4th Class

2017-2018

Cryptanalysis

تحليل الشفرة

أستاذة المادة : د. هالة بهجت

Basic Terminology

Suppose that someone wants to send a message to a receiver, and wants to be sure that no-one else can read the message. However, there is the possibility that someone else opens the letter or hears the electronic communication.

In cryptographic terminology, the message is called **plaintext** or **cleartext**. Encoding the contents of the message in such a way that hides its contents from outsiders is called **encryption**. The encrypted message is called the **ciphertext**. The process of retrieving the plaintext from the ciphertext is called **decryption**. Encryption and decryption usually make use of a **key**, and the coding method is such that decryption can be performed only by knowing the proper key.

Cryptography is the art or science of keeping messages secret. **Cryptanalysis** is the art of **breaking** ciphers, i.e. retrieving the plaintext without knowing the proper key. People who do cryptography are **cryptographers**, and practitioners of cryptanalysis are **cryptanalysts**.

Cryptography deals with all aspects of secure messaging, authentication, digital signatures, electronic money, and other applications. Cryptology is the branch of mathematics that studies the mathematical foundations of cryptographic methods.

Basic Cryptographic Algorithms

A method of encryption and decryption is called a **cipher**. Some cryptographic methods rely on the secrecy of the algorithms; such algorithms are only of historical interest and are not adequate for real-world needs. All modern algorithms use a **key** to control encryption and decryption; a message can be decrypted only if the key matches the encryption key.

There are two classes of key-based encryption algorithms, **symmetric** (or **secret-key**) and **asymmetric** (or **public-key**) algorithms. The difference is that symmetric algorithms use the same key for encryption and decryption (or the decryption key is easily derived from the encryption key), whereas asymmetric algorithms use a different key for encryption and decryption, and the decryption key cannot be derived from the

encryption key.

Symmetric algorithms can be divided into **stream ciphers** and **block ciphers**. Stream ciphers can encrypt a single bit of plaintext at a time, whereas block ciphers take a number of bits (typically 64 bits in modern ciphers), and encrypt them as a single unit.

Asymmetric ciphers (also called **public-key algorithms** or generally **public-key cryptography**) permit the encryption key to be public (it can even be published in a newspaper), allowing anyone to encrypt with the key, whereas only the proper recipient (who knows the decryption key) can decrypt the message. The encryption key is also called the **public key** and the decryption key the **private key** or **secret key**.

Modern cryptographic algorithms are no longer pencil-and-paper ciphers. Strong cryptographic algorithms are designed to be executed by computers or specialized hardware devices. In most applications, cryptography is done in computer software.

Generally, symmetric algorithms are much faster to execute on a computer than asymmetric ones. In practice they are often used together, so that a public-key algorithm is used to encrypt a randomly generated encryption key, and the random key is used to encrypt the actual message using a symmetric algorithm. This is sometimes called hybrid encryption.

- **Strength of Cryptographic Algorithms**

Good cryptographic systems should always be designed so that they are as difficult to break as possible. It is possible to build systems that cannot be broken in practice (though this cannot usually be proved). This does not significantly increase system implementation effort; however, some care and expertise is required. There is no excuse for a system designer to leave the system breakable. Any mechanisms that can be used to circumvent security must be made explicit, documented, and brought into the attention of the end users.

In theory, any cryptographic method with a key can be broken by trying all possible keys in sequence. If using **brute force** to try all keys is the only option, the required computing power increases exponentially with the length of the key. A *32* bit key takes 2^{32} (about 10^{9}) steps. This is something anyone can do on his/her home computer. A system with *40* bit keys takes 2^{40} steps - this kind of computation requires something like a week (depending on the efficiency of the algorithm) on a modern home

computer. A system with *56* bit keys (such as DES) takes a substantial effort (with a large number of home computers using distributed effort, it has been shown to take just a few months), but is easily breakable with special hardware. The cost of the special hardware is substantial but easily within reach of organized criminals, major companies, and governments. Keys with *64* bits are probably breakable now by major governments, and within reach of organized criminals, major companies, and lesser governments in few years. Keys with *80* bits appear good for a few years, and keys with *128* bits will probably remain unbreakable by brute force for the foreseeable future. Even larger keys are sometimes used.

However, key length is not the only relevant issue. Many ciphers can be broken without trying all possible keys. In general, it is very difficult to design ciphers that could not be broken more effectively using other methods. Designing your own ciphers may be fun, but it is not recommended for real applications unless you are a true expert and know exactly what you are doing.

One should generally be very wary of unpublished or secret algorithms. Quite often the designer is then not sure of the security of the algorithm, or its security depends on the secrecy of the algorithm. Generally, no algorithm that depends on the secrecy of the algorithm is secure. Particularly in software, anyone can hire someone to disassemble and reverse-engineer the algorithm. Experience has shown that the vast majority of secret algorithms that have become
public knowledge later have been pitifully weak in reality.

The key lengths used in public-key cryptography are usually much longer than those used in symmetric ciphers. This is caused by the extra structure that is available to the cryptanalyst. There the problem is not that of guessing the right key, but deriving the matching secret key from the public key. In the case of RSA, this could be done by factoring a large integer that has two large prime factors. In the case of some other cryptosystems it is equivalent to computing the discrete logarithm modulo a large integer (which is believed to be roughly comparable to factoring when the moduli is a large prime number). There are public key cryptosystems based on yet other problems.

To give some idea of the complexity for the RSA cryptosystem, a *256* bit modulus is easily factored at home, and *512* bit keys can be broken by university research groups within a few months. Keys with *768* bits are probably not secure in the long term. Keys with *1024* bits and more

should be safe for now unless major cryptographical advances are made against RSA; keys of *2048* bits are considered by many to be secure for decades.

It should be emphasized that the strength of a cryptographic system is usually equal to its weakest link. No aspect of the system design should be overlooked, from the choice algorithms to the key distribution and usage policies.

Cryptanalysis and Attacks on Cryptosystems

Cryptanalysis is the art of deciphering encrypted communications without knowing the proper keys. There are many cryptanalytic techniques. Some of the more important ones for a system implementer are described below.

- **Ciphertext-only attack**: This is the situation where the attacker does not know anything about the contents of the message, and must work from ciphertext only. In practice it is quite often possible to make guesses about the plaintext, as many types of messages have fixed format headers. Even ordinary letters and documents begin in a very predictable way. For example, many classical attacks use frequency analysis of the ciphertext, however, this does not work well against modern ciphers.

  Modern cryptosystems are not weak against ciphertext-only attacks, although sometimes they are considered with the added assumption that the message contains some statistical bias.

- **Known-plaintext attack**: The attacker knows or can guess the plaintext for some parts of the ciphertext. The task is to decrypt the rest of the ciphertext blocks using this information. This may be done by determining the key used to encrypt the data, or via some shortcut.

  One of the best known modern known-plaintext attacks is linear cryptanalysis against block ciphers.

- **Chosen-plaintext attack**: The attacker is able to have any text he likes encrypted with the unknown key. The task is to determine the key used for encryption.

A good example of this attack is the differential cryptanalysis which can be applied against block ciphers (and in some cases also against hash functions).

Some cryptosystems, particularly RSA, are vulnerable to chosen-plaintext attacks. When such algorithms are used, care must be taken to design the application (or protocol) so that an attacker can never have
chosen plaintext encrypted.

- **Man-in-the-middle attack**: This attack is relevant for cryptographic communication and key exchange protocols. The idea is that when two parties, A and B, are exchanging keys for secure communication (e.g., using Diffie-Hellman), an adversary positions himself between A and B on the communication line. The adversary then intercepts the signals that A and B send to each other, and performs a key exchange with A and B separately. A and B will end up using a different key, each of which is known to the adversary. The adversary can then decrypt any communication from A with the key he shares with A, and then resends the communication to B by encrypting it again with the key he shares with B. Both A and B will think that they are communicating securely, but in fact the adversary is hearing everything.

  The usual way to prevent the man-in-the-middle attack is to use a public key cryptosystem capable of providing digital signatures. For set up, the parties must know each others public keys in advance. After the shared secret has been generated, the parties send digital signatures of it to each other. The man-in-the-middle can attempt to forge these signatures, but fails because he cannot fake the signatures.

  This solution is sufficient in the presence of a way to securely distribute public keys. One such way is a certificate hierarchy such as X.509. It is used for example in IPSec.

- **Correlation** between the secret key and the output of the cryptosystem is the main source of information to the cryptanalyst. In the easiest case, the information about the secret key is directly leaked by the cryptosystem. More complicated cases require studying the correlation (basically, any relation that would not be expected on the basis of chance alone) between the observed (or

measured) information about the cryptosystem and the guessed key information.

For example, in linear (resp. differential) attacks against block ciphers the cryptanalyst studies the known (resp. chosen) plaintext and the observed ciphertext. Guessing some of the key bits of the cryptosystem the analyst determines by correlation between the plaintext and the ciphertext whether she guessed correctly. This can be repeated, and has many variations.

The differential cryptanalysis introduced by Eli Biham and Adi Shamir in late 1980's was the first attack that fully utilized this idea against block ciphers (especially against DES). Later Mitsuru Matsui came up with linear cryptanalysis which was even more effective against DES. More recently, new attacks using similar ideas have been developed.

Perhaps the best introduction to this material is the proceedings of EUROCRYPT and CRYPTO throughout the 1990's. There can be found Mitsuru Matsui's discussion of linear cryptanalysis of DES, and the ideas of truncated differentials by Lars Knudsen (for example, IDEA cryptanalysis). The book by Eli Biham and Adi Shamir about the differential cryptanalysis of DES is the "classical" work on this subject.

The correlation idea is fundamental to cryptography and several researchers have tried to construct cryptosystems which are provably secure against such attacks. For example, Knudsen and Nyberg have studied provable security against differential cryptanalysis.

- **Attack against or using the underlying hardware**: in the last few years as more and more small mobile crypto devices have come into widespread use, a new category of attacks has become relevant which aim directly at the hardware implementation of the cryptosystem.

  The attacks use the data from very fine measurements of the crypto device doing, say, encryption and compute key information from these measurements. The basic ideas are then closely related to those in other correlation attacks. For instance, the attacker guesses some key bits and attempts to verify the correctness of the guess by studying correlation against her measurements.

Several attacks have been proposed such as using careful timings of the device, fine measurements of the power consumption, and radiation patterns. These measurements can be used to obtain the secret key or other kinds information stored on the device.

This attack is generally independent of the used cryptographical algorithms and can be applied to any device that is not explicitly protected against it.

More information about differential power analysis is available at http://www.cryptography.com.

- **Faults in cryptosystems** can lead to cryptanalysis and even the discovery of the secret key. The interest in cryptographical devices lead to the discovery that some algorithms behaved very badly with the introduction of small faults in the internal computation.

  For example, the usual implementation of RSA private key operations are very susceptible to fault attacks. It has been shown that by causing one bit of error at a suitable point can reveal the factorization of the modulus (i.e. it reveals the private key).

  Similar ideas have been applied to a wide range of algorithms and devices. It is thus necessary that cryptographical devices are designed to be highly resistant against faults (and against malicious introduction of faults by cryptanalysts).

- **Quantum computing**: Peter Shor's paper on polynomial time factoring and discrete logarithm algorithms with quantum computers has caused growing interest in quantum computing. Quantum computing is a recent field of research that uses quantum mechanics to build computers that are, in theory, more powerful than modern serial computers. The power is derived from the inherent parallelism of quantum mechanics. So instead of doing tasks one at a time, as serial machines do, quantum computers can perform them all at once. Thus it is hoped that with quantum computers we can solve problems infeasible with serial machines.

  Shor's results imply that if quantum computers could be implemented effectively then most of public key cryptography will

become history. However, they are much less effective against secret key cryptography.

Current state of the art of quantum computing does not appear alarming, as only very small machines have been implemented. The theory of quantum computation gives much promise for better performance than serial computers, however, whether it will be realized in practice is an open question.

Quantum mechanics is also a source for new ways of data hiding and secure communication with the potential of offering unbreakable security, this is the field of quantum cryptography. Unlike quantum computing, many successful experimental implementations of quantum cryptography have been already achieved. However, quantum cryptography is still some way off from being realized in commercial applications.

- **DNA cryptography**: Leonard Adleman (one of the inventors of RSA) came up with the idea of using DNA as computers. DNA molecules could be viewed as a very large computer capable of parallel execution. This parallel nature could give DNA computers exponential speed-up against modern serial computers.

  There are unfortunately problems with DNA computers, one being that the exponential speed-up requires also exponential growth in the volume of the material needed. Thus in practice DNA computers would have limits on their performance. Also, it is not very easy to build one.

Refrences: http://euclid.colorado.edu/~hiba/crypto/cryptanalysis.html

Cryptanalysis

There are several types of attacks that a cryptanalyst may use to break a code, depending on how much information they have. A ciphertext-only attack is one where the cryptanalyst has a piece of ciphertext (text that has already been encrypted), with no plaintext (unencrypted text). This is probably the most difficult type of cryptanalysis, and calls for a bit of guesswork. In a known-plaintext attack, the cryptanalyst has both a piece of ciphertext and the corresponding piece of plaintext.

Other types of attacks may involve trying to derive a key through trickery or theft. The "man-in-the-middle" attack is one example. In this attack, the cryptanalyst places a piece of surveillance software in between two parties that communicate. When the parties' keys are exchanged for secure communication, they exchange their keys with the attacker instead of each other.

The ultimate goal of the cryptanalyst however, is to derive the key, so that all ciphertext can be easily deciphered. A brute-force attack is one way of doing so. In this type of attack, the cryptanalyst tries every possible combination until the correct key is identified. Although using longer keys make the derivation less statistically likely to be successful, faster computers, continue to make brute-force attacks feasible. Networking a set of computers together in a grid, combines their strength; their cumulative power can be used to break long keys. The longest keys used, 128-bit keys, remain the strongest, and less likely to be subject to a brute-force attack.

At its core, cryptanalysis is a science of mathematics, probability and fast computers; cryptanalyst's also usually require some persistence, intuition, guesswork and some general knowledge of the target.

Cryptanalysis also has an interesting historical element; the famous Enigma machine, used by the Germans to send secret messages, was ultimately cracked by members of the Polish resistance and transferred to the British.

There are two general approaches in order to attacking a conventional encryption scheme:

- **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-cipher text pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used. If the attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.
- **Brute-force attack:** The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

Table (1) summarizes the various types of cryptanalytic attacks, based on the amount of information known to the cryptanalyst. The most difficult problem is presented when all that is available is the *ciphertext only*. In some cases, not even the encryption algorithm is known, but in general, we assume that the opponent does know the algorithm used for

encryption. One possible attack under these circumstances if the brute-force approaches of trying all possible keys.

If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis to the cipher text itself, general applying various statistical tests to it. To us this approach, the opponent must have some general idea of the type of plaintext that is concealed, such as English or French text, a Window EXE file, a Java source listing, an accounting file, and so on.

Table (1): Types attacks on encrypted messages.

Type of Attack	***Known of Cryptanalyst***
Ciphertext only	• Encryption algorithm. • Ciphertext to be decoded.
Known plaintext	• Encryption algorithm. • Ciphertext to be decoded. • One or more plaintext-ciphertext pairs formed with the secret key.
Chosen plaintext	• Encryption algorithm. • Ciphertext to be decoded. • Plaintext message chosen by cryptanalyst, together with its corresponding cipher text generated with secret key.
Chosen cipher text	• Encryption algorithm • Cipher text to be decoded • Purported cipher text chosen by cryptanalyst, together with its corresponding decrypted plain generated with the secret key

Chosen text	• Encryption algorithm • Cipher text to b decoded • Plaintext massage chosen by cryptanalyst, together with its corresponding cipher text generated with the secret key. • Purported cipher text chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Table (1) lists two other types of attacks: chosen ciphertext and chosen text. These are less commonly employed as cryptanalytic techniques but are possible avenues of attack.

1- Cipher text – only attack: the cryptanalyst has the cipher text of several message , all of which have been encrypted using the same encryption algorithm, the cryptanalyst job is to recover the plain text of many message as possible, or better yet to deduce the key (or keys) used to encrypted the message. In order to decrypted other message encrypted with the same keys.

Given: $C_1=E_k(p_1)$, $C_2=E_k(p_2)$,….. $C_i=E_k(p_i)$

Deduce: either p_1, p_2, p_i, k; or an algorithm.

To infer p_{i+1} from $C_i=E_k(p_{i+1})$

2- Known – plain text attack: the cryptanalyst has access not only to the cipher text of several message, but also to the plain text of those message. His job is to deduce the key (or keys) used to encrypt the message or an algorithm to decrypt any new message encrypted with the same key, (or keys).

Given: $P_1, C_1=E_k(p_1)$, $P_2, C_2=E_k(p_2)$,….. $P_i, C_i=E_k(p_i)$

Deduce: either k or an algorithm.

To infer P_{i+1} from $C_i=E_k(P_{i+1})$

3- chosen– plain text attack: the cryptanalyst not only has access to the cipher text and associated plain text for several message. But be also chooses the plain text that gets encrypted.

This is more powerful than a known-plain text attack, because the cryptanalyst can choose specified plaintext blocks to encrypted, ones that might more information about the key. His job is to deduce the key (or keys) used to encrypt the message or an algorithm to decrypt any new message encrypted with the same key (or keys).

Given: $P_1, C_1=E_k(p_1)$, $P_2, C_2=E_k(p_2)$,….. $P_i, C_i=E_k(p_i)$

where the cryptanalyst gets to choose p_1, p_2, p_i

Deduce: either k or an algorithm.

To infer P_{i+1} from $C_i=E_k(P_{i+1})$

Other

4- chosen– cipher text attack.

5- Adaptive-chosen-plaintext attack: this is a special case of a chosen-plaintext attack. Note only can the cryptanalyst choose the plaintext that is encrypted , but he can also modify his choice based on the result of pervious encryption- in a chosen-plain text attack,a cryptan-alyst might just be able to choose one large block of plain text to be encrypted ; in an adaptive-chosen-plaintext attack he can choose a smaller block of plaintext and then choose another based on the result of the first, and so forth.

Two more definitions are worth to be noted. An encryption scheme is **unconditionally secure** if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available, that is no, matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext, simply because the required information is not there. With the exception of a scheme known as the one-time pad, there is no encryption algorithm that is unconditionally secure. Therefore, all that the uses of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria:

- The cost of breaking the cipher exceeds the value of the encrypted information.
- The time required to break the cipher exceeds the useful lifetime of the information.

An encryption scheme is said to be **computationally secure** if the foregoing two criteria are met. The rub is that it is very difficult to estimate the amount of effort required to cryptanalyze ciphertext successfully. Table (2) shows how much time is involved for various key spaces. Results are shown for four binary key sizes.

For each key size, the result is shown assuming that it takes $1\mu s$ to perform a single decryption, which is a reasonable order of magnitude for today's machines. The final column of table (2) considers the results for a system that can process 1 million keys per microsecond.

Table (2): Average time required for exhaustive key search.

Key Size (bits)	***Number of Alternative Keys***	***Time required at 1 encryption /*** μs	***Time required at*** 10^6 ***encryptions /*** μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\ \mu s = 35.8$ **minutes**	2.15 **milliseconds**
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1142$ **years**	10.01 **hours**
128	$2^{128} = 3.4 \times 10^{38}$	$2^{167}\ \mu s = 5.9 \times 10^{36}$ **years**	5.9×10^{30} **years**
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ **years**	6.4×10^{6} **years**

Transposition Ciphers

- ***intorduction***

Transposition ciphers jumble the letters of the message in a way that is designed to confuse the attacker, but can be unjumbled by the intended recipient.

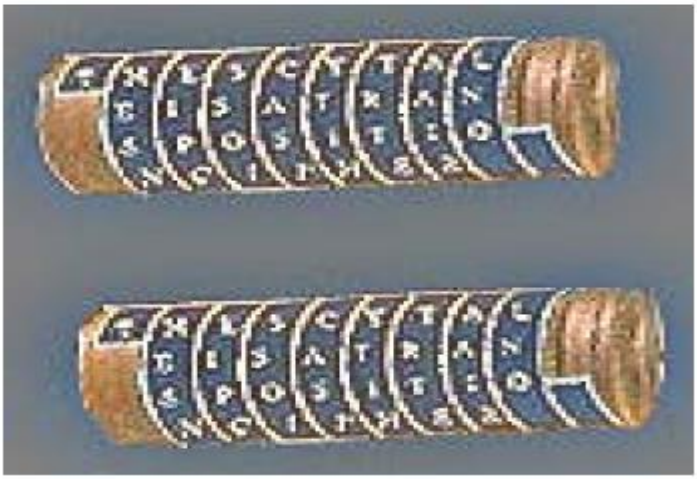
The concept of transposition is an important one and is widely used in the design of modern ciphers, as will be seen in subsequent chapters. Note
that the key must provide sufficient information to unscramble the ciphertext.

*** Scytale**

One of the earliest recorded uses of cryptography was the Spartan scytale
(circa 500 B.C.). **A** thin strip of parchment was wrapped helically around a
cylindrical rod and the message was written across the rod, with each letter
on a successive turn of the parchment. The strip was unwound and delivered
to the receiver. The message could then be decrypted with the use of an identical cylindrical rod. To anyone who intercepted the message, and did not understand the encryption technique, the message would appear to be a
jumble of letters. **A** clever cryptanalyst with access to a number of rods of various diameters will soon recover the plaintext.

For the scytale cipher, which is an example of a transposition cipher, the

key is the rod (or its diameter). This is a very weak cipher since the system
could be easily broken by anyone who understands the encryption method.



- **Scytale**

Example:

Encrypting

Suppose the rod allows one to write 4 letters around in a circle and 5 letters down the side of it. The plaintext could be: "Help me I am under attack"

To encrypt one simply writes across the leather...

```
_____________________________________________________________
    |   |   |   |   |   |   |
    |   | H | E | L | P | M |
    |___| E | I | A | M | U |__
        | N | D | E | R | A |   |
        | T | T | A | C | K |   |
        |   |   |   |   |   |   |
_____________________________________________________________
```

so the ciphertext becomes, "HENTEIDTLAEAPMRCMUAK" after unwinding.

Decrypting

To decrypt all one must do is wrap the leather strip around the rod and read across. The ciphertext is: "HENTEIDTLAEAPMRCMUAK" Every fifth letter will appear on the same line so the plaintext becomes

```
HELPM...return to the beginning once the end is reached
...EIAMUNDERATTACK
```

Insert spaces and the plaintext is revealed: "Help me I am under attack"

Columnar Transposition.

Suppose we have plaintext **SEETHELIGHT** and we want to encrypt this using a columnar transposition cipher. We first put the plaintext into the rows of an array of some given dimension. Then we read the ciphertext out of the columns. The key consists of the the number of columns in the array. For example, suppose we choose the key to be four, which means that we write the plaintext in four columns as

$$\begin{bmatrix} S & E & E & T \\ H & E & L & I \\ G & H & T & X \end{bmatrix},$$

Where the final **X** is used as to fill out the array. The ciphertext is then read from the columns, which in this case yields **SHGEEHELTTIX.** The intended recipient, who knows the number of columns, can put the ciphertext into an appropriate-sized array and read the plaintext out from the rows. Not surprisingly, a columnar transposition is not particularly strong. To perform a ciphertext only attack on this cipher, we simply need to test all
possible decrypts using c columns, where c is a divisor of the number of characters in the ciphertext.

Keyword Columnar Transposition

The columnar transposition cipher can be strengthened by using a keyword,

where the keyword determines the order in which the columns of ciphertext
are transcribed. We refer to this as a keyword columnar transposition cipher.

For example, consider encrypting the plaintext **CRYPTOISFUN** using a keyword columnar transposition cipher with keyword **MATH,** again using four columns. In this case, we get the array

$$\begin{array}{c} \begin{array}{cccc} M & A & T & H \end{array} \\ \hline \begin{bmatrix} C & R & Y & P \\ T & O & I & S \\ F & U & N & X \end{bmatrix} . \end{array}$$

The ciphertext is read from the columns in alphabetical order (as etermined by the keyword), so that, in this example, the ciphertext is

RO UPSXCTFYIN

Is it possible to conduct a ciphertext-only attack on a keyword columnar transposition cipher? It is certainly not as straightforward as attacking a non-keyword columnar cipher. Suppose we obtain the ciphertext ,

VOESA IVENE MRTNL EANGE WTNIM HTMEE ADLTR NISHO DWOEH

Which we believe was encrypted using a keyword columnar transposition.

Our goal is to recover the key and the plaintext. First, note that there are 45
letters in the ciphertext. Assuming the array is not a single column or row,
the array could have any of the following dimensions: 9 *x* 5. 5 x 9. 15 x **3** or **3** x 15. Suppose that we first try a 9 x ***5*** array. Then we have the ciphertext array in Table 1.1. We focus our attention on the top row of the array in Table 1.1. If we permute the columns as shown in Table 1.2,

we see the word **GIVE** in the first row and we see words or partial words in the other rows. Therefore, we have almost certainly recovered the key.

This method is somewhat ad hoc, but the process could be automated, provided we can automatically recognize likely plaintexts. In this example,

we have recovered the encryption key 24013 and the plaintext is **GIVE ME SOMEWHERE TO STAND AND I WILL MOVE THE EARTH.**

Table 1.1: Ciphertext Array

0	1	2	3	4
V	E	G	M	I
O	M	E	E	S
E	R	W	E	H
S	T	T	A	O
A	N	N	D	D
I	L	I	L	W
V	E	M	T	O
E	A	H	R	E
N	N	T	N	H

Table 1.2: Permuted Ciphertext Array

2	4	0	1	3
G	I	V	E	M
E	S	O	M	E
W	H	E	R	E
T	O	S	T	A
N	D	A	N	D
I	W	I	L	L
M	O	V	E	T
H	E	E	A	R
T	H	N	N	N

There are any ways to systematically mix the letters of the plaintext. For example, we can strengthen the columnar transposition cipher by allowing the permutation of columns and rows. Since two transpositions are involved, this is known as a double transposition cipher, which we briefly describe next.

Double Transposition Cipher

To encrypt with a double transposition cipher, we first write the plaintext into an array of a given size and then permute the rows and columns according to specified permutations. For example, suppose we write the plaintext **ATTACKATDAWN** into a **3** x 4 array:

$$\begin{bmatrix} A & T & T & A \\ C & K & A & T \\ D & A & W & N \end{bmatrix}.$$

Now if we transpose the rows according to (0,1,2) →(2,1,0) and then transpose the columns according to (0,1,2,3) → (3,1,0,2), we obtain

$$\begin{bmatrix} A & T & T & A \\ C & K & A & T \\ D & A & W & N \end{bmatrix} \longrightarrow \begin{bmatrix} D & A & W & N \\ C & K & A & T \\ A & T & T & A \end{bmatrix} \longrightarrow \begin{bmatrix} N & A & D & W \\ T & K & C & A \\ A & T & A & T \end{bmatrix}.$$

The ciphertext is read directly from the final array:

NADWTKCAATAT.

For the double t,ransposition, the key consists of the size of the matrix and the row and column permutations. The recipient who knows the key can simply put the ciphertext into the appropriate sized matrix and undo the permutations to recover the plaintext.

If Trudy happens to know the size of the matrix used in a double transposition, she can insert the ciphertext into a matrix of the appropriate size. She can then try to unscramble the columns to reveal words (or partial words).

Once the column transposition has been undone, she can easily unscramble the rows; This attack illustrates the fundamental principle of divide and conquer. That is, Trudy can recover the double transposition key in parts, instead of attacking the entire key all at Once. There are many examples of divide and conquer attacks throughout The remainder of this book. In spite of the inherent divide and conquer attack, the double transposition cipher is relatively strong---at least in comparison to many other classic cipher. The interested reader is directed to for a thorough cryptanalysis of the double transposition.

Cryptanalysis

Cryptanalysis requirements

The solution of nearly every cryptogram involves four basic steps–

1. Determination of the language used.
2. Determination of the general system used.
3. Reconstruction of the specific keys to the system.
4. Reconstruction of the plaintext.

1. Determination of the language used normally accompanies identification of the sender through traffic analysis or radio direction finding. If these forms of support are unavailable, or if an enemy uses several languages, the determination of the language may have to be made at a later stage of analysis.

2. Determination of the general system can come from several sources, such as

- A detailed study of the system characteristics, aided where necessary by character frequency counts, searches for repeated patterns, and various statistical tests. The study can extend beyond single messages to searching for patterns and repetitions between different messages with similar characteristics. This single step of system determination can be the most time consuming part of the analysis.
- Past history of system usage by the sender. In most cases, the user does not change systems regularly but uses the same system or set of systems from one day to the next. The specific keys may change regularly, but the general systems remain unchanged except at longer intervals.
- System indicators included with the traffic. Whenever the user has a choice of systems or a choice of keys within the system, the choice must be made known to the receiving cryptographer. The choice is usually communicated by some form of indicators, which can appear within the text of a message or as part of the externals. When the indicators reveal the choice of system, they are called system indicators or discriminates. When they denote specific frequently changing keys to the system, they are called message indicators. Once you learn just how indicators are used from day to day, they can provide a substantial assist to cryptanalysts.

3. Reconstruction of the specific keys to the system is an important step. Although the following step of plaintext recovery produces the most intelligence information, the full key reconstruction can speed recovery of future messages. The approach used to recover keys will vary greatly from system to system.

4. Reconstruction of the plaintext, although listed as the final step, will usually proceed simultaneously with the key reconstruction. Either step can come first, depending on the system and situation. Partial recovery of one aids in the recovery of the other. The two steps often proceed alternately, with each recovery of one helping in recovery of the other until a full solution is reached.

Substitution Ciphers

Like transposition, substitution is a crucial concept in the design of modern ciphers. in fact, Shannon's [133] two fundamental principles for the design of symmetric ciphers are confusion and diflusion, which, roughly, correspond to the classic concepts of substitution and transposition, respectively. These are still the guiding principles in the design of symmetric ciphers. In this section we discuss several classic substitution ciphers. We highlight, some of the clever techniques that can be brought to bear to attack such ciphers.

Simple Substitution

A simple substitution (or mono-alphabetic substitution) cipher is a generalization of the Caesar's cipher where the key can be any permutation of the alphabet. For the simple substitution, there are 26! = 288 keys available.

This is too many keys for any attacker to simply try them all, but even with This huge number of keys, the simple substitution cipher is insecure. Before We discuss the attack on the simple substitution, we consider a few special Types of related ciphers that have been used in the past.

Caesar's Cipher

in 50 R.C., Gaius Julius Caesar described the use of a specific cipher that, goes by the name of Caesar's c2pher.l In Caesar's cipher, encryption is accomplished by replacing each plaintext letter with its corresponding "shiftby- three" letter, that is, **A** is replaced by D, B is replaced by E, C is replaced by F, and so on. At the end of the alphabet, a wrap around occurs, with X replaced by **A, Y** replaced by B and Z replaced by C. Decryption is

accomplished by replacing each ciphertext letter with its corresponding left-shift-by-three letter, again, taking the wrap around into account.

Suppose we assign numerical values 0,1,. . . ,25 to the letters **A,** B, . . . , Z, respectively, Let pi be the ith plaintext letter of a given message, and ci the corresponding ith ciphertext letter. Then Caesar's cipher can be mathematically stated as ci = ***pi*** + **3** (mod 26) and, therefore, pi **=** ci **-** **3** (mod 26). In Caesar's cipher, the key is ***"3",*** which is not very secure, since there is only one key-anyone who knows that the Caesar's cipher is being used can immediately decrypt the message.

Simple Substitution Cryptanalysis

Trying all possible keys is known as an exhaustive key search, and this attack is always an option for Trudy. If there are N possible keys, then Trudy will, on average, need to try about half of these, that is; N/2 of the keys, before she can expect to find the correct key. Therefore, the first rule of cryptography is that any cipher must have a large enough keyspace so that an exhaustive search is impractical. However, a large keyspace does not ensure that a cipher is secure. To see that this is the case, we next consider an attack that will work against any simple substitution cipher and, in the general case, requires far less work than an exhaustive key search. This attack relies on the fact that statistical information that is present in the plaintext language "leaks" through a simple substitution. Suppose we have a reasonably large ciphertext message generated by a simple substitution, and we know that the underlying plaintext is English.