A7010E Homework 1

Nico Ferrari (nicfer-0@student.ltu.se)

September 17, 2020

1. How do you define or perceive Computer Security?

Computer security is a broad field, mainly focused in protecting the confidentiality, integrity, and availability of computing devices and networks, hardware, software and the data and information transmitted, processed and stored in them. With the development in networking, pervasive remote connections etc... the hardware remains the safest thing in computer security. Especially during the last days, these have increase drastically, leaving the hardware the only safe aspect in computer security in the companies. The three most important goals of Computer security can be explained with the *CIA model*:

- Confidentiality: is concerned with viewing of data or information. Computer security requires privacy in data and information, restricting the accessibility to certain data only to the authorized users.
- Integrity: it refers to the necessity of making sure that data transmitted, processed, and stored has not been changed either accidentally or maliciously from its original form.
- Availability: making sure that the data, software, hardware are available. It requires that the authorized users should be able to access the resources they have access to.

The objective of COmputer security can be described using the *AAA Model*:

- Authentication: which is the process of proving who you claim to be. Identification can be established via passwords, single sign-on (SSO) systems, biometrics, digital certificates, and public key infrastructure.
- Authorization: which means providing minimum necessary level of access that a user should have based on its credentials. For example, a user might be able to type commands, but only be permitted to show execute certain commands. This may be based on geographical location restrictions, date or time-of-day restrictions, frequency of logins, or multiple logins by a single user.
- Accounting: which means loggin
  keeping track of all the actions executed on a system. this is very important in order to analyze any possible incident.

As we can see, the AAA model describes the methods through which the three important goals in computer security can be realized.

2. Describe a scenario or real-world application where symmetric encryption is combined with asymmetric encryption? Explain in detail the functionality of each encryption approach and the interaction between the two approaches?

A common scenario where symmetric encryption is combined with asymmetric one, is when big amount of data must be encrypted and it is necessary to share secretly a key between two entities over a public channel. The asymmetric key is used to share a common 'secret', the symmetric key, among the entities and this one is used to transform the plain text into ciphertext. The choice of using symmetric key encryption instead

of public key encryption for large ciphertext generation is motivated by the fact that public key encryption is slower and requires more computation power, a critical point in constrained devices like IoT. In order to share a secret over a public channel, the Diffie–Hellman algorithm (based on public key exchange) for sharing keys can be used. Suppose *Alice* and *Bob* need to share a secret key in a channel which could be eavesdropped by *Eve*. *Alice* and *Bob* can publically agree over a modulus **p** and base **g**. Afterward the can decide their own secret value **a** and send to the other entity the value

$$A = g^a mod p \tag{1}$$

they can now calculate a secret common key with:

$$shared\_secret\_key = A^b mod p \tag{2}$$

where b is the private key of the entity and A is the received message from the other entity. Eve will be able to read the messages containing A, but not able to generate the secret key shared between Alice and Bob. The shared key can now be used as key for symmetric encryption to generate the cipher text, and it can be changed over the time with the same procedure.

A protocol using both types of encryption is SSL/TLS. In fact, during a TLS handshake, the client and server agree upon new keys to use for symmetric encryption, called "session keys". The TLS handshake itself makes use of asymmetric encryption to authenticate the identity of the website's origin server and for security while the two sides shared and generate the session keys.

3. What are the challenges you have faced though conducting lab hand-on(1)?Please give answers to the challenging questions you find in the lab hand-on, and support your answers by posting screenshots?

I haven't faced any particular challenge. the symmetric algorithms where easy to understand and easy to apply also with the use of just a paper and pen.

4. Write an essay(1-2 pages)on the comparison of single-key and public-key encryption approaches? The comparison should consider how each approach works, the advantages and the disadvantages, the possible applications,the practical challenges,and the future developments.

Essay written in the paper attached in the submission.

5. What is your general impression on the first two weeks of the course? Please highlight what you liked and what you disliked.

The practical part was guided and I found no challenge. It resulted easy to follow but for sure a good learning tool. Also the topics explained during the lessons were easy to follow and gave many hints and explained things that were new to me. Some problems with the VMs occurred during the week but overall I can say that i enjoyed the first week of lessons.

Comparison Symmetric and Asymmetric Cryptography

Nico Ferrari

I. Types of Cryptography

Cryptography is technique of securing information and communications through use of algorithms so that only those person for whom the information is intended can understand it and process it. Cryptography tries to ensure the man principles of Information Security: confidentiality, integrity, authenticity. In order to accomplish this goal, three types of cryptography are used: symmetric encryption, asymmetric encryption and hash functions [1].

- Symmetric: uses the same cryptographic key to encrypt and decrypt data. Since it uses a single key and relatively simple algorithms, this processes can encrypt and decrypt vast amounts of data efficiently. In order to use a symmetric encryption algorithm, the involved parties must share the same key and this can lead to security problems, especially if the entities are connected only through a public channel. If the key is stolen, an attacker could be able to read, edit and send data to the entities. This type is usually used for encrypting data stored on a device.
- Asymmetric: it uses a pair of different keys, yet mathematically related keys: a private one and a public one. The public key is used to encrypt data and it is made available to anyone, even over the Internet, while the private key is kept only by the user / machine that generated the key pair and is used for decryption. Asymmetric algorithms are much more complex and their goal is to make relatively easy to compute the public key from the private key but nearly impossible to do the reverse and generate the private key from the public key. This type of cryptography is usually used for confirmation of identity to authorize transactions for cryptocurrencies, digital signatures, management of digital certificates.
- Hash functions: These do not use any key. A hash value with fixed length is calculated from the plain text using a one way function, making impossible to reconstruct the plain text from the hash. Many operating systems use hash functions to encrypt passwords.

II. Advantages and Disadvantages of Symmetric vs. Asymmetric Cryptography

Now days both the types of algorithms are still in use and many times even together, like in the SSL protocol [2]. The reason of this is because they both have advantages and disadvantages which usually affect speed and security.

The public key algorithms known thus far are relatively computationally costly compared with most symmetric key algorithms of apparently equivalent security. In fact, asymmetric encryption requires larger keys compared to the symmetric ones, using more memory to store them and more complex mathematical operations which can be not supported by some devices.

The public-key schemes' arithmetic operations result much harder compared with the single key schemes and performance can be a bottleneck in constrained devices where small CPUs are prevalent, e.g., IoT devices, or on network servers that have to compute many public-key operations per second.

Because of the complexity of the public-key schemes arithmetic's operations, network processes might have performance issues. In fact, devices might create delays due to the asymmetric encryption and decryption operations. This becomes even more crucial in IoT devices such as wireless sensors, where it can result in drainage of the batteries, issues with memory capacity or even incapability of execution of these operations [3].

On the other hand, symmetric cryptography carries a high risk around key transmission, as the key used to decrypt the cipher text is the same used to encrypt the plain text and must be secretly shared among the different entities. Every time the key gets shared, the risk of interception by an unintended third party exists. Here asymmetric cryptography offers better security because it uses a pair of keys composed by a public key which only gets used to encrypt messages, making it safe for anyone to have, and a private key to decrypt messages that remains secret. Since the private key never needs to be shared, it helps ensure only the intended recipient can decrypt encoded messages creating a tamper-proof digital signature.

III. Future of Cryptography

As technology advances so does our ability to decrypt and encrypt data. In order to simplify the process of sharing keys over public keys, overcoming the disadvantages presented by the public key algorithms, new algorithms which use neural networks are implemented [4]. Since the unlimited amount of data which can be used and acquired to be used to train the neural networks, these approaches are becoming robust techniques [5].

In the field of IoT, where the devices are lacking of resources and computational power, the research is focusing on the optimization of the key distribution and generation.

Due to the characteristics of the quantum computer,many existing public key cryptography schemes will be no longer safe in the quantum computer. This because well-known discrete logarithm problem (DLP) or the integer factorization

problem, which are the base of the asymmetric cryptography operations, will no longer be difficult under quantum computer [6]. In the IoT field, this problem could be even more devastating: since, due to the smaller keys used to perform asymmetric cryptographic operation on constrained devices, would be even easier to decrypt the messages for a quantum computer. In this case, lighter solution suitable for these devices are being studied like the one in [7][8].

REFERENCES

[1] G. J. Simmons, "Symmetric and asymmetric encryption," *Secure Communications and Asymmetric Cryptosystems*, vol. 11, no. 4, pp. 241–298, 2019.

[2] P. C. K. Alan O. Freier, Philip Karlton, "The SSL Protocol Version 3.0," 1996.

[3] G. Margelis, X. Fafoutis, G. Oikonomou, R. J. Piechocki, T. Tryfonas, and P. Thomas, "Practical Limits of the Secret Key-Capacity for IoT Physical Layer Security," *Communications (ICC) 2017 IEEE International Conference on*, pp. 311–316, 2016.

[4] T. Godhavari, N. R. Alamelu, and R. Soundararajan, "Cryptography using neural network," *Proceedings of INDICON 2005: An International Conference of IEEE India Council*, vol. 2005, no. I, pp. 258–261, 2005.

[5] B. Arora, N. Khatri, V. Niranjan, and K. Gate, "Application of Artificial Neural Network in Cryptography," pp. 229–232, 2019.

[6] T. Zhou, J. Shen, X. Li, C. Wang, and J. Shen, "Quantum Cryptography for the Future Internet and the Security Analysis," *Security and Communication Networks*, vol. 2018, 2018.

[7] D. Liu, N. Li, J. Kim, and S. Nepal, "Compact-{LWE}: Enabling Practically Lightweight Public Key Encryption for Leveled {IoT} Device Authentication," 2017.

[8] D. Liu, N. Li, and J. Kim, "Compact-LWE : a Public Key Encryption Scheme," pp. 1–12.

Step 9

1. yes the tool helps to understand the different algorithms thanks to the brief explanation of them and the cryptanalysis tools.
2. Yes, I think that cryptology is a fundamental tool which helps to increase the security in each of our daily actions. Cryptology. The study of cryptosystems, helps in fact to achieve : *confidentiality*, when transmitting or storing data, one does not want an eavesdropper to understand the contents of the transmitted messages;
   Authentication: in order to give to the the receiver of a message a proof that the message comes from a certain entity and not from somebody else;
   Integrity:to give to the receiver of certain data evidence that no changes have been made by a third party

Problem 1

For this problem I will use the Vigenère cryptosystem, which consists of applying periodically *r* Caesar ciphers.
Having the key:welcome with length r=7, a plain text “this is a plain text to encrypt and decrypto” with length l=44, we need to decide the alphabet used. In this case we can choose if considering the spaces “ “ as part of the alphabet or not.
For this exercise, and to increase the characters of being part of the alphabet, the spaces have been considered as part of the alphabet, having then:
Alphabet A=[a,…,z,” “]
Then we identify [0,..,26] with A, where 26 in this way:

0	1	2	3	...	23	24	25	26
a	b	c	d	...	x	y	z	“ “

We could include in the alphabet also the punctuation, special characters, capital letters, numbers etc… but for this case Has been chosen to keep only the characters used in the plain text , which for simplicity has been converted to lower case.
We can now represent the plain text as a sequence of integers like this:

t	h	i	s		i	s		...
19	7	8	18	26	8	18	26	...

The key will be periodically repeated till it reaches the same length of the plain-text (44) so it becomes:

	text	**length**
plain-text	this is a plain text to encrypt and decrypto	44
Repeated key	welcomewelcomewelcomewelcomewelcomewelcomewe	44

And also the key can be represented as a sequence of integers:

w	e	l	c	o	m	e	w	...
22	4	11	2	14	12	4	22	...

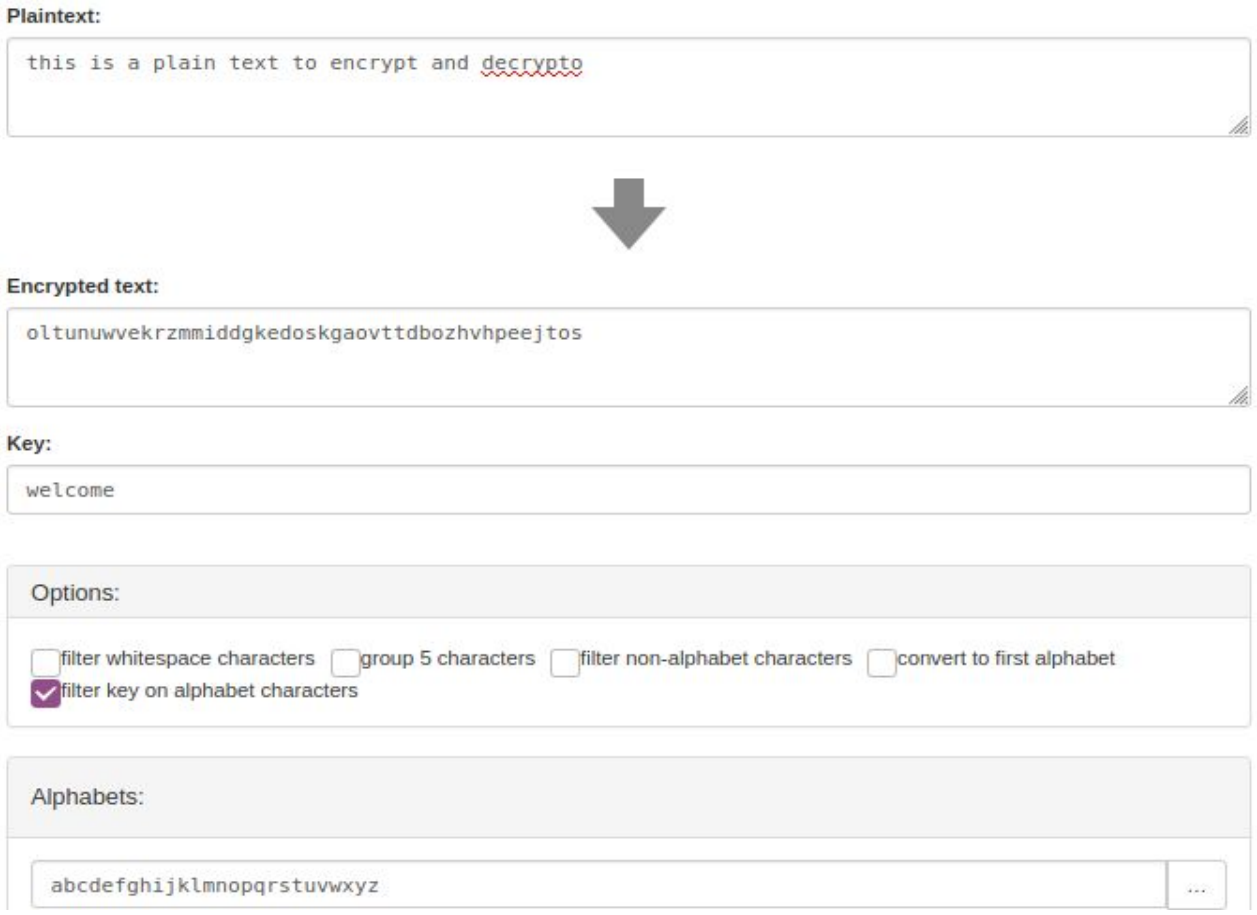
The cipher text is obtained by adding the key to the plain-text, where the addition is represented as: *char of ciphertext at position p = (Integer representation of the char at position p of the plain-text + integer representation of the char at position p of the repeated key) mod 27*, where 27 Is the size of the alphabet A.
Then the obtained sequence of integers is converted in the corresponding sequence of chars of the alphabet

Int. Repr.	14	11	19	20	13	20	22	...

cipher-text								
	o	l	t	u	n	u	w	..

The final cybertext is then “oltunuwvekrzmmiddgkedoskgaovttdbozhvhpeejtos”
in order to decrypt the message it is necessary to subtract the repeated key from the cipher text and we will obtain the same plain text we had in the beginning:
plaintext at position p = (Integer representation of the char at position p of the cipher text - integer representation of the char at position p of the repeated key) mod 27,

here is the verification with cryptool (online version because the VM was not working).



Problem 2

we have a cipher text “FRZDUGV GLH PDQB WLPHV EHIRUH WKHLU GHDWKV” obtained after applying to a plain-text a shift of 3 to each character (cesar algorithm with shift of 3). We suppose that the spaces are not shifted and left as spaces in the cipher text. In order to decrypt the message we must shift in the other direction each character of the ciphertext.
F→ C
R→ O
Z → W
D → A
….
and we obtain : “COWARDS DIE MANY TIMES BEFORE THEIR DEATHS” which is the plain-text.

Problem 3
plain-text : welcome to the world of cryptology with plain text, ciphers and deciphers
key: leap

To encrypt and decrypt using the Vigenere cryptosystem, beside the process explained in Problem 1, is possible to use the Vigenere table:

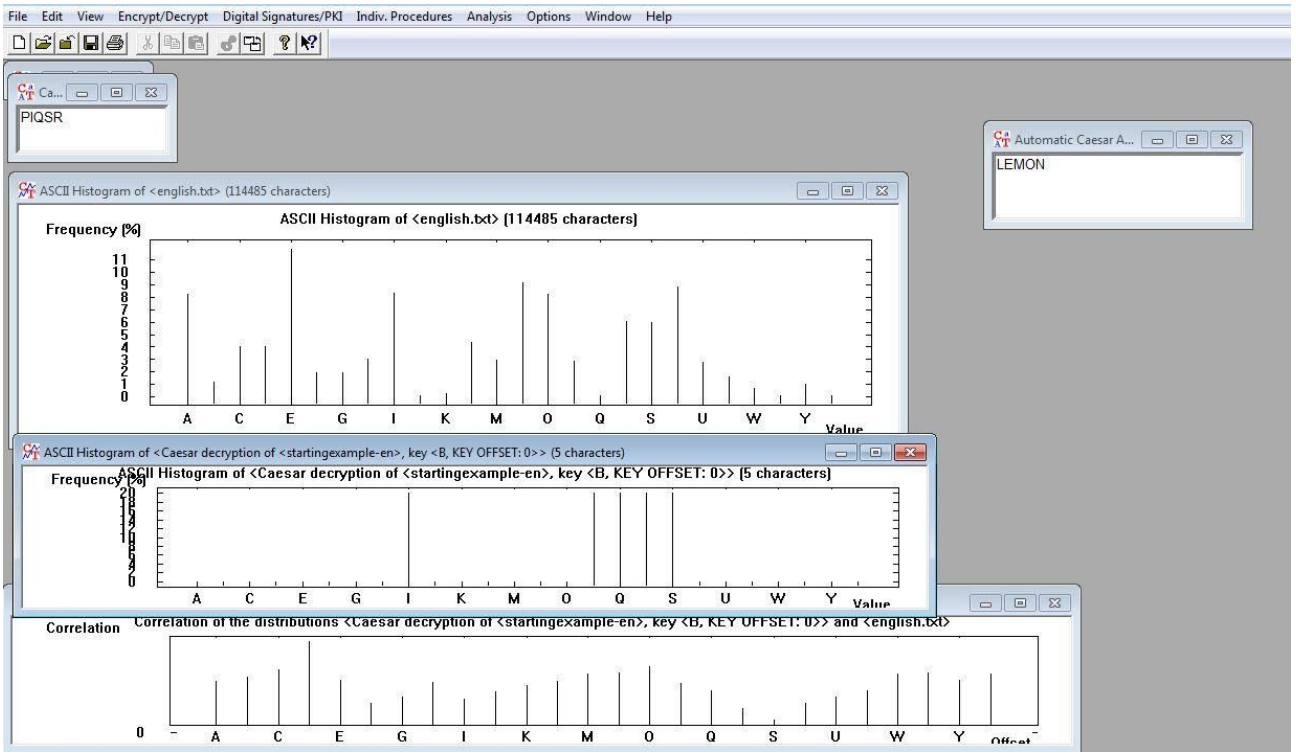
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T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

The first row of this table has the 26 English letters. Starting with the second row, each row has the letters shifted to the left one position in a cyclic way. For example, when B is shifted to the first position on the second row, the letter A moves to the end.
For simplicity, all spaces and punctuation are not changed and all letters are converted to lowercase, having an alphabet A=[a,…,z].
To encrypt, pick a letter in the plain text and its corresponding letter in the keyword, use the keyword letter and the plain-text letter as the row index and column index, respectively, and the entry at the row-column intersection is the letter in the ciphertext. For example, the first letter in the plain-text is “w” and its corresponding keyword letter is “l”. This means that the row of “L” and the column of “W” are used, and the entry “h” at the intersection is the encrypted result.

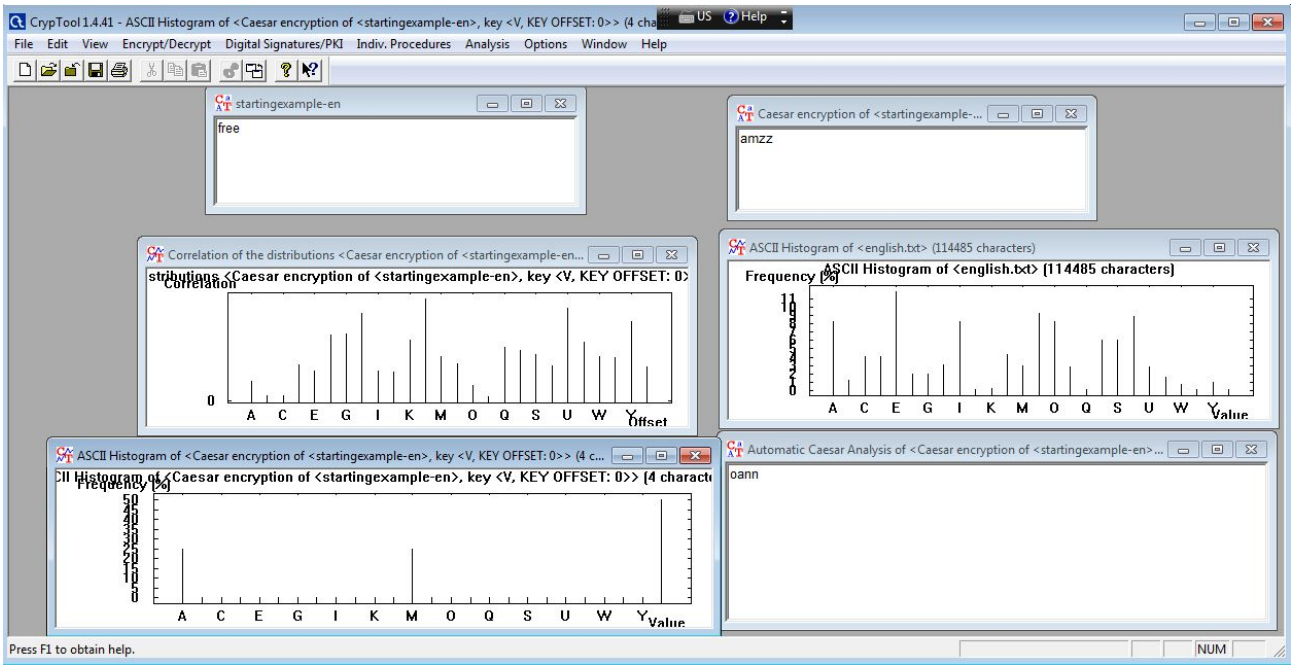
Afterwards, the second char of the plain text is “e” and for the key is “e”, so the intersection will be “l”.
the encrypted text will be:”hilrzqe iz xht hsrao sf rccpizpovj aiis tlptr ttix, cxalegd ens oicxalegd”.

To decrypt, we pick a letter in the ciphertext and its corresponding letter in the keyword, use the keyword letter to find the corresponding row, and the letter heading of the column that contains the ciphertext letter is the needed plain-text letter. To decrypt the first letter “h” in the ciphertext, we find the corresponding letter “l” in the keyword. Then, the row of “l” is used to find the corresponding letter “h” and that column provides the plain-text letter “w” . If we consider the second letter in the ciphertext “i”. This letter corresponds to the keyword letter “e" and row “E” is used to find “i”. Since “i” is on column “e”, the corresponding plain-text letter is “e”.

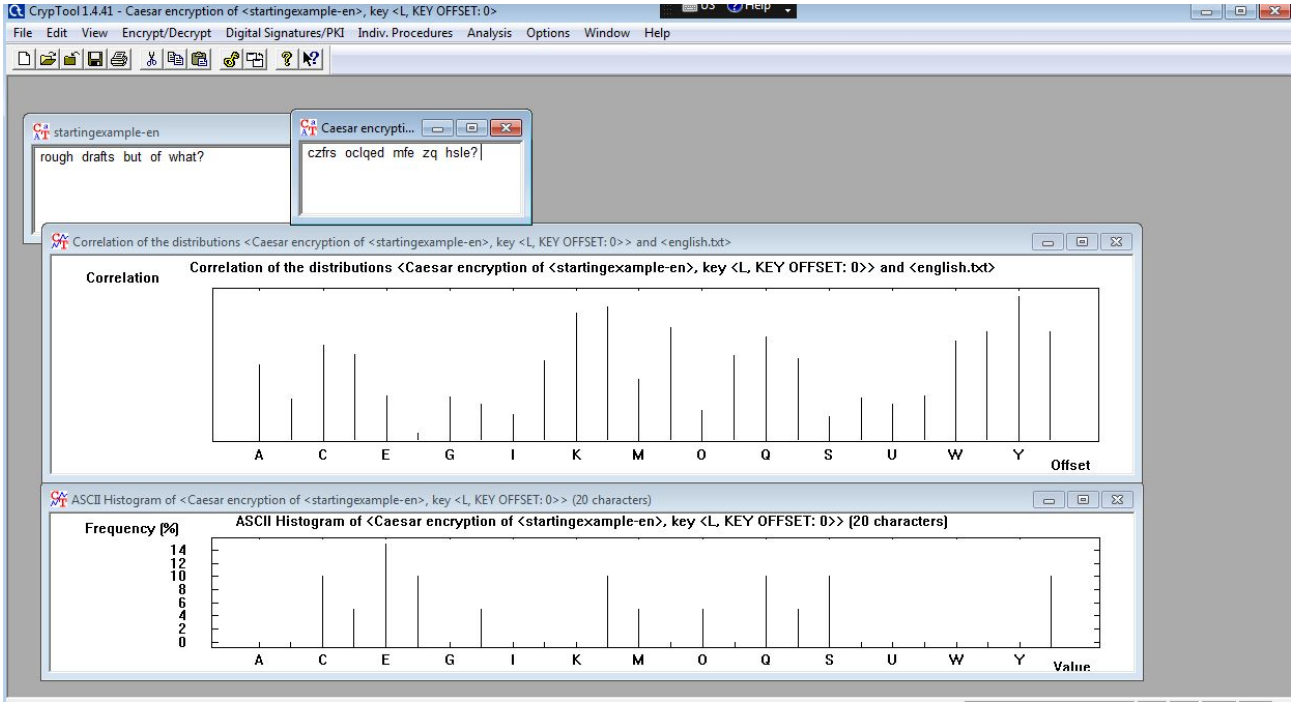
Problem 4:



Using the analysis tool, we can crack the key, finding “LEMON” as a decrypted key.
It would even be possible to crack it using brute force, since it is a short key, and then check which one is a common english word (if more than one we should try to decrypt the ciphertext with all the keys which are typical english words).
Now, using the method explained in the Problem 3 we can decrypt the ciphertext, finding the plain-text :”ATTACKATDAWN”.
To note that this analysis tool works good if we have to crack long encrypted texts. Short ones might lead to errors due to a possible frequency of infrequent letters in the used language as shown below:



Another case where this analysis tool does not work properly is where we try to crack encypted texts where the frequency of the letters of plain text is not correlated to the one of the language used to crack. For example, in english the most frequent letter is “e”, and if we encrypt a sentence without “e” could become difficult to crack using this analysis tool as shown below:



in this case the tool suggests a decryption using “Z” as key while “L” was used to encrypt. The sentence is taken from the book “A void” of George Perec, an entire book without the letter “e”.