

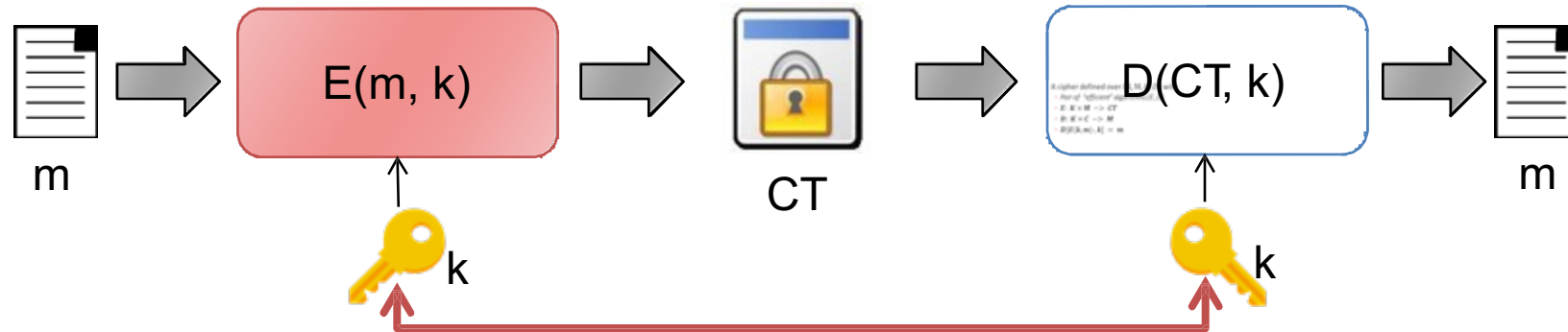
- $E(\cdot, \cdot) \rightarrow$ Encryption Algorithm
- $D(\cdot, \cdot) \rightarrow$ Decryption Algorithm
- $K \rightarrow$ Secret Key

We have two types of messages

- $M \rightarrow$ Plaintext (original message)
- $CT \rightarrow$ Ciphertext (encrypted message)

Common key and common cipher!

Symmetric Cipher



A cipher defined over (K, M, E, D) with

- Pair of “efficient” algorithms (E, D)
- $E: K \times M \rightarrow CT$
- $D: K \times C \rightarrow M$
- $D[E(k, m), k] = m$

Use cases

Single-use key

- *Key is only used to encrypt once*
 - e.g. Encrypted message: new key generated for each
 - e.g. Encrypted archive: different password for each
- *Only need key-agreement/transfer*

Multi-use key

- *Key used to encrypt many times*
 - e.g. Encrypted files: same key used to encrypt many
 - e.g. Encrypted communication: same key to encrypt messages
- *Needs more tricky functions*

Symmetric ciphers

Symmetric algorithms fall into two categories

- Block ciphers
- Stream ciphers

Used for different purposes

Both must provide confusion and diffusion

- Confusion: *relationship between key and ciphertext is obscured*
- Diffusion: *the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext*
 - Adversary must do more work to find statistical properties

Confusion/Diffusion in Caesar Cipher

Caesar cipher provides confusion and diffusion?

Confusion/Diffusion in Caesar Cipher

Caesar cipher provides confusion and diffusion?

Confusion → YES

- Yes, no relationship between key and ciphertext

Diffusion → NO

- Changing one symbol in the plaintext has a very predictable result
- Only changes one symbol in the output

XOR Operator

The XOR between $x, y \in \{0,1\}^n$

- Is the bit-wise add module 2: $(x + y) \bmod 2$

Has interesting properties for crypto

- Given x r.v. in $\{0,1\}^n$
- Given y uniform r.v. in $\{0,1\}^n$
- The r.v. $z = x \oplus y$ is uniform in $\{0,1\}^n$

x	y	$x \oplus y$		x	y		x	y	$x \oplus y$
0	0	0	0	P_0	$1/2$		0	0	$P_0/2$
0	1	1	1	P_1	$1/2$		0	1	$P_0/2$
1	0	1					1	0	$P_1/2$
1	1	0					1	1	$P_1/2$

Vernam Cipher / One Time Pad

First example of secure cipher

- Given a message $M = \{0,1\}^n$
- Given a secret key $K = \{0,1\}^n$

We define the symmetric cipher

- $CT := E(K, M) = K \oplus M$
- $D(K, CT) = K \oplus CT$
- $D(K, CT) = D[k, E(K, M)] = K \oplus (K \oplus M) = M$

M	0 1 0 0 1 0 1 1 1 0
K	1 1 1 0 1 0 1 0 1 0
CT	1 0 1 0 0 0 0 1 0 0

\oplus

Vernam Cipher / One Time Pad

One-Time Pad is so efficient

- Very fast encryption/decryption

What about length of the secret key?

What about randomness of the secret key?

One-Time Pad is a secure scheme?

Perfect Secrecy

A cipher (K, M, CT) has perfect secrecy if

- Given plaintexts $m_0, m_1 \in M \rightarrow |m_0| = |m_1|$
- Given ciphertext $c \in CT$
- Given the secret key $k \stackrel{R}{\leftarrow} K$

$$\mathbf{P}\{E(k, m_0) = c\} = \mathbf{P}\{E(k, m_1) = c\}$$

Knowing c cannot reveal anything about plaintext

Knowing c can't tell anything of m_0, m_1

No attacks on CT

Perfect secrecy requires: $|\mathcal{K}| \geq |\mathcal{M}|$

One-Time Pad: Perfect Secrecy

The only cryptosystem that provides perfect secrecy!

- Knowing the ciphertext doesn't give you any additional insight into the value of the plaintext

It is also unconditionally secure

- Cannot be broken even with *infinite* computational resources

An attacker with infinite resources can break a 10,000-bit key cipher in one time-step

- Have $2^{10,000}$ computers each try a key
- That's more computers than atoms in the universe!
- System is computationally secure
 - Adversary is computationally bounded

So, can we use OTP for everything?!

The bad news...

The key bits need to be truly random

- Does your computer have true random number generator?

The sender and receiver must have the same key stream

- How do you communicate the key stream securely?

A bit of the key stream can only be used once

- Key needs to be as long as the message
- That's a lot of bits over time
 - A lot to have to send securely

NOT practical for the vast majority of applications

The good news...

We can approximate a OTP with a stream cipher

Main idea:

- Use a shorter key to generate a key stream in a pseudo-random fashion
 - More practical
 - Can achieve computational security
 - Unfortunately, does not achieve perfect secrecy

Symmetric Cryptography

STREAM CIPHERS



Pseudo Random Generator

We need many unpredictable random numbers

- Very difficult to produce random
- Most behaviour are predictable
- Often depends on implementation



Pseudo Random Generator

Try to approximate random generators

- Takes a truly random key (*seed*)
- Expand the key into a “*random-looking*” sequence

$$\{0, 1\}^s \rightarrow \{0, 1\}^n, \quad n \gg s$$

Expansion

- Seed 128 – *bit* → Key 10^6 – *bit*

Efficiently computable using deterministic algorithm!

Famous PRG fails: MIT Magic Cookie

key = random()%256

Only 256 possible values of key: 2^8 *bruteforce*

Famous PRG fails: Kerberos

$srand(\mathbf{time}_{ms} \oplus \mathbf{time}_s \oplus \mathbf{pid} \oplus \mathbf{counter} \oplus \mathbf{id})$
 $\mathbf{key} = \mathbf{random}()$

Reduced entropy $32\text{bit} \rightarrow 20\text{bit}: 2^{20} \text{bruteforce}$

The most common random...

Random by *Linear Generation Algorithm*

- Take a module $p \leftarrow 2^{31}$
- Take two constant values a, b
- Generate values using linear combination

$$x_{i+1} = (a * x_i + b) \bmod p$$

- 32-bit of output's entropy
- Correlation between values!
- Don't use for crypto purposes!!!

- rand() function in Windows – an LCG with
 - **a = 214013, b = 2531011, p = 232**

Is it unpredictable?

Consider a Linear PRG

- Choose $K(\sim 2^{30})$, a , b , c , d
- Initialize $\{x_0, y_0\}$ with random values in $\{0, K\}$

$$PRG_n(K) = x_n \oplus y_n$$

- $x_{n+1} = a * x_n + b \text{ mod } K$
- $y_{n+1} = c * y_n + d \text{ mod } K$

Can you recover PRG_{i+1} given $PRG_{1..i}$?

```
import random

P = 295075276L    # about 2^28

class MyPrng(object):
    def __init__(self, p):    # generate seed with 56 bits of entropy
        self.p = p
        self.x = random.randint(0, p)
        self.y = random.randint(0, p)

    def next(self):
        # x_{i+1} = 2*x_{i}+5 (mod p)
        self.x = (2*self.x + 5) % self.p

        # y_{i+1} = 3*y_{i}+7 (mod p)
        self.y = (3*self.y + 7) % self.p

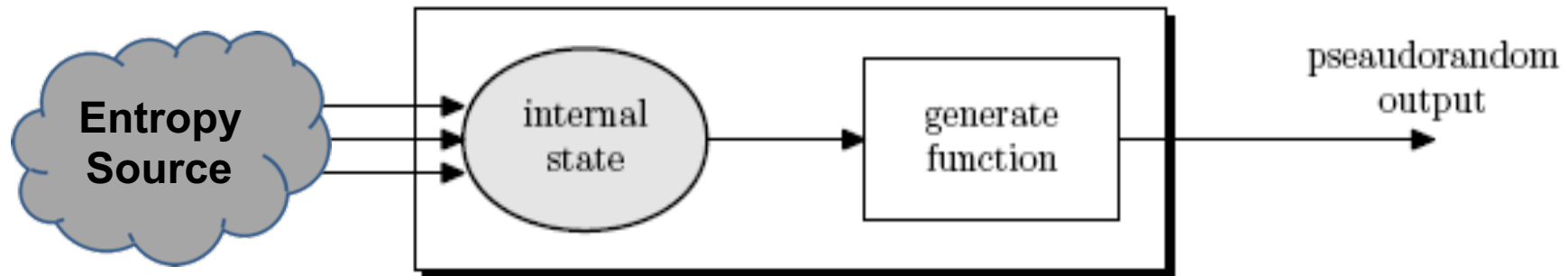
        # z_{i+1} = x_{i+1} xor y_{i+1}
        return (self.x ^ self.y)

if __name__ == '__main__':
    prng = MyPrng(P)
    for i in range(1, 10):
        print "output #%d: %d" % (i, prng.next())
```

Random in practice

Pseudo-random generators in practice

- Unix read from `/dev/random` and `/dev/urandom` (`cat ...`)
 - Maintains an “entropy pool” and a number of bits
 - Hashes pool and provides bits as output
 - Needs to continuously add entropy to internal state
- Needs entropy sources
 - Keyboard, mouse, etc...



“Looks random”

What does it mean?

Non-cryptographic applications:

- Should pass some statistical tests

Cryptography:

- Should pass all polynomial-time tests

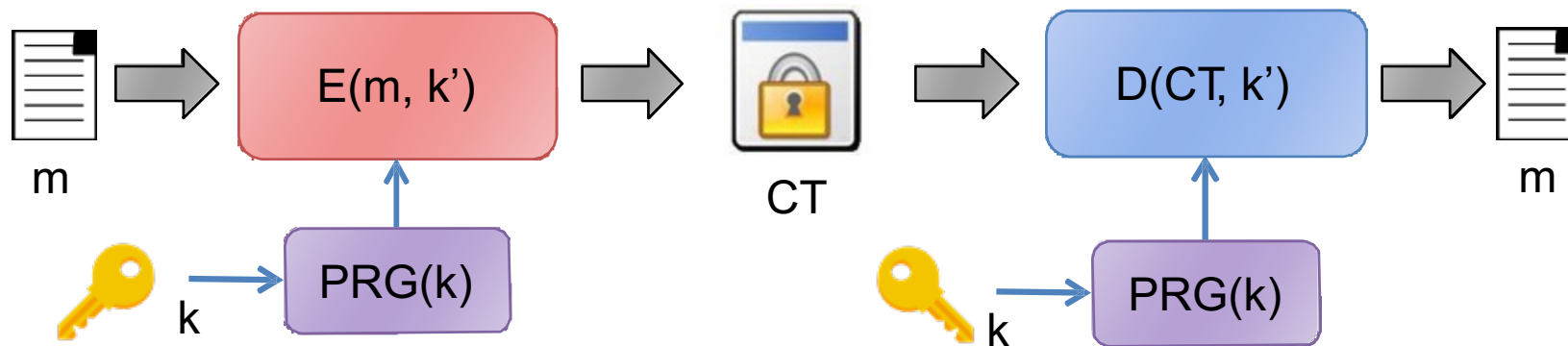
From OTP to Stream Cipher

Main Idea

- Replace random key with pseudo-random

Replace key with PRG

- $CT := E(K, M) = PRG(K) \oplus M$
- $D(K, CT) = PRG(K) \oplus CT$



Are Stream Ciphers secure?

Cannot have perfect secrecy

- Need to define differently the security notion
- Key is not truly random $k \stackrel{R}{\leftarrow} K$

Security depends on the specific PRG

- MUST be unpredictable

$$P\{PRG_{1..i}(k) = PRG_{i+1}(k)\} \simeq 1/2$$

Improve Stream Ciphers

Needs to use a freshness key

- Concatenate key and a nonce

$$PRG(k||r): \{0,1\}^s \times R \rightarrow \{0,1\}^n$$



- The pair $\langle k,r \rangle$ MUST be used only once!

$$CT_0 := E(k, m_0, r_0) = m_0 \oplus PRG(k||r_0)$$

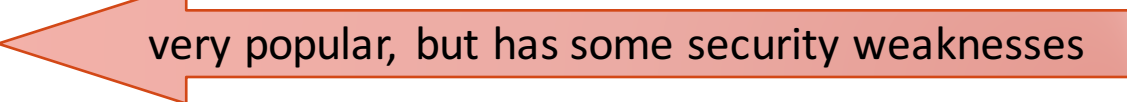
$$CT_1 := E(k, m_1, r_1) = m_1 \oplus PRG(k||r_1)$$

Popular Historical Stream Ciphers

Based on the linear feedback shift registers:

- A5/1 and A5/2 (used in GSM)  completely broken
- Content Scramble System (CSS) encryption (used in DVD)  completely broken

Other:

- RC4  very popular, but has some security weaknesses

Symmetric Cryptography

HOW TO NOT USE ONE-TIME PAD

Two-Time Pad

Don't change the OTP key

$$CT_0 = m_0 \oplus k$$

$$CT_1 = m_1 \oplus k$$

...

$$CT_n = m_n \oplus k$$

Attacker exploit it easily

$$CT_0 \oplus CT_1 = m_0 \oplus k \oplus m_1 \oplus k = m_0 \oplus m_1$$

$$CT_0 \oplus CT_i = m_0 \oplus k \oplus m_i \oplus k = m_0 \oplus m_i$$

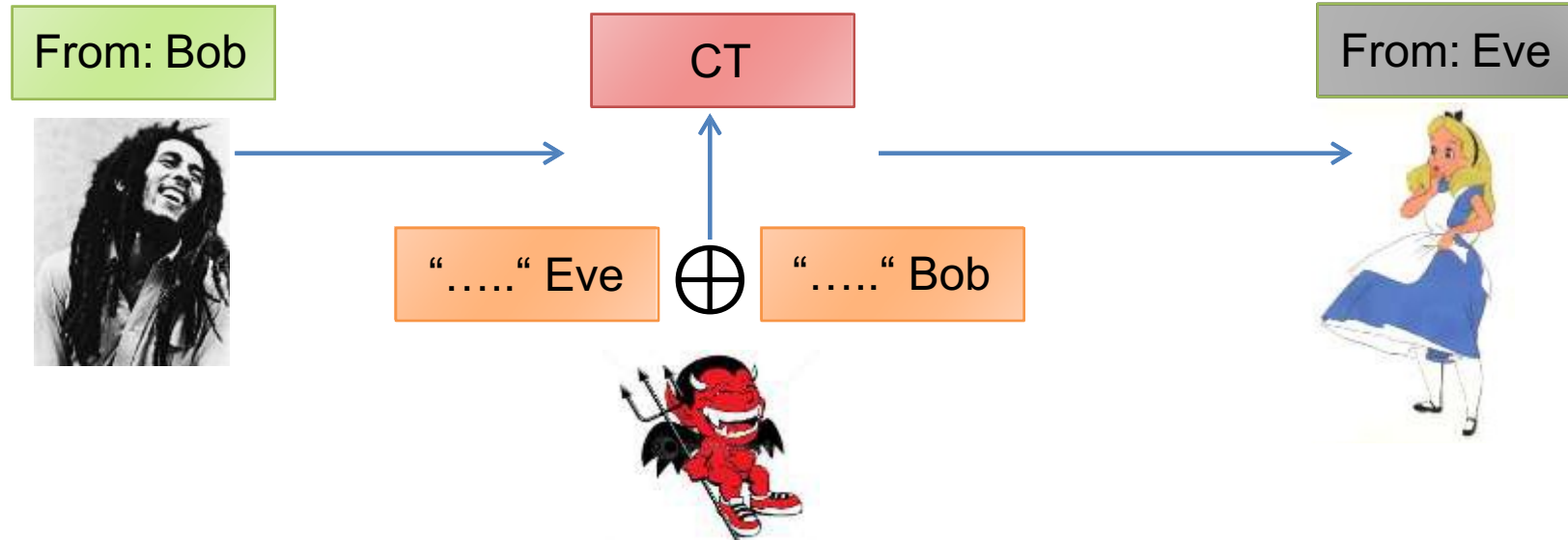
...

- Enough redundancy of ASCII and common languages
 - Simplified cryptanalysis

Message Integrity

One-Time Pad CT can be modified

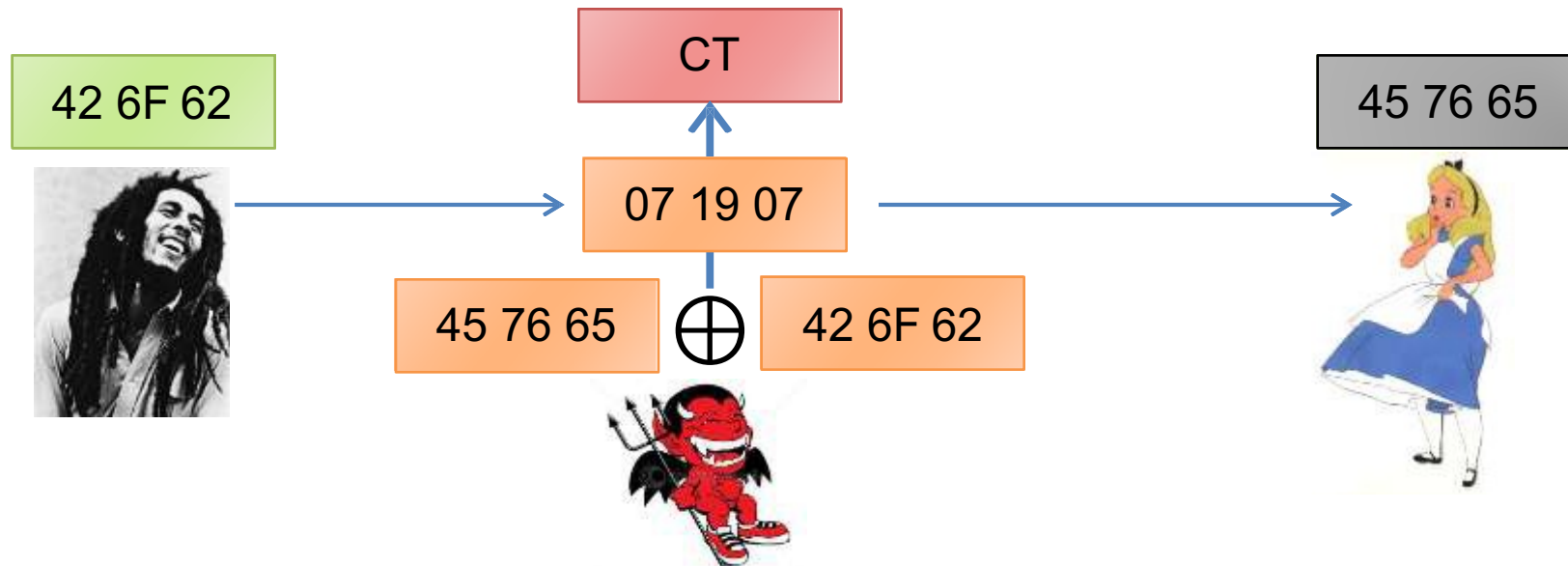
- Modifications are undetectable
- Modifications have predictable impact on PT
 - Not limited to *Denial Of Service*



Message Integrity

One-Time Pad CT can be modified

- Modifications are undetectable
- Modifications have predictable impact on PT
 - Not limited to *Denial Of Service*



Use Case: MS-PPTP

PPTP (Point to Point Tunneling Protocol)

- Allows to create virtual private networks using tunneling
- Implemented in Windows 98 and Windows NT
- Cryptographically broken by Microsoft since 2012

Same secret key used in the same way

- Usually created from the user password

