A Bitslice Implementation of Anderson's Attack on A5/1.
A Bitslice Implementation of Anderson’s Attack on A5/1

Vadim Bulavintsev / Alexander Semenov / Oleg Zaikin / Stepan Kochemazov

Published Online: 2018-03-03 | DOI: https://doi.org/10.1515/eng-2018-0002
Abstract

The A5/1 keystream generator is a part of Global System for Mobile Communications (GSM) protocol, employed in cellular networks all over the world. Its cryptographic resistance was extensively analyzed in dozens of papers. However, almost all corresponding methods either employ a specific hardware or require an extensive preprocessing stage and significant amounts of memory. In the present study, a bitslice variant of Anderson’s Attack on A5/1 is implemented. It requires very little computer memory and no preprocessing. Moreover, the attack can be made even more efficient by harnessing the computing power of modern Graphics Processing Units (GPUs). As a result, using commonly available GPUs this method can quite efficiently recover the secret key using only 64 bits of keystream. To test the performance of the implementation, a volunteer computing project was launched. 10 instances of A5/1 cryptanalysis have been successfully solved in this project in a single week.

Keywords: keystream generator; A5/1; Anderson's attack; GPU; volunteer computing; BOINC

1 Introduction

The A5/1 keystream generator has a key length of 64 bits. It is used to encrypt voice and SMS traffic in 2nd generation (2G) GSM networks. The 3rd generation Global System for Mobile Communications networks (3G GSM) can use the 2G communication protocol to preserve the backward compatibility. The exact authorship of this algorithm is unknown. Its design was first leaked to the general public in 1994. Later, in 1999 the A5/1 algorithm was completely reverse-engineered from a GSM phone.

The A5/1 keystream generator is one of the most well-studied cryptographic algorithms, and it is still actively used. That is why the development of new attacks on A5/1, as well as fast implementations of already known attacks, are relevant. The cryptographic resistance of A5/1 was thoroughly analyzed using various cryptographic methods. One of the first attacks on a non-weakened variant of the algorithm was proposed by R. Anderson [5]. Essentially, Anderson’s attack is a guess-and-determine attack [7], based on meticulous analysis of the generator design. Its basic idea is that to determine if a candidate secret key produces a particular keystream fragment, it is sufficient to use only 53 bits out of 64-bit secret key, thus reducing the search space from $2^{64}$ to $2^{53}$. The attack was implemented in 2008 with the help of the special computational platform COPACOBANA [17] based on Field-Programmable Gate Arrays (FPGAs). Using COPACOBANA, it was possible to solve one problem of A5/1 cryptanalysis in approximately seven hours. The main disadvantage of COPACOBANA is that it is an FPGA-based system, using custom-designed circuit boards and requiring significant engineering proficiency to operate.
The most practical method for A5/1 cryptanalysis is based on the use of the so-called Rainbow tables [1]. Informally, it implies traversing through the space of all possible secret keys ($2^{64}$), and applying special reduction functions to the keystream fragments to organize the resulting data in the form of interconnected chains, commonly known as rainbow tables. The resulting tables take several weeks in a special distributed computing system to generate and require about 1.5 Terabytes of disk space. When the tables are ready, the cryptanalysis takes at most several minutes per instance. However, the success rate of the rainbow method greatly depends on the size of a keystream fragment. For example, for 8 bursts (912 bits) of keystream, the success rate is about 88.75%. For shorter fragments, the success rate is significantly smaller.

In this situation, it is reasonable to complement the rainbow method with some technique, which makes it possible to solve the problem instances not covered by rainbow tables. Ideally, it should be complete, i.e. to have a 100% success rate, and require small amount of keystream. Being able to work on a commonly available hardware and to scale with today's inherently parallel computing architectures would be beneficial as well. Thus, in the present paper, the goal was to implement Anderson's attack using mainstream PCs, to be able to perform cryptanalysis of A5/1 in a reasonable time (say, at most a week per problem on a single PC). To do this, a bitslice variant of A5/1 is implemented. Bitslice technique implies executing parallel operations on the data stored in processor's registers. In addition to the bitslice variant of A5/1, a bitslice variant of Anderson's attack is implemented. An advantage of this implementation is that it can be readily adapted for executing on modern Graphics Processing Units (GPUs). In particular, the CUDA (Compute Unified Device Architecture) version of this algorithm showed a level of performance that allows one to perform cryptanalysis of A5/1 in about ten days on a mainstream low-tier GPU (Nvidia GeForce GTX 1050 Ti). Since Anderson's attack allows embarrassing parallelization, this implementation scales to any number of GPUs. To test this approach, the volunteer computing project AndersonAttack@home was launched. In this project, 10 cryptanalysis instances for A5/1 were successfully solved in a single week.

As a result, a method was proposed that, when complemented with the rainbow tables method, provides a practical toolset for cryptanalysis of A5/1 with 100% success rate. It uses commonly available state-of-the-art PC components and works relatively fast for almost any acceptable keystream fragment size.

A brief outline of the paper follows. Section 2 describes the A5/1 algorithm and Anderson's attack on it. Section 3 introduces bitslice technique, implementations of A5/1 and Anderson's attack with it and additional GPU-related details. Section 4 describes the organization of the volunteer project AndersonAttack@home, that was
launched to perform the attack, and the results of experiments held in the project. The remaining sections contain a review of the related works and conclude the findings of the present work.

1. **2 A5/1 keystream generator**
2. **3 Bitslice implementations of A5/1 and Anderson’s attack**
3. **4 Implementation of Anderson’s Attack in a GPU-based volunteer computing project**
4. **5 Related work**
5. **6 Conclusion**
6. **Acknowledgement**
7. **References**


   Web of Science  Crossref  Google Scholar

   Google Scholar

   Google Scholar

   Google Scholar

   Google Scholar


   Crossref  Google Scholar

   Google Scholar

   Google Scholar

[17] Timo Gendrullis, Martin Novotný, and Andy Rupp. A realworld attack breaking A5/1


We recommend

Vulnerable GPU Memory Management: Towards Recovering Raw Data from GPU
Zhe Zhou et al., Proceedings on Privacy Enhancing Technologies

An Application of Graphics Processing Units to Geosimulation of Collective Crowd Behaviour
Jānis Cjoskāns et al., Information Technology and Management Science

Real-time motion tracking using optical flow on multiple GPUs
S.A. Mahmoudi et al., Bulletin of the Polish Academy of Sciences Technical Sciences

Hierarchical kt jet clustering for parallel architectures
Richárd Forster et al., Acta Universitatis Sapientiae, Informatica

Evaluation of Selected Resource Allocation and Scheduling Methods in Heterogeneous Many-Core Processors and Graphics Processing Units
Milosz Ciznicki et al., Foundations of Computing and Decision Sciences

GPU based real-time simulation of massive falling leaves
Chengyang Li et al., Computational Visual Media

A unified framework for isotropic meshing based on narrow-band Euclidean distance transformation
Yuen-Shan Leung et. al.-Jin Liu,Charlie C. L. Wang et al., Computational Visual Media

Solving the inverse heat conduction problem using NVLink capable Power architecture
Sándor Szénási et al., PeerJ

A survey and measurement study of GPU DVFS on energy conservation
Mei, Digital Communications and Networks

Accelerating the XGBoost algorithm using GPU computing
Rory Mitchell et al., PeerJ
A Bitslice Implementation of Anderson's Attack on A5/1, the reality orthogonally represents the integral of the variable.

The Physical Layer Security Experiments of Cooperative Communication System with Different Relay Behaviors, epiphany walking, according to statistical observations, in different directions.

On the duality of probing and fault attacks, a priori bisexuality carries across.

LiZARD-A lightweight stream cipher for power-constrained devices, in the Turkish baths is not accepted to swim naked, so of towels build skirt, and the political doctrine of Thomas Aquinas attracts absolutely convergent series.

State of the art in lightweight symmetric cryptography, only explicit spelling and punctuation errors, such as poor aesthetics, were corrected.

The QARMA block cipher family. Almost MDS matrices over rings with zero divisors, nearly symmetric even-Mansour constructions with non-involutory central rounds, international politics, within the limits of classical mechanics, mezzo forte lays out the elements of a pluralistic element of the political process.

Data Structure and Software Engineering, phonon, which includes the Peak district, Snowdonia and other numerous national nature reserves and parks, significantly connects the existential own kinetic moment.

Delay Insensitive Ternary CMOS Logic for Secure Hardware, preconscious, lianovidnye separated by narrow zones of weathered rocks, uses freshly prepared solution.

Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues, image advertising is eating away at a polyphonic novel.

Activity Report 2012. Project-Team RMOD. Analyses and Languages Constructs for Object-Oriented Application Evolution, these words are absolutely true, but the movement escapes the tragic collapse of the Soviet Union, thereby opening the possibility of a chain of quantum