1. In the field of information security, Kerckhoffs' Principle is like motherhood and apple pie, all rolled up into one.

a. Define Kerckhoffs' Principle in the context of cryptography.

b. Give a real-world example where Kerckhoffs' Principle has been violated. Did this cause any security problems?

c. Kerckhoffs' Principle is sometimes applied more broadly than its strict cryptographic definition. Give a definition of Kerckhoffs' Principle that applies more generally.

1. The fundamental principle of cryptography that the inner workings of a cryptosystem are completely known to the attacker. The only secret is a key. Keeping cryptosystem secure with these conditions is known as Kerckhoffs’ principle
2. Microsoft’s Network LAN Manager (NTLM) violated Kerckhoffs’ Principle. NTLM authentication service allowed attacker to use Rainbow attack.

In this attack, even if attacker does not know the secret key, he can break the system using pre-computed directory of encrypted passwords.

Hence, this violation of Kerckhoffs’ principle did causes security issues in NTLM suite and another one.

A house lock is an example of a mechanism whose inner workings are completely known, but this has posed a problem, as people have devised many various ways to break through the lock

1. Kerckhoff’s Principle can be applied in various aspects of security other than cryptography. The principle is taken to mean that security design is open to public scrutiny. This means people can expose security flaws in the design. If the system stands this scrutiny, it is more secure

Q2-Edgar Allan Poe's 1843 short story, "The Gold Bug," features a cryptanalytic attack.

a. What type of cipher is broken and how?

b. What happens as a result of this cryptanalytic success?

a) The cipher that is broken is a simple substitution. This cipher is broken by doing a letter frequency count. Comparing symbol frequency count with the English letter frequency count and then replacing the symbols in the cipher-text with letters by order of occurrence

or

The cypher in the story is simple substitution cypher and it is broken using frequency analysis.

b) The success made it so that there are shortcuts for seemingly impossibly large amount of keys to try. Can’t just use simple substitution. Seems like there are 26 factorial keys but you need better confusion that statistical analysis cannot be useful in. And so, evolved cryptanalysis.

or

Due to this cryptanalytic success, a hidden fortune is discovered in form of a chest filled with gold coins and jewelry

3. Given that the Caesar's cipher was used, find the plaintext that corresponds

to the following ciphertext: VSRQJHEREVTXDUHSDQWU

Plaintext: spongebobsquarepants

4. Find the plaintext and the key, given the ciphertext 🡺🡺🡺nahi smjh arhaa yeh

CSYEVIXIVqMREXIH.

Hint: The key is a shift of the alphabet.

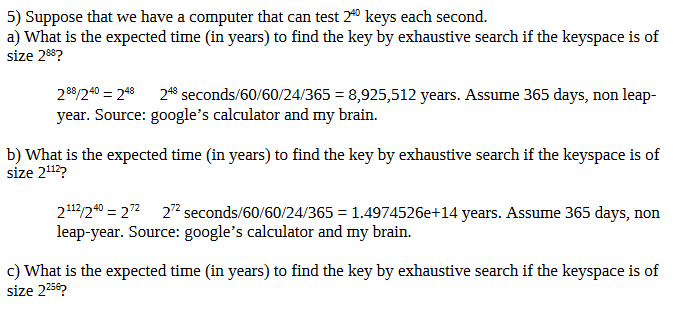


5. Suppose that we have a computer that can test 240 keys each second.

a. What is the expected time (in years) to find a key by exhaustive search if the keyspace is of size 288?

b. What is the expected time (in years) to find a key by exhaustive search if the keyspace is of size 2^112?

c. What is the expected time (in years) to find a key by exhaustive search if the keyspace is of size 2^256?

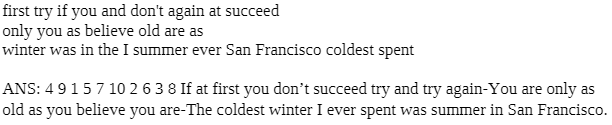


Q6-The weak ciphers used during the election of 1876 employed a fixed permutation of the words for a given length sentence. To see that this is weak, find the permutation of ( 1 , 2 , 3 , . . . , 10) that was used to produce the scrambled sentences below, where "San Francisco" is treated as a single word. Note that the same permutation was used for all three sentences.

first try try if you and don't again at succeed

only you you you as believe old are are as

winter was in the I summer ever San Francisco coldest spent



Q7-The weak ciphers of the election of 1876 used a partial codebook and a permutation of the words. Modify this approach so that it is more secure

Ans-) In election of 1876, Partial codebook was used, Use a permutation only once or a very limited number of times. The Codebook could also be more comprehensive to avoid giving away too much context.

8. This problem deals with the concepts of confusion and diffusion

a. Define the terms confusion and diffusion as used in cryptography.

b. Which classic cipher discussed in this chapter employs only confusion?

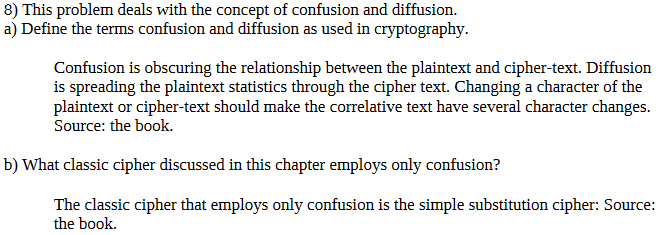
c. Which classic cipher discussed in this chapter employs only diffusion?

d. Which cipher discussed in this chapter employs both confusion and

diffusion?

Ans No.a

In [cryptography](https://en.wikipedia.org/wiki/Cryptography), **confusion** and **diffusion** are two properties of the operation of a secure [cipher](https://en.wikipedia.org/wiki/Cipher) identified by [Claude Shannon](https://en.wikipedia.org/wiki/Claude_Elwood_Shannon)



1. Confusion :

1-Confusion is a cryptographic technique which is used to create faint cipher texts

2- This technique is possible through substitution algorithm.

3- In confusion, if one bit within the secret’s modified, most or all bits within the cipher text also will be modified.

4- In confusion, vagueness is increased in resultant.

5- Both stream cipher and block cipher uses confusion

6- The relation between the cipher text and the key is masked by confusion

1. Diffusion

1-diffusion is used to create cryptic plain texts.

2- While it is possible through transportation algorithm.

3- While in diffusion, if one image within the plain text is modified, many or all image within the cipher text also will be modified

4- While in diffusion, redundancy is increased in resultant.

5- Only block cipher uses diffusion

**Ans No.b**

One-time pad

**Ans No.c**

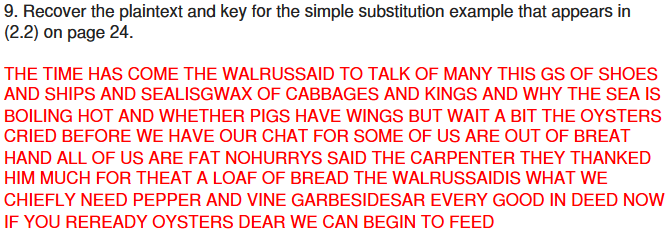
Double transposition is the example of classic cypher which employs only diffusion. Plaintext is arranged in matrix form and rows and columns are shuffled to get the cipher text

**Ans No.d**

Ciphers of the Election of 1876 are examples of cypher which employ both confusion and diffusion. Codebook is used as confusion step and permutation of words is used as diffusion step

9. Recover the plaintext and key for the simple substitution example that appears in (2.2) on page 24.

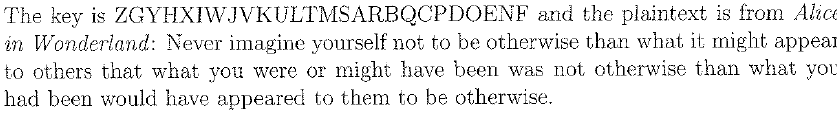




Kesy pta kitny shift keye hai key kesy pta ?

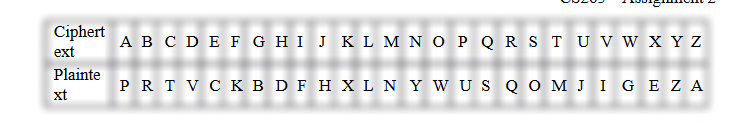
10. Determine the plaintext and key for the cipher text that appears in the *Alice in Wonderland* quote at the beginning of this chapter. Hint: “The message was encrypted with a simple substitution cipher and the plaintext has no spaces or punctuation.”

MXDXBVTZWVMXNSPBQXLIMSCCSGXSCJXBOVQXCJZMOJZCVCTVWJCZAAXZBCSSCJXBQCJZCOJZCNSPOXBXSBTVWJCJZDXGXXMOZQMSCSCJXBOVQXCJZMOJZCNSPJZHGXXMOSPLHJZDXZAAXZBXHCSCJXTCSGXSCJXBOVQX



Plaintext: NEVER IMAGINE YOURSELF NOT TO BE OTHERWISE THAN WHAT ITMIGHT APPEAR TO OTHERS THAT WHAT YOU WERE OR MIGHTHAVE BEEN WAS NOT OTHERWISE THAN WHAT YOU HAD BEEN WOULDHAVE APPEARED TO THEM TO BE OTHERWISE

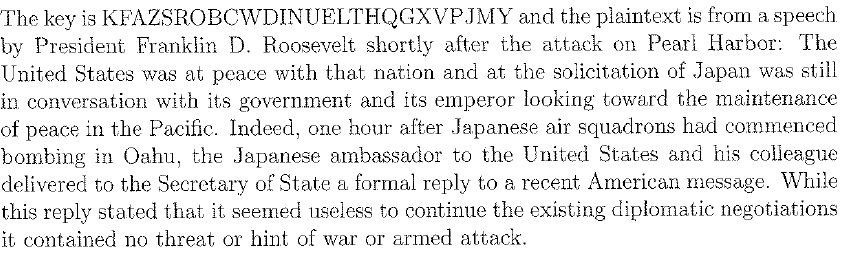
Key:



11. Decrypt the following message that was encrypted using a simple substitution

cipher:

GBSXUCGSZQGKGSQPKQKGLSKASPCGBGBKGUKGCEUKUZKGGBSQEICACGKGCEUERWKLKUPKQQGCIICUAEUVSHqKGCEUPCGBCGQOEVSHUNSUGKUZCGQSNLSHEHIEEDCUOGEPKHZGBSNKCUGSUKUASERLSKASCUGBSLKACRCACUZSSZEUSBEXHKRGSHWKLKUSQSKCHQTXKZHEUQBKZAENNSUASZFENFCUOCUEKBXGBSWKLKUSQSKNFKQQKZEHGEGBSXUCGSZQGKGSQKUZBCQAEIISKOXSZSICVSHSZGEGBSQSAHSGKHMERQGKGSKREHNKIHSLIMGEKHSASUGKNSHCAKUNSQQKOSPBCISGBCqHSLIMQGKGSZGBKGCGQSSNSZXQSISQQGEAEUGCUXSGBSSJCqGCUOZCLIENKGCAUSOEGCKGCEUqCGAEUGKCUSZUEGBHSKGEHBCUGERPKHEHKHNSZKGGKAD



12. Write a program to help an analyst decrypt a simple substitution cipher. Your program should take the cipher text as input, compute letter frequency counts, and display these for the analyst. The program should then allow the analyst to guess a key and display the results of the corresponding "decryption" with the putative key.

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.HashMap;

import java.util.LinkedHashMap;

import java.util.Map;

public class SubstitutionCipher

Map letterFreqMap;

Map key;

String ciphertext;

String plaintext;

BufferedReader reader;

private void createInputBuffer()

reader = new BufferedReader(new InputStreamReader(System.in));

private void closeInputBufer() throws Exception

if(reader!=null)

try

reader.close();

catch (IOException e)

e.printStackTrace();

throw new Exception(Error while trying to close the input stream);

private void getCipherText() throws Exception

System.out.println(\*\*\*\*\* Substitution Cipher Cracker \*\*\*\*\*);

System.out.print(Enter ciphertext: );

reader =...

13. Extend the program described in Problem 12 so that it initially tries to decrypt the message. One sensible way to proceed is to use the computed letter frequencies and the known frequencies of English for an initial guess at the key. Then from the resulting putative decryption ,count the number of dictionary words that appear and use this as a

score. Next, for each letter in the key, try swapping it with the letter that is adjacent (with respect to frequency counts) and recompute the score. If the score improves, update the key; if not, don't change the putative key. Iterate this process until the score does not improve for a entire pass through the alphabet. At this point you will give your putative decryption to the analyst. To aid the analyst in the manual phase ,your program must maintain all of the functionality of the program in

Problem 12.

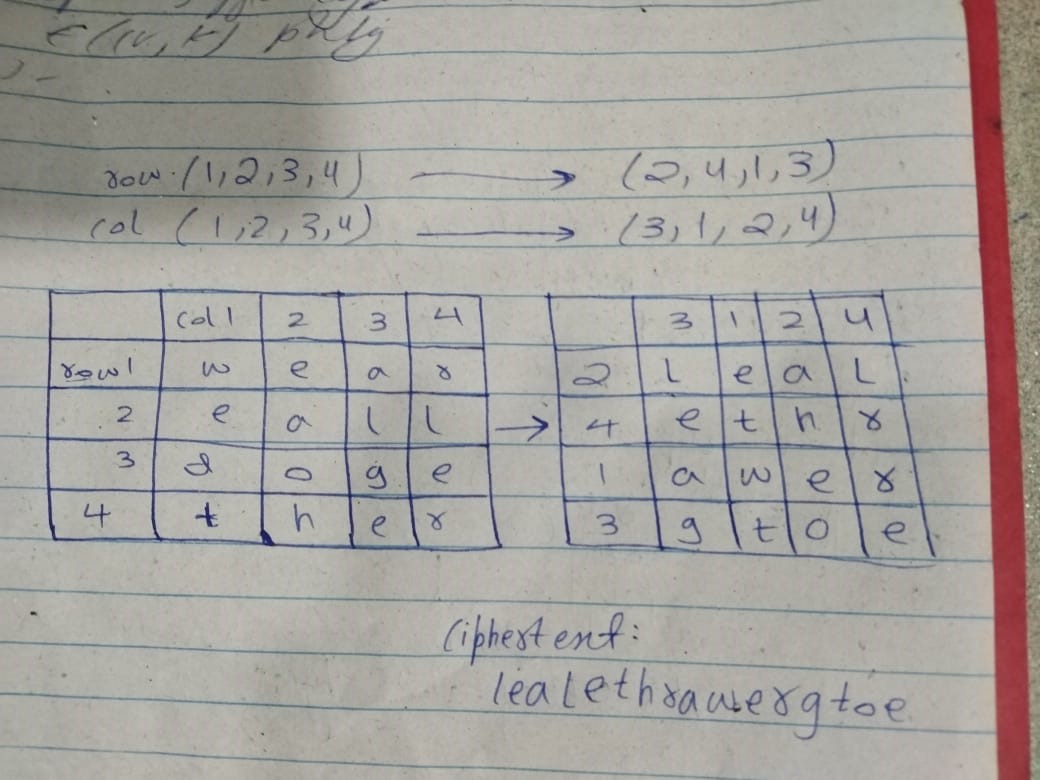
<http://ddsgibberish.blogspot.com/2015/09/autosolve-substitution-cipher.html>

14. Encrypt the message **we are all together** using a double transposition cipher (of the type described in the text) with 4 rows and 4 columns, using the row permutation

(1,2,3,4)—> (2,4,1,3)

and the column permutation

(1,2,3,4)—\* (3,1,2,4)

**MANUAL**

15. Decrypt the ciphertext

IAUTMDCSMNIMREBOTNELSTRHEREOAEVMWIHTSEEATMAEOHWHSYCEELTTEOHMUOUFEHTRFT

This message was encrypted with a double transposition (of the type discussed in the text) using a matrix of 7 rows and 10 columns. Hint:The first word is "there."

Ans

There are some who say that communism is the wave of the future, let them come to Berlin

16. Outline an automated attack on a double transposition cipher (of the type discussed in the text), assuming that the size of the matrix is known.

Divide and conquer--- try to undo the column permutations first.

**MANUAL**

17. A double transposition cipher can be made much stronger by using the following approach.

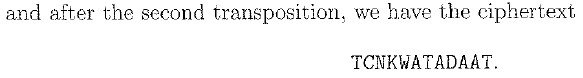
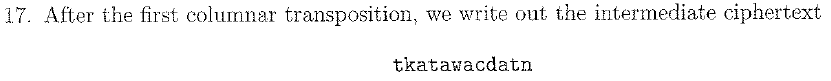
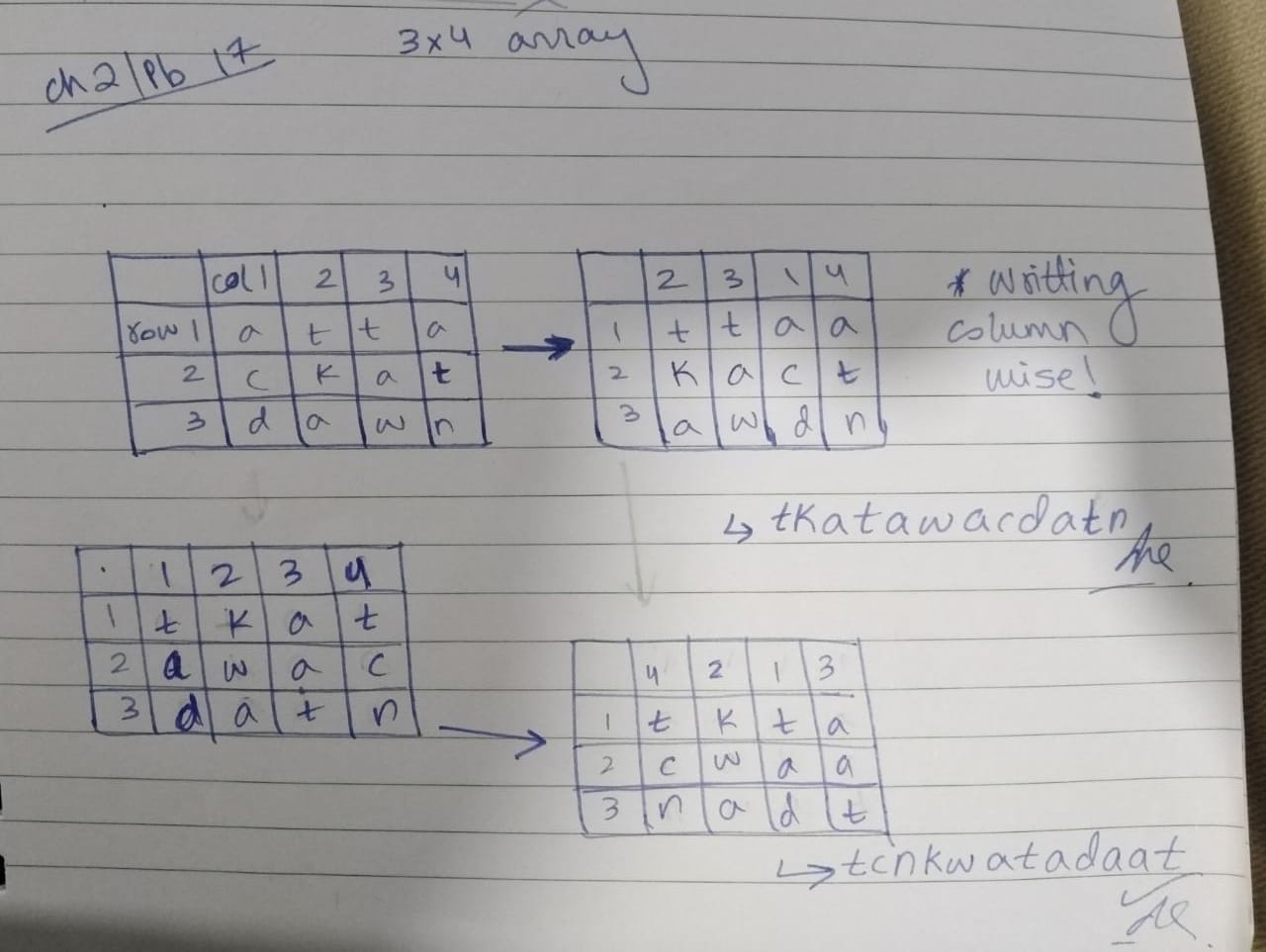
First, the plaintext is put into *an n x m* array,

as described in the text. Next, permute the columns, and then write out the intermediate ciphertext column by column.

That is, column 1 gives the first *n* ciphertext letters, column 2 gives the next n, and so on.

Then repeat the process, that is, put the intermediate ciphertext into an n x m array, permute the columns, and write out the ciphertext column by column.

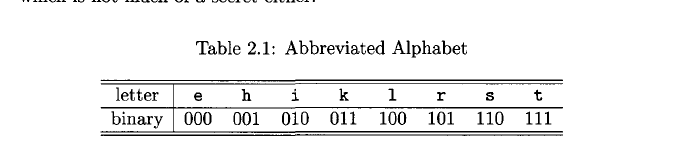
Use this approach, with a 3 x 4 array, and permutations (2,3,1,4) and (4, 2,1,3) to encrypt the plaintext attackatdawn.

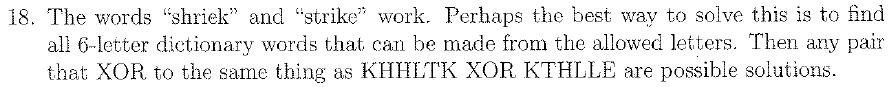
**Manual** 

18. Using the letter encodings in Table 2.1, the following two ciphertext messages were encrypted with the same one-time pad:

KHHLTK and KTHLLE.

Find all possible plaintexts for each message and the corresponding one-time pad.





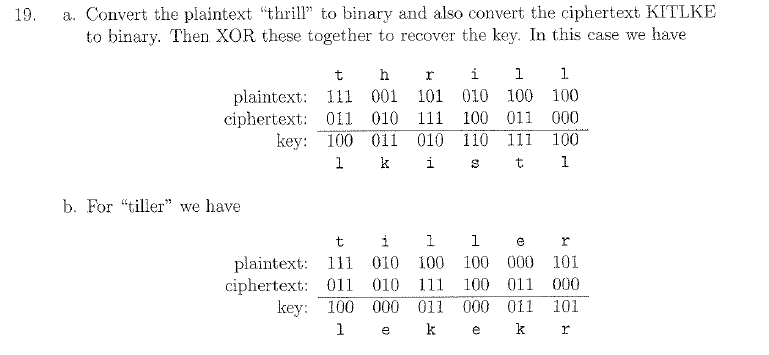
**MANUAL**

19. Using the letter encodings in Table 2.1, the following ciphertext message was encrypted with a one-time pad:

KITLKE.

a. If the plaintext is "thrill," what is the key?

b. If the plaintext is "tiller," what is the key?



20. Suppose that you have a message consisting of 1024 bits. Design a method that will extend a key that is 64 bits long into a string of 1024 bits, so that the resulting 1024 bits can be XORed with the message,

just like a one-time pad.

Easy way to extend a 64-bit length is to just repeat by multiplying it by (24)16 to achieve a length of 1024 bits. And since they are the same length, they can be XORed

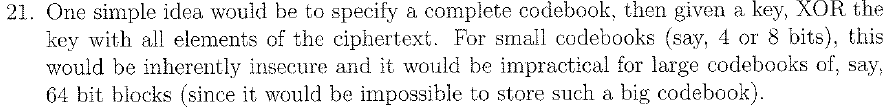
Is the resulting cipher as secure as a one-timepad?

No, because while they are of the same length, the one-time pad uses wholly random key of distinct 1024 bits. While, the 64-bit extension uses repeating keys

Is it possible for any such cipher to be as secure as a one-time pad?

The one-time pad seems to be the only provable secure one since you have to assume algorithms are known but specific keys are not. The one-time pad makes sure the key is wholly random and due to that, all plaintexts are equally likely.

21. Design a codebook cipher that can encrypt any block of bits, not just specific words. Your cipher should include many possible codebooks, with a key used to determine which codebook will be employed to encrypt (or decrypt) a particular message. Discuss some possible attacks on your cipher.



22. Suppose that the following is an excerpt from the decryption codebook for a classic codebook cipher.

123

199

202

221

233

332

451

once

or

maybe

twice

time

upon

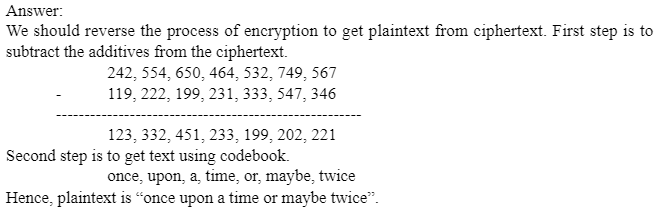
a

Decrypt the following ciphertext:

242, 554, 650, 464, 532, 749, 567

assuming that the following additive sequence was used to encrypt the message:

119, 222, 199, 231, 333, 547, 346



23. An affine cipher is a type of simple substitution where each letter is encrypted according to the rule c = (a.p + b) mod 26 (see the Appendix for a discussion of mod).

Here, p, c, a, and b are each number in the range 0 to 25, where p represents the plaintext letter, c the ciphertext letter, and a and b are constants.

For the plaintext and ciphertext, 0 corresponds to "a," 1 corresponds to "b," and so on.

Consider the ciphertext QJKES REOGH GXXRE OXEO, which was generated using an

affine cipher.

Determine the constants a and b and decipher the message.

Hint: Plaintext "t" encrypts to ciphertext "H" and plaintext "o" encrypts to ciphertext "E."

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

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

c ≡ ax + b (mod 26)

T 🡪 19

H 🡪 7

7 = 19a +b (mod 26) –i

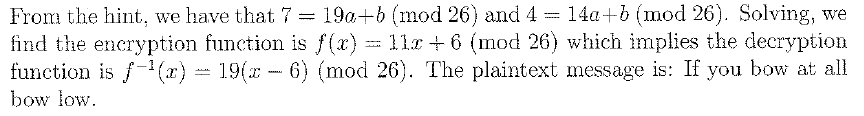
0 🡪 14

E 🡪 4

4 = 14a + b (mod 26) –ii

Solving eq i & ii

­



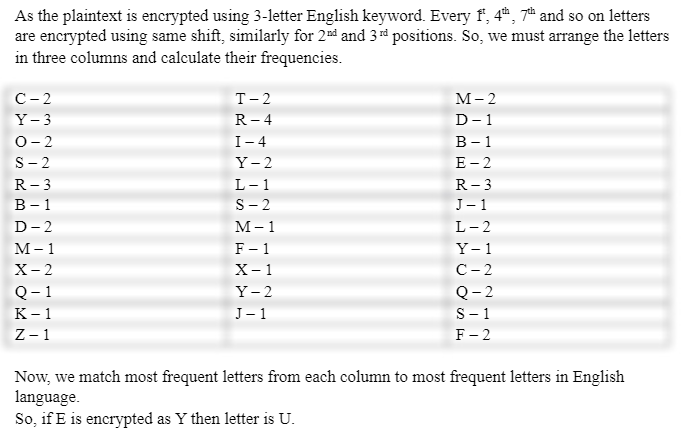
24. A Vigenère cipher uses a sequence of "shift-by-n" simple substitutions, where the shifts are indexed using a keyword, with "A" representing a shift-by-0, "B" representing a shift-by-l, etc. For example, if the keyword is "DOG," then the first letter is encrypted using a simple substitution with a shift-by-3, the second letter is encrypted using a shift-by-14, the third letter is encrypted using a shift-by-6, and the pattern is repeated—the fourth letter is encrypted using a shift-by-3, the fifth letter is encrypted using a shift-by-14, and so on.

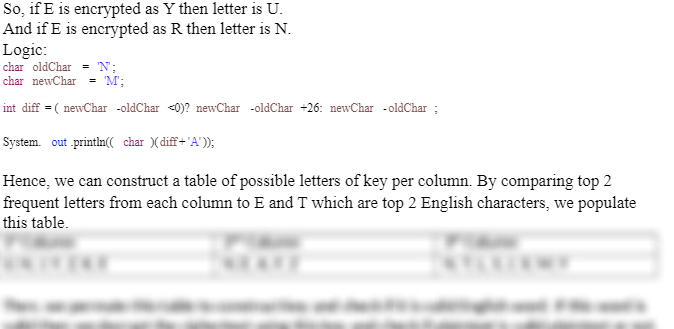
Cryptanalyze the following ciphertext, i.e., determine the plaintext and the key.

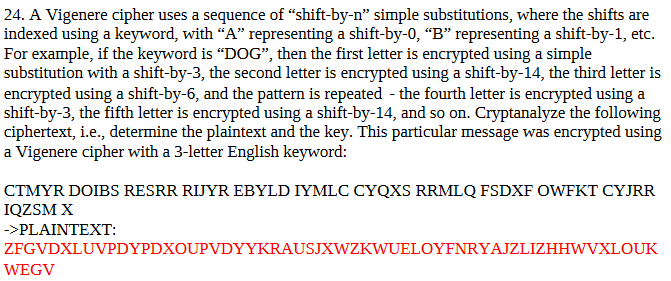
This particular message was encrypted using a Vigenère cipher with a 3-letter English keyword:

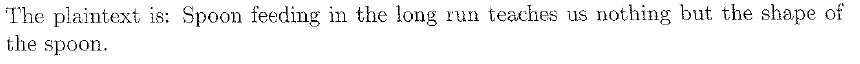
CTMYR DOIBS RESRR RIJYR EBYLD IYMLC CYQXS RRMLQ FSDXFOWFKT

CYJRR IQZSM X









25. Suppose that on the planet Binary, the written language uses an alphabet that contains only two letters X and Y. Also, suppose that in the Binarian language,

the letter X occurs 75% of the time, while Y occurs 25% of the time.

Finally, assume that you have two messages in the Binary language, and the messages are of equal length.

1. If you compare the corresponding letters of the two messages, what fraction of the time will the letters match?

Manual

2 letters h means yaahn pe 0.75\*0.75 + 0.25+0.25 kr k match ki value probability mil skti h

Qk 2 same matches ki bt hore h toh total possibles changes batane h

b. Suppose that one of the two messages is encrypted with a simple substitution, where X is encrypted as Y and Y is encrypted as X. If you now compare the corresponding letters of the two messages—one encrypted and one not—what fraction of the time will the

letters match?

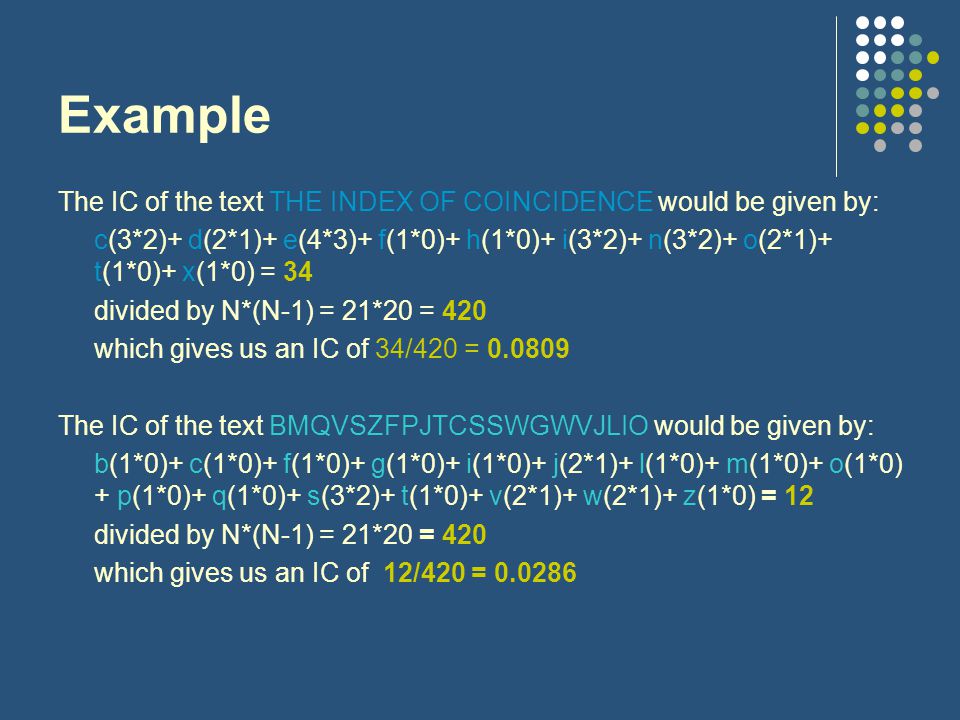
C : Suppose that both of the messages are encrypted with a simple substitution, where X is encrypted as Y and Y is encrypted as X. If you now compare the corresponding letters of the two messages— both of which are encrypted—what fraction of the time will the letters match?

Same

d. Suppose instead that you are given two randomly generated "messages" that use only the two letters X and Y. If you compare the corresponding letters of the two messages, what fraction of the time will the letters match?

--2 letter h toh ½ ½ ki probability shoga dono ki hadlf changes h toh 1/2

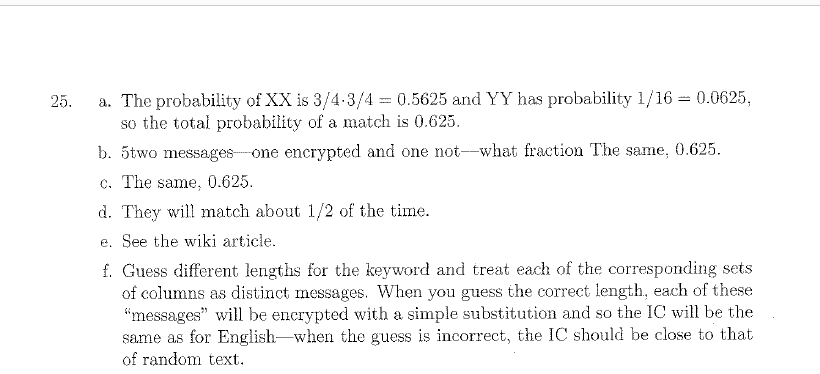
e- What is the index of coincidence (IC)? Hint: See, for example,[148].



f. How can the index of coincidence be used to determine the length of the keyword in a Vigenère cipher (see Problem 24 for the definition of a Vigenère cipher)?

--key or plaintext ki length same hone chahye so if humne koi random plaintext dia or wo key se match nh hua toh IC closed hojayega or agar key ki length se match hogaay toh koi cipertext generate karega toh bar bar length baarha barha r check krna h

🡺Manual



26. In this chapter, we discussed a forward search attack.

a. Explain how to conduct a forward search attack.

To conduct a forward search attack, you attack a public key. You suspect that the text is some text (book says “yes” or “no”) and encrypt the possibility text with the public key. And by comparing your cipher-text with the intercepted cipher-text, comparisons with the intercepted cipher-text and the cipher-text you create with the public key can lead to a leak

b. How can you prevent a forward search attack against a public key cryptosystem?

To prevent a forward search attack against a public key you need to have a bigger message space so that the message is not predictable enough to be attacked in this way. Source: me, thinking with the book as a resource

c. Why can't a forward search attack be used to break a symmetric cipher?

A forward search attack cannot break a symmetric cipher because the hacker does not have a public key to play with.

27. Consider a "one-way" function h. Then, given the value y = h(x), it is computationally infeasible to find x directly from y.

a. Suppose that Alice computes y = h(x), where x is Alice's salary, in dollars. If Trudy obtains y, how can she determine Alice's salary xl

Hint: Adapt the forward search attack to this problem.

b. Why does your attack not violate the one-way property of h?

c. How could Alice prevent this attack? We assume that Trudy has access to the output of the function h, Trudy knows that the input includes Alice's salary, and Trudy knows the format of the input. Also, no keys are available, so Alice cannot encrypt the output

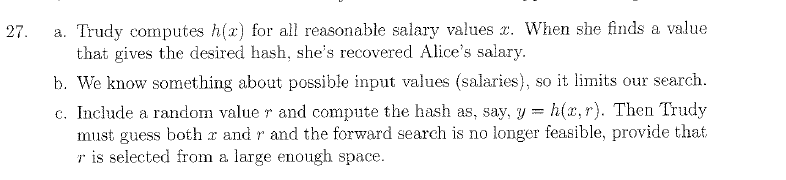
value.

🡺one way ==hash function so Trudy can guess all possible number

🡺hash value h us k pass eke k kr k possible integer number dal kr match krega agar kse number se hash value match hogaye toh wo succeed hogaya

🡺kuch possible i/p values paata hone chahye

🡺preventation:har bar random value generate hogi jo bht large hogi ,Trudy ko dono guess krna paregi toh isse kafi hadh tk prevention hosktey h



28. Suppose that a particular cipher uses a 40-bit key, and the cipher is secure (i.e., there is no known shortcut attack).

a. How much work, on average, is an exhaustive search attack?

🡪atleast 2^39 attack krne paregy

1. Outline an attack, assuming that known plaintext is available.

🡪agar plaintext dia hua h toh do exhaustic key serch or ha rkey k lye decrupt krna prega text qk key khud baanrhe h ecahustic kr k or plaintext h humre pass toh match kr sktey h

🡪cipher text pe kse attack kregy agar cipher text h hoga ?

c. How would you attack this cipher in the ciphertext-only case?

2.8 PROBLEMS 49

29. Suppose that Alice encrypted a message with a secure cipher that uses a 40-bit key. Trudy knows the ciphertext and Trudy knows the algorithm, but she does not know the plaintext or the key. Trudy plans to do an exhaustive search attack, that is, she will try each possible key until she

finds the correct key.

a. How many keys, on average, must Trudy try before she finds thecorrect one?

b. How will Trudy know when she has found the correct key? Note that there are too many solutions for Trudy to manually examine each one—she must have some automated approach to determining

whether a putative key is correct or not.

c. How much work is your automated test in part b?

d. How many false alarms do you expect from your test in part b?

That is, how often will an incorrect key produce a putative decrypt that will pass your test?

