Public Key Cryptography – Lab1

The substitution cipher

The set of keys is $\mathcal{K} = \{\sigma: \mathbb{Z}_n \to \mathbb{Z}_n \mid \sigma \text{ bijective}\}$. That is, a key is a permutation over $\mathbb{Z}_n$.
For a given encryption key, the corresponding decryption key is the encryption key's inverse function

$$\forall \sigma \in \mathcal{K}, \forall x, y \in \mathbb{Z}_n: \sigma(x) = e_\sigma(x) = y, \sigma^{-1}(y) = d_\sigma(y) = x$$

Example: $n = 27$

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

In practice, the decryption key can be obtained from the encryption key by using the following steps:

1. Reverse the associations (i.e. for $x \to y$, write $y \to x$)

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

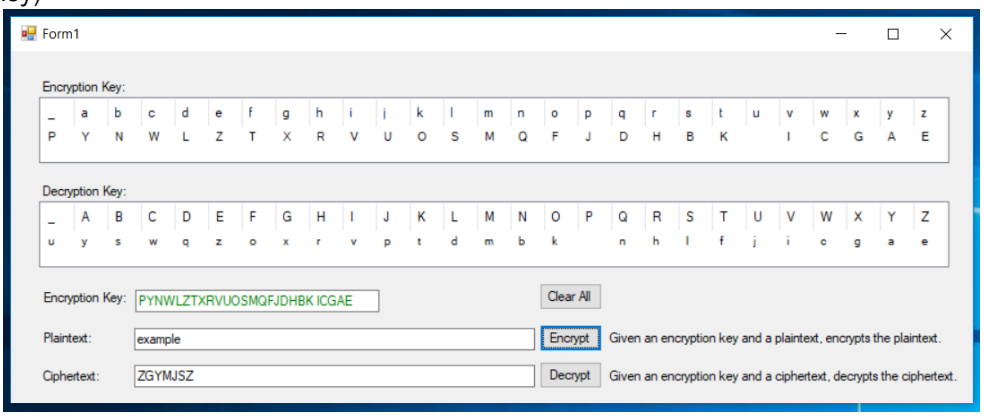
2. Reorder the table (i.e. sort the new x elements)

_	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
u	y	s	w	q	z	o	x	r	v	p	t	d	m	b	k	_	n	h	l	f	j	i	c	g	a	e

The application

The GUI consists of the following elements:

- two list views for the encryption key and the decryption key, respectively; used to easily view the keys
- a textbox where the user can enter **only** letters and space, without repeating two characters (these rules pre-validates the encryption key)
- a textbox for plaintext
- a textbox for ciphertext
- a **Clear All** button, used to clear the content of all textboxes
- an **Encrypt** button, used to encrypt the plaintext and produce the ciphertext
- a **Decrypt** button, used to decrypt the ciphertext and produce the plaintext



The functionality

Use case 1: encrypt a message	Use case 2: decrypt a ciphertext
1. the user inputs the encryption key in the textbox 2. the user inputs the plaintext to be encrypted (lowercase) in the **Plaintext** textbox 3. the user clicks the **Encrypt** button 4. the application validates the encryption key and the plaintext; if this step fails, an error is shown 5. the application encrypts the plaintext and writes the result inside the **Ciphertext** textbox	1. the user inputs the encryption key in the textbox 2. the user inputs the ciphertext to be decrypted (uppercase) in the **Ciphertext** textbox 3. the user clicks the **Decrypt** button 4. the application validates the encryption key and the ciphertext; if this step fails, an error is shown 5. the application computes the decryption key 6. the application decrypts the ciphertext and writes the result inside the **Plaintext** textbox

Vlad Cîncean / 933
cvie1883@scs.ubbcluj.ro