FPGA implementation of an optimized A5/3 encryption algorithm

Mahdi Madani, Camel Tanougast

FPGA Implementation of an Optimized A5/3 Encryption Algorithm

Mahdi Madani

Laboratory ImViA, University of Bourgogne Franche-Comté , Dijon, France
Email:mmadani49@gmail.com

Camel Tanougast

Laboratory LCOMS, University of Lorraine, Metz, France
Email:camel.tanougast@univ-lorraine.fr

Abstract

A radio link connecting users to network services is one of the most sensitive parts of mobile networks. This wireless channel is not protected physically to prevent unauthorized access to the carried information. Therefore, network providers use a security mechanism mainly based on cryptographic algorithms. For example, data protection (confidentiality) in the second and third generations of mobile networks is ensured using the A5/3 encryption algorithm (f8 function) standardized by the Third Generation Partnership Project (3GPP). In this work, we defined two main objectives for obtaining an optimized architecture of the A5/3 algorithm. The first one focuses on the optimization of the algorithm's kernel (the KASUMI block cipher) by simplifying its internal architecture. The second one aims at the optimization of the A5/3 algorithm using a single block of the simplified KASUMI, unlike the standard A5/3 algorithm based on five blocks of the basic KASUMI. As a result, good performance has been achieved by considering the tradeoff between high throughput and required hardware logic resources compared to previous works. The proposed architecture was implemented on several Xilinx Virtex Field Programmable Gate Arrays (FPGA) technology devices. The synthesis results obtained after place and route have demonstrated the feasibility and efficiency of our solution. This promising technique can be applied to provide real-time data protection on embedded applications of mobile networks.

Keywords: A5/3 encryption algorithm, KASUMI block cipher, Mobile security, Hardware optimization, FPGA implementation, Logic synthesis.

1. Introduction

Nowadays, mobile networks are widely used in our lives where much of our personal information is transmitted over weak radio links connecting users to

network services. As a result, all transmitted data are encrypted to ensure confidentiality and integrity. For example, the A5/1 and A5/2 stream ciphers are the first encryption algorithms used in mobile networks (second generation). They where cryptanalyzed a few years later [1–4] and replaced by the A5/3 algorithm which was also cryptanalyzed [5]. However, considering that the A5/3 algorithm is until widely used in Universal Mobile Telecommunication Systems (UMTS) networks (to perform the f8 confidentiality and f9 integrity functions), investigations are still proposed to improve the security optimization of the KASUMI algorithm [6–8]. Therefore, A5/3 optimization is yet very relevant. The technical specifications of this algorithm are standardized by the Third Generation Partnership Project (3GPP). The Quality of Services (QoS) is an important parameter considered by mobile networks. It is required to ensure communication in real-time (high throughput) using an embedded application (mobile phone). Therefore, it is necessary to implement the complex architecture of the encryption algorithm using a design that ensures low logical hardware resources and high throughput, which is the purpose of this work.

Our study consists to propose an optimized architecture of the A5/3 encryption algorithm used in the Global System for Mobile communications (GSM) and UMTS networks to ensure data confidentiality. This paper is an improvement of the work presented in the IEEE-2017 7th International Conference on Circuits, System and Simulation (ICCSS 2017) [9].

To achieve our objective, we divided the work into two main parts. In the first part, we proposed an optimized architecture of the KASUMI block cipher by simplifying its internal structure. We started to combine the initial S7 and S9 Substitution Boxes (S-boxes) in one Global S-Box (GS). Then, we replaced the initial FI function built upon S7 and S9 by a simplified FI' function based on the proposed GS. After that, we replaced the initial FO function founded on the initial FI function by a simplified FO' function based on the proposed FI' function. Finally, we used two initial FL function blocks and two simplified FO' function blocks to form the optimized KASUMI block cipher. In the second part, we proposed an optimized A5/3 algorithm based on the proposed optimized KASUMI to ensure the same functionalities (produce 228-bits each 4.615 ms [10]) of the original A5/3 algorithm made up five components of the original KASUMI block cipher [11, 12]. In this study, we used the synthesis architectural technique corresponding to the reuse by factorization of logic/arithmetic operators [13]. This technique allows the connection between different internal components which are managed by a tailored control unit based on Moore's Finite State Machine (FSM).

To illustrate the performance improvement, the proposed architecture was implemented in several Field Programmable Gate Arrays (FPGA) technology devices, and a comparison with previous works was performed. The synthesis implementation results show the good performance of the proposed solution in terms of throughput and hardware logic resources. Indeed, the proposed architecture is still able to ensure data confidentiality in real-time communications and is also able to minimize the required area for its implementation in mobile phones.

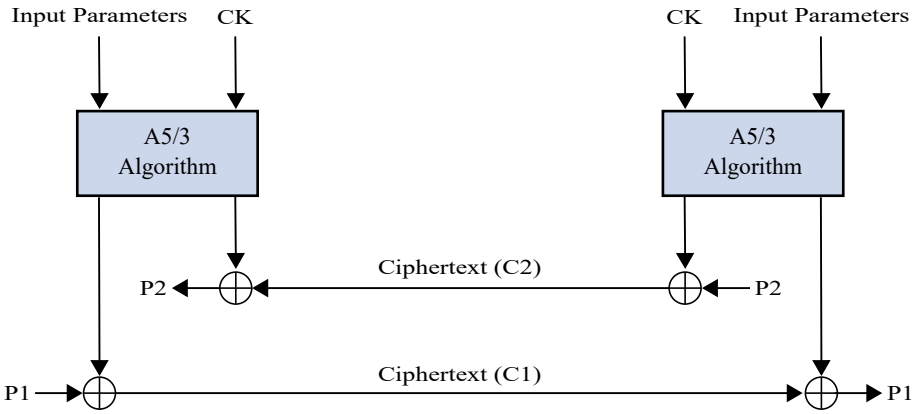


Figure 1: Mobile networks encryption/decryption.

The rest of this paper is organized as follows. We present briefly in Section 2 the original architecture of the A5/3 encryption algorithm. Section 3 is devoted to describing the optimized KASUMI block cipher, which forms the kernel of the principal algorithm. The proposed A5/3 architecture and its processing steps are presented in Section 4. The hardware implementation results in terms of internal functions and FPGA logic resources are presented in Section 5. Section 6 gives a synthetic discussion and performance comparison with previous works. Finally, conclusion and future work are given in Section 7.

2. A5/3 encryption algorithm overview

A5/3 is an encryption algorithm designed for data protection in mobile networks (GSM and UMTS networks) where data are transmitted in real-time [14, 15]. The communications between users and networks are synchronized in the uplink (mobile to the network) and downlink (network to mobile) directions. Therefore, the A5/3 algorithm treats two sequences at the same time. It generates two keystream sequences of size 144-bits under the control of the Cipher Key (CK) of 64-bits size and the input variable ($COUNT$) of 22-bits size [9]. The first generated keystream sequence is combined with the plaintext (P1) using a XOR operation to form the transmitted ciphertext (C1), and the second sequence is combined with the received ciphertext (C2) to recover the plaintext (P2). At the reception, the same process is used. It is based on symmetric encryption process as illustrated in Figure 1 [16].

The internal architecture of the A5/3 algorithm is based on five KASUMI components, as standardized by 3GPP (see Figure 2). To generate output keystream, it uses CA, CB, CC, CD, CE initialization standardized parameters (see Table 1), CK, and the fixed Key Modifier (KM) as inputs.

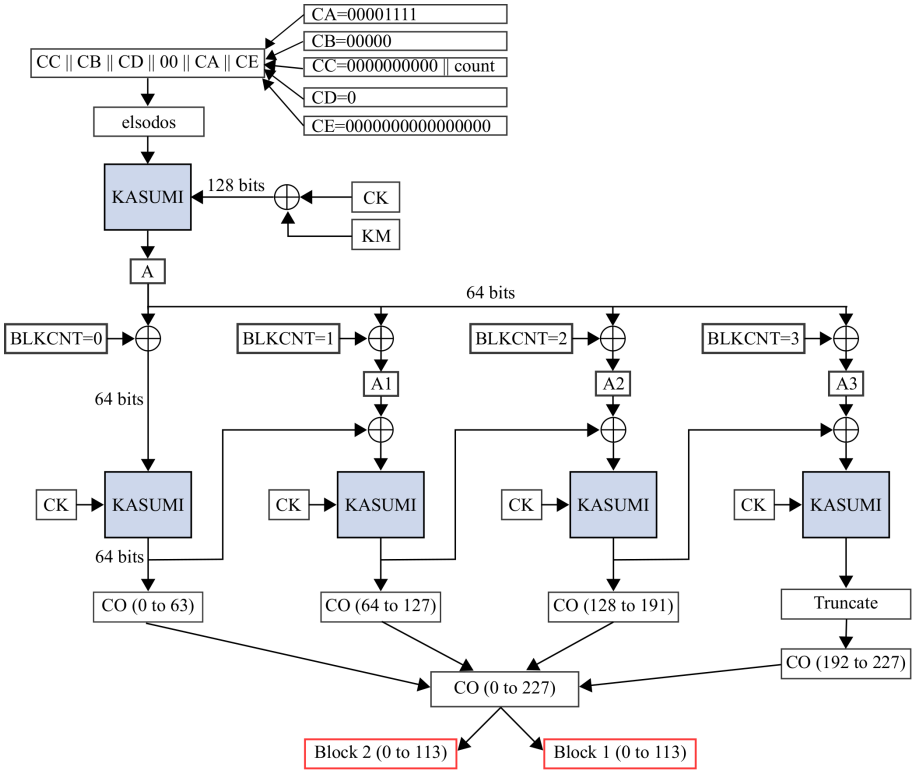


Figure 2: The A5/3 internal architecture [15].

	A5/3
CA	00001111
CB	00000
CC	INPUT
CD	DIRECTION
CE	0000000000000000
CK	Cipher Key repeated to fill 128-bits
CO	Block1/Block2 (114-bits/114-bits)
KM	0x55555555555555555555555555555555

Table 1: Standardized parameters for the A5/3 encryption algorithm.

3. KASUMI block cipher

This section presents an overview of the standard KASUMI block cipher and details the architecture of the optimized KASUMI block cipher used in order to enhance the implementation performance of the A5/3 encryption algorithm.

3.1. Standard KASUMI block cipher

KASUMI is a block cipher standardized by 3GPP standards [11]. It is based on the optimization of the $MISTY1$ block cipher [17]. It was designed by the Security Algorithms Group of Experts (SAGE), part of the group of European standards of the European Telecommunications Standards Institute (ETSI), which is the organism of the European normalization in telecommunications domain [12]. It is used to ensure the confidentiality and integrity of the user's data over GSM and UMTS networks [18–20]. Its architecture is formed by a Feistel structure of eight rounds (see Figure 3). At each round, it executes two main FL (see Figure 3(d)) and FO (see Figure 3(b)) functions to generate output blocks of 64-bits size under the CK control of 128-bits size. The FO and FL functions are executed in reverse order between odd and even rounds, as illustrated in the detailed KASUMI architecture presented in Figure 3(a). Our first objective in this work is to implement this complex architecture requiring many logical resources while maintaining good performance.

3.2. Optimized KASUMI block cipher

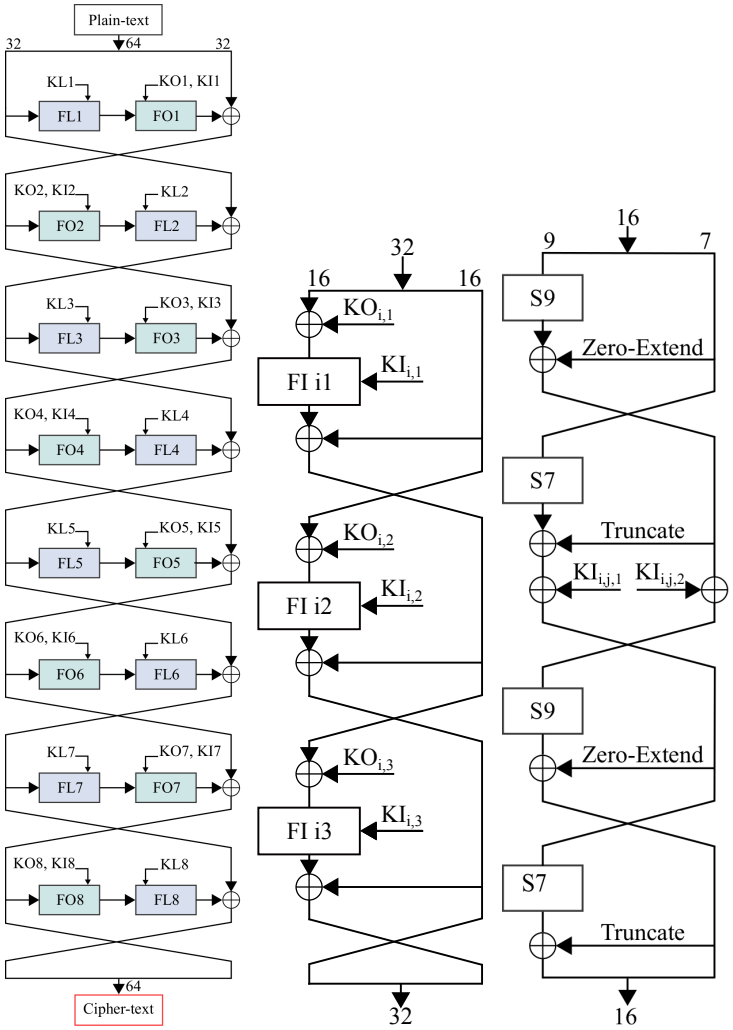
The first contribution of this work is the optimization of KASUMI architecture. Unlike the standard design based on the Feistel structure, we propose an optimized design based on two blocks of the main FL and simplified FO' functions. The purpose is to reduce the required and limited logic resources on embedded systems. The processing steps of the proposed architecture are detailed in the following subsections.

3.2.1. Global S-box

In the first step, we proposed to join S7 and S9 S-Boxes of the standard algorithm to form one specific Global S-box (GS) characterized by a 16-bits input/output block. It is designed in the way that the right part (7-bits) and the left part (9-bits) substitute S7 and S9 input/output blocks, respectively. The proposed GS is implemented using one combinational design performing a high execution time and consuming low cost of the logic FPGA resources (hardware area). This technique consists mainly to simplify the logical implementation and to perform the acceleration of operations execution.

3.2.2. Simplified FI' function

In the second step, we used the proposed GS to form a simplified FI' function which can be executed on two parallel rounds instead of four rounds in the original FI function. For this purpose, in the first round, the FI' input (16-bits) is divided into two parts, left L (9-bits) and right R (7-bits), to form the GS



(a) The KASUMI block cipher architecture.

(b) The KASUMI FO Function.

(c) The KASUMI FI Function.

(d) The KASUMI FL Function.

Figure 3: The KASUMI Feistel structure [11].

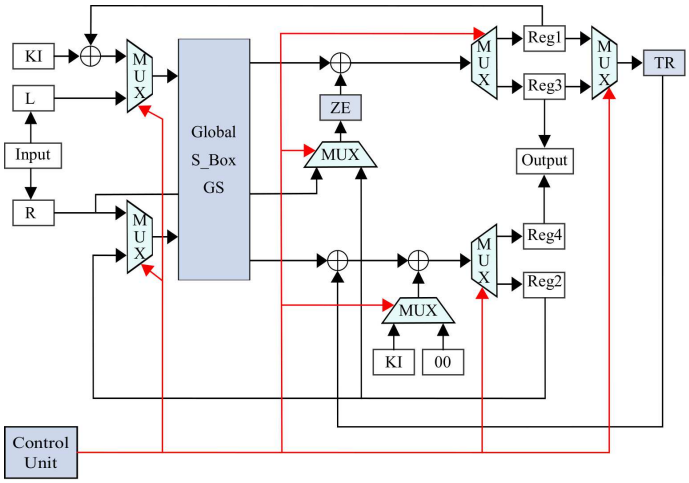


Figure 4: The simplified FI' architecture.

inputs. After this step execution, GS generates two outputs that are stored on Reg1 (9-bits) and Reg2 (7-bits) registers. In the second round, the content of Reg1 and Reg2 registers forms the novel GS inputs. After this execution step, GS generates two new outputs, which are saved on Reg3 (9-bits) and Reg4 (7-bits) registers. Thereby, the output of the simplified FI' function (16-bits) is set up by the concatenation of Reg3 and Reg4 registers. Consequently, the proposed simplification allows for twice-faster execution. The detailed architecture is given in Figure 4.

3.2.3. Simplified FO' function

In the third step, we considered our simplified FI' function to realize one simplified FO' function deeply difference with the original function based on the FI function. Indeed, the proposed FO' function is executed on three rounds. In the first round, the FO' input (32-bits) is divided into two parts, left L (16-bits) and right R (16-bits). L part is combined with subkey KO1 (16-bits) using bitwise XOR operation to form the first FI' input (16-bits). Then, the output is combined with the R part using a bitwise XOR operation to generate the first output saved on the Reg1 (16-bits) register. In the second round, the L part is combined with subkey KO2 (16-bits) using bitwise XOR operation to produce the second FI' input. Next, the output is combined with Reg1 using bitwise XOR operation to generate the second output stored on the Reg2 (16-bits) register. In the third round, Reg1 is combined with subkey KO3 (16-bits) using bitwise XOR operation to form the third FI' input. At this step, the output is combined with Reg2 using bitwise XOR operation to generate the third output to update the Reg1 register value. Thus, the output of the simplified FO' function (32-bits) is set up by the concatenation of Reg1 and Reg2 registers. The detailed architecture is depicted in Figure 5.

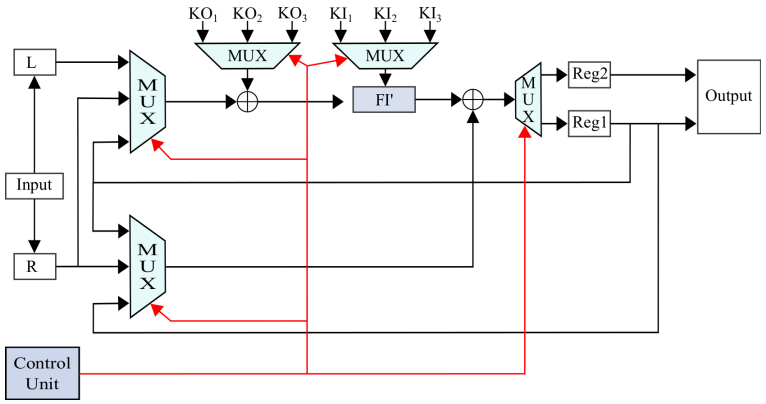


Figure 5: The simplified FO' architecture.

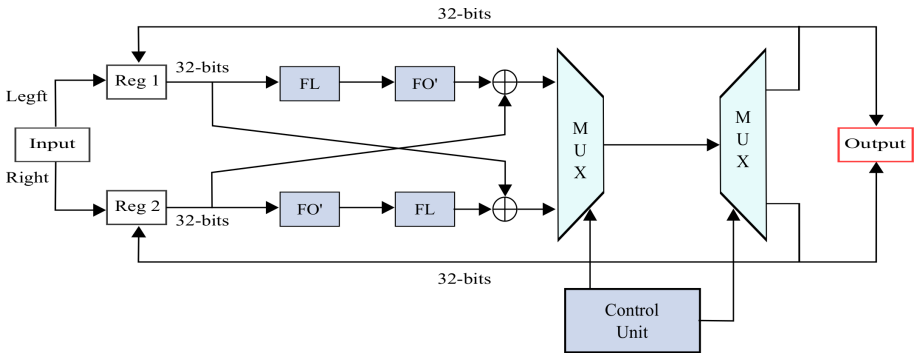


Figure 6: The optimized KASUMI block cipher architecture.

3.2.4. Simplified KASUMI block

Finally, we proposed to optimize the KASUMI block cipher by executing in the order the proposed FL and simplified FO' functions to perform the odd or even (reverse execution order) rounds. The switching execution between the odd and even rounds is managed by a specific control unit based on FSM. The final output keystream is generated after four rounds of execution of each block case. However, at each round before the final one, the FSM allows for updating the intermediate Reg1 and Reg2 registers to memorize the required input for the next round. The output keystream of the proposed architecture is carried out by concatenating the last contents of the Reg1 and Reg2 intermediate registers. Note that the initial state of the Reg1 and Reg2 intermediate registers are equal (32-bits) left and right parts of the initial input (64-bits), respectively. The detailed architecture of the proposed design is given in Figure 6.

The operating principle of the FSM unit is the control of the two-block functions based on the round loop by considering three different possible cases

as follows:

1. At the even round, block case one is executed and the $Reg2$ register is updated.
2. At the odd round, block case two is executed and the $Reg1$ register is updated.
3. At the final round, the $Reg1$ and $Reg2$ registers are concatenated to generate the output keystream.

4. Optimized A5/3 algorithm

In this section, we describe the proposed architecture and the processing steps of the optimized A5/3 encryption algorithm. The purpose is to improve the algorithm performance in terms of logic resources, ciphering throughput, and power consumption providing a good ciphering keystream designed for data encryption suitable for embedded applications.

Contrary to the regular A5/3 algorithm founded on five KASUMI block cipher components, the proposed architecture is built upon only one optimized KASUMI cipher block presented in the previous section. To generate output keystream (64-bits), two input words are required corresponding a key of 128-bits length and a principle input of 64-bits length. For each input, two cases are possible as shown in Figure 7. Therefore, the processing of the algorithm is divided into two operating modes:

4.1. Initialization mode

In this mode, inputs to the optimized KASUMI cipher block are:

1. The CA, CB, CC, CD, CE initialization standardized parameters (see Table 1), which are concatenated, and then loaded to form the first input signal of 64-bits length. Note that the remaining bits of the parameter CC are forced to zero.
2. The main CK is combined with KM using a bitwise XOR operation to form the second input signal of 128-bits length.

This mode is executed for one cycle, and the first 64-bits output word is saved in the reference register (A), which will be used during all following rounds in the keystream mode. Note that $Select$ signal is set up at "0".

4.2. Keystream mode

In this mode, inputs to the optimized KASUMI cipher block are:

1. The content of the A reference register which is combined using bitwise XOR operations with the contents of keystream word and the BLocK CouNTer dynamic register (BLKCNT) initially set to zero. The result of this operation forms the first input signal of 64-bits length.
2. The second input signal is CK of 128-bits length.

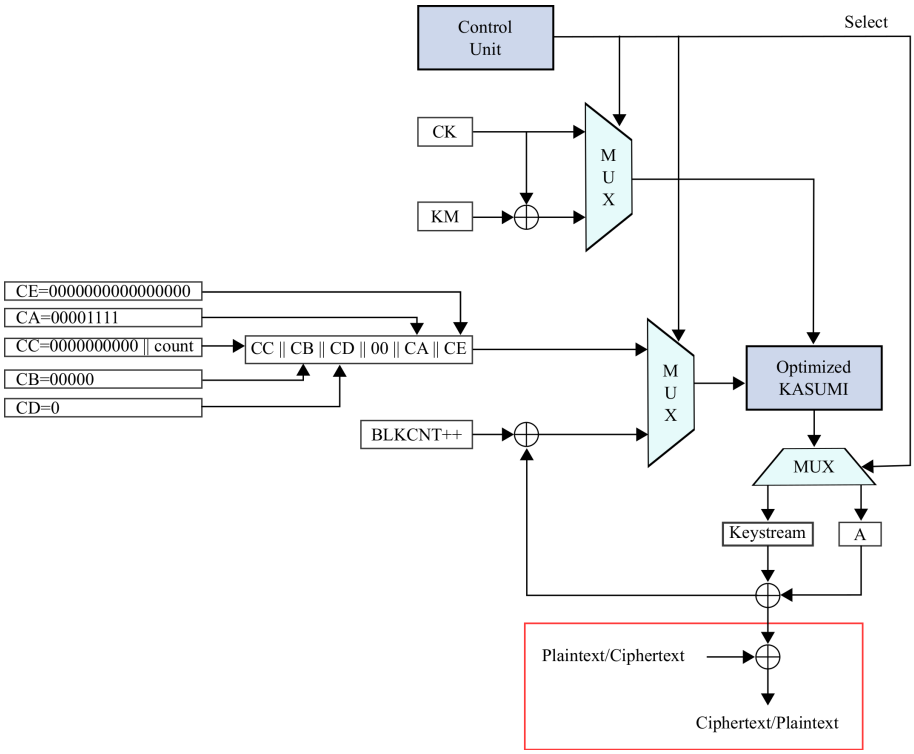


Figure 7: The optimized A5/3 algorithm architecture.

This mode is executed n times. n is the number of 64-bits blocks to be encrypted or decrypted by the algorithm. The generated output keystream is combined with the plaintext using a bitwise XOR operation to produce the ciphertext in encryption procedure, and the reverse operation is applied in the decryption procedure (see Figure 7). Note that the $BLKCNT$ value is incremented after each round, and the *Select* signal is set up at "1".

5. Hardware implementation results

In this section, we present the hardware implementation results in terms of internal components and FPGA logic resources used by the proposed architecture.

5.1. Required internal functions

The internal structure of the proposed architecture is based on one component of the optimized KASUMI block cipher (see Section 4). Indeed, the FL and simplified FO' functions are the two main units of the A5/3 algorithm. Table 2 summarizes the number of the required functions for the proposed A5/3 algorithm.

Component	Amount
Kasumi block cipher	1
FL function	2
FO' function	2

Table 2: The number of required functions for the proposed A5/3 algorithm.

Algorithm	Optimized A5/3 proposed design
Device	Virtex XC5vfx70t-1ff1136
Number of slice registers	1323 / 44800
Number of occupied slice	987 / 11200
Number of slice LUTs	1877 / 44800
Number of fully used LUT-FFpairs	1081 / 2119
Number of bonded IOBs	229 / 640
Number of BUFG/BUFGCTRLs	5 /32
Maximum frequency (Mhz)	250
Throughput (Mbps)	2000
Power (Watts)	1.46

Table 3: FPGA implementation results of the proposed algorithm.

5.2. FPGA implementation and logic resources

The Register Transfer Level (RTL) description of the proposed architecture has been implemented on Xilinx Virtex-5 FPGA (XC5vfx70t-1) through the ML507 Virtex development platform [21] using VHSIC Hardware Description Language (VHDL) structural description. Integrated Synthesis Environment (ISE) 13.2 of Xilinx tools have been used for this digital implementation allowing us to obtain the logic resource requirements and the associated real-time constraints. The synthesis results after a place and route show the performance of our implementation in Table 3. This table depicts the hardware resources in terms of the occupied logic slice, slice register, slice LUT, slice Flip-Flops numbers and power consumption. It shows that the proposed digital implementation on a Xilinx Virtex-5 device requires only 1323 slice registers, 1877 slice LUT, and no block RAMs. In addition, the low power consumption estimated by 1.46 Watts as well as the high throughput of 2000 Mbps is suitable for embedded applications designed for real-time communications (mobile networks).

6. Comparison and discussion

In this section, we give a comparison of the proposed architecture with previous works.

6.1. Internal structure and implementation

We consider in this subsection, the comparison between the proposed architecture with the structural and internal architectures of the most previous

	Original	[22]	[9]Initial work	Proposed work
Kasumi components	5	1	2	1
FL function	40	8	8	2
FO function	40	8	8	2
Decreased area	-	65.62%	78.75%	95%

Table 4: The A5/3 combinatory logic resources comparison results.

works on the A5/3 encryption algorithm (original A5/3, A5/3 presented in [22] and initial optimized A5/3 [9]).

The main unit of the A5/3 encryption algorithm is the KASUMI cipher block (5 components used) founded on main units which are FL and FO functions. The FL function is formed by 1 OR logic operator, 1 AND logic operator, and 2 XOR logic operators. The FO function is based on 3 FI functions and 6 XOR logic operators. The FI function is formed by 4 Substitution boxes ($S-boxes$) and 6 XOR logic operators (see Figure 3(c)). The proposed A5/3 architecture is formed by 3 multiplexers, 1 control unit, 4 XOR logic operators, and one optimized KASUMI cipher block built upon 2 FL and 2 simplified FO' functions, 2 multiplexers, 1 control unit, 2 XOR logic operators, and 2 intermediate registers. The simplified FO' function is formed by a simplified FI' function, 5 multiplexers, 1 control unit, 2 XOR logic operators, and 2 intermediate registers. The simplified FI' function is formed by GS, 1 control unit, 7 multiplexers, 4 XOR logic operators, and 4 intermediate registers.

Table 4 gives a comparison between the proposed architecture, the original architecture of the A5/3 encryption algorithm, our initial work presented in the IEEE-2017 7th ICCSS [9], and the implementation proposed in [22]. The comparison is performed with respect to the number of KASUMI component, and the number of internal FL and FO functions. In addition, the Figure 8 gives a graphical illustration to highlight the decreased area in the proposed design. According to those results, we conclude that the optimized A5/3 encryption algorithm (based on 2 FL and 2 modified FO' functions) proposed in this work, is the best one considering it allows to reduce the required area approximatively by $\approx 95\%$ compared to the original A5/3 algorithm (requiring 40 FL and 40 FO functions).

6.2. Time performance comparison

In this subsection, we compare the proposed architecture, in terms of maximum time-frequency and height throughput, to previous works [22–34] implementing the KASUMI, the A5/3 algorithm used in GSM and UMTS (by performing the f8 function) as described by the standard guidelines [11, 14, 18]

Through the analysis of the comparison results presented in Table 5, we acknowledge that the described design in this study achieves the highest throughput compared to previous works due to its high processing clock frequency. Therefore, the proposed architecture allows speed encryption of user's data,

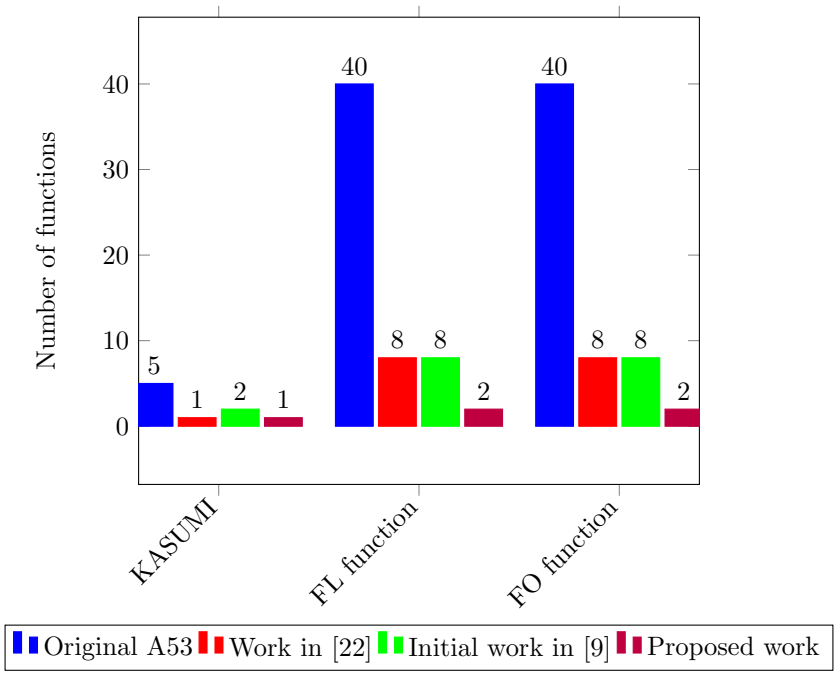


Figure 8: Comparison between the proposed and previous A5/3 architectures.

Architecture	Device	Freq (Mhz)	Debit (Mbps)
KASUMI in [23]	XCV300E-6BG 432	20.88	167.04
KASUMI in [24]	Virtex-E	20	110
KASUMI in [25]	XCV300E-6BG432	58.14	66.45
KASUMI in [25]	XCV300E-8BG432	68.13	77.86
KASUMI in [26]	XCV300E-8BG432	41.14	165
KASUMI in [27]	XCV300E-8BG432	41.63	222
KASUMI in [28]	-	71	68
KASUMI in [29]	XCV300E-8BG432	54	432
KASUMI in [30]	XCV300E-8BG432	79.45	318
KASUMI in [31]	Virtex-II	96.33	385.32
KASUMI in [32]	XCV300E-6BG432	31.93	36.09
KASUMI in [32]	XCV300E-8BG432	39.4	44.54
A5/3 in [22]	XC2V500-6FG456	130	166
f8_Comb in [23]	XCV300E-6BG432	20.52	162.1
f8_LUT in [23]	XCV200E-6FG456	33.14	261.8
f8_Comb in [33]	XCV300E-6BG432	16.93	135
f8_LUT in [33]	XCV600E-6BG432	46.56	372
f8_type1 in [24]	-	19.5	154
f8_type2 in [24]	-	52	411
f8_synth1 in [25]	XCV300E-8BG432	30.12	240.96
f8_synth2 in [25]	XCV300E-8BG432	25.80	206.40
f8_RIJNDAEL (ROM) [34]	XCV300E-8BG432	104	822
f8_RIJNDAEL (Logic) [34]	XCV300E-8BG432	53.5	423
Proposed A5/3	XC2V500-6FG456	183.37	1466.96
Proposed A5/3	XCV200E-6FG456	95.55	764.4
Proposed A5/3	XCV300E-6BG432	95.56	764.48
Proposed A5/3	XCV300E-8BG432	119.56	956.48
Proposed A5/3	XCV600E-6BG432	95.56	764.48
Proposed A5/3	XC5VFX70T-FF1136	250	2000

Table 5: A5/3, f8, and KASUMI time performance comparison results.

which is considered as a critical parameter in real-time applications, in particular for mobile network communications.

6.3. Generated outputs comparison

To prove the efficiency of the proposed implementation, we used the 3GPP test set vector [19] designed to help implementers in their realization. This document provides test data for the algorithms as well as details on the internal states of the algorithms when they process the input data. For example, the test set vectors 1 is defined as follows.

- The algorithm inputs are presented in Table 6.
- The plaintext and the corresponding ciphertext are presented in Table 7.

Key	=	$2BD6459F82C5B300952C49104881FF48$
$Count$	=	$72A4F20F$
$Bearer$	=	$0C$
$Direction$	=	1
$Length$	=	$798bits$

Table 6: 3GPP Test set vector1 inputs.

Plaintext	Ciphertext
7EC61272743BF161	D1E2DE70EEF86C69
4726446A6C38CED1	64FB542BC2D460AA
66F6CA76EB543004	BFAA10A4A093262B
4286346CEF130F92	7D199E706FC2D489
922B03450D3A9975	1553296910F3A973
E5BD2EA0EB55AD8E	012682E41C4E2B02
1B199E3EC4316020	BE2017B7253BBF93
E9A1B285E7627953	09DE5819CB42E819
59B7BDFD39BEF4B2	56F4C99BC9765CAF
484583D5AFE082AE	53B1D0BB8279826A
E638BF5FD5A6061	DBBC5522E915C120
3901A08F4AB41AAB	A618A5A7F5E89708
9B134880	9339650F

Table 7: 3GPP Test set vector1 plaintext/ciphertext.

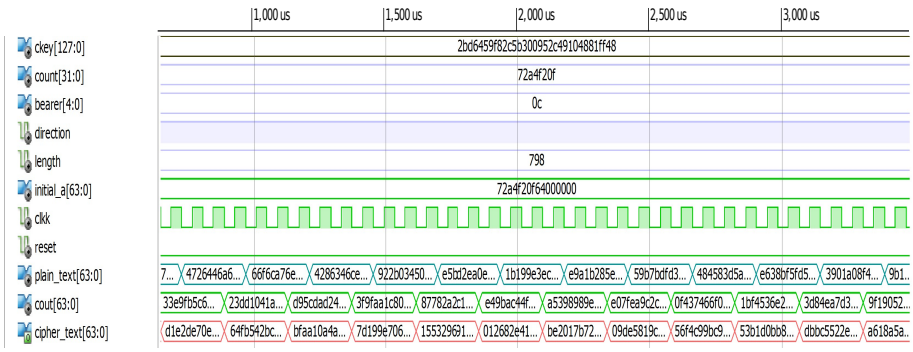


Figure 9: The optimized A5/3 algorithm output.

To validate the outputs generated by the proposed algorithm, we used the 3GPP reference data presented in Tables 6 and 7 and the ISE Simulator (ISim). The simulation results (VHDL test bench waveform) are shown in Figures 9.

By comparing the ciphertext generated by the original algorithm presented in Tables7 with the ciphertext generated by the optimized algorithm presented in Figures 9, it is clear that the proposed implementation generates the same and expected outputs as the original one. Consequently, we conclude that the optimized A5/3 design proposed in this study can ensure the same functionalities compared to the original A5/3 design.
Note that the proposed algorithm satisfies all of the test data sets provided in [19].

6.4. Discussion

According to the results presented in Tables 4 and 5, it is clear that the proposed architecture requires lower hardware resources (slice register, occupied slice, slice LUTs) and internal components (KASUMI block cipher, FL and FO functions) while allowing a high throughput to ensure the same functionalities of the standard algorithm. The proposed architecture presents one simple and low complexity implementation on a hardware device (FPGA technology). In addition, the characterizing high performance proves that the proposed architecture is more suitable for being used in GSM and UMTS networks. Consequently, our architecture can replace the old version of the A5/3 encryption algorithm while satisfying the network requirements (time constraint, size of data, etc.). Note that it is not significant to compare the occupied logic area of each cited works considering the different technologies used for implementations. For that, we mainly focused our comparison on time-performance. Finally, we conclude that the optimized A5/3 encryption algorithm proposed in this work is also suitable for embedded applications (handsets), in particular for mobile phone in our case study. In terms of security, the proposed architecture is robust against the linear and nonlinear cryptanalysis similarly to the original version of the A5/3

algorithm. However, in the last decade, KASUMI and A5/3 were the subjects of many cryptanalysis attacks [5, 35–40].

7. Conclusion

In this work, two main contributions were proposed. First, we have optimized the KASUMI block cipher, which forms the kernel of the GSM and UMTS security. We have also proposed an optimization of the A5/3 algorithm based on the modified KASUMI cipher block unlike the original algorithm founded on five KASUMI components. The proposed architecture has been performed and implemented on an FPGA Virtex technology using VHDL structural description. The hardware and synthesis implementation results after a place and route show good performance of the optimized A5/3 architecture. Consequently, the final proposed design allows us to decrease the occupied area in the handset by almost compared to the original version. Furthermore, our proposed algorithm increases the encryption speed to achieve a throughput of 2000 Mbps in Virtex-5 technology. By considering the obtained good trade-off between time performance and logic resources, compared with previous works, we conclude that the proposed design forms the best-designed algorithm to ensure data protection (confidentiality and integrity) in the second and third generations of mobile networks. Besides, this proposed architecture is still satisfying the real-time and embedded systems properties, while preserving the 3GPP standardized requirements (key length, data size, etc.). Therefore, it is possible to replace the old A5/3 encryption algorithm with the proposed optimized version, which can ensure the same functionalities with better performance.

The future works will mainly focus on the robustness enhancement of the KASUMI encryption algorithm, which is still largely used in countries using UMTS mobile networks. To achieve this objective, we consider this proposed optimized-architecture which can be combined with previous works [6–8] to increase data protection against the usual cryptanalysis attacks.

References

[1] E. Barkan, E. Biham, Conditional Estimators: an Effective Attack on A5/1, in: LNCS 3897, SpringerVerlag, 2006, pp. 1–19 (2006).

[2] A. Biryukov, Shamir, D. Wagner, Real time cryptanalysis of A5/1 on a PC, in: Fast Software Encryption Workshop 2000, Springer-Verlag, 2000, pp. 1–18 (2000).

[3] P. Ekdahl, T. Johansson, Another Attack on A5/1, IEEE Transactions on Information Theory 49 (1) (2003) 284–289 (2003).

[4] A5/1 Security Project, Creating A5/1 Rainbow Tables, Tech. rep., `http://reflextor.com/trac/a51/wiki`
`http://opensource.srlabs.de/`

http://events.ccc.de/congress/2009/Fahrplan/attachments/1519_26C3.Karsten.Nohl.GSM.pdf
http://events.ccc.de/congress/2009/Fahrplan/events/3654.en.html (2009).

[5] J. Hong, P. Sarkar, New Applications of Time Memory Data Tradeoffs, in: International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT), Vol. 3788, LNCS, 2005, pp. 353–372 (2005).

[6] M. Salman, R. Yugitama, Amiruddin, R. F. Sari, KAMIES: Security Optimization of KASUMI Algorithm by Increasing Diffusion Level, International Journal of Security and Its Applications 12 (3) (2018) 29–46 (2018).

[7] R. Muthalagu, S. Jain, Modifying the structure KASUMI to improve its resistance towards attacks by inserting FSM and S-Box, Journal of Cyber Security Technology 2 (2018) 37–50 (2018).

[8] R. Muthalagu, S. Jain, Reducing the time required by KASUMI to generate output by modifying the FL and FI functions, Iran Journal of Computer Science 2 (2019) 33–40 (2019). doi:https://doi.org/10.1007/s42044-018-0017-2.

[9] M. Madani, S. Chitroub, C. Tanougast, Two KASUMI Components for an Optimal Implementation of the A5/3 Algorithm, in: The International Conference on Circuits, System and Simulation (ICCSS 2017), IEEE Conference Publications, London, UK July 14-17, 2017, pp. 124–128 (2017).

[10] 3G Security; Technical Specification Group Services and system Aspects; Security related network functions, Technical Specification (TS) TS 43.020 V12.0.0, 3GPP (2013-03).

[11] 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI specification, Technical Specification (TS) TS 35.202 V15.0.0, 3GPP (Jun 2018-06).

[12] B. Noureddine, Security of Mobile Communications, 2009 (2009).

[13] T. Liu, C. Tanougast, S. Weber, A Framework of Architectural Synthesis for Dynamically Reconfigurable FPGAs, in: IEEE Circuits and Systems Society (Ed.), IEEE International SOC Conference 2008, IEEE Circuits and Systems Society, Newport Beach-California, USA, 2008, pp. 283–286 (2008).

[14] 3G Security; Specification of the A5/3 encryption algorithms for GSM and ECSD, and the GEA3 encryption algorithm for GPRS; Document 1: A5/3 and GEA3 specifications, Technical Specification (TS) TS 55.216 V12.0.0, 3GPP (2014-10).

[15] 3G Security; Specification of the A5/3 Encryption Algorithms for GSM and ECSD, and the GEA3 Encryption Algorithm for GPRS; Document 4: Design and evaluation report, Technical Specification (TS) TS 55.919 V12.0.0, 3GPP (2014-09).

[16] D. Basin, P. Schaller, M. Schläpfer, Applied Information Security, Springer-Verlag Berlin Heidelberg, 2011 (2011).

[17] M. Matsui, New Block Encryption Algorithm MISTY, in: . L. N. in Computer Science (Ed.), Fast Software Encryption-FSE'96, Springer-Verlag, 1997, pp. 54–68 (1997).

[18] 3G Security; Specification of the 3GPP confidentiality and integrity algorithms; Document 1: f8 and f9 specification, Technical Specification (TS) TS 35.201 V15.0.0, 3GPP (2018-06).

[19] 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 3: Implementors Test Data, Technical Specification (TS) TS 35.203 V15.0.0, 3GPP (2018-06).

[20] 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 4: Design Conformance Test Data, Technical Specification (TS) TS 35.204 V15.0.0, 3GPP (2018-06).

[21] Virtex 5 FPGA Configuration User Guide, ug702, Xilinx (2012).

[22] E. Vrentzos, G. Kostopoulos, O. Koufopavlou, Hardware Implementation of The a5/3 & a5/4 GSM encryption algorithms, in: WAC, 2006 (2006).

[23] K. Marinis, N. Moshopoulos, F. Karoubalis, K. Pekmestzi, On the hardware implementation of the 3GPP confidentiality and integrity algorithms, in: the 4th International Conference for the Information Security, ISC 2001, Malaga, Spain, 2001, p. 248–265 (October 1-–3 2001).

[24] H. Kim, Y. Choi, M. Kim, H. Ryu, Hardware Implementation of 3GPP KASUMI Crypto Algorithm, in: The 2002 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Phuket, Thailand, 2002, p. 317–320 (July 16—19 2002).

[25] A. Satoh, S. Morioka, Small and high-speed hardware architectures for the 3GPP standard cipher KASUMI, in: the 5th International Conference Information Security, ISC 2002 (2002), Lecture Notes in Computer Science 2433 Springer-Verlag, Sao Paulo, Brazil, 2002 (September 30-–October 2 2002).

[26] T. Balderas, R. Cumplido, An Efficient Hardware Implementation of the KASUMI Block Cipher for Third Generation Cellular Networks, in: Proc. GSPx, 2004 (2004).

[27] T. Balderas, R. Cumplido, An Efficient Reuse-Based Approach to Implement the 3GPP KASUMI Block Cipher, in: Proc. ICEEE 2004.

[28] H. Kim, S. Lee, Design and Implementation of a Private and Public Key Crypto Processor and Its Application to a Security System, IEEE Transactions on Consumer Electronics 50 (1) (2004) 214—224 (2004).

[29] P. Kitsos, M. D. Galanis, O. Koufopavlou, High-speed hardware implementations of the KASUMI block cipher, in: International Symposium on Circutis and Systems (ISCAS'4), Vol. 2, Vancouver, Canada, 2004, pp. 549—552 (May 23–26 2004).

[30] T. Balderas, R. Cumplido, High Performance Encryption Cores for 3G Networks, in: Proc. DAC 2005.

[31] T. Balderas, R. Cumplido, C. Feregrino-Uribe, On the design and implementation of a RISC processor extension for the KASUMI encryption algorithm, Computers and Electrical Engineering 34 (6) (2008) 531—-546 (2008).

[32] Y. Dai, I. Kouichi, Y. Jun, A Very Compact Hardware Implementation of the KASUMI Block cipher, in: LNCS 6033, 2010, pp. 293—307 (2010).

[33] K. Marinis, N. Moshopoulos, F. Karoubalis, K. Pekmestzi, An area optimized hardware implementation of the 3GPP confidentiality and integrity algorithms, in: the 8th Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2002, Brasov, Romania, 2002 (May 16—17 2002).

[34] P. Kitsos, Y. Sklavos, O. Koufopavlou, UMTS security: system architecture and hardware implementation, Wireless Communications And Mobile Computing 7 (2007) 483—-494 (2007).

[35] Y. Wentan, C. Shaozhen, Multidimensional zero-correlation linear cryptanalysis of the block cipher KASUMI, IET Information Security 10 (2016).

[36] Z. Wang, X. Dong, K. Jia, J. Zhao, Diffeential Fault Attack on KASUMI Cipher Used in GSM Telephony, Mathematical Problems in Engineering 2014 (2014) 7 pages (2014). doi:http://dx.doi.org/10.1155/2014/251853.

[37] O. Dunkelman, N. Keller, A. Shamir, A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony, Journal of Cryptology 27 (2014).

[38] K. Jia, L. Li, C. Rechberger, J. Chen, X. Wang, Improved Cryptanalysis of the Block Cipher KASUMI, in: International Conference on Selected Areas in Cryptography SAC 2012, Lecture Notes in Computer Science, volume 7707, Windsor, Canada, 2012, pp. 222–233 (Aug 2012).

[39] E. Biham, O. Dunkelman, N. Keller, A related-key rectangle attack on the full KASUMI, in: International Conference on the Theory and Application of Cryptology and Information Security ASIACRYPT 2005, LNCS, volume 3788, 2005, pp. 443–461 (2005).

[40] M. Blunden, A. Escott, Related Key Attacks on Reduced Round KASUMI, in: International Workshop on Fast Software Encryption FSE 2001, LNCS, volume 2355, 2002, pp. 277–285 (2002).