Pak. J. Engg. & Appl. Sci. Vol. 18 January, 2016 (*p.* 31–40)

Increasing Resistance Against Power Analysis Attacks Using Dual Key Scheme

Muhammad Uzair[1], Kashif Javed[2], Haroon A. Babri[3]

1. *University of Engineering & Technology, Lahore, Pakistan*
2. *University of Engineering & Technology, Lahore, Pakistan*
3. *University of Engineering & Technology, Lahore, Pakistan*

* **Corresponding Author:** E-mail: mm_uzair@hotmail.com

Abstract

Execution of a mathematically secure encryption algorithm on hardware is known to leak certain information to the side channels of the hardware. These side channels include current consumed from power supply and electromagnetic radiation emitted from cryptographic hardware. The information thus leaked can be utilized to mount an attack to reveal secret information about the algorithm (e.g. encryption key). This method of extracting the information is broadly classified as "Side Channel Attacks". A type of side channel attack called "Power Analysis" utilizes the power/current consumed information as a source of information leakage. Several measures including "hiding" have been proposed to counter these attacks. These counter measures are based upon inserting randomness or consuming nearly constant current thus reducing the value of this information. In this research, we propose a new hiding countermeasure which uses dual keys to perform cryptographic operations. This method cannot be bypassed by increasing the number of traces.

Key Words: Security, Differential Power Attacks, Hiding countermeasure

1. Introduction

Information security is becoming a very important factor of today's digital life. We are using various facilities like electronic bank transactions, Automatic Teller Machines (ATM), smart phones and digital identities like "smart national ID cards". Eavesdropping has become a serious threat to these transactions. Hack attacks to steal information from people and organizations have become today's norm. Credit card fraud and attacks on banking sectors are also increasing day by day. Various cryptographic algorithms such as Advance encryption standard (AES) [1], Data encryption standard (DES), Rivest Shamir Adleman (RSA) [2] and Elliptic curve cryptography (ECC) [3] are in use. These algorithms are mathematically secure, i.e. calculation of encryption/decryption keys is not computationally feasible given the input plain text and output cipher text. Mathematically these algorithms can only be broken by brute force but this requires huge computational power and time e.g. 10^{13} years [4] for breaking 128 bit AES.

The brute force approach is therefore not tempting for an eavesdropper. However, it was found [5] that when a symmetric encryption algorithm is executed on a PC, it emits certain side channel information in the electromagnetic and electrical current forms, which can be used to gain information about the algorithm. Similar is the case, when a smart card is swiped at some terminal (i.e., card reading device) and the decryption key of the cryptographic algorithm is used to verify the user identity[6]. These attacks, which are highly attractive for the eavesdropper are called side channel attacks (SCA) and were first studied by P Kocher [6], [7]. Kocher used the terms of differential power attacks (DPA) and differential EM attacks (DEMA) to indicate attacks launched on the basis of electrical current and electromagnetic waves, respectively. With the advancements in technology, dedicated hardware implementation of cryptographic algorithms has become popular, due to their efficiency, throughput and small form factor, in the form of ASICs and FPGAs.

In this paper, we have developed a Dual Key model for increasing resistance against these power attacks. This model utilizes a pair of true and false key to perform cryptographic operations simultaneously. By utilizing this model, we have observed the reduction of correlation between true key and power consumed by the cryptographic device. This reduction in correlation value makes it more difficult to recover secret key utilizing DPA attack. This model is implemented in VHDL and using simulations in MODELSIM [8], XPOWER [9] and MATLAB, we have compared the DPA resistance of our model with that of unprotected AES-128 implementation.

The remainder of the paper is organized into seven sections: DPA theory is presented in Section 2.Related work is presented in Section 3. In Section 4, we discuss shortcomings of existing hiding countermeasures. Our proposed model along with its mathematical foundation is presented in Section 5. In Section 6, we have presented implementation details of our evaluation. The performance of the proposed scheme is compared with unprotected implementation in Section 7. Conclusion is presented in section 8 along with future directions.

2. DPA Theory

In this section we present thebasics of DPA theory, which includes source of information leakage and details about mounting a DPA attack.

2.1 How Information Is Leaked

Nowadays most of transistors are made using CMOS technology, the most basic unit of which is a CMOS inverter, shown in Figure 1

The reason behind side channel leakage is simple and intrinsic to CMOS technology. Ideally a CMOS inverter consumes no power when input is at logic ‘0’.It draws electrical current from the power supply (Vcc), when input logic is changed from ‘0’ to ‘1’. This electrical current is used to charge output capacitance ‘Cout’ through gate ‘G1’. Once this Cout is fully charged, it consumes no power, while input is kept at logic ‘1’. It drawsno current from power supply when input is changed from logic ‘1’ to logic ‘0’. During this input change, ‘Cout’ is discharged through G2.

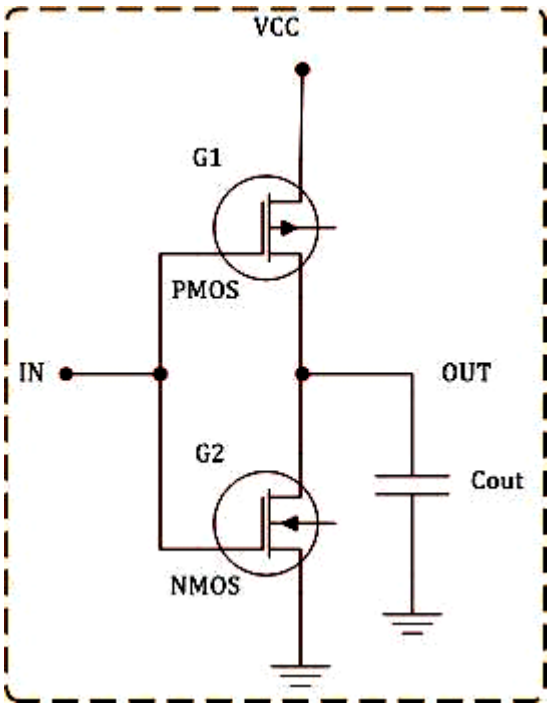

Fig. 1 A CMOS Inverter

In summary, CMOS inverter draws current from power supply, when input is changed from ‘0’ to ‘1’. This data dependent power consumption causes information leakage, which in case of cryptographic operations can be used to find some information about the secret key. Another point worth noting is that current flows through ‘Cout’ during both input combination i.e. ‘0’ to ‘1’ and ‘1’ to ‘0’. This flow of current through ‘Cout’ causes Electromagnetic waves (EM) to be generated [5], [6] and [7]. If attack is launched based on power consumption as source of SCA, only input change from ‘0’ to ‘1’ is event of interest while if attack is launched based on EM as source of SCA, both input changes are events of interest[6], [10]

2.2 Identifying the cryptographic Algorithm

For successful SCA, the attacker needs to have detail knowledge about the cryptographic algorithm. However if implemented algorithm is not known there are ways of getting around this problem. One way is to analyze the power traces to identify the implemented algorithm. Another possibility is to assume an algorithm and apply DPA to recover secret key. Once the key is recovered, it can be used to

verify that the algorithm we assumed is correct or not. This procedure can be repeated to find out the implemented algorithm. A third option is also available, in which the attacker has access to similar hardware e.g. Processor or FPGA. To find the implemented algorithm in such a scenario is pretty straight forward. The attacker has to implement some commonly used cryptographic algorithms in software or hardware and compare the power traces with the device under attack. These types of attacks are sometimes referred as template attacks [10]

2.3 Procedure of Mounting Power Analysis Attack

The setup for DPA is shown in Figure 2. A resistor or current probe is used to measure power consumption of the cryptographic hardware. These measured consumption values are then digitized using an oscilloscope and the digitized data is sent to a PC for statistical analysis. Power consumption data measured for one cryptographic operation is known as a 'Trace'. The number of data values 'D' present in a trace depends on the resolution of the oscilloscope used to digitize that trace. Normally a large number of traces 'N' are captured using this setup. This generates a Power matrix 'P' of size N x D. The attacker then makes a theoretical model of the algorithm under attack which is normally chosen based on intermediate values that are calculated based on device and algorithm knowledge. More details about creating a theoretical model are presented in the next sub section

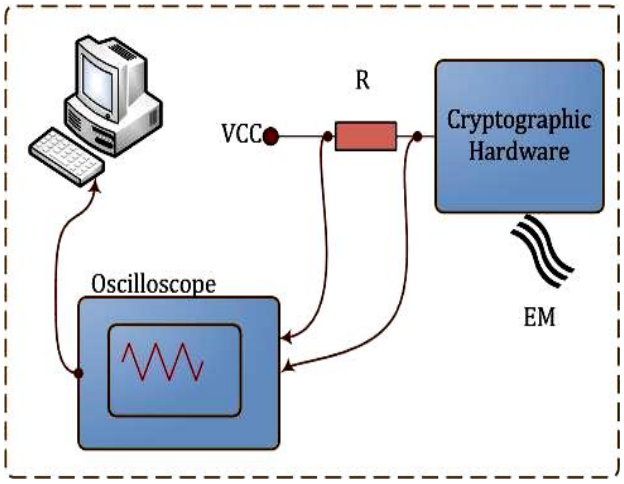

Fig. 2 Setup for DPA

2.4 The Process of Key Recovery

In DPA, the secret key is not recovered at once but it is recovered in chunks of few bits which are referred to as sub key. The size of sub key depends on the algorithm, device architecture and knowledge of device working. Modern cryptographic algorithms operate on byte by byte level, so the choice of 8 bits as length of sub key is realistic. The next step is to identify the weakest spot in the algorithm. For AES algorithm, the first round is considered most vulnerable since first 8 bits of the cryptographic key are directly XORED with the input/cipher text. Therefore the case of 128 bit key, and sub key length of 8 bit, cryptographic key will be recovered in 16 sub keys.

For each sub key, there are 2^8 possible choices. A theoretical model of device under attack (DUA) is used to generate matrices for each of the possible choices. i.e. 2^8 or 256matrices are generated each of size N x D. Once the attacker has these matrices, statistical methods are employed to calculate the correlation between matrix 'P' and matrices that are generated for possible choices of sub key. The matrix that generates highest correlation is the most possible value of sub key. Difference of the mean [6] and Pearson correlation [11] are two methods commonly used for calculating the correlation.

Once the attacker has determined one sub key, this is treated as known data and the theoretical model of DUA is updated accordingly. This ensures that the attacker is going on the right track. By iterating the above procedure, the complete key is recovered.

2.5 Countermeasures Against Power Analysis Attacks

Several counter measures that have been proposed are divided in to two main categories: masking and hiding [11]. In masking, the input data (e.g. data stored on smart card or information to be encrypted) is XORed with some random mask to reduce the correlation between the measured trace and input text. Hiding is a broad category in which the designer tries to hide the internal states by shuffling the execution order, inserting random noise, inserting dummy operations and Dual-Rail Pre-charge logic (DPL). DPL [12] tries to overcome the

data dependent power consumption of CMOS, by consuming same amount of power for each input transition.

3. Related Works

In [13], a countermeasure is introduced that includes random execution ordering scheme which randomly changes the execution order of instructions. This randomness is introduced at hardware level. This method utilizes the byte level independence present in AES encryption. The system does not operate on synchronous clock but instead uses a data driven model, which results in a non linear power trace. This countermeasure can be overcome by increasing the number of traces and trace alignment techniques. A similar technique is utilized in [14], where shuffling of different operation is achieved at algorithmic level and timing for these operations is changed at run time. This technique has very large area overhead and 6 to 7 times reduction in overall system performance.

Inverted ring oscillators are added within synchronous core in [15]. The frequency of these oscillators is also altered at run time to generate unpredictable clock, latency and synchronization problems in power traces. It is claimed that this method also adds some resistance against fault induction and it also does not affect crypto system performance. The complexity of this solution and its integration cost is very high and this countermeasure can be overcome by utilizing trace alignment techniques.

In [16], a true random sequence based ring oscillator is added to AES core. This method has only 6.2% area overhead. It also solves the reset problems which are associated with pseudo random number generation. It uses 12, three stagering oscillators which utilize feedback structure to improve resistance against DPA. The problem with this solution is that it is difficult to implement and just adds random noise to power traces which can be overcome by increasing the number of traces.

In [17], a countermeasure is proposed to randomly change voltage and clock frequency of the circuit to prevent DPA. This countermeasure utilizes the idea of power consumption reduction in digital circuits. Due to clock frequency scaling the exact time of operations is not fixed which produces trace alignment problems. Also due to voltage scaling, randomness is introduced in correlation values. This countermeasure can also be countered by increasing power traces and utilizing trace alignment techniques.

In [18], a hiding countermeasure is proposed in which random delays are inserted in the data path of the cryptographic processor. It changes the instantaneous current and adds random noise to power traces. This countermeasure can be overcome by increasing the number of traces to filter out noise.

In [19], a circuit is presented to suppress information leakage in digital circuit chips. It senses instantaneous current drawn by the crypto circuit and then an equivalent amount of current is shunted from the supply so that total current drawn from supply pins is always constant. The practical implementation of this countermeasure is only able to achieve this goal partially. This countermeasure is highly dependent on operating clock frequency and fabrication method used in chip manufacturing and can be overcome by increasing the number of traces.

According to NSA declassified document [23], it had asked to operate at least 10 ciphering machine at a time, in an effort to mask leaked information. The problem with this scheme is that these machine will be spatially at difference position so each machine will still emit side channel information on power lines as well as in electromagnetic side channels. Even if all the ciphering machines share a single power source, electromagnetic side channel leakage can be captured by appropriating placement of receiver [5].

4. Problems with Other Hiding Countermeasures

In this section, we discuss the problem with the existing hiding countermeasures.

4.1 Relation of SNR with Correlation

The correlation between hypothetical power consumption and exploitable power consumption is derived in [11] as:

$$\rho(H_t, P_{total}) = \frac{\rho(H_i P_e}{\sqrt{1+\frac{1}{4}}} \quad (1)$$

Where H_i is a random variable used to denote hypothetical power consumption of the device, P_{exp} is the exploitable power consumption, P_{total} is the total power consumption and SNR is signal to noise ratio.

It is clear from Eq.1 that correlation is directly proportional to SNR. Higher the SNR will result in stronger correlation values which increases the chance of successful DPA. Since most of the countermeasures discussed in Section 3 are merely adding noise to the system, this relation shows that these countermeasures can be defeated by increasing the number of traces.

4.2 Relation of SNR with number of traces

Another relationship is derived in [11], which relates SNR with number of traces required

$$n = \frac{28}{\rho^2_{ch.ct}} \approx \frac{1}{SNR} \quad (2)$$

Where 'n' is the number of traces required, $\rho_{ck,ct}$ is the actual correlation between theoretical and actual power matrices. Eq.2 clearly shows that if a countermeasure is merely reducing SNR, it can be countered with more number of traces.

5. The Proposed Dual-Key Scheme

As discussed in the previous section, most of the hiding countermeasure focuses on adding noise to cryptographic system which can be filtered out by increasing the number of traces as this noise does not have any correlation with the implemented cryptographic algorithm. Our proposed countermeasure does not merely add non-correlated noise but instead, adds another type of noise which is related to the implemented cryptographic algorithm.

Our proposed model uses a pair of cryptographic keys to perform cryptographic operation (encryption/ decryption) on plain/cipher text simultaneously. One key is called the true key, which is actually responsible for performing cryptographic operations, while the other key, called false key merely adds noise to the cryptographic operation being performed by the true key. It is critical that these two cryptographic cores don not share any resource other than the power supply and clock to avoid any critical information leakage. This also ensures that all the operations are being performed simultaneously by both keys.

This proposed algorithm is adding noise in two dimensions. The first dimension is that more traces will be required to distinguish one operation from the other, since two cryptographic operations being performed at the same time.

The second dimension is that when attacker tries to apply DPA on this implementation (as explained in section 2), he gets two sub keys for which correlation values is higher than rest of the sub keys. This behavior is in line with the fact that all operations are being performed simultaneously by both cryptographic keys. The actual values of these correlations depend on architecture of underline hardware. The attacker will choose one sub key out of these two as the correct sub key and will utilize this sub key as known data for the recovery of next sub key. If the attacker chooses the wrong value for any sub key, the probability of choosing next wrong sub key would be much higher as value of correlation for wrong key would be higher than the correct one.

Figure 3 shows the path available to attacker at any point. For AES-128, this tree will have a branch length of 16. At any point, the attacker has two choices to make with probability

$$P(x) = \begin{cases} p \\ 1- \end{cases} \quad (3)$$

Where p = probability of choosing correct path at any given point and 1-p = probability of choosing wrong path

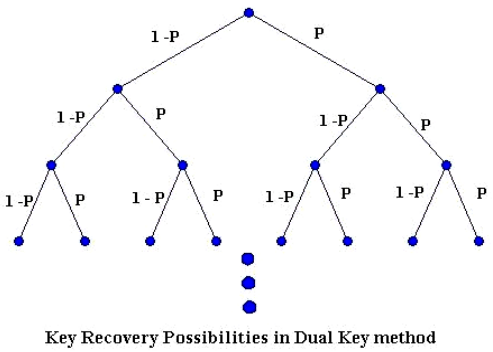

Fig. 3 Path choices available to attacker

The probability of choosing all the correct choices will be:

$$P(C) = p * p * ... * p = p^n \quad (4)$$

For 128 bit key length and 8 bit sub key, this equation will reduce to:

$$P(C) = p^{16} \quad (5)$$

For ideal scenario i.e. ideal hardware, measurement and perfect choice of hypothetical model, we can assume p = 0.5

$$P(C) = 0.5^{16} = 0.000015 \quad (6)$$

While probability of extracting wrong key would be:

$$P(W) = 1 - 0.000015 = 0.999985 \quad (7)$$

So the probability of correctly recovering the key is very small. We mention some important properties of this countermeasure:

This countermeasure can not be overcome by increasing the number of traces, as we are adding the noise which is related to implemented algorithm unlike other countermeasures which add unrelated random noise.

This countermeasure can be extended to add more resistance to cryptographic attack by increasing the number of keys to say 'k' number of keys, which are performing cryptographic operation simultaneously on the same plain/cipher text.Increasing the number of keys 'k' has two advantages; first the attacker requires more number of traces due to multiple cryptographic operations being performed at the same time. Second it adds more branches to the tree shown in Figure 3 which further reduces the probability of correct key recovery.

This countermeasure is also very easy to integrate in existing designs, since we are not modifying any cores internally but merely utilizing more instances of the core to perform multiple cryptographic operations simultaneously. So our solution for adding DPA resistance can be utilized without raising any IP integrity and licenses issues.

Please note that this countermeasure is different than operating multiple devices simultaneously as given in [23]. The reason in later case is, devices will be at different spatial position and will leak information through electromagnetic side channel and can be received by adjusting the position of receiver antenna. In our proposed method we have duplicated cores on the same chip, i.e. the distance is very small and same power source is utilized, so the above mentioned issue does not arise.

5.1 Relation with Success Rate of DPA Attack

In [20], a relation for success rate of DPA attack is derived.

$$S.R \approx P[C > W]^{(g-1)} \quad (8)$$

Where S.R = Success Rate, P = Probability, C = correlation coefficient of a correct key guess, W = correlation coefficient of a wrong-key guess and g = possible key guesses

It is clear from Eq.8 that the success rate of any DPA attack is dependent on possible guesses. In case of our dual key countermeasure, the number of possible key guesses doubles at each node of the tree which sadistically reduces the probability of success rate. With 'k' key model this relation will reduce to

$$S.R \approx P[C > W]^{(g-1)} \quad (9)$$

6. Implementation Detail

To test our Model, we have utilized the simulation method of testing against DPA resistance as employed in [21] and [22]. We implemented the AES core in Verilog for Xilinx Virtex-5 LX50T device. Once we have simulated our core in MODELSIM and verified its functionality, we have duplicated this core to add another cryptographic core which is performing cryptographic operations using a second key. This step was necessary since otherwise compiler drops some or all portion of 2nd core in optimization process, while we wanted to perform all operations in parallel. We use .ncdfilde generated with Xilinx ISE suit, along with 10,000 random inputs (128 bit each), as input to MODELSIM. It generates signal activity file (called .vcd file) after simulation.

This vcd file is then imported along with .ncd file in XILINX's XPOWER tool. This tool generates

power consumption for values in a text format in resolution of pico seconds. Once this tool has generated power consumption values, we arrange them in a matrix of size 10,000 x 256, where 10,000 represents the number of traces and there are 256 values in each trace. The data we have obtained at this point is simulation equivalent of actual power consumption values. By actual power consumption values, we mean the values we would have obtained by implementing encryption on FPGA and measuring the power consumption using oscilloscope as discussed in Section 2B. The only difference is that our values are more ideal and do not have effects of parasitic capacitance and noise which are present in real systems.

7. Performance Evaluation

To evaluate our countermeasure we have compared it with normal, unprotected implementation of AES-128. We only attacked the first sub key of 128 bit key for AES, as these first 8 bits are directly XORed with the plain text and represent the weakest spot in AES algorithm. Power consumptions values were generated as discussed in previous section. Theoretical model for this normal implementation was developed where we choose the hamming distance method for targeting the XOR that take place in first round of AES.

The next step was to calculate theoretical power consumption matrices for all possible values of sub key and then using the difference of mean method we calculated correlation values between these theoretical power matrices and actual power consumption matrices. Figure 4 shows the correlation graph for unprotected AES-128 implementation where correlation value is plotted against possible values of sub key. 3000 traces have been utilized for this DPA attack. We are able to detect correct sub key "74" whose correlation value is higher compared to other sub keys.

The next step was to increase the number of traces utilized in DPA attack and observe the behaviour of correlation graph with more number of traces. This DPA attack was repeated by utilizing 4000 traces. The resultant correlation graph is presented in Figure 5. As expected, by increasing number of traces the relative value of correlation for correct key has increased very much and we can clearly differentiate correct value of sub key compared to other values of sub key.

Increasing number of traces filter out the noise present in cryptographic circuit. Similar behaviour would be observed for the countermeasures which merely add noise to cryptographic circuits i.e. by increasing the number of traces, DPA countermeasures can be bypassed.

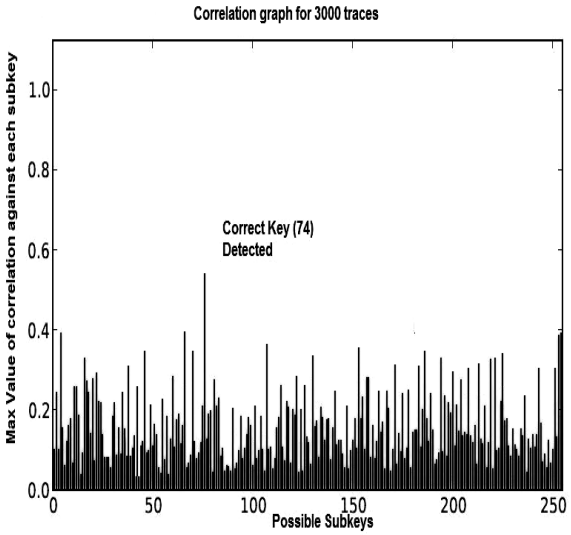

Fig.4 Correlation graph for unprotected AES (3000 traces)

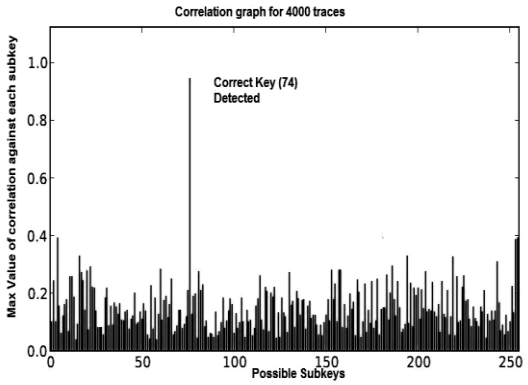

Fig. 5 Correlation graph for unprotected AES (4000)

The next step was to evaluate the performance of our dual key countermeasure and compare number of traces and waveforms with the unprotected AES-128 implementation. Same procedure was used to attack

dual key protected AES-128. 8000 traces are used for this DPA attack on AES implementation with dual key countermeasure. Figure 6 shows the correlation graph for DPA attack on dual key protected AES-128. As expected we have obtained two sub keys value for which correlation value is higher than other sub key's. The first one has value of "74" and other one with value "154". We could not differentiate these two keys for less than 7500 traces. This clearly shows that due to two encryptions being performed simultaneously, we need larger number of traces to differentiate these keys from others. One other point that is worth noting is the magnitude of correlation, which is reduced, compared to DPA attack on unprotected implementation of AES-128

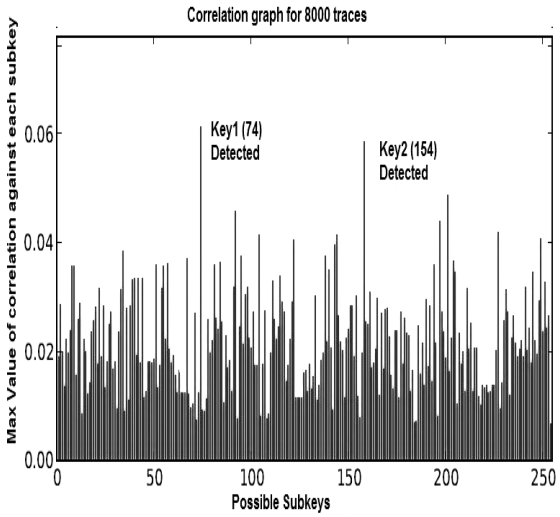

Fig. 6 Correlation graph for dual key protected AES (8000 traces)

In order to observe the effect of increase in number of traces on dual key countermeasure, we have repeated this DPA attack with 10,000 traces. Correlation graph utilizing 10,000 traces is presented in Figure 7. As expected, there is a very little increase in absolute values of correlation for different values of sub keys. In fact relative change between correlation values for true and false sub key is nearly zero, which shows that increasing number of traces cannot bypass dual key countermeasure.

In order to investigate the difference in relative correlation coefficient of Key1 and Key2, we repeated the experiment with different random input sets as described in Section 6. It turns out that relative values of correlation coefficient are random. However for actual DPA on physical hardware, this relative difference will depend on parasitic capacitance present between different elements. We have presented the outcome of one such experiment in Figure 8. Please note that we use 10000 traces for this experiment.

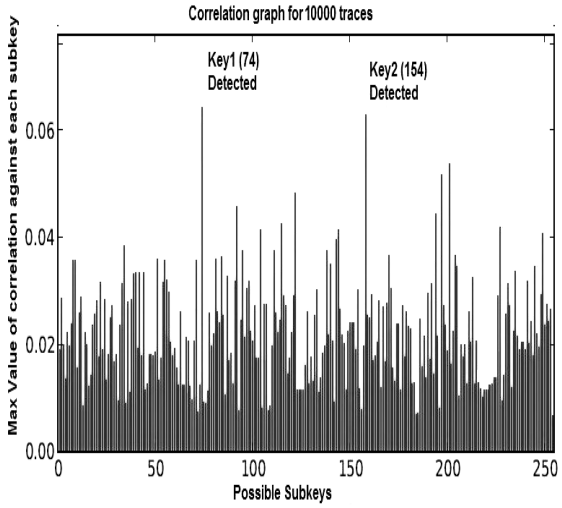

Fig. 7 Correlation graph for dual key protected AES (10000 traces)

There are certain trade-offs associated with this countermeasure. Energy consumption or power required for dual key countermeasure is double than normal (unsecure) implementation which implies shorter battery life. Similarly processing power or area required for implementing dual key countermeasure is double than normal (unsecure) implementation. As fabrication technology is shifting towards lower gate size, i.e. more logic is available on same chip size with reduced voltage level, the effect of increase in area and power overhead reduces. On the bright side, this countermeasure does not have any effect on system throughput i.e. number of the cryptographic operations per unit time remains the same.

8. Conclusion & Future Direction

Our proposed countermeasure has increased resistance against DPA attack. This countermeasure is unique in the sense that it cannot be overcome by increasing number of traces which is the problem with other hiding countermeasures. This countermeasure can be extended to use 'k' number of

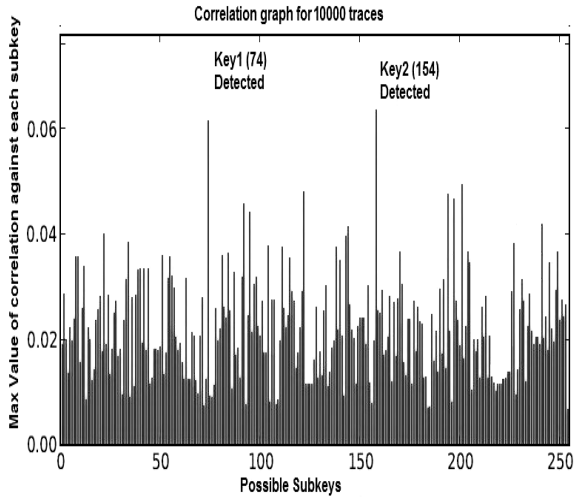

Fig. 8 Correlation graph for dual key protected AES with different random input set (10000 traces)

keys simultaneously increasing further resistance to DPA. This countermeasure can be integrating very easy to existing systems as IP modification is not required. Like any other countermeasure there is a tradeoff with power, area, throughput and DPA resistances. This countermeasure requires 2x Area and power while it has no degradation on throughput. In Future we will compare its effectiveness with other countermeasures and will introduce selective duplication at selective places in cryptographic algorithm

9. References

[1] N. I. of Standards, T. (NIST), Announcing the ADVANCED ENCRYPTION STANDARD (AES), Technical Report FIPS Publication 197, 2001.

[2] Rivest, R., Shamir A., Adleman L., A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM 21 (1978) 120–126.

[3] Koblitz, N., Elliptic curve cryptosystems, Mathematics of Computation 48 (1987) 203–209.

[4] Applied Cryptography: Protocols, Algorithms, and ource Code in C, John Wiley and Sons, 2nd edition, 1996.

[5] Introduction to Hardware Security and Trust, Springer, 2012.

[6] Kocher,P. C., Jae, J., Jun, B., Differential Power Analysis, in: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, pp. 388–397.

[7] P. Kocher, P. C., Timing Attacks on Implementations of Di E-Hellman RSA DSS and Other Systems, in: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, pp.104–113.

[8] Model Sim, ww.model.com last accessed October, 19th 2014

[9] Xilinx XPOWERwww.xilinx.com/products/design_tools/logic_design/verification/xpower.html last accessed October 19th 2014

[10] Digital Integrated Circuits: A Design Perspective, Prentice Hall, 1996.

[11] Power Analysis Attacks: Revealing the Secrets of Smart Cards, Springer, 2007.

[12] Tiri, K., Verbauwhede, I., A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation, in: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition(DATE), pp. 246–251.

[13] Bo, Y., Xiangyu, L., Cong, C., Yihe, S., Liji, W., Xiangmin, Z., An AES Chip with DPA resistance using hardware-based random order execution, Journal of Semiconductors 33 (2012) 065009.

[14] Medeiros, S. F., The Schedulability of AES as a Countermeasure against Side Channel Attacks, in: Security, Privacy, and Applied Cryptography Engineering, 2012, pp. 16–31.

[15] Zafar, Y., Har, D., A Novel Countermeasure to Resist Side Channel Attacks on FPGA Implementations, International Journal On Advances in Security 2 (2009).

[16] Liu, P.C., Chang, H.C., Lee, C.Y., A True Random-Based Differential Power Analysis

Countermeasure Circuit for an AES Engine, IEEE Transactions on Circuits and Systems II: Express Briefs 59 (2012) 103–107.

[17] Yang, S., Wolf, W., Vijaykrishnan, N., Serpanos, D. N., Xie, Y., Power Attack Resistant Cryptosystem Design: A Dynamic Voltage and Frequency Switching Approach, in: Proceedings of the conference on Design, Automation and Test in Europe, pp. 64–69.

[18] Bucci, M., Luzzi, R., Guglielmo, M., letti, A. T., A Countermeasure Against Differential Power Analysis Based on Random Delay Insertion, in: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 3547–3550.

[19] Ratanpal, G. B., Williams, R. D., Blalock, T. N., An On-Chip Signal Suppression Counter-measure to Power Analysis Attacks, IEEE Transactions on Dependable and Secure Computing 1 (2004) 179–189.

[20] Standaert, O.X., Peeters, E., Rouvroy, G., Quisquater J.J., An Overview of Power Analysis Attacks Against Field Programmable Gate Arrays, Proceedings Of the IEEE 94 (2006) 382–394.

[21] Amaar, A., Ashour, I., Shiple, M., Efficient Implementation of AES Algorithm Immune to DPA Attack, in: Proceedings of 14th international Conference on Modelling and Simulation, pp. 396–401.

[22] Strachacki, M., Szczepanski, S., Implementation of AES Algorithm Resistant to Differential Power Analysis, in: Proceedings of 15th IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 214–217.

[23] TEMPEST: A signal problem https:// www.nsa.gov/public_info/_files/cryptologic_spectrum/tempest.pdf last accessed October 5th 2014