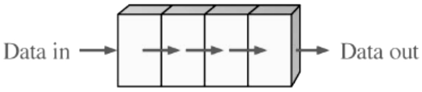
CRYPTOGRAPHY IN p-ADIC ANALYSIS

JOHAN VONK

1. Introduction

In computer science, one of the most important functions is one that creates pseudo-random numbers which appear to have no predictable pattern. Pseudo-random numbers are especially important to cryptography where they are used to encode or decode messages. One way of generating pseudo-random numbers is by using shift registers.

Definition 1.1. Shift Register
A shift register shifts all the values such that a new value is shifted into the first input and the last value is removed.



Definition 1.2. Linear Feedback Shift Register (LFSR)
The output of a standard shift register is manipulated using a linear function and fed back into its input resulting in a endless cycle of a sequence of patterns.

Definition 1.3. Feedback with Carry Shift Register (FCSR)
This is a Linear Feedback Shift Register whose function also includes a carry.

Definition 1.4. Seed
A seed is the initial value or values given to a LFSR or FCSR.

Most of these generators use an initial seed that is not predictable. One example of a possible seed is the digits after the thousandth digit of the current temperature. If one knows the seed, they can calculate what the numbers that result from the generator are making pseudo-random not truly. On the other hand, these numbers are widely used throughout computer science because a logic based computer can't generate something truly random. Imagine a FCSR as a linear function on the digits of a n-adic integer and a carry. This function results in a new digit for the n-adic number and a new carry. On the other hand, a LFSR does not have a carry and just uses the digits of the previous n-adic integer. We call the n-adic number a and the carry z. Each element, a_i of a is $a_i \in S = 0, 1, ...N - 1$. We say a FCSR has a length, r, if a has elements $a_0 a_1 ... a_{r-1}$. To iterate through the FCSR we find a new integer, a_r which like any a_i is $a_i \in S = 0, 1, ...N - 1$.
All in all, a FCSR creates a pseudo-random code based upon an initial seed.
By doing this over and over again, you generate an infinite but eventually periodic sequence of numbers in S, also known as a n-adic number. This can in turn generate a stream of seemingly pseudo-random numbers for use in various cryptography methods.
For example, if we had a 6-ary FCSR with a seed of 425 and a function that just sums the

Date: June 7, 2018.

Simon Rubinstein-Salzedo and the rest of Euler Circle.

last 3 values and takes it $\mod 6$. The first added digit would be $4+2+5=11 \mod 6=5$ and the next would be $2+5+5=12 \mod 6=0$ and the next would be $5+5+0=10 \mod 6=4$ and then $5+0+4=9 \mod 6=3$ then $0+4+3=7 \mod 6=1$ and $4+3+1=8 \mod 6=2$ and $3+1+2=6 \mod 6=0$ and so forth.
All in all,

$$4255043120352...$$

Obviously, this FCSR function doesn't generate truly pseudo-random number because it is so basic. Therefore, it would not take a computer very long to figure out this extremely simple function. But at first glance, these numbers already seem fairly random. Using this technique and more complicated functions using many digits in bizarre combinations, you could generate something that seems pseudo-random.
In cryptography, the only FCSR used is 2-ary because computers are binary.

Definition 1.5. Plain-text
A uncoded message.

Definition 1.6. Cipher
A mechanism to encode plain-text such that it can be transmitted securely.

Definition 1.7. Stream cipher
A cipher in which the digits of the plain-text are combined with a pseudo-random string to encode a message.

The F-FCSR cipher was a stream cipher developed by Thierry Berger, Francois Arnault, and Cedric Lauradoux. In p-adic terms, a F-FCSR computes the binary expansion of a 2-adic number p/q, where p and q are some integers, with q being odd. p is determined by a private key that can quickly decode the message and q is public and anybody can access it. Let assume that $q < 0 < p < |q|$. In this particular cipher, q is a negative prime whose 2-adic expansion is $n+1$ digits long. Additionally, $T = \frac{|q|-1}{2}$ is prime.
It turns out that this common cipher doesn't work because this FCSR does not create nonlinearity. The nonlinearity in this technique comes from the carry bit calculation which should make the result pseudo-random ends up not actually being random instead of being obvious like in our example. A feedback bit is determined based upon the previous carries. If the feedback bit is 0 and the carry is 0 than the carry must be 0 and if the carry is 1 than with probability $\frac{1}{2}$ it will turn to 0. If there are many consecutive 0s for the feedback bit, then the carry bit and therefore and future feedback bit will forever be 0. Similarly, the same is true for if the feedback bit is 1 for long enough.
Using this knowledge, and p-adic numbers, an attack can be devised that can find the key to the cipher very quickly and therefore decode the cipher without knowing the private key in a comparable amount of time to decoding it normally. Unfortunately, I don't have time to present on that right now but I suggest you read "Breaking the F-FCSR-H Stream Cipher in Real Time" by Martin Hell and Thomas Johansson. [1]

References

[1] Martin Hell and Thomas Johansson. *Breaking the F-FCSR-H Stream Cipher in Real Time.* `https://iacr.org/archive/asiacrypt2008/53500563/53500563.pdf`

Euler Circle, Palo Alto, CA 94306
E-mail address: johan.d.s.vonk@gmail.com