IEOR SEMINAR SERIES

Cryptanalysis: Fast Correlation Attacks on LFSR-based Stream Ciphers

presented by

Goutam Sen
Research Scholar
IITB Monash Research Academy.

Agenda:

- Introduction to Stream Ciphers
- Linear Feedback Shift Register(LFSR)
- Cryptanalysis of LFSR-based Stream Ciphers.
- Statistical Model
- Exponential-Time Correlation Attack
- Polynomial-Time Correlation Attack
- Computational Complexity and Limits of Attack
- References

A Cryptosystem or Cipher

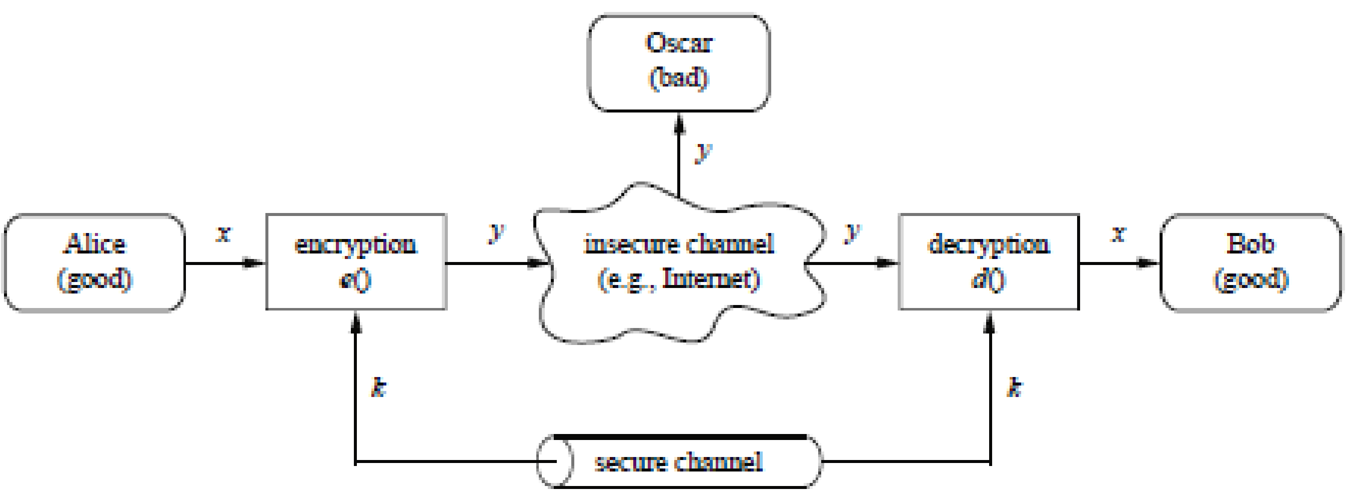
- 5-tuple Cryptosystem: $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$

$\mathcal{P}$ is a finite set of possible plaintexts;

$\mathcal{C}$ is finite set of possible ciphertexts;

$\mathcal{K}$ is the keyspace, finite set of possible keys;

For each $K \in \mathcal{K}$, there is an encryption rule $e_K \in \mathcal{E}$ and a corresponding decryption rule $d_K \in \mathcal{D}$. Each $e_K : \mathcal{P} \rightarrow \mathcal{C}$ and d_k: $\mathcal{C} \rightarrow \mathcal{P}$ are functions such that $d_K(e_K(x)) = x$ for every plaintext element $x \in \mathcal{P}$.



Block Ciphers vs. Stream Ciphers

Block Ciphers:

$x = x_1x_2 \ldots x_n$ for some integer $n \geq 1$ and $x_i \epsilon \mathscr{P}$

K: predetermined key(might be different for $\mathscr{E}$ and $\mathscr{D}$).

$y_i = e_K(x_i)$, where $e_K()$ is an injective function(one-to-one).

$y = y_1y_2 \ldots y_n$

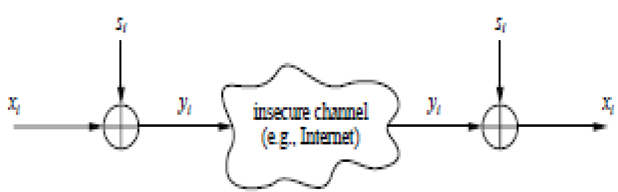
Encrypted with the same key $K \epsilon \mathscr{K}$

Stream Ciphers:

Keystream $K = k_1k_2k_3 \ldots$

Cipher $y = e_{k1}(x_1)e_{k2}(x_2)e_{k3}(x_3) \ldots$

- $\mathscr{P} = \mathscr{C} = \mathbf{Z_2}$
- $e_k(x) = (x+k)\%2$
- $d_k(y) = (y+k)\%2$
- Hardware implementation: XOR gate



Random Number Generators:

- True Random Number Generator (TRNG)
- Pseudo-Random Number Generator (PRNG)

  Example: Linear Congruential Generator(LCG)

$$s_0 = seed;$$
$$s_{i+1} = as_i + b \bmod m; \text{ for } i = 0,1,2\ldots$$

  - ➢ chi-square test for statistical randomness
  - ➢ not truly random, having periodicity.

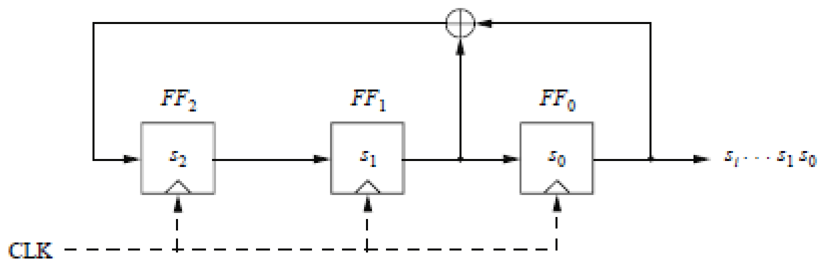
- Cryptographically Secure Pseudo-Random Number Generator (CSPRNG)

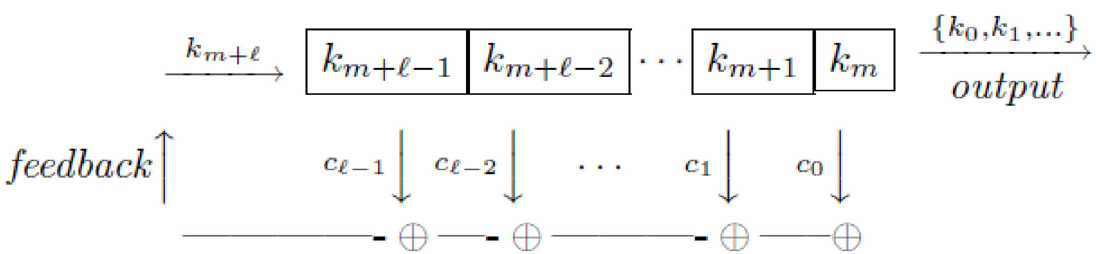
➢ statistical properties of truly random sequence

➢ Given n output bits $s_i, s_{i+1}, \ldots, s_{i+n-1}$
No polynomial time algorithm that can predict the next bit s_{n+1} with better than *50%* chance of success.

➢ Computationally infeasible to predict $s_{i+n}, s_{i+n+1}, \ldots$ *and also* $s_{i-1}, s_{i-2}, \ldots$

Linear Feedback Shift Register(LFSR)





$$k_{m+\ell} = \sum_{j=0}^{\ell-1} c_j k_{m+j}.$$ linear feedback

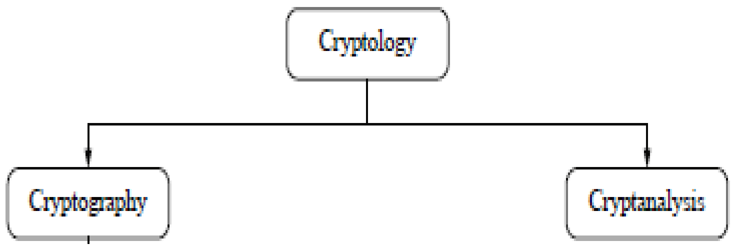
Properties of LFSR

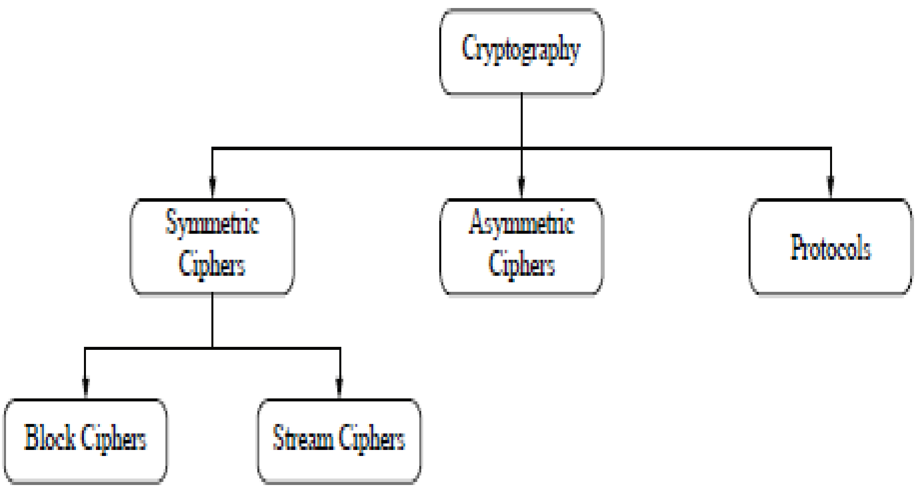
- Periodicity: 2^l-1 for maximum-length LFSR.
- Tap polynomial:

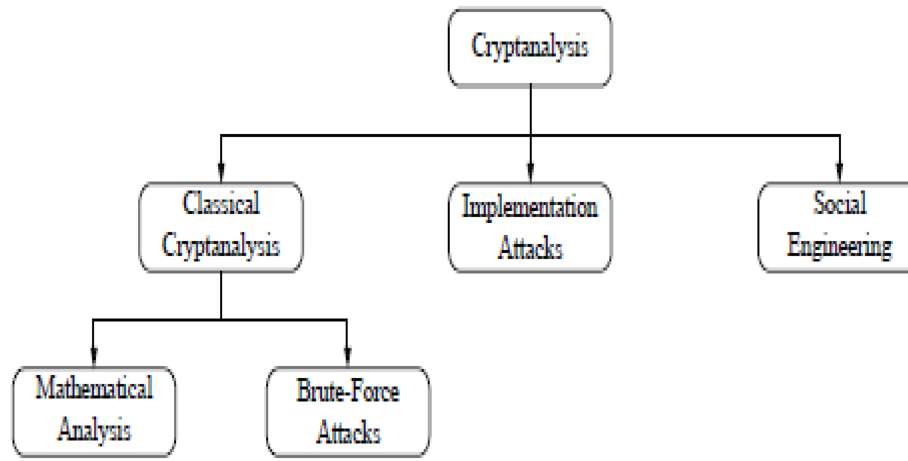
$$t(x) = x^\ell + c_{\ell-1}x^{\ell-1} + c_{\ell-2}x^{\ell-2} + \cdots + c_1 x + c_0$$

- Primitive polynomial(maximum-length LFSR)
  - ➢ $t(x)$ has no proper non-trivial factors
  - ➢ does not divide x^d+1 *for* $d<2^l-1$
- Linear complexity of a binary sequence $k = \{k_j\}$ is the length of the shortest LFSR that generates k.
- Berlekamp Massey Algorithm suggests that for a binary sequence $k = \{k_j\}$ having linear complexity L, there exists a unique LFSR of length L iff $L \leq n/2$

Cryptology, Cryptography and Cryptanalysis







Key length	Security estimation
56–64 bits	short term: a few hours or days
112–128 bits	long term: several decades in the absence of quantum computers
256 bits	long term: several decades, even with quantum computers that run the currently known quantum computing algorithms

Cryptanalysis

- Mathematical analysis to defeat cryptographic methods.
- Kerckhoff's Principle:

  To obtain security while assuming that Oscar knows the cryptosystem (i.e. encryption and decryption algorithms).
- Types of Attack:
  - Ciphertext only attack (knowledge of y)
  - Known plaintext attack (knowledge of x and y)
  - Chosen plaintext attack (temporary access to cryptosystem $x \rightarrow y$)
  - Chosen ciphertext attack (temporary access to decryption machinery $y \rightarrow x$)
- Objective: To determine the "key" so that 'target' ciphertext can be decrypted.

Cryptanalysis of LFSR-based stream ciphers

- $y_i = (x_i + k_i)\%2$
- $(k_1, k_2, \ldots, k_m)$ initial tuple.
- Linear recurrence:

$$z_{m+i} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2$$

- Known-plaintext attack:

  $x = x_1 x_2 \ldots x_n$

  $y = y_1 y_2 \ldots y_n$

  $k_i = (x_i + y_i)\%2$
- To reproduce the entire keystream, we require $n \geq 2m$, assuming m, the length of the LFSR, is known.
- What remains to compute is the tap sequence $c_0, c_1, c_2, \ldots, c_{m-1}$

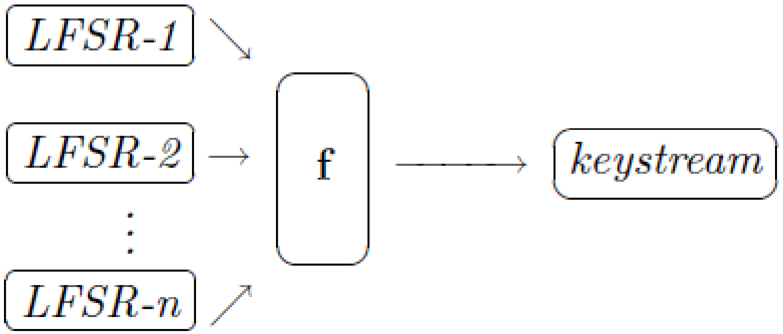
Matrix Form

$$(z_{m+1}, z_{m+2}, \ldots, z_{2m}) = (c_0, c_1, \ldots, c_{m-1}) \begin{pmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \vdots & \vdots & & \vdots \\ z_m & z_{m+1} & \cdots & z_{2m-1} \end{pmatrix}$$

If the coefficient matrix has an inverse (modulo 2), we obtain the solution

$$(c_0, c_1, \ldots, c_{m-1}) = (z_{m+1}, z_{m+2}, \ldots, z_{2m}) \begin{pmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \vdots & \vdots & & \vdots \\ z_m & z_{m+1} & \cdots & z_{2m-1} \end{pmatrix}^{-1}$$

Nonlinear Combination Generator



$$f(x_1, x_2, x_3, x_4, x_5) = 1 \oplus x_2 \oplus x_3 \oplus x_4 \cdot x_5 \oplus x_1 \cdot x_2 \cdot x_3 \cdot x_5.$$

- Siegenthaler shows that if the keystream is correlated to (at least) one of the LFSR sequences, the correlation attack against this individual LFSR significantly reduces a brute-force attack.
- Divide and Conquer:

  Attempt first to determine initial states of subset of LFSRs, in order to reduce complexity of search for right key.

Algebraic and Statistical Foundation

- Assume that ***N*** digits of the output sequence ***z*** are given.
- Correlation probability $p>0.5$ to an LFSR sequence **a**.

$$p = \text{Prob}(z_n = a_n) > 0.5.$$

- The LFSR in question has few feedback tabs, say *t*. (This is desired for the ease of hardware).
- Further assume that feedback connection is known(although not an essential restriction).
- LFSR sequence **a** is given by linear relation(for LFSR-length *k*)

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}.$$

$$\sum_{\{i:0 \le i \le k, c_i \ne 0\}} a_{n-i} = 0.$$

- Feedback polynomial: $c(X) = c_0 + c_1 X + c_2 X^2 + \cdots + c_k X^k$ (with $c_0 = 1$)

Algebraic and Statistical foundations

- Every polynomial multiple of $c(X)$ defines a linear relation for **a**.
- In particular, $\boldsymbol{c(X)^j = c(X^j)}$ for exponents $j=2^i$
- All having same number t number of feedback taps.
- Suppose a_n is fixed.
- Linear relations obtained by shifting and iterated squaring:

$$\begin{aligned} L_1 &= a + b_1 = 0, \\ L_2 &= a + b_2 = 0, \\ &\vdots \\ L_m &= a + b_m = 0, \end{aligned}$$

  where $a=a_n$ and each b_i, $i=1,\ldots,m$ is a sum of exactly t different terms of the LFSR sequence **a**.
- We substitute the digits of $\boldsymbol{z}$ at same index positions:

$$L_i = z + y_i, \qquad i = 1, \ldots, m,$$

Statistical Model

- Introducing a set of binary random variables $A = \{a, b_{11}, b_{12}, \ldots, b_{1t}, b_{21}, b_{22}, \ldots, b_{2t}, \ldots, b_{m1}, b_{m2}, \ldots, b_{mt}\}$

$$\begin{aligned} a + b_{11} + b_{12} + \cdots + b_{1t} &= 0, \\ a + b_{21} + b_{22} + \cdots + b_{2t} &= 0, \\ &\vdots \\ a + b_{m1} + b_{m2} + \cdots + b_{mt} &= 0. \end{aligned}$$

- Similarly introducing a set of binary random variables $Z = \{z, y_{11}, y_{12}, \ldots, y_{1t}, y_{21}, y_{22}, \ldots, y_{2t}, \ldots, y_{m1}, y_{m2}, \ldots, y_{mt}\}$

$$\mathrm{Prob}(z = a) = p \quad \text{and} \quad \mathrm{Prob}(y_{ij} = b_{ij}) = p.$$

$$b_i = b_{i1} + b_{i2} + \cdots + b_{it}$$

$$y_i = y_{i1} + y_{i2} + \cdots + y_{it}$$

$$L_i = z + y_i.$$

$$s = \mathrm{Prob}(y_i = b_i),$$

$$s(p, t) = ps(p, t-1) + (1-p)(1 - s(p, t-1)),$$

$$s(p, 1) = p.$$

Statistical Model(contd.)

- Consider random variables $L_1, L_2, \ldots, L_m$.
- The probability that the outcome of these random variable vanishes for a given set of exactly h indices is given by

$$ps^h(1-s)^{m-h} + (1-p)(1-s)^h s^{m-h}.$$

- For simplicity, assume that $L_1=0, L_2=0, \ldots, L_h=0$ and $L_{h+1}=1, L_{h+2}=1, \ldots, L_m=1$.

$$P(z=a \mid L_1=\cdots=L_h=0, L_{h+1}=\cdots=L_m=1) = \frac{ps^h(1-s)^{m-h}}{ps^h(1-s)^{m-h}+(1-p)(1-s)^h s^{m-h}},$$

$$P(z\neq a \mid L_1=\cdots=L_h=0, L_{h+1}=\cdots=L_m=1) = \frac{(1-p)(1-s)^h s^{m-h}}{ps^h(1-s)^{m-h}+(1-p)(1-s)^h s^{m-h}}.$$

- z corresponds to the fixed digit z_n, and a to the fixed digit a_n we wish to determine.

$p*$ as a function of h

p^* as function of number h
of relations satisfied ($p=0.75$)

h	p*
0	0.00011
1	0.00030
2	0.00085
3	0.00235
4	0.00649
5	0.01782
6	0.04797
7	0.12278
8	0.27995
9	0.51923
10	0.75000
11	0.89286
12	0.95859
13	0.98469
14	0.99443
15	0.99799
16	0.99927
17	0.99974
18	0.99991
19	0.99997
20	0.99999

$$m = m(N, k, t) \approx \log\left(\frac{N}{2k}\right)(t+1).$$

An Efficient Exponential-Time Attack

- To select *k* digits of ***z*** with the highest probability *p**
- LFSR sequence **a** can be constructed out of its any *k* digits solving linear equations for the initial state.
- The probability ***Q(p,m,h)*** that a fixed digit *z* satisfies at least *h* of *m* relations:

$$Q(p,m,h) = \sum_{i=h}^{m} \binom{m}{i} \left(p s^i (1-s)^{m-i} + (1-p)(1-s)^i s^{m-i} \right)$$

- The probability ***R(p,m,h)*** that *z=a* and at least *h* of *m* relations hold:

$$R(p,m,h) = \sum_{i=h}^{m} \binom{m}{i} p s^i (1-s)^{m-i}.$$

- So, the prob. for *z=a*, given that at least *h* of *m* relations hold is the quotient:

$$T(p,m,h) = R(p,m,h)/Q(p,m,h).$$

- ***Q(p,m,h).N*** are expected to satisfy at least *h* relations and these digits have probability ***T(p,m,h)*** of being correct.
- ***T(p,m,h)*** increases with *h*. So maximize h with ***Q(p.m.h)≥k***

Algorithm A

- *Step1*. Determine *m*.
- *Step2*. Find the maximum value of *h* such that $Q(p.m.h) \geq k$.
- *Step3*. Search for digits of ***z*** satisfying at least *h* relations and use these digits as a reference guess I_0 *of* **a** at the corresponding index positions.
- *Step4*. Find the correct guess by testing modifications of I_0 with Hamming distance 0,1,2,… by correlation of the corresponding LFSR sequence with the sequence ***z***.

- Observation: digits in the middle part of ***z*** satisfy more relations that the digits near the boundaries. This leads to slight modification of step3 as

  Step3': Compute new probability p^* for the given digits of ***z*** and choose *k* digits having highest probability p^*.
- Average number of erroneous digits is computed as $(1-T(p,m,h)).k$. Under favorable conditions(e.g., <<1), step4 is not necessary.

Computational Complexity of Algorithm A

- Computation time for Step 1-3 is negligible.
- Only estimate average number of trials in step4.
- Suppose exactly *r* among the digits found in step3 are incorrect.
- Max number of trials in step4 is

$$A(k, r) = \sum_{i=0}^{r} \binom{k}{i}.$$

- A well-known estimate using binary entropy function

$$H(0) = H(1) = 0,$$
$$H(x) = -x \log x - (1 - x) \log(1 - x) \qquad (0 < x < 1).$$

- Then

$$A(k, r) = \sum_{i=0}^{r} \binom{k}{i} \leq 2^{H(\theta)k}$$

with $\theta = r/k$.

- Algorithm A has computational complexity $O(2^{ck})$, where $c = H(r/k)$, $0 \leq c \leq 1$

A Polynomial-Time Attack

- We do not search for correct digits here. Instead, we assign new probability *p** to each digit of ***z*** iteratively and under some favorable conditions, complement all digits to get maximum correction effect.
- The probability $U(p,m,h)$ that at most h of m relations are satisfied:

$$U(p,m,h) = \sum_{i=0}^{h} \binom{m}{i} (ps^i(1-s)^{m-i} + (1-p)(1-s)^i s^{m-i}).$$

- The probability $V(p,m,h)$ that $z=a$ and at most h of m relations are satisfied:

$$V(p,m,h) = \sum_{i=0}^{h} \binom{m}{i} ps^i(1-s)^{m-i}$$

- The probability $W(p,m,h)$ that $z \neq a$ and at most h of m relations are satisfied:

$$W(p,m,h) = \sum_{i=0}^{h} \binom{m}{i} (1-p)(1-s)^i s^{m-i}.$$

- $U(p,m,h).N$ is the expected number of digits of z which satisfy at most h relations.
- Relative increase in correct digits after complementation:

$$I(p,m,h) = W(p,m,h) - V(p,m,h).$$

- For given p and m, choose $h=h_{max}$ so as to maximize $I(p,m,h)$.

- Taking $p*$ into account, we replace h_{max} by a corresponding probability threshold on $p*$

$$p_{\text{thr}} = \tfrac{1}{2}(p^*(p, m, h_{\max}) + p^*(p, m, h_{\max} + 1))$$

- Expected number of digits with $p*$ below p_{thr} is:

$$N_{\text{thr}} = U(p, m, h_{\max}) \cdot N.$$

- Generalized formula to compute $s(p,t)$:

$$s(p_1, \ldots, p_t, t) = p_t s(p_1, \ldots, p_{t-1}, t-1) + (1 - p_t)(1 - s(p_1, \ldots, p_{t-1}, t-1)),$$

$$s(p_1, 1) = p_1.$$

Algorithm B

- *Step1*: Determine m.
- *Step2*: Find the value of $h=h_{max}$ such that $I(p,m,h)$ is maximized. Compute p_{thr} and N_{thr}.
- *Step*3. Initialize the iteration counter $i=0$.
- *Step*4. For every digit of $\boldsymbol{z}$ compute the new probability p^* with respect to the individual number of relations satisfied. Determine the number N_w of digits with $p^*<p_{thr}$.
- *Step*5. if $N_w<N_{thr}$ or $i<\alpha$ increment i and go to *step*4.
- *Step*6. Complement those digits of $\boldsymbol{z}$ with $p^*<p_{thr}$ and reset the probability of each digit to the original value of p.
- *Step*7. If there are digits not satisfying linear recurrence, go to *step*3.
- *Step*8. Terminate with $\mathbf{a}=\boldsymbol{z}$.

Computational Complexity and Limits of Attack:

- $m=m(t,d)$, $d=N/k$.
- $h_{max}=h_{max}(p,m)$
- $I_{max}=I_{max}(p,t,d)$
- The expected number of digits corrected in one iteration $N_c=I_{max}(p,t,d).N$
- $N_c = F(p,t,d).k$ where
  $F(p,t,d)=I_{max}(p,t,d).d$
- If $F(p,t,d)\leq 0$, no correction effect. Attack will fail.
- For $F(p,t,d)\geq 0.5$, successful attack.

p with $F(p,t,d)=0.5$

	t								
d	2	4	6	8	10	12	14	16	18
10	0.761	0.880	0.980	0.980	0.980	0.980	0.980	0.980	0.980
10^2	0.595	0.754	0.824	0.863	0.889	0.905	0.917	0.926	0.934
10^3	0.553	0.708	0.787	0.832	0.861	0.882	0.897	0.908	0.918
10^4	0.533	0.679	0.763	0.812	0.844	0.867	0.883	0.896	0.906
10^5	0.525	0.663	0.748	0.800	0.833	0.857	0.875	0.889	0.900
10^6	0.519	0.650	0.737	0.789	0.825	0.849	0.868	0.883	0.894
10^7	0.515	0.641	0.727	0.781	0.817	0.843	0.862	0.877	0.890
10^8	0.514	0.634	0.720	0.774	0.812	0.838	0.858	0.874	0.886
10^9	0.512	0.628	0.714	0.770	0.807	0.833	0.854	0.870	0.882
10^{10}	0.510	0.621	0.709	0.764	0.802	0.830	0.850	0.866	0.879

An Example

- Consider the following situation

  $p=0.75$

  $t=4$

  $d=100$

  $N=10,000$

  $k=100$
- $F(p,t,d)=0.392$
- Parameters of Algorithm B:

  $p_{thr}=0.524$

  $N_{thr}=448$

	Number of digits with $p^* < p_{thr}$	Number of wrong digits with $p^* < p_{thr}$	Decrease of wrong digits	Number of wrong digits after correction
Round 1				
Iteration 1	430	246	62	2500
Iteration 2	615	416	217	2500
Correction ($615 > N_{thr}$)	0	0	0	2283
Round 2				
Iteration 1	70	44	18	2283
Iteration 2	314	254	194	2283
Iteration 3	921	743	565	2283
Correction	0	0	0	1718
Round 3				
Iteration 1	49	48	47	1718
Iteration 2	654	643	623	1718
Correction	0	0	0	1086
Round 4				
Iteration 1	110	110	110	1086
Iteration 2	712	708	704	1086
Correction	0	0	0	382
Round 5				
Iteration 1	86	86	86	382
Iteration 2	342	342	342	382
Iteration 3	382	382	382	382
Correction	0	0	0	0

Complexity and Limits of Attack:

- Algorithm B grows linearly with LFSR length k i.e., is of order $O(k)$.
- $F(p,t,d)<0.5$ has led to successful attack. Same is reported even for $F(p,t,d)=0.1$
- Definite barrier with $F(p,t,d)\leq 0$

p with $F(p,t,d)=0$

	t								
d	2	4	6	8	10	12	14	16	18
10	0.584	0.739	0.804	0.841	0.864	0.881	0.894	0.904	0.912
10^2	0.533	0.673	0.750	0.796	0.827	0.849	0.865	0.878	0.890
10^3	0.521	0.648	0.727	0.776	0.809	0.833	0.852	0.866	0.878
10^4	0.514	0.629	0.709	0.760	0.795	0.821	0.841	0.856	0.869
10^5	0.511	0.620	0.699	0.752	0.787	0.815	0.834	0.850	0.863
10^6	0.509	0.612	0.692	0.745	0.782	0.809	0.830	0.846	0.860
10^7	0.508	0.605	0.684	0.738	0.775	0.803	0.825	0.842	0.855
10^8	0.507	0.601	0.680	0.733	0.771	0.800	0.821	0.838	0.852
10^9	0.506	0.597	0.676	0.729	0.768	0.797	0.818	0.836	0.850
10^{10}	0.505	0.592	0.671	0.725	0.764	0.793	0.815	0.832	0.847

Suggestion:

- Any correlation to an LFSR with less than 10 taps should be avoided.

References:

- Christof Paar and Jan Pelzl, *Understanding Cryptography*, Springer, 2010
- Douglas R. Stinson, *Cryptography Theory and Practice*, 3rd ed., Chapman and Hall/CRC, Taylor & Francis group, 2006
- Mark Stamp and Richard M. Low, *Applied Cryptanalysis: Breaking Ciphers in the Real World*, John Wiley and Sons, Inc., publication, Wiley-Interscience, 2007
- Nigel Smart, *Cryptography: An Introduction*, 3rd Ed., University of Bristol.
- Richard A. Mollin, *An Introduction to Cryptography*, 2nd ed.,Chapman and Hall/CRC, Taylor & Francis group, 2007.
- Willi Meier and Othmar Staffelbach, Fast Correlation Attacks on Certain Stream Ciphers, *Journal of Cryptology*(1989) 1:159-176.
- T. Siegenthaler, Decrypting a class of stream ciphers using ciphertext only, *IEEE Trans. Comput.*,34, 81-85, 1985.
- S. Palit, B. Roy and A. De, "A Fast Correlation Attack for LFSR-Based Stream Ciphers," *ACNS 2003, Lecture Notes in Computer Science*, vol. 2843, pp. 331-342, 2003

Acknowledgements:

- Dr. Sarbani Palit, Professor, Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Calcutta.

Thank You