International Journal of Information & Network Security (IJINS)
Vol.1, No.2, June 2012, pp. 119~126
ISSN: 2089-3299

A Reconfigurable Cryptography Coprocessor RCC for Advanced Encryption Standard AES/Rijndael

S. El Adib, N. Raissouni, A. Chahboun, A. Azyat, M. Lahraoua, N. Ben Achhab, A. Abbous, O. Benarchid
Innovation & Telecoms Engineering Research Group. Remote Sensing & Mobile GIS Unit.
University Abdelmalek Essaadi. Mhannech II, B.P 2121 Tetuan, Morocco

Article Info

Article history:

Received Jun 12th, 2012
Revised Jun 20th, 2012
Accepted Jun 26th, 2012

Keyword:

Security
AES
FPGA
Reconfig. Crypto Processor
VHDL-Xilinx

ABSTRACT

The market trend of secure products is to offer more users' services and security. Thus, electronic devices must be flexible and reconfigurable in the way they permit executing further algorithms than those designed for. In this paper, in order to encrypt/decrypt data blocks, a Reconfigurable Cryptography Coprocessor (RCC) for Advanced Encryption Standard (AES/Rijndael) is developed. The AES offers a good combination of security, performance, efficiency, implementability and flexibility. We propose a RCC by using a Systolic Processor (SP) based on: i) Processing Element (PE) array, and ii) Controller with a Finite State Machine (FSM) and a memory. The advantages are: i) provide a solution to compute all matrix format data and ii) the PE array's data path is reconfigurable via the FSM. Finally, the conception and implementation were carried out by using Very High Speed Integrated Circuit Hardware Description (VHDL) language and Xilinx ISE 7.1 simulator.

Corresponding Author:

Naoufal Raissouni, Ph.D,
National School for Applied Sciences of Tetuan, Abdelmalek Essaadi University,
Innovation & Telecoms Engineering Research Group. Remote Sensing & Mobile GIS Unit,
Mhannech II, B.P 2121 Tetuan, Morocco.
Email: nraissouni@uae.ma

1. INTRODUCTION

Security of data is becoming an important challenge for a wide spectrum of applications, including communication systems secure storage supports, digital video recorders, smart cards, cellular phones [1]. Most of encryption and decryption models are implemented for specific algorithm. It is easy to implement hardware for a single algorithm. With such models, it is not possible to treat different encryption algorithms [2], furthermore the corresponding market is now oriented towards more flexibility. Thus, electronic devices must be flexible and reconfigurable in the way they permit executing further algorithms than those designed for. The main objective of the present paper is to design a module of a reconfigurable cryptographic coprocessor capable of executing on Advanced Encryption Standard (AES/Rijndael) [3], [4], [5] and using the basic arithmetic and logic operations. VHDL and logic synthesis tools have been used to design RCC. RCC Architecture is based on 4*x*4 Processing Elements (PE) systolic array [6]; it belongs to the class of the flexible hardware implementations, and allows a user implementing other cryptographic algorithm under specific conditions. Finally, test results and performance evaluation is presented.

2. ADVANCED ENCRYPTION STANDARD AES ALGORITHM

An encryption algorithm converts a plain text message into cipher text message which can be recovered only by authorized receiver using a decryption technique. The AES-Rijndael [3], [4] algorithm is an iterative private key symmetric block cipher. The input and output for the AES algorithm each consist of sequences of 128 bits (block length). Hence, N_b = Block length/32 = 4. The Cipher Key for the AES algorithm is a sequence of 128, 192 or 256 bits (Key length), in our environment always 128 bits, is ordered in a similar fashion. The algorithm consists of a single AddRoundKey, using the cipher key, followed by 9 regular rounds each consisting of 4 steps and a final round. The steps of the regular rounds are: Sub Bytes, Shift Rows, Mix Column and AddRoundKey. The final round skips the Mix Column step. Every round requires its own round key. These keys can be generated in advance or one per round by adding an extra round step.

2.1. SubBytesTransformation

Each input byte of the state matrix is independently replaced by another byte from a look-up table called Sbox [7]. Sbox is a 256-entry table composed of two transformations: First each input byte is replaced with its multiplicative inverse in $GF(2^8)$ [8] with the element $\{00\}$ being mapped onto itself; followed by an affine transformation over $GF(2^8)$.

2.2. ShiftRows Transformation

Cyclically shifts bytes in each row by a certain offset. First row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, third and fourth rows are shifted by offsets of two and three respectively. In this way, each column of output state of Shift Rows step is composed of bytes from each column of input state.

2.3. MixColumns Transformation

Operates on the State column-by-column, treating each column as four-term polynomial. Columns are considered as polynomials over $GF(2^8)$ and multiplied by modulo x^4+1 with fixed polynomial: $a(x) = \{03\}\ x^3 + \{01\}\ x^2 + \{02\}\ x$ as given by [9].

2.4. AddRoundKey Transformation

128-bits of state are XORed with 128-bits of round key. The operation is viewed as column wise operation between 4 bytes of state column and one word of round key. Each round key is 4-word (128-bits) array generated as product of previous round key: a constant that changes each round, and series of S-Box lookups for each 32-bits word of the key. Key schedule Expansion generates a total of $N_b\ (N_r + 1)$ words; where *Nr* is number of rows.

Decryption process is direct inverse of encryption process. All transformations applied in encryption process are inversely applied to this process. Hence, last round values of both data and key are first round inputs for decryption process and follows in decreasing order. AES decryption can be performed by using same algorithm flow. However, all four steps in round transformation are replaced with their own inverses and round keys for encryptions are used in reverse order.

3. RECONFIGURABLE CRYPTOGRAPHIC COPROCESSOR (RCC) ARCHITECTURE

Reconfigurable Cryptographic Coprocessor is presented (RCC) may encrypt or decrypt data by using encryption algorithm. RCC is built around systolic array of processing elements (PEs) [10], [12], Control Unit and Memory Unit. RCC encrypts and decrypts data efficiently and flexibly. RCC is designed to provide an efficient way to implement the Advanced Encryption Standard (AES/Rijndael) (see Figure 1).

3.1. Memory Unit Module

Memory Unit Module is consisting of 4 RAMS working in parallel (see Figure 1). Memory can store 32-bits data and instruction words. Data include encrypting or decrypting I/O data, as well as all data needed by encrypting/decrypting process. Therfore, AES; Sbox, INVSbox, logarithm and anti-logarithm tables and round keys are stored in RAM0, RAM1, RAM2 and RAM3, respectively. Each RAM is 512*x*8-bits dual port allowing better access time and performance.

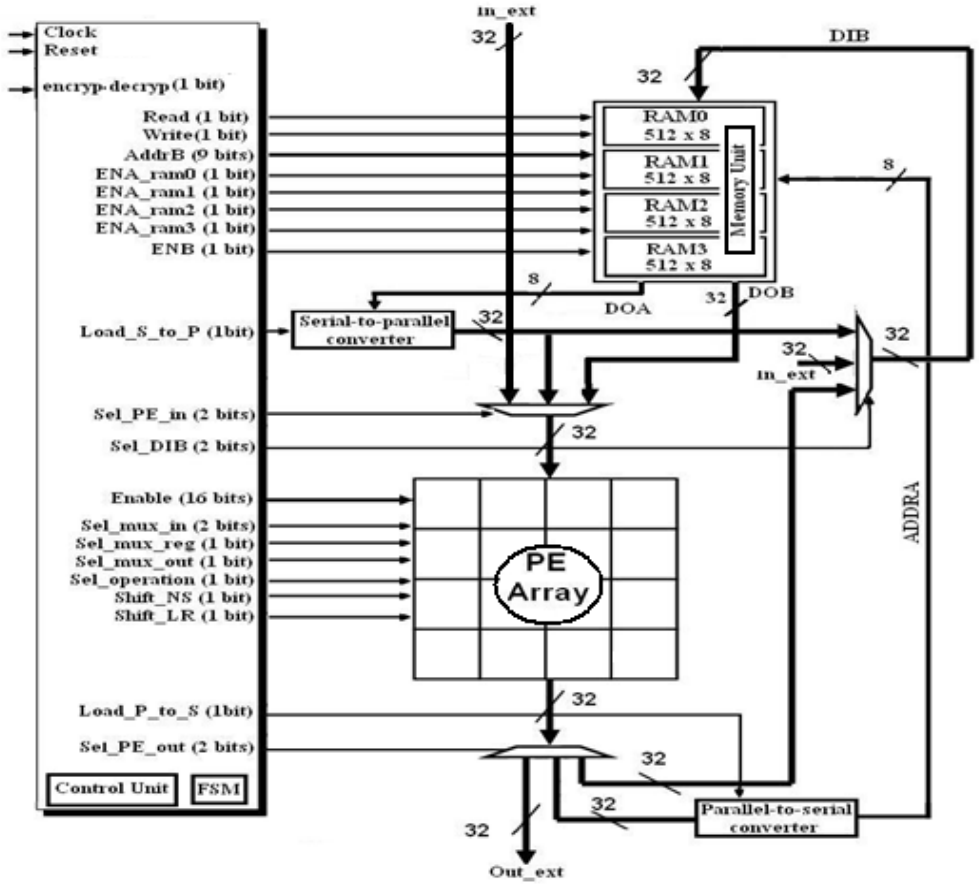

Figure 1. Architecture of the Reconfigurable Cryptographic Coprocessor (RCC)

3.2. Control Unit Module

Control Unit with a Finite State Machine (FSM) is the main module of the architecture. The main function is to control other modules according to instructions stored in memory. FSM controller is responsible for driving RCC either for key expansion or for data encryption (Table 1).

Table 1. Control Unit interface signals

Signal Name	Width (bits)	Type	Description
Encrdecry	1	Input	Select the encryption or decryption
Sel_DIB	2	Output	Select input of 4 RAM (port B)
ADDRB	9	Output	Address the 4 RAM (port B)
ENA_ram0	1	Output	Enable RAM0 (port A)
ENA_ram1	1	Output	Enable RAM1 (port A)
ENA_ram2	1	Output	Enable RAM2 (port A)
ENA_ram3	1	Output	Enable RAM3 (port A)
ENB	1	Output	Enable 4 RAM (port B)
Enable_PE	16	Output	Enable PE (one wire for each PE)
Sel_PE_in	2	Output	Select input of PE array between RAM and external source
Shift_LR	1	Output	Shift byte of each PE into its neighbor, from west to east (or vice-versa)
Shift_NS	1	Output	Shift byte of each PE into its neighbor, from north to south (or vice-versa)
Sel_mux_in	2	Output	Manage input multiplexor inside each PE
Sel_mux_out	1	Output	Manage output multiplexor inside each PE
Sel_mux_reg	1	Output	Manage register multiplexor inside each PE
Load_P_to_S	1	Output	Enable convert data parallel to data serial
Load_S_to_P	1	Output	Enable convert data serial to data parallel
Sel_operation	1	Output	Select the operation (XOR) or addition

3.3. PE array

Processor Element (PE) array is configured to receive input commands and data (to encrypt/decrypt) from the Control Unit (CU), configured to receive and transmit encrypted/decrypted data. It consists of a systolic array of 4x4 Processing Elements (Figure 2). All PE are identical and controlled by CU.

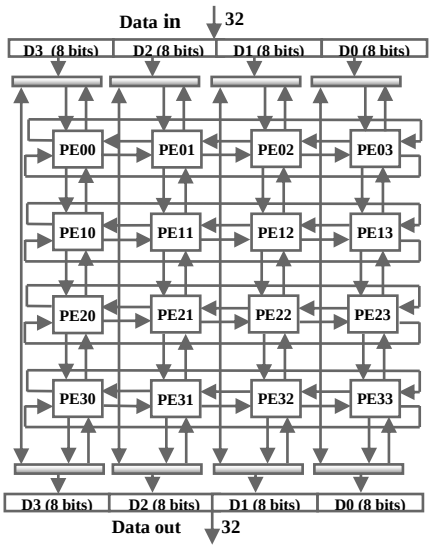

Figure 2. Architecture of Processing Element (PE) array

PE includes I/O ports, input-select multiplexor, an operations unit, register multiplexor, register, 4 tri-state buffer and an output multiplexor (Figure 3):

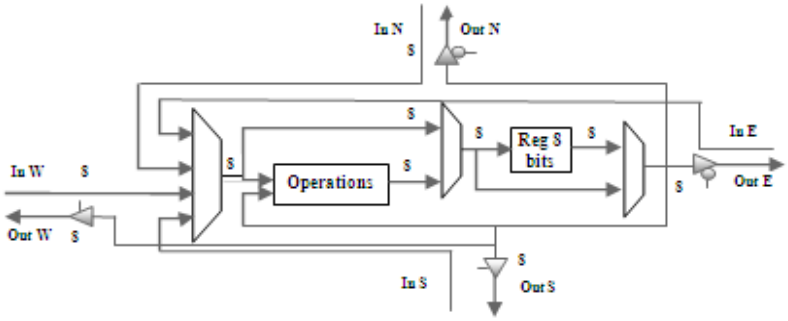

Figure 3. PE Block diagram.

4. AES EXECUTION ON RCC

RCC utilizes AES-128, where the plain text consists of 128-bits data blocks and each block may be managed as a matrix of 4*x*4 bytes. Each element of AES matrix may be mapped to PE of PE matrix. Load row operation copies 32-bits word from external source or Memory unit, and writes it into row of PE array. Encryption process includes 10 rounds involving the following transformations: Subbytes, Shiftrows, Mixcolumns and Add-round-key

4.1. SubBytes transformation

On receiving the encryption request, cipher data and round keys are fetched from memory byte by byte at every cycle. Two data bytes are XORed with each other for add round key transformation. FSM extracts all bytes in PE array byte by byte, is becomes the address of the RAM0 and gets substituted with corresponding data bytes residing at those addresses. Each byte stored in PE array is substituted with corresponding byte in substitution box table (S-Box table) stored in RAM0 at address "FF00". FSM may manage one word of 4 bytes and may provide all needed commands for substitution of all 4*x*4 1-byte data. After byte substitution, data is directed to Shift rows transformation.

4.2. Shift rows transformation

Each row of PEs in PE array is wrapped around in a cylindrical fashion which is considered a circular shift register, particularly useful in shift rows transformation. FSM activates Shift_LR signal to shift all bytes from west to east of PE array. Shift_LR may be active during 4 cycles. Enable signals activate only PE that needs to be shifted: at the first cycle only 2^{nd}, 3^{rd} and 4^{th} rows of the PE are enabled. At second cycle only 3^{rd} and 4^{th} rows of PE are enabled. Finally, at the third cycle row number 4 is enabled. After Shift rows, data is directed to Mixcolumn stage for matrix multiplication.

4.3. MixColumns transformation

Mixcolumn transformation operates on the state column-by-column. RCC performs the following matrix multiplication:

$$S'(x) = A(x) \otimes S(x) \tag{1}$$

$$A(x) = \begin{bmatrix} '02' & '03' & '01' & '01' \\ '01' & '02' & '03' & '01' \\ '01' & '01' & '02' & '03' \\ '03' & '01' & '01' & '02' \end{bmatrix} \tag{2}$$

S(x), data transformed by PE array, and *A*(*x*), matrix of multiplicative vectors. Equation (1) may be performed by using logarithm and anti-logarithm tables (see Tables 2 and 3, respectively).

Table 2. Logarithm Table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0	0	19	1	32	2	1A	C6	4B	C7	1B	68	33	EE	DF	3
1	64	4	E0	E	34	8D	81	EF	4C	71	8	C8	F8	69	1C	C1
2	7D	C2	1D	B5	F9	B9	27	6A	4D	E4	A6	72	9A	C9	9	78
3	65	2F	8A	5	21	F	E1	24	12	F0	82	45	35	93	DA	8E
4	96	8F	DB	BD	36	D0	CE	94	13	5C	D2	F1	40	46	83	38
5	66	DD	FD	30	BF	6	8B	62	B3	25	E2	98	22	88	91	10
6	7E	6E	48	C3	A3	B6	1E	42	3A	6B	28	54	FA	85	3D	BA
7	2B	79	A	15	9B	9F	5E	CA	4E	D4	AC	E5	F3	73	A7	57
8	AF	58	A8	50	F4	EA	D6	74	4F	AE	E9	D5	E7	E6	AD	E8
9	2C	D7	75	7A	EB	16	B	F5	59	CB	5F	B0	9C	A9	51	A0
A	7F	C	F6	6F	17	C4	49	EC	D8	43	1F	2D	A4	76	7B	B7
B	CC	BB	3E	5A	FB	60	B1	86	3B	52	A1	6C	AA	55	29	9D
C	97	B2	87	90	61	BE	DC	FC	BC	95	CF	CD	37	3F	5B	D1
D	53	39	84	3C	41	A2	6D	47	14	2A	9E	5D	56	F2	D3	AB
E	44	11	92	D9	23	20	2E	89	B4	7C	B3	26	77	99	E3	A5
F	67	4A	ED	DE	C5	31	FE	18	D	63	8C	80	C0	F7	70	7

Table 3. Anti-Logarithm Table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	1	3	5	0F	11	33	55	FF	1A	2E	72	96	A1	F8	13	35
1	5F	E1	38	48	D8	73	95	A4	F7	2	6	0A	1E	22	66	AA
2	E5	34	5C	E4	37	59	EB	26	6A	BE	D9	70	90	AB	E6	31
3	53	F5	4	0C	14	3C	44	CC	AF	D1	68	B8	D3	6E	B2	CD
4	4C	D4	67	A9	E0	3B	4D	D7	62	A6	F1	8	18	28	78	88
5	83	9E	B9	D0	6B	BD	DC	7F	81	98	B3	CE	49	DB	76	9A
6	B5	C4	57	F9	10	30	50	F0	0B	1D	27	69	BB	D6	61	A3
7	FE	19	2B	7D	87	92	AD	EC	2F	71	93	AE	E9	20	60	A0
8	FB	16	3A	4E	D2	6D	B7	C2	5D	E7	32	56	FA	15	3F	41
9	C3	5E	E2	3D	47	C9	40	C0	5B	ED	2C	74	9C	BF	DA	75
A	9F	BA	D5	64	AC	EF	2A	7E	82	9D	BC	DF	7A	8E	89	80
B	9B	B6	C1	58	E8	23	65	AF	EA	25	6F	B1	C8	43	C5	54
C	FC	1F	21	63	A5	F4	7	9	1B	2D	77	99	B0	CB	46	CA
D	45	CF	4A	DE	79	8B	86	91	A8	E3	3E	42	C6	51	F3	0E
E	12	36	5A	EE	29	7B	8D	8C	8F	8A	85	94	A7	F2	0D	17
F	39	4B	DD	7C	84	97	A2	FD	1C	24	6C	B4	C7	52	F6	1

$$C = a \times b \tag{3}$$

C can be computed by using logarithm tables in the following way:

$$C = Log'\big((Log\ a) + (Log\ b)\big) \tag{4}$$

PE array are substituted by using logarithm table stored in RAM2 (Table 2) at address "FF00". Data are fetched from PE array byte by byte, is becomes the address of the RAM2 and gets substituted with corresponding data bytes residing at those addresses. Each byte stored in PE array is substituted with corresponding byte in logarithm table. FSM extracts logarithm of all bytes in PE array, adding by substituted bytes to matrix (2) and writes bytes to PE array. Logarithms are then copied to Memory Unit for further computations, PE array may be XORed by columns and written into first row of PEs. Results may be stored in Memory unit. Once all 4x4 bytes computed, FSM copy PE logarithms previously saved in memory unit. According to (4), FSM may substitute them by using antilogarithm table stored in RAM3 (Table 3).

4.4. Add-Round-Key transformation

Final transformation of given round combines key value with transformed data. Keys are loaded from Memory Unit into PE array and XORed with data stored in PE array. This completes one round and output of AddRoundKey is written back into Memory Unit. Finally, at the end of 10 rounds, the fully encrypted data is available in memory.

5. IMPLEMENTATION RESULTS

The synthesizable Cryptography Coprocessor core was described in VHDL using ModelSim 6.4b simulator and synthesized using Xilinx ISE7.1i. The target device selected was Xilinx XC4VLX25 [12]. The synthesis results show that the synthesized device uses only 2252 slices. The device uses only 4 BRAMS 512x8-bits dual-port wide and can be operated at a clock frequency of 189.215 MHz. Throughput reaches value of 43.25 Mbps for encryption. The synthesis and mapping results of Cryptography Coprocessor design are summarized in Table 4.

Table 4. Results of FPGA implementation of Reconfigurable coprocessor cryptography

Target FPGA device	Virtex XC4VLX25-10
Max. clock frequency	189.215 MHz
Number of Slices	2252
Number of Slice Flip Flops	1700
Number of 4 input LUTs	2096
Block RAMS	4
Encryption throughput	43.25 Mbps

6. CONCLUSION

Reconfigurable Cryptographic Coprocessor (RCC) has been designed for Advanced Encryption Standard (AES-Rijndael) algorithm to encrypt/decrypt data. Both encryption and decryption operations were carried out. Results validate the functionality and operability of the proposed coprocessor. The RCC is capable of using other types of Symmetric Block Cipher Algorithms (SBCA). Reconfigurability characteristics also allow the processor to be updated with new algorithms, as well as other applications. Synthesizable RCC core was described in VHDL using ModelSim 6.4b simulator and synthesized using Xilinx ISE7.1i. Finally, throughput reaches value of 43.25 Mbps for encryption with XC4VLX25 Device of Xilinx Virtex Family.

ACKNOWLEDGEMENTS

This work was supported in part by the Ministry for Higher Education, Management Training and Scientific Research under CSPT Grants for "Integration and application of GIS and GPS on mobile systems" and "Ad-hoc wireless sensor networks for remote sensing algorithm validation" projects. The authors would like to express gratitude to external anonymous referees whose comments and suggestions improved this manuscript.

REFERENCES

[1] C. Mucci, L. Vanzolini, A. Lodi, A. Deledda, R. Guerrieri, "Implementation of AES/Rijndael on a dynamically reconfigurable architecture", 978-3-9810801-2-4/DATE07 © 2007 EDAA A. Author, Book title. Publisher, Place (Year).
[2] M. Askar, T.Egemen, "Design and SystemC Implementation of a Crypto Processor for AES and DES Algorithms", *Information Security Cryptology Conference With International Participation*, Decembre 2007, pages (pp. 145-149).
[3] J. Daemen, V.Rijmen, "The Rijndael Block Cipher" , AES proposal, *First Candidate conference (AESI)*, August 20-22, 1998.
[4] J. Daemen, V. Rijmen, "The Design of Rijndael, AES-The Advanced Encryption Standard", *Springer-Verlag Berling Heidelberg* New York 2002.
[5] "Specication for the advanced encryption standard (AES)", *Federal Information Processing Standards Publication* 197, 2001.
[6] S. Sharma, B. Sudarshan, "Design of an Efficient Architecture for Advanced Encryption Standard Algorithm Using Systolic Structures", *International Conference of High Performance Computing HiPC* 2005).
[7] M. T. Tran, D. K. Bui and A. D. Duong, "Gray S-Box for Advanced Encryption Standard", in Proc. of *International Conference on Computational Intelligence and Security*, Vol. 1, pp. 253-258, 2008.
[8] M. Jung, et al. "A Reconfigurable Coprocessor for Finite Field Multiplication in GF(28) ", In *IEEE Workshop on Heterogeneous reconfigurable SoCs*, 2003.
[9] Jarvinen et al. "A fully pipelined memory less 17.8 Gbps AES-128 encryptor", *International Symposium on Field Programmable Gate Arrays*. In 2003 ACM/SIGDA 11th International.
[10] P. Frison, E. Gautrin, D. Lavenier, and J. L. Scharbarg, "Réseaux systoliques spécifiques à base du processeur", *Institut National de Recherche en Informatique et en Automatique (INRIA)*, France, Tech. Rep., 1990.
[11] S. El Adib and N. Raissouni, "Towards developing a Reconfigurable Cryptography Coprocessor RCC for Symmetric Block Cipher Algorithms SBCA: Application to Wireless Sensor Networks WSN communication security", *EuroMediterranean Scientific Congress on Engineering*, Algeciras 2011.
[12] Xilinx, Inc., "Virtex Field Programmable Gate Arrays", *http://www. xilinx.com*.

BIOGRAPHY OF AUTHORS

Samir El Adib received the degree in Informatics, Electronics, Electrotechnics, and Automatics (IEEA) and M.S. degree in automatic and data processing from University Abdelmalek Essaadi (UAE), Tetuan, Morocco, in 2004 and 2006 respectively. Currently, he is a member of Remote-Sensing & Mobile-GIS Unit/Telecoms Innovation & Engineering Research group. His main research interests are FPGAs in custom-computing applications, and more concretely, applications of reconfigurable hardware to cryptography.

Naoufal Raissouni received the M.S., and Ph.D. degrees in physics from the University of Valencia, Spain, in 1997, and 1999, respectively. He has been a Professor of physics and remote sensing at the National Engineering School for Applied Sciences of the University Abdelmalek Essaadi (UAE) of Tetuan, since 2003. He is also heading the Innovation & Telecoms Engineering research group at the UAE, responsible of the Remote Sensing & Mobile GIS unit. His research interests include atmospheric correction in visible and infrared domains, the retrieval of emissivity and surface temperature from satellite image, huge remote sensing computations, Mobile GIS, Adhoc networks and the development of remote sensing methods for land cover dynamic monitoring.

Asaad Chahboun obtained his degree from Central School of Arts and Businesses in Brussels Belgium, M.S. in Telecommunication Systems from University Abdelmalek Essaadi in 2006. He was Maintenance responsible engineer of radiology and medical imaging materials, at FREELANCE SERVICE enterprise Rabat-Morocco, from 1995 to 2003. Currently, he is a member of Remote-Sensing & Mobile-GIS Unit/Telecoms Innovation & Engineering Research group. His current areas of research are Wireless Sensor Networks, Routing in Ad hoc Network, Network Security, remote sensing, the development of remote sensing methods for land cove dynamic monitoring and Grid computing.

Abdelilah Azyat received the degree in Informatics, Electronics, Electrotechnics, and Automatics (IEEA) and M.S. degree in Bioinformatics from University Abdelmalek Essaadi (UAE), Tetuan, Morocco, in 1995 and 2004 respectively. Currently, he is a member of Remote-Sensing & Mobile-GIS Unit/Telecoms Innovation & Engineering Research group. His research interests include remote sensing, spatiotemporal studies of thermal infrared satellite imagery of the earth's surface and the development of remote sensing methods for land cove dynamic monitoring, GIS and Mobile GIS.

Mohammed Lahraoua received the degree in Chemistry and M.S. degree in Bioinformatics from University Abdelmalek Essaadi (UAE), Tetuan, Morocco, in 1992 and 2004 respectively. Currently, he is a member of Remote-Sensing & Mobile-GIS Unit/Telecoms Innovation & Engineering Research group. His present investigations include remote sensing algorithms, thermal infrared satellite imagery of the earth's surface and the development of remote sensing methods for land cove dynamic monitoring and Grid computing.

Nizar Ben Achhab received the degree in Informatics, Electronics, Electrotechnics, and Automatics (IEEA) and M.S. degree in Bioinformatics from University Abdelmalek Essaadi (UAE), Tetuan, Morocco, in 1995 and 2004 respectively. Currently, he is a member of Remote-Sensing & Mobile-GIS Unit/Telecoms Innovation & Engineering Research group. His present research include remote sensing multitemporal and spatiotemporal studies of thermal infrared satellite imagery of the earth's surface, the insitu LST measurements and the development of remote sensing methods for land cove dynamic monitoring. It includes also, Grid computing, Multithreading, GPGPU, hardware and software methods for time computing optimizations.

Abdellah El Abbous obtained his engineer degree from the National Institute of Telecommunications (INPT) in 2010, Rabat, Morocco. He is currently working toward the Ph.D. He has been with the Remote Sensing & Mobile GIS Research Unit at National School of Applied Sciences, Abdelmalek Essaadi. His main research activities are in the fields of remote sensing, image processing, urban planning and environmental planning. In particular, his interests include buildings extraction, classification, and change detection.

Omar Benarchid received Engineer degree in telecommunications and networks engineering from the National School of Applied Sciences, University of Abdelmalek Essaadi, Tetuan, Morocco, in 2011. He is currently working toward the Ph.D. He has been with the Remote Sensing & Mobile GIS Research Unit at National School of Applied Sciences, Abdelmalek Essaadi. His main research activities are in the fields of remote sensing, image processing, urban planning and environmental planning. In particular, his interests include buildings extraction, classification, and change detection.