Elements of Cryptography (Part 2)

Lec 6 Ciphers

2016

Outline

- The need for cryptography
- Ciphers classification

Cryptography

- The process of disguising a message in such a way as to hide its substance is called *encryption*
  - a message is called *plaintext*
  - the encrypted message is called *ciphertext*
  - The algorithm used is called *cypher*
  - the process of turning ciphertext back into plaintext is called *decryption*
- The art and science of keeping messages secure is called *cryptography*
  - *cryptanalysis* is the art and science of breaking ciphertext

Auguste Kerckhoffs (1883)

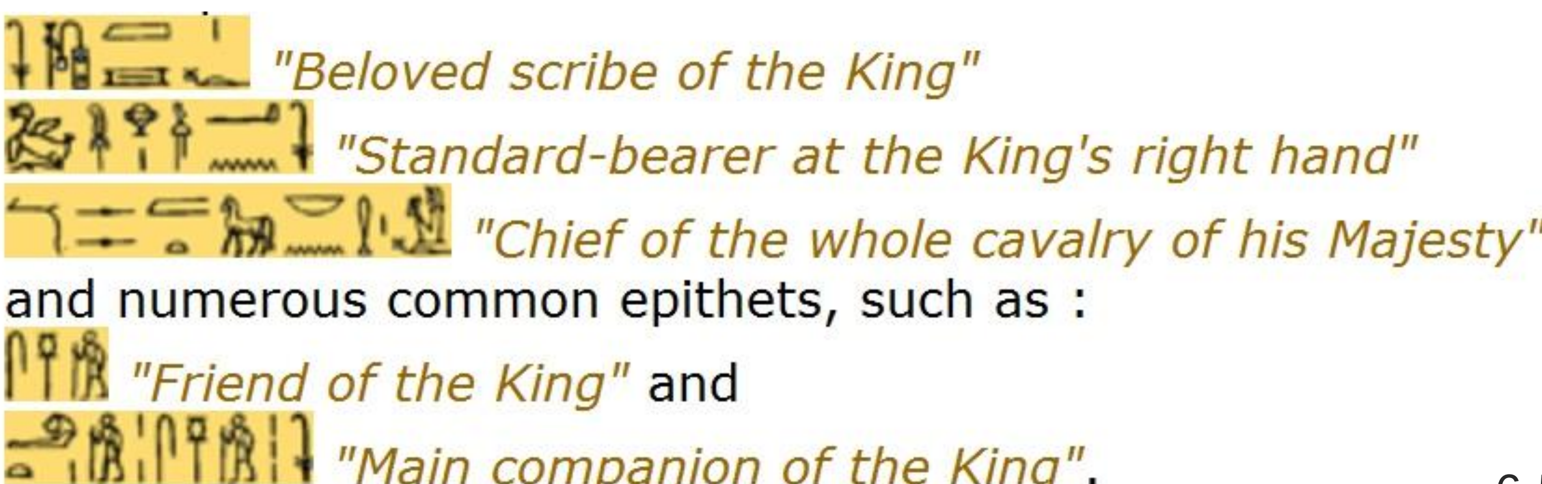
A cryptosystem should be secure

even if everything about the system,

except the key, is public knowledge

The Beginning

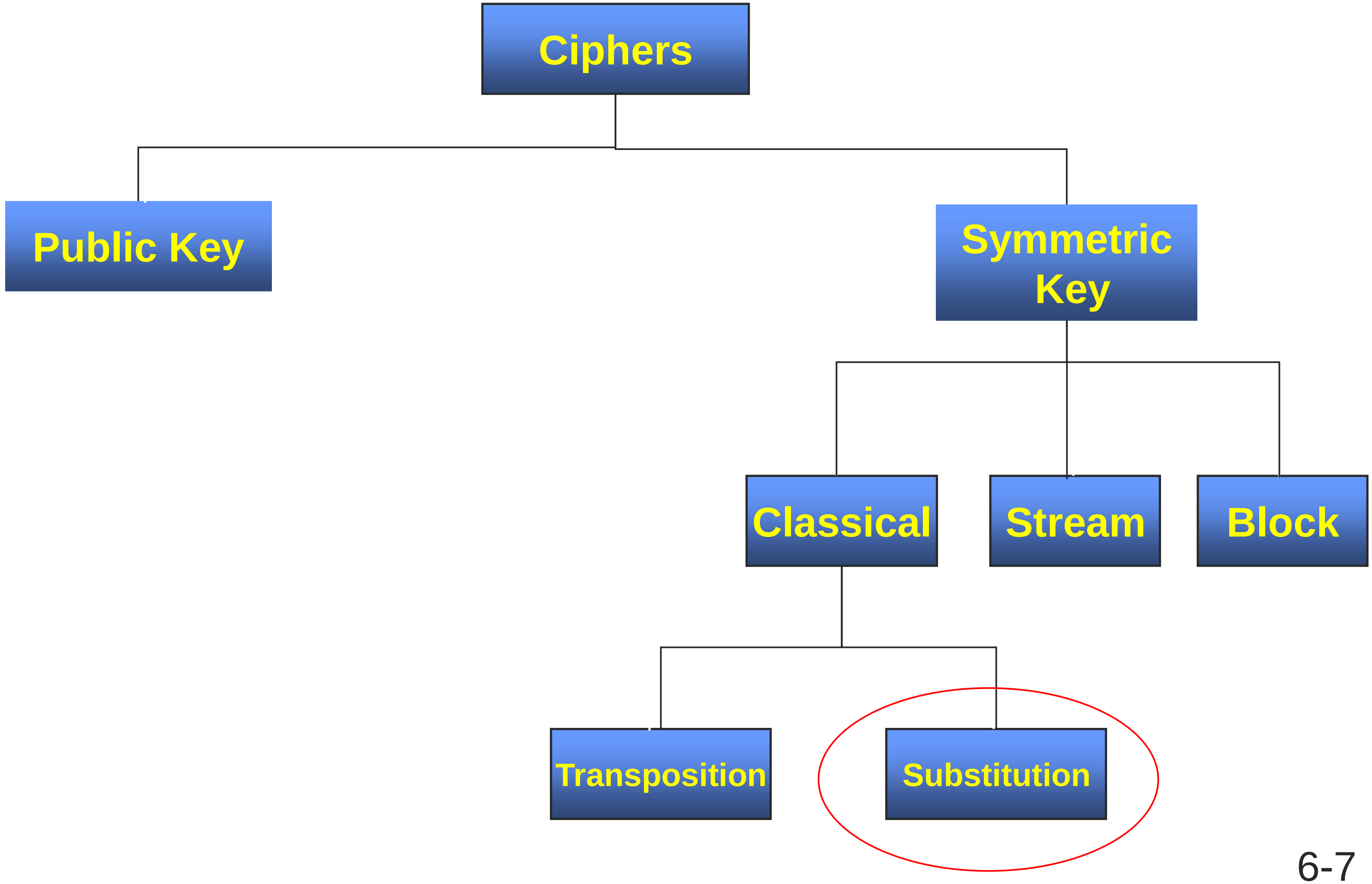
- One of the first recorded uses of secret writing occurred in 1900 BC
  - An inscription carved into the rock of the main chamber of the tomb of an Egyptian nobleman uses some unusual symbols in place of standard hieroglyphics
  - Its purpose was (perhaps) to impress the reader by adding dignity and authority to the message



Nowadays

- We want to protect information
- Ciphers classification

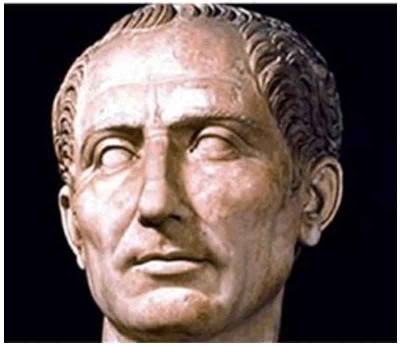
Cipher Classification



Substitution Cipher

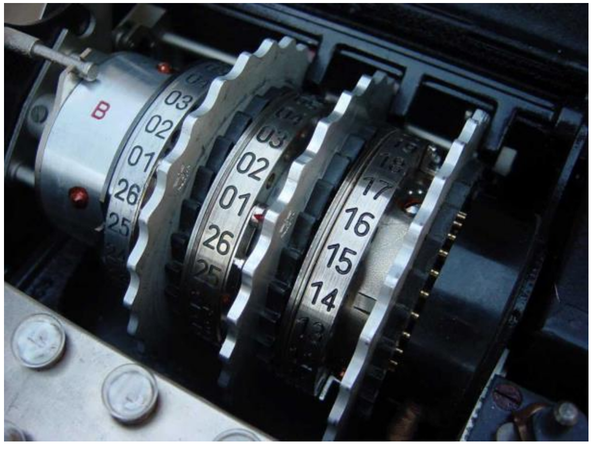
Caesar cipher

100 BC



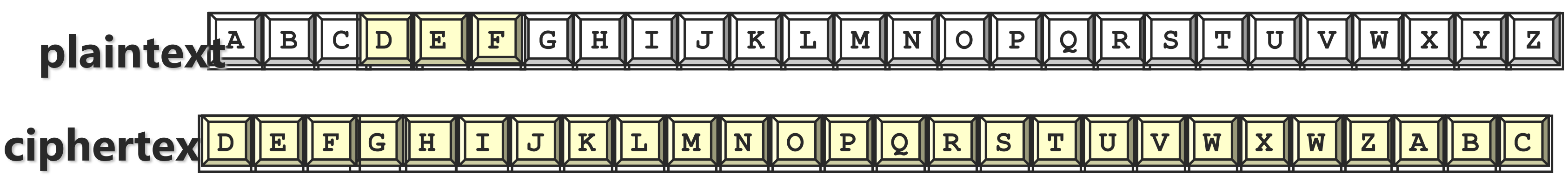
German Enigma

WWII



Substitution Cipher

- Each character in the plaintext is substituted for another character in the ciphertext
  - The Caesar Cipher replaces each plaintext character by the character 3 positions to the right   (key = 3)



Caeser Cipher

plaintext	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
ciphertext	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

MESSAGE

`the word privacy does not appear in the united states constitution`

`wkh zrug sulydfb grhv qrw dsshdu lq wkh xqlwhg vwdwhv frqvwlwxwlrq`

NOTE: the shift could be any value from 1 to 25

How would you break the Caesar cipher?

Vigenere Cipher

- Caeser cipher is a monoalphabetic cipher
- Vigenere cipher is an example of a polyalphabetic cipher where the substitution pattern varies
  - a plaintext “e” may be replaced by a ciphertext “p” one time and a ciphertext “w” another
  - the Vigenere cipher does this using a table
- Blaise de Vigenère

  Traicté des Chiffres ou Secrètes Manières d'Escrire (1586)

Vigenere Table

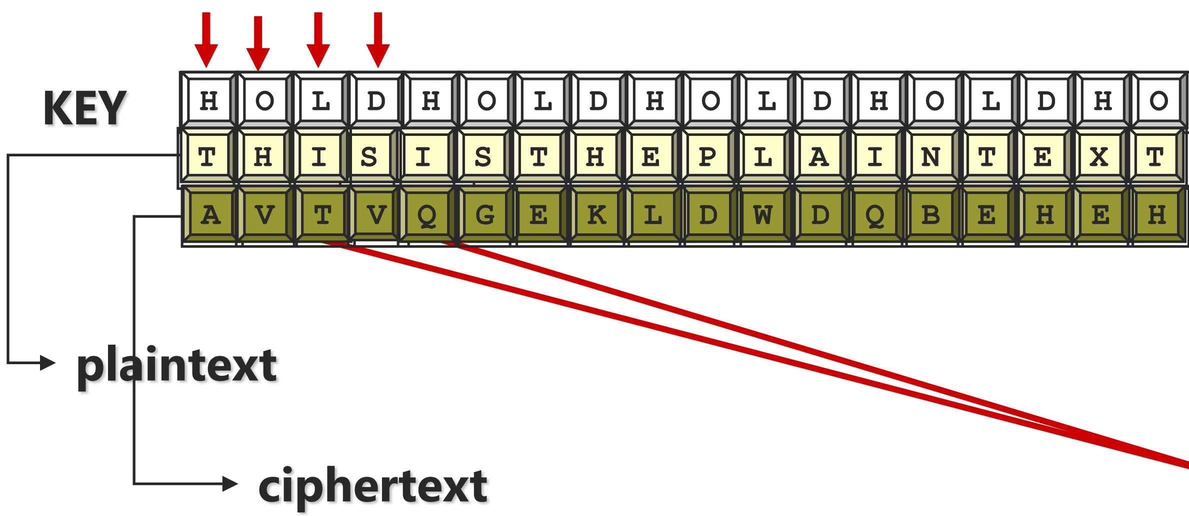
keys chars on the top of the table

plaintext chars
on the side

	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 Operation

- A keyword is selected and it is repeatedly written above the plaintext
  - **EXAMPLE: using the keyword “hold”**



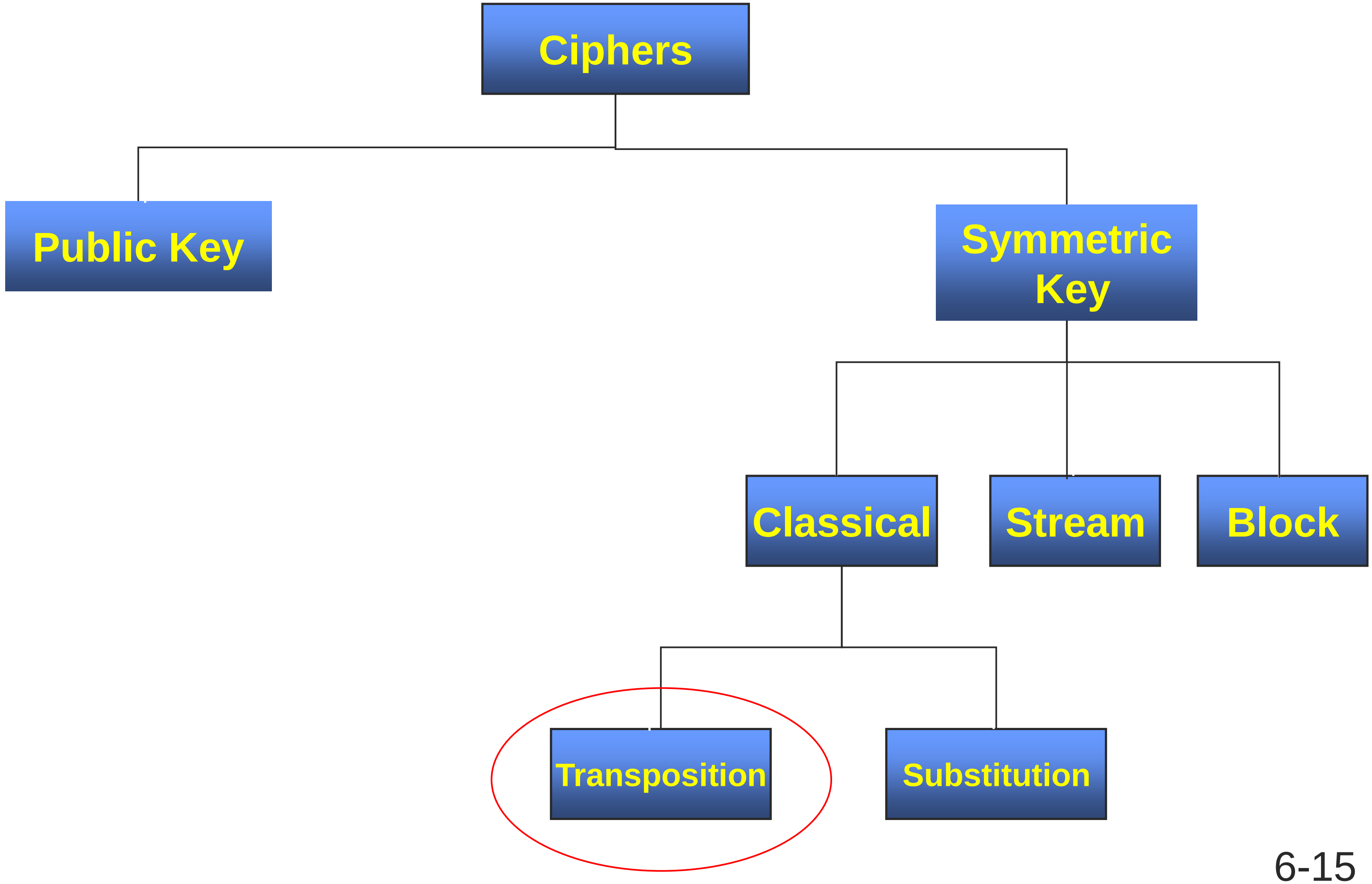
	a	b	c	d	e	f	g	h	i	.	.	.
a	a	b	c	d	e	f	g	h	i			
b	b	c	d	e	f	g	h	i	j	.	.	.
n	c	d	e	f	g	h	i	j	k	.	.	.
d	d	e	f	g	h	i	j	k	l	.	.	.
e	e	f	g	h	i	j	k	l	m	.	.	.
f	f	g	h	i	j	k	l	m	n	.	.	.
g	g	h	i	j	k	l	m	n	o	.	.	.
h	h	i	j	k	l	m	n	o	p	.	.	.
i	i	j	k	l	m	n	o	p	q	.	.	.
j	j	k	l	m	n	o	p	q	r	.	.	.
k	k	l	m	n	o	p	q	r	s	.	.	.
l	l	m	n	o	p	q	r	s	t	.	.	.
m	m	n	o	p	q	r	s	t	u	.	.	.
n	n	o	p	q	r	s	t	u	v	.	.	.
o	o	p	q	r	s	t	u	v	w	.	.	.
p	p	q	r	s	t	u	v	w	x	.	.	.
q	q	r	s	t	u	v	w	x	y	.	.	.
r	r	s	t	u	v	w	x	y	z	.	.	.
s	s	t	u	v	w	x	y	z	a	.	.	.
t	t	u	v	w	x	y	z	a	b	.	.	.
u	u	v	w	x	y	z	a	b	c	.	.	.

Breaking Vigenere

- For more than 300 years cryptanalysts worked on the problem of breaking a polyalphabetic cipher (like the V-cipher)

- In 1863, a Polish Infantry officer, Friedrich W. Kasiski, published a short book (95 pages) which changed the nature of cryptography
  - He had found a simple solution to the polyalphabetic ciphers
  - He died in 1881 without realizing that he had started a revolution in cryptography

Cipher Classification



Transposition Ciphers

- Like jigsaw puzzles in that all the pieces are present but are merely disarranged
  - rather than substitute letters, rearrange letters in the text
- Most transpositions involve a geometric figure (square, rectangle, . . .)
  - the letters are inscribed in the figure upon some agreed direction
  - the letters are then transcribed or rewritten according to another direction to form the ciphertext

Rail-Fence Cipher

- Plaintext: *this is a test*
- Ciphertext: *tiehsstsiat*
- Key: *(lines=3, columns=11)*

Inscribe by a zigzag pattern

extract by rows

t				i				e		
	h		s		s		t		s	
		i				a				t

Other Figures

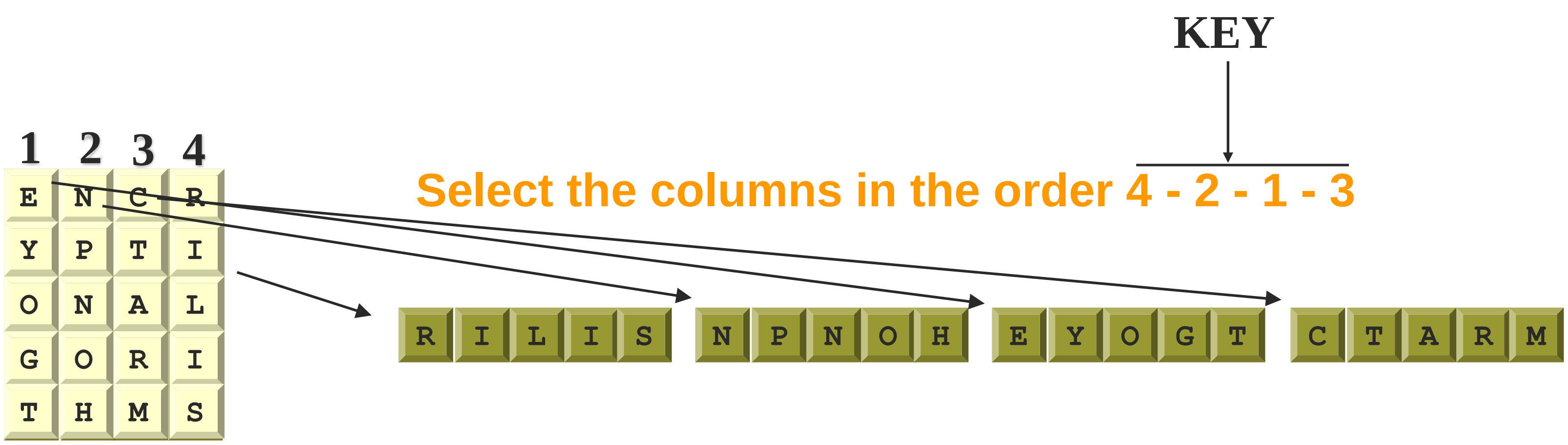
- Use a triangle
  - Plaintext: *You must do that*
  - Ciphertext: *tuhosayuttmdnoow*
  - Key: *(lines=4, columns=7)*

Inscribe by rows

			Y			
		o	u	m		
	u	s	t	d	o	
t	h	a	t	n	o	w

Column Transposition Ciphers

- Write the plaintext into a matrix by rows, then generate the ciphertext by selecting the columns in a given order
  - Plaintext: *encryption algorithms*



Cryptanalysis: Breaking Transpositions

- We will look at the process of breaking a keyed columnar transposition with completely filled rectangles

- Given the following ciphertext, what is the first thing you must determine?

```
NETEFLTDSRTSSTFMDCETDRHXSWHOHOEEADUOUUFIRRRRS
NEROTCFIEMEDSHARTCPJAOEGEWNLHOEPMWAWERUVAAINA
TSDDSOEOACEHNTLHFLAURAEENOTOTSSOSYSTNNCGEMETT
YDYRRNEOOERESTHINR
```

Tasks

- There are three tasks involved in breaking a column transposition cipher:
  1. Find possible rectangle sizes
     - in a completely filled transposition, the number of characters is the product of the number of rows and the number of columns
     - so, factor the number of characters to determine possible row and column sizes
  2. Select the correct rectangle
  3. Find the column order

Example

- First, factor the message length of the example ciphertext
  - our message has 153 letters and 153 has 3, 9, 17, and 51 as factors
  - possible rectangle sizes (columns by row):

    3 x 51   51 x 3   9 x 17   17 x 9

- 9 x 17 and 17 x 9 are the most probable because the other two have poor distributions of rows vs columns

Which Rectangle?

- Since the factors only supply possible column sizes - test each possibility by doing a vowel count
  - any line of plaintext should contain about 40% vowels
  - So, count the vowels in each row of each possible rectangle
  - the one with the best match to 40% is the best choice for the actual rectangle

Rectangle 1

- The **9 x 17 rectangle** - 3.6 vowels per row expected (9 x 0.4)

1	2	3	4	5	6	7	8	9	vowels	difference
N	C	U	F	G	A	N	S	E	2	1.6
E	E	O	I	E	A	T	S	Y	7	3.4
T	T	U	E	W	I	L	O	R	4	.4
E	D	U	M	N	N	H	S	R	2	1.6
F	R	F	E	L	A	F	Y	N	3	.6
L	H	I	D	H	T	L	S	E	2	1.6
T	X	R	S	O	S	A	T	O	3	.6
D	S	R	H	E	D	U	N	O	3	.6
S	W	R	A	P	D	R	N	E	2	1.6
R	H	R	R	M	S	A	C	R	1	2.6
T	O	S	T	W	O	E	G	E	4	.4
S	H	N	C	A	E	E	E	S	4	.4
S	O	E	P	W	O	N	M	T	3	.6
T	E	R	J	E	A	O	E	H	5	1.4
F	E	O	A	R	C	T	T	I	4	.4
M	A	T	O	U	E	O	T	N	5	1.4
D	D	C	E	V	H	T	Y	R	2	1.6

Total Difference
20.6

Rectangle 2

- The second possible **rectangle is 17 x 9** (expected per row vowel count = 6.8)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Vowels	Difference
N	R	E	O	U	N	M	P	L	E	T	C	A	O	N	Y	E	8	1.2
E	T	T	H	U	E	E	J	H	R	S	E	U	T	N	D	R	6	.8
T	S	D	O	F	R	D	A	O	U	D	H	R	S	C	Y	E	6	.8
E	S	R	E	I	O	S	O	E	V	D	N	A	S	G	R	S	7	.2
F	T	H	E	R	T	H	E	P	A	S	T	E	O	E	R	T	6	.8
L	F	X	A	R	C	A	G	M	A	O	L	E	S	M	N	H	5	1.8
T	M	S	D	R	F	R	E	W	I	E	H	N	Y	E	E	I	7	.2
D	D	W	U	R	I	T	W	A	N	O	F	O	S	T	O	N	6	.8
S	C	H	O	S	E	C	N	W	A	A	L	T	T	T	O	R	5	1.8

Total Difference = 8.4

Which of the two is the most likely rectangle?

Find the Column Order: Letter Affinity

- Once the rectangle size is determined, the column order must be discovered
  - Advantage is taken of all the characteristics of the plaintext language

- First, in all languages there are certain letters usually of medium or low frequency which combine with other letters to form diagrams of high frequency
  - H (medium frequency) combines with T to form TH (highest frequency)
  - H combines with C (medium frequency) to form CH
  - V (low frequency) combines with E to form VE (medium frequency in military text)

Other Heuristics

- When there is an H, attempts should be made to combine it with a T or a C
- AV should be combined first with an E
- AK should be combined first with a C

Pilot Letters

- Second, there is usually in every language at least one letter which can be followed by only certain other letters forming an *obligatory sequence* or *invariable digraph*
  - Q is always followed by U
  - J can only be followed by a vowel
  - X can be preceded only by a vowel and, except at the end of a word, X can only be succeeded by a vowel or C, H, P, T

- Letters such as these with limited affinity are called pilot letters

Anagramming

- **Breaking a transposition cipher is a process of anagramming by selecting a pilot letter and trying to form digrams with the other letters in its row**

for example, select the J in column 8 and match it with . . .

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
N	R	E	O	U	N	M	P	L	E	T	C	A	O	N	Y	E
E	T	T	H	U	E	E	J	H	R	S	E	U	T	N	D	R
T	S	D	O	F	R	D	A	O	U	D	H	R	S	C	Y	E
E	S	R	E	I	O	S	O	E	V	D	N	A	S	G	R	S
F	T	H	E	R	T	H	E	P	A	S	T	E	O	E	R	T
L	F	X	A	R	C	A	G	M	A	O	L	E	S	M	N	H
T	M	S	D	R	F	R	E	W	I	E	H	N	Y	E	E	I
D	D	W	U	R	I	T	W	A	N	O	F	O	S	T	O	N
S	C	H	O	S	E	C	N	W	A	A	L	T	T	T	O	R

Centiban Weights

- The US government studied a set of 5,000 digraphs and produced a table of what are called **centiban weights**
- Higher values indicate that the character occurs more often

Centiban Table

Second Letter

First Letter

	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	33	45	61	73	13	38	45	25	64	13	25	76	61	89	25	58	00	82	80	83	59	48	33	00	58	00
B	38	00	00	00	66	00	00	00	25	13	00	45	13	00	38	00	00	25	13	13	25	00	00	00	48	00
C	67	00	33	13	76	13	00	61	48	00	38	42	13	13	80	00	00	38	13	61	38	00	13	00	13	00
D	76	38	38	51	77	51	25	25	73	13	00	33	42	38	63	42	25	58	59	62	42	33	38	00	13	00
E	78	38	76	88	81	66	38	48	73	13	00	74	61	99	58	67	58	94	86	79	33	67	48	48	38	13
F	42	00	25	13	55	56	13	00	80	00	00	25	13	00	80	13	00	53	33	56	33	00	13	00	13	00
G	48	00	25	13	61	25	13	67	42	13	00	25	13	33	45	25	00	42	33	38	25	00	13	00	00	00
H	67	13	33	25	67	42	00	00	77	00	00	13	25	33	67	13	13	64	38	74	51	00	13	00	13	00
I	51	25	69	45	59	55	67	00	00	00	25	70	53	92	80	48	00	73	78	73	00	72	00	62	00	25
J	18	00	00	00	25	00	00	00	00	00	00	00	00	00	25	00	00	00	00	00	25	00	00	00	00	00
K	13	00	13	00	45	00	00	00	25	00	00	13	00	13	00	00	00	00	13	00	00	00	00	00	00	00
L	74	33	33	53	79	33	13	13	67	00	00	73	25	13	59	33	00	25	45	51	25	25	25	00	55	00
M	78	45	33	13	72	13	00	13	53	00	00	00	59	00	55	51	00	25	38	25	25	00	00	00	25	00
N	72	25	67	85	87	53	73	38	75	13	25	42	42	51	66	33	13	38	71	93	48	33	33	00	42	00
O	48	38	51	58	33	72	25	33	42	13	25	67	72	92	45	72	00	89	61	67	79	48	51	13	25	00
P	61	13	13	13	70	25	00	33	45	00	00	59	38	13	64	56	00	66	45	51	33	13	13	00	13	00
Q	00	00	00	00	00	00	00	00	00	00	00	00	13	00	00	00	00	13	00	00	62	00	00	00	00	00
R	80	25	53	64	96	45	48	33	75	13	13	42	53	48	74	59	00	56	75	81	42	42	38	00	53	00
S	71	33	59	42	84	58	25	72	77	00	13	25	33	38	62	55	00	42	67	88	56	13	38	00	13	00
T	74	33	45	45	91	48	13	92	82	00	00	42	45	48	84	25	13	64	67	67	42	00	78	00	80	13
U	42	33	33	33	56	13	51	00	42	00	00	45	42	68	13	25	00	75	58	58	00	13	00	00	00	00
V	45	00	00	00	87	00	00	00	58	00	00	00	00	00	13	00	00	00	00	13	00	00	00	00	00	00
W	58	00	00	00	69	00	00	38	59	00	00	13	00	25	67	00	00	13	13	00	00	00	00	00	13	00
X	25	00	25	13	13	13	00	13	25	00	00	00	00	13	13	25	00	13	13	48	00	00	00	00	00	00
Y	45	25	38	38	53	56	13	13	33	00	00	25	25	45	55	33	00	38	56	62	13	00	13	00	00	00
Z	13	00	00	00	25	00	00	00	13	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Possible Pairings

- Try both JU pairings in the example and rank each digram by the centiban numbers

8	5	rank	8	13	rank
P	U	33	P	A	61
J	U	25	J	U	25
A	F	38	A	R	82
O	I	42	O	A	48
E	R	94	E	E	81
G	R	42	G	E	61
E	R	94	E	N	99
W	N	25	W	O	67
N	S	71	N	T	93
		464			617

Result: pair columns 8 and 13

Form Trigrams

- The digram JU should be followed by a consonant, preferably N or S - columns 15 and 11 are candidates

8	13	15	rank
P	A	N	89
J	U	N	68
A	R	C	53
O	A	G	45
E	E	E	81
G	E	M	61
E	N	E	87
W	0	T	67
N	T	T	67
			618

8	13	11	rank
P	A	T	83
J	U	S	58
A	R	D	64
O	A	D	73
E	E	S	86
G	E	O	58
E	N	E	87
W	O	O	45
N	T	A	74
			628

Look for Words

- Continue this process looking for new digraphs and for possible words
- For example, are there any possible words in this text?

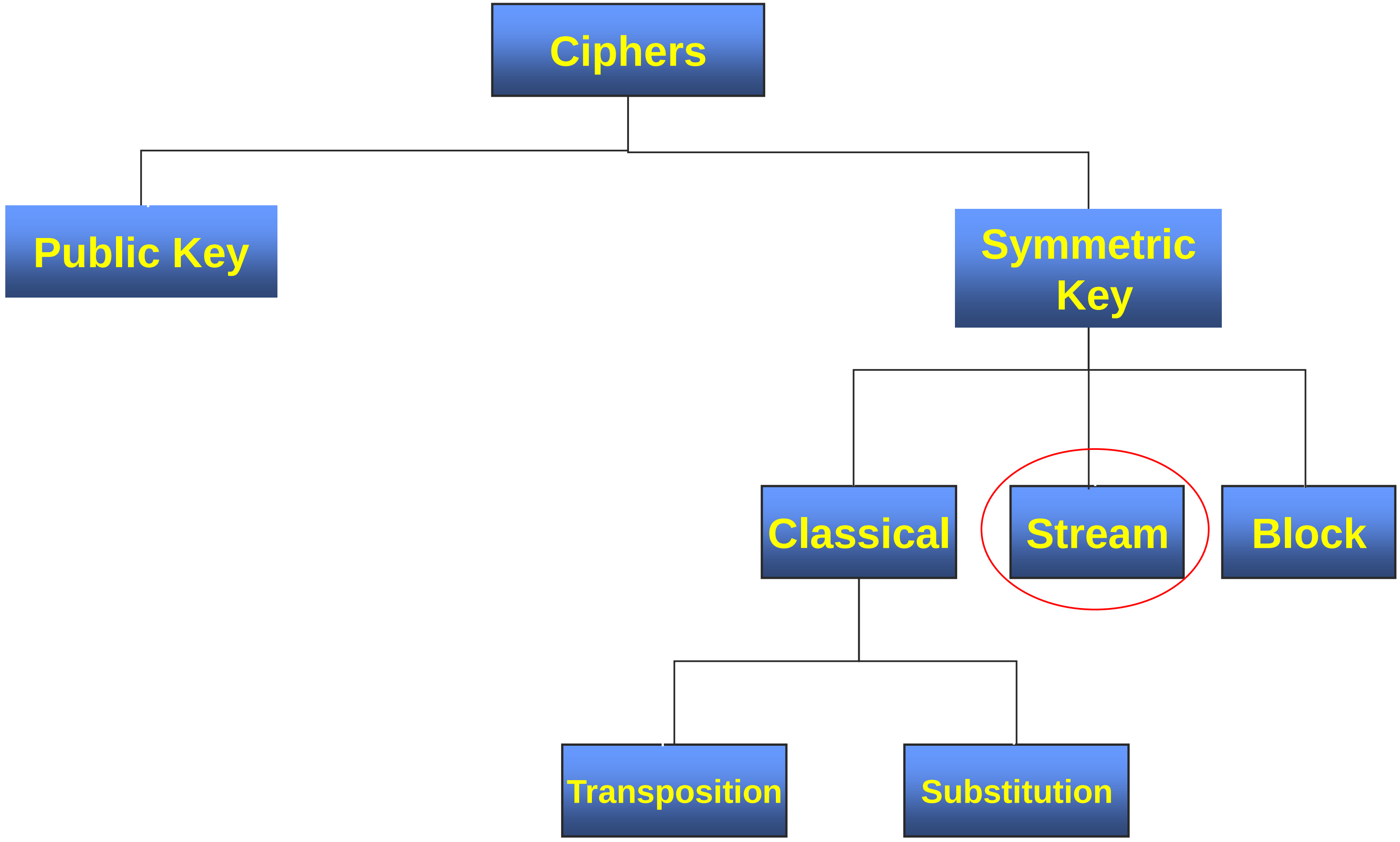
8	13	11	1	2	3	4	5	6	7	9	10	12	14	15	16	17
P	A	T	N	R	E	O	U	N	M	L	E	C	O	N	Y	E
J	U	S	E	T	T	H	U	E	E	H	R	E	T	N	D	R
A	R	D	T	S	D	O	F	R	D	O	U	H	S	C	Y	E
O	A	D	E	S	R	E	I	O	S	E	V	N	S	G	R	S
E	E	S	F	T	H	E	R	T	H	P	A	T	O	E	R	T
G	E	O	L	F	X	A	R	C	A	M	A	L	S	M	N	H
E	N	E	T	M	S	D	R	F	R	W	I	H	Y	E	E	I
W	O	O	D	D	W	U	R	I	T	A	N	F	S	T	O	N
N	T	A	S	C	H	O	S	E	C	W	A	L	T	T	O	R

Try centiban weights or look for other words

Solution

3	6	17	7	16	8	13	11	2	14	9	10	1	12	4	5	15
E	N	E	M	Y	P	A	T	R	O	L	E	N	C	O	U	N
T	E	R	E	D	J	U	S	T	T	H	R	E	E	H	U	N
D	R	E	D	Y	A	R	D	S	S	O	U	T	H	O	F	C
R	O	S	S	R	O	A	D	S	S	E	V	E	N	E	I	G
H	T	T	H	R	E	E	S	T	O	P	A	F	T	E	R	E
X	C	H	A	N	G	E	O	F	S	M	A	L	L	A	R	M
S	F	I	R	E	E	N	E	M	Y	W	I	T	H	D	R	E
W	I	N	T	O	W	O	O	D	S	A	N	D	F	U	R	T
H	E	R	C	O	N	T	A	C	T	W	A	S	L	O	S	T

Cipher Classification



Stream Ciphers

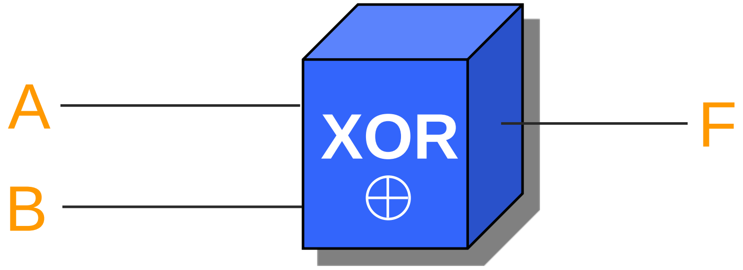
- Computer-based approach to cipher systems
  - data in computer is stored, processed, and transmitted in binary form (as 0’s and 1’s)
  - Letters are represented as binary bits (ASCII code)
- Encryption done at bit level
  - Execute at hight speed (higher than block ciphers)
  - Can have security vulnerabilities
- Commonly used in hardware applications
  - Pay-per-view TV encryption
  - Mobile phone conversation encryption

Bit Level Ciphers

- Using computers, ciphers are implemented at the bit level (ex. RC4)
  - that is, we can now substitute or transpose 0’s and 1’s
- For example, an A is **ASCII** is 0100 0001, so if I randomly change some 0’s to 1’s and some 1’s to 0’s the result might be 0010 1011 which is a “+”
- The problem is, how can I randomly change bits and yet still be able to recover the plaintext?
  - to do this we will use a binary function called the exclusive-OR (**XOR**)

XOR Function

- It is a two input, one output binary function where the output is 1 if the inputs are different and the output is 0 if the inputs are the same
  - this can be expressed in a “truth table” which lists all the inputs and outputs



A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

A will be the plaintext and **B** the key

Bit Stream

- The pattern of inputs and outputs may look like:

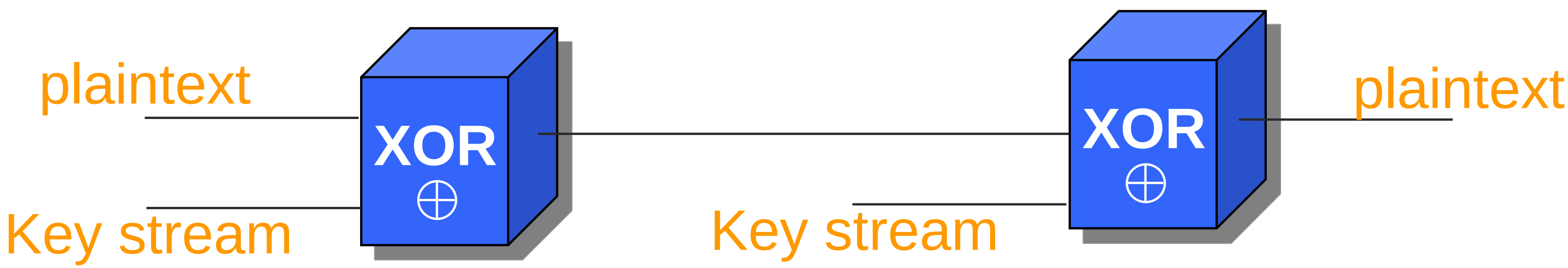
```
plaintext:  1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 0 1
      key:  0 0 1 1 1 0 1 0 1 0 0 1 1 0 0 1 1 0 0
ciphertext: 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 1 0 1
      key:  0 0 1 1 1 0 1 0 1 0 0 1 1 0 0 1 1 0 0
plaintext:  1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 0 1
```

Plaintext bits encrypted once at a time

Problem: How do we recover the plaintext from knowledge of the ciphertext and key?

Simple Stream Cipher

- Set up a known pattern (sequence) of 1's and 0's to use as a key
- Apply the key to the plaintext bit stream using an **XOR** function
- Recover the plaintext using the same key pattern on the ciphertext bit stream



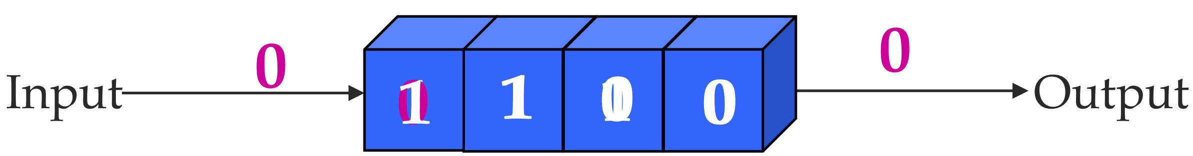
How to Generate a Key?

- A short sequence of key bits would be easy to remember but not very secure
- A long sequence of key bits would be secure but hard to remember
- PROBLEM: How can we generate a long random-appearing sequence of 0's and 1's yet easy to reproduce by legitimate users?
- ANSWER: Construct a ***Linear Feedback Shift Register*** - **LFSR**

Shift Register

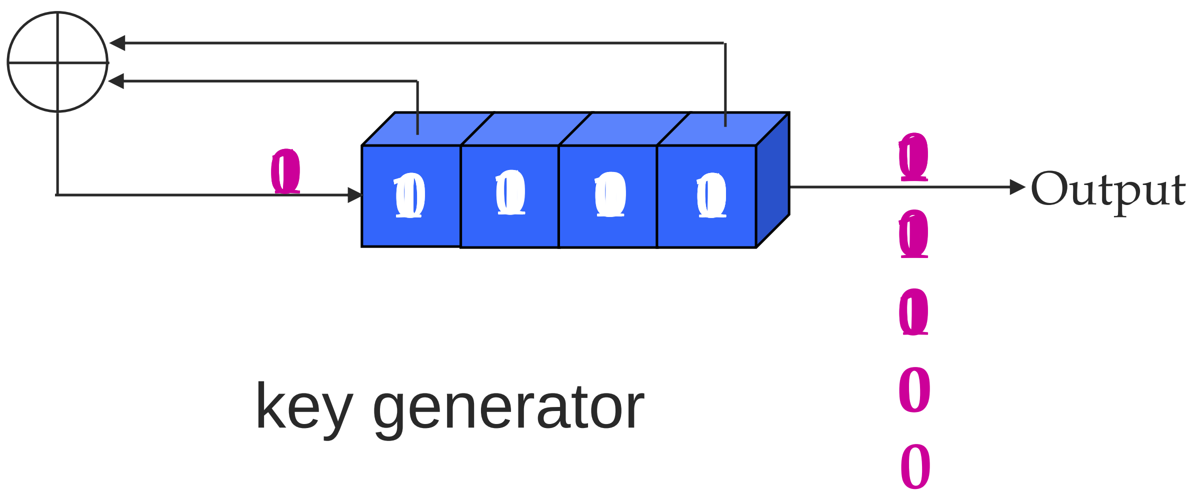
- A shift register is a hardware device which:
  - saves bits
  - shifts bits

- For example, a 4-bit shift register looks like:



Add Feedback

- Take some of the bits in the shift register, combine them with an **XOR**, and feedback the result as the input



key generator

Breaking a Stream Cipher

- One way is using an insertion attack
  - Intercept the ciphertext
  - Insert a known bit somewhere in the plaintext and get the modified plaintext encrypted with the same keystream
  - Knowledge of the single bit will compromise the plaintext

Insertion Attack

Assume the following ciphertext is intercepted:

```
p1 p2 p3 p4 p5 . . .
k1 k2 k3 k4 k5 . . .
c1 c2 c3 c4 c5 . . .
```

All we know is the ciphertext

Now insert a bit p after p1 and observe the new ciphertext:

```
p1 p  p2 p3 p4 p5 . . .
k1 k2 k3 k4 k5 k6 . . .
c1 c  d3 d4 d5 d6 . . .
```

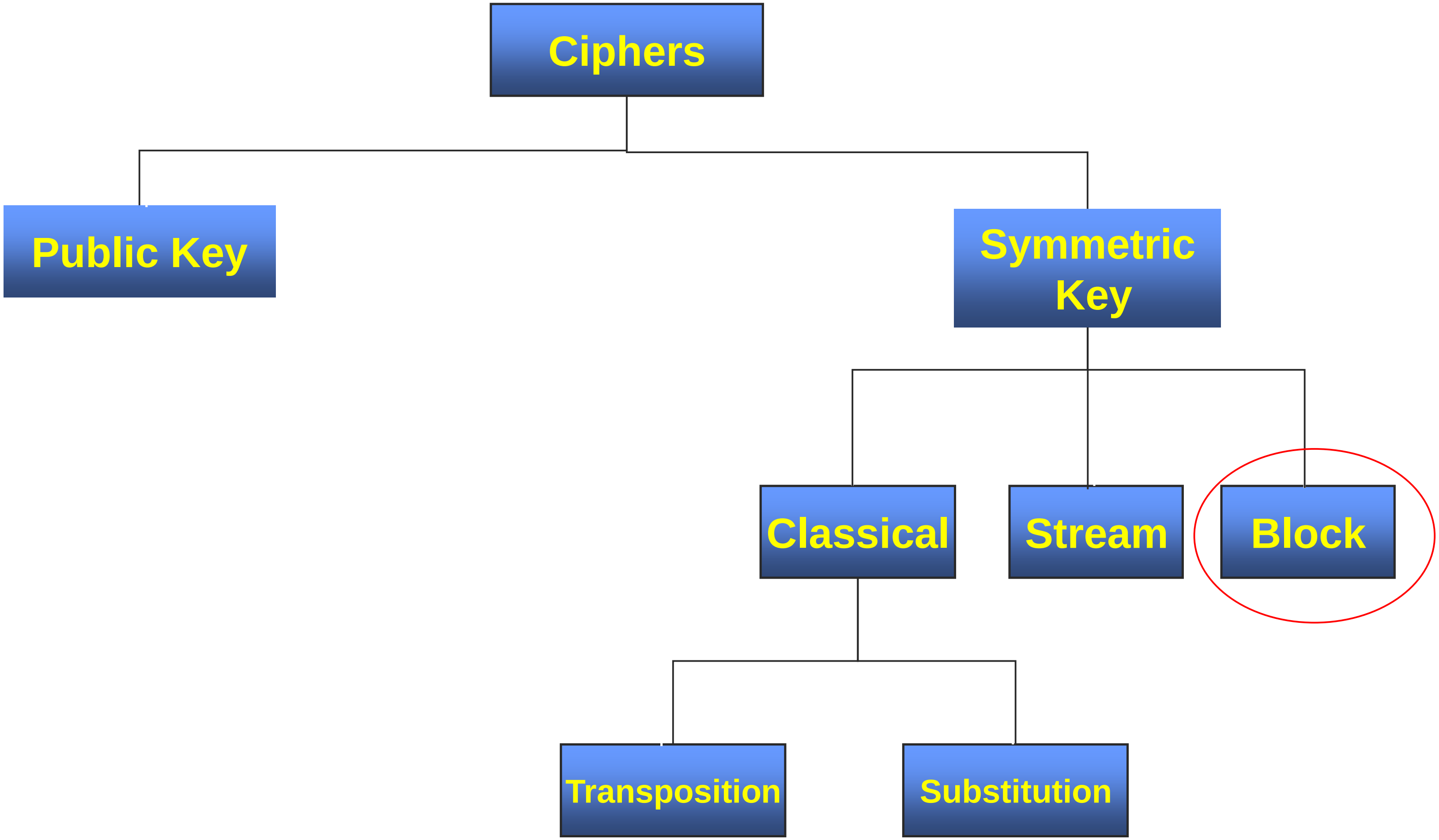
Using the two ciphers and the one bit of plaintext:

1. Find k2 using c and p
2. Find p2 using c2 and k2
3. Find k3 using p2 and d3
4. Find p3 using c3 and k3
5. Etc.

Stream Ciphers - Conclusions

- A binary keystream is mixed with the binary plaintext stream to produce a binary ciphertext stream
  - **XOR** function is used
  - The binary keystream can be generated by a **LFSR**
  - The user only has to remember how to get the key generator started

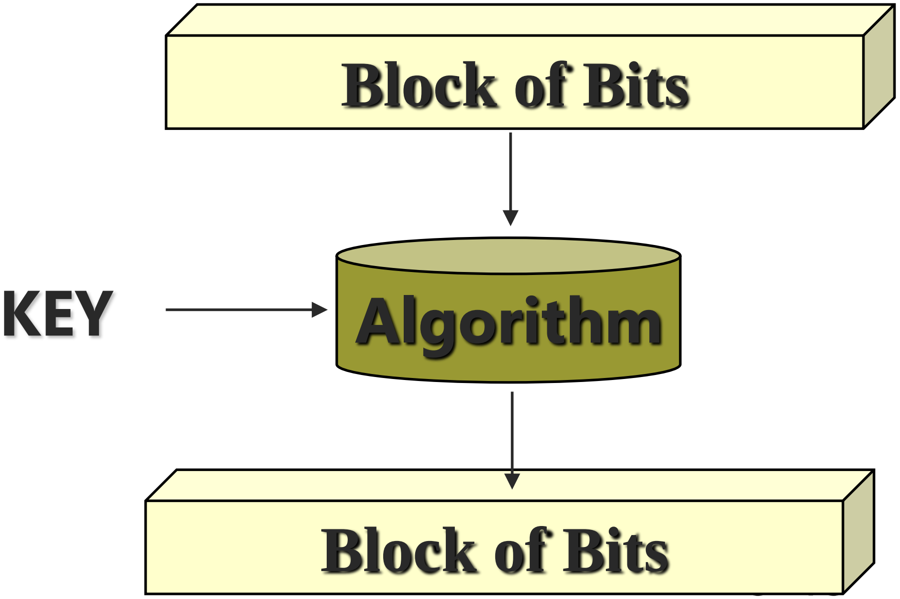
Cipher Classification



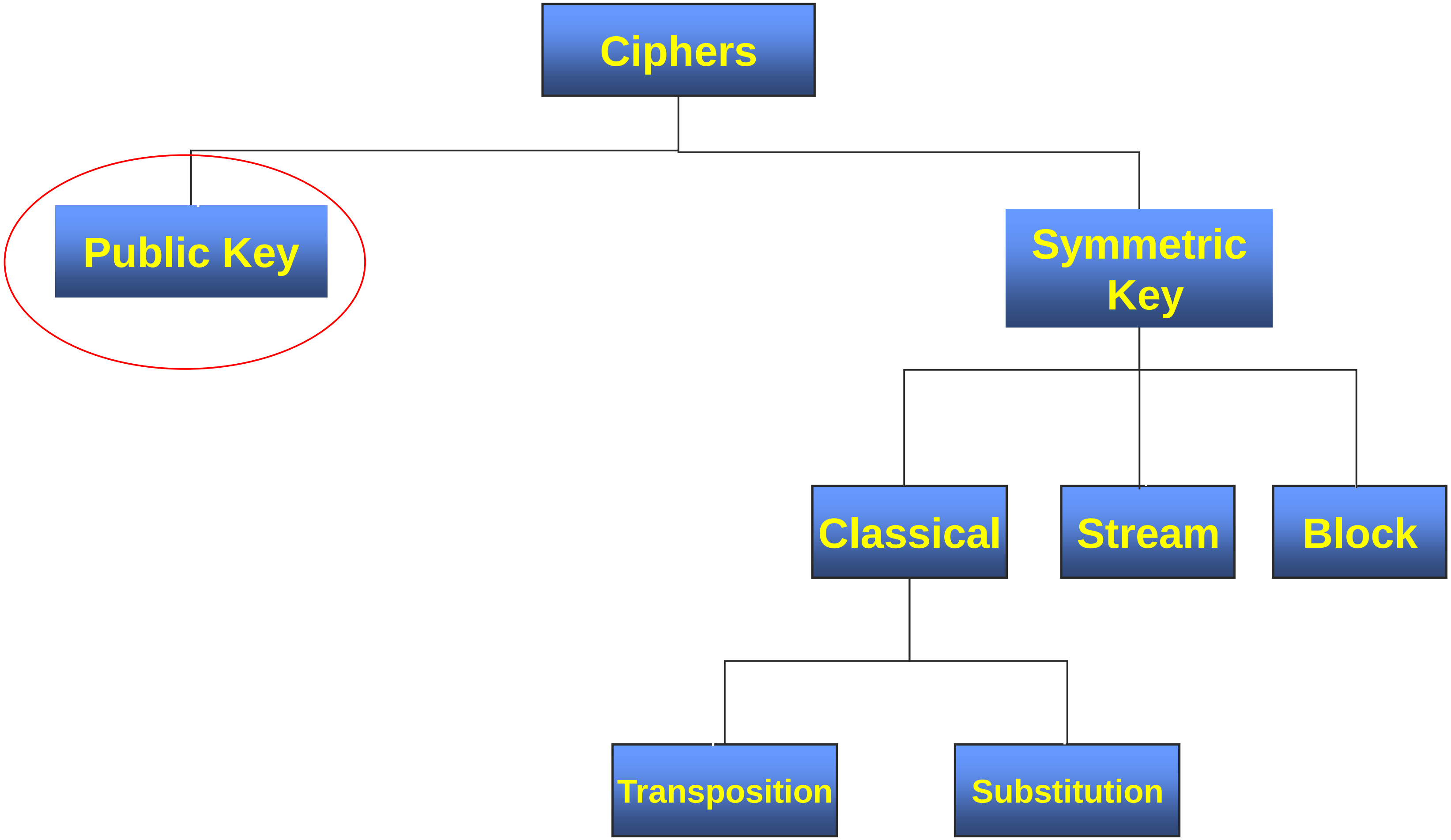
Block Cipher

- Today's most widely used ciphers
  - Define the size of a block of bits
  - Encipher the complete block at one time
  - DES, 3DES, AES, etc.
- Suitable where encryption

  is done in software



Cipher Classification



Public Key Ciphers

- Most are based on some intractable problem
  - Factoring large numbers
  - Finding the logarithm of a number

    (discrete logarithm problem)

RSA

- Invented by Rivest, Shamir, and Adleman, in 1977
- It is based on the idea of factorizing integers on their prime factors
- Used in most web browsers in the SSL protocol

The **RSA** Algorithm

1. *Bob* chooses **secrete** primes p and q and computes $n = pq$
2. *Bob* chooses e s.t. $GCD(e, (p-1)(q-1)) = 1$
3. *Bob* computes d s.t $ed \bmod ((p-1)(q-1)) = 1$
4. *Bob* makes n and e public
5. *Alice* encrypts m as $c = m^e \bmod n$ and sends c to *Bob*
6. *Bob* decrypts by computing $m = c^d \bmod n$

Note: **(e,n**) is the public key of Bob and **(d,n)** is his private key

RSA

- Encrypt: $c = m^e \bmod n$
- Decrypt: $m = c^d \bmod n$
- The value of d that works is found by

$$ed \bmod \varphi(n) = 1$$

- $\varphi(n)$ is the **Euler function** – number of integers in {1, 2, …,n} which are relatively prime to n
  - If n is prime then $\varphi(n) = n-1$
  - If p and q are relatively prime then $\varphi(pq) = \varphi(p)\,\varphi(q)$

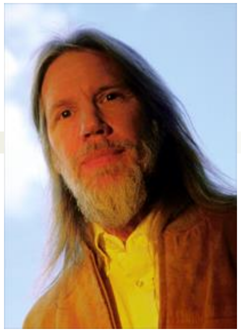
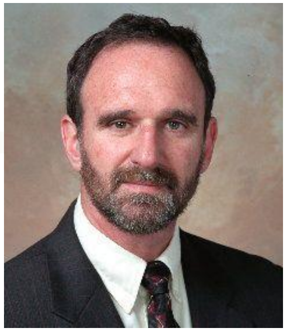
RSA

- If small numbers are chosen for p and q then the private key d is easy to guess
  1. Try to factor n to guess p and q
  2. Solve $ed \bmod ((p-1)(q-1)) = 1$ to find d
- Hence, p and q should be very large prime numbers
- Security due to the cost of factoring large numbers (hard)
- Performance is a serious issue
  - Long integer arithmetic is needed

Key Agreement

- Since public key algorithms are slow, they are often used to securely transmit keys for faster block ciphers
- However, there are protocols other than public key systems for agreeing on a common block key.
  - One of the key exchange methods developed is called the *Diffie-Hellman Key Agreement* system

Diffie-Hellman Key Agreement (1976)

- Bob and Alice want to agree on a secure key without meeting in person so they decide to use the Diffie-Hellman protocol
  - First they agree on two numbers:
    - p – a large prime number
    - g – a random number less than p
    - Both p and g are public so they can select them over an insecure channel
  - Alice selects a secret random number, ***a*** and sends Bob the value $g^a \bmod p$
  - At the same time Bob selects a secret random number, ***b*** and sends the value $g^b \bmod p$ to Alice

Diffie-Hellman Key Exchange

- Alice uses her secret number and the value Bob sent her to calculate:

$$(g^b \bmod p)^a \bmod p = k$$

- Bob uses his secret number and the value Alice sent him to calculate:

$$(g^a \bmod p)^b \bmod p = k$$

- They both end up with the same number, k
  - This is their common key

Observations

- Neither Bob nor Alice have any idea what the final key will be
- Neither Bob nor Alice shares their secret number with each other
- Eve can have access to g, p, and the values g^a mod p and g^b mod p
    - The only way she can find k is to solve (for a and b):
        - g^a mod p
        - g^b mod p
    - This is equivalent to the discrete logarithm problem:
        - Hard problem
        - Given $f : x \rightarrow g^x$ mod p

          find x from the value g^x mod p

          f is a one-way function

Example

- If Alice and Bob agree on the values: $p = 113$ $g = 23$
  - then Alice selects the secret value 4 and sends Bob the value

    $$23^4 \bmod 113 = 53$$

  - While Bob selects the secret value 11 and sends Alice the value

    $$23^{11} \bmod 113 = 27$$

  - They both calculate the common key:

Bob	Alice
$53^{11} \bmod 113 = 2$	$27^4 \bmod 113 = 2$