Classical Cipher - Part 1

Giulia Mauri

Politecnico di Milano

email: *giulia.mauri@polimi.it*
website: *http://home.deib.polimi.it/gmauri*

April 15, 2015

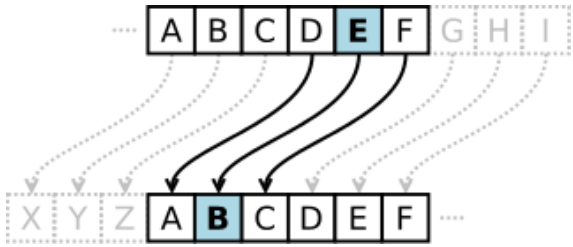
Overview

References:

[1] SAGE: http://sagemath.org
[2] PYTHON: http://docs.python.org

The Shift Cipher - History

The *shift cipher* was often used by Julius Caesar, who used it in his private correspondence. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. The cipher illustrated in figure uses a left shift of three, so that each occurrence of E in the plaintext becomes B in the ciphertext.



In the 19th century, the personal advertisements section in newspapers would sometimes be used to exchange messages encrypted using simple cipher schemes. Caesar ciphers can be found today in children's toys such as secret decoder rings. In 2011, Rajib Karim was convicted in the United Kingdom of "terrorism offences" after using the Caesar cipher to communicate with Bangladeshi Islamic activists discussing plots to blow up British Airways planes or disrupt their IT networks.

The Shift Cipher - Definition

Definition (Shift Cipher)

Let $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$. For $0 \leq K \leq 25$, define

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

and

$$d_K(y) = (y - K) \bmod 26$$

$(x, y \in \mathbb{Z}_{26})$.

It is defined over $\mathbb{Z}_{26}$ since there are 26 letters in English alphabet. We set up a correspondence between alphabetic characters and residues modulo 26.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

The Shift Cipher - Definition

Then, the shift cipher could be defined over $\mathbb{Z}_m$ for any modulus m.
Thus, $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_m$.

- $\mathcal{P}$ is a finite set of possible plaintexts;
- $\mathcal{C}$ is a finite set of possible ciphertexts;
- $\mathcal{K}$, the keyspace, is a finite set of possible keys;

Note (1)

Any secure encryption scheme must have a key space that is not vulnerable to exhaustive search.
This condition is NECESSARY but not SUFFICIENT.

Note (2)

Encrypting the same message two times with two different keys $K1$ and $K2$ is equivalent to encrypting only one time with $K3 = K1 + K2$. In this case, encrypting multiple times does not bring any advantage.

$$e_{K2}(e_{K1}(x)) = (x + K1 + K2) \bmod 26$$

The Shift Cipher - Sage Implementation

Here is an example over the upper-case letters of the English alphabet.

```
sage:  S = ShiftCryptosystem(AlphabeticStrings()); S
Shift cryptosystem on Free alphabetic string monoid on A-Z
sage:  x = S.encoding("The shift cryptosystem generalizes the
Caesar cipher."); x
THESHIFTCRYPTOSYSTEMGENERALIZESTHECAESARCIPHER
sage:  K = 7
sage:  y = S.enciphering(K, x); y
AOLZOPMAJYFWAVZFZALTNLULYHSPGLZAOLJHLZHYJPWOLY
sage:  S.deciphering(K, y)
THESHIFTCRYPTOSYSTEMGENERALIZESTHECAESARCIPHER
sage:  S.deciphering(K, y) == x
True
```

The Shift Cipher - Sage Implementation

Generate a random key for encryption and decryption:

```
sage:  K = S.random_key()
```

Decrypting with the key K is equivalent to encrypting with its corresponding inverse key.

```
sage:  S.enciphering(S.inverse_key(K), y) == x
True
```

For example, deciphering the message x can be accomplished by either shifting right by 19 (enciphering) or to the left by -7 (deciphering).

```
sage:  x = S.enciphering(19, y)
sage:  x = S.deciphering(7, y)
```

There exists also the function `brute_force(y)` that attempt a brute force cryptanalysis of the ciphertext y.

The Shift Cipher - Brute Force

Exercise

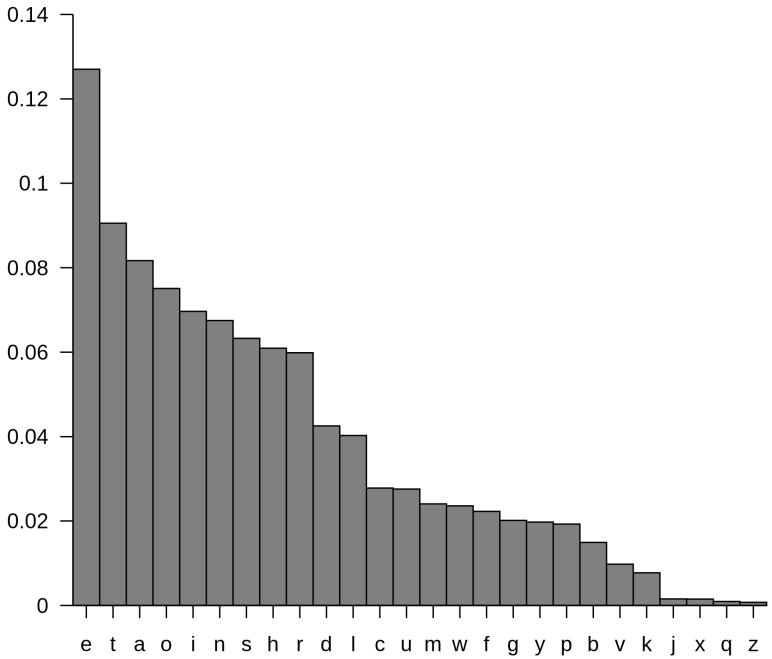
Crack this ciphertext: "LRZZOACZZQTDZYPESLEXLVPDFDHTDPC"

Solution

```
sage: S = ShiftCryptosystem(AlphabeticStrings())
sage: m = "LRZZOACZZQTDZYPESLEXLVPDFDHTDPC"
sage: y = S.encoding(m)
sage: for i in range(26):
....:     print i, S.deciphering(i, y)
0 LRZZOACZZQTDZYPESLEXLVPDFDHTDPC
1 KQYYNZBYYPSCYXODRKDWKUOCECGSCOB
...
10 BHPPEQSPPGJTPOFUIBUNBLFTVTXJTFS
11 AGOODPROOFISONETHATMAKESUSWISER ← Shift = 11
12 ZFNNCOQNNEHRNMDSGZSLZJDRTRVHRDQ
...
```

The Shift Cipher - Frequency Analysis

Many techniques of cryptanalysis use statistical properties of the English language. Various people have estimated the relative frequencies of the 26 letters.



letter	probability	letter	probability
A	.082	N	.067
B	.015	O	.075
C	.028	P	.019
D	.043	Q	.001
E	.127	R	.060
F	.022	S	.063
G	.020	T	.091
H	.061	U	.028
I	.070	V	.010
J	.002	W	.023
K	.008	X	.001
L	.040	Y	.020
M	.024	Z	.001

The Shift Cipher - Frequency Analysis

Exercise

Decipher the following message by using frequency analysis:
"VUJLBWVUHTPKUPNOAKYLHYFDOPSLPWVUKLYLKDLHRHUK
DLHYFVCLYTHUFHXBHPUAHUKJBYPVBZCVSBTLVMMVYN-
VAALUSVYLDOPSLPUVKKLKULHYSFUHWWPUNZBKKLUSFAOLY
LJHTLHAHWWPUNHZVMZVTLVULNLUASFYHWWPUNYHWWPUN-
HATFJOHTILYKVVYAPZZVTLCPZPALYPTBAALYLKAHW
WPUNHATFJOHTILYKVVYVUSFAOPZHUKUVAOPUNTVY".

Solution

```
sage: S = ShiftCryptosystem(AlphabeticStrings())
sage: m = "VUJLBWVUHTPK [...] HUKUVAOPUNTVY"
sage: y = S.encoding(m)
sage: fd = y.frequency_distribution().function()
sage: sorted(fd.items())
```

The Shift Cipher - Frequency Analysis

Solution

[(A, 0.062256809338521 4), (B, 0.027237354085603 1), (C, 0.011673151750972 8), (D, 0.015564202334630 4), (F, 0.035019455252918 3), (H, 0.089494163424124 5), (I, 0.007782101167315 18), (J, 0.019455252918287 9), (K, 0.058365758754863 8), (L, 0.105058365758755), (M, 0.011673151750972 8), (N, 0.035019455252918 3), (O, 0.031128404669260 7), (P, 0.073929961089494 2), (R, 0.003891050583657 59), (S, 0.031128404669260 7), (T, 0.046692607003891 1), (U, 0.093385214007782 1), (V, 0.085603112840466 9), (W, 0.046692607003891 1), (X, 0.003891050583657 59), (Y, 0.073929961089494 2), (Z, 0.031128404669260 7)]

The Shift Cipher - Frequency Analysis

Solution

```
sage: print AlphabeticStrings().alphabet().index("E")
4
sage: print AlphabeticStrings().alphabet().index("L")
11
sage: S.deciphering(11-4, y)
ONCEUPONAMIDNIGHTDREARYWHILEIPONDEREDWEAKANDWEARYOVER
MANYAQUAINTANDCURIOUSVOLUMEOFFORGOTTENLOREWHILEINODDED
NEARLYNAPPINGSUDDENLYTHERECAMEATAPPINGASOFSOMEONE
GENTLYRAPPINGRAPPINGATMYCHAMBERDOORTISSOMEVISITERIMUTTERED
TAPPINGATMYCHAMBERDOORONLYTHISANDNOTHINGMOR
```

The Affine Cipher - Definition

Definition (Affine Cipher)

Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$ and let

$$\mathcal{K} = \{(a, b) \in \mathcal{Z}_{26} \times \mathcal{Z}_{26} : gcd(a, 26) = 1\}.$$

For $K = (a, b) \in \mathcal{K}$, define

$$e_K(x) = (ax + b) \bmod 26$$

and

$$d_K(y) = a^{-1}(y - b) \bmod 26$$

$(x, y \in \mathbb{Z}_{26})$.

Note, with $a = 1$, the affine cipher is the same as the shift cipher. However, the key space is larger, making this cipher slightly more secure than the shift cipher.

The Affine Cipher - Sage Implementation

Encryption and decryption over the capital letters of the English alphabet.

```
sage:  A = AffineCryptosystem(AlphabeticStrings()); A
Affine cryptosystem on Free alphabetic string monoid on A-Z
sage:  x = A.encoding("The affine cryptosystem generalizes
the shift cipher."); x
THEAFFINECRYPTOSYSTEMGENERALIZESTHESHIFTCIPHER
sage:  a, b = (9, 13)
sage:  y = A.enciphering(a, b, x); y
CYXNGGHAXFKVSCJTVTCXRPXAXKNIHEXTCYXTYHGCFHSYXK
sage:  A.deciphering(a, b, y)
THEAFFINECRYPTOSYSTEMGENERALIZESTHESHIFTCIPHER
sage:  A.deciphering(a, b, y) == x
True
```

You can use the functions `random_key()`, `inverse_key()`, `brute_force(y)` also for the Affine cipher.

Homework 1 - The Caesar Cipher

Exercise

The ciphertext "KDDKMU" has been encrypted with a Caesar cipher.
Find the plaintext using a brute force attack.
Hint: use as reference slide 8 to obtain all the possible plaintexts.

Homework 1 - The Caesar Cipher

Exercise

The ciphertext "KDDKMU" has been encrypted with a Caesar cipher.
Find the plaintext using a brute force attack.
Hint: use as reference slide 8 to obtain all the possible plaintexts.

Solution

```
sage: S = ShiftCryptosystem(AlphabeticStrings())
sage: m = "KDDKMU"
sage: y = S.encoding(m)
sage: for i in range(26):
....:     print i, S.deciphering(i, y)
0 KDDKMU
1 JCCJLT
...
9 BUUBDL
10 ATTACK
```

← *Shift = 10, corresponds to the only meaningful word*

Homework 2 - The Affine Cipher

Exercise

Encrypt the plaintext "CLEOPATRA" using an affine cipher with encryption function $7x + 8$.
Hint: use as reference slide 14 to obtain the ciphertext.

Homework 2 - The Affine Cipher

Exercise

Encrypt the plaintext "CLEOPATRA" using an affine cipher with encryption function $7x + 8$.
Hint: use as reference slide 14 to obtain the ciphertext.

Solution

```
sage: A = AffineCryptosystem(AlphabeticStrings())
sage: x = A.encoding("CLEOPATRA")
sage: a, b = (7, 8)
sage: y = A.enciphering(a, b, x); y
WHKCJILXI
sage: A.deciphering(a, b, y)
CLEOPATRA
```

Homework 3 - The Affine Cipher

Exercise

The ciphertext MZDVEZC has been obtained using an affine cipher with encryption function $y = 5x + 12$. *Find the plaintext.*
Hint: invert the encryption function and compute $x = 5^{-1}(y - 12)$.

Homework 3 - The Affine Cipher

Exercise

The ciphertext MZDVEZC has been obtained using an affine cipher with encryption function $y = 5x + 12$. Find the plaintext.
Hint: invert the encryption function and compute $x = 5^{-1}(y - 12)$.

Solution

```
sage:  inv = 5^(-1)%26
21
sage:  21 * (-12) % 26
8
```

The decryption function is $x = 21y + 8$.

```
sage: A = AffineCryptosystem(AlphabeticStrings())
sage: y = A.encoding("MZDVEZC")
sage: a, b = (21, 8)
sage: x = A.enciphering(a, b, y); x
ANTHONY
```