International Journal of Security and Its Applications
Vol. 11, No. 10 (2017) pp.35-44
http://dx.doi.org/10.14257/ijsia.2017.11.10.04

Fast Hardware Implementation of AES-128 Algorithm in Streaming Output Feedback Mode for Real Time Ciphering

Syed Izhar Hussain Zaidi and Samreen Amir Hussain

Dawood University of Engineering and Technology,
Karachi, Pakistan
izhar_hussain@hotmail.com and samreen.amir4@gmail.com

Abstract

Information security is one of the main challenges faced today in purview of the increase in trend of online purchases. Many ciphering schemes are available that provide data encryption. But the strength of stream ciphers available is very squat as compared to block ciphers. To ensure better security, block ciphers are used in various ways for streaming application at the cost of increased computational load. This paper discusses the implementation of AES-128 in output feedback (OFB) mode for real-time streaming applications. The target performance parameter for the implementation of the algorithm is speed as well as reduced memory resources. Implementation techniques for various blocks of the algorithm have been discussed for achieving the target performance. The implementation is functionally tested on Virtex – 6 FPGA. The performance achieved in terms of latency, speed, memory resources and other logic resources is also presented. This shows the effectiveness of the proposed hardware implementation for real-time streaming cipher applications.

Keywords*: AES-128, streaming encryption, OFB mode, FPGA, block Cipher*

1. Introduction

Security is the most challenging aspects in wireless and network related applications. Wireless internet and its networks applications are growing at an enormous rate, Internet Of Things (IOT) as an example use wireless as well as wired data, hence the importance and the value of the exchanged data over the wireless media types is on an increase. The search for the best solution to protect against the data intruders' along with providing these services in time is one of the most dominating research in the cryptographic related communities. Cryptography is types of computer security that converts information from its original form into an encrypted unreadable form. The main characteristics that differentiate one algorithm from another is its ability to secure the data against illegitimate use and efficiency in encryption/decryption at transmitting and receiving end. Advance Encryption Standard (AES) was selected as a method of encrypting electronic data by National Institute of Standard and Technology (NIST) in 2001 [1]. This standard is now used worldwide and remains an active research area for implementation on both software and hardware platforms. Cryptographic algorithms are usually classified into two main categories in which first one is Symmetric Encryption Algorithms, and the second one is an Asymmetric Encryption Algorithms. The difference between these two algorithms is that the symmetric algorithms uses a single key for both encryption and decryption while the asymmetric algorithms use two different keys *i.e*. One for encryption and the other for decryption. It is worth mentioning that for wireless communication encrypting bulks of data is required for which symmetric encryption algorithm is used. It is because the symmetric algorithms are generally faster and can data be re-send if problem in

Received (March 19, 2017), Review Result (August 16, 2017), Accepted (September 8, 2017)

ISSN: 1738-9976 IJSIA

transmission system. With the popularity and requirement of high throughput wireless data communication either through WiFi or 3G/4G GSM mobile network, security of data remains a point of concern for users as well as for operators. RC4 encryption is being used in different wireless environments including WiFi (WEP), which is well known to be breached even when used with TLS protocol. Microsoft has already recommended disabling RC4 through Security Advisory "2868725: Recommendation to disable RC4" [2]. Other methods used in GSM arena for encryption purposes is KASUMI which uses the A5/3 key stream generator which is shown to be vulnerable to related-key attack [3]. Similarly, due to large distances, high delays, sync issues and round-trip time, complex security algorithms are not used for communication purposes in space vehicles and satellite links. Telemetry and control links are especially susceptible to attacks and is in active debate within global space/satellite community [4]. SHA-2 is also known as a one-way algorithm and it is susceptible to infringement and researchers are currently using it in combination with OFB to make it reversible with an added advantage of making it more secure [5]. In this paper, a high speed, low latency and low resource consuming design for AES-128 encryption is presented for streaming applications. The design can be used for ciphering real-time streams from drones to satellites or other telemetry related systems. The design can also be used for securing Ethernet streams, wireless hotspots and in 3G/4G communication in GSM arena. This design requires a very low logic resources and hence can be implemented as an ASIC for secure streaming applications as well as securing space/satellite communications including telemetry. The next section discusses the AES in Block Cipher and feedback mode in detail, section 3 highlights the related work in this field, and section 4 is the implementation of the proposed design in an FPGA and finally in the conclusion we have discussed the results of our work and have given future recommendation.

2. AES Block Cipher Modes

All modes of operation recommended by the NIST require breaking the input into blocks of a given size. That size should be equal to the algorithm input block size. As AES [1] is based on the principle of substitution and permutation known as a substitution-permutation-network. The AES starts with the Key Expansion function with the round being repeated 10, 12 and 14 times according to the key size of 128, 192 and 256 bits respectively giving an expanded key to be used uniquely for each round. These rounds consist of ShiftRows (SR), SubBytes (SB), MixColumns (MC) and AddRoundKey (ARK). Only the last round omits the MixColumn step. Similarly, the decryption is done in reverse mode as shown in the block diagram in Figure 1.

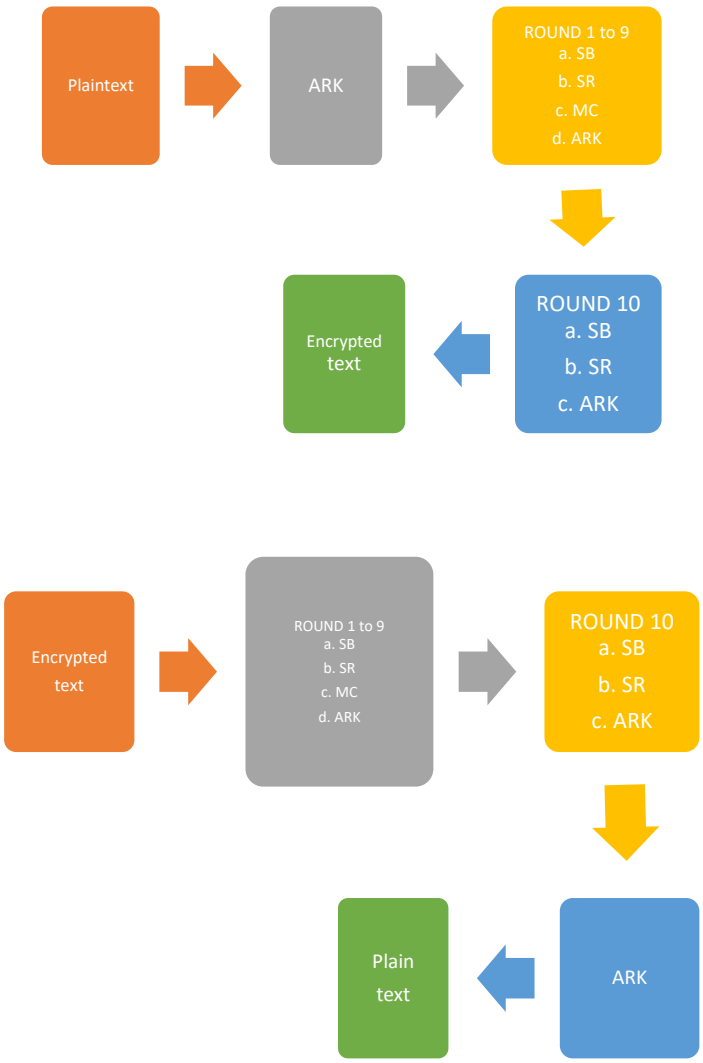


Figure 1. AES Block Diagram

The block cipher as its name implies is applicable on a fixed length group of bits called the block. Its mode of operation depends on repeatedly applying the cipher on the stream which is larger than the block. The different modes of operation for block ciphers are ECB, CBC, PCBC, CFB, OFB and CTR [6]. All block ciphers use an initial variable key known as Initialization Vector (I-V) to randomize encryption and to produce distinct cipher text. For streaming applications, the Output Feedback Mode (OFB), used in this work, transforms the block cipher into synchronous stream cipher. This repeated synchronization patterns are required as they help in re-synchronization of wireless digital receivers. OFB generates key stream blocks, which are then XORed with the stream of data to get the encrypted text. One more property of OFB is that flipping a bit in the encrypted text produces a flipped bit in the plain text at the same location allowing error correcting codes to function flawlessly. Table 1 shows the inherent parallelism of scheme that can be implemented on hardware. The symmetry of the XOR operation assists in encryption and decryption being the same as shown in Figure 2.

Table 1. Table Summarizing Implementation Scheme on Hardware: Courtesy Wikipedia

OFB	
Output Feedback	
Encryption parallelizable:	No
Decryption parallelizable:	No
Random read access:	No

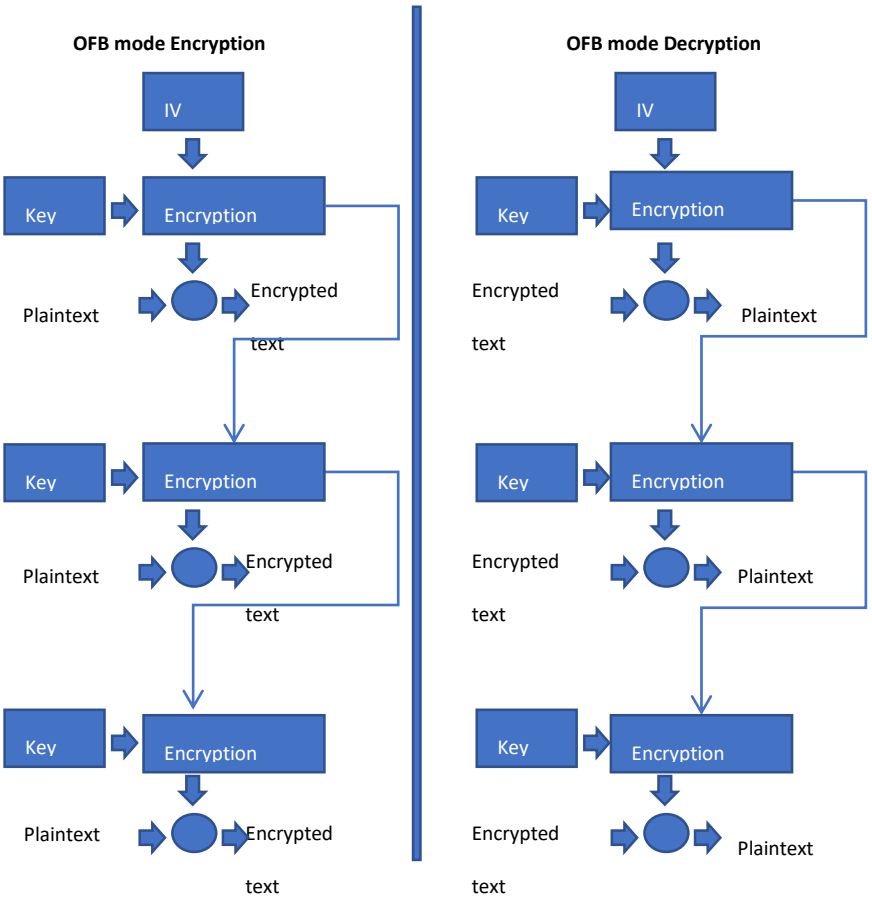


Figure 2. OFB Encryption/Decryption

The advantage of OFB is that it can be implemented on bulk user transmissions and the second advantage is that transmission related error can be corrected by simple error correcting codes. Compared with other DES modes, the error generally accumulates, for example in ECB mode, a single bit error in transmission will cause on the fifty percent of

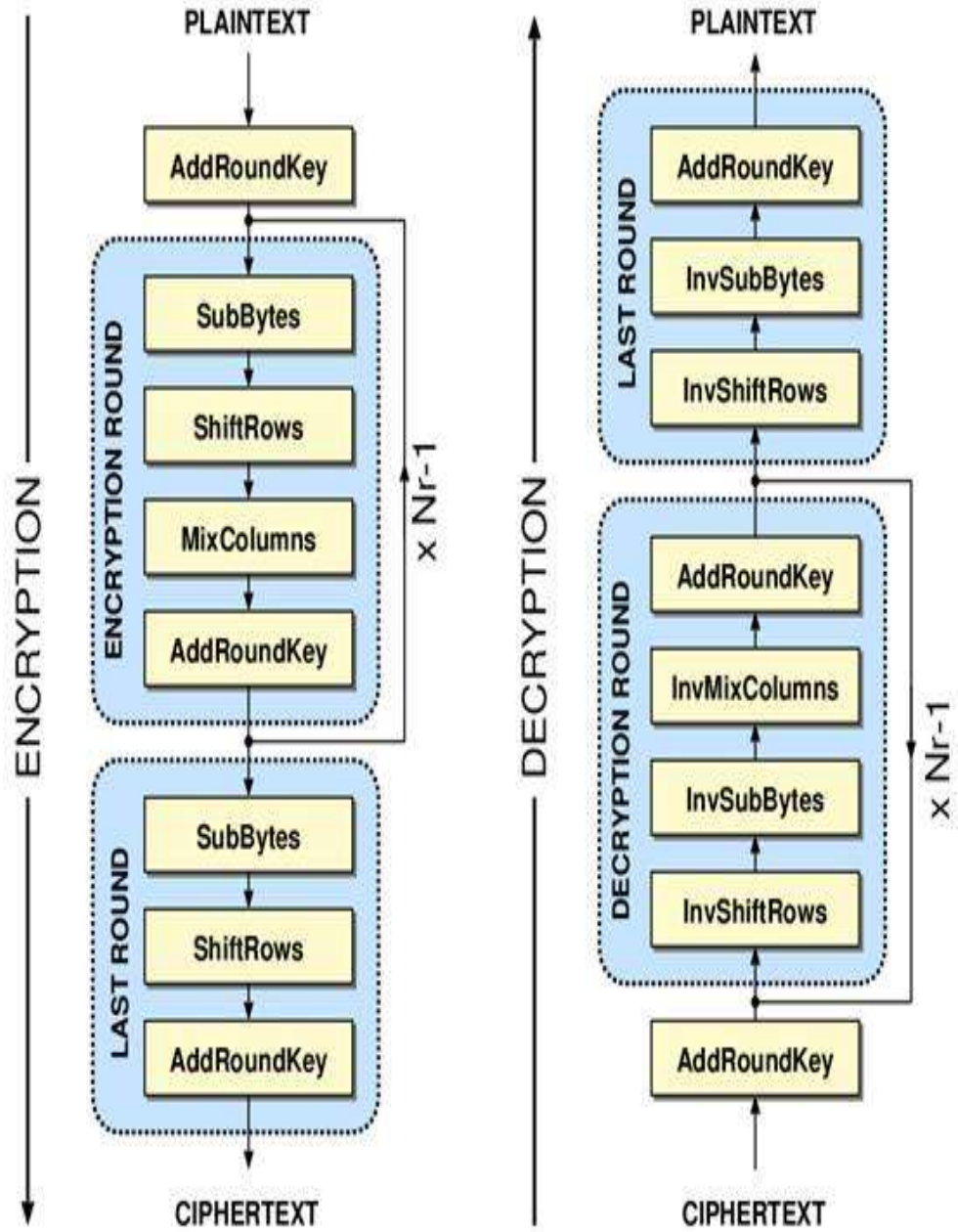
the data of the decrypted block to be in error. The same is in the case of CBC as well making CBC the prime encryption method for digital transmission. The only disadvantage is the requirement of complex methods for ensuring synchronization. However, with the advent of synchronization schemes like HDLC, SDLC or like bi-sync, it can be assured that the errors are minimum.

3. Related Work

Cache based attacks have posed question on the future of the AES [8], the authors Endre Bangerter at al in their research have recommended to use hardware based AES encryption/decryption to block cache based attacks. Hence, the use of hardware based AES in wireless applications such as satellite telemetry and Wi-Fi still has a chance of being a prime focus and the way forward. OFB mode being synchronous is a powerful candidate for all wireless implementations and it is more secure as compared to RC4. From an implementation point of view Tim Good at al [9] has compared AES block cipher and RC4 implementations on an FPGA. In this implementation, the design achieved 2.3 Mbps with a 70 MHz clock with 174 slices and 2 blocks of memory which is 27% smaller as compared to RC4. James S, Grabowski *et al* [10] has implemented all five modes of operation in Xilinx Virtex Pro FPGA with a throughput of 480.427 Mbps for a 128-bit key and the overall design occupied 7452 slices of FPGA. Another implementation without using memory is presented by Chi-Wu Huang *et al* [11] achieving a baud rate of 0.23 Mbps utilizing 2200 gate counts. This 8-bit CFB/OFB implementation was intended for wireless Bluetooth applications. Our implementation specifically targets the overall wireless band with special focus on high speed wireless internet and high throughput telemetry applications.

4. Design and Implementation

As adding and shifting in digital hardware implementations is easier, hence it makes ARK and SR blocks effortlessly implementable. A high speed and low resourced implementation strategy for key-expansion, SB and MC blocks of AES-128 and synchronization for OFB mode targeted for FPGA(s) is discussed below:



Source: http://www.iis.ee.ethz.ch/~kgf/acacia/fig/aes.png

Figure 3. Inside AES

A) Key expansion and S-box

As discussed in section 2, the 16-byte key input to the system is expanded to form eleven so called round keys, each of 16 bytes, to be used for the different rounds. Here it can be seen that the key expansion needs to be performed only once and then the same expanded key is used throughout. So, instead of expanding the key in the hardware, the proposed design makes use of a register file as shown in Figure 4.

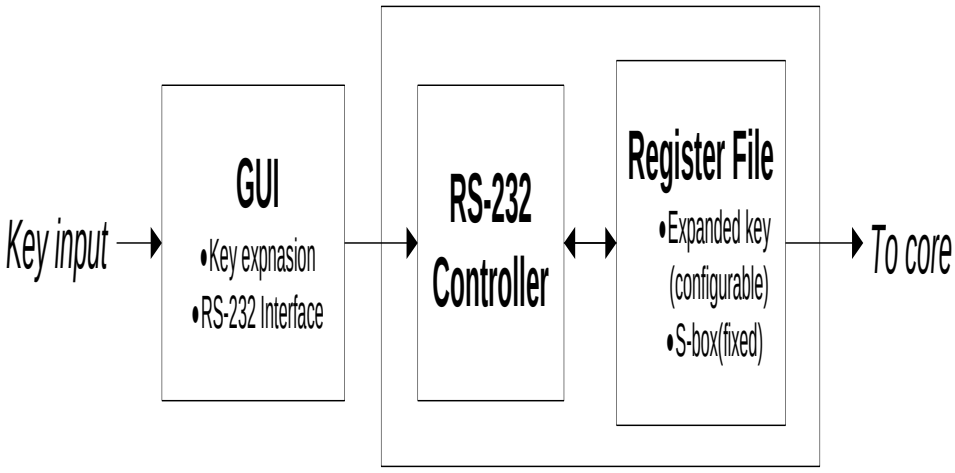


Figure 4. AES Key Configuration

The user needs to enter the 16-byte key to the GUI which expands it and sends it to the hardware using a specified packet format through RS232 interface. The RS232 controller in the device receives the packet containing the expanded key and updates the register file accordingly. The user can confirm the correct configuration of the expanded key in the register file by reading it back using the GUI. The register file also contains the s-box table to be used for the byte substitution step. The byte to be substituted is made available on the input lines of the s-box table module and the substituted value is available on the output lines readily. Hence a combinatorial substitution of bytes can be carried out. The register file is implemented using the register array instead of any memory block. In this way, the logic resources, to be used for key expansion, and memory resources are preserved.

B) Mix Column

Mix column is the most complex operation to be discussed as well as to be performed. However, the simplest way to describe this operation as stated in the standard [1] is that each column is multiplied by a given matrix to get the new valued for that column as shown in Figure 5. Instead of usual multiplication and addition, Galois Field operations are used.

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < \boldsymbol{Nb}.$$

Figure 5. MixColumn Operation (Nb = 4 for AES-128)

Here it can be seen that the multiplications are with three numbers only – 01, 02 and 03. For the ease of computation, the resulting expressions of above matrix multiplication can be simplified to involve multiplication by just one number – 02. Simplification for one element is shown below:

$$S'_{0,c}- = (\{02\} \cdot S_{0,c}) + (\{03\} \cdot S_{1,c}) + S_{2,c} + S_{3,c}$$
$$S'_{0,c}- = (\{02\} \cdot S_{0,c}) + (\{02 + 01\} \cdot S_{1,c}) + S_{2,c} + S_{3,c}$$
$$S'_{0,c}- = (\{02\} \cdot S_{0,c}) + (\{02\} \cdot S_{1,c}) + (\{01\} \cdot S_{1,c}) + S_{2,c} + S_{3,c}$$
$$S'_{0,c}- = (\{02\} \cdot S_{0,c}) + (\{02\} \cdot S_{1,c}) + S_{1,c} + S_{2,c} + S_{3,c}$$

Similarly, all the expressions can be reduced to just two operations – Galois field addition and Galois field multiplication by 2. GF addition corresponds to a XOR operation and a simplified way of performing the GF multiplication by 2 is presented in [1]. GF multiplication by {02} can be implemented at the byte level as a left shift and a subsequent conditional bitwise XOR with {1b} – if MSB is high perform the XOR operation otherwise not [1].

C) Synchronization

For using the block cipher in streaming application using the Output Feedback (OFB) mode synchronization is of vital importance. Since the decryption of a block of data depends on the previous block of data so it is necessary to know the block boundaries otherwise all the data will be lost. In order to achieve synchronization of the block boundaries in the proposed design a 32-bit sync pattern (FE6D2840) is used. IRIG-106 standard [12] states that the maximum length of a major frame should not exceed 1024 bytes. So, as per the standard, the sync pattern is inserted after every 63 ciphered blocks giving a major frame length of 1012 bytes. The insertion of the extra frame sync pattern increases the data rate as compared to the input. However, the increase in data rate is just about 0.4%. Such a small amount of change in data rate falls within the jitter tolerance of the transmitting/receiving systems and hence can be mitigated by increasing the throughput by 0.4% without any change in the configuration of the transmitter/receiver systems. It should be noted that losing a sync pattern will result in a loss of 63 blocks of data.

5. Functional Validation

The implemented system is tested using two ML605 Virtex-6 kits. One of the kits serves as encryptor and the other one serves as decryptor. UART interfaces are used as data transmission protocol. The test setup is shown in Figure 6.

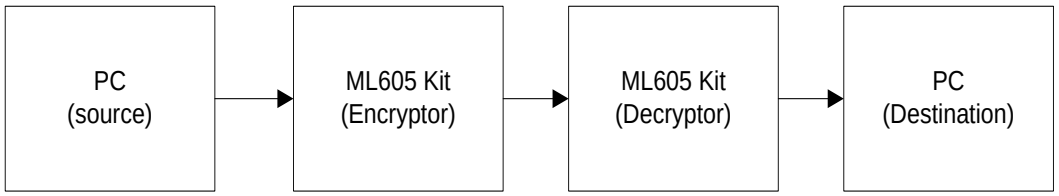


Figure 6. Verification Test Setup

The source PC sends a stream of pseudo random sequence through the serial port. Encryptor receives the stream and transmits it via the UART interface to the decryptor. Decryptor decrypts the stream and sends the deciphered stream to the destination PC. The data received on the destination PC is compared against the known pseudo random sequence and mismatches are reported. The implemented core is found to be correctly functional.

6. Performance

The performance parameters depend on the application. There is always a trade-off between the resources consumed and the speed. The proposed design offers performance that is acceptable in all scenarios. Table 2 shows the quantified values of the performance parameters achieved by implementation of the design for Virtex-6 FPGA. It is clearly indicated that the proposed design provides high speed and low latency while consuming very low logic resources and zero memory resources.

Table 2. Performance Parameters

Max Operating Clock	467MHz
Latency	21 operational cycles
Min Latency (no. of operating cycles * max operating clock)	45ns
Max Data Rate (block size *max operating clock/Number of clock cycles)	2.846 Gbps
Logic Resources	288 (out of 37680 *i.e.* < 1%)
Memory Resources	0

Although Virtex-6 FPGA is not the optimum platform to implement the OFB mode, it has been used to prototype and show the effectiveness of this work. The above table shows that no memory is involved during the process making it a number one contender to implement it in a high-performance ASIC tightly coupled with a processor. The emphasis on hardware based implementation is for low latency and to avoid CACHE based attacks as discussed in the earlier sections. The throughput can be further enhanced by utilizing optimum performance of DSP blocks *i.e.* increasing multiply-accumulate in a single clock cycle.

7. Conclusion and Future Work

A high speed, low latency and low resource consuming design for AES-128 encryption is presented for streaming applications. Compared to software implementation and implementation tools like Lab View/Matlab, the overheads of programming can be reduced by low-level hardware implementations increasing speed and reducing latency. This hardware implementation can be used for ciphering real-time streams from satellites, wireless internet modems, 3G/4G mobile GSM networks and other telemetry systems. The XOR operation symmetry helps in utilizing exactly the same operation in deciphering the data. The design can also be used for securing Ethernet streams on TLS and SSL protocols. The design requires very low logic resources and hence can be implemented for small sized FPGAs and ASIC's. The future work will be to optimize it further for GSM applications especially for 3G/4G mobile internet access and assessing new set of modes (such as GCM, CCM and EAX). The throughput of our system can also be increased by properly utilizing Digital Signal Processor functionality of fast Multiply-accumulate. The synchronization issue can be resolved via connection oriented system rather than implementing on connectionless systems. Another potential application is use of different internet bands allocated to user in GSM arena using Software Defined Radio (SDR) technology. The concept of Internet of Things (IOT) has also led to the security of user

premises and data of personal nature to be secured by implementing end-to-end encryption. This solution might be the answer of many high speed, high throughput applications currently circulating in marketplace.

References

[1] "Announcing the Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197", United States National Institute of Standards and Technology (NIST), **(2001)**.
[2] https://blogs.technet.microsoft.com/srd/2013/11/12/security-advisory-2868725-recommendation-to-disable-rc4/.
[3] O. Dunkelman, N. Keller and A. Shamir, "A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony", Cryptology ePrint Archive, Report 2010/013, **(2013)**.
[4] S.M.J. Shah, A. Nasir and H. Ahmed, "A Survey Paper on Security issues in Satellite Communication Network Infrastructure", International Journal of Engineering Research and General Science, vol. 2, Isuue 6, **(2014)**.
[5] R.K. Ibrahim, R. A J. Kadhim and A.S.H. Alkhalid, "Incorporating SHA-2 256 with OFB to realize a Novel Encryption Method", Computer Networks and Information Security (WSCNIS), 2015 World Symposium, DOI: 10.1109/WSCNIS.2015.7368295, IEEE Conference Publications, **(2015)**, pp. 1-6.
[6] www.wikipedia.org/wiki/Block_cipher_mode_of_operation#Output_feedback_.28OFB.29 Accessed on 15 Dec 2016.
[7] M. McLoone and J.V. McCanny, "Generic architecture and Semiconductor intellectual property cores for advanced encryption standard crypotography, Generic architecture and semiconductor intellectual property cores for advanced encryption standard cryptography", Computers and Digital Techniques, IEE Proceedings, ISSN: 1350-2387, INSPEC Accession Number: 7717169, DOI: 10.1049/ip-cdt:20030499, Sponsored: IET, Publisher: IET, vol. 150, Issue 4, **(2003)**, pp. 239–244.
[8] E. Bangerter, D. Gullasch and S. Krenn, "Cache Games-Bringing Access-Based Cache Attacks on AES to Practice", Proceeding SP '11 Proceedings of the 2011 IEEE Symposium on Security and Privacy, IEEE Computer Society Washington, DC, USA © 2011 ISBN: 978-0-7695-4402-1 DOI: 10.1109/SP.2011.22, **(2011)**, pp. 490-505.
[9] T. Good and M. Benaissa, "AES as Stream Cipher on a small FPGA", in System Science and Engineering (ICSSE), 2013 International Conference on, ISSN : 2325-0909, Print ISBN: 978-1-4799-0007-7, DOI: 10.1109/ICSSE.2013.6614638, IEEE, **(2013)**, pp. 87–92.
[10] J. S. Grabowski and A. Youseef, "An FPGA implementation of AES with support for counter and feedback modes", Published in: Microelectronics, 2007. ICM 2007. Internatonal Conference on, Print ISBN: 978-1-4244-1846-6, INSPEC Accession Number: 9965418, DOI: 10.1109/ICM.2007.4497657, Publisher: IEEE, **(2007)**, pp. 39–42.
[11] C.W. Huang, S.W. Kuo and C.J. Chang, "Embedded 8-bit AES in wireless Bluetooth application", Published in: System Science and Engineering (ICSSE), 2013 International Conference on, ISSN : 2325-0909, Print ISBN: 978-1-4799-0007-7, DOI: 10.1109/ICSSE.2013.6614638, Publisher: IEEE, **(2013)**, pp. 87–92.
[12] Telemetry Standards, IRIG Standard 106-13 (Part 1), Chapter 4, **(2013)**.

