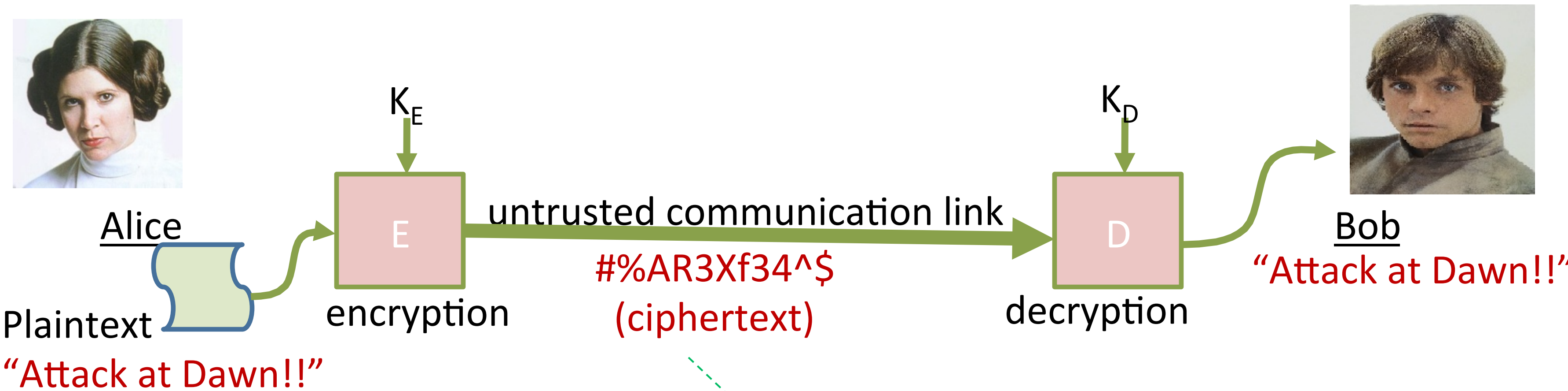
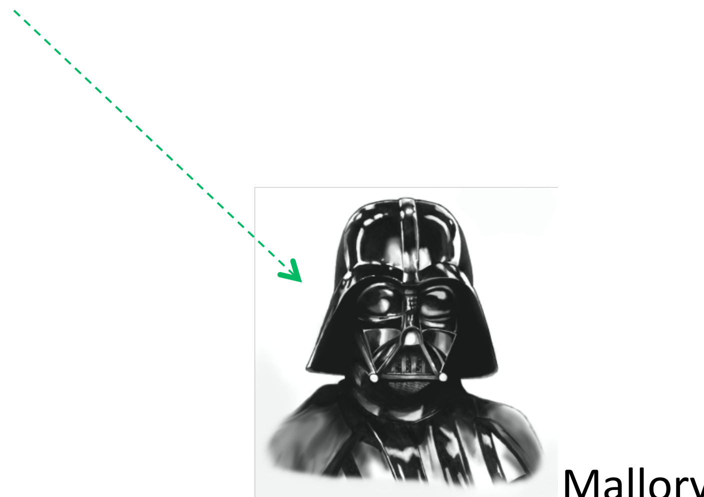
Classical Cryptography

Chester Rebeiro
IIT Madras

Ciphers



Are of 2 types
- Symmetric Key
- Asymmetric Key

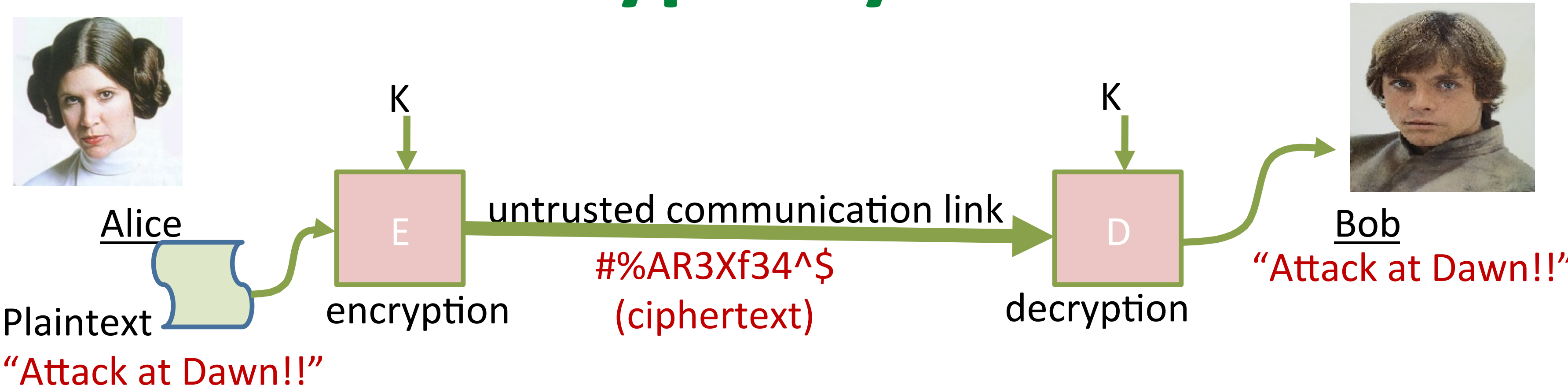


Only sees ciphertext.
cannot get the plaintext message
because she does not know the key

Ciphers

- Symmetric Key Algorithms
  - Encryption and Decryption use the same key
  - i.e. $K_E = K_D = K$ (kept secret)
  - Examples:
    - Block Ciphers : DES, AES, PRESENT, etc.
    - Stream Ciphers : A5, Grain, etc.
- Asymmetric Key Algorithms
  - Encryption and Decryption keys are different
  - $K_E \neq K_D$ (K_E kept public; K_D kept secret)
  - Examples:
    - RSA
    - ECC

A CryptoSystem



A **cryptosystem** is a five-tuple (P,C,K,E,D), where the following are satisfied:

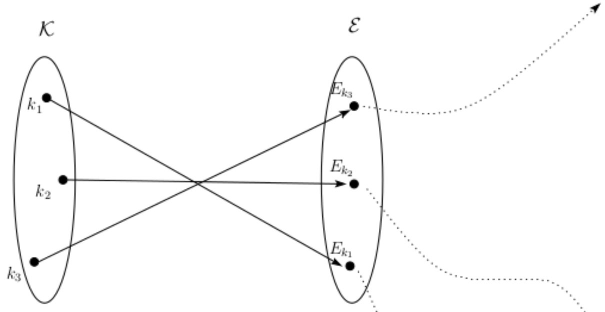
- P is a finite set of possible **plaintexts**
- C is a finite set of possible **ciphertexts**
- K, the **keyspace**, is a finite set of possible **keys**
- E is a finite set of encryption functions
- D is a finite set of decryption functions
- $\forall K \in$ K

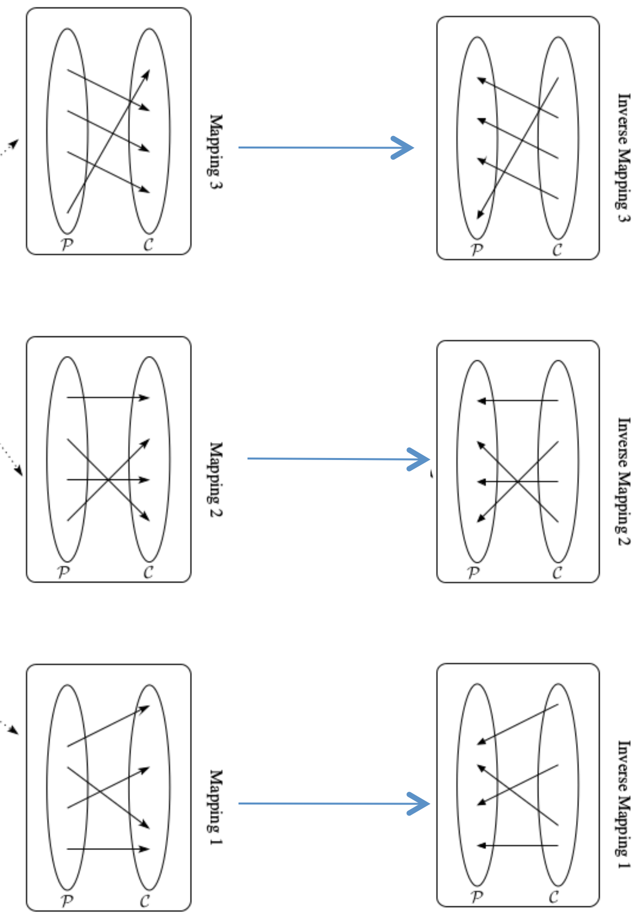
  Encryption Rule : $\exists e_K \in$ E, and

  Decryption Rule : $\exists d_K \in$ D

  such that (e_K: P→C), (d_k: C→P) and $\forall x \in$ P, $d_K(e_K(x)) = x$.

Pictorial View of Encryption





Depending on the value of the key, a mapping between the P and C is chosen.

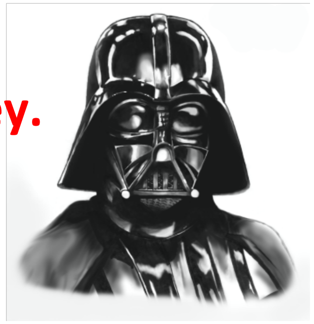
The encryption map then fixes a Mapping between C and P

Decryption is the exact inverse of encryption.

If we know the key, then mapping is easily done.

If we don't know the key, mapping is difficult. Mallory needs to first figure out which of the keys is used.

Attacker's Capabilities (Cryptanalysis)



Mallory wants to some how get information about the secret key.

- Attack models
  - ciphertext only attack
  - known plaintext attack
  - chosen plaintext attack

    Mallory has temporary access to the encryption machine. He can choose the plaintext and get the ciphertext.
  - chosen ciphertext attack

    Mallory has temporary access to the decryption machine. He can choose the ciphertext and get the plaintext.

Kerckhoff's Principle for cipher design

- Kerckhoff's Principle
  - The system is completely known to the attacker. This includes encryption & decryption algorithms, plaintext
  - only the key is secret

- Why do we make this assumption?
  - Algorithms can be leaked (secrets never remain secret)
  - or reverse engineered

history of A5/1: https://en.wikipedia.org/wiki/A5/1

Facts about e_K

- It is injective (one-to-one)
  - i.e. $e_k(x_1) = e_k(x_2)$ iff $x_1 = x_2$
  - Why?
    - If not, then Bob does not know if the ciphertext came from x_1 or x_2
- If P = C, then the encryption function is a permutation

  C is a rearrangement of P

A Shift Cipher

- Plaintext set : P = {0,1,2,3 …, 25}
- Ciphertext set : C = {0,1,2,3 …, 25}
- Keyspace : K = {0,1,2,3 …, 25}
- Encryption Rule : $e_K(x) = (x + K) \bmod 26$,
- Decryption Rule : $d_k(x) = (x - K) \bmod 26$

  where $K \in$ K and $x \in$ P
- *Note:*
  - *Each K results in a unique mapping* e_K: P→C and d_K:C→P
  - $d_k(e_K(x)) = x$
  - *The encryption/decryption rules are permutations*

CR

Using the Shift Cipher

with K=3

	0	1	2	3	4	5	6	7	8	9	10	11	12
plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
ciphertext	D	E	F	G	H	I	J	K	L	M	N	O	P

	13	14	15	16	17	18	19	20	21	22	23	24	25
plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

attackatdawn → DWWDFNDWFDZQ

Shift Cipher Mappings

- Each K results in a unique mapping e_K: P→C and d_K:C→P
- The mappings are injective (one-to-one)

$y_1 , y_2 \in C$
$d_K(y_1) \neq d_K(y_2)$

plaintext	a	b	c	d	…	x	y	z
	0	1	2	3		23	24	25
	K=8							
ciphertext	8	9	10	11		5	6	7
	I	**J**	**K**	**L**		**F**	**G**	**H**
	K=10							
ciphertext	10	11	12	13		7	8	9
	K	**L**	**M**	**N**		**H**	**I**	**J**
	K=13							
ciphertext	13	14	15	16		10	11	12
	N	**O**	**P**	**Q**		**K**	**L**	**M**

Encryption Rule

$e_K(x) = (x + K) \bmod 26,$

Decryption Rule

$d_k(x) = (x - K) \bmod 26$

How good is the shift cipher?

- A good cipher has two properties
  - Easy to compute
    - Satisfied
  - An attacker (Mallory), who views the ciphertext should not get any information about the plaintext.
    - Not Satisfied!!
    - The attacker needs at-most 26 guesses to determine the secret key ….
      - This is an exhaustive key search (known as brute force attack)

Cryptanalysis of Shift Cipher

By Brute Force...
Ciphertext : "DWWDFNDWGDZQ"

- There are only 26 possible keys, so 26 possible decryptions
- Try all of them
  - key=0, "dwwdfndwgdzq"
  - key=1, "cvvcemcvfcyp"
  - key=2, "buubdlbuebxo"
  - key=3, "attackatdawn" . . . makes sense
  - key=4, . . .
  - key=25, . . .
- Only key=3 makes sense, thus it is likely to be the key
- ... too easy!!!

Puzzle

- Cryptanalyze, assuming a shift cipher

“COMEBSDISCKCCDBYXQKCSDCGOKUOCD
VSXU”

History & Usage

- Used by Julius Caesar in 55 AD with K=3. This variant known as Caesar's cipher.
- Augustus Caesar used a variant with K=-1 and no mod operation.
- Shift ciphers are extremely simple, still used in modern times
  - By Russian Soldiers in first world war
  - Last known use in 2011 (by militant groups)

CR

Interesting Read: https://en.wikipedia.org/wiki/Caesar_cipher

Substitution Cipher

- Plaintext set : P = {a,b,c,d,…,z}
- Ciphertext set : C = {A,B,C,D,…,Z}
- Keyspace : K = {π | such that π is a permutation of the alphabets}
  - Size of keyspace is 26!
- Encryption Rule : $e_{\pi}(x) = \pi(x)$,
- Decryption Rule : $d_{\pi}(x) = \pi^{-1}(x)$

Substitution Cipher Example

Key is some permutation of the alphabets

plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
ciphertext	Z	J	H	K	F	G	I	D	B	E	A	C	P

plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext	Y	M	S	L	V	N	X	O	T	R	Q	U	W

Plaintext : “attackatdawn”

Ciphertext : “ZXXZHAXKZRY”

26! permutations possible. Thus possible keys are $26! \approx 4 \times 10^{26}$ .... rules out brute force!!!

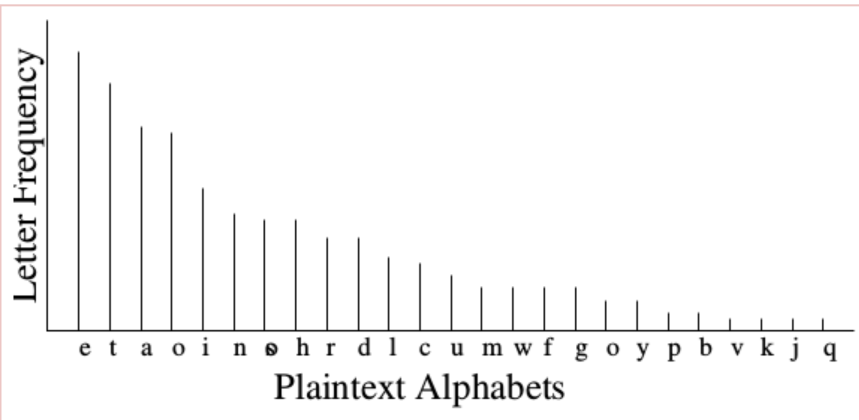
Note that the shift cipher is a special case of the substitution cipher which includes only 26 of the 26! keys

Cryptanalysis of Substitution Cipher (frequency analysis)

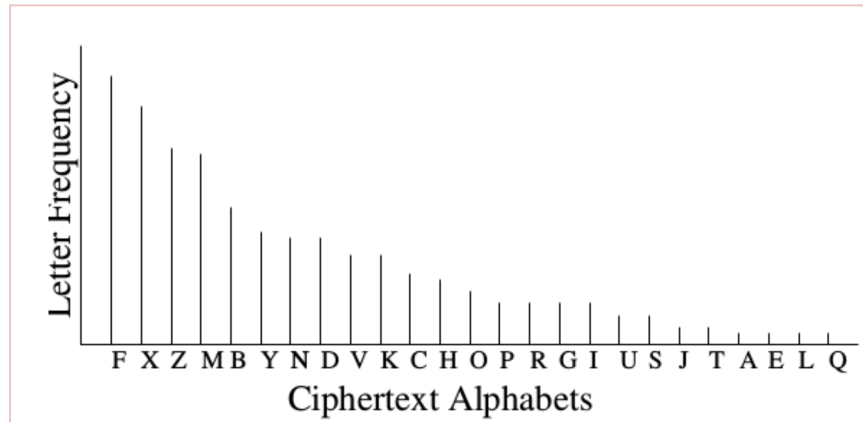
Languages do not have uniform probabilities

- Unigram probabilities of alphabets
  - E has probability 0.12 (12%)
  - T,A,O,I,N,S,H,R each have probabilities between 0.06 and 0.09
  - D,L each have probabilites around 0.04
  - C,U,M,W,F,G,Y,P,B each have probabilities between 0.015 and 0.028
  - V,K,J,X,Q,Z each occur less than 0.01
- 30 common digrams are TH, HE, IN, ER, AN, RE, AT,. . .

Cryptanalysis of Substitution Cipher
(from their frequency characteristics)



Frequency analysis of plaintext alphabets



Frequency analysis of ciphertext alphabets

Usage & Variants

- Evidence showed that it was used before Caesar's cipher
- The technique of 'substitution' still used in modern day block ciphers
- Frequency based analysis attributed to Al-kindi, an Arab mathematician (in AD 800)

Polyalphabetic Ciphers

- Problem with the simple substitution cipher :
  - A plaintext letter always mapped to the same ciphertext letter (mono alphabetic)

    eg. ‘Z’ always corresponds to plaintext ‘a’
  - facilitating frequency analysis

- A variation (polyalphabetic cipher)
  - A plaintext letter may be mapped to multiple ciphertext letters
  - eg. ‘a’ may correspond to ciphertext ‘Z’ or ‘T’ or ‘C’ or ‘M’
  - More difficult to do frequency analysis (but not impossible)

  - Example : Vigenere Cipher, Hill Cipher

Vigenère Cipher

- Let the key be (2,5,8,7,9,12) of size 6
- Let the message to be encrypted be "attackatdawn"
- Convert message to integers modulo 26
  - "attackatdawn" becomes (0, 19, 19, 0, 2, 10, 0, 19, 3, 0, 22, 13)
- To encrypt, group them in terms of 6 and add the corresponding key

$|\text{keyspace}| = 26^m$
(where m is the length of the key)

plaintext (x)	a	t	t	a	c	k	a	t	d	a	w	n
	0	19	19	0	2	10	0	19	3	0	22	13
key (k)	2	5	8	7	9	12	2	5	8	7	9	12
(x + k) mod 26	2	23	1	7	11	22	2	24	11	7	5	25
ciphertext	C	X	B	I	K	W	C	Y	K	H	F	Z

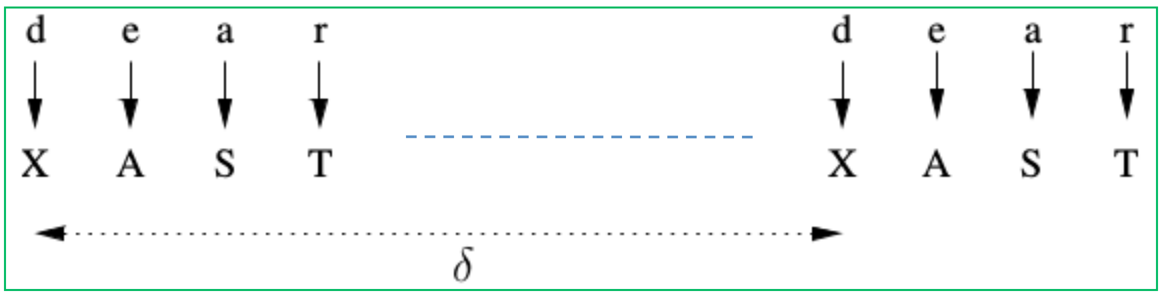
Cryptanalysis of Vigenère Cipher

- Frequency analysis more difficult
  (but not impossible)
- Attack has two steps

1. Determine the length m of the key
2. Determine $K = (k_1, k_2, k_3, \cdots k_m)$ by finding each k_i separately

Determining Key Length *(Kaisiki Test)*

- **Kasiski test** by Friedrich Kasiski in 1863
- Let *m* be the size of the key
- observation: two identical plaintext segments will encrypt to the same ciphertext when they are δ apart and $(m \mid \delta)$



when (m divides δ)

- If several such δs are found (i.e. $\delta_1, \delta_2, \delta_3, \ldots.$) then
  - $m|\delta_1, m|\delta_2, m|\delta_3, \ldots.$
  - Thus m divides the gcd of $(\delta_1, \delta_2, \delta_3, \ldots.)$

Example

```
Key:         ABCDABCDABCDABCDABCDABCDABCD
Plaintext:   CRYPTOISSHORTFORCRYPTOGRAPHY
Ciphertext:  CSASTPKVSIQUTGQUCSASTPIUAQJB
```

The distance between the two "CSASTP" is 16.
The key length is either 16,8,4,2, or 1.

CR

Increasing Confidence of Key Length *(Index of Coincidence)*

- Consider a multi set of letters of size N

  say ***s*** = *{a,b,c,d,a,a,e,f,e,g,.....}*

- Probability of picking two 'a' characters (without replacement) is

$$\frac{n_0}{N} \times \frac{n_0 - 1}{N - 1}$$

n_0 : Number of occurrences of 'a' in S

probability the first pick is 'a'

probability the second pick is 'a'

- Sum of probabilities of picking two similar characters is

$$I_c = \sum_{i=0}^{25} \frac{n_i(n_i - 1)}{N(N-1)}$$

index of coincidence

Index of Coincidence

- Consider a random permutation of the alphabets (as in the substitution cipher)

s = {a,b,c,d,a,a,e,f,e,g,.....} → *S = {X,M,D,F,X,X,Z,G,Z,J,.....}*

- Note that : $n_a = n_X$; thus the value of I_c remains unaltered

- Number of occurrence of an alphabet in a text depends on the language, thus each language will have a unique I_c value

English	0.0667	French	0.0778
German	0.0762	Spanish	0.0770
Italian	0.0738	Russian	0.0529

Index of Coincidence, NSA Declassified Document
https://www.nsa.gov/public_info/_files/friedmanDocuments/Publications/FOLDER_231/41760429079956.pdf

Modular Arithmetic

Modular Arithmetic

slides in Mathematical Background

Affine Cipher

- A special case of substitution cipher
- Encryption: $y = ax + b \pmod{26}$
- Decryption: $x = (y - b)a^{-1} \pmod{26}$
  - plaintext : $x \in \{0,1,2,3, \ldots 25\}$
  - ciphertext : $y \in \{0,1,2,3, \ldots 25\}$
  - key : (a,b)
    - where a and b $\in \{0,1,2,3, \ldots 25\}$ and
    - $\gcd(a, 26) = 1$

why need this condition?

- Example: a=3, b=5
  - Encryption: x=4; y = (3*4 + 5)mod 26 = 17
  - Decryption: $x = (y - b)a^{-1} \bmod 26$

$a.a^{-1} = 1 \bmod 26$. The inverse exists only if a and 26 are coprime

$a^{-1} = 9$ (Note that 3 * 9 mod 26 = 1)

(17 - 5)*9 mod 26 = 4

When gcd(a,26) ≠ 1?

- Let *gcd(a, 26) = d > 1*
  - then *d|a* and *d|26*   *(i.e. d mod 26 = 0)*
  - y = ax + b mod 26

    Let ciphertext y = b ;   ax = 0 mod 26

    In this case x can have two decrypted values : 0 and d.

    Thus the function is not injective…. cannot be used for an encryption

What is the ciphertext when (1) $x_1 = 1$ and (2) $x_2 = 14$ are encrypted with the Affine cipher with key (4, 0)?

Usage & Variants of Affine Cipher

- Ciphers built using the Affine Cipher
  - Caesar's cipher is a special case of the Affine cipher with a = 1
  - Atbash
    - b = 25, $a^{-1} = a = 25$
    - Encryption : y = 25x + 25 mod 26
    - Decryption : x = 25x + 25 mod 26

Encryption function same as decryption function

CR

Hill Cipher

- Encryption: y = xK (mod 26)
- Decryption: x = yK^{-1} (mod 26)
  - plaintext : $x \in \{0,1,2,3, ... 25\}$
  - ciphertext : $y \in \{0,1,2,3, ... 25\}$
  - key : K is an invertible matrix
- example

$$K = \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \qquad K^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} \qquad K \bullet K^{-1} = 1 \bmod 26$$

h i l l

plaintext

h i l l

(7,8)(11,11)

encryption

$$\begin{bmatrix} 7 & 8 \end{bmatrix} \times \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} (\bmod 26) = \begin{bmatrix} 23 & 8 \end{bmatrix}$$

decryption

$$\begin{bmatrix} 23 & 8 \end{bmatrix} \times \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} (\bmod 26) = \begin{bmatrix} 7 & 8 \end{bmatrix}$$

h i l l → (7,8)(11,11) → (23,8)(24,9) → XIYJ

plaintext ciphertext

Cryptanalysis of Hill Cipher

- ciphertext only attack is difficult
- known plaintext attack

$$(7,8)(11,11) \times \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \longrightarrow (23,8)(24,9)$$

known plaintext — corresponding ciphertext

Form equations and solve to get the key

$$7k_{11} + 8k_{21} = 23 \qquad 7k_{12} + 8k_{22} = 8$$

$$11k_{11} + 11k_{21} = 24 \qquad 11k_{12} + 11k_{22} = 9$$

Permutation Cipher

- Ciphers we seen so far were substitution ciphers
  - Plaintext characters substituted with ciphertext characters

h i l l → XIYJ

plaintext → ciphertext

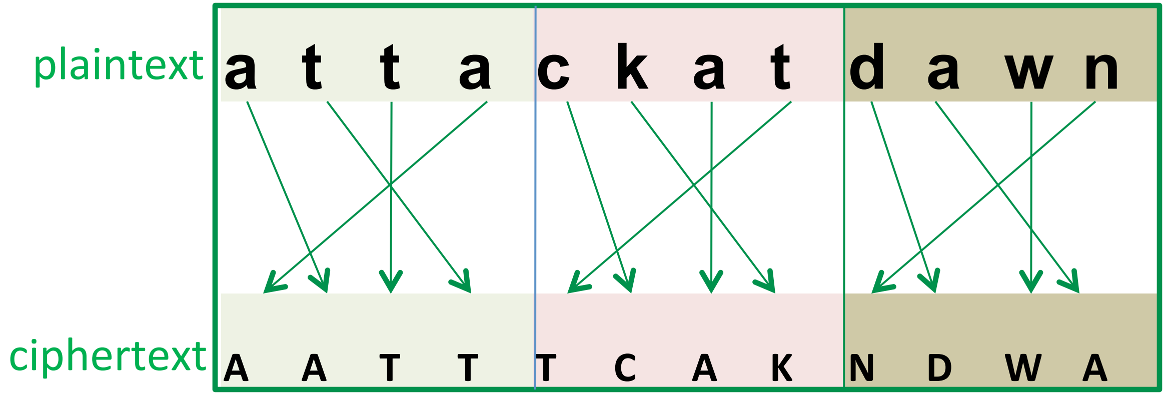
- Alternate technique : permutation
  - Plaintext characters re-ordred by a random permutation

h i l l → LIHI

plaintext → ciphertext

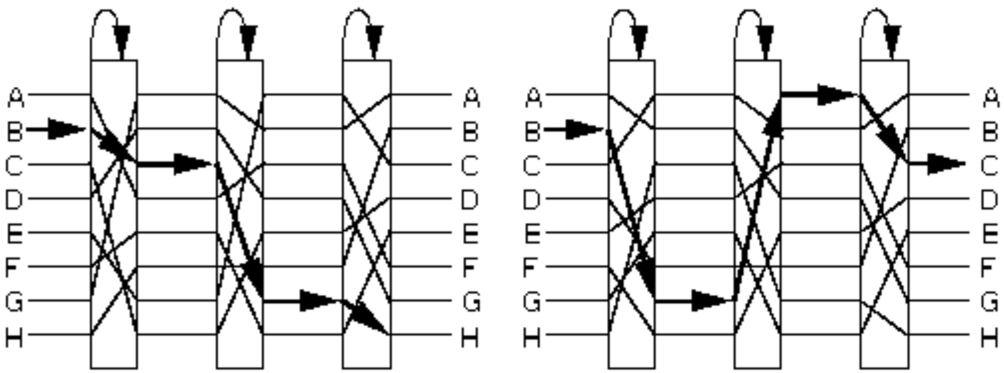
Permutation Cipher

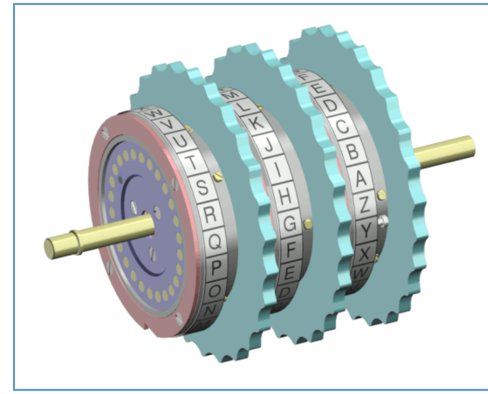
- Example plaintext : attackatdawn
  - key : (1,3,2,0) here is of length 4 and a permutation of (0,1,2,3)
    - It mean's 0^{th} character in plaintext goes to 1^{st} character in ciphertext (and so on…)



- cryptanalysis : 4! possibilities

Rotor Machines (German Enigma)

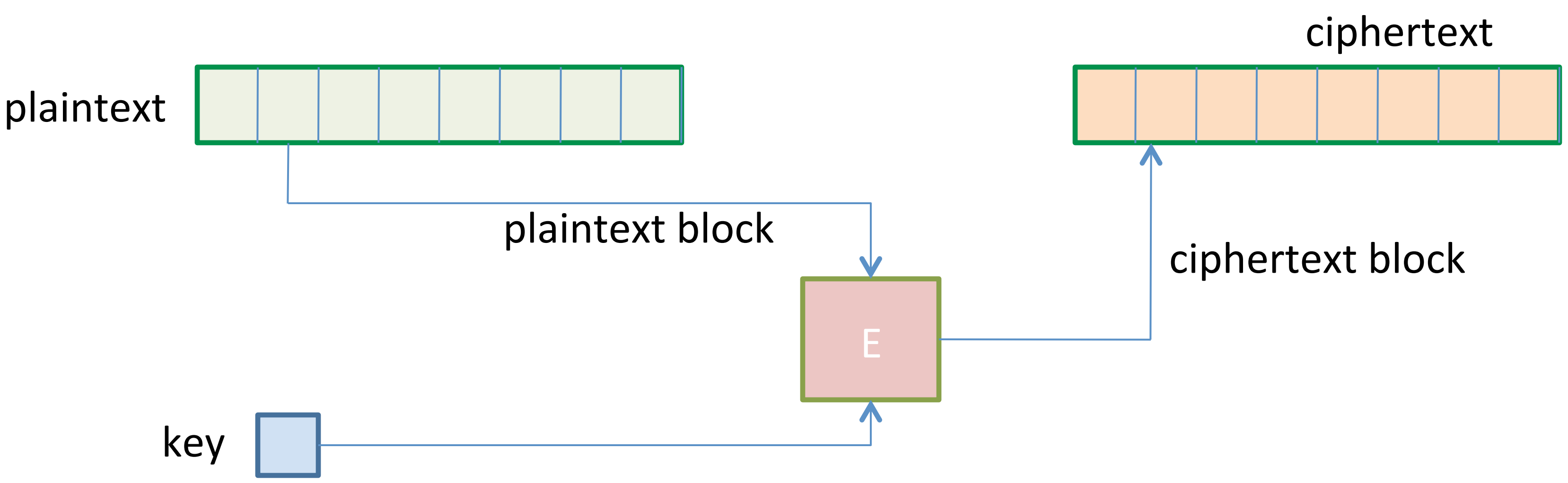




- Each rotor makes a permutation
  - Adding / removing a rotor would change the ciphertext
- Additionally, the rotors rotates with a gear after a character is entered
- Broken by Alan Turing

Block Ciphers

- General principal of all ciphers seen so far
  - Plaintext divided into blocks and each block encrypted with the same key
  - Blocks can vary in length starting from 1 character

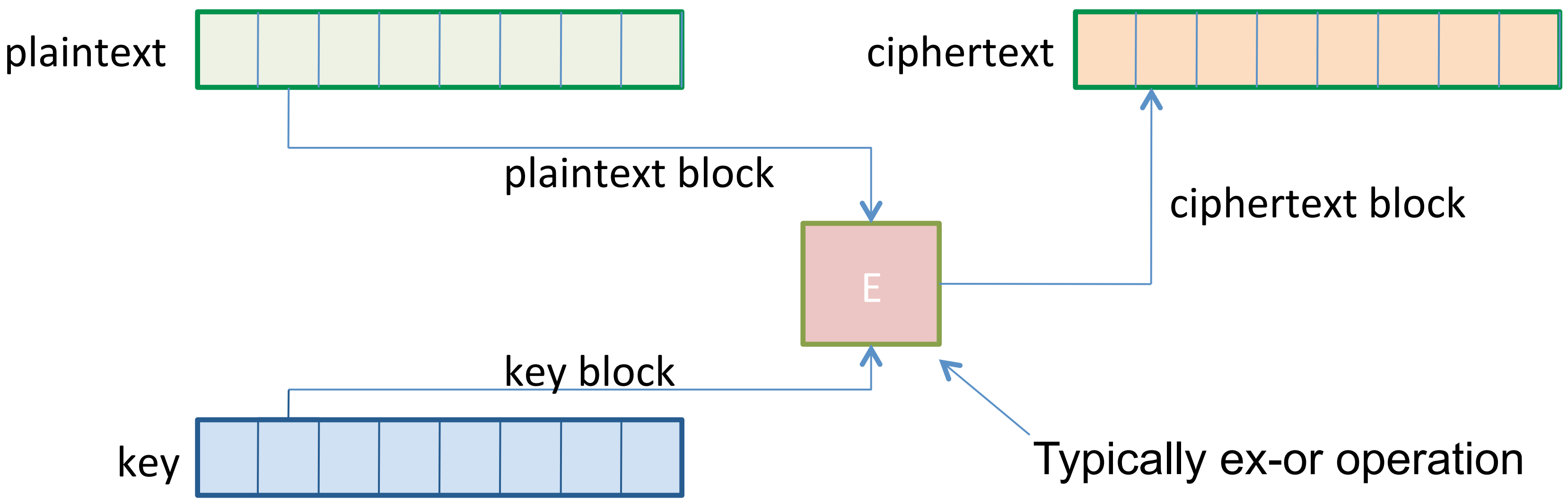


- examples: substitution ciphers, polyalphabetic ciphers, permutation ciphers, etc.

Stream Ciphers

Typically a bit, but can also more than a bit

- Each block of plaintext is encrypted with a different key



Formally, $y = y_1 y_2 y_3 \ldots = e_{k_1}(x_1) e_{k_2}(x_2) e_{k_3}(x_3) \ldots$

Observe: the key should be variable length… we call this a key stream.

Stream Ciphers (how they work)

stream cipher output :

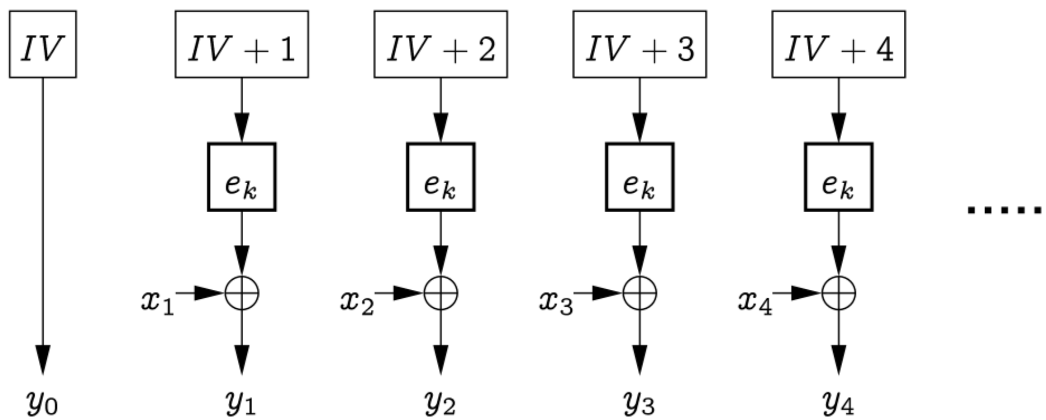
$$y = y_1 y_2 y_3 \cdots$$

$$y_1 = x_1 \oplus k_1 ; y_2 = x_2 \oplus k_2 ; y_3 = x_3 \oplus k_3 , \ldots .$$

How to generate the i^th key : $k_i = f_i(K, k_1, k_2, k_3, \ldots, k_{i-1})$

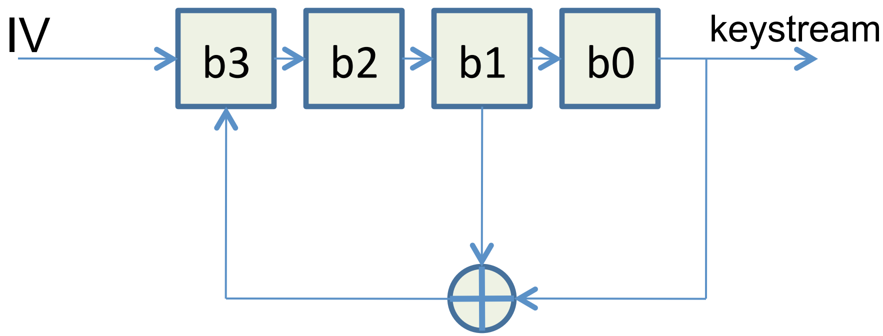
i^{th} key is a function of K and the first i-1 plaintexts

$k_1, k_2, k_3, \ldots, k_i$ Is known as the keystream



Generating the keystream in practice

- Using LFSRs (Linear feedback shift registers)



Initialization Vector →

b3	b2	b1	b0
1	0	0	0
0	1	0	0
0	0	1	0
1	0	0	1
1	1	0	0
0	1	1	0
1	0	1	1
0	1	0	1
1	0	1	0
1	1	0	1
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
1	0	0	0