

FSR - Feedback Shift Registers

Nusa Zidaric, Mark Aagaard, and Guang Gong

University of Waterloo, Waterloo, ON, Canada N2L 3G1
{nzidaric,maagaard,ggong}@uwaterloo.ca

This presentation describes the GAP package FSR, which implements feedback shift registers and filtering functions. It allows creation, initialization and running of both linear and nonlinear FSRs (LFSRs and NLFSRs), and can compute some of their properties, e.g. the internal state size of any FSR or the period of an LFSR, and evaluate filtering functions, named FILFUNs.

While the LFSR and NLFSR differ only in feedback, a FILFUN is an object of type FSR, whose functionality is defined by a multivariate polynomial, but without feedback, shifting, or storing. The main functionality of all three objects is implemented with three methods: *LoadFSR*, *StepFSR* and *RunFSR*. The latter two can be used as a regular and an external step and run. A stand-alone simple (N)LFSR object is self-contained: it is updated by the computed feedback value (regular step and run). The external *StepFSR* allows arbitrary filters to be added to the feedback of any (N)LFSR or it can be used e.g. to mask the output of the filtering function. The external step and run are implemented because of their common use in cryptography, thus the FSRs can be used directly as building blocks of many ciphers.

Feedback shift registers (FSR) play an important role in stream cipher design. A milestone in stream cipher design is the eSTREAM project [1], launched in 2004. All 3 hardware portfolio ciphers, Grain, MICKEY and Trivium, as well as the software portfolio cipher Sosemanuk, use FSRs. The stream cipher ACORN [2], a round 3 CAESAR candidate [3], is based on 6 LFSRs. Last but not least, the two stream ciphers used for encryption and integrity of communications in mobile networks, Snow3G and ZUC[4,5], both use LFSRs. Another application area for LFSRs are the cyclic redundancy codes (CRC) used in many communication and data storage devices for error-detection and correction. Less noticeable is the use of LFSRs in algorithms for finite field arithmetic: e.g. a serial circuit implementing multiplication by x followed by reduction modulo the field defining polynomial $f(x)$ can be implemented as a LFSR with the feedback $f(x)$.

The remainder of the FSR package consists of helper functions (for example to compute the degree of the feedback polynomial or format the output w.r.t. the selected basis), output functions that can store entire runs to files (*.txt or *.tex) and drawing functions that can automatically generate tikz code. More detail can be found at <https://github.com/nzidaric/gap-fsr> and its manual.

References

1. Robshaw, M.: “The eSTREAM Project”, New Stream Cipher Designs - The eSTREAM Finalists, Springer-Verlag, Berlin Heidelberg, 2008
2. Wu, H.: “ACORN: A Lightweight Authenticated Cipher (v1)”, <http://competitions.cr.yp.to/round1/acornv1.pdf>
3. <https://competitions.cr.yp.to/caesar.html>
4. ETSI/SAGE Specification version 1.1: “Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 2: SNOW 3G Specification”, Sept. 2006
5. ETSI/SAGE Specification version 1.6: “Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification”, June 2011