Logistic Map Cipher v0.4

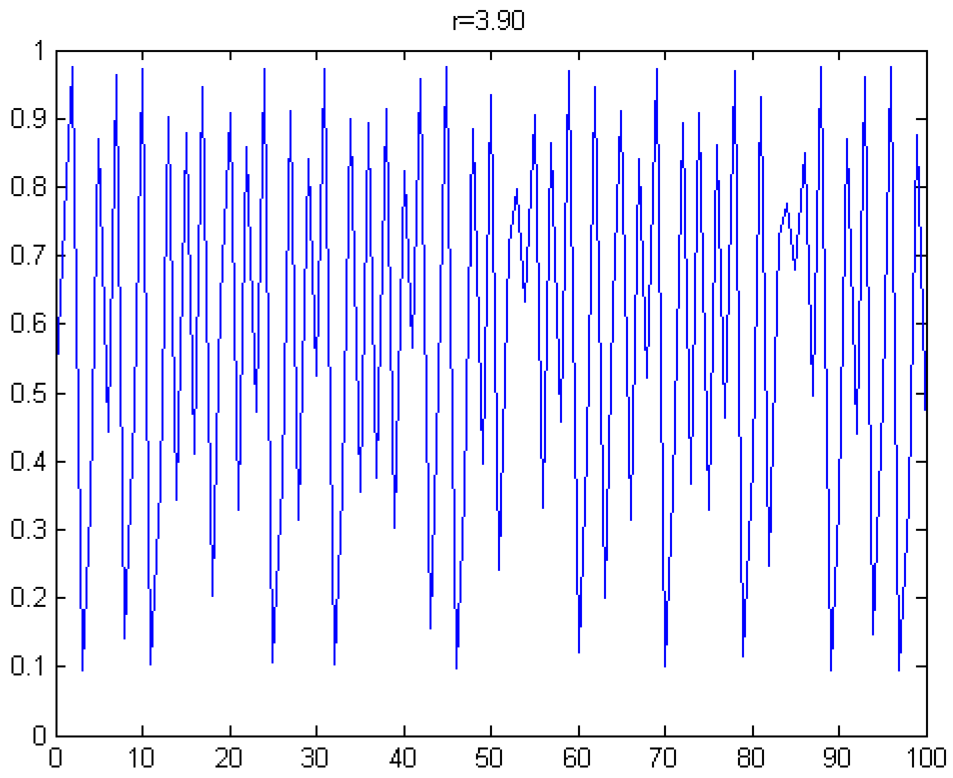
Eleftherios "Lefty" Ioannidis

1 How it works

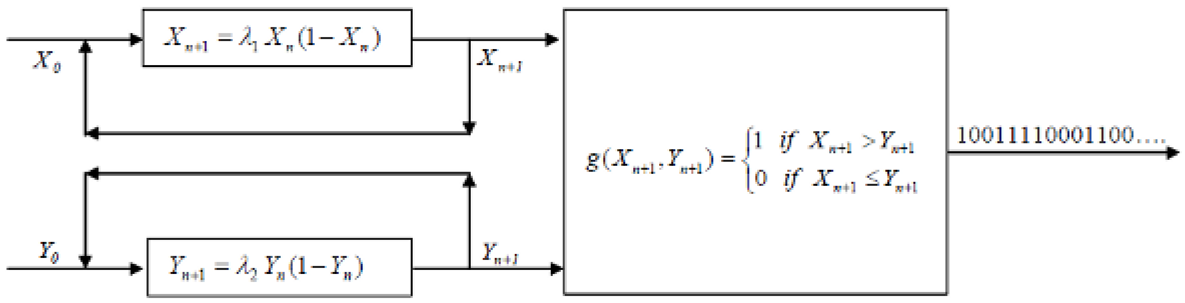
The Logistic Map cipher, hereby refered as LMCipher, utilizes the pseudo-random properties of the logistic map, a recurence relation of degre 2 with interesting chaotic properties:

$$x_{n+1} = rx_n(1 - x_n)$$

where $x_n \in (0, 1)$ and $r \in (1, 4)$. It has been shown to exhibit chaotic behaviour for values of $r \approx 3.9$.



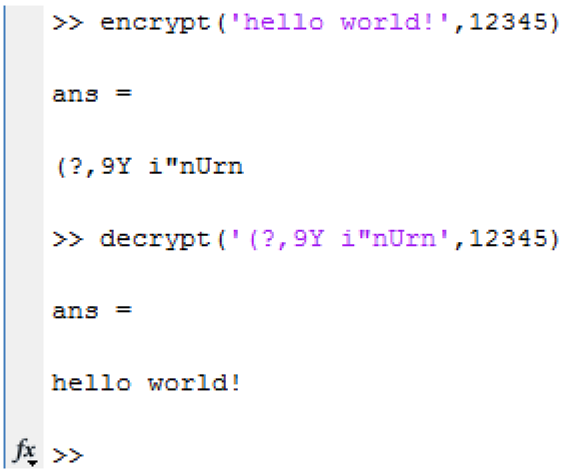
The algorithm choses values for r near 3.9 based on the given key. Then a pseudo-random bit generator using two logistic maps running simultaneously, as depicted below, is initiated. Using this algorithm we can achieve great levels of statistical randomness, making this algorithm impossible to crack using statistical cryptanalysis. For more information on the generator's statistical improbability read[2]. This cipher is also diffiicult to crack using exaustive search, for a sufficiently long key.


$X_{n+1} = \lambda_1 X_n(1-X_n)$
X_0
X_{n+1}
Y_0
$Y_{n+1} = \lambda_2 Y_n(1-Y_n)$
Y_{n+1}
$g(X_{n+1}, Y_{n+1}) = \begin{cases} 1 & if\ X_{n+1} > Y_{n+1} \\ 0 & if\ X_{n+1} \le Y_{n+1} \end{cases}$
10011110001100....

Using the pseudo-random bit generator (prbg.h) library, a one time pad is created. Then the input file is XOR'd to the one time pad. The result is the password protected output file. When tested by the NIST statistical test, the LM pseudo-random bit generator gives statistical results ideal for cryptography. This first version of LM is an 8-bit block cipher, even though it could work just as well as a stream cipher.

2 Matlab source code

The cipher encryption/decryption matlab routines can be found at `http://appslinalgebra.sourceforge.net/` At this moment Matlab .m files and C/C++ source are available. Find the latest UNIX/Linux version at: `http://lmcipher.sourceforge.net/` To use the Matlab m files, uncompress the archive in your working directory and use them as depicted below:



```
>> encrypt('hello world!',12345)

ans =

(?,9Y i"nUrn

>> decrypt('(?,9Y i"nUrn',12345)

ans =

hello world!

>> 
```

3 C/C++ code

To use C/C++ source use: $make && sudo make install for UNIX based distributions and then run:
$lmcipher $< key >$ [*inputfile*] [*outputfile*].
Use $lmcipher −−help for details.

References

[1] Schneier B. Applied Cryptography-Protocols, algorithms and source code in C. John Wiley & Sons, New York, USA, 1996.

[2] Vinod Patidar and K. K. Sud, *A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing*. Informatica 33 (2009) 441452 [p.441].

[3] Wikipedia, the Free Encyclopedia: `http://en.wikipedia.org/wiki/Logistic_map`