CHAPTER 4

AN IMMUNE AES AGAINST DPA

4.1 OVERVIEW

Cryptographic devices are exposed to different types of side-channel attacks. The attacks due to leakage of information are carried out by calculating either one of the sweep out power emission or leakage of electromagnetic emission during cryptographic process, and exploiting the secret key on the cryptographic device. The Differential Power Analysis (DPA) introduced by Kocher et al (1999) and the correlation power analysis (CPA) introduced by Brier et al (2004) are known to be the most prevailing and well-known attack techniques in the side-channel attacks. The two defensive measures against side channel attacks were Hiding and Masking. The hiding aims at matching the power utilization during cryptographic processing and masking aims at randomizing the side-channel leaked data.

4.2 AES DESCRIPTION

AES has become the well accepted substitute for its antecedent and is at present used in a various applications where safety is a concern. Because of these properties AES is made another better algorithm to work with power analysis of side-channel attacks. The flow chart that follows depicts a typical AES with 128 bit key cipher in operation.
An AES-128 cipher starts with a 128-bit block of plaintext along with a 128-bit key and executes ten AES rounds to calculate a 128-bit ciphertext as shown in Figure 4.1. In contrast with DES which is based on a Feistel network, AES is based on a replacement and permutation network. In each stage of AES, the individual value in each round is computed on a 4x4 byte array known as the state. In every round of AES which include four main stages; SubBytes, ShiftRows, MixColumns and AddRoundKey. The Rijndael key schedule is used in AES for key scheduling to allow for a dissimilar sub-key to be generated for each round.

### 4.3 POWER ANALYSIS ATTACK

Few methods to carry out power analysis are

- SPA – Simple Power Analysis
4.3.1 SPA

It is performed along time axis the attacker tries to obtain the key with one or few trace. Likewise to visual inspection of power trace (or) template base SPA attack Visual inspection only have possibility of success. Attack could be micro controller implement of AES cipher where round function are implemented with instruction of microprocessor like add, xor, branch memory read or memory write instruction occurs when key bit is one and another instruction occurs only when that bit is zero. It is feasible to obtain the secrete key.

Power utilization depends on the data. This is defined by a mean vector and a covariance matrix in DPA and CPA.

The following steps are used in both types of attack are

1. Select an intermediate result of encryption
2. Compute the power consumed for that operation
3. Determine the hypothetical in-between value
4. The intermediate value is to be mapped to the power consumption value
5. Compare the hypothetical power consumption value with power traces

DPA or CPA attack starts with assembling of power traces. The steps involved are explained in Figure 4.2.
Encryption or decryption power traces are recorded with oscilloscope and known input data (plain text) stored data which is selected arbitrarily by random number generation. The calculation of hypothetical intermediate values was done and these values are mapped on power model. This map model result gives theoretical power utilization. The sample data of the calculated power trace result in a correlation with theoretical power consumption.

Figure 4.2 Steps of CPA and DPA
4.3.2 Differential Power Analysis

A differential power analysis (DPA) attack is analysed by correlating the power consumption of a chip in a smart card and the secret key used for encryption algorithm. Differential Power Analysis (DPA) Kean Rong Boey et al (2010) and Paul Kocher et al (1999) attack needs large set of power consumption traces measured while for different data inputs the device is operational. The utilized power during a fixed amount of time reveals high data information. DPA is most widely used attack than SPA, it uses statistical analysis and noise modification techniques and no depth of acquaintance about the cryptographic device is needed. The essential condition for the attack to be powerful is the existence of one or more in-between variables in the algorithm that can be given as a function depending on a few number of key bits and on known input or output information known and by calculating and comparing some easy statistic, such as the average value, on the partitioned curve at some points in time. The unbeaten attack exists if the exact guess about the key bits result in a major differentiation between the average curves at one or more number of points in time.

In addition to large-scale power variations due to the instruction sequence, there are manipulated data values with correlated effects. The variations are small values and are sometimes overshadowed by measurement errors and other noise. In these cases, it is still often possible to break the system using statistical functions tailored to the target algorithm. To implement the DPA attack, an attacker first observes m encryption operations and captures power traces T1 : : : m[1 : : k] containing samples each. The attacker records the cipher text 1 : : : m. The attacker computes ak -sample differential trace D[1 : : k] by finding the difference between the average of the traces for which a certain intermediate value V is one and the average of the traces for which is zero. Thus D[j] is the average over C [1 : : m] of the
effect use to the value represented by the selection function $D$ on the power consumption at point $j$.

In particular,

$$\Delta_D[j] = \frac{\sum_{i=1}^{m} D(C_i, b, K_s) T_i[j]}{\sum_{i=1}^{m} D(C_i, b, K_s)} - \frac{\sum_{i=1}^{m} (1-D(C_i, b, K_s)) T_i[j]}{\sum_{i=1}^{m} (1-D(C_i, b, K_s))}$$

$$\approx 2 \left( \frac{\sum_{i=1}^{m} D(C_i, b, K_s) T_i[j]}{\sum_{i=1}^{m} D(C_i, b, K_s)} - \frac{\sum_{i=1}^{m} T_i[j]}{m} \right)$$

(4.1)

If $K_s$ is incorrect, the bit computed using $D$ will differ from the actual target $b_i$ for about half of the cipher text $C_i$. The selection function is thus effectively uncorrelated to what was actually computed by the target device. If a random function is used to divide a set into two subsets, the difference in the averages of the subsets should approach zero as the subset sizes approach infinity.

Thus, because trace components uncorrelated to $D$ will diminish with $1/p_m$, causing the differential trace to become at (the actual trace may not be completely at, as $D$ with $K_s$ incorrect may have a weak correlation to $D$ with the correct $K_s$). If $K_s$ is correct, however, the computed value for $D(C_i: b: K_s)$ will equal the actual value of target bit $b$ with probability 1. The selection function is thus correlated to the value of the bit considered. Other data values, measurement errors, etc., that are not correlated to $D$ approach zero. Four values of $b$ correspond to each $S$, providing confirmation of key block guesses. After finding all eight $K_s$ yields the entire 48-bit round sub
key. DPA can use known plaintext or known cipher text and can find encryption or decryption keys.

The superior level of DPA is the higher-order DPA attack. It is a generalization of the DPA attack in which the power consumption traces are processed by applying statistic methods to more points in time. DPA attack is more powerful than other types of attacks but can be more complex. DPA attacks calculates the power levels consumed at different parts of the chip and apply statistical analysis to overcome countermeasures, such as noise addition, to obscure individual bits it is applied. The identification of kind of computational operations done by a device is done through power usage measurement. An analysis will disclose several bits of the crypto key at a time; the process is repeated to ultimately produce the entire key. The DPA attacks are hazardous because they avoid the hardware and software security that vendors have put in place. Because these types of attacks are non-invasive, it's possible for an intruder to negotiate an embedded system without leaving a trace.

To avoid DPA attacks, experts suggest that enterprises use smart cards that will erratically generate a new key from the old key each time the card is used, thus rendering captured keys become ineffective.

4.3.3 Power Characteristics

The possibility of power analysis attack depends on the availability of power approximation method using hamming distance and hamming weight power models. The DPA power trace database is shown in Figure 4.3 and depicts the 10 rounds of AES-128 encryption.
Figure 4.3 A sample of AES power trace which shows the power consumption for ten rounds of an AES-128 encryption

Differences in instantaneous power utilization are related to the bit values that are being manipulated. As bit values change, the related essential hardware, consumes power on a much lower scale. Due to slight power deviation, detection becomes more difficult. It requires modifications to the hardware or statistical techniques to identify and correlate the values. As more number of data bit changes from logic 0 to logic 1, more power is consumed.

Power models map the data value process to power utilization. The familiar Power models are (i) Hamming weight as expressed in equation 4.3.

\[ hw(d) = d(j), d(j-1) \]  \hspace{1cm} (4.3)

If \( d \) contain \( m \) independent and consistently dispersed bits, the entire word has an common hamming weight of \( \mu_h = m/2 \) and a variance \( = m/4 \) the data values before or after this data value are ignored.
(ii) Hamming distance

It is based on the switching of bits one state to another state. If there is no difference between 0 to 1 and 1 to 0

\[ \text{HD}(X, X') = \text{HW}(X \oplus X') \quad X' = 0 \text{ and when } X \text{ is uniform random variable} \]

It is robust for CMOS technology.

\[ W = a \cdot \text{HW}(X \oplus X') + b \]  \hspace{1cm} (4.4)

b-independent variable; a-is a scalar gain between the hamming distance and w is consumed power during transitions 0-1 and 1-0 as shown in equation 4.4.

4.4 FUNDAMENTALS OF DPA ATTACK

The fundamental of differential power analysis attack described here relies on two properties of the AES encryption. The first property is the truth that the individual AES S-box outputs create responsive information that can be correlated to the power consumed data recorded from the end device. The second one is the fact that each AES S-box input uses only 6 bits of the 48-bit subkey used for that round, which is small enough that they may be intensely searched to obtain the highest correlation. The main objective of the attack implemented is to disclose the secret key used in the first round. Once this subkey is identified, the full key used in the AES encryption can be found by general search methods. The DPA attack schedule is done in the similar way to the power characteristics program, however, this time we must exactly guess and find out the key bits used and select those that produce the highest correlation for every S-box value. For each S-box, we select the 6 bits from
the original plaintext that is used, which is found by the initial permutation and the expansion permutation inside the Feistel function. For example, in the first S-box these bits are (7, 57, 49, 41, 33, 25) of the plaintext. Because the output of the S-box is dependent on the unknown 6-bit key, we simply generate an array of $2^6$ potential S-box outputs, one for each key presumption (from 0 to 63). For each potential 4 bit S-box output, the Hamming distance is computed between that output and the consequent bits in the original plaintext. The end result is an array of 64 sensitive data values, one for every potential key. With the help of correlation coefficient equation, the correlation between each one of the 64 sensitive data values and the calculation of power consumed information is done. This process is repeated after the construction of many power traces in order to find the correlation coefficient. After an adequate number of traces are obtained, then we can observe the existence of a peak in the correlation which corresponds to the correct key.

The general advantages of DPA attack is described in this section. The attack can be done on a known-plaintext or known-ciphertext situation, which is inherently better than attacks that require plaintext and its corresponding ciphertext pair. The DPA attack is also extremely quick when compared to other extensive search methods, taking meager minutes as opposed to several days for exhaustive key searches. The main weakness of the DPA attack is that it requires physical access to the device and it requires noiseless power traces from the target device. Even if noise is present that can be reduced in the power trace acquisition by obtaining many samples using the same plaintext and key and averaging them.

DPA performs intermediary value of single bit (or) a set of several bits. Generally attack device requires a lot of power traces which shows power utilization. The First order DPA attack is done at one time, whereas higher order DPA is done at many times. Higher order DPA is performed on
masked encryption. Paul Kocher has explained about DPA and evaluated the difference of means of power traces. The relationship between the H and the T columns using a binary matrix H Manipulation logic bit. Logic 1 is different in terms of power consumption than manipulating a bit to a logic 0. Here m is the number of bits.

\[ m_{1,i,j} = \frac{1}{n_{1,i}} \cdot \sum_{j=0}^{n} h_{1,i,j} \cdot t_{1,i,j} \]

\[ m_{0,i,j} = \frac{1}{n_{0,i}} \cdot \sum_{j=1}^{n} (1 - h_{1,i,j}) \cdot t_{1,i,j} \]

\[ n_{1,i} = \sum_{j=1}^{n} h_{1,i} \]

\[ n_{0,i} = \sum_{j=1}^{n} (1 - h_{1,i,j}) \]

\[ R = M_1 - M_0 \]

4.5 PROPOSED WORK

The DPA is the most prevailing method based on statistical analysis and characteristics of the captured power traces. It is more efficient even when the algorithm is unknown. It is generally executed by comparing the power traces captured using the known/unknown key. They are correlated within each other to guess the correct key.

When analysis is made based on why attack is made possible, it is found that

- Usage of registers
- Availability of pure cipher text
- Power variation/correlation provided during dispreading with matched key even when architecture is unknown

- Approximately 9000 power traces are required

4.5.1 Counter Measures

The power consumption is independent of intermediary data value of the cryptographic counter measure that cannot assure that attack will be saved. It is intended to make the life of the attack more difficult using Protocol counter measure or Implementation counter measure.

- Protocol counter measure

The cryptographic protocol changes in such a way that attacker is unable to recover enough leakage.

- Implementation counter measure

Here the protocol remains unchanged. It is mainly anxious about the implementation that is done by masking and hiding.

4.5.2 Masking

Masking is a counter measure that randomizes the intermediary values that are processed by cryptographic devices. “The masking counter measure, for each intermediary value v is hidden with a random value m, is called mask”. A logical xor is also known as modular addition.

\[ V_m = v + m \mod(n) \]
When $n$ is defined by cryptographic algorithm another masking technique is modular multiplication $V_m = v^* m \pmod{n}$. Masking implementation can be attacked using higher order DPA or CPA attacks.

### 4.5.3 Hiding

In Hiding, one can conceal information by processing the sensitive data at random time intervals. Another hiding is to have some fake logic running that creates extra noise.

**Costs of countermeasures**: Countermeasure comes with a cost. Although security is an important factor, it is viable like performance, power and area. “The challenge is not to build a secure system, to build a secure system with acceptable cost.”

**Side channel analysis tool**: Side channel attacks, called inspector. The Inspector supports various cryptographic cryptanalysis algorithms, various data acquisition modules, filtering tools editing, statistical analysis and more.

### 4.5.4 SCA Vulnerability of AES S-Box and Existing Counter Measures

Differential power analysis (DPA) is an important type of side-channel attack. DPA attack can take out secret keys through statistically analyzing power consumption measurements from a cryptosystem Bodhisatwa Mazumdar et al (2012). For the DPA attack, normally the following steps are carried out by the attackers: (1) First gather the power consumption measurements from the encrypted device with arbitrary inputs; (2) Second, classify the collected results by using deciding function; (3) repeat (1) with a theoretical key; (4) Fourth step, sort out the results to the present existing sets; (5) As a fifth step, find the average power calculation in
each sets; (6) Match the different results until the correct key is found. If the theoretical key is the real key, it can be identified apparently by the large spikes in the differential power traces. Else, we can come to the conclusion that the key is incorrect. S-Box is the most crucial component, and it helps to determine the power consumption and throughput of not only the Sub Bytes operation but also the AES hardware implementation. Therefore, our research focuses on the S-Box design.

**Counter measures against power analysis attacks:** DPA attacks are possible, because the power consumption depends on the intermediary results of the executed algorithm. Therefore there is an endeavour to make the consumption of the cryptographic devices data-independent or at least to reduce the consumption. In order to protect the cryptosystems against power analysis attacks, two appropriate approaches were summarized in Mehdi Masoomi et al (2010) and Zheng Yuan et al (2011): hiding and masking. How these countermeasures smash the connection between the intermediate results of cryptographic algorithm and the power consumption. We can differentiate them as software and hardware counter measures. Further attention will be devoted to the software implementations.

**Hiding approach:** Hiding techniques try to eliminate the link power utilization the processed intermediary values and the operations were performed. Link is broken down when one of the following conditions is fulfilled.

1. Arbitrary amount of power is consumed in each clock cycle of the device.

2. The same amount of power is consumed in each clock cycle of the device.
4.6 SIMULATION RESULTS

To attain these conditions the operations of the executed algorithm are performed at different moment of time for each execution or power utilization characteristics of the operations are directly changed. The simulation results for power traces captured are shown below in Figures from 4.4 to 4.7.

Figure 4.4 Captured Power variation during the final round of the AES Encryption

Figure 4.5 Captured power variation in register
Figure 4.6 Initial trace without register power variation

Figure 4.7 Power Trace for Combinational Logic Circuit Design

In order to prevent information from hackers impure cipher text is generated as shown in Figure 4.8, so that it is made complicated for them to reveal original plain text.
The description of the measured parameter for the proposed method is compared with conventional methods shown in Table 4.1.

Table 4.1 Number of Power Traces for Proposed Vs Existing Methods

<table>
<thead>
<tr>
<th>Author</th>
<th>No. of Power Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chin Chi Tiu[19]</td>
<td>9000</td>
</tr>
<tr>
<td>Yu HAN et al.[85]</td>
<td>6000</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>17230</td>
</tr>
</tbody>
</table>

4.7 ANOTHER DPA COUNTERMEASURE PROPOSED

The main purpose of the DPA countermeasure circuit is to break the dependency between the measured power traces and the predicted power values. As shown in Figure 4.9, 16 identical subcircuits were contained in the DPA countermeasure circuit. Each subcircuit, composed of 12 digitally controlled ring oscillators which are controlled to randomly enable different number of ring oscillators. A global enable signal is also applied to turn off the subcircuit to reduce power consumption.
The random number generator is designed based on linear feedback shift registers (LFSRs) with dynamic feedback configuration to make the random sequence more unpredictable. Each subcircuit is controlled by a data byte of the AES data block and the random byte from the pseudo random number generator.

When a ring oscillator is enabled, it will consume additional power to change the power consumption characteristic. An 8-bit input is obtained by XORing one data byte with a random mask, and eight ring oscillators are directly controlled by this 8-bit input. An internally designed random number generator is generated using random mask, whose randomness dominates the DPA resistance of our proposed countermeasure circuit. The other four oscillators are controlled by pairs of these eight inputs. This way, the amount of power consumption added to the whole chip corresponds to the masked data.

LFSR are very predictable. Predicting the next output bit is equivalent to knowing the feedback pattern of the LFSR and the states.
Mathematicians (Berlekamp-Massey) found that given an N-bit LFSR with unknown feedback pattern, then only 2N bits are needed to predict bit 2N + 1. So let's say we have an 8-bit LFSR, then we need only 16 bits of the RNG stream before it becomes predictable. The LFSR are unsuited for everything that should be Unpredictable.

**Figure 4.10 States of LFSR.**

The Figure 6.2 shows all 15 states of LFSR where the 16th state is the same as the first state. The real issue for PRNG is that the value of bit N+1 be predicted when someone observes the first N bits.

4.8 **OSCILLATOR-BASED CIRCUIT**

**Figure 4.11 Oscillator-based circuit**
To be unpredictable, the LFSR should be long we have many techniques of using maximal-length \( P(x) = x^N + \ldots + 1 \) with \( N \gg 64 \) (E.g. 65,536). But this becomes very expensive to design 64K flip-flops. The next technique is the use of Non-linear Combination Generator.

**Figure 4.12 Non-linear Combination Generator**

### 4.9 NON-LINEAR COMBINATION GENERATOR

Non-linear Combination Generator consists of 43-bit LFSR and 37-bit cellular automata shift register (CASR). The 43-bit LFSR defined by

\[
P(X) = X^{43} + X^{41} + X^{20} + X + 1
\]

Where we need to have only 3XOR, 43 taps with a Maximal Length of \( 2^{43}-1 \). Since all-zero pattern cannot appear we have a Bias of \( 2^{43} \).
The 37-bit cellular automata shift register combines previous and next state register into current state register which is similar to 37 intertwined state machines (automata). Thus the CASR has a Maximal Length of $2^{37}-1$ and the Bias of $2^{-37}$. Therefore we can design a 32 bit random sequence with this Combined Generator.

However, there is still a security weakness with this architecture. During the system reset, pseudorandom number generator would be the same. Therefore, the additional power consumption added by the DPA countermeasure circuit in each cycle would be the same if the attacker resets the system before recording power traces.

### 4.10 PROPOSED DPA COUNTERMEASURE CIRCUIT

To overcome the security weakness in the pseudo random-based architecture, there is a need for true random sequence for the DPA countermeasure circuit. But, most of the true random number generators are analog circuits whose power consumption are very much higher. A digital...
method to generate random data by using ring oscillators in Fibonacci and Galois configurations was proposed by Galois.

![Architecture of the DPA countermeasure circuit](image)

**Figure 4.14 Architecture of the DPA countermeasure circuit**

Instead of designing an extra random number, the proposed DPA countermeasure circuit shown in the Figure 4.14 can generate by itself a true digital random sequence.

The ring oscillators in the DPA countermeasure circuit after some modifications can be shared as random sources of the true random number generator. The proposed DPA countermeasure circuit consists of four Galois ring oscillator sets (GaRO), four Fibonacci ring oscillator sets (FiRO), and eight postprocessing circuits. The Fibonacci and Galois Ring Oscillator sets are composed of four Fibonacci and Galois ring oscillators, respectively. The random and data bytes directly control the 12 3-stage ring oscillators which forms the DPA countermeasure circuit to dynamically change the power consumption. Hence, an extra random number generator is required.

Two practically important concretizations which are generalizations of the ring oscillator structure are proposed. Other concretizations, are also possible based on LFSRs or programmable linear cellular automata. Based on
a cascade of inverters, both the generalizations replace the simple circular feedback defining a ring oscillator by a more complex feedback incorporating a number of XOR logic gates. The complex feedback corresponds to the well-known Fibonacci or Galois configurations of an LFSR. Instead of the delay elements the inverters are used.

The feedback connections are chosen in such a way that there are no fixed points in the corresponding state-transition function to satisfy the basic condition. The individual inverters delays are the same and the delays could be adjusted by composing each of them as a cascading odd number of elementary inverter logic gates. Both generalizations reduce to the classical ring oscillator when the number of inverters is only one.

### 4.11 FIBONACCI AND GALOIS RING OSCILLATORS

![Fibonacci ring oscillator](image)

**Figure 4.15 Fibonacci ring oscillator**

The Fibonacci ring oscillator, shown in Figure 4.15, consists of a number ‘r’ of inverters connected in a cascade so that the output of each but the last inverter is directly used as the input to the next inverter. The feedback connections are specified by the binary coefficients \( f_i \) with the convention that the corresponding switch is closed if \( f_i = 1 \) and open if \( f_i = 0 \), in which case, the corresponding 2-input XOR gate is not present. The feedback coefficients and the corresponding inverters are indexed from left to right.
The Galois ring oscillator, shown in Figure 4.16 consists of a number ‘r’ of inverters connected in a cascade so that the output of each except the last inverter is used to form the input to the next inverter and the output of the last inverter directly defines the feedback signal. The binary coefficients in the ring oscillator \( f_i \) is similar to Fibonacci ring oscillator. The feedback coefficients and the corresponding inverters are indexed from right to left.

### 4.12 COMBINED OSCILLATOR

The robustness as well as randomness can be further increased by XOR-ing the outputs of two oscillators, one from the Galois and the other from the Fibonacci configuration. Thus the output signals properties of the two configurations are also combined.
The corresponding elementary RNG, including a sampling unit implemented as a D-type flip-flop, is shown in Figure 4.17. The lengths of the two oscillators minus one should preferably be mutually prime, first, for the period of the corresponding pseudorandom sequence to be maximized and, second, for the interlocking or coupling effect to be minimized. In particular, it is suggested that the lengths differ only by one, where the even length corresponds to the Fibonacci ring oscillator.

The random source to generate one random sequence is generated by the combination of two FiROs and two GaROs. A total of 32 ring oscillators (including Fibonacci and Galois ring oscillators) are required in the DPA countermeasure circuit for generating 8 independent random bits for each data. These eight random sources are sampled by flip-flops for further postprocessing. After postprocessing, these eight random bits are XORed with data bytes from the cryptographic circuit to dynamically enable oscillators in the FiRO and GaRO. The FiRO and GaRO work as random sources to generate random data and also work as the digitally controlled ring oscillators to counteract the DPA attack.

The proposed architecture incorporates a true random number generator into the DPA countermeasure circuit to resist the DPA attack and the reset problem mentioned earlier system by postprocessing circuits. The postprocessing circuits are composed of LFSRs with different initial seeds. The postprocessing circuit removes the bias of the random source. The feedback value is XORed with that from the random source in each postprocessing circuit. Thus after the system is reset, the postprocessing circuit starts from a deterministic state and the generated random sequence would also be different as the random source is added into the feedback of the LFSR.
4.13 TOOLS USED FOR SIMULATION

The Integrated Software Environment (ISE®) is the Xilinx® design software suite that allows taking the design from design entry through Xilinx device programming. The ISE Project Navigator is used to manage and process the design through the following steps in the ISE design flow. Our project is simulated using VLSI software Modelsim Altera 6.6c and synthesized using XILINX ISE 9.2i device.

4.13.1 Design Entry

Design entry is the first step in the ISE design flow. During design entry, source files are created based on the design objectives. The top-level design file can be created using a Hardware Description Language (HDL), such as VHDL, Verilog, or ABEL, or using a schematic. Multiple formats can be used for the lower-level source files in the design.

4.13.2 Synthesis

After design entry and optional simulation, synthesis is done. During synthesis, HDL language designs become netlist files which are accepted as input to the implementation step.

XCF

XST Constraint File (XCF) is the one in which synthesis, timing, and specific implementation constraints are specified that are to be propagated to the NGC file.
**RTL Schematic**

This is a schematic representation of the pre-optimized design shown at the Register Transfer Level (RTL). This representation is in terms of generic symbols, such as adders, counters, multipliers and gates, and is generated after the HDL synthesis phase of the synthesis process. Viewing the schematic may help to discover design issues early in the design process by viewing an RTL Schematic - XST.

**Technology Schematic**

This is a schematic representation of an NGC file shown in terms of logic elements optimized to the target architecture or "technology". It is generated after the optimization and technology targeting phase of the synthesis process. Viewing this schematic allows to see a technology-level representation of the HDL optimized for a specific Xilinx architecture. This schematic helps to discover design issues early in the design process by viewing a Technology Schematic - XST. In Incremental Synthesis mode, multiple NGC and NGR files are generated by XST, which each represent a single user design partition.

**Core Files**

These files can be in either NGC or EDIF format. Cores are not modified by XST. It uses them to inform area and timing optimization surrounding the cores.

**Synthesis Report**

This report contains the results from the synthesis run, including area and timing estimation by Viewing Messages and Reports
4.13.3 Implementation

After synthesis, design implementation is run, which converts the logical design into a physical file format that can be downloaded to the selected target device. From Project Navigator, the implementation process can be run in one step, or each of the implementation processes can be run separately.

Implementation processes vary depending on the target if it is a Field Programmable Gate Array (FPGA) or a Complex Programmable Logic Device (CPLD).

Verification

The functionality of the design can be verified at several points in the design flow. Simulator software can be used to verify the functionality and timing of the design or a portion of the design. The simulator interprets HDL code into circuit functionality and displays logical results of the described HDL to determine correct circuit operation. Simulation allows creating and verifying complex functions in a relatively small amount of time. In-circuit verification can also be done after programming the device.

4.14 SIMULATION AND SYNTHESIS RESULTS

The algorithm proposed in this chapter is simulated then implemented by using Verilog language, while reduce the number of input/output ports which reduces the chip area and to increase the frequency thereby reducing delay.
SIMULATION USING MODELSIM ALTERA 6.6C PLATFORM

All files are compiled by using Modelsim Altera 6.6c and then the waveforms are obtained from wave block.

Figure 4.18 Simulation of pseudorandom sequence generator

The random number generator is designed based on linear feedback shift registers (LFSRs) with dynamic feedback configuration to make the random sequence more unpredictable. Each sub circuit is controlled by a data byte of the AES data block and the random byte from the pseudo random number generator. Figure 4.18 shows the simulation of pseudo random sequence generation of 32 bit sequence. The function used for the generation of the above sequence is \( F(x) = x^{32} + x^{22} + x^2 + 1 \).

Figure 4.19 Simulation of modified pseudorandom generator
Figure 4.19 shows the simulation of pseudorandom sequence generator of 4 bit which shows that the random sequence is repeated after 15, the $2^n-1$th simulation.

Figure 4.20 shows the simulation of pseudorandom sequence generator to produce 128 bit random sequence.

Figure 4.21 Combined simulation of AES and PRNG
Figure 4.21 shows the simulation of AES algorithm with the key provided by pseudorandom sequence generator.

Figure 4.22 Simulation of Firo Garo

The true random number generators are analog circuits with much higher power consumption. Galois proposed a digital method to generate random data by using ring oscillators in Fibonacci and Galois configuration. The Fibonacci and the Galois ring oscillator consists of a series of inverters connected with feedback polynomial \( f(x) = \sum_{i=0}^{r} f_i x^i \)  \( f_0 f_r = 1 \)

The coefficient \( f_i = 1 \) indicates that the path is connected, whereas \( f_i = 0 \) indicates no connection. Figure 4.22 shows the simulation of basic Firo Garo combination to produce the desired polynomial sequence.
Figure 4.23 Simulation of post processing using LFSR

The post processing circuits are composed of LFSRs with different initial seeds. The purpose of the post processing circuit is to remove the bias of the random source. In each post processing circuit, the feedback value is XORed with that from the random source. In this way, even the post processing circuit starts from a deterministic state after the system is reset, the generated random sequence would not be the same because the random source is added into the feedback of the LFSR.

Figure 4.23 shows the simulation of post processing circuit using LFSR and CASR to generate unpredictable random bits.
The randomness is combined with pseudo random and, in the latter case, new randomness can possibly be introduced by effectively sampling multiple ring oscillator signals at various points in the LFSR circuit. If the ring oscillator signals are used only to clock linear circuits, then the RNG sequence may be predictable by guessing the limited phase or frequency uncertainties and by solving the linear equation.

Figure 4.24 shows the simulation of true digital random sequence generator which generates unpredictable the key needed for AES algorithm.
Figure 4.25 Simulation of modified generator

Figure 4.25 shows the simulation of modified true digital random sequence generator which generates unpredictable the key needed for AES algorithm.

Figure 4.26 Combined simulation of AES and True DRNG
The proposed DPA countermeasure circuit that can generate a true random sequence of itself. Since the DPA countermeasure circuit is composed of several digital ring oscillators, these oscillators can be shared as random sources of the true random number generator after some modifications. Notice that the proposed DPA countermeasure circuit consists of four Fibonacci ring oscillator sets (FiRO), four Galois ring oscillator sets (GaRO), and eight post processing circuits.

Figure 4.26 shows the simulation of AES algorithm for encryption with the key provided by true random sequence generator.

Figure 4.27 Combined simulation of AES and modified True DRNG

Figure 4.27 shows the AES encrypted data using modified digital random generation, which includes Firo Garo ring oscillators and postprocessing logics. The security level of an AES engine can be improved by the proposed DPA countermeasure circuit. In addition, another minor improvement is that the area overhead can be reduced due to hardware sharing of ring oscillators for generating random power and random sources.
4.15 SYNTHESIS RESULT OF PRNG

The synthesis result of the pseudorandom generator to provide the pseudorandom key is given below in Figure 4.28.

**Figure 4.28 Synthesis result of PRNG**

The Figure 4.28 shows the pseudorandom sequence generation with 136 number of flipflops.

**Synthesis Result of True Drng**

The synthesis result of the true digital sequence generator is given below.
Figure 4.29 Synthesis result of the True DRNG

Figure 4.29 shows the simulation of true digital random sequence generator which generates unpredictable the key needed.

Synthesis Result Of Modified Generator

The synthesis result of the modified generator is given below in Figure 4.30.
Figure 4.30 Synthesis result of modified generator

The synthesis result shows that it uses 132 number of Flip Flops and 130 number of LUTs in the design.

Synthesis Result of AES Algorithm

The synthesis result of the final AES algorithm with the digitally generated random key is given below in Figure 4.31.
Figure 4.31 Synthesis of AES algorithm with digital random sequence

The synthesis result shows the number of slice used and the frequency of operation.

4.16 COMPARISON RESULT

Proposed random generator for AES is compared with existing pseudo random generator for AES.
TIME AND DELAY COMPARISON

Table 4.2 Time and delay Comparison

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Max output period (ns)</th>
<th>Min input period (ns)</th>
<th>Min period (ns)</th>
<th>Best Time Period (ns)</th>
<th>Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prng</td>
<td>4.182</td>
<td>3.419</td>
<td>2.72</td>
<td>2.468</td>
<td>0.087</td>
</tr>
<tr>
<td>Drng</td>
<td>4.182</td>
<td>3.419</td>
<td>3.389</td>
<td>2.738</td>
<td>0.087</td>
</tr>
<tr>
<td>Modified Drng</td>
<td>4.182</td>
<td>4.308</td>
<td>2.394</td>
<td>2.394</td>
<td>0.087</td>
</tr>
</tbody>
</table>

The Table 4.2 shows the Comparison of Proposed random generator for AES with existing pseudo random generator for AES on the basis of time and delay.

Table 4.3 Area and Frequency Comparison

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Slices</th>
<th>FF Slices</th>
<th>LUTs</th>
<th>Frequency(MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prng</td>
<td>79</td>
<td>136</td>
<td>117</td>
<td>367.67</td>
</tr>
<tr>
<td>Drng</td>
<td>70</td>
<td>120</td>
<td>121</td>
<td>295.068</td>
</tr>
<tr>
<td>Modified Drng</td>
<td>76</td>
<td>132</td>
<td>130</td>
<td>417.632</td>
</tr>
</tbody>
</table>

The table 4.3 shows the Comparison of Proposed random generator for AES with existing pseudo random generator for AES on the basis of area and frequency.
Figure 4.32 Comparison of Time Vs Delay

The Figure 4.32 shows the Comparison of Proposed random generator for AES with existing pseudo random generator for AES on the basis of time and delay.

Figure 4.33 Comparison of Area Vs Frequency

The Figure 4.33 shows the Comparison of Proposed random generator for AES with existing pseudo random generator for AES on the basis of area and frequency.
4.17 SUMMARY

Hence the proposed work explains and demonstrates a DPA attack on AES, providing a detailed study on the steps the opponent should perform to make it successful. Using load instruction, xor operation and the average of the load from the SBOX the forth byte of the secret key in AES is publicized. With similar fashion, the rest of the keys can be also revealed. The implementation result shows compact resource utilization of minimum 50% and the increased number of traces. The key extraction is done and the power traces are captured. The proposed work illustrates the considerable reduction in resource utilization and the secret key cannot be guessed even after 17230 traces are when compared to previous method requires 9000 traces. Thus, our proposed method strongly outshine in effectiveness, computational requirement and robustness.

The true digital random number generator is used to solve the problem in pseudorandom architecture. This work not only counteracts the DPA attack but also to self-generate a true digital random sequence. The security level of AES engines can be further enhanced while the area overhead remains same in the architecture. The modification done in the proposed architecture improves the security by hiding the details used for random key generation with reduced area utilization and increased speed.

This new method with digital post processing of raw random data which can provide both randomness extraction and computationally secure has achieved the needed speed and frequency with suitable software implementations. This architecture in addition to speed, uses minimum number of slices thereby reducing the area requirement. The method proposed
is interesting for many security applications of computers and other electronic
devices needing a physical source of true random numbers.