An End-to-End Hardware Approach Security for the GPRS

P. Kitsos, N. Sklavos and O. Koufopavlou
University of Patras/Electrical and Computer Engineering Department, Patras, Greece
e-mail: pkitsos@ee.upatras.gr

***Abstract*—An end-to-end security architecture and its VLSI implementation for the GPRS is proposed in this paper. The security offered by GPRS is similar to that offered by the Global Mobile System (GSM). Three algorithms are needed. The A3 and A8, for authentication and ciphering key generation, and the GEA3 algorithm for data confidentiality. The A3 and A8 are based on the RIJNDAEL block cipher, while the GEA3 is based on the KASUMI block cipher. For both ciphers efficient implementations are proposed. The whole design was coded using VHDL language and for the hardware implementations of the designs FPGA devices were used. Detailed analysis is shown, in terms of frequency, throughput, and covered area.**

I. INTRODUCTION

The General Packet Radio Services (GPRS) offers to the users continuous connection to Internet and Intranet. Some of the services may require high level of security, for example the financial transaction over the Internet. The GPRS has inherited most of the security threats that exists in the Global Mobile System (GSM) system. In addition the GPRS encounters new and great challenges. This since GPRS employs IP technology and it is connected to the Internet. The technical security offered by GPRS is similar to that offered by the GSM. Confidentiality, integrity and authentication are the services that devices and networks should cover [1].

In order to cover the GPRS security features three algorithms are used. The A3 algorithm [2] is used for authentication procedure, the A8 algorithm [2] is used for encryption key generation, and finally the GEA3 algorithm [3] is used for data confidentiality. The A3 and A8 algorithms are based on the RIJNDAEL block cipher [4], while the GEA3 algorithm is based on the KASUMI block cipher [5]. The performance of the proposed RIJNDAEL block cipher implementation is slight slower than other previous designs [6]-[9] in terms of throughput, but the implementation is compact enough in order to integrate better in the Subscriber Identification Card (SIM). The GEA3 algorithm is integrated in the mobile equipment and is used for bulk encryption. So, the performance demands are very high and an efficient implementation of the KASUMI block cipher is needed. The proposed GEA3 and KASUMI implementations outperforms all the previous published designs [10]-[14].

The paper is organized as follows: In section II the GPRS security architecture is described. In section III the proposed GPRS security VLSI implementation is described. The synthesis results for the FPGA implementation are shown in section IV, and the paper conclusions are given in section V.

II. GPRS SECURITY FEATURES

The SIM contains the identity of the subscriber. A Mobile Equipment (ME) with the SIM inserted they together form a Mobile Station (MS). The primary function of the SIM is to authenticate an MS before it gets access to the network. The SIM contains the Individual Subscriber Authentication Key Ki, the ciphering key generating algorithm (A8), the authentication algorithm (A3), as well as a Personal Identification Number (PIN). The GEA3 algorithm is implemented in the ME. Figure 1, shows the block diagram of the GPRS security in the MS.

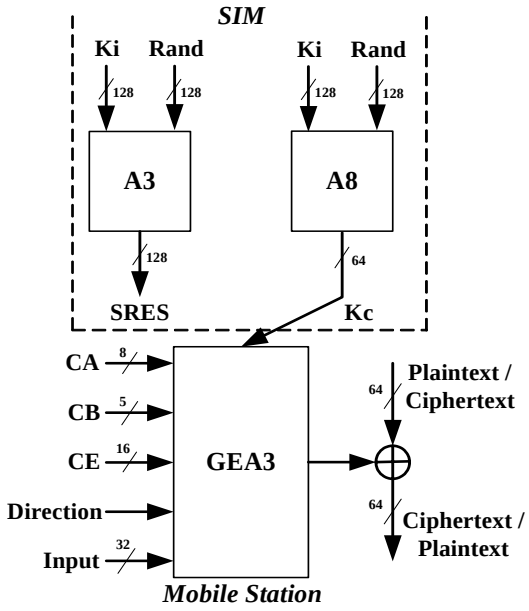

Figure 1. The GPRS security block diagram

The Ki is 128 bits. The purpose of the algorithm A3 is to allow authentication of a mobile subscriber's identity. The algorithm A3 must compute an expected response SRES from a random challenge RAND sent by the network. For this calculation, algorithm A3 is used with the secret authentication key Ki. If the authentication is passed, the A8 algorithm uses the Ki with the 128 bits authentication RAND to generate the 64 bits Ciphering Key, Kc. The GEA3 algorithm is integrated in the ME and is used for encryption the data during a data transfer under the ciphering key, Kc. This algorithm uses the *Input* and *Direction* for synchronization purpose. In addition some predefine constants, *CA*, *CB*, and *CE* are used.

III. GPRS SECURITY VLSI IMPLEMENTATION

For the implementation of algorithms A3 and A8 the *f2*, *f3*, *f4*, functions of the UMTS MILENAGE [15] are used. The implementation of these algorithms is shown in Fig. 2.

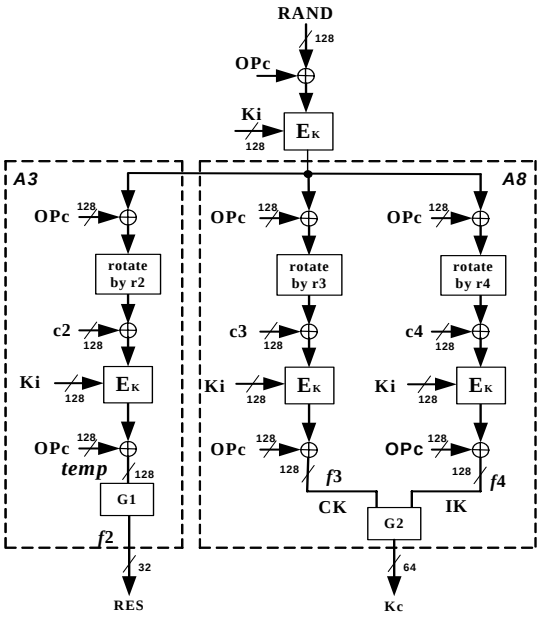

Figure 2. The A3 and A8 architectures

The constants ci, are stored and accessed from the ROM blocks. The OPc value is stored and accessed from the RAM. With E_K the RIJNDAEL cipher is denoted. In the A3 algorithm, the *temp* signal is equal to 128-bit. For the *SRES* production the 64 least significant bits are used, by the function *G1* in the following way: SRES=temp(0 to 31) XOR temp(32 to 63). For the *Kc* production, the outputs of the *f3* and *f4* function are used, by the function *G2*, in the following way: Kc=CK(0 to 63) XOR CK(64 to 127) XOR IK(0 to 63) XOR IK(64 to 127).

The GEA3 is a stream cipher that encrypts/decrypts blocks of data, between 1 to M bytes (max. 65536 bytes) in length, by using a ciphering key K'_C. The K'_C is defined as $K'_C = K_C \| K_C$. The GEA3 stream generator is based on a KASUMI cipher in a form of Output Feedback Mode (OFB) [16], and generates the output Key stream in multiples of 64 bits. The implementation of the GEA3 algorithm is illustrated in Fig. 3. The GEA3 data mapping pads the KASUMI initial value and set the value of the counter BLKCNT. The CA, CB, and CE parameters are fixed and stored in the data mapping subunit. At the initialization phase, the system parameters *CA*, *CB*, *Input*, *CE*, and *Direction* are padded in order to make a 64-bits *Initial Input*. During the initialization process (first loop execution) the MUX subunit selects the IN1 (*Initial Input*) and the KASUMI produces the initial Key Stream (KS) by using the modified K'c. This initial KS is stored in a register and used for the next iterations. In all the next iterations, the MUX selects the second input (IN2) and the K'c is used by the KASUMI. The Block Count (BLKCNT) counter is set initially to 0, and after each iteration, is increased by one. The

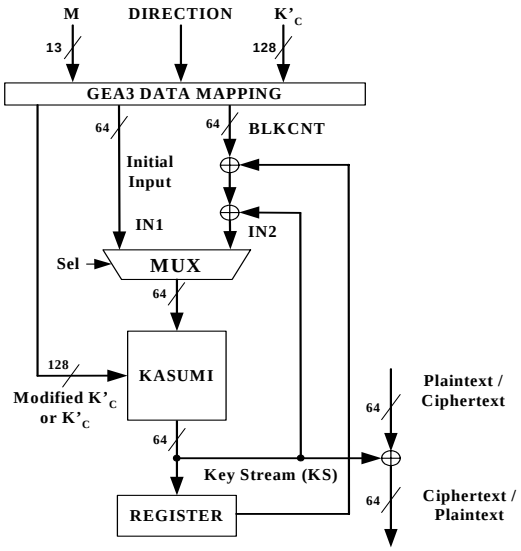

Figure 3. The GEA3 algorithm implementation

maximum value of the counter is (8M/64), which is the number of iterations. The input M defines the plaintext/ciphertext length (# of bits).

A. RIJNDAEL Block Cipher implementation

The proposed hardware implementation of the RIJNDAEL block cipher is shown in Fig. 4. This implementation is similar to the [17], but with reduced the hardware resources. The different transformations of the algorithm architecture operate on the intermediate result, called State. The State can be pictured as a rectangular array of bytes. This array has four rows. The number of columns (Nb) is equal to the block length divided by 32. The Key is also considered as a rectangular array with the same number of rows as State. The number of columns (Nk) is equal to the key length divided by 32. The number of rounds (Nr), depends on the values Nb and Nk. For block and key length equal to

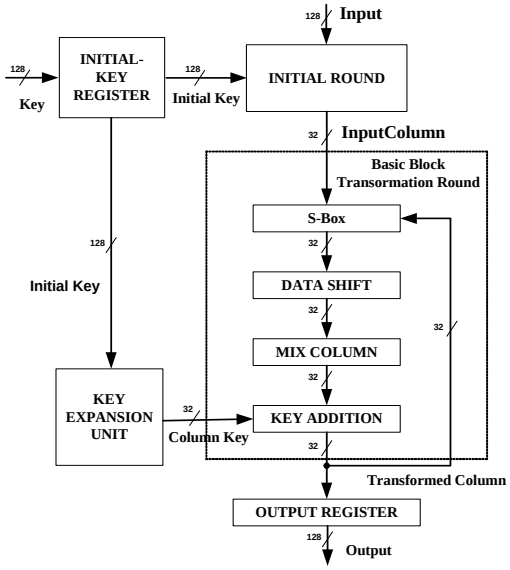

Figure 4. The RIJNDAEL block cipher implementation

128 bits both values of Nb and Nk are equal to 4 and the Nr is defined as 10. The proposed architecture consists of the Key Expansion unit, the Basic Block Transformation Round, the Initial Round and the appropriate registers. 41 clock cycles are needed for the completion of a 128-bit plaintext transformation. The Basic Block Transformation Round is composed of four building blocks: S-BOXes, Data Shift, Mix Column and Key Addition. In order to achieve high-speed performance the S-BOXes are implemented by ROM. In the proposed implementation four [256x8]-bit ROM blocks were used. The implementation of the S-BOXes requires the implementation of two different mathematical functions: 1) the multiplicative inverse of each byte of the State in the finite field GF(2^8) and 2) an affine mapping transformation over GF(2). The multiplicative inverse function produces a byte, which is the input of the affine mapping transformation function. This is defined as: Out[i] = In[i] XOR In[(i+4) mod 8] XOR In[(i+5) mod 8] XOR In[(i+6) mod 8] XOR In[(i+7) mod 8] XOR C(i) where In[i] is the i-th bit of the input byte, and C(i) is the i-th bit of a byte predefined constant C, as the algorithm specifications defines.

B. KASUMI Block Cipher Implementation

The proposed KASUMI cipher consists of the two main components. The Key Scheduling Unit, which is responsible for the round keys generation, and the KASUMI Core, which executes the basic encryption procedure. The KASUMI Core implementation uses two pipeline stages. The even round of KASUMI cipher has different structure of the odd round. The odd rounds are denoted as Odd Round Cell (ORC) and the even rounds are denoted as Even Round Cell (ERC).

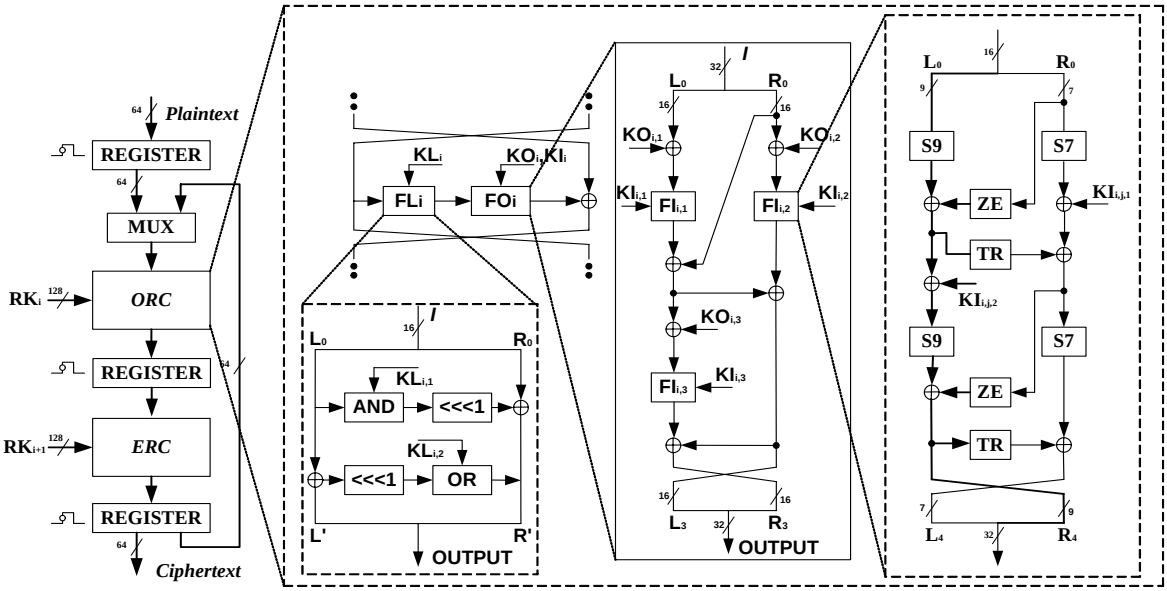

Figure 5. The KASUMI block cipher implementation

Figure 5 shows the implementations of the KASUMI and ORC. In the ERC the order of the functions FLi and FOi is reversed. As it is previously mentioned the GEA3 algorithm uses the KASUMI cipher in OFB mode of operation. This mode, in order to work correctly, demands the output block of the previous KASUMI execution. So, the pipeline technique is used only in order to decrease the critical path and only one data block can process at any time. The proposed Key Expansion Unit architecture is implemented by shift registers in order to produce a number of sub-keys. The rest of the sub-keys are generated by bit-wise XOR operations with the constants Cj. These constants are stored in the 8x16 bits ROM memory. At total 40 16-bit sub-keys are generated. With the appropriate concatenations of the sub-keys the round keys are generated. The round keys are computed and stored in a 52x16 bit register file.

IV. Synthesis Results and Evaluation

The proposed architecture (Fig. 1) was captured by using VHDL with structural description logic. The VHDL code was simulated and verified by using the official test vectors, provided by the 3GPP standard [15], [18].

The synthesis results of the proposed RIJNDAEL block cipher and the A3/8 unit are shown in Table I. The FPGA device XILINX V400E-FG676 was used.

TABLE I.
RIJNDAEL AND A3/8 UNIT IMPLEMENTATION RESULTS

	RIJNDAEL block cipher	**A3/A8 Unit**
Function Generators	2387	9548
CLBs	1194	4750
D Flip Flop	715	2960
F (MHz)	78	70
Throughput (Mb/s)	243	218

The performance comparison with previous published works is shown in Table II. In addition, the synthesis results of the proposed GEA3 and KASUMI block cipher implementations, are shown in Table III. The FPGA device XILINX V200E-FG456 was used.

TABLE II.
RIJNDAEL BLOCK CIPHER IMPLEMENTATIONS PERFORMANCE MEASUREMENTS

Architectures	F (MHz)	Throughput (Mb/s)
[6]	14,1/31,8	300/1940
[7]	-	353
[8]	25,9	331
[9]	-	750 (BEST)
Proposed	78.3	244

TABLE III.
KASUMI AND GEA3 IMPLEMENTATION RESULTS

	KASUMI Block Cipher	GEA3 Algorithm
Function Generators	2442	2687
CLBs	768	900
D Flip Flop	1405	1623
F (MHz)	34	33
Throughput (Mb/s)	544	363

Performance comparisons between the proposed KASUMI cipher implementation and implementations in [10]-[14] are given in Table IV.

TABLE IV.
KASUMI TIME PERFORMANCE COMPARISONS

Architecture	F (MHz)	Throughput (Mb/s)
[10]-[11]	35.35	70.70
[12]	20	110
[12]	60	410
[13]	33.14	265.12
[13]	28.38	227.04
[14]	7.3	233.6 (BEST)
Proposed	34	544

The GEA3 algorithm is almost the same with the UMTS algorithm *f*8. Because no other previous GEA3 implementations are referred, comparisons with the previous *f*8 implementation are made (see Table V).

Table V.
GEA3 Time Performance Comparisons

Architectures	F (MHz)	Throughput (Mb/s)
[10]	33.14	53
[11]	46.56	73.5
[12]	19.5	103
[12]	52	321
[13]	30.12	238
[13]	25.80	204
Proposed	33	363

V. CONCLUSIONS

A hardware implementation of the GPRS security was presented in this paper. The proposed system performs all the necessary security features that GPRS demands. The main architectural units of the system are based on the RIJNDAEL and KASUMI block ciphers. Efficient implementations for both ciphers are proposed. The system was synthesized, placed, and routed by using FPGA devices. It is an efficient design for devices with GPRS applications.

REFERENCES

[1] 3GPP TS 43.020 V4.0.0 3rd Generation Partnership Project; Technical Specification Group Services and system Aspects; Security related network functions

[2] ETSI/SAGE. Specification of the MILENAGE-2G Algorithms: an Example Algorithm Set for the GSM Authentication and Key Generation Functions A3 and A8. ETSI/SAGE, May 2002.

[3] ETSI/SAGE. Specification of the A5/3 Encryption Algorithms for GSM and EDGE, and GEA3 Encryption Algorithm for GPRS, Document 1: A5/3 and GEA3 Specifications. ETSI/SAGE, May 2002.

[4] Joan Daemen and Vincent Rijmen: "AES Proposal: Rijndael", http://csrc.nist.gov/encryption/aes/round2/AESAlgs/Rijndael/Rijndael.pdf.

[5] KASUMI specification, Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 2, ETSI/SAGE, December 1999.

[6] A. J. Elbirt, W. Yip, B. Chetwynd, C. Paar, "An FPGA Based Performance Evaluation of the AES Block Cipher Candidate Algorithm finalists", 3rd Advanced Encryption Standard (AES) Candidate Conference, New York, USA, April 13-14, 2000.

[7] A. Dandalis, V.K. Prasanna, J.D.P. Rolim, " A Comparative Study of Performance of AES Final Candidates Using FPGAs", 3rd Advanced Encryption Standard (AES) Candidate Conference, New York, April 13-14, 2000.

[8] K. Gaj and P. Chodowiec, "Comparison of the Hardware Performance of the AES candidates using Reconfigurable Hardware", 3rd Advanced Encryption Standard (AES) Candidate Conference, New York, USA, April 13-14, 2000.

[9] V. Fischer and M. Drutarovsky, "Two Methods of Rijndael Implementation in Reconfigurable Hardware", CHESS 2001, France, May 14-16, 2001.

[10] K. Marinis, N. K. Moshopoulos, F. Karoubalis, and K. Z. Pekmestzi, "On the Hardware Implementation of the 3GPP Confidentiality and Integrity Algorithms", 4th International Conference for the Information Security, ISC 2001 Malaga, Spain, pp. 248-265, October 1-3, 2001.

[11] K. Marinis, N. K. Moshopoulos, F. Karoubalis, and K. Z. Pekmestzi, "An Area Optimized Hardware Implementation of the 3GPP Confidentiality and Integrity Algorithms", 8th Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2002, Brasov, Romania, May 16-17, 2002.

[12] HoWon Kim, YongJe Choi, MooSeop Kim, and HeuiSu Ryu, "Hardware Implementation of 3GPP KASUMI Crypto Algorithm," The 2002 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Vol 1., pp. 317 - 320, July 16-19, 2002, Phuket, Thailand.

[13] Akashi Satoh, Sumio Morioka, "Small and High-Speed Hardware Architectures for the 3GPP Standard Cipher KASUMI", 5th International Conference Information Security, ISC 2002 Sao Paulo, Brazil, September 30 - October 2, 2002, LNCS 2433 Springer 2002.

[14] Guy-Armand Kamendje, "FPGA Architectures for High Speed UMTS Encryption", 2nd Asian International Mobile Computing Conference (AMOC 2002), 14-17 May 2002, Malaysia.

[15] 3GPP TS 35.206 V4.0.0, Technical Specification Group Services and System Aspects, 3G Security, Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*: Document 1: General, April 2001.

[16] "Recommendation for Block Cipher Modes of Operation. Methods and Techniques". NIST, Technology Administration. http://csrc.nist.gov/encryption/modes/Recommendation/Modes01.pdf.

[17] N. Sklavos, and O. Koufopavlou, Architectures and VLSI Implementations of the AES-Proposal Rijndael", IEEE Transaction on Computers, Vol. 51, No. 12, December 2002, pp. 1454-1455.

[18] ETSI/SAGE. Specification of the A5/3 Encryption Algorithms for GSM and EDGE, and GEA3 Encryption Algorithm for GPRS, Document 2: Implementators' Test Data. ETSI/SAGE, May 2002.