Optimized Lightweight Bio-Cryptosystem for Wsn Applications

P. Prakathi , S. Nithiyalakshmi, D. Renuka, **:**Mr. A. S. Loganathan,

UG Student, Final year ECE Department, GRT Institute of Engineering and Technology, Tiruttani, Tiruvallur District.

UG Student, Final year ECE Department, GRT Institute of Engineering and Technology,Tiruttani, Tiruvallur District.

UG Student, Final year ECE Department, GRT Institute of Engineering and Technology, Tiruttani, Tiruvallu District.

Assistant Professor, Department of Electronics and Communication, GRT Institute of Engineering and Technology, Tiruttani, TiruvallurDistrict.

Abstract:*In recent days IOT devices and WSN sensor nodes are becomes a popular target for implementing cryptographic block ciphers, as a well-designed security solution that can combine some of the algorithmic flexibility and cost efficiency of an equivalent software implementation with throughputs that are comparable to any custom designs. The recently selected Advanced Encryption Standard (AES) is slowly replacing other tiny ciphers as the building block of choice for secure systems and is well suited to an any power compatible system implementation. Here we have described some key generation and multi round physical transformation models for encrypting the information and possible biometric schemes(RETINA) that can be used for authentication along with cryptography on networked embedded computers. Public-key infrastructures are secured, but only to the extent that private keys of individuals are maintained secret. Usually this involves securing the private key(s) using a password, a PIN or a token. Biometrics alone does not provide a great deal of safety, but a combination of biometrics will provide a higher degree of security for embedded computing devices. Finally we improve the performance of the proposed system in terms of transformation level and its efficiency will be proved through hardware synthesis.*

I. Introduction

Objectives

In today's digital world, encryption is emerging as a disintegrable part of all communication networks and information processing systems, for protecting both stored and in transit data. Encryption is the transformation of plain data (known as plaintext) into unintelligible data (known as cipher text) through an algorithm referred to as cipher. There are numerous encryption algorithms that are now commonly used in computation, but the U.S. government has adopted the Advanced Encryption Standard (AES) to be used by Federal departments and agencies for protecting sensitive information. The National Institute of Standards and Technology (NIST) have published the specifications of this encryption standard in the Federal Information Processing Standards (FIPS) Publication 1997.

Any conventional symmetric cipher, such as AES requires a single key for both encryption and decryption, which is independent of the plaintext and the cipher itself. It should be impractical to retrieve the plaintext solely based on the cipher text and the encryption algorithm, without knowing the encryption key. Thus, the secrecy of the encryption key is of high importance in symmetric ciphers such as AES. Software implementation of encryption algorithms does not provide ultimate secrecy of the key since the operating system, on which the encryption software runs, is always vulnerable to attacks.

. Key Based Approach

Different versions of AES algorithm exist today (AES128, AES196 and AES256) depending on the size of the encryption key. In this project, a hardware model for implementing the AES 128 algorithm was developed using the Verilog hardware description language. A unique feature of the design proposed in this project is that the round keys, which are consumed during different iterations of encryption, are generated in parallel with the encryption process.

Cryptographic Systems Are Generally Classified On The Following Basis:

(1). Type Of Operations Used To For Transforming Plaintext To Cipher TEXT: Most encryption algorithms are based on 2 general principles,

a) **Substitution**, in which each element in plain text is mapped to some other element to form the cipher text.

b) **Transposition,** in which elements in plaintext are rearranged to form cipher text.

(2). NUMBER OF KEYS USED: If both the sender and the receiver use a same key then such a system is referred to as Symmetric, single-key, secret-key or conventional encryption. If the sender and receiver use different keys, then such a system is called Asymmetric, Two-key or public-key encryption.

(3). PROCESSING OF PLAIN TEXT: A Block cipher processes the input one block at a time, producing an output block for each input block. A stream cipher processes the input elements continuously producing output elements on the fly.

Most of the cryptographic algorithms are either symmetric or asymmetric key algorithms.

Cipher Transformations:

The AES cipher either operates on individual bytes of the State or an entire row/column. At the start of the cipher, the input is copied into the State and then, an initial Round Key addition is performed on the State. Round keys are derived from the cipher key using the Key Expansion routine. The key expansion routine generates a series of round keys for each round of transformations that are performed on the State. It consists of the following four steps:

(a) SubBytes
(b)ShiftRows
(C) MixColumns
(d) AddRoundKey

. Inputs, Outputs And The State:

The plaintext input and ciphertext output for the AES algorithms are blocks of 128 bits. The cipher key input is a sequence of 128, 192 or 256 bits. In other words the length of the cipher key, N_k is either 4, 6 or 8 words which represent the number of columns in the cipher key. The AES algorithm is categorized into three versions based on the cipher key length. The number of rounds of encryption for each AES version depends on the cipher key size.

In the AES algorithm, the number of rounds is represented by N_r, where $N_r = 10$ when $N_k = 4$, $N_r = 12$ when $N_k = 6$, and $N_r = 14$ when $N_k = 8$. The following table illustrated the variations of the AES algorithm. For the AES algorithm the block size (N_b), which represents the number of columns comprising the State is N_b= 4.

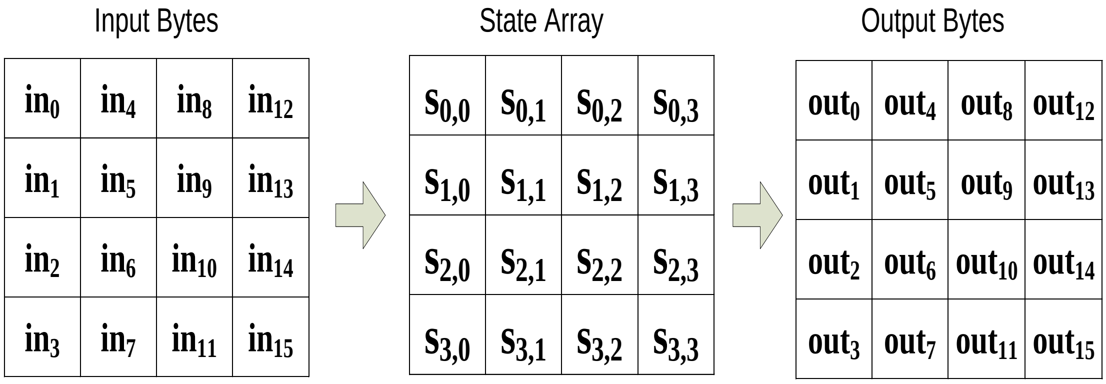
AES Version	Key Length (N_k words)	Block Size (N_b words)	Number of Rounds (N_r rounds)
AES128	4	4	10
AES192	6	4	12
AES256	8	4	14

Table 1 – AES Variations

The basic processing unit for the AES algorithm is a byte. As a result, the plaintext, ciphertext and the cipher key are arranged and processed as arrays of bytes. For an input, an output or a cipher key denoted by a, the bytes in the resulting array are referenced as a_n , where n is in one of the following ranges:

Block length = 128 bits, 0 <= n < 16
Key length = 128 bits, 0 <= n < 16
Key length = 192 bits, 0 <= n < 24
Key length = 256 bits, 0 <= n < 24

All byte values in the AES algorithm are presented as the concatenation of their individual bit values between braces in the order {b7, b6, b5, b4, b3, b2, b1, b0}. All the AES algorithm operations are performed on a two dimensional 4x4 array of bytes which is called the State, and any individual byte within the State is referred to as $s_{r,c}$, where letter ‘r’ represent the row and letter ‘c’ denotes the column. At the beginning of the encryption process, the State is populated with the plaintext. Then the cipher performs a set of substitutions and permutations on the State. After the cipher operations are conducted on the State, the final value of the state is copied to the cipher text output as is shown in the following figure.



Input Bytes

in_0	in_4	in_8	in_{12}
in_1	in_5	in_9	in_{13}
in_2	in_6	in_{10}	in_{14}
in_3	in_7	in_{11}	in_{15}

State Array

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

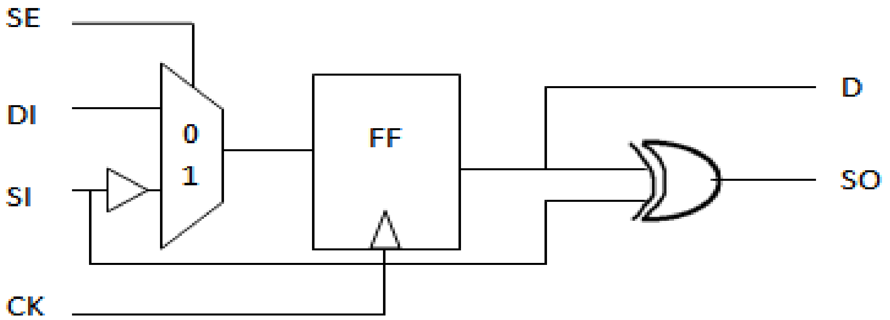
Output Bytes

out_0	out_4	out_8	out_{12}
out_1	out_5	out_9	out_{13}
out_2	out_6	out_{10}	out_{14}
out_3	out_7	out_{11}	out_{15}

Figure 1.1 State Population and Results

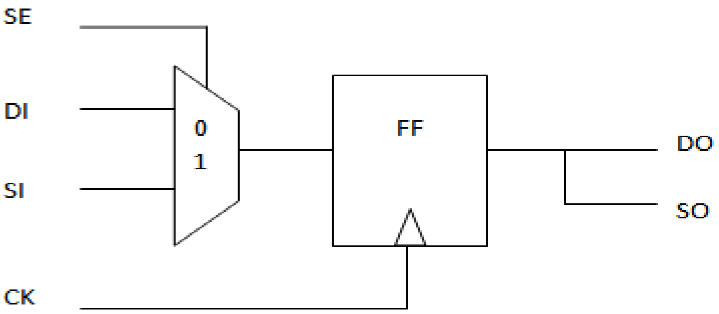
Block Diagram Explanation:

Due to the security and test ability requirements as mentioned above, a novel robust secure scan-based test approach is proposed as a counter measure against scan-based differential cryptanalysis.

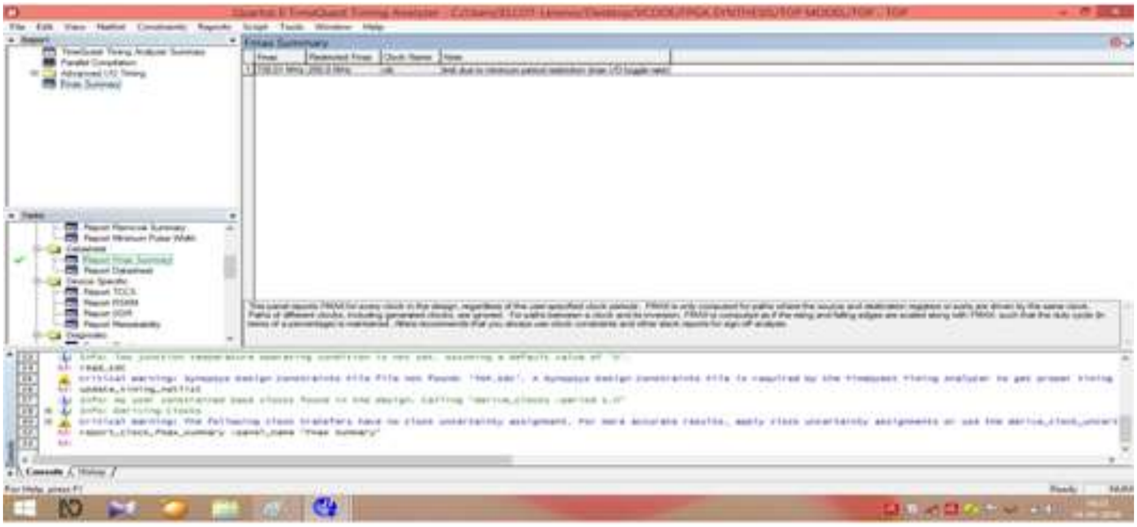


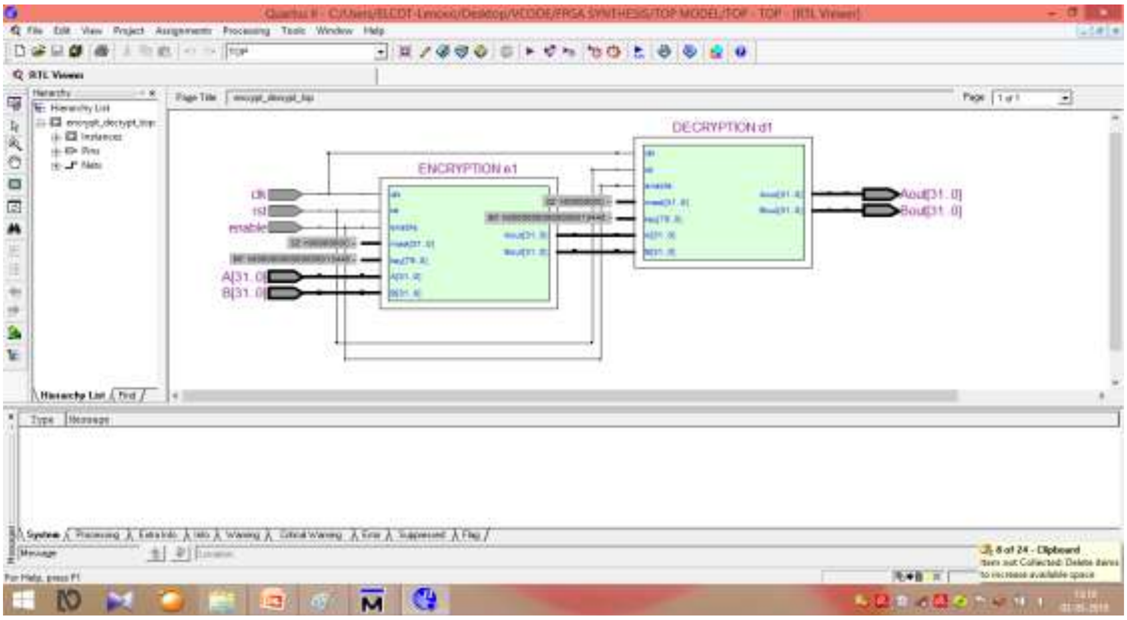
side channel attacks:

Scan test has been widely adopted as a default testing technique among most VLSI designs, including crypto cores. Unfortunately, these scan chains might be used as a “side channel” to recover the secret keys from the hardware implementations of cryptographic algorithms, for example scan-based attacks on Data Encryption Standard (DES), Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) have been illustrated respectively.



performancereport





II. Conclusion

Here in this we carried out implementation of PRESENT cryptographic algorithms with scan based testing futures. It has been previously demonstrated that scan chains introduced for hardware testability open a back door to potential attacks. Here, we propose a level based masking and RSFF based flip flop masking as a scan-protection scheme that provides testing facilities both at production time and over the course of the circuit's life. Compared to regular scan tests, this technique has no impact on the quality of the test or the model-based fault diagnosis. Here we proved that RSFF based AES will give better hardware complexity & power optimization with considerable delay enhancement. An accurate SFF-based analysis approach was introduced for crypto core with single and multi FF characterizations. The proposed approach was derived from the SFF method. The method avoids the use of a large number of masking parameters to minimize the required resources for area- and power-efficient built-in testing applications. Modelsim based pre simulation results of a crypto implementation showed the feasibility of the approach. For a QUARTUS II based hardware synthesis report proved the efficiency of proposed method.