Application of FSM Machine and S-Box in KASUMI Block Cipher to Improve Its Resistance Against Attack

Raja Muthalagu and Subeen Jain
(Corresponding author: Raja Muthalagu)

Department of Electrical and Electronic Engineering, Birla Institute of Technology and Science, Pilani
345055 Dubai International Academic City, Dubai, United Arab Emirates
(Email: raja.m@dubai.bits-pilani.ac.in)

(Received July 2, 2017; revised and accepted Oct. 22 & Nov. 5, 2017)

Abstract

In this paper, modifications in the original KASUMI block cipher is proposed by introducing a finite-state machine (FSM) and substitution box (S- box) to provide better confidentiality and integrity function in global system for mobile communications (GSM) and 3G networks. The FSM constitute the nonlinear combiner of SNOW 3G block cipher and it uses two S-box to provide strong diffusion. The addition of FSM in KASUMI is introducing the non-linearity in output bits and it will increase the complexity for an attacker to make an attack. Also, few changes are made in a KI and KL keys that are used in a different rounds of KASUMI to prevent various attacks such as a rectangle attack, sandwich attack, single key attack, etc. The simulation results show the performance improvement of the proposed modified KASUMI design is compared with the conventional KASUMI in terms both the encryption speed and encryption time taken.

Keywords: Finite-state Machine (FSM); KASUMI; S-box; SNOW-3G

1 Introduction

The rapid growth of mobile communications has increased the requirement of having secure network/communication between the users. Multiple ways of attacking or hacking a network are used by attackers. As the wireless mode of communications provides feasibility and ease to the users, the security of information being exchanged within two users or group of users is always at threat. Many algorithms have been proposed which are used for different encryption purposes [5, 6, 16]. Some have proved resistant towards attacks while some have high security issues with them and hence, proved as weaker one by attackers. For having secure network, encryption services and algorithms involved need to be robust and secure enough to provide end-to-end secure transmission of data among various users. It poses a challenge for designers to design highly secure and attack-resistant algorithm for encryption of data.

The KASUMI block cipher is used for providing encryption services in mobile networks like GSM, universal mobile telecommunications system (UMTS) and general packet radio service (GPRS). In UMTS, KASUMI is used in the confidentiality (f8) and integrity algorithms (f9) with names UEA1 and UIA1, respectively. In GSM, KASUMI is used in the A5/3 key stream generator and in GPRS in the GEA3 key stream generator. The KASUMI is evolved from MISTY1 algorithm to provide users safe and secure way for exchange of data. The KASUMI is a slightly modified version of MISTY1. And it provides easy hardware implementation that meets security requirement of 3G mobile communications. In [3, 4, 11, 12, 14] attacks related to KASUMI were discussed which indicate that KASUMI is weak algorithm. Though these attacks were very powerful and posed a threat on it, they were not considered due to impractical assumptions made as suggested in [11] by 3GPP society and thus, inapplicable to real-life attack on full KASUMI. But as there are chances of these or other attacks related to them being carried out practically, new algorithms are designed and worked upon to provide high confidentiality and security of data. The experimental study of the obtained encryptor from various researchers are demonstrated its effectiveness in protecting from many existing types of attacks aimed at block cipher algorithms [13]. Also Reference [10] present a new concept called a certificateless key insulated encryption scheme (CL-KIE).

In this paper, it proposed the modified KASUMI block cipher to improve its resistance against attack. The SNOW-3G block cipher is a another encryption algorithm for mobile networks and the concept of SNOW-3G is widely used in our proposed method. The SNOW-3G is used as UEA2 and UIA2 algorithm for providing confidentiality and integrity in 3rd Generation Partnership Project (3GPP) [7, 8]. It is seen as strong enough for carrying any attack as it has Rijndael's SR box and SQ box and LFSR. It also uses three 32-bit registers R1, R2, R3 and 16 s-boxes and each having capacity to hold 32-bits. As suggested in [9] inclusion of R3 had increased resistance of SNOW-3G towards algebraic attacks along with use of two S-box, and it can be used strengthen the proposed modified KASUMI. The FSM is known to provide resistance towards differential and linear attack and, algebraic attack as it uses $\mathbf{S}_1$ and $\mathbf{S}_2$ 32-bit boxes along with three registers R1, R2, and R3. The proposed KASUMI is using a part of SNOW-3G security module which are FSM machine and two S-boxes. We have taken into use three 32-bit Shift Registers two of them providing input to FSM. Besides introducing SNOW-3G, small change in KL keys of 1st and 8th round as well as change in KI keys is also made which is discussed in further section of paper.

This paper is organized in following way: Section 2 gives briefly an overview of KASUMI. Different functions and keys used in each rounds for their respective functions are described. Section 3 contains brief description of SNOW-3G and describes about its two modules LFSR and, FSM. Functioning of initialization and key-stream mode for generation of key-stream is also discussed. Section 4 discusses about our proposed work related to changes in KASUMI. Section 5 provides results of our work done. Finally conclusions are given in Section 6.

Throughout this paper, $\oplus$ stands for EX-OR operation, $\|$ stands for Concatenation operation and $\boxplus$ is addition modulo 2^{23}.

2 Overview of KASUMI

KASUMI is modified form of cipher algorithm MISTY1. It is Fiestel network of 8-rounds taking input of 64-bit and giving output of 64-bit by using 128-bit key for each round. Functions of KASUMI are FI, FO and FL functions performed by them are completely different from each other and they use key values for doing their operations. The key values are KI for FI function, KO for FO function and, KL for FL function for all 8 rounds. Figure 1 provides Fiestel structure of 8-round KASUMI algorithm [2]. For odd numbered rounds, function FL comes before FO, FI functions while in even numbered rounds, FO, FI comes before FL function. Brief description of three functions of algorithm is given below along with key-value operations done in them.

2.1 FL Function

This function has two rounds of operation as shown in Figure 1 It takes 32-bit input and performs operation on it using 32-bit key, KL, to produce 32-bit output. Key, KL, is divided in two 16-bit values for each of the two rounds as shown below:

$$\mathbf{KL}_{\mathrm{i}} = \mathbf{KL}_{\mathrm{i},1} \| \mathbf{KL}_{\mathrm{i},2}$$

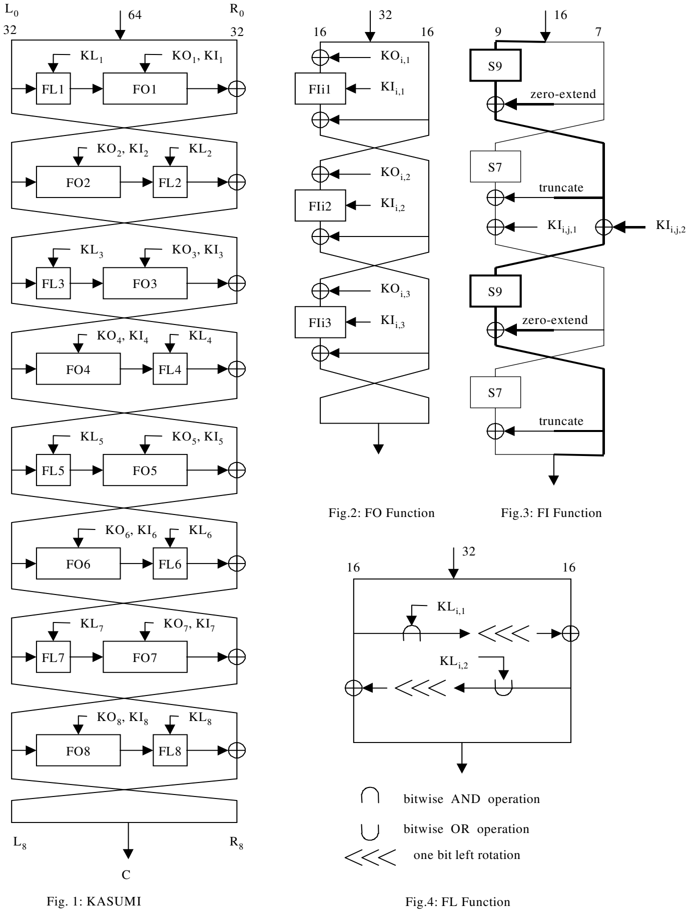

Figure 1: The original KASUMI algorithm, FO, FI and FL functions.

Input is divided in two 16-bit values, **L** (left) and **R** (right) as given below:

$$\begin{aligned}\mathbf{R}^{'} &= \mathbf{R} \oplus ROL(\mathbf{L} \cap \mathbf{KL}_{\mathrm{i},1}) \\ \mathbf{L}^{'} &= \mathbf{L} \oplus ROL(\mathbf{R} \cap \mathbf{KL}_{\mathrm{i},2})\end{aligned}$$

$\mathbf{R}^{'}$ and $\mathbf{L}^{'}$ are each 16-bit values obtained after performing operations on **R** and **L**. These values obtained are then concatenated to make 32-bit output.

2.2 FO Function

This function has three rounds of operation as shown in Figure 1. It first takes 32-bit input, divides it in two equal 16-bit values and, performs operation on it using 48-bit key KO and, another 48-bit sub-key KI used in FI function. Figure for FO function is shown in Figure 1. The 48-bit sub-keys are subdivided into three 16-bit sub-keys where:

$$\begin{aligned}\mathbf{KO}_{\mathrm{i}} &= \mathbf{KO}_{\mathrm{i},1} \| \mathbf{KO}_{\mathrm{i},2} \| \mathbf{KO}_{\mathrm{i},3} \\ \mathbf{KI}_{\mathrm{i}} &= \mathbf{KI}_{\mathrm{i},1} \| \mathbf{KI}_{\mathrm{i},2} \| \mathbf{KI}_{\mathrm{i},3}\end{aligned}$$

For each integer j (number of rounds within FO) with $1 \leq \mathrm{j} \leq 3$, $\mathbf{R}_{\mathrm{j}}$ and $\mathbf{L}_{\mathrm{j}}$ are given as:

$$\begin{aligned}\mathbf{R}_{\mathrm{j}} &= FI(\mathbf{L}_{\mathrm{j}-1} \oplus \mathbf{KO}_{\mathrm{i,j}}, \mathbf{KI}_{\mathrm{i,j}}) \oplus \mathbf{R}_{\mathrm{j}-1}\\ \mathbf{L}_{\mathrm{j}} &= \mathbf{R}_{\mathrm{j}-1}\end{aligned}$$

We then, return the 32-bit concatenated value obtained $(\mathbf{L}_3 \| \mathbf{R}_3)$ after completion of FO function.

2.3 FI Function

This function involves four rounds of operation. FI function takes 16-bits of input data and gives 16-bits of output. Data is split into 9-bit, say $\mathbf{L}_0$, which goes to left side of FI function and, 7-bit value, say $\mathbf{L}_0$, which goes to right side of FI function. Two **S**-boxes used are **S9** box used when 9-bit data is operated and, **S7** box used when 7-bit data is operated. FI-box is given in Figure 1. Truncation of 9-bit data done by removing 2 most significant bits when it is operated with 7-bit data and, 7-bit data is zero padded by adding two zeros as most significant bits when it is operated with 9-bit data. We define the following series of operations:

$$\begin{aligned}\mathbf{L}_1 &= \mathbf{R}_0,\\ \mathbf{R}_1 &= \mathbf{S9}[\mathbf{L}_0] \oplus \mathbf{ZE}(\mathbf{R}_0),\\ \mathbf{L}_2 &= \mathbf{R}_1 \oplus \mathbf{KI}_{\mathrm{i,j},2},\\ \mathbf{R}_2 &= \mathbf{S7}[\mathbf{L}_1] \oplus \mathbf{TR}(\mathbf{R}_1) \oplus \mathbf{KI}_{\mathrm{i,j},1},\\ \mathbf{L}_3 &= \mathbf{R}_2,\\ \mathbf{R}_3 &= \mathbf{S9}[\mathbf{L}_2] \oplus \mathbf{ZE}(\mathbf{R}_2),\\ \mathbf{L}_4 &= \mathbf{S7}[\mathbf{L}_3] \oplus \mathbf{TR}(\mathbf{R}_3),\\ \mathbf{R}_4 &= \mathbf{R}_3\end{aligned}$$

The function returns the 16-bit value $(\mathbf{L}_4 \| \mathbf{R}_4)$. The Figure 1 shows the structure of KASUMI [2] along with three functions used in it.

3 Overview of SNOW-3G

SNOW-3G is word-oriented cipher algorithm which performs operations by using 128-bit key and, 128-bit Initialization Vector (IV). It has two security modules, LFSR (Linear Feedback Shift Register) and FSM [9]. The LFSR is made up of 16 s-blocks each having the capacity to hold 32-bit data and, the feedback is defined by a primitive polynomial over the finite field GF (2^{32}). The second module is FSM which consists of three registers **R1**, **R2** and **R3** and it perform functions by using two substitution boxes **S1** and **S2**. The algebraic operations used in the FSM are EX-OR and addition modulo 2^{32}.

The SNOW-3G algorithm is working under two different modes, one is initialization mode and other is key-stream generation mode. In initialization mode, 32-bit output **F** is generated and it is discarded at the beginning. After that, the algorithm goes into key-stream mode generation and produces 32-bit output called as **F**. The **F** value generated from key-stream is used in the feedback part of LFSR as shown in Figure 2 Various 32-bit values of the key-stream are generated which are then used for encryption of different input data. Brief description about working of LFSR and, FSM is given below.

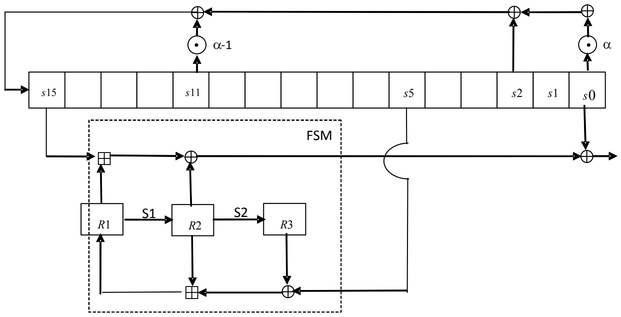

Figure 2: The SNOW-3G algorithm [9]

3.1 LFSR

It has 16 **s**-boxes $\mathbf{s}_0, \mathbf{s}_1, ..., \mathbf{s}_{15}$ and each box having the holding capacity of 32-bit. As seen from Figure 2, the feedback values are taken only from $\mathbf{s}_0, \mathbf{s}_2$ and $\mathbf{s}_{11}$ boxes and given as feedback to the $\mathbf{s}_{15}$ after some mathematical feedback computations that are MUL_α and DIV_α.

3.2 FSM

The FSM takes two input values from LFSR which are $\mathbf{s}_5$ and $\mathbf{s}_{15}$ and it produces 32-bit output word **F** defined as follows:

$$\mathbf{F} = (\mathbf{s}_{15} \boxplus \mathbf{R1}) \oplus \mathbf{R2}$$

The registers (**R1**, **R2** and **R3**) are updated with the inflow of new values. Intermediate value **r** is calculated before getting in operation with register **R1** and so, it is given as:

$$\mathbf{r} = \mathbf{R2} \boxplus (\mathbf{R3} \oplus \mathbf{s}_5)$$

The values corresponding to the Registers **R3**, **R2**, **R1** are computed in the following way:

$$\begin{aligned}\mathbf{R3} &= \mathbf{S2}(\mathbf{R2})\\ \mathbf{R2} &= \mathbf{S1}(\mathbf{R1})\\ \mathbf{R1} &= \mathbf{r}\end{aligned}$$

4 Proposed Work on KASUMI

In this proposed design, the FI function is completely removed from all 8 rounds of KASUMI. But the keys that are used in FI function are retained. The three 16-bit keys of FI function used in each round of KASUMI are now used differently. For a particular round, three KI

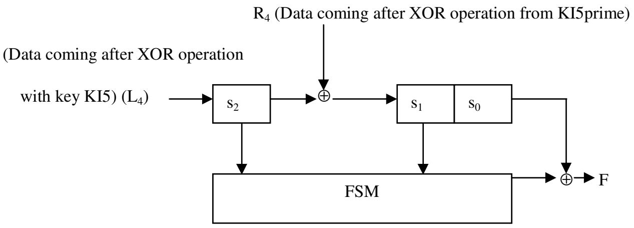
R4 (Data coming after XOR operation from KI5prime)
(Data coming after XOR operation
with key KI5) (L4)
S2
S1
S0
FSM
F

Figure 3: Structure used at end of the 4th round of KASUMI

keys, $\mathbf{KI}_{i1}$, $\mathbf{KI}_{i2}$ and $\mathbf{KI}_{i3}$ are used. The first two 16-bit KI sub-keys, $\mathbf{KI}_{i1}$ and $\mathbf{KI}_{i2}$ are concatenated to form a 32-bit key. Similarly, $\mathbf{KI}_{i2}$ and $\mathbf{KI}_{i3}$ are concatenated to form another 32-bit key, i.e.,

$$\begin{aligned}\mathbf{KI}_1 &= \mathbf{KI}_{i1}\|\mathbf{KI}_{i2}\\ \mathbf{KI}_2 &= \mathbf{KI}_{i2}\|\mathbf{KI}_{i3}\end{aligned}$$

The same operation is performed with KI keys of the other rounds of KASUMI to form two separate 32-bit key. Final key values are given below and their implementation is shown in Figure 4 where 'n' denotes n^{th} round value,

$$\begin{aligned}\mathbf{KI3} &= (\mathbf{KI}_{i1}[n]\|\mathbf{KI}_{i2}[n]) \oplus\\ &\qquad (\mathbf{KI}_{i1}[n+2]\|\mathbf{KI}_{i2}[n+2])\\ \mathbf{KI3}_{\text{prime}} &= (\mathbf{KI}_{i2}[n]\|\mathbf{KI}_{i3}[n]) \oplus\\ &\qquad (\mathbf{KI}_{i2}[n+2]\|\mathbf{KI}_{i3}[n+2])\end{aligned}$$

Similarly, other sets of 32-bit key values are defined as follows:

$$\begin{aligned}\mathbf{KI5} &= (\mathbf{KI}_{i1}[n+3]\|\mathbf{KI}_{i2}[n+3]) \oplus\\ &\qquad (\mathbf{KI}_{i1}[n+6]\|\mathbf{KI}_{i2}[n+6])\\ \mathbf{KI5}_{\text{prime}} &= (\mathbf{KI}_{i2}[n+3]\|\mathbf{KI}_{i3}[n+3]) \oplus\\ &\qquad (\mathbf{KI}_{i2}[n+6]\|\mathbf{KI}_{i3}[n+6]).\\ \mathbf{KI7} &= (\mathbf{KI}_{i1}[n+1]\|\mathbf{KI}_{i2}[n+1]) \oplus\\ &\qquad (\mathbf{KI}_{i1}[n+4]\|\mathbf{KI}_{i2}[n+4])\\ \mathbf{KI7}_{\text{prime}} &= (\mathbf{KI}_{i2}[n+1]\|\mathbf{KI}_{i3}[n+1]) \oplus\\ &\qquad (\mathbf{KI}_{i2}[n+4]\|\mathbf{KI}_{i3}[n+4]).\\ \mathbf{KI8} &= (\mathbf{KI}_{i1}[n+5]\|\mathbf{KI}_{i2}[n+5]) \oplus\\ &\qquad (\mathbf{KI}_{i1}[n+7]\|\mathbf{KI}_{i2}[n+7])\\ \mathbf{KI8}_{\text{prime}} &= (\mathbf{KI}_{i2}[n+5]\|\mathbf{KI}_{i3}[n+5]) \oplus\\ &\qquad (\mathbf{KI}_{i2}[n+7]\|\mathbf{KI}_{i3}[n+7]).\end{aligned}$$

It is well known that 8 rounds are involved in KASUMI. Let's say 0th round is the first round of KASUMI. If $[n+(integer)] > 7$, then MSB will be masked and value corresponding to binary value is obtained. Then the obtained value will be considered for further operations (for example, the binary value 110 gets changed to the binary value 010 after masking MSB). Two n^{th} round values are required for producing 32-bit KI key and it will be chosen as per the procedure given in the above eqnarray*s. These KI keys are used as corresponding round values.

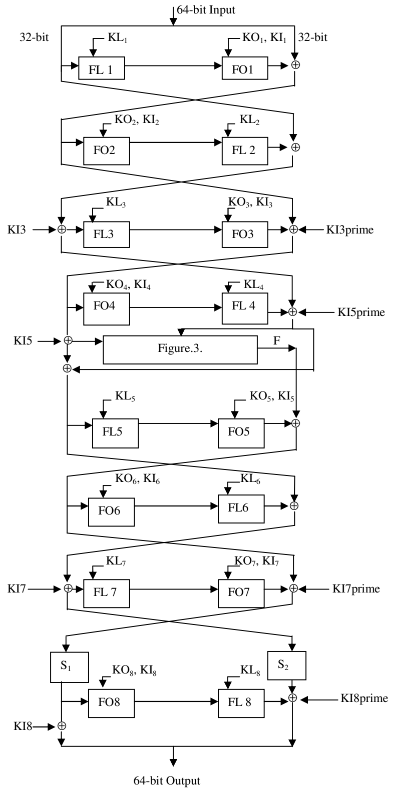

Figure 4: Proposed modified KASUMI algorithm (using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ at the 7th round)

In proposed design, in addition to making changes in KI key, small change in second KL-key produced by KL function is also done in a proposed design. This change is done only in last and first rounds. As suggested in reference [15], KL key value used in the 8th round of the KASUMI is weak and it is having some bits as same as that of the KL key value that used in the second round of the KASUMI. The chances for the attack can be extended to more rounds of the algorithm if the attack has been done already in the 6th or 7th rounds of the KASUMI. The changes in the KL function is introduced only in the first and last rounds of the KASUMI. The **S3** box is used to changing KL key value of 2nd stage of FL function that present in the first and last rounds. This S-box is same as that used in SNOW-3G (32-bit Rijndael's S-box) but it performs the 16-bit data operation as the length of the key is 16-bit.

Another important change that made in the proposed design is the insertion of FSM machine along with three shift registers and each of it is having the capacity of holding 32-bits data as is shown in Figure 3 The functioning of FSM is same as in SNOW-3G. But the SNOW-3G is basically well known for key-stream generation but in our case it is used different purpose. The FSM is used only after the end of 4th round, as shown in Fig.7 and Fig.8. The output which is generated from FSM for the first time will be discarded. After discarding this first value from

FSM, next output bits generated will be used as an input for next round of the KASUMI. In purposed design the FSM is initialized for only one time rather than 32 times as done in SNOW-3G algorithm. The input to FSM is given through two shift registers that are $\mathbf{s}_1$ and $\mathbf{s}_2$. The left input value from the end of fourth round is entered into $\mathbf{s}_2$. This $\mathbf{s}_2$ is then EX-ORed with right input value that coming from the end of the fourth round. This value produced is then passed into $\mathbf{s}_1$ shift register followed by $\mathbf{s}_0$. Fig.6. illustrates the the above explanations.

The 32-bit value of $\mathbf{F}_1$ is generated from FSM and it is EX-ORed with $\mathbf{s}_0$ to produce the final output $\mathbf{F}$. This $\mathbf{F}$ is referred as a right input data ($\mathbf{R}_5$) for next round (i.e. 5th round) of the KASUMI. The left 32-bit input data ($\mathbf{L}_5$) is r 4th round of the KASUMI.

$$\begin{aligned}\mathbf{L}_5 &= \mathbf{L}_4 \oplus \mathbf{R}_4 \\ \mathbf{R}_5 &= \mathbf{F} = (\mathbf{F}_1)(32 - bitoutoutfromFSM) \oplus \mathbf{s}_0\end{aligned}$$

Third modification which have been done in the proposed design is the insertion of $\mathbf{S}_1$ and $\mathbf{S}_2$ boxes which are based on Rijndael's $\mathbf{S}_\mathrm{R}$ and $\mathbf{S}_\mathrm{Q}$ box, respectively. The functionality of the $\mathbf{S}_1$ and $\mathbf{S}_2$ boxes are same as in SNOW-3G. But these boxes are used only after the end of the 7th round. It considers two different cases, one is the use of both the $\mathbf{S}_1$ and $\mathbf{S}_2$ boxes while other use only the $\mathbf{S}_1$ box at the end of the 7th round which are illustrated in Fig.7 and Fig8, respectively. Since the $\mathbf{S}_2$ box is not strong as $\mathbf{S}_1$ box, it considered to use both the $\mathbf{S}_1$ and $\mathbf{S}_2$ boxes [8]. The 32-bit output data from left side is EX-ORed with '**KI8**' key value and the 32-bit right side data is EX-ORed with **KI8** prime' key to obtain the final 32-bit left and 32-bit right data. This is illustrated in Figure 4 and Figure 5 As given in [15], the last two rounds are observed to be weaker. In order to strong, the nonlinearity output can be produced with the help of **S**-box that can be leads to increase the complexity for the attacker.

The FSM is implemented after the 4th round, but this produced more delay in the encryption process. If the FSM is implement before 4th round, then also no difference in encryption time. Our focus was to merely propose modifications in KASUMI to make it more robust and also it consumes approximately the same time for encryption compared to standard KASUMI.

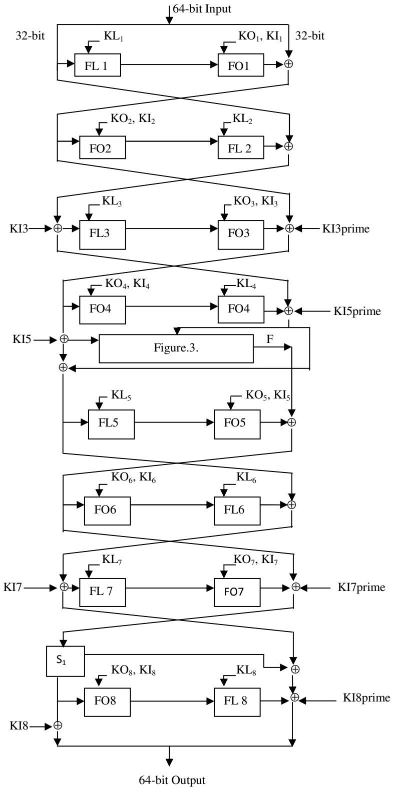

Figure 5: Proposed modified KASUMI algorithm (using only $\mathbf{S}_1$ at the 7th round)

5 Simulation Results

The proposed algorithm was tested using National Institute of Standards and Technology (NIST) [14] statistical test suite where all 15 tests were passed. This suite is used for testing of randomness produced in designed/modified algorithm. The encryption time required by the modified algorithm indicates an increase by few seconds compared to original KASUMI. Comparison is done based on the 'No. of iterations' which means number of times algorithm remains in operational mode continuously before producing final output value. There are Federal Information Processing Standard (FIPS) or Standardization Administration of China (SAC) standards also used in previous works as alternate standard for testing an algorithm. In this proposed design, a large array of numbers at a single time is given to find out an approximate time taken when the number of iterations are entered. The testing was based on the software (MobaXtermlinux application) and it done on Intel (R) Core(TM) i7-4770 CPU @ 3.4GHz.

As one can be seen from the simulation results and the tables presented in this paper, the performance improvement of the proposed modified KASUMI design is compared with the conventional KASUMI in terms both the encryption speed and encryption time taken. In addition to that, it also present the simulation result for comparing the two different proposed design based on different usage of the **S**-boxes ($\mathbf{S}_1$ box alone and combined $\mathbf{S}_1$ and $\mathbf{S}_2$). The Table 1 illustrate the values for encryption speed and encryption time taken for various number of numbers entered for the proposed modified KASUMI algorithm (using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ box at the 7th round). And Table 2 illustrate the values for encryption speed and encryption time taken for various number of numbers entered for the proposed modified KASUMI algorithm (using only $\mathbf{S}_1$ box at the 7th round). And Table 3 illustrate the values for encryption speed and encryption time taken for various number of numbers entered for the conventional KASUMI algorithm.

First, Figure 6 shows the performance comparisons of the conventional KASUMI with that of the proposed modified KASUMI (using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ at the 7th round) in terms of encryption time taken Vs number of numbers entered. The main purpose of this simulation to show the performance in terms of encryption time taken of the proposed modified KASUMI is almost identical to conventional KASUMI. Figure 7 shows the performance comparisons of the conventional KASUMI with that of the proposed modified KASUMI (using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ box at the 7th round) in terms of encryption speed Vs Number of numbers entered. As can be seen from the Figure 7, the proposed modified KASUMI speed is reduced for higher values of number of numbers entered compared to the conventional KASUMI. Figure 8 and Figure 9 also shows the similar type of performance comparisons as in Figure 6 and Figure 7 but for the case proposed modified KASUMI (using only $\mathbf{S}_1$ at the 7th round). Again, the performance degradation of our proposed design over the conventional design is clearly observed from Figure 8 and Figure 9.

Figure 10 examines the performance comparison of the proposed modified KASUMI using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ box at the 7th round Vs proposed modified KASUMI using only $\mathbf{S}_1$ box at the 7th round in terms of encryption time taken Vs number of numbers entered. From this result, we can observe that while only the $\mathbf{S}_1$ at the 7th round is used, the time taken for the for encryption is reduced by 1 to 2 seconds for the cases of $5 * 10^7$ to $1 * 10^8$ iterations. Figure 11 shows the proposed modified KASUMI using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ box at the 7th round Vs proposed modified KASUMI using only $\mathbf{S}_1$ box at the 7th round in terms of encryption speed Vs number of numbers entered. As can be seen from the Figure 11, the encryption speed of the proposed modified KASUMI using only $\mathbf{S}_1$ box at the 7th is increased by 20 to 60 Kbps for the cases of $5 * 10^7$ to $1 * 10^8$ iterations. Though, the encryption speed of the proposed modified KASUMI using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ box at the 7th round would get reduced little, but it can be acceptable to have a delay of 1 to 2 by achieving more resistance against attacks.

Table 1: proposed modified KASUMI algorithm (using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ box at the 7th round)

No. of iterations	Encry. time taken in sec.	Encry. speed in bps
$1 * 10^6$ to $2 * 10^6$	1	2000000.00
$4 * 10^6$	2	2000000.00
$6 * 10^6$	3	2000000.00
$8 * 10^6$	4	2000000.00
$1 * 10^7$	5	2000000.00
$2 * 10^7$	11	1818181.82
$4 * 10^7$	22	1769132.49
$5 * 10^7$	28	1785714.29
$6 * 10^7$	34	1764705.88
$7 * 10^7$	40	1750000.00
$8 * 10^7$	46	1739130.43
$9 * 10^7$	51	1764705.88
$1 * 10^8$	57	1754385.96

Table 2: proposed modified KASUMI algorithm (using only $\mathbf{S}_1$ box at the 7th round)

No. of iterations	Encry. time taken in sec.	Encry.n speed in bps
$1 * 10^6$ to $2 * 10^6$	1	2000000.00
$4 * 10^6$	2	2000000.00
$6 * 10^6$	3	2000000.00
$8 * 10^6$	4	2000000.00
$1 * 10^7$	5	2000000.00
$2 * 10^7$	11	1818181.82
$4 * 10^7$	22	1818181.82
$5 * 10^7$	27	1821851.85
$6 * 10^7$	33	1818181.82
$7 * 10^7$	38	1832105.26
$8 * 10^7$	44	1818181.82
$9 * 10^7$	49	1826734.69
$1 * 10^8$	55	1818181.82

6 Conclusion

In this paper,it is proposed the use the FSM machine, Rijndael's SR box and SQ box in the existing KASUMI algorithm to make it as more robust against attacks. Also it is proposed to modify the KL keys by using SR box. And this changes are is done in the KL keys which are corresponding to the 1st and 8th. I addition to that, it is also proposed to use KI keys in different ways by using Ex-OR and concatenation operations in t^{th} and 8th as explained in this paper. From the simulation results, it observed that the proposed modified KASUMI algorithm will take more encryption time compared to standard KASUMI algorithm, but the proposed modified KASUMI algorithm reduce linear and differential attacks in KASUMI.

Table 3: Original conventional KASUMI algorithm

No. of iterations	Encry. time taken in sec.	Encry.speed in bps
$1 * 10^6$ to $2 * 10^6$	1	2000000.00
$4 * 10^6$	2	2000000.00
$6 * 10^6$	3	2000000.00
$8 * 10^6$	4	2000000.00
$1 * 10^7$	5	2000000.00
$2 * 10^7$	10	2000000.00
$4 * 10^7$	20	2000000.00
$5 * 10^7$	26	2000000.00
$6 * 10^7$	30	1973076.92
$7 * 10^7$	35	2000000.00
$8 * 10^7$	40	2000000.00
$9 * 10^7$	45	2000000.00
$1 * 10^8$	51	1970784.31

References

[1] 3GPP TR 55.919 V6.1.0, *Specification of the A5/3 Encryption Algorithms for GSM and ECSD, and the GEA3 Encryption Algorithm for GPRS*, Technical Report 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Document 4: Design and evaluation Report (Release 6), 2002-2012.

[2] 3GPP TS 35.202 V14.0.0, *Specifications of the 3GPP Confidentiality and Integrity Algorithms*, Technical

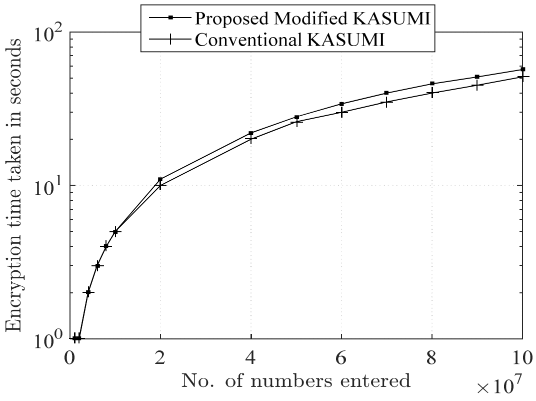

Figure 6: Performance comparisons of the conventional KASUMI Vs the proposed modified KASUMI (using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ at the 7th round) in terms of encryption time taken Vs number of numbers entered.

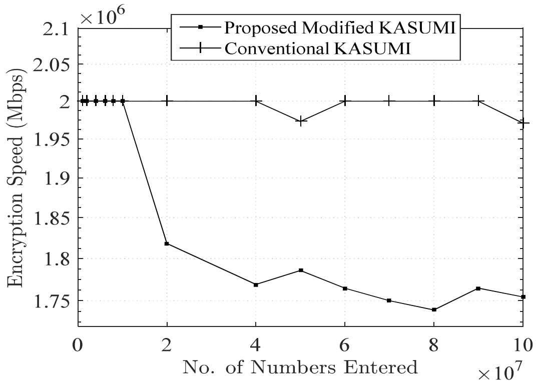

Figure 7: Performance comparisons of the conventional KASUMI Vs the proposed modified KASUMI (using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ at the 7th round) in terms of encryption speed Vs number of numbers entered.

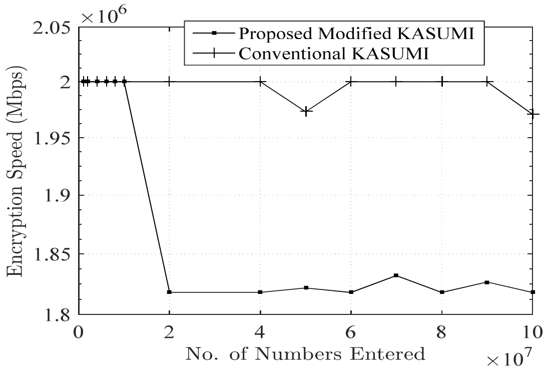

Figure 8: Performance comparisons of the conventional KASUMI Vs the proposed modified KASUMI (using only $\mathbf{S}_1$ at the 7th round) in terms of encryption time taken Vs number of numbers entered.

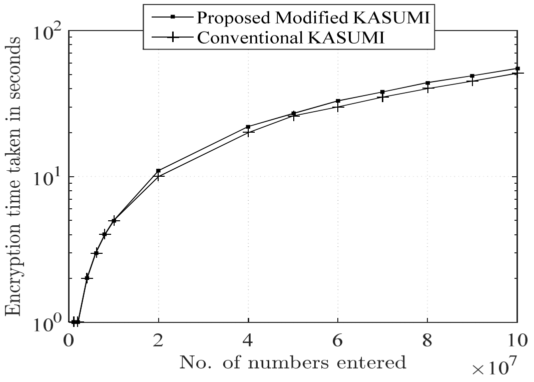

Figure 9: Performance comparisons of the conventional KASUMI Vs the proposed modified KASUMI (using only $\mathbf{S}_1$ at the 7th round) in terms of encryption speed Vs number of numbers entered.

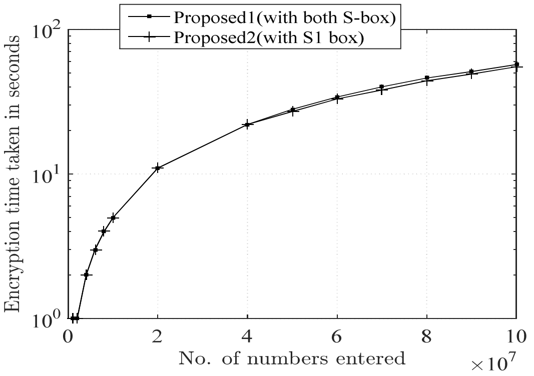

Figure 10: Proposed modified KASUMI using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ box at the 7th round Vs proposed modified KASUMI using only $\mathbf{S}_1$ box at the 7th round in terms of encryption time taken Vs number of numbers entered.

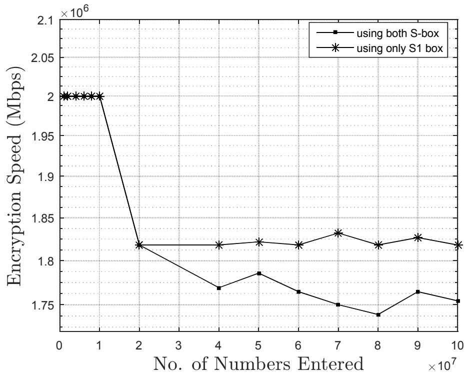

Figure 11: Proposed modified KASUMI using combined $\mathbf{S}_1$ and $\mathbf{S}_2$ box at the 7th round Vs proposed modified KASUMI using only $\mathbf{S}_1$ box at the 7th round in terms of encryption speed Vs number of numbers entered.

Report 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Document2: KASUMI Specifications V14.0.0, 2017.

[3] E. Biham, O. Dunkelman, N. Keller, "The rectangle attack - Rectangling the serpent," in *Proceedings of the EUROCRYPT*, Lecture Notes in Computer Science, pp. 340–357, Springer, May 2001.

[4] E. Biham, O. Dunkelman, N. Keller, "Related-key rectangle attack on full kasumi," in *Proceedings of the 11th International Conference on Theory and Application of Cryptology and Information Security*, pp. 443–461, Dec. 2005.

[5] C. S. Chen, X. Yu, Y. X. Xiang, X. Li, T. Li, "An improved DPA attack on DES with forth and back random round algorithm," *International Journal of Network Security*, vol. 19, no. 2, pp. 285-294, 2017.

[6] K. Chetioui, G. Orhanou, S. El Hajji, "New protocol e-DNSSEC to enhance DNSSEC security," *International Journal of Network Security*, vol. 20, no. 1, pp. 19-24, 2018.

[7] H. Choudhury, B. Roychoudhury, D. Kr. Saikia, "Security extension for relaxed trust requirement in non-3GPP access to the EPS," *International Journal of Network Security*, vol. 18, no. 6, pp. 1041-1053, 2016.

[8] ETSI/SAGE, *Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 and UIA2*, Technical Report Document 5: Design and Evaluation Report, Ver. 1.0, 2006.

[9] ETSI/SAGE, *Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 and UIA2*, Technical Report Document 2: SNOW 3G Specification, Ver. 1.1, 2006.

[10] L. He, C. Yuan, X. Hu, and Z. Qin, "An effcient and provably secure certificateless key insulated encryption with applications to mobile internet," *International Journal of Network Security*, vol. 19, no. 6, pp. 940-949, 2017.

[11] N. Keller, O. Dunkelman, A. Shamir, "A practical-time attack on the A5/3 cryptosystem used in third generation GSM telephony," *IACR Cryptology Eprint archive*, 2010.

[12] T. Saito, "A single-key attack on 6-round KASUMI," *IACR Cryptology Eprint archive*, 2011.

[13] M. Styugin, "Establishing systems secure from research with implementation in encryption algorithms," *International Journal of Network Security*, vol. 20, no. 1, pp. 35-40, 2018.

[14] Z. Wang, X. Dong, K. Jia, J. Zhao, "Differential fault attack on kasumi cipher used in gsm telephony," *Mathematical Problems in Engineering*, vol. Article ID 251853, pp. 7, 2014.

[15] W. Yi, S. Chen, "Multidimensional zero-correlation linear cryptanalysis of the block cipher KASUMI", *IET Information Security*, vol. 10, no. 4, pp. 215-221, 2016.

[16] H. Zhu, Y. Zhang, and Y. Sun, "Provably secure multi-server privacy-protection system based on Chebyshev chaotic maps without using symmetric cryptography," *International Journal of Network Security*, vol. 18, no. 5, pp. 803-815, 2016.

Biography

Raja Muthalagu received his Ph.D. in Wireless Communication from National Institute of Technology (NIT), Tiruchirappalli, India in 2014. He joined the Department of Electrical and Electronics Engineering, BITS, Pilani, Dubai Campus, in 2015, where he is currently a full Assistant Professor. His research interests include orthogonal frequency division multiplexing (OFDM), multiple-input and multiple-output (MIMO) systems, and network security.

Subeen Jain received hid B.E. (Honors) in Electronics and Communications engineering from BITS-Pilani Dubai campus. His areas of interest include security algorithms mainly related to mobile security and networking and, areas related to telecommunications.