Implementation of a Brute Force Attack on the A5/1 Keystream Generator in a GPU-based Volunteer Computing Project

Vadim Bulavintsev, Alexander Semenov and Oleg Zaikin

Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk, Russia

Stream ciphers

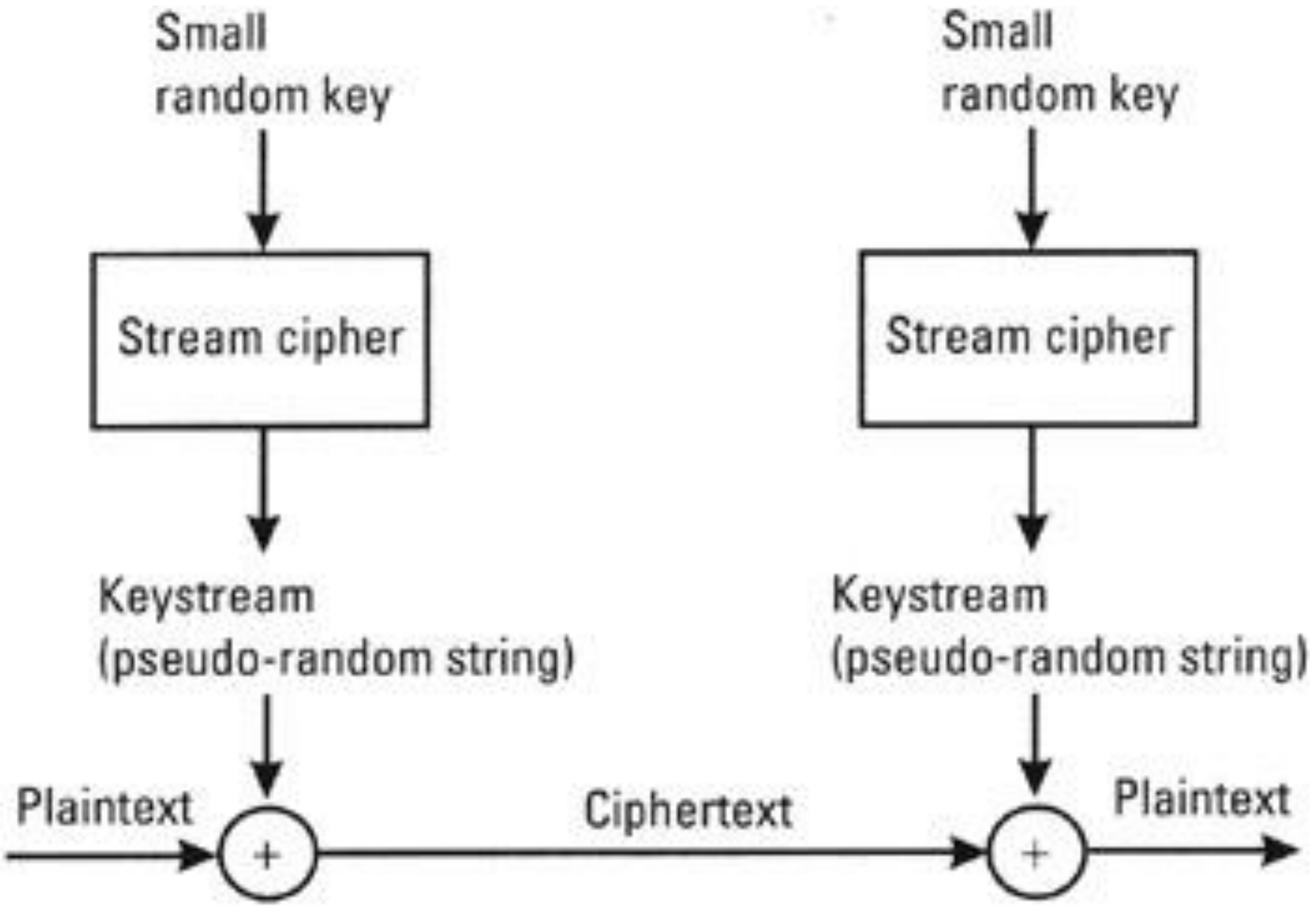
A **stream cipher** is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (**keystream**).

In a stream cipher, each plaintext digit is encrypted one at a time with the corresponding digit of the keystream, to give a digit of the ciphertext stream.

Good stream ciphers must be resistant and fast.

Stream ciphers are used for online ciphering of data. For example, in GSM and Skype.

Stream ciphers



Keystrem generator

Keystream generator is main part of each stream cipher.
Keystream generator – *the discrete function.*

$$f = \{f_n\}_{n\in N},\ f_n : \{0,1\}^n \rightarrow \{0,1\}^*$$

$$\begin{tikzcd} {} \arrow[r, "x = (x_1,\ldots,x_n)"] & \boxed{f_n} \arrow[r, "y = (y_1,\ldots,y_m)"] & {} \end{tikzcd}$$

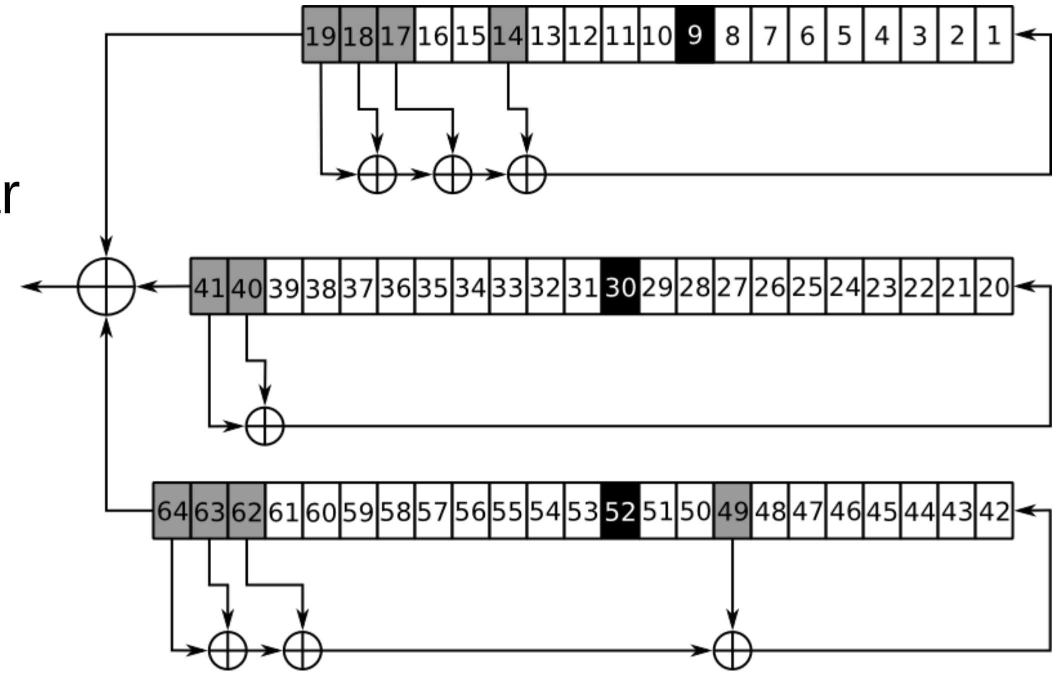
$x = (x_1, \ldots, x_n)$ – initial state.
y $= (y, \ldots, y_m)$ – keystream.

Given *x*, one can effectively produce *y*. In opposite, given *y*, it is very hard to restore x (inversion problem).

Cryptanalysis problem: given known algorithm f and a keystream fragment, one need to find an initial state.

A5/1 generator

- A5/1 is used in GSM
- It consists of 3 linear feedback shift registers (LFSRs) (19, 22 and 23 cells).
- Initial value is 64 bits.



At each tick, output of the generator is 2-modulo summation of all LFSRs outputs.

Nonlinearity of the generator is achieved by irregular LFSRs clocking. At each time 2 or 3 LFSRs are shifted.

A5/1 cryptanalysis

- A5/1 was developed in France in 1980-s.
- 1994 год – Anderson attack.
  Anderson, Ross A5 - The GSM Encryption Algorithm. 1994
- 2009 – Rainbow tables for A5/1
- 2012 – 10 cryptanalysis problems were solved in SAT@home (2 weeks for 1 problem on average).
- **2016 – GPU-implementation of the Anderson attack.**

Anderson attack

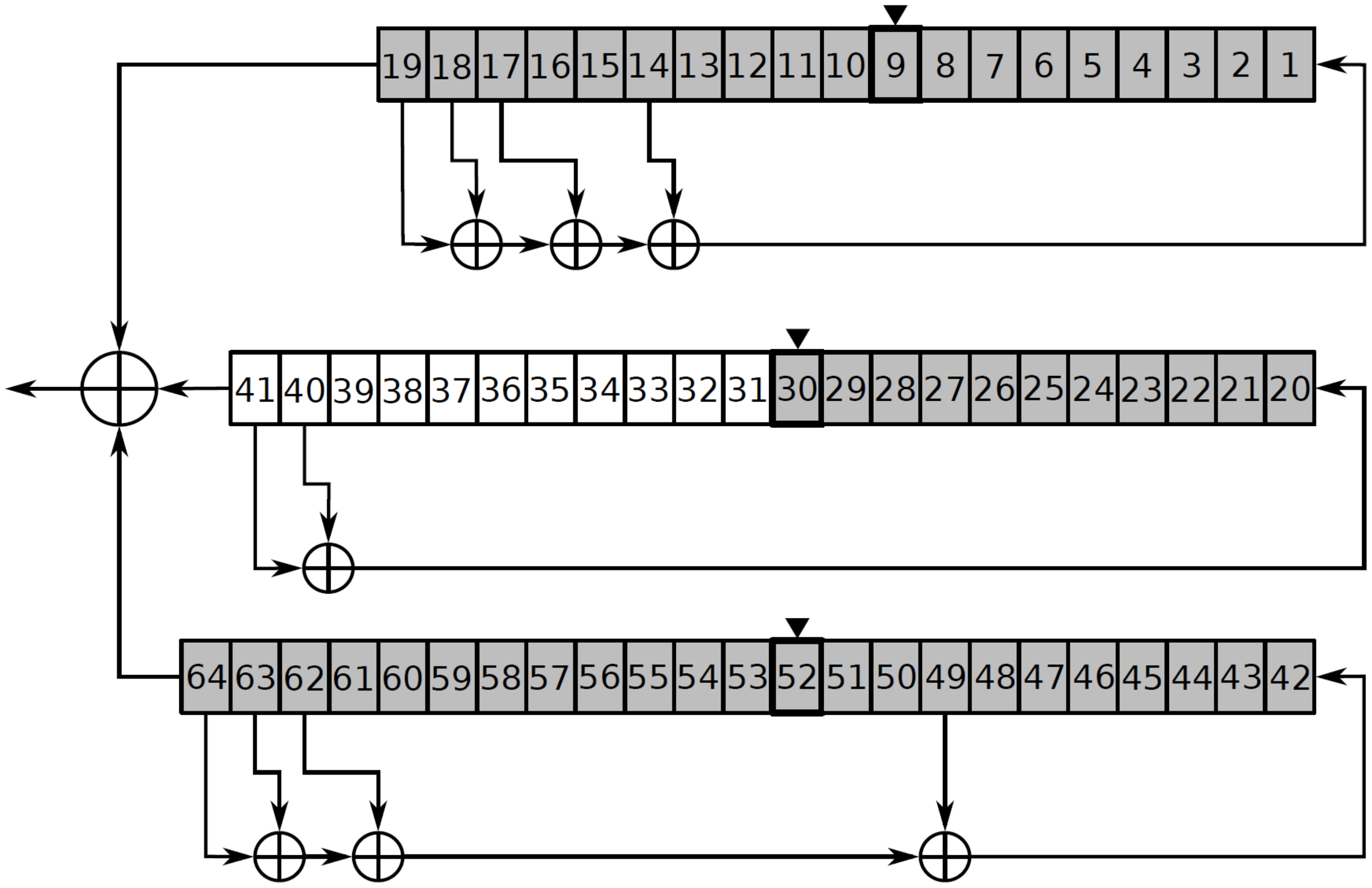
1. Guess states of the LFSRs R1 and R3, and also guess the values of 11 (out of 23) cells of the LFSR R2 (i.e. guess the values of 53 out of 64 cells).

2. Given a known keystream fragment, effectively calculate the remaining 12 bits of R3.

3. Check the obtained initial state (64 bits) by producing the keystream fragment, and comparing it with the given keystream.

In fact, it is a brute force algorithm on the reduced search space. Search space is reduced from 2^{64} to 2^{53} .

Anderson attack



GPU implementation

Anderson attack suits well for GPU.

We made a CUDA implementation. Bit-slicing technique allowed us to achieve very high speed.

Speed on Intel Core i7 930 is 37 millions keys per second.

Speed on NVIDIA GeForce GTX 1050 Ti is 11950 millions keys per second (323 times higher).

It means that given one such GPU, we can process all search space (for one cryptanalysis problem) in 22 days.

AndersonAttack@home

- In October 2016 we launched the BOINC-based project AndersonAttack@home.
- No checkpoints, quorum of 2.
- Deadline of 1 day.
- We generated workunits for 10 cryptanalysis problems.
- 2^12 = 4096 workunits were produced for each problem by varying values of the first 12 out of 53 cells. So, in each workunit the values of 12 out of 53 cells are fixed
- The values of the remaining 41 cells (2^41) are varied on a host.
- On average, it took about 10 minutes to process 1 workunit.

AndersonAttack@home

- 41 920 workunits were generated in total.
- 142 hosts of 78 crunchers took part in the experiment.
- The project’s performance was about 5 teraflops.
- All 10 problems were solved in 5 days.
- All workunits were processed in 7 days.

AndersonAttack@home

- The performance was similar to that of a computing cluster, equipped with 30 modern GPU.

- It is quite hard (at least for us) to get access to such cluster.

- Short deadline helped us to solve all the tasks very fast.

- In fact we had been successfully using our BOINC-based project in unusual way (“cluster-style”) – to quickly solve a relatively simple problem.

Connection with the umbrella project Optima@home

- The described experiment is quite small. However, it is interesting from the scientific point of view.

- The best way: don’t launch new BOINC project for such experiments, but launch it in an umbrella project.

Thank you for your attention!