MysteryTwister

THE CRYPTO CHALLENGE CONTEST

Lady Liz Challenge

Author: Elijah Cross

February 2025

Lady Liz Cipher

Lady Liz is a cipher blending modern and classic cryptography. It encrypts the 95 printable ASCII characters in pairs using a two-character initialization vector (or IV), two keys that are permutations of the printable ASCII set, and a method called “relative encryption”, after which a new IV and key permutations are determined before continuing.

This cipher is named after my mother, who suddenly passed away in 2014. I’ve been working to make the cipher secure so that Elizabeth Cross can live forever.

Firstly, we introduce the cipher using an example.

Example encryption

In this example, the initial IV will be `JP`, and the two keys will be:

```
Super !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRTUVWXYZ[\]^_`
   abcdfghijklmnoqstvwxyz{|}~
```

```
Mario !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLNOPQRSTUVWXYZ[\]^_`
   bcdefghjklmnpqstuvwxyz{|}~
```

First, divide the characters of the plaintext:
`Our princess is in another castle!`
into pairs (add a random character to the end when necessary). Spaces are also encrypted:

```
Ou
r
pr
in
ce
ss
 i
s
in
 a
no
th
er
 c
as
tl
e!
```

IV and key setup

Next, apply the IV to the plaintext pair to create an intermediate block (IB) — for each character in the plaintext pair, find its position in the first key of the key pair using 1-indexing and add it to the position of the corresponding IV character.
For example, the `O` in `Ou` is at position 53, and the `J` in `JP` is at position 48. Add these ($53+48=101$) and if the sum exceeds 95, subtract 95 ($101 \mod 95 = 6$). The character at this final position (6) in the first key (the space) becomes the first character of the IB.
Repeat for the second character: `u` (position 2) and `P` (position 54) gives $2+54=56$.
The character at position 56 in the first key, `R`, becomes the second character of the IB.
After each iteration, the IV becomes an IB and a new IB is calculated.

Encryption

To encrypt the IB: Find the distance between the characters (space and `R`) in the first key of the key pair, "wrapping around" the key if necessary.

The distance from the space to `R` in the first key is 50. The first character in the ciphertext will be the 50th character in the second key of the key pair, `L`.
Then, in the second key of the key pair, find the distance from that ciphertext character to the first character of the IB — in this case, from `L` to the space. It's a distance of 51.

Again, the 51st character in the other key in the key pair determines the ciphertext character, `M`.
The ciphertext `LM` will also be the new IV to apply to the next plaintext pair. Note: if a pair is a character repeated (like `XX`), the distance from it to itself is 95.

Steps before the next encryption

If the IB characters that were just encrypted are different, swap their positions in the first key of the key pair. Next, if the ciphertext characters that were just created are different, swap their positions in the second key of the key pair.

The value of the first distance calculated (50) decides the order in which the keys will be used next. If it is even, the order stays the same; if it is odd, swap the order.

Finally, both keys in the set will shift their characters forward by the value of the second distance calculated (51).

These swaps and shifts lead to the keys being configured as follows:

```
GHIJKLMNOPQ TUVWXYZ[\]^_`abcdfghijklmnoqstvwxyz{|}~SuperR!"#$%&'()
   *+,-./0123456789:;<=>?@ABCDEF
```

and

```
GHIJKMNOPQRSTUVWXYZ[\]^_`bcdefghjklmnpqstuvwxyz{|}~Lario !"#$%&'()
   *+,-./0123456789:;<=>?@ABCDEF
```

Continuing in this fashion, the final ciphertext as a single string will read:

```
LMLPB';qd?Nb} 33yUNe$hs8P=}pLb]N*]
```

Decryption

To decrypt this example, begin with the keys in the original configurations. Starting with the second key of the key pair, and at the first character of the ciphertext pair `L`), move forward the distance that the second character of the pair represents according to the first key of the key pair `M`, or 51), landing on the space — the first IB character.

Then, in the first key of the pair, and from the character that was landed on in the second key of the pair, move forward the distance that the first character of the ciphertext pair `L`) represents according to the second key of the pair. Starting at the space character and moving forward a distance of `L` (50) leads to `R`.

To get the original plaintext pair from the IB, use the first key of the pair and subtract the initial IV's character positions from the IB's character positions, adding 95 if the result is 0 or negative.

After performing the same swaps as in the encryption process, the values the ciphertext characters originally represented are used to perform the correct shifts for the key cycle and the order of characters in both keys before continuing decryption. The previous ciphertext block is the new IV.

Challenge

Recover the English plaintext from the given ciphertext. You may confirm you've done so by submitting the first three surnames that appear in the plaintext, separated by commas without any spaces. The challenge was created using `lady-liz.py`, a Python script for encryption and decryption.

Additional files

- `ciphertext.txt` – Ciphertext to decrypt.
- `lady-liz.py` – Python script for encryption and decryption.