CRYPTOGRAPHY #01

Stream ciphering and PRNG

Jacek Tchórzewski, jacek.tchorzewski@pk.edu.pl

1. Basic Notation

a) **Cryptology** - is a branch of science concerning methods of secure information exchange and storage. Cryptology can be divided into cryptography and cryptanalysis.
b) **Cryptography** - creating and investigating ciphers, transformations and other securing methods. The idea is to represent a message in a form that only a legitimate user can reveal it.
c) **Cryptanalysis** - part of cryptology oriented on breaking secure algorithms, transformations, and other securing methods. The idea is in finding breaches and weaknesses which allow an unauthorized user to reveal (wholly or partially) a secured message.
d) **Cryptogram** - is a presentation of the message in a form that only authorized recipients can reveal the content.
e) **Cipher** - algorithm which creates appropriate, secure cryptograms.
f) **Symmetric Ciphering** – a scheme that involves one, the same cryptographic key for ciphering and deciphering purposes. Before communication begins, a party which has generated a key has to distribute it via a secure channel. Symmetric schemes can be divided into 2 groups: Block Ciphers and Stream Ciphers.
g) **Block Ciphers –** cryptographic key can be smaller than the message which will be encrypted. The message is divided into blocks and each block is being ciphered separately with cryptographic key usage. Block Ciphers can use few operational modes (such as Electronic Codebook, Cipher Block Chaining, etc.), thus encryption itself depends on the key and the operational mode. Example: AES, DES.
h) **Stream Ciphers** – also called one-time-pad ciphers (for example Vernam Cipher). The key is at least as long as the message and can be used once. The key generation may involve secure Pseudo-Random Number Generators (PRNGs) such as Blum Blum Shub.
i) **GCD(a, b)** – Greates Common Divisor. The highest number which is a divisor of *a* and *b* at the same time. For example: GCD(4, 6) = 2, GCD(21, 16) = 1, GCD(21, 14) = 7.

2. Vernam Ciphering and Deciphering Schemes

   Developed in 1917 by Gilbert Vernam stream cipher, which can not be broken (under some assumptions). The whole scheme based on XOR operation. Let's denote message to be ciphered by M, cryptographic key by K and ciphertext by C. Than, ciphering looks as follows:

   $$C = M \oplus K$$

   And deciphering scheme as follows:

   $$M = C \oplus K$$

   Where key K can be used only once, must have at least the same length as M and must be generated randomly.

3. Blum Blum Shub PNRG

   BBS is a pseudorandom number generator (PNRG) presented in 1986 by L. Blum, M. Blum, and M. Shub. BBS is slow, however, is also secure (under some assumptions), thus can be used in cryptography. The basic algorithm looks as follows:

   1) Find two distict, prime numbers *p* and *q* such that: *p mod 4 = 3* and *q mod 4 = 3*.
   2) Calculate *n = p * q*.
   3) K = 0.
   4) Find random seed *s* from range (1, n-1) such that binary length of *s* isn't lower than 0.25 of binary length of *n,* and *GCD(n, s) = 1.*
   5) Calculate $x_1 = (s*s) \bmod n.$
   6) Calculate $x_t = (x_{t-1} * x_{t-1}) \bmod n.$
   7) Append Least Significant Bit of x_t to the K.
   8) Repeat steps 6) and 7) until K has appropriate length.

   Security of BBS is as high as hard is factorization of *n*, thus numbers *p* and *q* should be big enough.

Exercise 1 (warm up):

What values will be stored in *a* and *b*? Why?

1) a = 4
2) b = 14
3) $\boldsymbol{a} = \boldsymbol{a} \oplus \boldsymbol{b}$
4) $\boldsymbol{b} = \boldsymbol{a} \oplus \boldsymbol{b}$
5) $\boldsymbol{a} = \boldsymbol{a} \oplus \boldsymbol{b}$
6) a = ?, b = ?

Exercise 2:

Write a program accordingly to the steps below. What will be bit length of *n*? Execute program multiple times.

1) Generate number *p*. *p* can be any random number, however it must have exactly 128 bits.
2) Generate number *q*. *q* can be any random number, however it must have exactly 64 bits.
3) Calculate *n = p * q*.
4) len(n) = ?

Exercise 3:

Write a program which will:

1) Generate pseudorandom cryptographic key K with usage of BBS generator.
2) *p* and *q* both must have exactly 512 bits.

Exercise 4:

Write a program which will:

1) Read an input from a keybord (only ASCII characters).
2) Cipher it with Vernam Cipher usage and key K generated in Ex. 3.
3) Display ciphertext.
4) Decipher the ciphertext.
5) Display original message.