CO-INS:Information and Network Security

UNIT-II (Part-I)

Course Instructors:
Soma Saha
Veerendra Srivastava

Soma Saha (PhD)

Department of Computer Engineering
SGSITS Indore, India

March 4, 2021

UNIT-II (Part-I): Learning Objectives

Upon completion of this unit, you should be able to

LO1 Define the terms and concepts of symmetric-key ciphers

LO2 Discuss the two broad categories of traditional symmetric-key ciphers with focus on different cipher cryptanalysis

LO3 Show the idea behind stream ciphers and block ciphers

Cryptography: Symmetric-Key Cipher

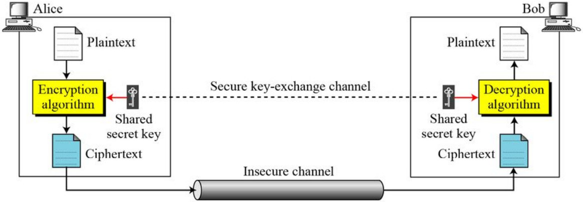


Figure 1: Symmetric-Key Encipherment.

- $c = E_s(p, k)$
- $p = D_s(c, k)$
- D_s = Decryption function (symmetric)

- c = ciphertext
- p = plaintext
- k = secret key
- E_s = Encryption function (symmetric)

Soma Saha

Symmetric-Key Encipherment: Message exchange Prove

- We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates p_1; we prove that p_1 = p:

Symmetric-Key Encipherment: Message exchange Prove

- We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates p_1; we prove that $p_1 = p$:

- $\mathit{Alice} : c = E_s(p, k)$
- Bob: $p_1 = D_s(c, k) = D_s(E_s(p, k), k) = p$

Symmetric-Key Encipherment: Message exchange Prove

- We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates p_1; we prove that $p_1 = p$:

- $Alice : c = E_s(p, k)$
- Bob: $p_1 = D_s(c, k) = D_s(E_s(p, k), k) = p$

- **Kerckhoff's Principle**: A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.
- Kerckhoffs's principle was reformulated (or possibly independently formulated) by American mathematician **Claude Shannon** as ”the enemy knows the system”, i.e., ”one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them”. In that form, it is called **Shannon's maxim**.(Source: wikipedia)

Symmetric-Key Encipherment: How to compute #keys?

- If there are *m* people in a group who need to communicate with each other, how many keys are needed?

Symmetric-Key Encipherment: How to compute #keys?

- If there are *m* people in a group who need to communicate with each other, how many keys are needed?

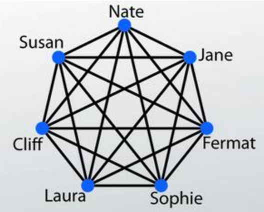


Figure 2: Example scenario.

Cryptography-Cryptanalyis-Cryptology

- Cryptography - Science and art of creating secret codes.
- Cryptanalyis - Science and art of breaking those codes.
- The study of cryptanalysis helps us create better secret codes.

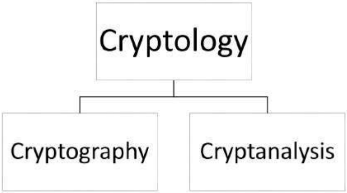
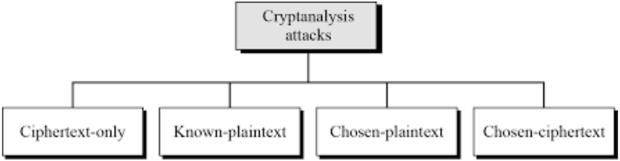


Figure 3: Cryptography-Cryptanalyis-Cryptology.

Cryptanalyis Attacks

- Classification based on encryption techniques:
  1. Ciphertext-only attack
  2. Known-plaintext attack
  3. Chosen-Plaintext attack
  4. Chosen-Ciphertext attack
  5. Chosen-text attack



Cryptanalysis attacks

Ciphertext-only

Known-plaintext

Chosen-plaintext

Chosen-ciphertext

Figure 4: Classification of cryptanalysis attacks based on encryption techniques.

Ciphertext-Only Attack

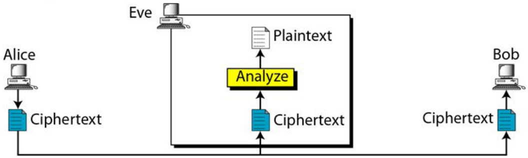


Figure 5: Ciphertext-only attack.

- Adversary, Eve/Darth has access to only some ciphertext.
- Assumption: Eve/Darth knows the algorithm and can intercept the ciphertext.

Ciphertext-Only Attack: various methods

- **Brute-Force Attack/Exhaustive-key-search Attack:** Eve/Darth tries to use all possible keys.
- Assumption:
  - Eve/Darth knows the algorithm.
  - Eve/Darth knows the key domain (the list of all possible keys).
- Application: Using the intercepted cipher, Eve/Darth decrypts the ciphertext with every possible key until the plaintext makes sense.
- Prevention: The number of possible keys must be very large.

Ciphertext-Only Attack: various methods..cont..1

- **Statistical Attack:** The cryptanalyst can benefit from some inherent characteristics of the plaintext language to launch a **statistical attack**.
- Example:
  - The letter E is the most frequently use letter in English text.
  - The cryptanalyst finds the mostly-used character in the ciphertext and assumes that the corresponding plaintext character is E.
  - After finding a few pairs, the analyst can find the key and use it to decrypt the message.
- Prevention: The cipher should hide the characteristics of the languagee.

Ciphertext-Only Attack: various methods..cont..2

- **Pattern Attack:** Some ciphers may hide characteristics of the language, but may create some patterns in the ciphertext. A cryptanalist may use a **pattern attack** to break the cipher.
- Prevention: Important to use ciphers that make the ciphertext as random as possible.

Known-Plaintext Attack

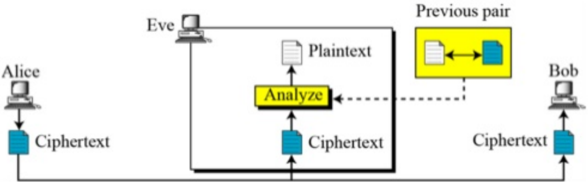


Figure 6: Known-Plaintext Attack.

- **Known-Plaintext Attack:** Eve/Darth has access to some plaintext/ciphertext pairs in addition to the intercepted ciphertext that she/he wants to break.
  - The plaintext/ciphertext pairs have been collected earlier.
  - This type of attacks are less likely to happen, because…

Chosen-Plaintext Attack

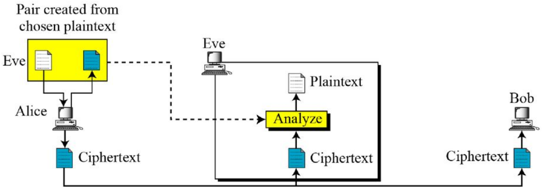


Figure 7: Chosen-Plaintext Attack.

- **Chosen-Plaintext Attack:** This attack is similar to the known-plaintext attack, but the plaintext/ciphertext pairs have been chosen by the attacker herself.
  - Example: If Eve/Darth has access to Alice's computer, she/he can choose some plaintext and intercept the captured ciphertext.
  - She/he does not have the key because the key is normally **embedded in the software** used by the sender.
  - Easy to implement, but less likely to happen, because..

Soma Saha

Chosen-Ciphertext Attack

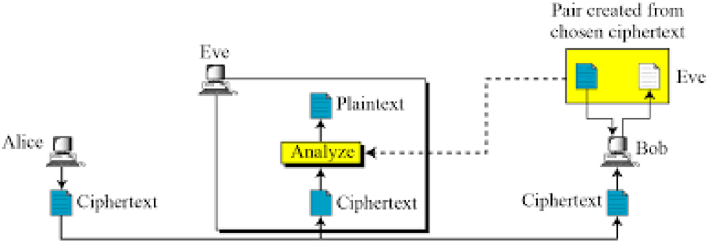
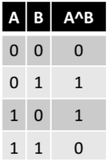


Figure 8: Chosen-Ciphertext Attack.

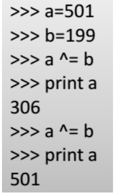
- **Chosen-ciphertext attack** is similar to the chosen-plaintext attack, except that Eve/Darth chooses some ciphertext and decrypts it to form a ciphertext/plaintext pair.
- This can happen if Eve/Darth has access to Bob's computer.

Hypothetical bad symmetric encryption algorithm: XOR (Content courtesy: Tyler Bletsch, Duke Univ.)

- *A* lot of encryption algorithms rely on properties of XOR
  - Can think of $A^{\wedge}B$ as "Flip a bit in *A* if corresponding bit in *B* is 1"
  - If you XOR by same thing twice, you get the data back
  - XORing by a random bit string yields NO info about original data
    - Each bit has a 50% chance of having been flipped
- Could consider XOR itself to be a symmetric encryption algorithm (but it seems dreadful at it!) - can be illustrative to explore
- Simple XOR encryption algorithm:
  - $E(p,k) = p^{\wedge}k$ (keep repeating *k* as often as needed to cover *p*)
  - $D(c,k) = c^{\wedge}k$ (same algorithm both ways!)

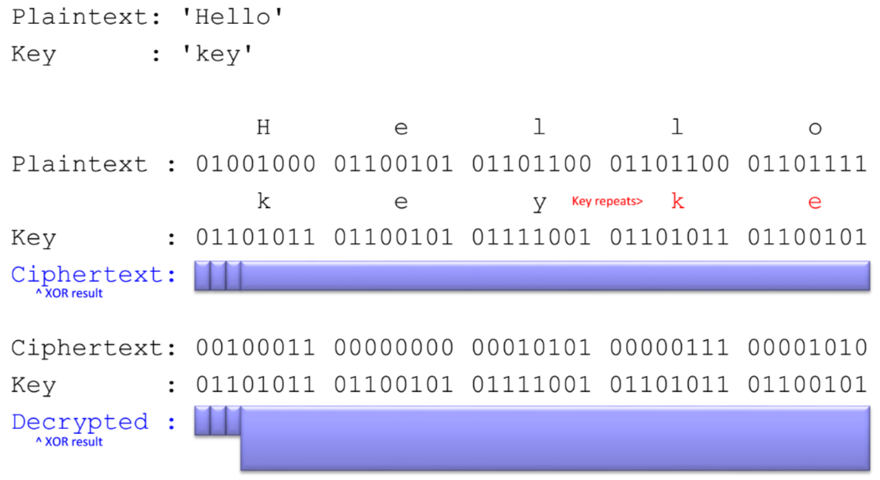


A	B	A^B
0	0	0
0	1	1
1	0	1
1	1	0



```
>>> a=501
>>> b=199
>>> a ^= b
>>> print a
306
>>> a ^= b
>>> print a
501
```

XOR “encryption” demo (Content courtesy: Tyler Bletsch, Duke Univ.)



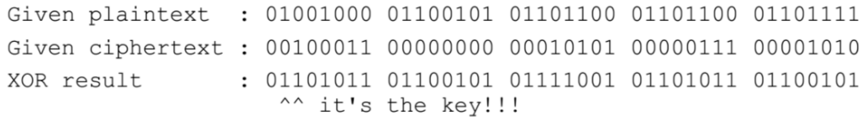
```
Plaintext: 'Hello'
Key      : 'key'

                H          e          l          l          o
Plaintext : 01001000  01100101  01101100  01101100  01101111
                k          e          y  Key repeats>  k          e
Key       : 01101011  01100101  01111001  01101011  01100101
Ciphertext:
    ^ XOR result

Ciphertext: 00100011  00000000  00010101  00000111  00001010
Key       : 01101011  01100101  01111001  01101011  01100101
Decrypted :
    ^ XOR result
```

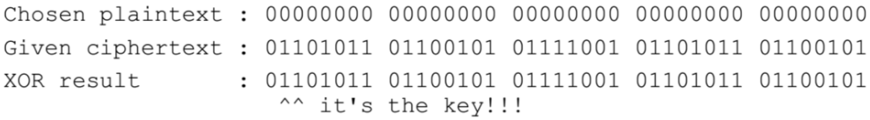
Attacking XOR (Content courtesy: Tyler Bletsch, Duke Univ.)

Figure 9: Known-Plaintext Attack:



```
Given plaintext   : 01001000 01100101 01101100 01101100 01101111
Given ciphertext  : 00100011 00000000 00010101 00000111 00001010
XOR result        : 01101011 01100101 01111001 01101011 01100101
                     ^^ it's the key!!!
```

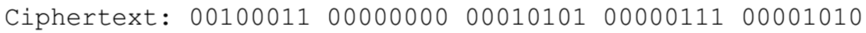
Figure 10: Chosen-Plaintext Attack:



```
Chosen plaintext  : 00000000 00000000 00000000 00000000 00000000
Given ciphertext  : 01101011 01100101 01111001 01101011 01100101
XOR result        : 01101011 01100101 01111001 01101011 01100101
                     ^^ it's the key!!!
```

Attacking XOR..contd.. (Content courtesy: Tyler Bletsch, Duke Univ.)

Figure 11: Ciphertext-Only Attack:



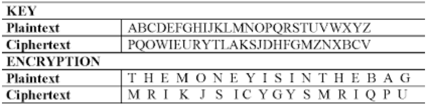
```
Ciphertext: 00100011 00000000 00010101 00000111 00001010
```

- “I assume the plaintext had ASCII text with lowercase letters, and in all such letters bit 6 is 1, but none of the ciphertext has bit 6 set, so I bet the key is most/all lower case letters”
- “The second byte is all zeroes, which means the second byte of the key and plaintext are equal”
- etc…

** **Conclusion: XOR is a dreadful encryption algorithm**

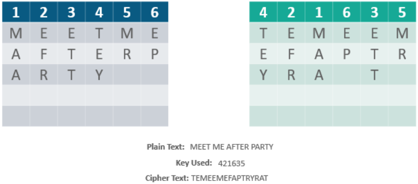
Categories of Traditional Ciphers

- Traditional Symmetric-Key Ciphers
  - **Substitution Ciphers:** We replace one symbol in the ciphertext with another symbol.
  - **Transposition Ciphers:** We reorder the position of symbols in the plaintext.



KEY	
Plaintext	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext	PQOWIEURYTLAKSJDHFGMZNXBCV
ENCRYPTION	
Plaintext	T H E M O N E Y I S I N T H E B A G
Ciphertext	M R I K J S IC YGY S M R I Q P U

Figure 12: Substitution Cipher.



1	2	3	4	5	6
M	E	E	T	M	E
A	F	T	E	R	P
A	R	T	Y		

4	2	1	6	3	5
T	E	M	E	E	M
E	F	A	P	T	R
Y	R	A		T	

Plain Text: MEET ME AFTER PARTY

Key Used: 421635

Cipher Text: TEMEEMEFAPTRYRAT

Figure 13: Transposition Cipher.

Soma Saha

Substitution Ciphers: Types

- **Substitution Cipher**

1. **Mono-alphabetic** : It only uses one alphabet to substitute.
   - Additive/Shift/Caeser
   - Multiplicative
   - Affine
   - Monoalphabetic Substitution Cipher

2. **Poly-alphabetic**: It may use two or more alphabets to substitute.
   - Auto-Key Cipher
   - Vigenere Cipher
   - Playfair cipher
   - Hill Cipher
   - One Time Pad
   - Rotor Cipher

Additive Cipher/Shift Cipher

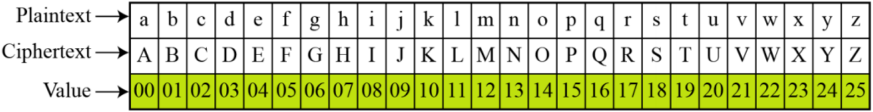
- Each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
- For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.
- **Caesar Cipher** is a particular case (for k = 3).
  - ***Julius Caesar***, who used it in his private correspondence; he used fixed $\#3$ for shifting.

Additive Cipher/Shift Cipher

- Each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
- For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.
- **Caesar Cipher** is a particular case (for k = 3).
  - ***Julius Caesar***, who used it in his private correspondence; he used fixed #3 for shifting.
- **Mathematically**,
  - $Z_{26} = \{0, 1, 2, ..., 24, 25\}$
  - $P = C = K = Z_{26}$
  - For $k \in K$,
    $E_k(x) = (x + k)$ mod 26 for $x \in P$
    $D_k(y) = (y - k)$ mod 26 for $y \in C$

Additive/Shift Cipher: Example

- The plaintext is ordinary English text.
- Correlation between alphabetic characters and integer: $A = 0, B = 1, ..., Y = 24, Z = 25$.



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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Additive/Shift Cipher: Working Principle

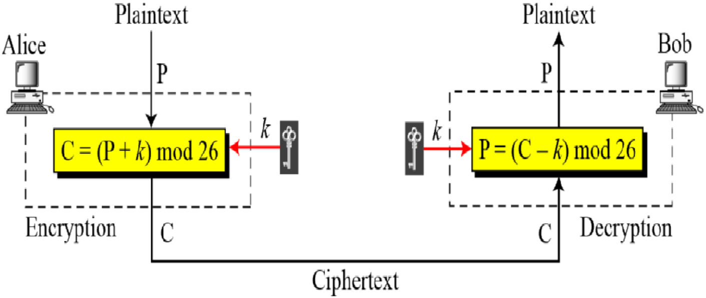
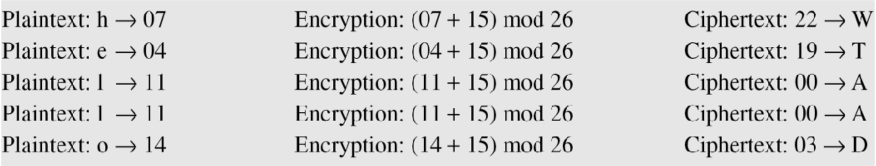


Figure 14: Additive/Shift Cipher (**Note: When the cipher is additive, the plaintext, ciphertext, and key are integers in** Z_{26}).

Additive/Shift Cipher: Encryption

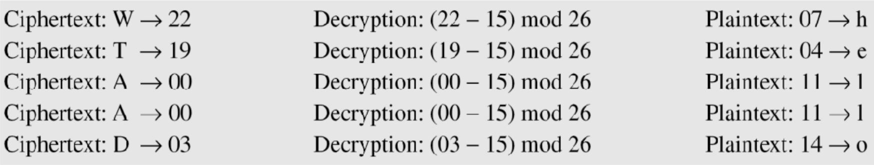
- key $k = 15$
- Plaintext is “hello”
- Corresponding sequence of integers:
  07 04 11 11 14
- we add 15 (key) to each value (reducing modulo 26):
  22 19 00 00 03
- Convert the sequence of integers to alphabetic characters:
  W T A A D



Plaintext: h $\rightarrow$ 07	Encryption: $(07 + 15) \bmod 26$	Ciphertext: 22 $\rightarrow$ W
Plaintext: e $\rightarrow$ 04	Encryption: $(04 + 15) \bmod 26$	Ciphertext: 19 $\rightarrow$ T
Plaintext: l $\rightarrow$ 11	Encryption: $(11 + 15) \bmod 26$	Ciphertext: 00 $\rightarrow$ A
Plaintext: l $\rightarrow$ 11	Encryption: $(11 + 15) \bmod 26$	Ciphertext: 00 $\rightarrow$ A
Plaintext: o $\rightarrow$ 14	Encryption: $(14 + 15) \bmod 26$	Ciphertext: 03 $\rightarrow$ D

Additive/Shift Cipher: Decryption

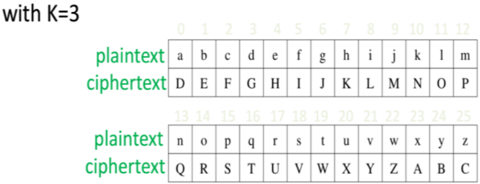
- ciphertext : “WTAAD”
- convert the ciphertext to sequence of integers:
  22 19 00 00 03
- subtract 15 from each value (reducing modulo 26):
  07 04 11 11 14
- convert the sequence of integers to alphabetic characters: ”hello”



Ciphertext: W → 22	Decryption: (22 – 15) mod 26	Plaintext: 07 → h
Ciphertext: T → 19	Decryption: (19 – 15) mod 26	Plaintext: 04 → e
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: D → 03	Decryption: (03 – 15) mod 26	Plaintext: 14 → o

Caesar Cipher

- **Caesar Cipher** is the earliest known (and the simplest).
- It involves replacing each letter of the alphabet with the letter standing three places further down. This is then wrapped around on itself when the end is reached.
- For example:



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

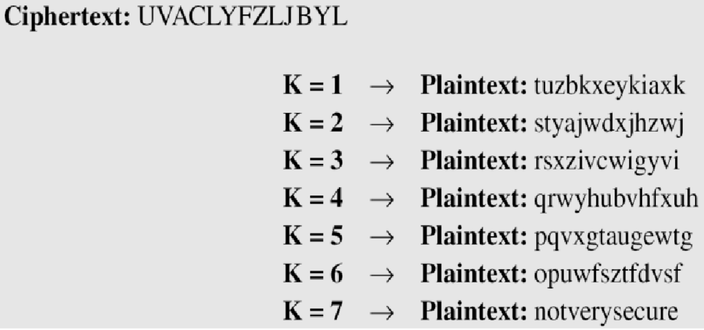
attackatdawn → DWWDFNDWFDZQ

Additive/ Shift Cipher: Example

- Eve/Darth has intercepted the ciphertext **“UVACLYFZLJBYL”**. Show how she/he can use a brute-force attack to break the cipher!!

Additive/ Shift Cipher: Example

- Eve/Darth has intercepted the ciphertext **"UVACLYFZLJBYL"**. Show how she/he can use a brute-force attack to break the cipher!!



Ciphertext: UVACLYFZLJBYL

K = 1 → **Plaintext:** tuzbkxeykiaxk

K = 2 → **Plaintext:** styajwdxjhzwj

K = 3 → **Plaintext:** rsxzivcwigyvi

K = 4 → **Plaintext:** qrwyhubvhfxuh

K = 5 → **Plaintext:** pqvxgtaugewtg

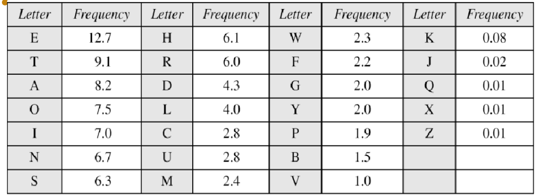
K = 6 → **Plaintext:** opuwfsztfdvsf

K = 7 → **Plaintext:** notverysecure

Additive/Shift cipher: Secure?? (Cryptanalysis)

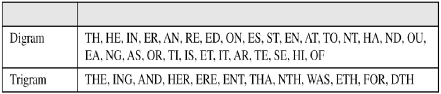
- Shift Cipher is not Secure
- Brute-force cryptanalysis easily performed on the shift cipher by trying all 25 possible keys.
- Given a ciphertext string, Eve/Darth successively try the decryption process with k = 0, 1, 2, etc. until get a meaningful text.
- Additive ciphers are also subject to **statistical attacks**.

Additive/Shift ciphers: Prone to Statistical Attacks!!



Letter	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

Figure 19: Frequency of occurrence of letters in an English text of 100 characters.



Digram	TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF
Trigram	THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH

Grouping of digrams and trigrams based on their frequency in English.

Multiplicative Ciphers

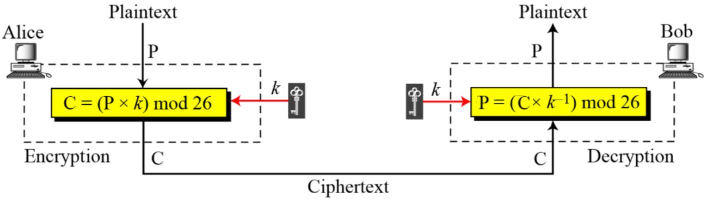


Figure 21: Multiplicative Cipher

- **In a multiplicative cipher, the plaintext and ciphertext are integers in Z_{26}; the key is an integer in Z$_{26}*$.**

Soma Saha

Multiplicative Cipher: Example

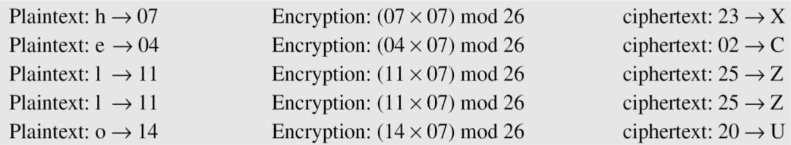
- What is the key domain for any multiplicative cipher, if keys are from English alphabet set?

Multiplicative Cipher: Example

- What is the key domain for any multiplicative cipher, if keys are from English alphabet set?
- **The key needs to be in $Z_{26}*$. This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.**

Multiplicative Cipher: Example

- What is the key domain for any multiplicative cipher, if keys are from English alphabet set?
- **The key needs to be in $Z_{26}*$. This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.**



Plaintext: h → 07	Encryption: (07×07) mod 26	ciphertext: 23 → X
Plaintext: e → 04	Encryption: (04×07) mod 26	ciphertext: 02 → C
Plaintext: l → 11	Encryption: (11×07) mod 26	ciphertext: 25 → Z
Plaintext: l → 11	Encryption: (11×07) mod 26	ciphertext: 25 → Z
Plaintext: o → 14	Encryption: (14×07) mod 26	ciphertext: 20 → U

Figure 22: Use of multiplicative cipher to encrypt the message “hello” with a key of 7.

Affine Ciphers

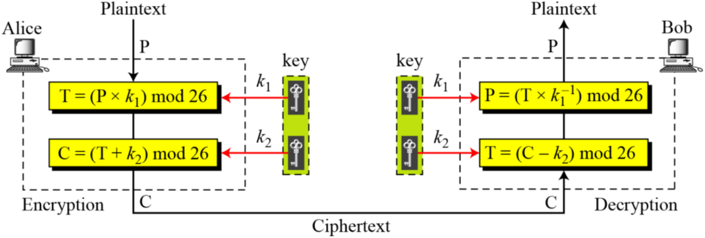


Figure 23: Affine Cipher: A combination of additive and multiplicative cipher with a pair of keys.

$$C = (P \times k_1 + k_2) \bmod 26 \qquad\qquad P = ((C - k_2) \times k_1^{-1}) \bmod 26$$

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2

Key Domain

- The affine cipher uses a pair of keys in which the first key is from $Z_{26}*$ and the second is from Z_{26}.
- What is the size of key domain for Additive/Shift cipher?

Key Domain

- The affine cipher uses a pair of keys in which the first key is from $Z_{26}*$ and the second is from Z_{26}.
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?

Key Domain

- The affine cipher uses a pair of keys in which the first key is from $Z_{26}*$ and the second is from Z_{26}.
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?
- What is the size of key domain for Affine cipher?

Key Domain

- The affine cipher uses a pair of keys in which the first key is from $Z_{26}*$ and the second is from Z_{26}.
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?
- What is the size of key domain for Affine cipher?

(i) 25, (ii) 11, (iii) 26x12 - 1 = 312 -1 = 311

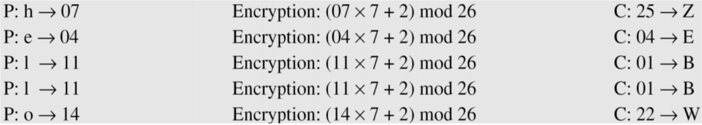
Key Domain

- The affine cipher uses a pair of keys in which the first key is from $Z_{26}*$ and the second is from Z_{26}.
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?
- What is the size of key domain for Affine cipher?

(i) 25, (ii) 11, (iii) 26x12 - 1 = 312 -1 = 311

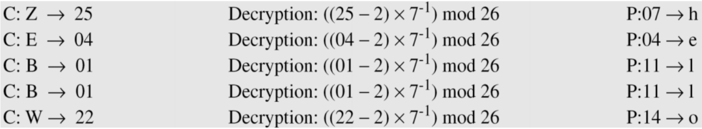
- **The additive cipher is a special case of an affine cipher in which $k_1 = 1$. The multiplicative cipher is a special case of affine cipher in which $k_2 = 0$.**

Affine Cipher: Example



P: h → 07	Encryption: $(07 \times 7 + 2) \bmod 26$	C: 25 → Z
P: e → 04	Encryption: $(04 \times 7 + 2) \bmod 26$	C: 04 → E
P: l → 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 → B
P: l → 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 → B
P: o → 14	Encryption: $(14 \times 7 + 2) \bmod 26$	C: 22 → W

Figure 24: Use of an affine cipher to encrypt the message “hello” with the key pair (7,2).

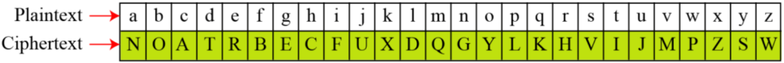


C: Z → 25	Decryption: $((25 - 2) \times 7^{-1}) \bmod 26$	P:07 → h
C: E → 04	Decryption: $((04 - 2) \times 7^{-1}) \bmod 26$	P:04 → e
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P:11 → l
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P:11 → l
C: W → 22	Decryption: $((22 - 2) \times 7^{-1}) \bmod 26$	P:14 → o

Figure 25: Use of an affine cipher to decrypt the message “ZEBBW” with the key pair (7,2) in modulus 26.

Monoalphabetic Substitution Cipher

- Additive, multiplicative, and affine ciphers have small key domains; therefore, they are very vulnerable to brute-force attack.
- **Monoalphabetic Substitution Cipher:** A better solution is to create a mapping between each plaintext character and the corresponding ciphertext character.
  - Alice and Bob can agree on a table showing the mapping for each character.

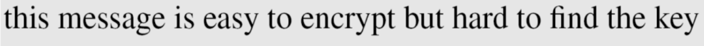


Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	N	O	A	T	R	B	E	C	F	U	X	D	Q	G	Y	L	K	H	V	I	J	M	P	Z	S	W

Figure 26: An example key for monoalphabetic substitution cipher.

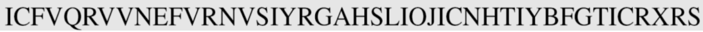
Monoalphabetic Substitution Cipher: Examples

- We can use the key in Figure to encrypt the message



this message is easy to encrypt but hard to find the key

- The ciphertext is



ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

Monoalphabetic Substitution Cipher: Secure?? (Cryptanalysis)

- Each alphabetic character is mapped to a unique alphabetic character. **One-to-One**
- We use arbitrary monoalphabetic substitution, so the key space for monoalphabetic subsitition cipher is $26!$, or almost $4x10^{26} \approx 2^{88}$ possible permutations, which is a very large number. Thus, **brute-force** seems infeasible.

Monoalphabetic Substitution Cipher: Secure?? (Cryptanalysis)

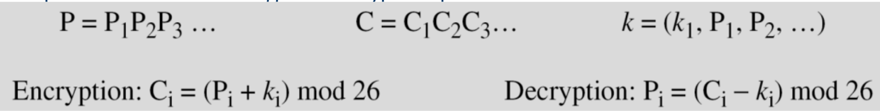
- Each alphabetic character is mapped to a unique alphabetic character. **One-to-One**
- We use arbitrary monoalphabetic substitution, so the key space for monoalphabetic subsitition cipher is $26!$, or almost $4x10^{26} \approx 2^{88}$ possible permutations, which is a very large number. Thus, **brute-force** seems infeasible.
- **However, a Monoalphabetic Substitution Cipher is insecure against frequency analysis.**

Polyalphabetic Cipher

- In polyalphabetic substitution, each occurrence of a character may have a different substitute.
- The relationship between a character in the plaintext to a character in the ciphertext is **one-to-many**.

Autokey Cipher

- *"autokey"* implies that the subkeys are automatically created from the plaintext cipher characters during the encryption process.

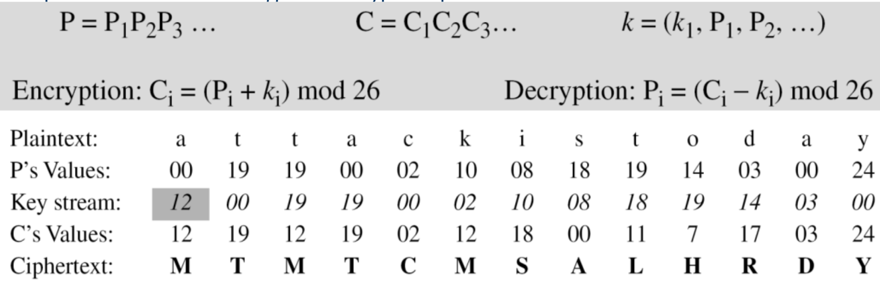


$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3 \ldots$ $k = (k_1, P_1, P_2, \ldots)$

Encryption: $C_i = (P_i + k_i) \bmod 26$ Decryption: $P_i = (C_i - k_i) \bmod 26$

Autokey Cipher

- *"autokey"* implies that the subkeys are automatically created from the plaintext cipher characters during the encryption process.



$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3\ldots$ $k = (k_1, P_1, P_2, \ldots)$

Encryption: $C_i = (P_i + k_i) \bmod 26$ Decryption: $P_i = (C_i - k_i) \bmod 26$

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P's Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	*12*	*00*	*19*	*19*	*00*	*02*	*10*	*08*	*18*	*19*	*14*	*03*	*00*
C's Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	**M**	**T**	**M**	**T**	**C**	**M**	**S**	**A**	**L**	**H**	**R**	**D**	**Y**

Figure 27: **Assume that Alice and Bob agreed to use an autokey cipher with initial key value $k_1 = 12$. Now Alice wants to send Bob the message "Attack is today". Enciphering is done character by character.**

Autokey Cipher: Secure?? (Cryptanalysis)

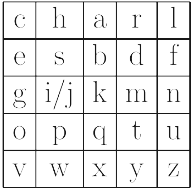
- The autokey cipher hides the single-layer frequency statistics of the plaintext. But..
- Vulnerable to brute-force attack as the additive cipher.

Autokey Cipher: Secure?? (Cryptanalysis)

- The autokey cipher hides the single-layer frequency statistics of the plaintext. But..
- Vulnerable to brute-force attack as the additive cipher.
- The first sub-key can be from one of the 25 values.
- We need polyalphabetic ciphers that not only hide the characteristics of the language but also have large key domains.

Playfair Cipher

- Variant of polyalphabetic cipher, used by the British army during World War I.
- The secret key in this cipher is made of 25 alphabet letters arranged in a 5x5 matrix (letter I and J are considered the same when encrypting).



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

Figure 28: An example of a secret key in the Playfair cipher.

Playfair Cipher: Rules..

- Plaintext: “meet me at the bridge”
  - Split the sentence into digrams removing spaces, ‘x’ used to make even number of letters: **me et me at th eb ri dg ex**

Playfair Cipher: Rules..

- Plaintext: “meet me at the bridge”
  - Split the sentence into digrams removing spaces, ‘x’ used to make even number of letters: **me et me at th eb ri dg ex**
  - Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x:
    “balloon” would be treated as **ba lx lo on**

Playfair Cipher: Rules..

- Plaintext: “meet me at the bridge”
  - Split the sentence into digrams removing spaces, ‘x’ used to make even number of letters: **me et me at th eb ri dg ex**
  - Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x:

    “balloon” would be treated as **ba lx lo on**
  - Two plaintext letters in the same row are each replaced by the letter to the right, with the first element of the row circularly following the last.

    eb is replaced by sd

    ng is replaced by gi (or gj as preferred)

Playfair Cipher: Rules..cont

  - Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last.
    dt would be replaced by my
    ty would be replaced by yr
  - Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter.
    me becomes gd

- Ciphertext therefore is:
  "gd do gd rq pr sd hm em bv"

Playfair Cipher: Secure?? (Cryptanalysis)

- Brute force attack is difficult as the size of the key domain is 25!
- Single letter frequency is obscured.
- But digrams are preserved.
- A cryptanalyst can use a ciphertext-only attack based on the digram frequency test to find the key.

Vigenere Cipher

- Simplest polyalphabetic substitution cipher, designed by Blaise de Vigenere, a sixteenth-century french mathematician.
- Consider the set of all Caesar ciphers:
  $\{C_a, C_b, C_c, \ldots, C_z\}$
- Key: e.g. security
- Encrypt each letter using $C_s, C_e, C_c, C_u, C_r, C_i, C_t, C_y$ in turn.
- Repeat from start after C_y.
- Decryption simply works in reverse.

Vigenere Cipher: Mathematical Representation

- Let m be a positive integer
- $P = C = K = (Z_{26})^m$
- For $k = (k_1, k_2, \ldots, k_m) \in K$,

1. $e_k(x_1, x_2, \ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$
2. $d_k(y_1, y_2, \ldots, y_m) = (y_1 - k_1, y_2 - k_2, \ldots, y_m - k_m)$

- All above operations are performed in Z_{26}

Vigenere Cipher: Example

- Correspondence between alphabetic characters and integer: $A = 0, B = 1, ..., Y = 24, Z = 25.$
- $m = 6.$
- Keyword is “CIPHER”, this corresponds to the numerical equivalent $k = (2, 8, 15, 7, 4, 17)$

Vigenere Cipher: Example..contd...

- Plaintext : “thi**s**crypto**s**y**s**temi**s**not**s**ecure”.
- Encryption: add modulo 26

$$\begin{array}{cccccc|cccccc|cccc}
19 & 7 & 8 & 18 & 2 & 17 & 24 & 15 & 19 & 14 & 18 & 24 & 18 & 19 & 4 & 12 \\
2 & 8 & 15 & 7 & 4 & 17 & 2 & 8 & 15 & 7 & 4 & 17 & 2 & 8 & 15 & 7 \\
\hline
21 & 15 & 23 & 25 & 6 & 8 & 0 & 23 & 8 & 21 & 22 & 15 & 20 & 1 & 19 & 19
\end{array}$$

$$\begin{array}{cc|cccccc|ccc}
8 & 18 & 13 & 14 & 19 & 18 & 4 & 2 & 20 & 17 & 4 \\
4 & 17 & 2 & 8 & 15 & 7 & 4 & 17 & 2 & 8 & 15 \\
\hline
12 & 9 & 15 & 22 & 8 & 25 & 8 & 19 & 22 & 25 & 19
\end{array}$$

- Ciphertext: “VPX**Z**GIAXIV**W**P**U**BTTM**J**PWI**Z**ITWZT”.

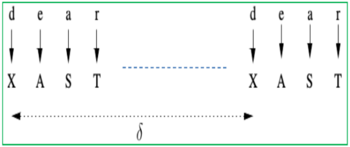
Vigenere Cipher: Secure? (Cryptanalysis)

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

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

Determining key length: Kaisiki Test(content courtesy: Chester Rebeiro)

- 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 $(m|\delta)$



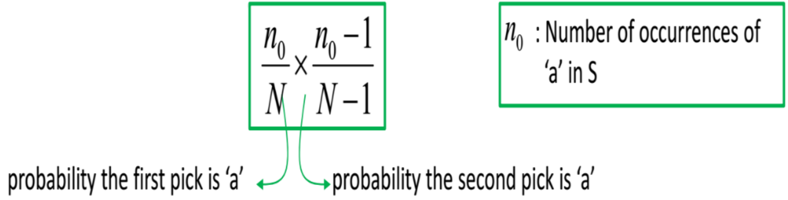
when (m divides δ)

- 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 $(\delta_1, \delta_2, \delta_3, \ldots)$

Soma Saha

Increasing Confidence of Key Length (Index of Coincidence)

- Consider a multi set of letters of size *N* say $s = \{a, b, c, d, a, a, e, f, e, g, \ldots\}$
- Probability of picking two 'a' characters (without replacement) is



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

n_0 : Number of occurrences of 'a' in S

probability the first pick is 'a'

probability the second pick is 'a'

Index of Coincidence..contd..1

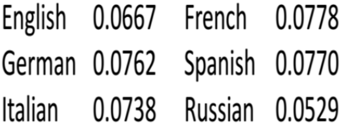
- Sum of probabilities of picking two similar characters is

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

index of coincidence

Index of coincidence.. cont..2

- Consider a random permutation of the alphabets (as in the substitution cipher) $s = \{a, b, c, d, a, a, e, f, e, g, \ldots\} -- > S = \{X, M, D, F, X, X, Z, G, Z, J, \ldots\}$
- 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

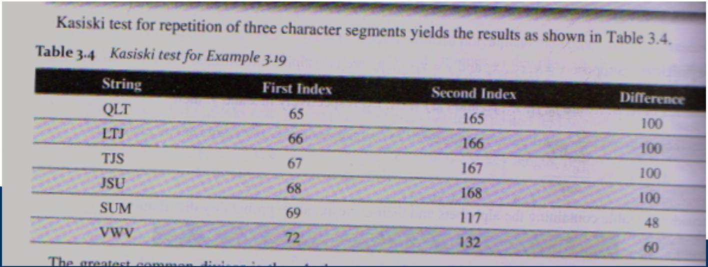


English	0.0667	French	0.0778
German	0.0762	Spanish	0.0770
Italian	0.0738	Russian	0.0529

Vigenere Cipher: Cryptanalysis Example

Let us assume we have intercepted the following ciphertext:

LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTHVTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFZMVVWLGYHCUSWXQHKVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWVWJWIXGFWLTSHGJOUEEHHVUCFVGOWICQLTJSUXGLW



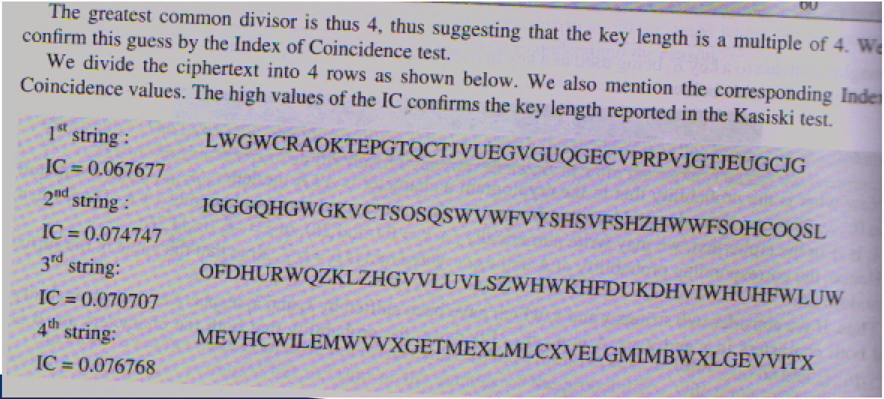
Kasiski test for repetition of three character segments yields the results as shown in Table 3.4.

Table 3.4 *Kasiski test for Example 3.19*

String	First Index	Second Index	Difference
QLT	65	165	100
LTJ	66	166	100
TJS	67	167	100
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60

The greatest common divisor is th

Vigenere Cipher: Cryptanalysis Example



The greatest common divisor is thus 4, thus suggesting that the key length is a multiple of 4. We confirm this guess by the Index of Coincidence test.

We divide the ciphertext into 4 rows as shown below. We also mention the corresponding Inde Coincidence values. The high values of the IC confirms the key length reported in the Kasiski test.

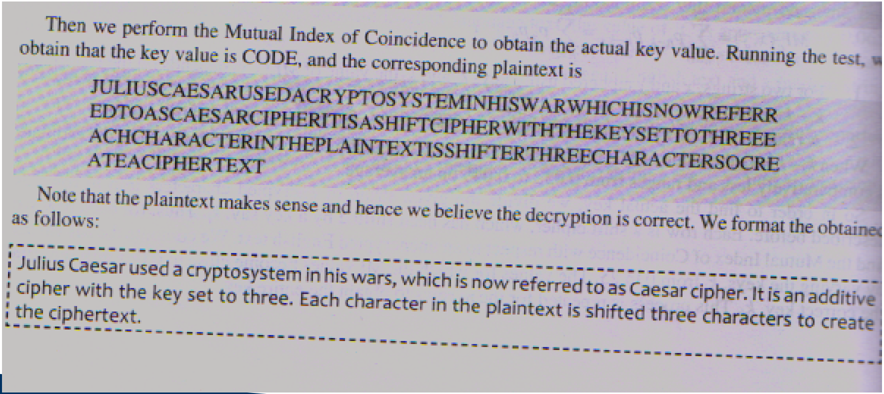
1st string : LWGWCRAOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG
IC = 0.067677

2nd string : IGGGQHGWGKVCTSOSQSWVWFVYSHSVFSHZHWWFSOHCOQSL
IC = 0.074747

3rd string: OFDHURWQZKLZHGVVLUVLSZWHWKHFDUKDHVIWHUHFWLUW
IC = 0.070707

4th string: MEVHCWILEMWVVXGETMEXLMLCXVELGMIMBWXLGEVVITX
IC = 0.076768

Vigenere Cipher: Cryptanalysis Example



Then we perform the Mutual Index of Coincidence to obtain the actual key value. Running the test, w obtain that the key value is CODE, and the corresponding plaintext is

JULIUSCAESARUSEDACRYPTOSYSTEMINHISWARWHICHISNOWREFERR
EDTOASCAESARCIPHERITISASHIFTCIPHERWITHTHEKEYSETTOTHREEE
ACHCHARACTERINTHEPLAINTEXTISSHIFTERTHREECHARACTERSOCRE
ATEACIPHERTEXT

Note that the plaintext makes sense and hence we believe the decryption is correct. We format the obtained as follows:

Julius Caesar used a cryptosystem in his wars, which is now referred to as Caesar cipher. It is an additive cipher with the key set to three. Each character in the plaintext is shifted three characters to create the ciphertext.

Hill Cipher (Content courtesy: Chester Rebeiro)

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

Hill Cipher.. (Content courtesy: Chester Rebeiro)

- example

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

plaintext

$\underbrace{hi}\underbrace{ll}$ → (7,8)(11,11)

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

encryption

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

decryption

$$hill \rightarrow (7,8)(11,11) \longrightarrow (23,8)(24,9) \rightarrow \text{XIYJ}$$

plaintext ciphertext

Cryptanalysis of Hill Cipher (content courtesy: Chester R. and Debdeep M.)

- ciphertext only attack is difficult
- known plaintext attack

$$\underset{\text{known plaintext}}{(7,8)(11,11)} \times \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \longrightarrow \underset{\text{corresponding ciphertext}}{(23,8)(24,9)}$$

Form equations and solve to get the key

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

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

One-Time Pad

- One of the goals of cryptography is perfect secrecy.
- A study by **Shannon** has shown that perfect secrecy can be achieved if each plaintext symbol is encrypted with a key randomly chosen from a key domain.
- This idea is used in a cipher called **one-time pad**, invented by **Vernam**.
- The key has the same length as the plaintext and is chosen in random.
- The key is changed each time the sender sends a new message.

One-Time Pad.. contd..

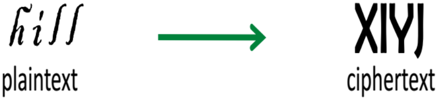
- For example, an additive cipher can be easily broken because the same key is used to encrypt every character.
- However, even this simple cipher can become a perfect cipher if the key that is used to encrypt each character is chosen randomly from the key domain (00, 01, 02, ..., 25) -i.e if the first character is encrypted using the key 04, the second character is encrypted using the key 02, the third character is encrypted using the key 21; and so on.
- **ciphertext-only attack is impossible**.
- Other types of attacks are also impossible if the sender changes the key each time she/he sends a message, using another random sequence of integers.

One-Time Pad: Feasibility

- A one-time pad is a perfect cipher, but it is almost impossible to implement commercially.
- If the key must be newly generated each time, how can Alice tell Bob the new key each time she has a message to send?
- **There are some occasions when a one-time can be used. For example, if the president of a country needs to send a completely secet message to the president of another country, she/he can send a trusted envoy with a random key before sending the message.**

Permutation Cipher

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



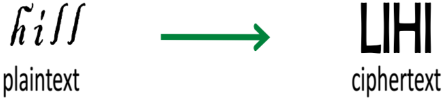
hill

plaintext

XIYJ

ciphertext

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



hill

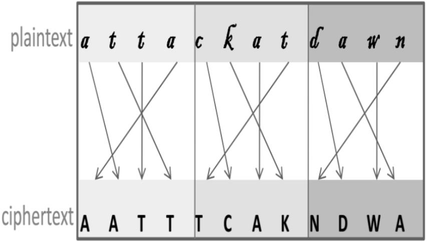
plaintext

LIHI

ciphertext

Permutation Cipher

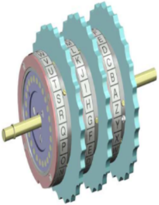
- Example plaintext: ***attackatdawn***
  - key: (1,3,2,0) here is of length 4 and a permutation of (0,1,2,3)
  - It refers 0^{th} character in plaintext goes to 1^{st} character in ciphertext (and so on..)

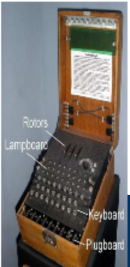


- cryptanalysis : 4! possibilities

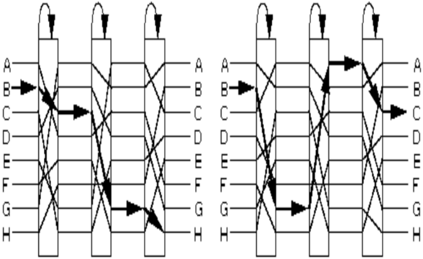
Rotor Cipher Machines (German Enigma)

- Before modern ciphers, rotor machines were most common complex ciphers in use.
- Widely used in WW2.
- Used a series of rotating cylinders.
- Implemented a polyalphabetic substitution cipher of period K.





Rotor Cipher 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

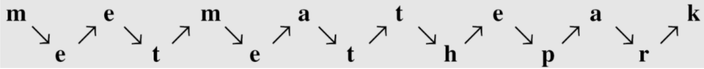
Transposition Ciphers

- A transposition cipher reorders symbols.
- Classification:

1. Keyless Transposition Ciphers
2. Keyed Transposition Ciphers
3. Combining Two Approaches

Keyless Transposition Ciphers

- Reorders the symbols.
- Simple transposition ciphers, which were used in the past, are keyless.
- Example:
  - A good example of a keyless cipher using the first method is the **rail fence cipher**. The ciphertext is created reading the pattern row by row. For example, to send the message **“Meet me at the park”** to Bob, Alice writes:



m e m a t e a k

e t e t h p r

  - She then creates the ciphertext **“MEMATEAKETETHPR”**.

Keyless Transposition Cipher: Example 2

- Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.

1	2	3	4	5	6
M	E	E	T	M	E
A	F	T	E	R	P
A	R	T	Y		

4	2	1	6	3	5
T	E	M	E	E	M
E	F	A	P	T	R
Y	R	A		T	

Plain Text: MEET ME AFTER PARTY

Key Used: 421635

Cipher Text: TEMEEMEFAPTRYRAT

Keyed Transposition Ciphers

- The keyless ciphers permute the characters by using writing plaintext in one way and reading it in another way.
- The permutation is done on the whole plaintext to create the whole ciphertext.
- Another method is **to divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately**.

Keyed Transposition Cipher: Example

- Alice needs to send the message **"Enemy attacks tonight"** to Bob..

e	n	e	m	y	a	t	t	a	c	k	s	t	o	n	i	g	h	t	z

- The key used for encryption and decryption is a permutation key, which shows how the character are permuted.

Encryption ↓

3	1	4	5	2
1	2	3	4	5

↑ Decryption

Keyed Transposition Cipher: Example

- Alice needs to send the message **“Enemy attacks tonight”** to Bob..

enemy attac kston ightz

- The key used for encryption and decryption is a permutation key, which shows how the character are permuted.

Encryption ↓

3	1	4	5	2
1	2	3	4	5

↑ Decryption

- The permutation yields

EEMYN TAACT TKONS HITZG

Bibliography: Books and Resources

- Cryptography and Network Security: Principles and Practice by William Stallings
- Cryptography and Network Security by Behrouz A Forouzan and Debdeep Mukhopadhyay
- Principles of Information Security by Michael E. Whitman and Herbert J. Mattord.
- Cisco platform, and Internet.
- Published research papers, study materials from researchers of security domain.