International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, Jul - Aug. 2013 pp-2442-2445 ISSN: 2249-6645

Implementation of Cryptography Architecture with High Secure Core

R. Vijaya Durga[1], K. V. Ramana Rao[2], K. Prasanna Kumar[3]
[M.tech] Department of Electronics and Communication Engineering, Pydah College of engineering and Technology, Affliated to JNTUK, Ghambheeram, Visakhapatnam-531163

ABSTRACT: *The continuous increase in demand for security in electronic systems and communication systems which lacks a secure architecture has resulted in the need to provide cryptography architecture with high secure core. The hardware implementation of the cryptography core which incorporates multiple algorithm for security purpose was already developed but if the architecture is capable of switching between the algorithms used for encryption /decryption as controlled by the host computer dynamically, then the security over the data path will be increased by making the attempt for hacking too difficult. The switching between heterogeneous algorithms will also increase the confusion level. This proposed architecture implements three symmetric algorithms namely the standard AES, standard DES and proposed modified DES (MDES) algorithms. Representing these algorithms in the functional block level and also using the new concept of common S-Box , results in operations that are common to all the three algorithms, allows us to merge them in a single architecture and thus there is an area reduction of 14.5% in cryptography core with 2 S-Boxes rather than using 11 S-Box. The operation of this cryptography core is controlled by the control signals, selecting which algorithm to work at time, making it difficult to hack the information transferred through the data line.*

I. INTRODUCTION

The cryptography technique users always have demanded high security for the data they transfer, and this demand increases day-by-day as the information transferred is hacked unlawfully. To meet this needs, though some works with different architectures are designed, this project introduces a simple new technique having three algorithms onto a single core, making the architecture to function as any one of the algorithm at different time instance helps in increasing the security level. In the proposed cryptography architecture we have implemented the private key algorithm and the algorithms used here are standard AES which is a 128-b block cipher algorithm, standard DES and MDES both of which are 64-b block cipher algorithms.

A cryptography system can be implemented either in software or hardware. Software implementation allows multiple algorithms to be supported on the same platform but they are usually slow and are considered to be more vulnerable to side channel attack, which uses physical measurements on the devices, for example the power consumption of the processor to detect the encryption / decryption key while on the other hand the hardware implementation is comparatively faster and more secure as compared to that of software. The main concept used in order to increase the security is that, the inputs to core module is a 128-b data and 128-b key and the output is also 128-b which generally confuses the hacker since AES generates an output which is 128-b data but the other two algorithm generates output of 64-b data which is hidden in the 128-b data that is coming out of the module and also the control signal which selects the algorithms operation increases the security level because before the hacker hacks the information the encrypted/decrypted data will be transmitted since he doesn't know which algorithm is used at what time as it is been controlled by the host medium with the control signal being sent via the secure data channel with the help of private key algorithm.

II. STRATEGY

Some of the strategy used in designing algorithm is as follows. In designing the Key Schedule module for AES encryption, we had used on the fly key method i.e. the keys are generated in accordance with the working of each round in AES flow. Instead of pre-generating the keys required for the encryption process which consumes lot of time and reduces the frequency and speed of operation, but by using this strategy keys are generated whenever the round operation works helping in increasing the throughput.

The technique used to design common S-Box is show in the Fig.1 we know that the number of S-Boxes used for DES algorithm is 8 and each box contain 64 elements of 4-b each. So the number of elements as a whole is 64 X 8 = 512 elements. And the common S-Box that we are going to use is (16 x 16 = 256) capable of holding 256 elements of 8-b each this is nothing but the S-Box that we use for AES and DES algorithms. Now, what we do is we club two elements starting from starting element till end and represent them in the hexadecimal value and dump it into the common S-Box. As already told the total elements in DES is 512 and each is of 4-b. Now, we just club two elements from DES S-Box i.e. of 4-b each and represent it as 8-b elements that fits into the Common S-Box. As show in the example the element 14 and 4 of 4-b each is clubbed and represented as E4 and the value is dumped into the first location of the Common S-Box and as such this is done for the rest of the elements and thus all the elements of DES S-Box is dumped into Common S-Box and hence the required functionality is achieved. But there is a small change because of this common box in the cipher function of the DES algorithm i.e. instead of using 6 i/p □ 4o/p LUT , we require an LUT of 8 i/p □ 8 o/p. And by using this technique we now require 2 S-Box as a whole instead of 11 S-Box, one for AES- inv-S-Box which is mandatory and other S-Box that is common to all the three algorithms and because of this strategy there is area reduction in the whole of the architecture.

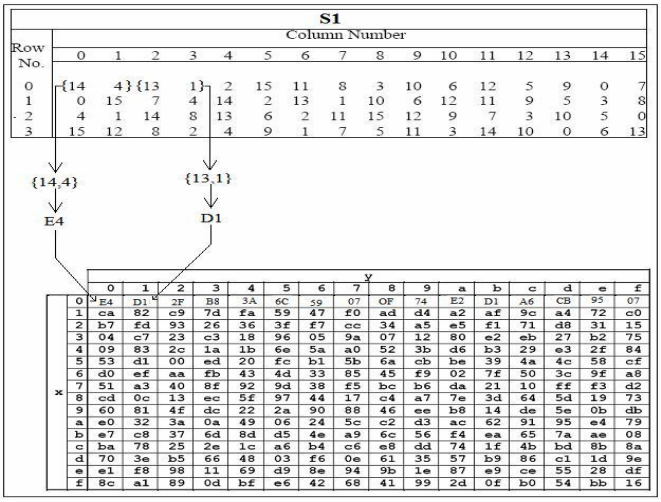

Fig.1. Common S-Box Technique

III. CRYPTOGRAPHY CORE ARCHITECHTURE

The complete cryptography architecture is designed using Verilog HDL language and is synthesized using Xilinx and cadence RTL compiler. The cryptography core functions as standard AES or standard DES or MDES according to the selection of the control signals. The block diagram of the complete cryptography architecture is shown in the Fig.2.

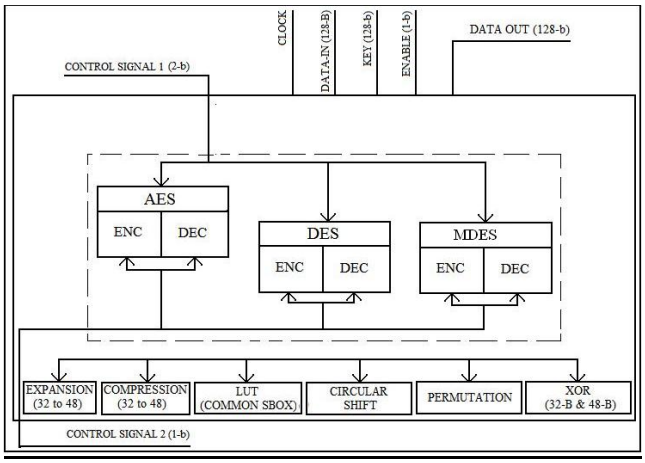

Fig.2. Cryptography architecture

IV. CRYPTOGRPHY CORE OPERATION

The working of the core is started with the help of enable signal. As soon as the architecture is given enable signal, the structure start working as any of these algorithms on the basis of the control signal provided the algorithms works in flow with the clock signal.

ENABLE	CONTROL1	CONTOL2	OUTPUT
0	??	?	HIGH IMPEDANCE
1	00	1	AES ENCRYPTION
1	00	0	AES DECRYPTION
1	01	1	DES ENCRYPTION
1	01	0	DES DECRYPTION
1	10	1	MDES ENCRYPTION
1	10	0	MDES DECRYPTION

Table.1. Working Operation of Cryptography Core

The Table.1 reveals the combinations of the control signal and is corresponding algorithm selection for functioning of the architecture.

The necessary operations required for direct implementation of the cryptography algorithm is shown in Below Table.2.

ALGORITHM	Expand / Permutation	LUT Logical	XOR	Circular/Logical Right/Left shift	LUT (S-box)
AES	-	8in=> 8out		32 (32 by 8b)	2
DES	32,48,56,64	6in=> 4out	32, 48	28(by 1b or 2b)	8
MDES	32,48,56,64	8in=> 8out	32,48 ,64	28(by 1b or 2b)	1

Table.2.Basic Operation Required For Implementation of the algorithm

This gives us an idea on what are the operations that governs the working condition of the algorithm.

ALGORITHM	Expand / Permutation	LUT Logical	XOR	Circular/Logical Right/Left shift	LUT (S-box)
AES	-	8in=> 8out		32 (32 by 8b)	2
DES	32,48,56,64	6in=> 4out	32, 48	28(by 1b or 2b)	8
MDES	32,48,56,64	8in=> 8out	32,48 ,64	28(by 1b or 2b)	1

Table.3. Basic Operation Required For Implementation after Reordering

The procedure for designing this cryptography core is to express the high level operations of each algorithm in terms of basic arithmetic and logical operations to maximize the Number of common operations among the algorithms and also using the common S-Box technique results in minimize the overall required architecture area. Hence some of the operations that are common to the algorithms are expansion (32 to 48), compression (48 to 32), LUT, permutation, xor operation, which gets plugged into architecture of the algorithm whenever it is called for functioning. Table.3 shows the operations required for each algorithm after reorganization of the architecture.

V. SECURITY ENHANCEMENT IN CRYPTOGRAPHY CORE

In the working operation of the cryptography core a simple technique is used in order to increase the security level to the data being transferred. Fig.3 shows technique show the technique used.

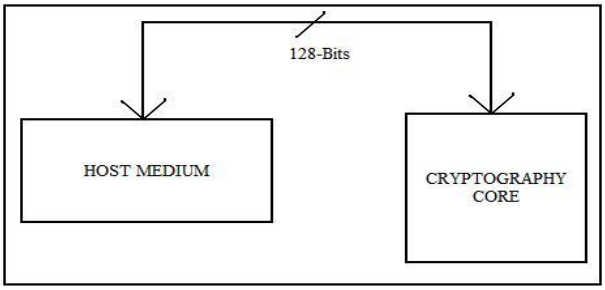

Fig.3. Security Enhancement Technique

Here the data input and the key to core is of 128-b and the data out is also128-b. In general when the data's are encrypted using AES algorithm, the output transferred is 128-b, but when the other two algorithms are used for encryption process the data out should be 64-b as these two algorithms are 64-b block cipher algorithms as explained before, but the data coming out of this core is of 128-b, i.e. the 64-b encrypted/decrypted data is embedded into the 128-b data coming out.

This helps in increasing the security level because in general if it is going to be 64-b data, the hacker has to try all 2^{64} combinations to data out to crack and find the information which takes a longer time, but here as the important data is embedded into 128-b data coming out, it's a tough job to hack the information and apart from this, the selection of which algorithm to work at what time by the host medium too increase security level.

VI. EXPERIMENTAL RESULTS

The Table.4 shows the result of the cell usage and other details when the design is targeted to ASIC and synthesized using the Xilinx's RTL compiler. Here the cell usage of cryptography core using 2 S-Box is less when compared to that of using 11 S-Box.

ALGORITHM	No of Cells Used	Power(w)	Time Period(ns)	Frequency of Operation (MHz)
CRYPCORE(11 S-BOX)	288153	0.156	24.597	40.683
CRYPCORE(2 S-BOX)	245838	0.082	24.733	40.431

Table.4. Area and Timing Report for Design Using Xilinx's RTL Compiler Targeting 180 nm

VII. CONCLUSION

A simple technique to address the important issue of performance improvement of the cryptography core is discussed in this work and also explains the design of cryptography core for electronic devices and other desired applications used for secure data transmission using private Key algorithm. Through the simulation and synthesize results, we have inferred that performance level of the cryptography core is increased and we have achieved this by expressing the operations of the three algorithms namely the standard AES, standard DES and MDES in terms Of basic arithmetic and logical operations that maximizes the number of common blocks among them and also using common S-Box technique, resulting in merging of these three algorithms in a single core. And, this design also increasing the security level by controlling the core to Operate in different algorithms at a particular time with the help of control signal which is transferred via secure data line using private key algorithm, making it too difficult for hacking and other external attacks. A few extra works that can be added to this design in future to enhance its capabilities are implementing the private key cryptography algorithm onto this core, which helps in secure transfer of keys and the control signals used for the operation of the core.

REFERENCES

[1] Data Encryption Standards (DES), Oct.1999.FIPS standards publications.
[2] Advanced Encryption Standard (AED), Nov.2001.FIPS standards publication 197.
[3] AES and DES Algorithm, "Wikipedia-Free Encyclopaedia".
[4] Atul Kahate, "Cryptography and Network Security", 2003.
[5] William stalling, ''Cryptography and Network Security Principles and Practice''.
[6] John Daemen and Vincent Rijmen, "AES proposal: Rijndael".
[7] J.Bhaskar, "Verilog HDL primer and Verilog HDL synthesize".