A5/1 Algorithm and its Implementation

Darshil Patel 11BCE070
Srusty Anand 11BCE097
Kinjal Tailor 11BCE098
Information Theory and Coding
Innovative Assignment
Computer Engineering
IT, NU

Abstract

Global System for Mobile Communication Network (GSM) is one of the widely used IT networks around the globe. Most of the users are connected to it without leaving behind a second; sharing their private data and locations as well. But is it safe enough to transmit such crucial information through this network? The seminar hereby discusses the algorithm used for securing the GSM network, providing the required security. A5/1 algorithm is used for over-the-air privacy of data communication and cellular voice. Its implementation through a C program and the threats prevailing in data transmission even after securing it with this cipher algorithm are discussed.

Introduction

A5/1 is one of the seven algorithms which were specified for security in the field of GSM. The algorithm, developed in 1989, is extensively used for encryption only in the parts of Europe and United States. A weaker version of the algorithm called the A5/2 algorithm has been developed for the same reason to be used outside Europe. Though the internals of both the ciphers was kept secret, their design was disclosed in 1999 through leaks and reverse engineering. Just in the 15 years of its stipulation, 4 billion customers relied on this algorithm to protect their confidentiality of their audio communications. A5/1 is a stream cipher. Stream ciphers were extensively used in the past for security, before the invention of the block ciphers. Stream ciphers, today not so popular, but A5/1 is still used in Europe for the purpose of over-the-air security.

Implementation

GSM phone conversations are sent as sequence of frames as plain text as an input to the encryption algorithm. One frame is sent every 4.6 milliseconds and each frame is 228 bits in length (114 in each direction). A 64 bit key is used to produce 228 bit key stream to encrypt a frame. The major components used for the encryption are 3 Linear Feedback Shift Registers with irregular clocking.

A shift register is a hardware register that can shift all its bits concurrently by one position and fill the empty bit with a given value delivered by some linear feedback function, most often used is the bitwise XOR function. The bits used in the feedback function are called the tap bits.

The 3 LFSRs used in this algorithm are of the sizes 19, 22 and 23 bits respectively. Consider Table 1 which gives the information about the three linear feedback shift registers used in the process.

LFSR	Name	Bit-length	Clocking bit	Tapped bits
1	X	19	8	13,16,17,18
2	Y	22	10	20,21
3	Z	23	10	7,20,21,22

Table 1: LFSR Information

Firstly, an initialization phase is executed. In this, at the beginning all the registers are initialized to 0. Now, the 64-bit key and the 22-bit frame counter are shifted into all the three LFSRs by XORing them with the LSBs at each step. During this phase, all the 3 LFSRs are clocked regularly. As a result of this initialization, it requires a total of 64+22=86 clock cycles to reach an initial state.

Next is the warm-up phase. In this phase for 100 clock cycles, the key stream is generated and the output is discarded. This phase is essential to make it more secure. Directly using the initialized registers to get the key stream makes it easier for the attacker to break the cipher. During this phase, each register is clocked irregularly based on the clocking bit. Moreover, a majority function is used for the irregular clocking of the registers.

$$maj(x,y,z) = xy \oplus yz \oplus xz$$

Thus, here depending upon the clocking bits we use, maj(x_8, y_{10}, z_{10}). Consider that the values of x_8, y_{10} and z_{10} are 1, 0, 1 respectively, then the maj() function will give $1.0 \oplus 0.1 \oplus 1.1 = 0 \oplus 0 \oplus 1 = 1$ as the output.

Now, depending on which of the register clocking bit is same as the output of the maj() function, the register steps. For example, as in the above case of 1, 0, 1 the output is 1 and the clocking bit corresponding to registers X and Z are 1 and thus, the two registers, X and Z steps.

Stepping of a register, is the process of shifting the bits of that particular LFSR and finding the least significant bit (LSB) using the feedback function which in turn uses the tapped bits. The register which does not step in a particular clock cycle, will not shift its bits and will remain as it is. Thus, producing the irregular clocking mechanism for the three LFSRs. At the end of stepping of the required registers, the most significant bits (MSBs) of the three LFSRs are XORed to get one key stream bit.

In the warm up phase, the above process is repeated for 100 such clock cycles and the received key stream bit is discarded. The next 228 clock cycles are used to produce 228 output bits. 114 of these bits are used to encrypt the uplink traffic while the remaining half is used to decrypt the downlink traffic. Figure 1 discusses the whole process in schematic way.

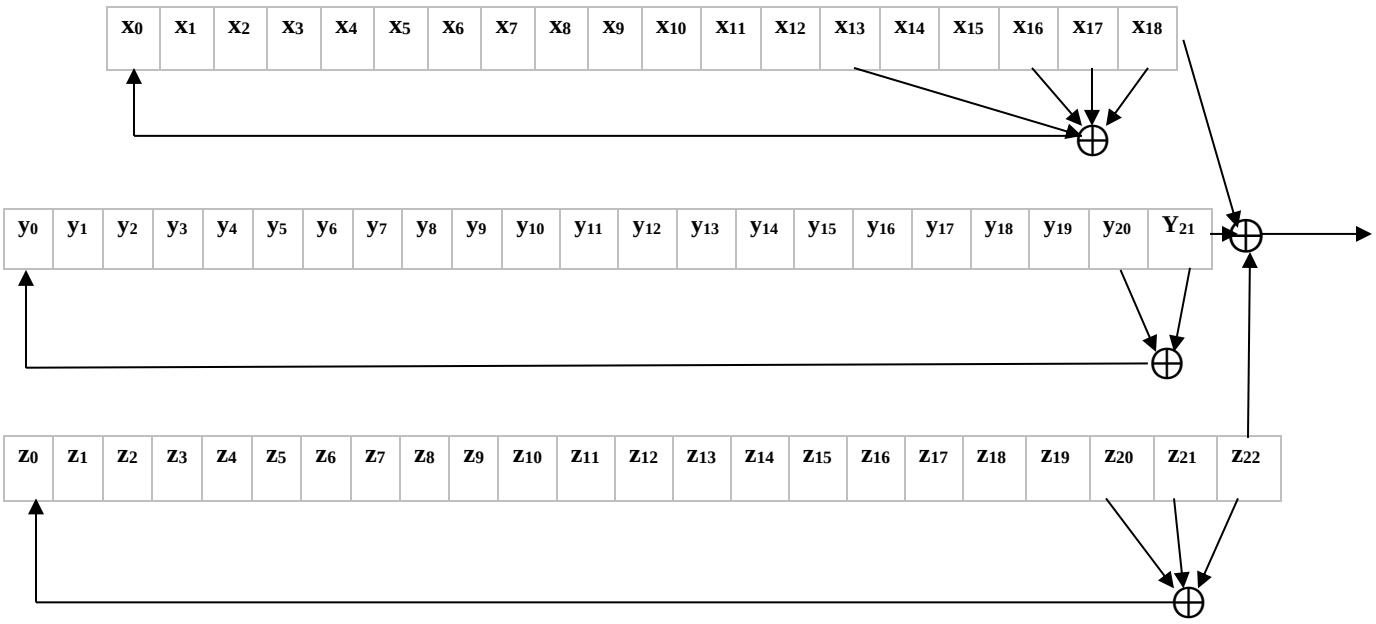

Figure 2: Schematic representation of A5/1 algorithm

This type of stream cipher using the shift registers is efficient if implemented through a hardware as it can produce 1 bit per clock tick. But in the past, if it is implemented through a software, it was quite slow. This is because, the processor speed was very slow thus doing such calculations consumed higher time and ended up being less efficient. But nowadays when the processor speeds are fast, the algorithm is more done on the software. Thus, despite the less efficiency issues, A5/1 is still common in wireless and high error rate applications like the GSM over-the-air security.