Journal of Applied Sciences 12 (3): 201-210, 2012
ISSN 1812-5654 / DOI: 10.3923/jas.2012.201.210
© 2012 Asian Network for Scientific Information

Survey and Analysis of Hardware Cryptographic and Steganographic Systems on FPGA

Sundararaman Rajagopalan, Rengarajan Amirtharajan,
Har Narayan Upadhyay and John Bosco Balaguru Rayappan
Faculty, School of Electrical and Electronics Engineering, SASTRA University,
Tirumalaisamudram, Thanjavur-613401, India

Abstract: Little brooks make great rivers-says a proverb. Information science involves not only the efforts made for gathering, acquiring or collecting the data that corresponds to the information but also contains the ways to save it, protect it and preserve it. The meaning of the proverb however stresses upon how to protect and secure information, as a small leakage will pave way for entire loss of information which should be protected. True, there have been various methods, approaches and algorithms proposed in the past and will emerge in future too in the areas of secure information transmission. Cryptography and steganography have been primary sources for information security. The bird's eye view on the literature pertaining to the above mentioned two giants of information security spots a number of algorithms developed on both software as well as hardware platforms. This study does the survey from the literature on different cryptographic and image steganographic methods implemented on a reconfigurable hardware like FPGAs in the past. The analysis of various methods proposed earlier is also an important objective of this study.

Key words: Cryptography, steganography, information security, FPGA

INTRODUCTION

Marking, the great revolutions in the computer communication networks, lot of researchers further proceed to get even better and effective forms. But the fear of all is one, The Security. Every person while communicating wants security of the best form. Even after the advent of digital communications and many more digitalization techniques, the information security threat is always posing the researchers. Cryptography provides a layer of security in cases where the medium of transmission is susceptible to interception, by translating the message into a form that cannot be read by an unauthorized third party (Kumar and Baskaran, 2010). The main objectives of information security are Confidentiality, Data Integrity, Authentication and Non-Repudiation (Schneier, 2007). Secret information sharing between Alice and Bob is a term synonymous with Con?dentiality and their privacy in an open channel. Con?dentiality is a service offered to the authorised user to keep away the content of private information from eavesdroppers. Data integrity is a service deals with alteration of data by an attacker/hacker Eve. To assure data integrity, the secure system should posses some mechanism to detect data manipulation by attackers. Authentication is a service related to identification of both parties in an open channel. Non-repudiation is a service in crypto system, which prevents both parties from denying the previous actions. When disputes arise due to one of the participants denying later in time that certain actions were taken is not of her will, then it should be taken care off. While cryptography is the act of scrambling the information (Zaidan *et al*., 2010), steganography (Karzenbeisser and Perircolas, 2000; Thenmozhi *et al*., 2012) is an art of hiding information (Bender *et al*., 1996) in fictious covers wherein the cover may be audio, video, image (Amirtharajan and Balaguru, 2009, 2012a, b; Amirtharajan *et al*., 2011; Cheddad *et al*., 2010) or text. Its counter attack called steganalysis deals with the detection of its presence or its length (Qin *et al*., 2009, 2010). Digital watermarking is another information security approach that looks for copyright preserving of an item (Karzenbeisser and Perircolas, 2000). The scope of this article is limited to hardware cryptographic and steganographic systems. Hmood *et al*. (2010) explained about the security and capacity of the steganographic algorithm. Most of the earlier cryptographic and steganographic algorithms are implemented in software platforms. However in literature an equal importance is given to the hardware approach oriented crypto

Corresponding Author: Sundararaman Rajagopalan, Department of Electronics and Communication Engineering,
School of Electrical and Electronics Engineering, SASTRA University, Thanjavur-613401,
Tamil Nadu, India Tel: +914362264101/144 Fax: +914362264120

implementations (Kitsos *et al*., 2004). Several reasons can back the hardware approach to information security. As per the KPMG report commissioned by n Cipher on Key Management Policy and Practices Framework, the superiority of hardware-based security over software based security is emphasized .Normally in software based cryptography, the plain text, keys and related algorithms are stored in the unprotected portion of the computer. This will provide option for an entity to do modification, duplication or substitution. The hardware based security systems called as Hardware Security Modules (HSM) provide a way for shielding the information from harm in the memory of a tamper proof security service.

Earlier work in hardware cryptographic and steganographic systems have been implemented in ASIC as well as FPGA Platforms. The use of reconfigurable hardware platforms allow the user to have multiple stream and block cipher algorithms to be present on the logical elements so that as the requirement changes, the suitable algorithm can come to the fore. Hardware cryptography has a lot of potential in the loop of chaotic dynamic systems. In the chaotic based cryptography, one approach uses hardware-based synchronized chaotic circuits, where the plain text message is hidden in the spectral domain of the chaotic signal (Yang *et al*., 1997). Kitsos *et al*. (2004) have analysed the hardware implementation of 64 block ciphers like Triple DES, IDEA, CAST-128 and KHASAD. The implementation provides high speed execution of algorithm with additional feature like parallelism on the FPGA. A lot of elliptic curve cryptographic architectures on FPGA were reported in the past. NIST standard curves have been used by many implementations (Lutz and Hasan, 2004; Gura *et al*., 2003; Jarvinen *et al*., 2004; Rodriguez-Henriquez *et al*., 2004; De Dormale and Quisquater, 2007). The Elliptic Curve Crypto systems (ECC) (Al-Somani *et al*., 2006) use the finite field arithmetic as the vital architecture. As multiplication, division and modular inverse are a part of many algorithm sensitive crypto systems, the architectures proposed have concentrated on the power, area and design objectives like speed of computation of arithmetic units. FPGA has been the right candidate for implementing the computationally intensive operations for ECC (Al-Somani *et al*., 2009; Chelton and Benaissa, 2008). Similarly few works have been found in the literature pertaining to steganography on FPGAs.

CRYPTOGRAPHY ON FPGA

So many architectures have been proposed in the past in the area of software based cryptographic algorithm design. The main objective of the present task is to review some of the popular cryptographic and steganographic algorithms and their hardware implementations. We discuss DES, AES and HASH cryptographic algorithms and their past FPGA implementations.

FPGAs are the most suitable hardware platforms for implementing the various crypto algorithms by way of different architectures. FPGA based implementation of algorithm needs the code to be present in a Hardware Description Language. Any one of the two HDLs namely VHDL or Verilog HDL has been the design entry sources for carrying the algorithm to the hardware. In Literature most of the FPGA Implementations have been carried out with the help of either Xilinx or Altera FPGAs. As far as the architectures of FPGA are concerned, two main divisions are employed in practise. One is Fine Grained and the other is coarse grained. The Fine grained architecture is suitable for low level data manipulation, especially in bit level manipulation. The coarse grained architecture is used for high level or block level modification. Even though High speed and low area are the two main objectives of any FPGA implementations, there are other expectations from FPGA at the cost of exploiting its fullest resources. Normally FPGAs offer a numerous advantages as follows:

- Reconfigurable hardware architectures to support multiple keys, message sizes and change of algorithms
- Huge Internal RAM in many FPGAs facilitate internal storage of cipher text of heavy payload
- High intrusion protection
- Specific Hardware dependability as the bit stream differs for different FPGAs
- High speed computation of block units of cryptographic algorithms
- Ability to keep multi Processors inside a single chip to create MPSoC architectures
- Experimentation results showcase the much better efficiency of FPGAs over software platforms

Apart from the advantages mentioned above, security of the device is a concern when a whole system is considered. In Viretx-4 devices of Xilinx, the bit stream is encrypted using Xilinx ISE tool. When configuration of the bit stream is done, the virtex device decrypts the bit stream with a key and configures the device according to the developed algorithm.

Block ciphers on FPGA: A number of block ciphers are available in cryptography (Schneier, 2007). Some of them include DES, AES, Triple Des, IDEA, GOST, BLOWFISH etc. This section presents the survey of block ciphers on FPGA platform reported in the literature.

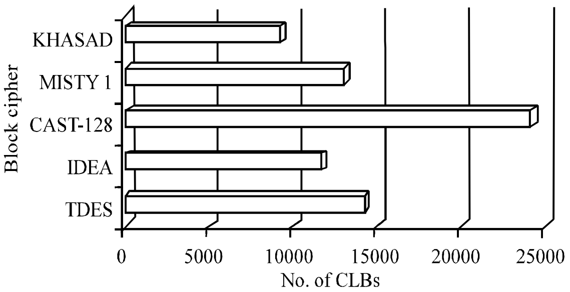


Fig. 1a: Area comparison of various 64 Bit Block ciphers on FPGA

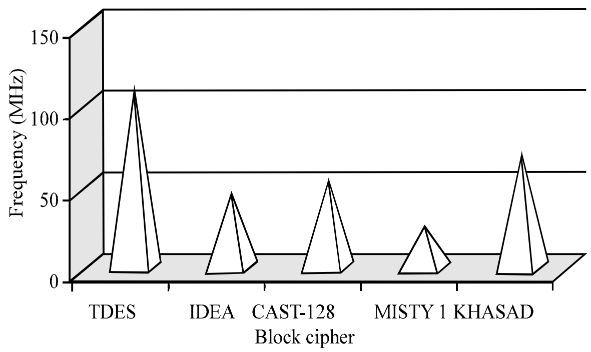


Fig. 1b: Frequency comparison of various 64 Bit Block ciphers on FPGA

Data Encryption Standard (DES) has been a world wide standard called as Data Encryption Algorithm by ANSI and DEA. The DES is a 64 bit block cipher. It uses 56 bit key. As per the DES, 16 rounds of functions have to be performed. Initially 64 bit block will be divided into two halves of 32 bits in each. After that Expansion permutation, S box substitution and P-box permutation have to be performed on the two halves of 32 bit sub blocks. These steps are repeated in all the 16 rounds of operation (Stallings, 2007; Schneier, 2007). FPGA implementation of various block ciphers including Triple DESS has been proposed in Kitsos *et al.* (2004). The block ciphers considered in the article were Triple-DES, IDEA, CAST-128, MISTY1 and KHAZAD. Basic Looping Architecture (BLA) implements only one round of each block cipher and full loop rolling architecture (FULA) implements entire rounds of operation. Virtex 1600EBG560-6 FPGA has been used for implementing all the block ciphers. Figure 1a and b show the LUTs consumption and frequency comparison for FULA of the proposed block ciphers.

Advanced Encryption Standard (AES) has been formulated from Rijndel algorithm. Rijndael is a symmetric block cipher which takes two inputs, namely, the plaintext block to be encrypted and the secret key. It applies an iterative procedure at the end of which an output ciphertext block is produced. During a single iteration, a set of transformations, called a rounds are applied to the state data block. For each round, a round key is generated through a process called key scheduling. The detailed algorithm for AES is given below through a block diagram in Fig. 2.

Table 1: Performance of AES-128 bit block by Li *et al.* (2011)

Design	Encryption only	Decryption only	Encryption and decryption
Slices	626	645	830
4input LUTs	858	872	1452
BRAM	14	14	14
Latency (ns)	264	281.6	321.8
Max clock (MHz)	166.7	156.2	136.7
Throughput (Mbps)	2133.76	2000	1749.76
Efficiency (Mbps/slices)	3.41	3.1	2.1

The AES takes different block sizes namely 128,192 and 256 bits. Also the key size can also be in the size of blocks which are 128, 192 and 256 bits. In many implementations 128 block AES has been employed. In the 128 bit block AES, a 4x4 state Matrix is formed by arranging the 128 bits into 16 bytes. The entire operations are performed over the state matrix. Initially the state matrix is subjected to a transformation using a sub key. Then the transformed state matrix undergoes a loop where four consecutive operations are done. The loop can run for 10, 12 or 14 times. In each loop, first Byte Substitution (BS) is done in which, a non-linear independent substitution operation of state matrix byte with another byte is carried out. The second step is a Shift Row (SR) step, wherein the specific rows are shuffled. In the third step, Mix Column (MC) operation is performed to adjust the columns. The last step in a loop is Add Round Key (ARK) step, where the resultant matrix of the third step is XORed with group key.

Many hardware implementations of AES algorithm on FPGA have been reported in literature. Low cost implementation of AES is proposed. Rais and Qasim (2009) and Drimer *et al.* (2008) proposes high speed AES. Li *et al.* (2011) proposed AES architecture implementation on FPGA. They have implemented the algorithm in Xilinx Virtex FPGA XCV1x25-10. The analysis reported in the work includes encryption, decryption and combined architectures on FPGA. Xilinx ISE 10.1 has been used from the creation of design entry to bit file generation. Table 1 shows the AES on FPGA performance results.

This implementation of AES algorithm on Virtex FPGA yielded 1749.76 Mbps throughput consuming only 14 Block RAMs and 830 slices.

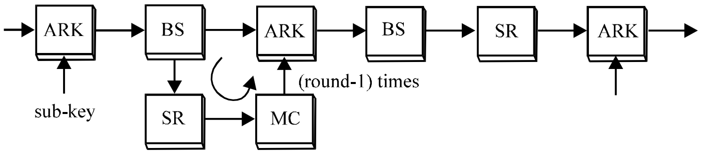


Fig. 2: AES Block Diagram

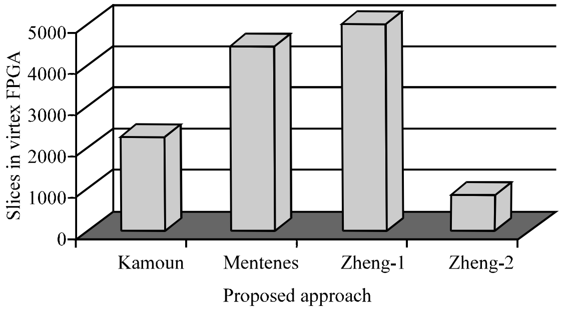


Fig. 3: Hardware consumption analysis of AES implementations on Virtex FPGA proposed

Yuan *et al.* (2011) reported a new type of hardware AES implementation has been reported. In this paper Masked SBox for AES approach is discussed. Different types of masking that are used in practise according to masking functions are Boolean masking, Additive masking, Multiplicative masking and mixed masking. Wolkerstorfer *et al.* (2002) proposed a similar type of AES S box approach. But, Yuan *et al.* (2011), an attempt has been made to create SBox which works with GF(2^4). This method consumes only $2^4 \times 2^4 \times 2^4 \times 4$ bits. Six pre computed tables have been used here. This algorithmhas been implemented in Virtex -5 FPGAs XCVLX30 and XCVLX50. The algorithm takes 20 K gates and 885 slices with Boolean masking technology. Figure 3 shows the FPGA slicescomparison of various masking based AES hardware implementations on Virtex FPGA. Kamoun *et al.* (2008) and Mentens *et al.* (2004) earlier proposed similar kind of architectures on Virtex 4 and XCV800-4 FPGAs, respectively. Zheng-1 in the chart means XCVLX50 and Zheng-2 stands for XCVLX30 FPGA.

Instead of S Boxes which are usual modules in every AES implementations, several other methods have been suggested on using T Boxes employed in FPGAs for AES (McLoone and McCanny, 2003; Bulens *et al.*, 2008). A T Box consumes four times more memory than S Box (ie) 8 kb. An efficient utilization internal Block RAM of Virtex 5 FPGA for fitting the T Box has been proposed by Kundi *et al.* (2010). This approach uses only one BRAM for a 32 bit data path and 4 BRAMs for 128 bit data path. Using the Digital Clock Manager (DCM) and simple multiplexer logic an efficient memory utilization has been suggested. This method was implemented in XC5Vlx110-3ff676 FPGA and it consumed only 4 BRAMS out of 128 (3%) and 199 (1%) slices, thereby supporting a throughput of 32.128 Gbps.

In another implementation work (Borkar *et al.*, 2011), the AES algorithm has been implemented using XCV600BG560-6 FPGA. The maximum operating frequency reported in this work is 140.390 MHz. An encryption/decryption throughput of 352 Mbits/second has been obtained during the AES implementation. This method which uses Ciper FeedBack (CFB) mode for block encryption and decryption, consumes 1853 slices and 391 Bonded IOBs of Virtex FPGA considered for implementation. The hardware consumption is 26% of the total capacity of FPGA. In this implementation the 128 bit input message is encrypted with 128 bit key and after 10 rounds of operations, the cipher text is obtained. Example I/O relation is given below:

- **Plaintext:** 00112233445566778899aabbccddeeff(128 bits)
- **Key:** 000102030405060708090a0b0c0d0e0f (128 bits)
- **Output/cipher text:** 69c4e0d86a7b0430d8cdb78070b4c55a (128 bits)

The hardware implementation of AES algorithm (Papaefstathiou *et al.*, 2004) proposes four different types of AES implementation styles on FPGA platform. As the main approach towards AES hardware implementations follow an architecture that consists of four steps namely key expansion, Initial permutation, Round Permutation and Final permutation, the difference is shown in the objectives considered for implementation and in the basic hardware element used for implementation. Area and performance are the two objectives and Flip-Flop and Latch are the two hardware elements upon which the architecture has been built. Four hardware implementation schemes proposed are:

- Area optimized AES encryption-Flip-Flop approach (AOAESFF)
- Performance optimized AES encryption-Flip-Flop approach (POAESFF)

Table 2: AES Place and Route Results proposed by Papaefstathiou *et al.* (2004)

Hardware design	Latency (nse)	Throughput (Gbps)	Power consumption (mW)
Encryption			
AOAESFF	55.2	2.31	79.24
POAESFF	43.2	2.96	112.48
AOAESLA	56.2	2.28	84.82
POAESLA	49.7	2.57	129.09
Decryption			
AOAESFF	110.4	1.15	163.72
POAESFF	96.0	1.33	193.45
AOAESLA	92.22	1.39	196.47
POAESLA	87.6	1.46	225.49

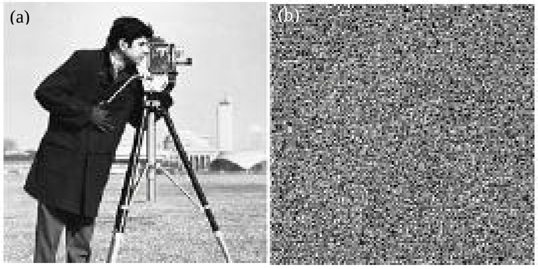


Fig. 4(a-b): (a) Original cameraman image by Chen (2003) and (b) Encrypted cameraman image by Chen (2003)

- Area optimized AES encryption-latch approach (AOAESLA)
- Performance optimized AES encryption-Flip-Flop approach (POAESLA)

Similarly the decryption architectures have also been discussed with Flip-Flop as well as Latch approaches. The analysis has been done in the above schemes and a detailed Latency, throughput and power consumption details have been discussed. Table 2 shows the AES Place and Route results.

Stream ciphers on FPGA: A combination of two stream ciphers LFSR and Hash function to make an hardware efficient stream cipher has been proposed (Deepthi and Sathidevi, 2009). Hash function H(M) that works on an arbitrary length message M, returns a fixed length hash value h. In network applications where data and control packets will be running continuously this type of technique will hold the key for secure communication. The stream cipher was implemented in Altera ACEX-EP1K100QC208-3 FPGA. The LFSR based Toeplitz hash block used in the work has a 5 bit Register to process the computed data. It consumes five D Flip-Flops, five XOR gates and six AND gates.

Swap based hardware cryptographic system: A new type of hardware cryptographic system has been proposed (Chen, 2003). The system performs random permutation and random transformation. The method uses the combination of swap and XOR/XNOR functions under the control of binary sequence from a chaotic system. In this method adjacent signal values are swapped and then they are XORed/XNORed together to generate a system which has low computational complexity and high security.The swap operation utilizes the 1-D logic map suggested (Schuster, 1984; Parker and Chua, 1987). The parameters involved in construction of crypto algorithm of this type are μ and x(n). The μ will take a value which lies between 0 and 1 (ie) $0<x(0)<1$. x(0) is the initial value for this computation. The successive states of the chaotic 1-D map are derived from the relation $x(n+1) = \mu x(n)\ (1 - x(n)$.

For example when $n = 0$, $x(1) = \mu\, x(0)\ (1\text{-}x(0)$, when $n = 1$, $x(2) = \mu\, x(1)\ (1\text{-}x(1))$ and so on.

Like this the remaining x(n) values where n= 0, 1,2….. may be computed. The preceding 24 bits below the binary representation of x(n) , n = 1,2,3,4….are used to find out the sequence b(0),b(1),……etc.

If the signal length is N, in the initial step, two seeds seed1 and seed2 are found out for 0 to (N/16)-1 times. After finding the seeds according to the swap control bit the adjacent values are swapped. The swap operation is given by a function $\text{swap}_{b()}$(g(m), g(n)) where the subscript b(.) is the decider of the swap operation. If the swap control b(.) results in 1, the two elements will be swapped otherwise they preserve the original positions. Once the swapping step is done, the data elements are XORed or XNORed upon the generated output of the b(.) which is a function of k and p [5]. This b(.) results in a two bit value which may be 00,01 ,10 or 11. So four such XOR or XNOR operations have to be performed with a switch case. The result of this case operation g'(m) which is the decrypted message. This work considers the original data as a grayscale image. The encryption process is done on the image. Figure 4a and b show the original and encrypted images.

The hardware implementation consists of three parts namely Signal Encryption Unit (SEU), Chaotic Binary Sequence Generator (CBSG) and Signal Decryption Unit (SDU). The CBSG is responsible for the generation of b(0),b(1)….etc.,. It takes x(0) and μ as inputs and generates the b(.) series. The hardware consumption of the proposed design is as follows:

- No.of 24 x 24 Multipliers - 2
- No.of 24-Bit subtractors-2
- No.of 2:1 MUX/DEMUX-352
- No.of 8:1 MUX/DEMUX- 8

- No.of D Flip-Flops-8
- No.of Two Input XOR gates-32
- No.of Two Input XNOR gates-32
- No.of Frequency Divider-4

The above discussed design has been implemented both in Altera FPGA EPF10K100ARC240-1 and Avanti 0.35 µm cell library. In the FPGA implementation, the crypto algorithm consumes 51,000 gates (51% of logic cell), data sampling rate larger than 80.6 MHz and throughput rate larger than 0.64 Gbps. The Avanti cell implementation takes 7675 gates to implement the crypto algorithm. The data sampling rate is 343.6 MHz and a throughput rate larger than 2.74 Gbps are achieved through this second approach of hardware implementation.

Pseudo random Numbers are used in cryptography for randomizing the message bits in a move to utilize the deterministic sequences with good statistical properties as close as possible to random numbers. The PN sequences is generated by a pattern defined by the Pseudo Random PRNGs. The PRNGs use Linear Feedback Shift Registers (LFSRs) for the PN sequence generation with a primitive polynomial. LFSRs normally use a combination of sequence of D Flip-Flops and xor gate/gates. The LFSRs are normally loaded with a seed value of N bits (a non zero value) where N is the number of bits generated by PRNG during the application of every clock pulse. A new time variant PRNG with sigma Delta Modulator hardwarearchitecture for cryptographic application has been proposed (Gonzalez-Diaz *et al*., 2011). The proposed architecture uses an accumulator which adds a constant seed 'X' with the quantization error 'e0' that is delayed by a clock pulse. This accumulation operation is performed for N stages where in every stage 1 bit from the Auxilary LFSR is utilized for computation purpose. The PRNG architecture has been implemented in a CMOS 0.35 µm process where it consumed 474 digital cells. The FPGA implementation of the same approach needed 57 slices in Virtex 4 FPGA XC4VFX12. Earlier works reported on Mersenne Twist PRNG methods (Tian and Benkrid, 2009; Sriram and Kearney, 2006) consume more hardware elements (slices) in the Virtex FPGA compared to the approach suggested (Gonzalez-Diaz *et al*., 2011). Figure 5 displays a time variant accumulator architecture.

Another Block cipher in this queue is BLOWFISH. Blowfish algorithm uses 64 bit block with a variable length key of up to 448 bits. Like DES, the 64 bit block is splitted into two halves of 32 bits. After that key expansion and 16 rounds of data encryption function have to be carried out (Schneier, 2007). The blowfish cipher is fast, secure, simple and variably secure. Blowfish algorithm proposed by Kumar and Baskaran (2010) used Virtex XCV50-BG256-6 FPGA that consumed 1608 slices and 2984 four input LUTs.

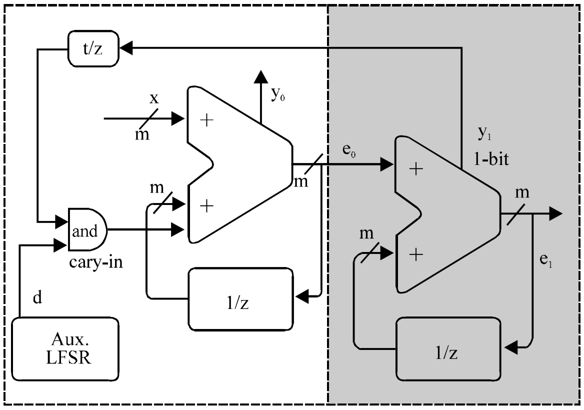


Fig. 5: Time variant coefficient accumulator with M = 2^m

STEGANOGRAPHY ON FPGA

Even though a number of articles are present in literature pertaining to software implementation of various spatial and transform domain steganographic algorithms (Padmaa *et al*., 2011; Amirtharajan and Balaguru, 2009, 2012a, b; Janakiraman *et al*., 2012) and steganalysis (Qin *et al*., 2009, 2010), very few works have been reported on how to implement the information hiding methodology on reconfigurable hardware platforms. Most of the reported works implemented the steganographic module on FPGA in spatial domain. A paper by Farouk and Saeb (2004) reported secret key steganographic micro architecture on Spartan FPGA. The proposed architecture consists of two important parts. One is address generator and the other one is control unit. The address generator unit is used to generate the address of the cover image randomly so that the particular pixel value can be chosen for hiding a bit. The address generator consists of memory of pointers, shuffler and shift and concatenate units. As per the architecture, the memory of pointers issue a 512 bit command to the shuffler unit which in turn generate a 64 bit block address with the help of a 8 bit key.

The address generator then produces 17 bit value with which we can address 128 kbytes. Totally 64 bits generated from shuffler addresses 64 blocks each having 256 locations. Again this can address only 64*256 locations. This is increased by using the 8-bit key as upper bits of the pointer so that 128 kbytes can be addressed. The cover image is stored in external SDRAM which takes more number of clock cycles compared to

SRAM for reading , modifying and writing the contents. Normally it takes 9 clock cycles for addressing cover image and message locations. This is reduced by considering the message bytes consecutively for hiding the information. The control unit consists of status register, control logic gates, state register and a 3:8 decoder. The control unit has been responsible for monitoring and controlling the information hiding process. The proposed architecture on XC2S100tq144-6 FPGA consumes 1195 slices (99%), 2 DLLs (50%), 1 Global clock (25%) and 43 External IOBs (46%).

A steganographic context technique on cyclone II FPGA has been proposed by (Gomez-Hernandez *et al*., 2008). The method uses noisy area and the regions with abrupt gray level changes as the pixel locations to embed the information. Gray scale images stored in FPGA were the cover images and this approach processes the cover image block by block for information hiding. First the image is divided into group of 3×3 blocks. Then the 3×3 block is again splitted into four 2×2 blocks in such a way that the center pixel of a 3×3 block will be present in all the 2×2 blocks. The suitability is checked for information embedding by comparing the pixels of the 2×2 block. If the compatibility check is positive, then a single bit will be embedded in the center pixel of a 3×3 block. The method has been tested in software platform with a computer operating at 1.4 GHz and 512 MB of RAM memory. Cyclone II EP2C35F672C6 FPGA has been the hardware platform. The time taken to perform the above algorithm in software is 8.089 sec with a 512×512 image as cover whereas the same approach in FPGA took only 0.0325 sec which is 252 times faster. A total of 204 logic elements have been consumed to work out the algorithm on FPGA and a throughput of 61.54 Mbps has been achieved.

Another work reported by Sundararaman and Upadhyay (2011) discusses about steganographic system architecture on FPGA with LFSR based information hiding approach. The important part of this work is the design of LFSR for various size images. Various primitive polynomials were used to construct the LFSR circuits using the D Flip-Flops XOR gates combination. The LFSR has been used as address generator in this work to generate addresses pseudo randomly. The cover image has been stored in external Static RAM which has accessing speed of 10 ns which is faster than the SDRAM which was considered by Farouk and Saeb (2004). So in this aspect, the present methodology is faster than the method in Farouk and Saeb (2004). Three clock cycles of 100 MHz, have been used to generate the pixel address, to read the cover image pixel and to modify the content, then to store it back to the stegoimage pixel location. The architecture has UART for communicating with a Graphical User Interface to transmit the message bytes to be embedded in the image. Without knowing the baudrate with which the pixels are transmitted through the UART transmitter, the pixel reception will not be proper when stego image is received for retrieving the information. This approach uses a particular FPGA that has a specific bit file without which the information cannot be retrieved. Figure 6a and b show the MSE and PSNR analysis of hardware stego images. Figure 7 shows the GUI with LabVIEW for decrypting the message from the stego image. The FPGA used for implementing the proposed architecture is Cyclone II EP2C20F484C7.Total Logical

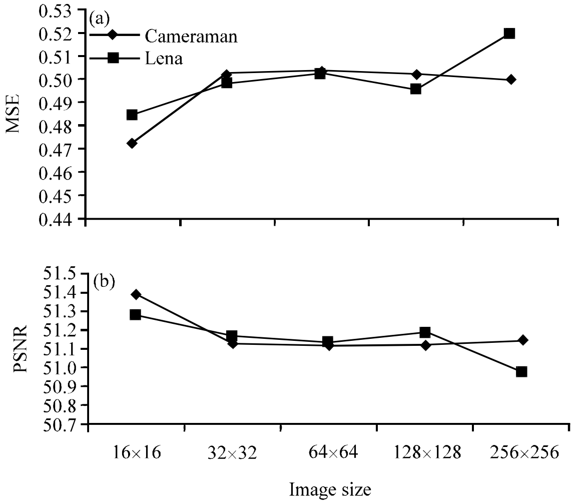
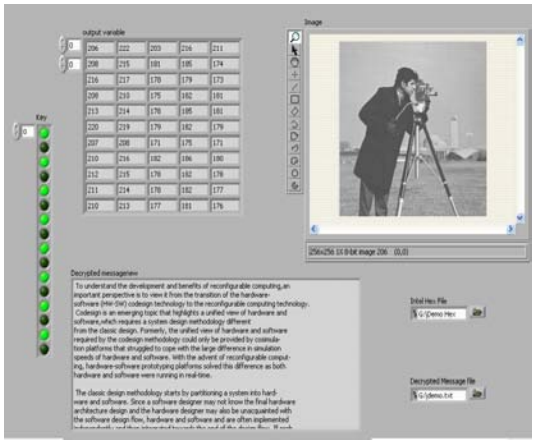


Fig. 6(a-b): (a) MSE and (b) PSNR for Cameraman and Lena Images (k = 1)



206	222	203	216	211
208	215	181	185	174
216	217	178	179	173
208	210	175	182	181
213	214	178	185	181
220	219	179	182	179
207	208	171	175	171
210	216	182	186	180
212	215	178	182	178
211	214	178	182	177
210	213	177	181	176

256x256 1X 8-bit image 206 (0,0)

Decrypted messageview

To understand the development and benefits of reconfigurable computing,an important perspective is to view it from the transition of the hardware-software (HW-SW) codesign technology to the reconfigurable computing technology. Codesign is an emerging topic that highlights a unified view of hardware and software,which requires a system design methodology different from the classic design. Formerly, the unified view of hardware and software required by the codesign methodology could only be provided by cosimulation platforms that struggled to cope with the large difference in simulation speeds of hardware and software. With the advent of reconfigurable computing, hardware-software prototyping platforms solved this difference as both hardware and software were running in real-time.

The classic design methodology starts by partitioning a system into hardware and software. Since a software designer may not know the final hardware architecture design and the hardware designer may also be unacquainted with the software design flow, hardware and software and are often implemented

Intel Hex File: G:\Demo.Hex

Decrypted Message file: G:\demo.txt

Fig. 7: LabVIEW GUI for Data retrieval from stego image (LFSR hiding approach)

Elements present in this FPGA are 18752. The stego architecture using a 16×16 image consumed 954 Logic Elements (LEs), a 32×32 image consumed 3398 Les and a 64×64 image consumed 11493 LEs. The images of these sizes were stored in internal block RAM of FPGA. Likewise for stego system architecture which used 128×128 image 331 LEs were consumed and for a 256×256 image only 340 LEs were consumed as these images were stored in External SRAM.

Recently, Rajagopalan *et al.* (2011) proposed another FPGA based steganographic system that employs a pixel guide block for hiding the message in grayscale images.

ANALYSIS AND RECOMMENDATIONS

In the case of block cipher implementation with FPGA, many works reported used Virtex FPGA compared to the Altera FPGAs. Low cost FPGAs like Spartan and cyclone have been used very little as we found in the literature. In the few works which have concentrated on Stream ciphers on FPGAs, LFSR based algorithms have been analysed (Deepthi and Sathidevi, 2009). Also cellular automata for hardware cryptography are an important direction where more research works can be done, where the software implementation using Matlab is available by Jaberi *et al.* (2012). With this architecture, hardware implementation is also possible. PRNGs generated with LFSRs (Gonzalez-Diaz *et al.*, 2011) have been generated with delta modulators. The effect of combining this architecture for stream ciphers needs to be analysed. Normally memory is a big issue on hardware platforms when the case is image processing, cryptography or steganography as these handle huge payload and images. When it is a Xilinx device, internal BRAMs or when it is an Altera FPGA M4K memory elements are suitable candidates for internal processing of the crypto algorithms or stegano algorithms. The accessing time to these Block RAMs or M4K is very less when compared to accessing the external memory like SRAMs, Flash or SDRAMs. Eventhough Memory management aspects have been discussed in some papers, more work can be done on running the security algorithm on logic blocks of FPGA by storing the cover or payload in internal/external memories thereby providing an option fordetailed memory analysis like power consumption during the memory access, reducing the total time for encryption by clever memory handling etc.

The steganographic algorithms proposed in FPGA platform in the past, have been implemented in spatial domain. All the articles published have considered image as cover for performing data hiding. Some of the works have not focussed on MSE and PSNRerror metrics (Farouk and Saeb, 2004; Gomez-Hernandez *et al.*, 2008). As the robustness of the stego module on hardware is decided by the error metrics like MSE, PSNR and Mean Structure Similarity Index (MSSIM), attention can be focussed on analysing these issues. Moreover image processing functions can be a part of steganographic modules on FPGA thereby making much better information hiding on cover images. Further to this, future work can also focus on audio, video and text as covers on FPGA. As modern day technologies converge towards system on chips, hardware cryptographic and steganographic systems can be a part of MPSoCs.

CONCLUSION

A survey and background discussion has been done on cryptographic and stego architecture implementations on reconfigurable hardware platform in this paper. As most of the present day Network on Chips demand an efficient security methodologies to support secure data packets transmission, these approaches can form a part of NoCs. As more bit wise operations are supported by FPGAs the prominence can be given to these operations while designing hardware crypto and stego architectures. Moreover as transform domain techniques for steganography like DWT, DCT etc., demand floating point operations on reconfigurable hardware which needs special purpose blocks to handle, focus towards this area may provide bright results in the aspect of improving the security.

REFERENCES

Al-Somani, T.F., E.A. Khan, A.M. Qamar-ul-Islam and H. Houssain, 2009. Hardware/software co-design implementations of elliptic curve cryptosystems. Inform. Technol. J., 8: 403-410.

Al-Somani, T.F., M.K. Ibrahim and A. Gutub, 2006. High performance elliptic curve GF (2^m) crypto-processor. Inform. Technol. J., 5: 742-748.

Amirtharajan, R. and R.J.B. Balaguru, 2009. Tri-layer stego for enhanced security-a keyless random approach. Proceedings of the IEEE International Conference on Internet Multimedia Services Architecture and Applications, December 9-11, 2009, Bangalore, India, pp: 1-6.

Amirtharajan, R., R.R. Subrahmanyam, P.J.S. Prabhakar, R. Kavitha and J.B.B. Rayappan, 2011. MSB over hides LSB: A dark communication with integrity. Proceedings of the 2011 IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application (IMSAA), December 12-14, 2011, Bangalore, Karnataka, India.

Amirtharajan, R. and R.J.B. Balaguru, 2011. Data embedding system. WIPO Patent Application WO/2011/114196. http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=WO2011114196&F=0

Amirtharajan, R. and J.B.B. Rayappan, 2012a. An intelligent chaotic embedding approach to enhance stego-image quality. Inform. Sci., (In Press). 10.1016/j.ins.2012.01.010

Amirtharajan, R. and J.B.B. Rayappan, 2012b. Inverted pattern in inverted time domain for icon steganography. Inform. Technol. J., (In Press).

Bender, W., D. Gruhl, N. Morimoto and A. Lu, 1996. Techniques for data hiding. IBM Syst. J., 35: 313-336.

Borkar, A.M., R.V. Kshirsagar and M.V. Vyawahare, 2011. FPGA implementation of AES algorithm. Proceeings of the 2011 3rd International Conference on Electronics Computer Technology (ICECT), April 8-10, 2011, Kanyakumari, pp: 401-405.

Bulens, P., F.X. Standaert, J.J. Quisquater, P. Pellegrin and G. Rouvroy, 2008. Implementation of the AES-128 on Virtex-5 FPGAs. Prog. Cryptol., LNCS, 5023: 16-26.

Cheddad, A., J. Condell, K. Curran and P. Mc Kevitt, 2010. Digital image steganography: Survey and analysis of current methods. Signal Process., 90: 727-752.

Chelton, W.N. and M. Benaissa, 2008. Fast elliptic curve cryptography on FPGA. IEEE Trans. Very Large Scale Integrat Syst., 16: 198-205.

Chen, H., 2003. A new cryptography system and its VLSI realization. J. Syst. Architect., 49: 355-367.

De Dormale, G.M. and J.J. Quisquater, 2007. High-speed hardware implementations of elliptic curve cryptography: A survey. J. Syst. Architect., 53: 72-84.

Deepthi, P.P. and P.S. Sathidevi, 2009. Design, implementation and analysis of hardware efficient stream ciphers using LFSR based hash functions. Comput. Security, 28: 229-241.

Drimer, S., T. Guneysu and C. Paar, 2008. DSPs, BRAMs and a pinch of logic: New recipes for AES on FPGAs. Proceedings of the 16th International Symposium on Field-Programmable Custom Computing Machines, April 14-15, 2008, Palo Alto, CA, pp: 99-108.

Farouk, H. and M. Saeb, 2004. Design and implementation of a secret key steganographic micro-architecture employing FPGA. Proc. Des. Automat. Test Eur. Conf. Exhibit., 3: 212-217.

Gomez-Hernandez, E., C. Feregrino-Uribe and R. Cumplido, 2008. FPGA hardware architecture of the steganographic context technique. Proceedings of the 18th International Conference on Electronics, Communications and Computers, March 3-5, 2008, Puebla, pp: 123-128.

Gonzalez-Diaz, V.R., F. Pareschi, G. Setti and F. Maloberti, 2011. A pseudorandom number generator based on time-variant recursion of accumulators. IEEE Trans. Circuits Syst. II: Express Briefs, 58: 580-584.

Gura, N., S.C. Shantz, H. Eberle, S. Gupta and V. Gupta, 2003. An end-to-end systems approach to elliptic curve cryptography. Cryptogr. Hardware Embedded Syst., LNCS, 2523: 15-22.

Hmood, A.K., H.A. Jalab, Z.M. Kasirun, B.B. Zaidan and A.A. Zaidan, 2010. On the Capacity and security of steganography approaches: An overview. J. Applied Sci., 10: 1825-1833.

Jaberi, A., R. Ayanzadeh and A.S.Z. Mousavi, 2012. Two-layer cellular automata based cryptography. Trends Applied Sci. Res., 785: 68-77.

Janakiraman, S., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2012. Pixel forefinger for gray in color: A layer by 0layer stego. Inform. Technol. J., 11: 9-19.

Jarvinen, K., M. Tommiska and J. Skytta, 2004. A scalable architecture for elliptic curve point multiplication. Proceedings of the 2004 IEEE International Conference on Field-Programmable Technology, December 6-8, 2004, The University of Queensland, Brisbane, pp: 303-306.

Kamoun, N., L. Bossuet and A. Ghazel, 2008. SRAM-FPGA implementation of masked S-Box based DPA countermeasure for AES. Proceedings of the 3rd International Design and Test Workshop, December 20-22, 2008, Monastir, pp: 74-77.

Karzenbeisser, S. and F. Perircolas, 2000. Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, UK., ISBN: 1580530354, pp: 240.

Kitsos, P., N. Sklavos, M.D. Galanis and O. Koufopavlou, 2004. 64-bit Block ciphers: hardware implementations and comparison analysis. Comput. Electr. Eng., 30: 593-604.

Kumar, P.K. and K. Baskaran, 2010. An ASIC implementation of low power and high throughput blowfish crypto algorithm. Microelectr. J., 41: 347-355.

Kundi, D.S., A. Aziz and N. Ikram, 2010. Resource efficient implementation of T-Boxes in AES on Virtex-5 FPGA. Inform. Process. Lett., 110: 373-377.

Li, C., Q. Zhou, Y. Liu and Q. Yao, 2011. Cost-efficient data cryptographic engine based on FPGA. Proceedings of the 2011 4th International Conference on Ubi-Media Computing (U-Media), July 3-4, 2011, Sao Paulo, pp: 48-52.

Lutz, J. and A. Hasan, 2004. High performance FPGA based elliptic curve cryptographic co-processor. Proc. Int. Conf. Inform. Technol.: Cod. Comput., 2: 486-492.

McLoone, M. and J.V. McCanny, 2003. Rijndael FPGA implementations utilising look-up tables. J. VLSI Signal Process., 34: 261-275.

Mentens, N., L. Batina, B. Preneel and I. Verbauwhede, 2004. An fpga implementation of rijndael: Trade-offs for side-channel security. Proceedings of the IFAC Workshop on Programmable Devices and Systems, November 17-19, 2004, Krakow, Poland, pp: 493-498.

Padmaa, M., Y. Venkataramani and R. Amirtharajan, 2011. Stego on 2^n:1 platform for users and embedding. Inform. Technol. J., 10: 1896-1907.

Papaefstathiou, I., V. Papaefstathiou and C. Sotiriou, 2004. Design-space exploration of the most widely used cryptography algorithms. Microprocess. Microsyst., 28: 561-571.

Parker, T.S. and L.O. Chua, 1987. Chaos: A tutorial for engineers. Proc. IEEE, 75: 982-1008.

Qin, J., X. Sun, X. Xiang and Z. Xia, 2009. Steganalysis based on difference statistics for LSB matching steganography. Inform. Technol. J., 8: 1281-1286.

Qin, J., X. Xiang and M.X. Wang, 2010. A review on detection of LSB matching steganography. Inf. Technol. J., 9: 1725-1738.

Rais, M.H. and S.M. Qasim, 2009. Efficient hardware realization of advanced encryption standard algorithm using Virtex-5 FPGA. IJCSNS, 9: 59-63.

Rajagopalan, S., S. Janakiraman, H.N. Upadhyay and K. Thenmozhi, 2011. Hide and seek in silicon: Performance analysis of quad block equisum hardware steganographic systems. Proceedings of the International Conference on Communication, Technology and System Design, December 7-9, 2011, Coimbatore, Tamilnadu, India.

Rodriguez-Henriquez, F., N.A. Saqib and A. Diaz-Perez, 2004. A fast parallel implementation of elliptic curve point multiplication over GF(2^m). Microprocess. Microsyst., 28: 329-339.

Schneier, B., 2007. Applied Cryptography: Protocols, Algorithm and Source Code in C. 2nd Edn., Wiley, India.

Schuster, H.G., 1984. Deterministic Chaos: An Introduction. Physik-Verlag, Weinheim, ISBN-13: 978-3527293155,.

Sriram, V. and D. Kearney, 2006. An area time efficient field programmable mersenne twister uniform random number generator. Proceedings of the 2006 International Conference on Engineering of Reconfigurable Systems and Algorithms, June 26-29, 2006, Las Vegas, USA, pp: 244-246.

Stallings, W., 2007. Cryptography and Network Security: Principles and Practice, Prentices. 5th Edn., Prentice Hall, New Delhi.

Sundararaman, R. and H.N. Upadhyay, 2011. Stego system on chip with LFSR based information hiding approach. Int. J. Comput. Appl., 18: 24-31.

Thenmozhi, K., P. Praveenkumar, R. Amirtharajan, V. Prithiviraj, R. Varadarajan and R.J.B. Balaguru, 2011. OFDM+CDMA+stego = secure communication: A review. Res. J. Inform. Technol., (In Press).

Tian, X. and K. Benkrid, 2009. Mersenne twister random number generation on FPGA, CPU and GPU. Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems, July 29-August 1, 2009, San Francisco, CA, pp: 460-464.

Wolkerstorfer, J., E. Oswald and M. Lamberger, 2002. ASIC implementation of the AES S boxes. Proc. RSA Conf. Topics Cryptogr., LNCS, 2271: 67-78.

Yang, T., C.W. Wu and L.O. Chua, 1997. Cryptography based on chaotic systems. IEEE Trans. Circuits Syst. I: Fundamental Theory Appl., 44: 469-472.

Yuan, Z., Y. Wang, J. Li, R. Li and W. Zhao, 2011. FPGA based optimization for masked AES implementation. Proceedings of the 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), August 7-10, 2011, Seoul, pp: 1-4.

Zaidan, B.B., A.A. Zaidan, A.K. Al-Frajat and H.A. Jalab, 2010. On the differences between hiding information and cryptography techniques: An overview. J. Applied Sci., 10: 1650-1655.