

TP 01: Overview of Cryptography

Jean-Sébastien Coron

Université du Luxembourg

1 The Vernam Cipher

The *Vernam Cipher* (see [1]) is a stream cipher defined on $\{0, 1\}$. It takes as input a binary message $m_1 m_2 \dots m_t \in \{0, 1\}^t$ and a binary key $k_1 k_2 \dots k_t \in \{0, 1\}^t$ of the same length and outputs a ciphertext $c_1 c_2 \dots c_t \in \{0, 1\}^t$ where

$$c_i = m_i \oplus k_i, 1 \leq i \leq t$$

If the key is randomly chosen and used only once, the Vernam cipher is called the *one-time pad*.

1) Implement the Vernam Cipher in C. Your program must take as input a string of character and output the resulting ciphertext. The key string will be generated using the (insecure) C random generator.

The Vernam cipher can proven perfectly secure if the key string is random and only used once. Namely, given a ciphertext, any t -bit binary string of plaintext is equally likely. This implies that an attacker who does not know the key obtains no information about the plaintext.

2) However, if the key is reused then the cipher can be attacked. Assume that the same key-bits are repeated every 32-bits, i.e. we have

$$K = k_1 k_2 \dots k_{32} k_1 \dots k_{32} k_1 \dots k_{32} \dots$$

Show how the plaintext can be recovered without knowing the key K . Show that your attack works by implementing it in C.

2 Décomposition

Ecrire un programme `factor` prenant en entrée un entier n et affichant la décomposition en produit de nombres premiers de n , en utilisant l'algorithme "naïf".

```
$ factor 150
(2,1) (3,1) (5,2)
```

car $150 = 2^1 \cdot 3^1 \cdot 5^2$.

Executez votre programme pour des entiers de plus en plus grands. Que constatez-vous ?

3 Algorithme d'Euclide

Ecrire un programme `pgcd` prenant en entrée deux entiers et renvoyant leur `pgcd`, en utilisant l'algorithme d'Euclide.

```
$ pgcd 12 15  
3
```

References

1. A.J. Menezes, P. C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*