Attacks on A5/1 Cryptography Algorithm

Jordan Fernando / 13510069
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
fernandojordan.92@gmail.com

***Abstract*—This paper will discuss about A5/1 stream cipher algorithm and attacks on A5/1 algorithm. This algorithm is used to encrypt over-the-air communication in GSM. The A5/1 algorithm is used widely by cellular that use GSM today. There are several attacks that can be used to crack the A5/1 algorithm, but all of them require amazing computing power so that not just some people with some computer could crack the A5/1 algorithm. There are also ways to improve A5/1 algorithm security. The first section will discuss about introduction on what is A5/1 and the history. The second section will describes how to implements A5/1 algorithm. Third section will describes some of the attacks that can be used against A5/1 algorithm. Fourth section will discuss about ways to improve security. Fifth section will show the conclusion. Sixth section describes the terms usage and definition. And the last section shows the references that are used for this paper.**

***Index Terms*—A5/1, GSM, stream cipher, attacks, security**

I. Introduction

Privacy is a very important thing for each person. Nowadays, people use mobile phone in their daily life. Communication using mobile phone is very common that almost every message is sent using call, messaging, and even chatting. By using mobile phone, people need standard in the communication. The standard that are popular nowadays are Global System for Mobile Communications (GSM) and Wideband Code Division Multiple Access (W-CDMA).

In this paper we will discuss more only about GSM. GSM is a standard developed by European Telecommunications Standards Institute to describe protocols for second generation digital cellular networks used by mobile phone. In order to use GSM network standard, people need a Subscriber Identity Module (SIM) card. The SIM is a smart card that contains user's subscription information and phone book. This allows the user to retain his or her information by keep using the same card. GSM also designed with security. The system was designed to authenticate the subscriber using Personal Identification Number (PIN). In the communication, GSM uses several cryptographic algorithms such as A5/1, A5/2, and A5/3 stream ciphers to ensure over-the-air voice privacy.

Algorithm A5/1 is a stream cipher that is used to encrypt over-the-air communication in the GSM cellular telephone standard. Initially it was kept secret for Europe and United States GSM communication, but became public knowledge through leaks and reverse engineering.

There is also A5/2 algorithm which was a deliberate weakening of A5/1 that is used outside Europe and United States. In 1999, Ian Goldberg and David A. Wagner cryptanalyzed A5/2 in the same month it was published. It was shown that A5/2 was extremely weak such that it can be cryptanalyzed in the day it was published and by only using low end equipment to break it in real time. Since July 1, 2006, the GSM Association (GSMA) mandated that GSM mobile phones will not support A5/2 cipher any longer, because of the weakness, and A5/1 was being used for global instead. In 2000, around 130 million GSM customers relied on A5/1 to protect the confidentiality of their voice communication; by 2011, it was 4 billion. This paper will only discuss about A5/1 algorithm and the attack that can be used against it.

II. A5/1 Algorithm

A5/1 algorithm is an encryption algorithm that is using stream cipher. Stream cipher is a symmetric key cipher where the plaintext digits are combined with a pseudorandom cipher digit stream. The pseudo random is using a key. In a stream cipher each plaintext digit is encrypted one at a time with the corresponding digit of the generated cipher digit stream, to give a digit of cipher stream. By using the stream cipher means that A5/1 algorithm is very fast so that it can be used to encrypt voice on the call.

GSM transmission is organized as sequences of frames every 4.615 milliseconds. Each frame contains 114 bits representing the digitized A to B communications and 114 bits representing digitized B to A communication. Each conversation can be encrypted by using a new session key K. For each frame, K is mixed with a publicly known frame counter Fn, and the result serves as the initial state of a generator which produces 228 pseudo random bits. These bits are XOR'ed by the two parties with the 114+114 bits of the plaintext to produce 114+114 bits of cipher text.

A5 is built from three short linear feedback shift registers (LFSR) of length 19, 22, and 23 bits, which are denoted by R1, R2, and R3. The rightmost bit in each register is labeled as bit zero. The taps of R1 are at bit positions 13. 16, 17, 18; the taps of R2 are at bit positions 20, 21; and the taps of R3 are at bit positions 7, 20, 21, 22 (see Figure 1). Each register has a single "clocking" tap. For R1 it is bit 8, R2 is bit 10, R3 is bit 10). At each clock cycle, the majority function of the clocking taps is calculated and only registers that the clocking taps agree with the majority bit are clocked. This process is called the stop/go clock control. When a register is clocked, the taps are XOR'ed together, the register is shifted to left and the result is stored at the rightmost bit of the left-shifted register.

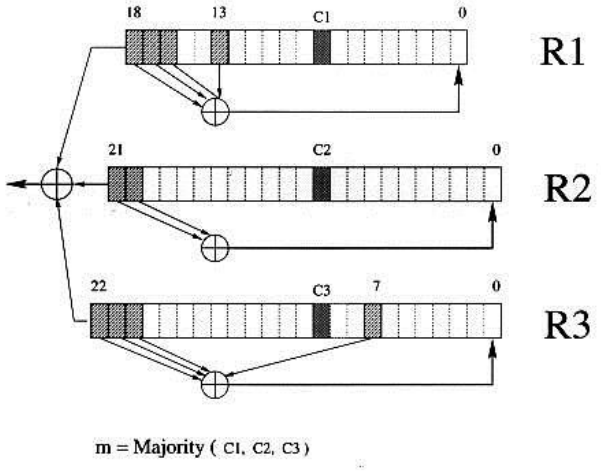


Figure 1: The A5/1 stream cipher.

The process of generating pseudo random bits from the session key K and the frame counter Fn is carried out in four steps:

- The three registers are zeroed, and then clocked for 64 cycles (ignoring the stop/go clock control). During each period each bit of K (from LSB to MSB) is XOR'ed in parallel into the LSB's of the three registers.
- The three registers are clocked for 22 additional cycles (ignoring the stop/go clock control). During this period the bits of Fn (from LSB to MSB) are again XOR'ed in parallel into the LSB's of the three registers. The contents of the three registers at the end of this step are called the initial states of the frame.
- The three registers are clocked for 100 additional clock cycles with the stop/go clock control without producing any outputs.
- The three registers are clocked for 228 additional clock cycles with the stop/go clock control in order to produce the 228 output bits. At each clock cycle, one output bit is produced as the XOR of the msb's of the three registers.

The pseudo random bits that are generated will be XORed with the plaintext and then sent to the other user. The other user can decrypt the messages if they know the key by processing the pseudo random key generator in the same way to process the pseudo random bits and XORed it with the cipher text.

Since the bit that are sent is in total of 228, so every time the cellular phone receive the messages the process of encryption and decryption run at the same time, 114 bits are used for encryption and the other 114 bits are used for decryption.

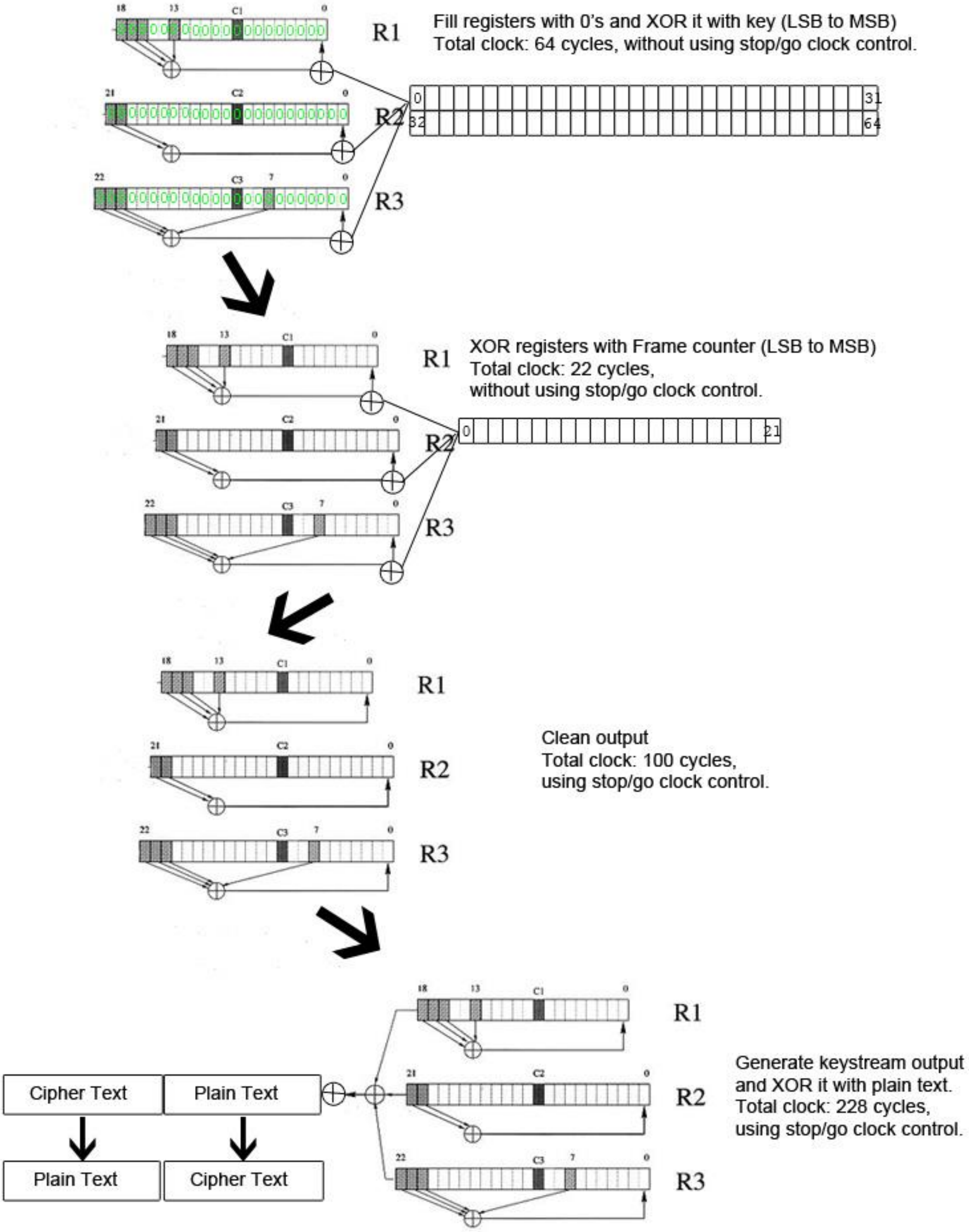
This algorithm is very straightforward and doesn't require much resource to process so that it can be used on cellular phone. The algorithm can also be simulated easily using a computer, and that's why we can also make a call using computer software such as Skype. The total process of encryption and decryption can be seen on Figure 2.

III. ATTACK

In this paper, we can assume that the attacker knows about the outputs of the A5/1 algorithm during some initial period of the conversation, and his/her goal is to find the key in order to decrypt the remaining part of the conversation.

Here are some extreme attacks descriptions:

1. Brute Force Attack
   In brute force attack, we need to compute exhaustive search with the time complexity of O(2^{64}) and it is meaningless because we need about in total of 18446744073709551616 computations which takes about 3-6 years in real time for just one frame of 228 bits. The decryption is not in real time and the conversation maybe meaningless after 3-6 years.
2. Table Look Up Attack
   Using table look up attack, we need in total of 2^{64} memories to store the look up table in total. That cost about 18446744 Terra Bytes in total, and we also need a lot of computing power to process such big data like that.



Fill registers with 0's and XOR it with key (LSB to MSB)
Total clock: 64 cycles, without using stop/go clock control.

XOR registers with Frame counter (LSB to MSB)
Total clock: 22 cycles,
without using stop/go clock control.

Clean output
Total clock: 100 cycles,
using stop/go clock control.

Generate keystream output
and XOR it with plain text.
Total clock: 228 cycles,
using stop/go clock control.

Figure 2. Overview of the A5/1 Algorithm process

The security of the A5/1 encryption algorithm was analyzed in several papers. The known attacks can be summarized as this:

- Briceno[2] found out that in all the deployed versions of the A5/1 algorithm, the 10 least significant of the 64 keys bits were always set to zero. The complexity of exhaustive search is thus reduced to O(2^{54}).
- Anderson and Roe[1] proposed an attack based on guessing the 41 bits in the shorter R1 and R2 registers, and deriving 23 bits of the longer R3 register from the output. However, they occasionally have to guess additional bits to determine the majority-based clocking sequence and thus the total complexity of the attack is about O(2^{45}). This attack needs more than one month to find one key.
- Golic[3] described an improved attack which requires O(2^{40}) steps. This attack is based on the solution of a system of linear equations. This attack takes more time than the previous algorithm.
- Golic[3] describes a general time-memory tradeoff attack on stream ciphers which was independently discovered by Babbage[2] two years earlier), and concludes that it is possible to find the A5/1 key in 2^{22} probes into random locations in a computed table with 2^{42} 128 bit entries. Since such a table requires a 64 terabyte hard disk, the space requirement is unrealistic. Alternatively, it is possible to reduce the space requirement to 862 gigabytes, but then the number of probes increases to O(2^{28}). Since random access to the fastest commercially available PC disks requires about 6 milliseconds, the total probing time is almost three weeks. And it can only be used to attack GSM phone conversations which last more than 3 hours, and it is unrealistic.

As we can see in the papers showing the attack to A5/1 stream cipher algorithm, that all of the attack requires a very large memory to compute the tables. If we want to reduce the total memory to be used, we need more time to attack the A5/1 algorithm. In that case, A5/1 algorithm is still safe for now, because most people can’t attack it easily. It needs a lot of cost to attack A5/1 algorithm because of the computation needed. Only computer such as Cost-Optimized Parallel Code Breaker (COPACOBANA) that can be used to crack A5/1 algorithm.

IV. Security improvement

The attacks that are described before require a lot of computation time or memories. We can improve more security in A5/1 algorithm but still retain the speed. The idea is to manipulate the encryption of the plaintext from using normal XOR operation into using block cipher encryption.

The block cipher will use the output key stream from A5/1 algorithm as the key to encrypt the data. So, in this block cipher, one block can be set to 114 bit. We can use Feistel network algorithm to encrypt the plaintext iteratively with at most consists of 30 iterations so that the algorithm can retain the speed. In the block encryption function, the message will not only be XOR’ed with the plain text, but also inversed and shifted to fulfill confusion and diffusion principle from Shannon. The key will also change every time in the iteration and the number of iteration is computed by using the key.

By adding the block cipher, rather than just XOR’ed the plaintext with the generated output of A5/1 algorithm, we have added the complexity of the A5/1 algorithm that require more than just compute all the possibility of the key with computing power.

V. Conclusion

From this paper, we can conclude that A5/1 stream cipher algorithm is safe enough from most of the attacks because of the need of a lot computing power and memory. But that restriction will also fade as the technology keeps on growing. We can improve A5/1 stream algorithm by combining it with block cipher algorithm and can still retain the speed of the algorithm because the algorithm needs to be used for transmitting data in real time. There are still many possibility of another improvement on A5/1 to become more secure and retain the speed.

VI. Terms Usage and Definition

Here is a list of some terms that are used in this paper with the definitions:

- Plain Text

  Plain text is the term used to describe the original message that is readable by everyone that we want to encrypt in order to limit the accessibility of the message to just some people or a person.
- Cipher Text

  Cipher text is the term used to describe the encrypted message that can only be decrypted by some people or a person by using a certain key that is given.
- Encryption

  Encryption is the term used to describe the process of converting plain text into cipher text by using some algorithm with provided keys.
- Decryption

  Decryption is the term used to describe the process of converting back cipher text into plain text by using some algorithm with provided keys.
- Cryptanalyst

  Cryptanalyst is a person who has job to analyze the encryption algorithm and try to find the plain text or the key of a cipher text. Some cryptanalysts decode the message by using known plain text.

That kind of attack is known as known-plain text attack. Cryptanalyst often find the security hole in some encryption algorithm and such thing can define whether the algorithm is safe enough to use or not.

- Stream Cipher

  Stream Cipher is a modern cryptography algorithm that is used to encrypt plain text by using a generated stream. The stream is generated by using a key with pseudo random algorithm and called as keystream. In stream cipher, each digit of plain text is encrypted one at a time with the corresponding digit of the keystream. A digit is typically a bit and the encryption is using an XOR operation. Stream cipher operates at high speed and low hardware complexity. The examples of stream cipher are A5/1, A5/2, FISH, RC4, SEAL, and SNOW.

- Key

  Key is a term that is used to describe a sequence of bit that is used to encrypt or decrypt the messages. Corresponding to the type of key, cryptography can be divided into two sections that are public key cryptography and symmetric key cryptography.

  In public key cryptography, the key that is used for encryption and decryption are different. Only the target of the message has the key for decryption (called as private key), all other people except the target can have the key for encryption (called as public key). To send a message from A to B, A first must request B's public key, and then A encrypt the message using B's public key and then send it to B. Then B can decrypt the message using his / her secret key. However, public key cryptography algorithm requires more time in encrypting and decrypting messages. The examples of public key cryptography are RSA, ElGamal, and DSS.

  In symmetric key cryptography, the key that is used for encryption and decryption are the same. So, there is a need to pass the key to the target in a safe way. Symmetric key cryptography algorithm works a lot faster than public key algorithm. The examples of symmetric key cryptography algorithm are DES, A5/1, A5/2, AES, Blowfish, and Serpent.

- XOR

  XOR is an operator in computing which is the extension of exclusive or. By using XOR operator, the value will result in true or 1 if the XOR operand has different values. Here is the truth table for XOR:

  Table 1. XOR truth table

Value A	Value B	Result
0	0	0
0	1	1
1	0	1
1	1	0

- LSB

  LSB is the abbreviation of Least Significant Bit. It points to the bit that is the rightmost in a byte. It is least significant because even if the value of the bit changes, it has only little effect on the byte value.

- MSB

  MSB is the abbreviation of Most Significant Bit. It points to the bit that is the leftmost in a byte. It is most significant because if the value of the bit changes, it will has a major effect on the byte value.

- Clock

  Clock is a term that is used to describe one process in a Central Processing Unit. When the clock start, the CPU executes a process and then moves to next process. The speed of clock in each CPU is different depends on the type of CPU.

- Bits Shifting

  Bits shifting is a method to shift the bits in registers either to the left or to the right. If the bits are shifted to the right, then the empty bit in the left will be filled with zero. If the bits are shifted to the left, then the new empty bit on the right will also be filled with zero.

- Registers

  Registers is a term that describe a sequence of bits.

- Brute Force

  Brute force is a basic algorithm method where we try to search for all possible solution naively in order to get the solution of a problem. This method will always get a solution, but the solution can be inefficient in time.

- COPACOBANA

  COPACOBANA, the Cost-Optimized Parallel Code Breaker, is an FPGA-based machine which is optimized for running cryptanalytical algorithms.

- Block Cipher

  Block cipher is a modern cryptography algorithm that is used to encrypt plain text by using a block encryption. The plain text is encrypted per blocks, the size of the block is usually defined in bytes. By using block cipher, the cryptanalyst cannot do the frequency analysis to crack the encryption algorithm. The examples of block cipher are blowfish, DES, AES, and RC5.

- Feistel Network

  Feistel Network is a symmetric structure used in the construction of block ciphers. The Feistel structure has the advantage that encryption and decryption operations are very similar, even identical in some cases, requiring only reversal of key schedule. Feistel Network can be used iteratively. Feistel Network is used by many block cipher algorithm such as DES and AES.

REFERENCES

[1] R. Anderson, M. Roe, A5, http://jya.com/crack-a5.htm, 1994.

[2] M. Briceno, I. Goldberg, D. Wagner, A pedagogical implementation of A5/1, http://www.scard.org, May 1999.

[3] J. Golic, Cryptanalysis of Alleged A5 Stream Cipher, proceedings of EUROCRYPT'97, LNCS 1233,pp.239{255, Springer-Verlag 1997.

[4] http://cryptome.org

[5] http://www.copacobana.org/

[6] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/kriptografi.htm

[7] http://www.amathnet.cz/Portals/0/workshopy/Beskydy%202012/dokumenty/122012dop/A5_1-TMTO_tabulky.ppt

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 Maret 2013

Jordan

Jordan Fernando / 13510069