Entropy of the Internal State of an FCSR in Galois Representation

Andrea Röck
INRIA Paris - Rocquencourt, France

Fast Software Encryption
Lausanne, February 12, 2008

Outline

- FCSR
- Entropy after one Iteration
- Final Entropy
- Lower Bound
- Conclusion

Part 1

FCSR

Context

- **Feedback with Carry Shift Registers (FCSRs):**
  - Similar to LFSRs but instead of XORs they use additions with carry.
  - Introduced by **[Goresky Klapper 93]**, **[Marsaglia Zamand 91]** and **[Couture L'Ecuyer 94]**.
- Binary FCSRs in **Galois** architecture **[Goresky Klapper 02]**.
- Used in the eSTREAM candidate F-FCSR **[Arnault *et al.* 05]**.
- **Entropy** of inner state when **all values** for the initial states are allowed, *e.g* first version of F-FSCR-8.

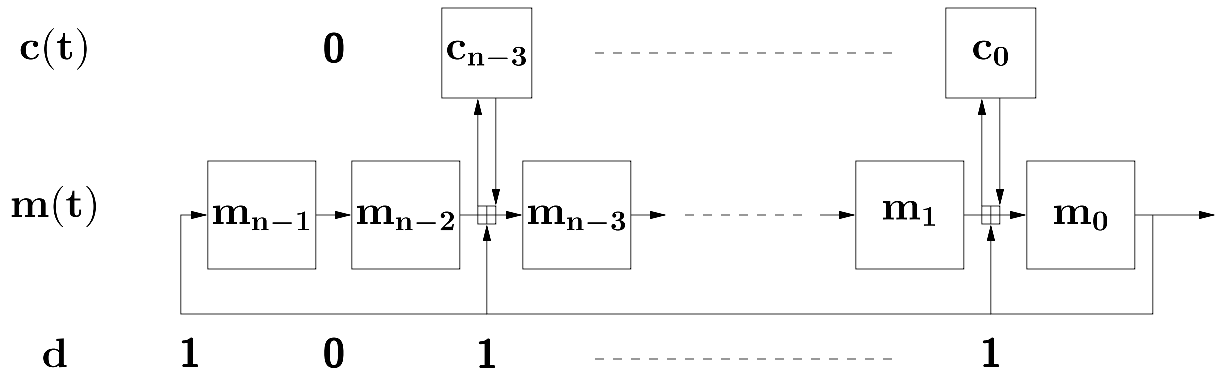
FCSRs

- The output of an FCSR is the 2-adic expansion of

$$\frac{p}{q} \leq 0.$$

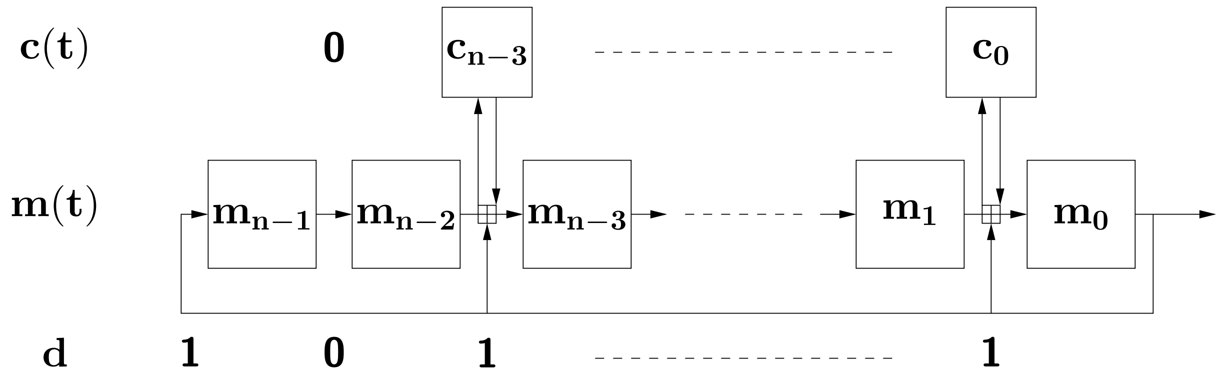
- The output of an FCSR has the maximal period of $|q|-1$ if and only if 2 has order $|q|-1$ modulo q.

FCSR in Galois architecture (1)



- n: Size of main register.
- $2^n > d \geq 2^{n-1}$: Integer which determines the feedback positions. Carry bit if $d_i = 1$.
- $(m(t), c(t))$: State at time t with
  - $m(t) = \sum_{i=0}^{n-1} m_i(t)2^i$: 2-adic expansion of the main register.
  - $c(t) = \sum_{i=0}^{n-1} c_i(t)2^i$: 2-adic expansion of the carry register, where $c_i(t) = 0$ for $d_i = 0$.
- **In our case:** $q = 1 - 2d < 0$ and $p = m(0) + 2c(0) \leq |q|$.

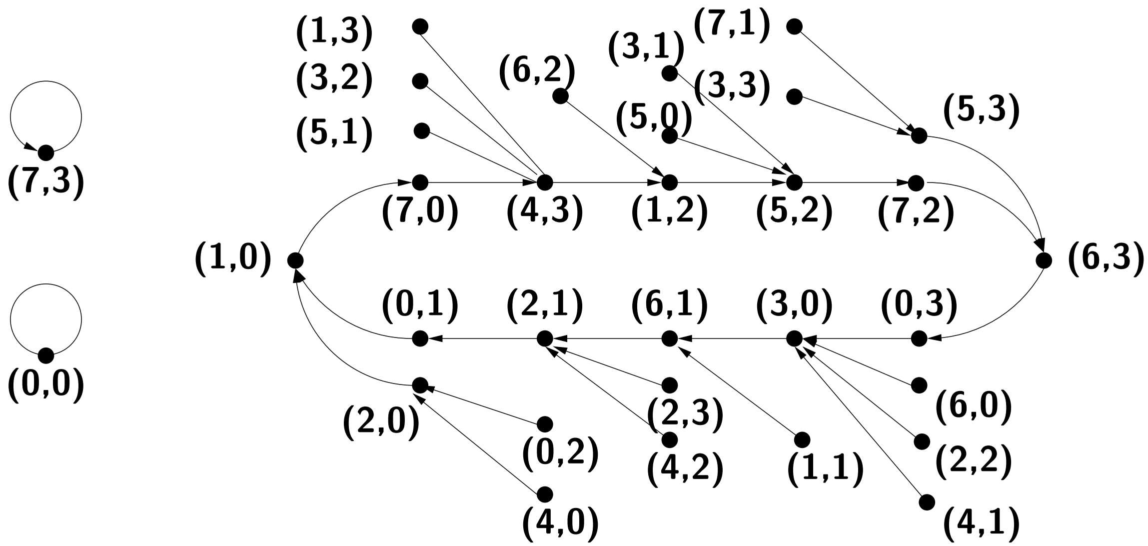
FCSR in Galois architecture (2)



- **Update function:**

$$
\begin{aligned}
m_{n-1}(t+1) &= m_0(t), \\
d_i = 1 : m_i(t+1) &= (m_0(t) + c_i(t) + m_{i+1}(t)) \quad \mod 2, \\
c_i(t+1) &= (m_0(t) + c_i(t) + m_{i+1}(t)) \quad \div 2, \\
d_i = 0 : m_i(t+1) &= m_{i+1}(t).
\end{aligned}
$$

Entropy



- We have
  - n bits in the main register and
  - $\ell = HammingWeight(d) - 1$ carry bits.
- Initial Entropy: $n + \ell$ bits.
- Entropy after one iteration: $H(1)$.
- Final Entropy: H^f.

Part 2

Entropy after one Iteration

Idea

- **Initial entropy:** $n + \ell$.
- **Question:** Entropy loss after one iteration?
- **Method:**
  - Counting the number of $(m(0), c(0))$'s which produce the same $(m(1), c(1))$.
  - Using the equations of the update function.
  - Only possible if there are positions i such that $d_i = 1$ and $m_{i+1}(0) + c_i(0) = 1$.
- **Entropy after one iteration:**

$$H(1) = \sum_{j=0}^{\ell} 2^{n-j} \binom{\ell}{j} \frac{2^j}{2^{n+\ell}} \log_2 \left(\frac{2^{n+\ell}}{2^j} \right) = n + \frac{\ell}{2}.$$

Part 3

Final Entropy

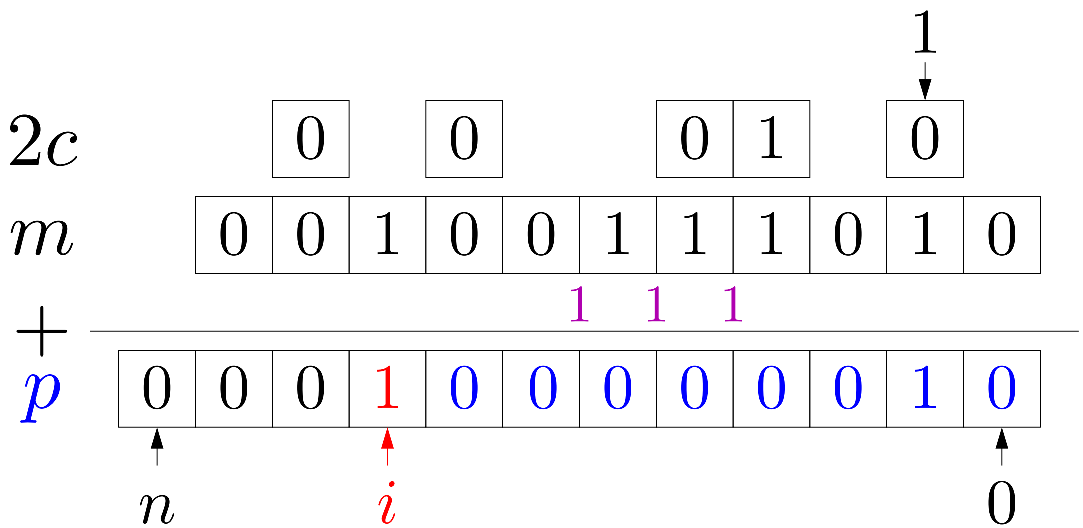
Final Entropy

- **Goal:** Entropy when we reached the cycle.
- **Proposition [Arnault Berger Minier 08]:** Two states (m, c) and (m', c') are equivalent, *i.e.* $m + 2c = m' + 2c' = p$, if and only if they eventually converge to the same state after the same number of iterations.
- **Idea:** How many (m, c)'s create the same $p = m + 2c$?
- **Probability:** $\frac{v(p)}{2^{n+\ell}}$, where $v(p) = \#\{(m, c) | m + 2c = p\}$ for all $0 \leq p \leq |q|$.
- **Final Entropy:**

$$H^f = \sum_{p=0}^{|q|} \frac{v(p)}{2^{n+\ell}} \log_2 \left(\frac{2^{n+\ell}}{v(p)} \right)$$

Algorithm (1)

- **Method:** Get $v(p)$ by looking at bit per bit addition of m and $2c$.



Algorithm (2)

- **4 different Cases:** $i = \lfloor \log_2(p) \rfloor$.
  - **Case 1**: $1 < i < n$ and $d_{i-1} = 0$.
  - **Case 2**: $1 < i < n$ and $d_{i-1} = 1$.
  - **Case 3**: $i = n$ and $2^n \leq p \leq |q|$.
  - **Case 4**: $0 \leq p \leq 1$ ("$i = 0$").
- **For each case:**
  - Which p's are in this case.
  - What is their value of $\frac{v(p)}{2^{n+\ell}} \log_2 \left(\frac{2^{n+\ell}}{v(p)} \right)$.
- **Complexity:** Works in $O\left(n^2\right)$ if $S_1(k) = \sum_{x=2^{k-1}+1}^{2^k} x \log_2(x)$ and $S_2(k) = \sum_{x=1}^{2^k-1} x \log_2(x)$ are known for $k \leq \ell$.

Approximation

- $S_1(k) = \sum_{x=2^{k-1}+1}^{2^k} x \log_2(x)$ and $S_2(k) = \sum_{x=1}^{2^k-1} x \log_2(x)$ can be approximated by using

$$\frac{1}{2}\Big(x \log_2(x) + (x+1)\log_2(x+1)\Big) \approx \int_x^{x+1} y \log_2(y)\, dy$$

  for large x.

- Result for some arbitrary values of d.

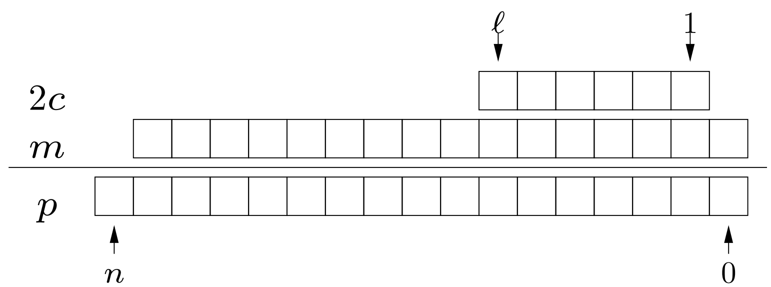
n	d	ℓ	H^f	lb H^f	ub H^f	lb H^f , $k > 5$	ub H^f, $k > 5$
8	0xAE	4	8.3039849	8.283642	8.3146356	8.3039849	8.3039849
16	0xA45E	7	16.270332	16.237686	16.287598	16.270332	16.270332
24	0xA59B4E	12	24.273305	24.241851	24.289814	24.273304	24.273305
32	0xA54B7C5E	17		32.241192	32.289476	32.272834	32.272834

Part 4

Lower Bound

Lower Bound of the Final Entropy

- Proof that final entropy is $\geq n$ for all FCSRs in Galois architecture by using previous algorithm.
- **Induction Base:**
  An FCSR has a final entropy larger than n if the feedback positions are all grouped together at the least significant position.



- **Induction Step:**
  If we move a feedback position one position to the left, the final entropy increases.

Part 5

Conclusion

Conclusion

- After one iteration, we loose already $\ell/2$ bits of entropy.
- We have presented an algorithm which computes the final state entropy of an Galois FCSR.
- The algorithm works in $O(n^2)$ if the values of the sums $\sum_{x=2^{k-1}+1}^{2^k} x \log_2(x)$ and $\sum_{x=1}^{2^k-1} x \log_2(x)$ are known. Otherwise we need $O(2^\ell)$ steps to compute these sums.
- The approximation of the sum works very well for large k.
- The final entropy is larger than n bits.