Cryptography

Symmetric Encryption – Classical Algorithms

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
and
Centre for Artificial Intelligence and Machine Learning (CAIML)
Indian Statistical Institute, Kolkata

March 12, 2021

Fundamental idea of symmetric encryption

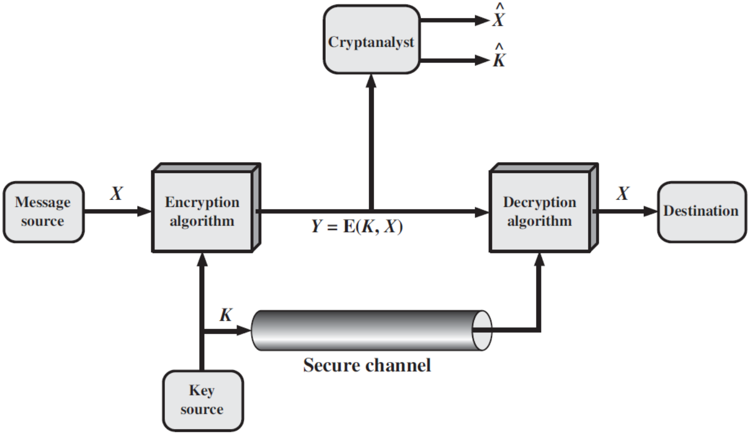
Given the plaintext P and the key K, we have the following rules:

- Encryption: Ciphertext $C = e(P, K)$.
- Decryption: Plaintext $P = d(C, K) = d(e(P, K), K)$.

Here, $e()$ and $d()$ denote the encryption and decryption functions, respectively.

<u>Note</u>: It is called symmetric because the key (K) is same for both encryption and decryption.

Modeling symmetric cryptosystem



Visualization of a symmetric cryptosystem

Caesar cipher

The Caesar cipher, made by Julius Caesar, involves replacing each letter of the alphabet with the letter three places further down in the same alphabet.

Plaintext	n	e	v	e	r		o	d	d		o	r		e	v	e	n
Ciphertext	Q	H	Y	H	U		R	G	G		R	U		H	Y	H	Q

Assuming that the letters in English alphabet are coded by the numbers between 0 to 25 (in the same order), we have the following rules in general:

- Encryption: Ciphertext $C = e(P, K) = (P + K)\%26$.
- Decryption: Plaintext $P = d(C, K) = (C - K)\%26$.

For the conventional Caesar cipher, $K = 3$.

Breaking the Caesar cipher

We can consider a generalization of the Caesar cipher by taking any arbitrary value of K.

If it is known that a given ciphertext is a generalized Caesar cipher, then a brute-force cryptanalysis is easy to perform.

1. Set $K = 0$.
2. Use K as the key and try to decrypt the ciphertext.
3. Set $K = K + 1$.

Note: This scheme will work if the language of the plaintext is known and easily recognizable.

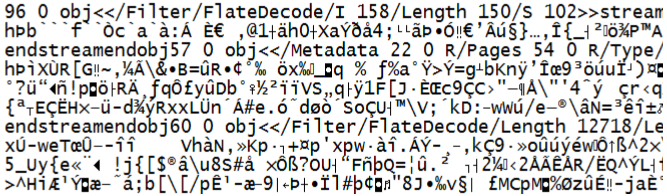
Monoalphabetic ciphers

In monoalphabetic ciphers the letters are substituted arbitrarily by some pre-defined rule.

While using monoalphabetic ciphers, which is a generalization of Caesar ciphers, the possible key space increases to 26! for English alphabet.

Breaking the monoalphabetic ciphers

If the nature of the plaintext (in noncompressed form) is known, then the patterns of the language might give us important clues.

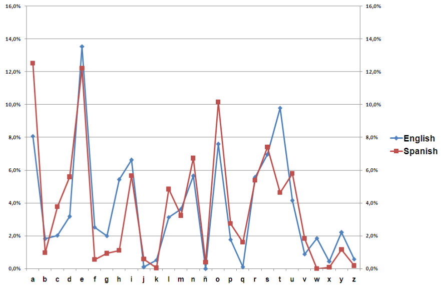


```
96 0 obj<</Filter/FlateDecode/I 158/Length 150/S 102>>stream
hþb```f``Òc`a`à:Á È€ ,@1†äh0†XaÝðå4;ʟʟãÞ•Ó‼€'Âú§}…,Î{_┤²ö¾P™A
endstreamendobj57 0 obj<</Metadata 22 0 R/Pages 54 0 R/Type/
hÞìXÙR[G‼~,¼Ã\&•B=ûR•¢°‰ öx‰_q % ƒ%a°Ÿ>Ý=g┴bKnÿ'Îœ9³öúuÏ┘)¤
°?ü“◄ñ!pö├RÄ¸ƒqÔ£yûDb°♀½²ïïVS„q├ÿ1F[J·ÈŒc9ÇC›"–¶Å\"'4ˆý çr‹q
{ª┬EÇËH×–ü-d¾ýRxxLÜn´Á#e.ó˜døò´SoÇU┤™\V;´kD:-wWú/e–®\âN=³êî±
endstreamendobj60 0 obj<</Filter/FlateDecode/Length 12718/Le
xÚ-weTœÛ--îî    VhàN,»Kp·┐+¤p'xpw·àî.ÁÝ-¸-,kÇ9·»oûúýéwÔ↑ß^2×
5_Uy{e«¨◄ !j{[$®â\u8S#å ×Ôß?OU┤“FñþQ=¦û.² ┐┤2¼‹2ÅÃÊÅR/ËQ^ÝL┤
>^HiÆ¹Ýæ–˜á;b[\[/pÊ¹-æ–9|←Þ†•Ïl#þ¢♬"8J•‰v§| £MCpM%ØzûÉ‼-jaÈ
```

Understanding the language patterns is hard from the compressed message

Breaking the monoalphabetic ciphers

Every language has some regularity. E.g., any English text contains high frequency monograms (e, t), digrams (th) and trigrams (the). This frequency pattern is useful for guessing the plaintext.



Relative frequency of letters in English and Spanish

Note: Breaking the ciphertext is hard if it is not sufficiently large.

Polyalphabetic ciphers

Polyalphabetic ciphers work in two phases as follows:

1. A set of related monoalphabetic substitution rules is used.
2. A key determines which particular rule is chosen for a given transformation.

As an example, the following rules can be stated:

- Encryption: Ciphertext $C = e(P, K) = (P_i + K_{i\%m})\%26$.
- Decryption: Plaintext $P = d(C, K) = (C_i - K_{i\%m})\%26$.

Playfair ciphers

Playfair cipher is the best-known multiple-letter cipher. It treats digrams in the plaintext as single units and translates them into the corresponding ciphertext digrams using a 5×5 matrix.

C	R	Y	P	T
O	G	A	H	B
D	E	F	I/J	K
L	M	N	Q	S
U	V	W	X	Z

The playfair matrix constructed from the key 'cryptography'

The matrix is prepared by filling in the letters of the keyword (without duplicates) in a row major fashion. The remainder of the matrix is filled in with the remaining letters in alphabetical order, and while doing so either I or J is chosen by the encipherer.

Playfair ciphers

Plaintext is encrypted by taking two letters at a time and using the following rules:

1. Repeating plaintext letters that are in the same pair are separated with a filler letter. E.g., ‘balloon’ and ‘baboon’ would be treated as ‘ba lx lo on’ and ‘ba bo on’, respectively (using ‘x’ as the filler letter).
2. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. E.g., ‘bo’ is encrypted as ‘OG’.
3. 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. E.g., ‘rv’ is encrypted as ‘GR’.
4. 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. E.g., ‘cf’ and ‘dh’ becomes ‘YD’ and ‘IO’ (or ‘JO’ as decided a priori), respectively.

Breaking the playfair ciphers

If the number of letters is 26 there can be a total $26 \times 26 = 676$ digrams. So, the identification of individual digrams is more difficult.

Furthermore, the relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult.

However, it still leaves much of the structure of the plaintext language intact and therefore a few hundred letters of ciphertext are generally sufficient for breaking it.

Hill ciphers

In general, the Hill ciphers can be represented as the following:

- Encryption: Ciphertext $C = e(P, K) = PK\%26$.
- Decryption: Plaintext $P = d(C, K) = CK^{-1}\%26 = PKK^{-1}$.

The encryption algorithm takes L successive plaintext letters and substitutes for them L ciphertext letters. The substitution is determined by L linear equations in which each character is assigned a numerical value (a = 0, b = 1, ..., z = 25). It can thus be visualized as a matrix. E.g., for $m = 3$, the system can be described as follows.

$$c_1 = (k_{11}p_1 + k_{21}p_2 + k_{31}p_3)\%26$$

$$c_2 = (k_{12}p_1 + k_{22}p_2 + k_{32}p_3)\%26$$

$$c_3 = (k_{13}p_1 + k23p_2 + k33p_3)\%26$$

Hill ciphers

The encryption with Hill ciphers can be represented as follows.

$$(c_1\ c_2\ c_3) = (p_1\ p_2\ p_3)\begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix}\%26$$

which is same as $PK\%26$.

Breaking the Hill ciphers

Although the Hill cipher is strong against a ciphertext-only attack, it is easily broken with a known plaintext attack.

For an $m \times m$ Hill cipher, suppose we have m plaintext-ciphertext pairs, each of length m. Now, as we use an invertible matrix for getting the plaintext back, a random guess can work by trials with additional plaintext-ciphertext pairs.

Vernam ciphers

Vernam ciphers work on binary data (bits) rather than letters. The encryption and decryption process of Vernam ciphers can be expressed as follows.

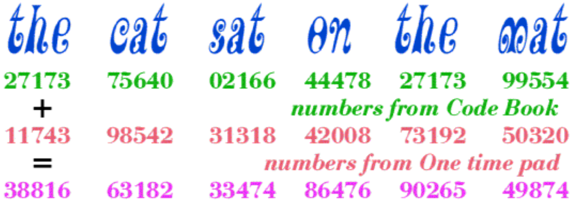
- Encryption: Ciphertext $C_i = e(P_i, K) = P_i \bigoplus K_i$.
- Decryption: Plaintext $P_i = d(C_i, K_i) = C_i \bigoplus K_i$.

Here, C_i is the i^{th} binary digit of plaintext, P_i is the i^{th} binary digit of plaintext, K_i is the i^{th} binary digit of the key, and $\bigoplus$ denotes the exclusive-or (XOR) operation.

The strength of this technique lies in the construction of key. Generally, a running loop of tape is used that eventually repeats the key. As a result, the system works with a very long but repeating key.

One-time pad

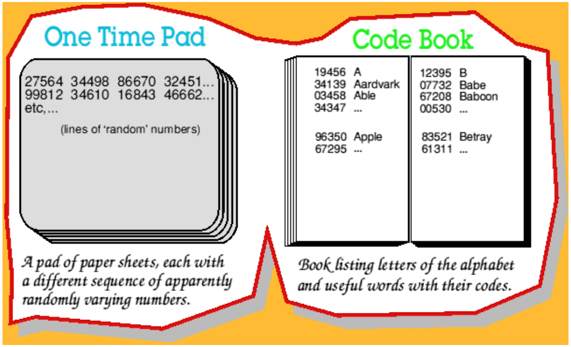
Random keys play important roles here. A separate one-time pad is used for every piece (generally represented as numbers) of code.



the	cat	sat	on	the	mat
27173	75640	02166	44478	27173	99554
+			*numbers from Code Book*		
11743	98542	31318	42008	73192	50320
=			*numbers from One time pad*		
38816	63182	33474	86476	90265	49874

An example of one-time pad

Breaking the one-time pad



A large code book of one-time pads is required to be maintained for large plaintexts

Transposition approaches

In transposition ciphers, some sort of permutation is done on the plaintext letters. We can arrange the plaintext “attack today!” with a rail fence of depth three as follows:

a a t a

- t c o y

- - t k d !

So, the ciphertext will be “AATATCOYTKD”.

The fences used in railways

Transposition approaches

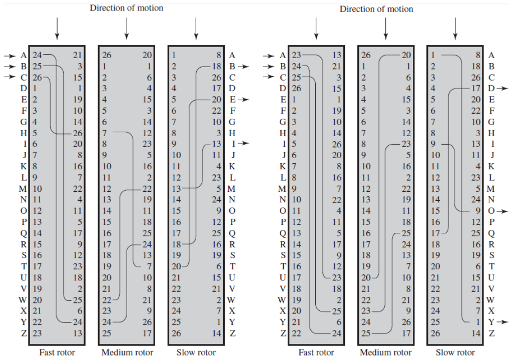
Transposition can also be done in a more complex way by reading the columns as per the ordering highlighted in the key to make it further secure.

```
Key:           4 3 1 2 5 6 7
Plaintext:     a t t a c k p
               o s t p o n e
               d u n t i l t
               w o a m x y z
Ciphertext:    TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

An example of transposition cipher

Rotor machines

In a rotor machine, each cylinder has 26 input pins and 26 output pins, with internal wiring that connects each input pin to a unique output pin.



A three-rotor machine with three connections

Rotor machines

An n-rotor machine provides $26 \times 26 \cdots \times 26$ (n times) $= (26)^n$ different substitution alphabets. We consider the value of n as period of the rotor machine.

A period of small length in a rotor machine thwarts any practical possibility of a straightforward plaintext-attack on the basis of letter frequency.