Complete Security Stack FPGA Implementation of The Software Defined Radio on ZYNQ.

Abdelrhman M. Abotaleb[1, 2], Abdulkareem M.Abotaleb[1], Amr G. Wassal[2], and Hassan Mostafa[1,3].
[1]Electronics and Communications Engineering Department, Cairo University, Giza 12613, Egypt.
[2]Computer Engineering Department, Cairo University, Giza 12613, Egypt.
[3]Nanotechnology Department at Zewail City for Science and Technology, Cairo, Egypt.
aabotaleb@nu.edu.eg, abdulkareem7070@gmail.com, wassal@eng.cu.edu.eg,
hmostafa@uwaterloo.ca

Abstract**—** The increasing deployment of the Internet of Things (IoT) in every life aspects makes the different communications methodologies used in IoT applications are vulnerable to cyber-attacks, either through passive attacks, intercepting un-authorized messages between two communications parties, or through masquerading active attacks, being able to modify or even fabricate new message to communications parties. This work implements the whole Software Defined Radio (SDR) with GSM, UMTS, and Bluetooth communications block then integrating the communications blocks with focus in optimizing the added security layers, including the authentication to overcome masquerading and encryption to overcome passive attacks , for the 2G (Comp128, A3, A5 and A8), 3G and 4G (the KASUMI, F8 and F9) and the Bluetooth (SAFER+,E1, E2 and E3) algorithms on the Field programmable gate array of the ZYNQ System-On-Chip while focusing on minimizing the utilization and power consumption of the security functions part.

Keywords**—***System-on-Chip, FPGA, KASUMI, SAFER, COMP128.*

I. Introduction

The wireless Software defined radio, which is implemented on the system of chip, is an essential part of the reconfigurable communications can make use of the dynamic partial reconfiguration ability to reduce the power needed while doing the proper switching from a certain standard to another and keep the communication alive during the running time, when this software radio offers the plenty of standards including the 2G, 3G, 4G and Bluetooth, this will make it an excellent candidate for the internet of thing applications [1].

The major issue that need careful attention is the security concern, so not only making the software defined radio and other similar Internet of thing applications more accessible and easier to use, but also due to the scalable use of the communication devices, it puts a higher concern to make the intended and authorized user communications are secure as possible otherwise it would be useless [2], and [3].

The main focus of the current work is to provide the security layers necessary for the SDR including the authentication, confidentiality and integrity security functions.

The security layers are implemented in the programmable logic part of the ZYNQ to offer the outperforming capabilities of fast hardware and real parallelism, avoiding pseudo-parallelism and extra layers provided by operating systems in the pure software implementation of the security function, also the RTL implementation of the security functions provides fast interconnections with the physical layer part of the communications standard, the RTL communications part is implemented based on the design shown in [4].

II. Literature Review

Selected literatures here are describing just a single part of algorithm; the whole system implementation of the whole security chain integrated with the communications chain is a genuine addition in the current work.

In 2017, Yasir, et al [5] proposes three different strategies for the F8 and F9 3G security functions implementations as the following:

1- Pipelining design, which is achieved by unrolling the 8 rounds of the KASUMI; the throughput reaches 13.6 Gbps versus 6346 Mbps in the original KASUMI.
2- Moderate area/Moderate speed, achieved by making odd and even rounds of KASUMI on +ve and -ve triggering.
3- Low Area/Low speed.

For each strategy, Substitution blocks are implemented in both ROM , LUTs and logic combinational gates, the logic combinational implementation of S-BOX gives better utilization but adds extra propagation delay.

In 2018, Yasir, et al. [6] introduced the combined block substitution method to optimize the 3G security functions implementations by applying resource sharing between different combinational logic operations, and also utilizes the pipelining idea of running the KASUMI rounds previously developed in [5].

Table 1 shows the reduction on the utilization of the S-boxes of the 3G security functions over straight forward utilization, i.e. without using resource sharing.

Table 1: Gate count reduction by running different strategies of compact S9-Box, S7-Box on the Xilinx FPGA reported in [6].

	S9 s-box		**S7 s-box**	
	AND	XOR	AND	XOR
Straight Forward	86	97	104	77
Proposed	36	68	41	52
Reduction%	58	30	61	32

In [7] implements both the encryption and decryption of the SAFER+ (Secure and Fast Encryption Routine) used in the Bluetooth, but doesn't include the remaining SAFER+ based Bluetooth security functions namely E1, E2, and E3 which are used for the authentication and key generation blocks.

This work starts by describing one round of the SAFER+ that is the basic unit of the Bluetooth authentication algorithms, then deduce that six or more rounds of the SAFER is secure enough against the differential cryptanalysis, then it propose an improvement that can reduce the number of logic elements required to implement a single round of SAFER+ by using the concept of data mapping and data dumping for the exponential and log functions used in the crypto standard of the SAFER which results in area optimization on the same throughput of the original SAFER.

Table 2 summarizes the gate level count reduction and frequency enhancement achieved throughout the proposed implementation of SAFER+.

Table 2: Utilization, Speed, and throughput reports by running the original SAFER+ versus the optimized SAFER+ in [7].

Type	Original	Proposed Optimized
Gate level count	233839	200013
Frequency	20 MHz	44 MHz
Throughput	320Mbits/sec	704Mbits/sec

III. PROPOSED DESIGN

In the current work, the distinctive is the implementation of whole security blocks of 2G, 3G, 4G, and the Bluetooth then optimize their gate count, finally integrating them to SDR communications block.

Security algorithms implemented are as the following:

1. ***COMP128v1*** includes ***A3*** and ***A8*** algorithms for the GSM authentication and ciphering key generation algorithms.
2. ***A5/1*** is the GSM encryption algorithm.
3. ***KASUMI*** is the core 3G/4G security block, it includes ***FI, FL,*** and ***FO*** for block cipher sub-functions , key scheduling [***KL, KI,*** and ***KO*** generation units] , ***F8*** [confidentiality algorithm] , and ***F9*** [Data integrity algorithm].
4. ***SAFER+***, which is the Bluetooth 2.0 core for encryption, ***Ar*** block, ***E1***, ***E2***, and ***E3*** used for authentication , key generation Bluetooth SAFER based algorithms and finally the ***E0*** Bluetooth encryption algorithm.

Then applying the following ***optimizations*** :

1. Compact design of KASUMI utilizes combined substitution block common method as in [6].
2. Compact area authentication and key generation SAFER+ based for Bluetooth using resource sharing as in [7].

Finally, integrating the security functions with the SDR, showing that the added security blocks have small gate count and power consumption in comparison to the SDR communications blocks.

IV. 2G SECURITY CHAIN

This section illustrates the implemented security algorithms of the GSM, which is basically the COMP128 standard for the authentication and the A5 algorithm for the cryptography.

COMP128 is composed of two sub-algorithms known as **A3**, which is used for the authentication based on challenge-response one-way authentication and **A8,** which is used for the encryption key generation (and inherently the encryption key exchange).

The **A5** algorithm is used for the voice coded data encryption/decryption with the generated encryption key resulted from A8 algorithm [8]

COMP128 standard details was initially confidential by ETSI, till been partially leaked by some of operator's employee on 1997 then it is completely revealed by doing some reverse engineering on 1998[9].

Once the mobile equipment begins to search for a nearby base station in order to connect to the cellular networks, it goes through the authentication/encryption key generation process; The process of the signaling is as the following:

1. The mobile equipment (ME) is sending its IMSI (an identifier which is unique for each SIM card).
2. The authentication center in the core of the cellular network responds with a challenge (i.e. RAND = a 128-bit random number) being sent to the mobile hand set.
3. Each SIM card processes a unique authentication key (128-bit secret number), in contrast to the IMSI which could be read easily from the SIM card, this ki is secret and couldn't be revealed by the mobile user.
4. The A3 Algorithm in ME responds with a signed response SRESR (64-bit), verifying the user.
5. A8 Algorithm produce another 64-bit number (kc) which is the encryption key which will be used fatherly to encrypt/decrypt the GSM TDMA bursts as in [8].

A. The COMP128 algorithm

COMP128 do both A3 and A5 authentication and ciphering key generation functions, its main building block is a 5 compression rounds working on a byte level, each compression round use a table from leaked 5 tables for substitution and compression, the 5 compression rounds are done iteratively inside a bigger loop of 8 passes, and this bigger loop doesn't include only the compression round but starts by assigning the least 16 bytes of x to ki then follows the compression rounds by a permutation of the intermediate variable x bits.

B. A5/1 encoded voice encryption algorithm

A5/1 is the first version used for over-the-air GSM encoded voice encryption in Europe and United States and was developed in 1987 and it was export restricted; later in a weaker version called A5/2 is developed in 1989 for GSM encryption outside Europe and United states.

Both A5/1 and A5/2 are confidential and was kept secret till being reversed engineered in 1999 by Marc Briceno from a GSM telephone.

A5/1 procedure is as following:

1- For each frame, accept both frame_number (22-bit) and the kc (64-bit).

2- The algorithm uses 3 LFSRs to produce 114-bits output.

3- At the transmitter, the output is furthermore XORed with the burst to produce encrypted burst.

4- At the receiver, the output is XORed with the ciphered burst which decrypts the original burst encoded voice/data.

Table 3 shows the internal design of LFSRs used in the A5/1 encryption algorithm.

Table 3: **A5/1 Ciphering Sequence Generation details as in [9]**

	LFSR1	LFSR2	LFSR3
Length:	19 bits	22 bits	23 bits
Clocking bit:	8	10	10
Tapped bits:	13-16- 17 - 18.	20- 21.	22- 21- 20-7.
Polynomial:	$x^{19}+x^{18}+x^{17}+x^{14}+1$	$x^{22}+x^{21}+1$	$x^{23}+x^{22}+x^{21}+x^{8}+1$

V. 3G and 4G Security Chain

KASUMI is the core for two main security algorithms used in the 3G cellular communications and early 4G cellular communications, named: F8 for confidentiality (encryption and decryption) and F9 for integrity (message authentication), In this section KASUMI is described. KASUMI is a Feistel cipher with 8 rounds. It operates on a 64-bit data block and uses a 128-bit key, the full details of the KASUMI encryption is illustrated at [10], [11], and [12].

KASUMI operates on a 64-bit input I using a 128-bit key K to produce a 64-bit output OUTPUT, The input I is divided into two 32-bit strings the left L_0 and the right R_0, where $\boldsymbol{I = L_0 \| R_0}$.

Then for each integer i with $1 \leq i \leq 8$ we define:

$$\boldsymbol{R_i = L_{i-1}, L_i = R_{i-1} \oplus f_i(L_{i-1}, RK_i)}$$

f_i denotes the round function with L_i-1 and round key RK_i as inputs, The result OUTPUT is equal to the 64-bit string ($L_8 \| R_8$) offered at the end of the eighth round.

$$\boldsymbol{OUTPUT = KASUMI[I]_K.}$$

A. Function fi:

It takes 32-bit input (L_{i-1}: left part of the previous round result) and the round key (RK_i which comprises sub-key triplet of (KL_i, KO_i, KI_i)).

$$\mathbf{f_i(I, RK_i) = FO(\ FL(\ I, KL_i\ ), KO_i, KI_i\ )\quad i = 1, 3, 5, 7.}$$
$$\mathbf{f_i(I, RK_i) = FL(\ FO(\ I, KO_i, KI_i\ ), KL_i\ )\ i = 2, 4, 6, 8.}$$

B. Function FL:

The $\boldsymbol{KL_i}$ sub-key is split into two 16-bit sub-keys, named $\boldsymbol{KL_{i,1}}$ and $\boldsymbol{KL_{i,2}}$ where $\mathbf{KL_i = KL_{i,1} \| KL_{i,2}}$.

Figure 1 shows the block diagram of the FL function, where <<< means shift left, ∪ is OR, and ∩ is AND.

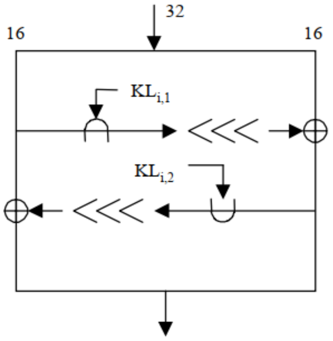

Fig. 1: FL function block diagram

C. Function FO:

Function FL accepts 32-bit input $\boldsymbol{I}$ and two sets of sub-keys: a 48-bit sub-key $\boldsymbol{KO_i}$, and a 48-bit sub-key $\boldsymbol{KI_i}$, and produces 32-bit output.

The input data $\boldsymbol{I}$ is split into two 16-bit halves, left half $\boldsymbol{L_0}$ and right half $\boldsymbol{R_0}$ where $\boldsymbol{I = L_0 \| R_0}$.

The 48-bit sub-keys are furthermore subdivided into three 16-bit sub-keys where $\mathbf{KO_i = KO_{i,1} \| KO_{i,2} \| KO_{i,3}}$ and $\mathbf{KI_i = KI_{i,1} \| KI_{i,2} \| KI_{i,3}}$.

Then three rounds of computations are done obeying the next equation in a feistel network structure: $\mathbf{R_j = FI(L_{j-1} \oplus KO_{i,j}\ ,\ KI_{i,j}\ ) \oplus R_{j-1}}$, $\mathbf{L_j = R_{j-1}}$, The 32-bit output value is ($\boldsymbol{L3} \| \boldsymbol{R3}$).

D. Function FI:

Function $\boldsymbol{FI}$ accepts 16-bit input $\boldsymbol{I}$ and 16-bit sub-key $\boldsymbol{KI_{i,j}}$, and produces 16-bit output.

FI uses substitution boxes, S7 maps a 7-bit input to a 7-bit output, and S9 maps a 9-bit input to a 9-bit output.

E. Key Scheduling.

The 128-bit secret key K stored in the USIM card is subdivided into 8 parts each part is of 16-bit length, so

$$\mathbf{K = K_1 \| K_2 \| K_3 \| K_4 \| K_5 \| K_6 \| K_7 \| K_8.}$$

A second array of 8 sub-keys named K1', K2', …, K8' are computed according to equation: $\mathbf{K_i' = K_i \oplus C_i}$; i=1, 2, 8 where C_i is tabulated in table 4 below.

Table 4: Array of constants C1, C2, ..., C8 in KASUMI

C1	0x0123	C5	0xfedc
C2	0x4567	C6	0xba98
C3	0x89ab	C7	0x7654
C4	0xcdef	C8	0x3210

VI. Bluetooth Security Chain

Bluetooth security architecture consists of four big blocks. First, E0 block is constructed to perform the encryption which is driven from the Massey-Rueppel algorithm. Second, E1 Block is used to perform the authentication which is based on Secure and Fast Encryption Routine (SAFER+) algorithm. Third, E21 or E22 block performs pairing feature which is also based on SAFER+ algorithms. Finally, E3 block which is used to generate encryption key as found in [13].

It is recognized that encryption, authentication and pairing blocks are used the Bluetooth address (BD_ADDR) which is 48-bit IEEE address and unique for each Bluetooth unit. The Bluetooth addresses are publicly known, and can be obtained via MMI interactions, or, automatically, via an inquiry routine by a Bluetooth unit as found in [14].

VII. FPGA Implementation Performance Analaysis

The whole security blocks RTL are coded using Verilog and simulations done using Vivado 2018.2 IDE.

A. GSM Security chain simulation results

A.1 *The COMP128v1 timing results*

Every **single pass** requires **145** cycles to finish and the whole top level A3-A8 Authentication block implementing the **COMP128v1** algorithm needs **1160** cycles to finish the computation of the encryption key Kc and the SRES.

A.2 *The A5/1 timing results*

The RTL Verilog behavioral simulation which also coincides the post implementation timing simulation needs 187 cycle from the beginning of asserting Start_A51 for the initialization of the LFSRs: 1 cycle to put zeros, 64 cycles to load the Kc, and 22 cycles to load frame number followed by 100 cycles for irregular clocking to do mixing, Then the RTL takes exactly 114 cycle to produce the output ciphering stream so a total 301 cycles.

The F8 Confidentiality algorithm needs 1636 cycles to be finished for input plaintext of size 256 , this is expected as the input plain text is treated as 4 blocks, each of 64-bit size, so total number of KASUMI's needed to be done are one KASUMI computation for the initialization, and four KASUMI computations each after other for each block key stream computations and as each KASUMI computation needs 323 cycle to be finished and there's extra 4 cycles between each KASUMI, so is reasonable to need 1635 cycles

VIII. SDR VERSUS SECURITY FUNCTIONS UTILIZATION AND POWER CONSUMPTION COMPARISON.

The synthesis of the whole COMP 128v1 algorithm configures contains only LUTs, registers, and F7 muxes, no other FPGA components (i.e. block RAM – DSPs -… etc.) is needed for this block, both the whole top-level of the COMP128 (A3 and A8 are included) and the A5/1 utilization are only 2.44% of the ZYNQ XC7Z020 or 1301 LUTs.

The utilization of the F8 confidentiality algorithm is proved to be on slice LUTs and slice registers only and summing up 15.71% of the ZYNQ XC7Z020 or 8357 LUTs.

While the largest utilization of the security modules is counted for the Bluetooth part SAFER+ based reaching 18082 LUTs, when comparing the security modules utilization against the communications blocks transmitters and recievers utilization we found 2G security functions represent only 2.1% of the 2G SDR , while 3G Security module consumes 13% of the SDR and the Bluetooth SAFER+ based consumes 29.9%.

The utilization results are summarized in figure 2 against the whole SDR in[4]

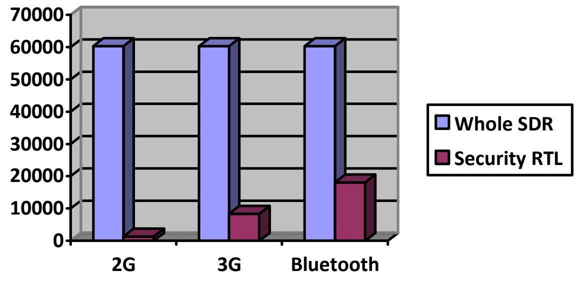

Fig. 2: Area utilization of security functions vs. the whole DSR.

The power consumption comparison, illustrated in figure 3 also show a small percentage of power consumption for the security modules equals 4.7%, 17.8%, and 20.86% of the 2G, 3G and Bluetooth security functions repressively, when compared to the whole SDR in [4].

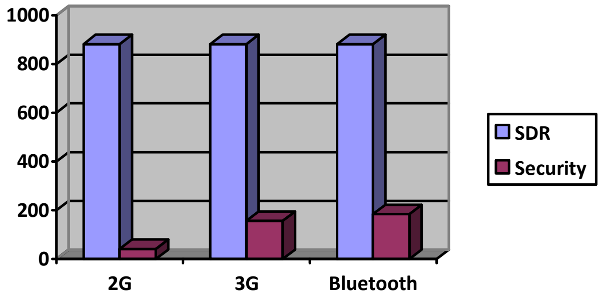

Fig.3: Power consumption (mWatt) of the security functions vs. the whole SDR.

The final conclusion is the proposed co-implementation of the whole security functions in the FPGA consumes very little utilization, power consumption and provides fast clocking response required in the real time communications security.

ACKNOWLEDGMENT

This work was partially funded by ONE Lab at Zewail City of Science and Technology and Cairo University, NTRA, ITIDA, ASRT, Mentor Graphics, NSERC.

REFERENCES

[1] Sadek, A., Mostafa, H., Nassar, A., & Ismail, Y. "Towards the implementation of Multi-band Multi-standard Software-Defined Radio using Dynamic Partial Reconfiguration.", 2017 International Journal of Communication Systems, 30(17), e3342. doi:10.1002/dac.3342.

[2] Khaled Khatib , Mostafa Ahmed , Ahmed Kamaleldin , Mohamed Abdelghany and Hassan Mostafa. "Dynamically Reconfigurable Power Efficient Security for Internet of Things Devices." 2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST).

[3] Mohamed A. Bahnasawi, et al. "ASIC-Oriented Comparative Review of Hardware Security Algorithms for Internet of Things Applications" 2016 28th International Conference on Microelectronics (ICM). doi:10.1109/icm.2016.7847871.

[4] Hosny, Sherif, et al. "A Software Defined Radio Transceiver Based on Dynamic Partial Reconfiguration." 2018 New Generation of CAS (NGCAS), 2018, doi:10.1109/ngcas.2018.8572253.

[5] Yasir, et al. "Performance Comparison of KASUMI and Hardware Architecture Optimization of f8 and f9 Algorithms for 3g UMTS Networks." 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2017, doi:10.1109/ibcast.2017.7868088.

[6] Yasir, et al. "Compact and High Speed Architectures of KASUMI Block Cipher." SpringerLink, Springer, Dordrecht, 22 Feb. 2018, link.springer.com/article/10.1007/s11277-018-5606-8.

[7] J.Umesh rao. "Implementation of Data Encryption & Decryption for the Safer+ Algorithm Using Verilog HDL" International Journal of Scientific and Engineering Research Volume 3, issue 4, April-2012-ISSN 2229-551

[8] Heine, Gunnar, and Holger Sagkob. GPRS: Gateway to Third Generation Mobile Networks. Artech House, 2003, Page 50, 'How to integrate the A5/1 to the physical layer of the GSM'.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[9] Briceno, Marc; Goldberg, Ian; Wagner, David (1998), Implementation of COMP128, archived from the original on 2009-03-18.

[10] Digital cellular telecommunications system (Phase 2+);Universal Mobile Telecommunications System (UMTS);3G security; Security architecture (3GPP TS 33.102 version 11.5.1 Release 11), V11.5.1 (2013-07).

[11] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security;Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 3: Implementors' Test Data' (Release 7), V7.0.0 (2007-06)

[12] D Dhebar, The mechanics of 3G cryptography: a step-by-step guide to every original input-tooutput algorithm of 3G/UMTS, Silverykey Books, 2010.

[13] Gehrmann, Christian, et al. Bluetooth security. Artech House, 2004.

[14] "Core Specifications | Bluetooth Technology Website." Bluetooth, www.bluetooth.com/specifications/bluetooth-core-specification.