# A5-BASED GSM CRYPTOSYSTEM IMPLEMENTATION AND ANALYSIS

Daniel Okunbor, Fayetteville State University Fayetteville, NC 28301, U.S.A. (diokunbor@uncfsu.edu)
Chinyere Eghosa Amado, University of Calabar, Calabar, Nigeria (chi_baby201144@yahoo.com)
Rakesh Sharma,University of Maryland Eastern Shore, Princess Anne, MD 21853, U.S.A. (rsharma@umes.edu)

**ABSTRACT**

In this research, we explore cryptosystem that is implemented in the Global System for Mobile (GSM) Communications. GSM is a standard developed by the European Telecommunications Standards Institute (ETSI) to describe the protocols for second-generation (2G) digital cellular networks used by mobile phones. It was first deployed in Finland in July 1991. It is recently considered the de facto global standard for mobile communications – with over 90% market share, operating in over 219 countries and territories. A commonly used cryptosystem for GSM communications is the A5 ciphering algorithm. The A5 encryptions are stream ciphers based on a combination of Linear Feedback Shift Registers (LFSRs) with irregular clocking and non-linear combiner. The A5/1 uses 3 LFSRs and A5/2 uses 4 LFSRs. In this research, we will develop a general-purpose software Python library for A5 implementation.

## INTRODUCTION

The Global System for Mobile Communications or GSM for short has become the de facto global standard for mobile communications. The work on GSM started in 1982 when a study group called Global Special Mobile group was established immediately after the Conference of European Posts and Telegraphs (CEPT) that was held in 1982 and following the announcement of the use of 900 MHz spectrum for Pan-European mobile communication systems. Recognizing the possibility of having systems across national boundaries and learning from experiences from the immediate success of the Nordic Mobile Telephone (NMT) systems that was developed by a group of Scandinavian companies, the CEPT deemed it necessary to form the Global Special Mobile (GSM) group to study and develop a new Pan-European Mobile System. The group was provided a set of criteria to follow for the new cellular technology, including "good subjective speech quality, low terminal and service cost, support for international roaming, ability to support handheld terminals, support for range of new services and facilities, spectral efficiency, and finally ISDN compatibility" (electronics-notes.com).

In the 1991, the responsibility of GSM was shifted to the European Telecommunications Standards Institute and the GSM which was initially used to represent the group name, was subsequently changed to Global System for Mobile Communications. However, the "GSM" is a trademark owned by the GSM Association. GSM describes protocols for second-generation (2G) digital cellular networks used by mobile devices. GSM adopted Time Division Multiple Access (TDMA) for access within individual frequency channels and Frequency Division Multiple Access (FDMA) for access between channels. The Memorandum of Understanding (MOU) for the adoption GSM standards was signed by telecommunication operators from 12 members countries in 1987. This is believed to most significant milestone in the history of GSM and the desire for digitalization of mobile networks. It became clearly evident between 1982 and 1987 that the original vision of GSM was not sustainable. This agreement painstakingly brought the whole of European to rally in support of GSM bringing about a revolution in mobile networks.

As a standard, GSM provides a unified way in which mobile networks are designed and deployedin European countries. When the GSM Technical Specification was developed in 1987, Ministers from four (4) European Union countries lend their support at what is now called Bonn Ministers' Declaration. Similar standards were concurrently being developed in Asia and America. However, the explosive nature of the growth of GSM allowed its adoption in many other non-European enclaves. Mobile operators have pledged support, investment wise to GSM-enabled mobile networks. Note that I used GSM-enabled mobile networks here, this is because GSM is truly not a network but a collection of standards or protocols.GSM is the most widely accepted standard in telecommunications globally. It is a circuit-switched system that divides each 200 kHz channel into eight 25 kHz with bands 900 MHz and 1800 MHz.Presently, there are more than one billion mobile subscribers in more than 210 countries for GSM-enabled mobile networks.GSM provides basic to advanced voice and data services including roaming service. Roaming is the ability to use your GSM phone number in another GSM network. The GSM architecture is a layered model that

is designed to allow communications between two different systems. Similar to the OSI reference model for data communication, each layer passes suitable notifications to ensure the transmitted data has been formatted, transmitted, and received accurately. The schematic representation of the GSM architecture is depicted below.

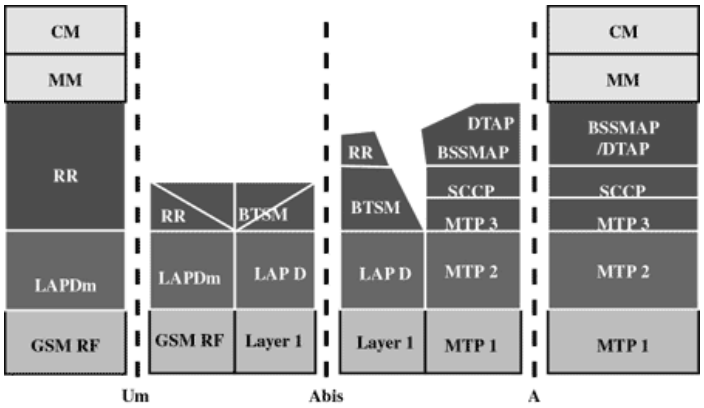


**Figure 1**. GSM Architecture (Source: https://www.tutorialspoint.com/gsm/gsm_quick_guide.htm)

Layer 3 consists of connection management (CM), the mobility management (MM), radio resource management (RR). Layer 2 is the data-link layer (based on modified LAPD -Link Access Protocol for the D channel, also referred to as LAPDm with m for mobile and the Message Transfer Part (MTP) of the Base Transceiver Station -BTS and Base Station System -BSS ) & protocol is used at the A interface) and Layer 1 is the physical layer. The sequence of operations in a GSM-based mobile network is indicated in the figure below:

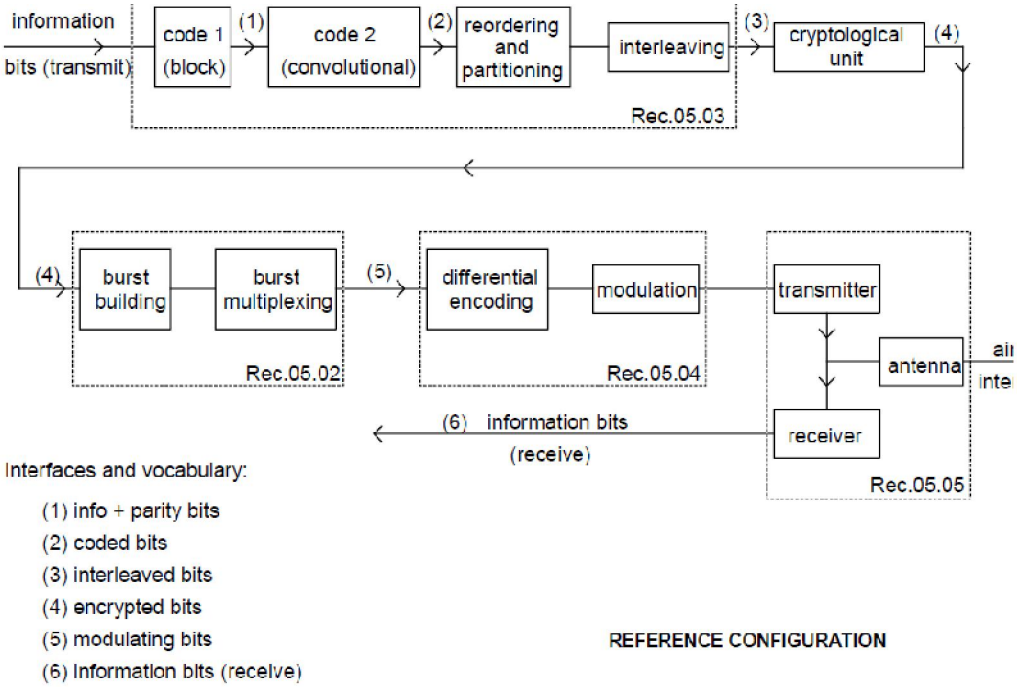


**Figure 2**. Reference Configuration (Source: www.etsi.org)

From the reference configuration above, we see that cryptological unit encrypts the bit frames from the interleaving process. The cryptosystem currently in use in the step of the configuration is the A5 encryption, which is the focus

of this research. It is important the ascertain in the GSM scheme where encryption is applied. A detailed analysis revealed the section of the GSM hyper frame where the encrypted bits are stored. The frame is

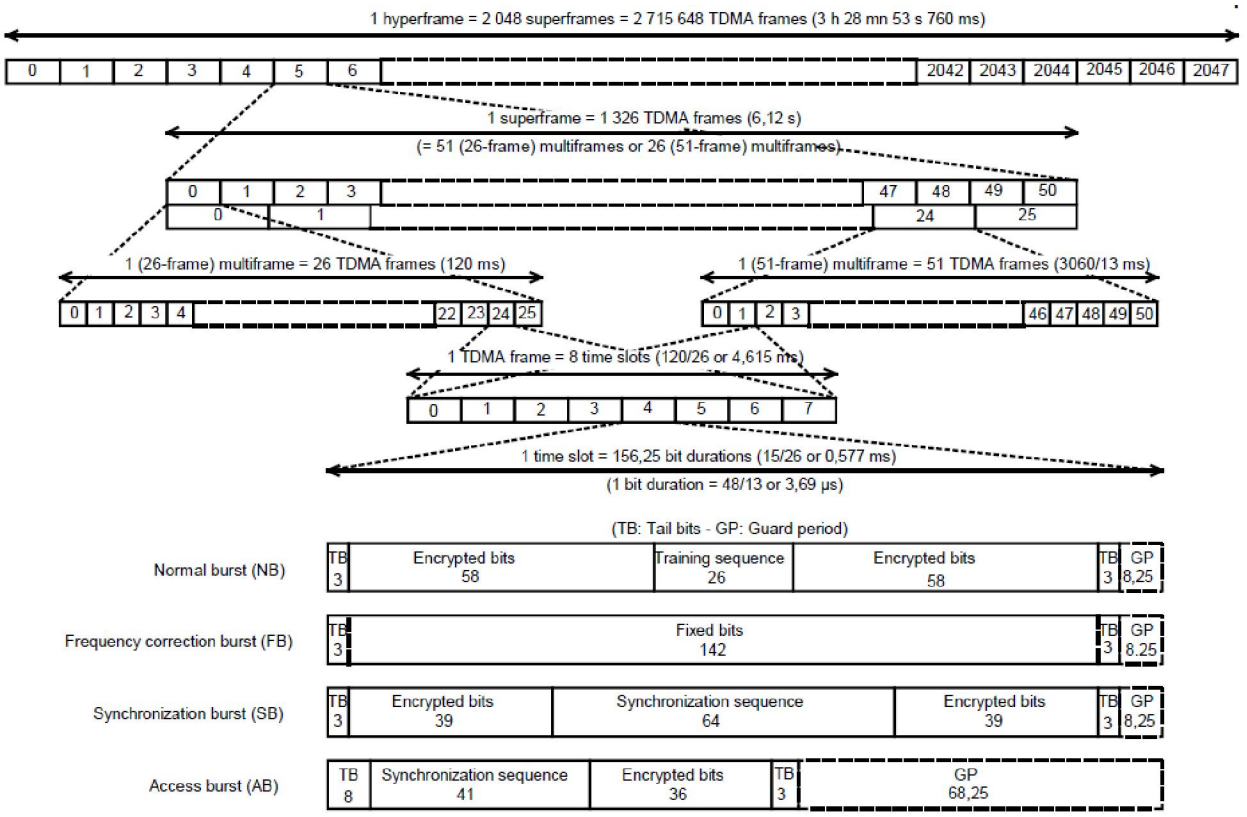


**Figure 3**. Time Frames and Bursts (Source: www.etsi.org)

From Figure 3, encrypted bits are used in the normal (116 bits), synchronization (78 bits) and access (36 bits) bursts.

This paper will provide a detailed analysis of linear feedback shift register (LSFR) as a stream cipher for use in A5. The paper is organized as follows: Section II will present the current state of affairs in LFSR and A5 encryption; Section III will describe in detail the linear feedback shift registers and its practical applications; and Section IV will examine A5 implementation using Python and parallel considerations.

## LITERATURE REVIEW

As described in the preceding section, GSM has become the communication protocol for mobile networks. According to the GSM Association reports, the number of GSM-based networks surpassed five (5) billion in February 2017 and GSM-based networks have gained over 80% of the total global mobile market with primary holdings in Asian, European and African countries. It is estimated that this number will increase to 5.7 billion subscribers by the end of this decade (www.gsma.com, 2017). It is this astronomical growth brings with it security concerns. The motivations for security in cellular telecommunications systems are 1) secure conversations, 2) Signaling data from interception, 3) To prevent cellular telephone fraud. The primary security is based on the encrypted bits in the normal, synchronization and access bursts as described in the previous section. The security methods standardized for the GSM System makes it the most secure cellular telecommunications standard currently available. Although the confidentiality of a call and anonymity of the GSM subscriber is only guaranteed on the radio channel, this is a major step in achieving end-to- end security. The subscriber's anonymity is ensured through the use of temporary identification numbers. The confidentiality of the communication itself on the radio link is performed by the application of encryption algorithms and frequency hopping which could only be realized using digital systems and signaling (Technology, 2013).

The security mechanisms of GSM are implemented in three different system elements:The Subscriber Identity Module (SIM), The GSM handset, and The GSM network.The SIM contains the IMSI, the individual subscriber authentication key (Ki), the ciphering key generating algorithm (A8), the authentication algorithm (A3), as well as a Personal Identification Number(PIN). The GSM handset contains the ciphering algorithm (A5). The encryption algorithms (A3,A5, A8) are present in the GSM network (Technology, 2013). The GSM network authenticates the identity of the subscriber through the use of a challenge-response mechanism. A 128-bit random number (RAND) is sent to the mobile station (MS). The MS computes the 32-bit signed response (SRES) based on the encryption of the random number (RAND) with the authentication algorithm (A3) using the individual subscriber authentication key (Ki). Upon receiving the signed response (SRES) from the subscriber, the GSM network repeats the calculation to verify the identity of the subscriber. If the received SRES agrees with the calculated value, the MS has been successfully authenticated and may continue. If the values do not match, the connection is terminated and an authentication failure indicated to the MS (Technology, 2013). The A3 and A8 are implemented in combination in COMP128 in the GSM standard. A3 encryption accepts 128-bit random number and 128-bit private key and produces a 32-bit SRES signed response, while the A8 accepts 128-bit random number and 128-bit private key and produces a 64-bit cipher key. A combination of A3 and A8 in COMP128 produces 32-bit SRES and 64-bit cipher key (54-bit with 10 zeros added). The 64-bit key is used as a session key for A5 encryption. Both encryptions are based on compression function that allows for bit substitutions to produce 32-bit and 64-bit outputs. This paper focuses on A5 encryption, readers interested in these algorithms should read (Brumley, 2003). A5 encryption n is a stream cipher that is based on linear feedback shift register. A stream cipher encrypts one bit or one character at a time. A bit is encrypted and the encrypted bit sent to the receiver, following which the next bit is encrypted and sent to the receiver until all bits are sent. In the following sections, the linear feedback shift register and A5 algorithm will be described.

## I. Linear Feedback Shift Registers

Linear feedback shift registers (LFSRs) are used in cryptography and coding theory (Schneier, 1996). They consist of two primary components, the feedback function and the shift register. A register comprises of bit stream and therefore, a shift register is one that the bit stream to shift one bit to the right, i.e., towards the least significant bit (LSB). The feedback function updates the left-most bit using other XOR (exclusive OR) of bits in the register. Because of the finiteness of the bit stream, the LFSR will eventually result in repeating bit sequence and the maximum duration for this to happen is called the period. Because an n-bit stream LFSR has $2^n$-1 possible bit streams, the maximum periodis $2^n$-1 and the degree is n. The pictorial representation of LFSR is shown in the figure below.

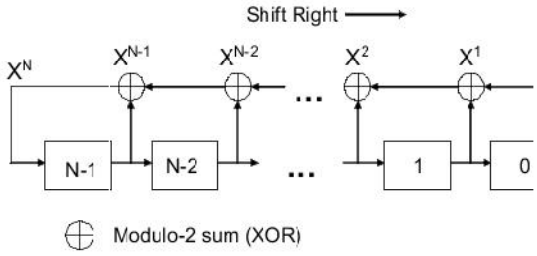


**Figure 4**. Linear Feedback Shift Register

The mathematical representation is

$$s_{i+n} = \sum_{j=0}^{n-1} p_j * s_{i+j}(mod\ 2)\,, s_i, p_j \epsilon \{0,1\}; i = 0,1,2, \ldots$$

where $p_i$ is the contribution of the *i*-th register and $(s_0, s_1, \ldots, s_{n-1})$ represent the initial state of the registers. The implementation of the linear feedback shift register in Python is given below. This is an iterable class that that can be used to generate bit stream given the initial state bit stream and the contribution bit stream in strings. For example, given "001" and "110" as the initial bit stream and combination bit stream, after 30 iterations, the bit stream (reversed) will be:**10100111010011101001110100111010 0.**

According to (Schneier, 1996), the basic approach to designing a keystream generator using LFSRs is to take one or more LFSRs, generally of different lengths and with different feedback polynomials. The key is the initial state of the LFSRs and the LFSRs are shifted to generate one bit (this process is called **clocking**). The output bit is preferably a nonlinear function (called combining function), of some of the bits of the LFSRs.

Table 1: LFSR in Python

```
#Linear Feedback Shift Register
#Author: Daniel Okunbor, Fayetteville State University
#The class LFSR is an iterator that grows the start_bitStream each
#time you call the next function
# Usage: lfsrString = LFSR("001", "110")
#next(lfsrString) --> '0100'
#next(lfsrString) --> '10100'
#next(lfsrString) --> '110100'
class LFSR:

def __init__(self, start_bitStream, coeffBits):
        self.__outputBitStream = self.__reverseBits__(start_bitStream)
        self.__coefficientBitStream = self.__reverseBits__(coeffBits)
self.__size = len(coeffBits)

def __iter__(self):
        return self

    def __next__(self):
        self.__outputBitStream = self.__outputBit__() +\
                    self.__outputBitStream
        return self.__outputBitStream
    #linear combination of product of start bits and coefficients
def __outputBit__(self):
outBit=0
        for i in range(self.__size):
outBit += int(self.__coefficientBitStream[i])* \
int(self.__outputBitStream[i])
        return str(outBit%2)
    #Reverses the bits stream
def __reverseBits__(self, stringBits):
        return stringBits[-1:-len(stringBits)-1:-1]
```

Typically, LFSRs are based on primitive polynomials in modulus 2. Primitive polynomials are very much like prime numbers, but in polynomial form and they are irreducible and only divisible by $x^{2^n-1}+1$. Primitive polynomials produce maximum period that is essential in LFSRs. An LFSR based on primitive polynomial (contribution bit stream) will result is a more secure key pseudo generation for encryption and are less susceptible to attacks. Generating primitive polynomials mod 2 can be a daunting task, however, there are several tables of primitive polynomials in the literature, see for example (Schneier, 1996). The A5 encryption as described in the following section is based on LFSRs with distinctive primitive polynomials.

## II. A5 Encryption and Python Implementation

As indicated in preceding Sections, A5 is an encryption for GSM-enabled mobile networks. The A5 encryption is used to generate the key for encrypted bits in three distinct phases of the GSM architecture as described in Section

II. A3 and A8 encryptions are used to support the A5 algorithm. The A5 keystream is used encrypt the plaintext. The A5 encryption which metamorphosed into different versions was originally keep secret until they were disclosed in 1994 and reverse engineered in 1999

There currently A5 encryption algorithms featured in GSM, namely, A5/0, A5/1, A5/2 and A5/3. A5/0 uses no encryption with the ciphertext being the same as the plaintext. A5/1 is claimed to the original A5 algorithm and was designed specifically for use in European countries and A5/2, which is a weaker model was developed for export to other countries. Version A5/3 is described as a strong encryption that was created as part of the 3rd Generation Partnership Project. While A/2 and A/3 may be interesting model for research, this paper focuses only on A5/1. Several variants of A5 and LFSR-based encryptions claimed to possess better security properties than A5/1 have also been developed, see (Schneier, 1996; Paar and Petzl, 2010).

A5/1 is based on 3 LFSRs whose degrees are 19, 22 and 23 for a total of 64-bit long. Their primitive polynomials and associated periods are indicated below:

$$LFSR\ 1: x^{19} + x^{18} + x^{17} + x^{14} + 1, \quad (2^{19} - 1)$$
$$LFSR\ 2: x^{22} + x^{21} + 1, \quad (2^{22} - 1)$$
$$LFSR\ 3: x^{23} + x^{22} + x^{21} + x^{8} + 1, \quad (2^{23} - 1)$$

The pictorial representation is shown in the figure below.

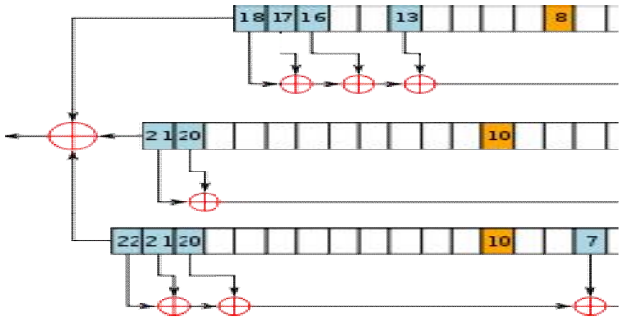


**Figure 5**: A5/1 Registers

The clocking bits are 8,10, and 10 for LFSRs 1, 2 and 3, respectively. The clocking is based on a majority function $f(x, y, z) = (x', y', z')$, defined as follows:

$$f(x, y, z) = sum(x, y, z), majority = \begin{cases} = 1, if\ sum > 1 \\ = 0, otherwise \end{cases}$$

The register is only shifted if the clocking bit is equal to the majority. The Python implementation of the A5/1 using the LFSR class is available on request. We developed two major functions pseudoBit and pseudo5Bit that will generate a single bit using A5 encryption.A sample output keystream obtained from the execution code with simple input stream is:

**01011110111101101011111100001010000111101110101100111000101000010110101100110101100011101111100000110**

For a full implementation of A5, it is imperative we first generate many bits before generating the keystream in other to ensure confusion in the keystream. A complete A5 algorithm to generating the keystream can be found in the literature (Schneier, 1996; Sadkhan and Jawad, 2015).

## REFERENCES

Technology (2013). GSM security and encryption, https://www.slideshare.net/RKUnited/gsm-securityand-encryption-16723070.

Schneier, B. (1996). Applied Cryptography, Protocols, Algorithms and Source Code in C, Second Edition, *John Wiley & Sons, Inc*.

Paar, C. and Petzl, J. (2010). Understanding Cryptography, A Textbook for Students and Practitioners, *Springer-Verlag*.

Sadkhan, S.B. and Jawad, N. H. (2015). Simulink Based Implementation of Developed A5/1 Stream Cipher Cryptosystems, *International Conference on Communication, Management and Information Technology*, *Procedia Computer Science 65*, 350-357.