Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 4, April 2016, pg.491 – 498

ANALYZING AND IMPROVING THE SECURITY OF CRYPTOGRAPHIC ALGORITHM AGAINST SIDE CHANNEL ATTACK

V. Sindhiya,
M.E, Department of Computer Science,
St.Joseph College of Engineering,
Sriperumbudur, India.
sindhiyapadma@gmail.com

Mr. M. Navaneetha Krishnan,
Research Scholar,
Manonmaniamsundarnar University,
Thirunelveli, India.
mnksjce@gmail.com

Dr.R.Ravi,
Prof & HOD, Department of IT,
St.Francis Xavier Engg College,
Thirunelveli, India.
directorresearch@francisxavier.ac.in

Abstract*—A side channel attack is a kind of attack based on side channel information obtained from the encryption device in addition to brute force attack. Common types of side channel attack are Timing attack, Power monitoring attack, Acoustic cryptanalysis attack, Differential fault analysis, Data remanence attack, row hamour attack. Timing attack utilizes timing in turn and supplementary input that are provided during customary implementation in encryption device. In Power Monitoring attack, it events the power consumption of the cryptographic device and are prevented by revealing the relationship between immediate power consumption and secret key information. Data remanence is the recuperation of deleted data that leftovers even after several attempts made to remove it. It makes involuntary revelation of sensitive information unrestricted into media. A new Rijndael algorithm, a AES algorithm is proposed to secure side channel attack against attacker module. Therefore in AES, the encrypted text will be sent into the several rounds of encryption with the key and the decryption of file will be carried with the same key. This new approach has been proposed to overcome the side channel attack using various implementations techniques.

***Keywords*—Side channel attack, Rijindael algorithm, AES algorithm, Data remanence**

I. INTRODUCTION

Embedded systems are most open to the elements to Side Channel Attacks. Side channel attack has been an valuable and those convenient attack on many decisive embedded systems that take up cryptographic algorithm for security. In Cryptographic algorithm, its security falls under two basis of classification; they are under mathematical protection and their application. Side Channel attack exploits the correspondence between physical leakage of a crypto system and its secret key to reclaim the key. Various measures are provided to embed system from side channel scrutiny. Random masking introduces random numbers into system to mask the secret key. It usually takes large implementation over transparency as it provides a copy of the original algorithm. These methods spread the side channel outflow from a single point onto more point to decrease the amount of leakage from the device. In addition to these, methods like random shuffling and delay are also used to mask the secret key. Permutation is applied on AES to stand firm the first

Order differential power analysis based on the fact that AES is a block cipher where 16 operation on in each round are independent.

Comparing with permutation, it is found that random sort index is easy to implement but it is highly weak. When compared to masking, shuffling does not require modifications of the algorithm. Manual implementation still requires a influence of specific algorithm and may not be fully broken by independence between operation in complex algorithm. Recent work designates a emerging trend towards automating the application of countermeasures against SCA to increase the security of the systems. The main involvement of the work lies in a agenda of source code conversion to randomize operations without any influence of the causal systems, which is generally applicable to other cryptographic algorithms also.

Involuntary instruction sensitivity quantification and local arbitrary recharging with substitute code segments take the edge off power leakage with automatic security assessment and substantiation. Regarding to assumption it is classified into two types namely profile and non-profile .A profiling phase is where the rival is provided with a training device under test that describe physical leakages and perform a secret key mining. Non-profiled Side Channel Attack requires the latter phase and assumes less accurate leakage model. Based on the key recovery procedure, Side Channel Attacks fall into two categories: divide-and-conquer that provides distinguishers for small key chunks that are then combined and analytical SCA that recover the entire key at once.

Analytical side channel attack is very active in crypto community. conventional Side Channel Attack exploits a divide and conquer strategy and recover several pieces of secret key independently .For ASCA, both the cipher and leakage are represented with algebric equations and the full furtive key is recovered at once by solving these equations with different methods. Since leakages are of many other rounds can be utilized this attack has less measurement difficulty than traditional SCA. The software implementation of AES, where collision-based SCA is shared with algebraic cryptanalysis is called algebraic side-channel collision analysis (ASCCA).This collision based ASCA differs signicantly from the usual Algebric Side Channel Attack which do not involve any collisions in it.

In ASCCA, the rival detects the internal collisions in two rounds by comparing the patterns of the power traces and then converting it to a equation to be solved. When there is excessively much noise and deduction sets are too large, their exist too many solution for the equation system. If both the plain text and cipher text are integrated in the equation set, the output solution should be correct but the solver may not output the solution in a logical amount of time. ASCCA only needs five power traces to pull through the master key of AES.. In ASCA, template attack is used to figure out the Hamming weight (HW) or the accurate value of transitional states. General equation solver might output a satisfied or optimized solution but not the correct one, which reduces the success rate.

The literal reduced key search space of AES for the given amount of leakages is not studied in earlier Algebric Side Channel attack. It is vital to find a new come within reach of that would lessen a manipulation of the solver and reduce the time density of existing analytical side channel attack on AES, especially, when considering the error tolerant circumstances. Our main idea is enthused by the simple power attack technique and the low data complexity attack technique. The work is utilized in incomplete diffusion feature in the AES key extension to recover the secret key of AES with a distinct power trace. The algebric techniques are used to connote both the cipher and their deductions.

Since there are more leakages in AES encryption course of action, attack might work under unknown plain text and cipher text. A fanatical attack may achieve a better performance than obtainable analytical side channel attack by using a considerable method called incomplete diffusions. This technique is called as incomplete diffusion analytical side channel analysis (IDASCA).computation of one byte does not rely on sixteen bytes of all rounds. The last AES round does not have Mix column operation. The master key can be received by analyzing all of the master keys.. In practice, it was practical that noise is the main issue for vigorous ASCA.. To restrain this issue, two types of solutions are provided. One solution is to cluster the deductions together into sets, then change them into algebraic equations. In this method, there are many variants, such as multiple deductions-based ASCA (MDASCA).

Instead of general equation solver to solve the round reduced AES, it is motivating to make use of the incomplete diffusion feature. It provides higher level of security when affected by various side channel attack and to get through the time convolution problem. As a cipher, AES has proven consistent. The only successful attacks against it have been side channel attacks on limitation found in the implementation or the key administration of assured AES based encryption products. It does not use brute force attack to break the cipher but develop flaws in the ways it has been implemented. Wide ranging experiment is conducted on AES using microcontroller.

IDASCA can help to improve the understanding of the different ASCA and make analytical side channel attacks more practical. Although there still remains a tradeoff between the toughness and in formativeness for IDASCA at the moment, further it may reduce these gaps. IDASCA enumerates and computes the satisfied candidates of four bytes. It can also be used to make the most of the probability of different deductions of the candidates by modifying the algorithm.

II. RELATED WORK

Pei Luo and Adam ding [1] Towards Secure cryptographic software implementation against side channel power analysis attacks. A tool is deliberate to detect independence among the statements at the source code level for automation operation shuffling to defend against power analysis side channel attack .Shuffling mitigates the susceptibility of cryptographic system against power analysis. transitional variables are produces when leakage occurs

Nicolas Veyrat-Charvillon, Benut Gerard, Francois-Xavier Standaert [2] Security evaluations beyond computing power. how to analyze side channel attack that you cannot mount used rank estimation algorithm. It provides tight bounds for the security level of leaking cryptographic devices. They are able to analyze the full complexity of standard channel attacks, in terms of their time, data, and memory complexity. This leads to uncomfortable situation where security is based on enumerable key size

X. J. Zhao et al [3] An enhanced Algebric Side channel attack for error tolerance and new leakage model exploitation. The Algebraic side-channel attack (ASCA) is a powerful cryptanalysis technique which is highly different from conventional side-channel attacks. It is studied in three aspects namely enhancement, analysis and application. To enhance ASCA, they propose a general method, called Multiple Deductions-based ASCA (MDASCA).It leads to a great threats with its distinction in error tolerance and new leakage model.

Charles Bouillaguet, Patrick Derbez, and Pierre-Alain Fouque [4] Automatic Search of attacks on round reduced AES and its applications. It make obvious the strength of the tools. It also describes the versatile and powerful algorithms for searching guess and determine and meet in the middle attacks. These tools can be used in the framework of fault tolerance and are vulnerable to various types of network attacks.

ItaiDinur and Adi Shamir [5] Side Channel Attacks on Block ciphers. It make official the opinion of the leakage on iterated block ciphers. They develop a new variant of cube which can tolerate noise. This describes the efficient leakage attacks on two of the best known block ciphers. Original cube attack requires exceedingly clean data, whereas the information provided by the side channel attacks are quite noisy.

S. Chari, J. Rao, and P. Rohatgi [6] A template attack is not amenable to methods such as SPA and DPA, can simply be devastated using a template attacks with a single sample. The realization of these attacks in such agreeable circumstances is due to the manner in which noise inside each sample is handled. It focuse on precisely modeling noise, and using this to fully extract information. SPA and DPA can easily be broken using template attacks with a single sample. Other application includes attacks on DES that are DPA resistance hardware.

III. PROPOSED METHOD

In the proposed system, we can recover the secret key of AES with a single power trace even when both the plaintext and cipher text are unknown, which is much more powerful and requires less power traces than Differential Power Analysis. Power analysis attack is a side channel attack which deals the power consumption of the cryptographic device during normal execution. A challenge is then made to bare the relationship between power consumption and secret key information. The problem which is present in the existing system is that, it does not prevent the data from user, which leads to the loss of data. Timing attacks slows down the system speed and also the path of the file which sends the data, so there will be delaying in the file transfer. The Rijndael algorithm is used to defend against various side channel attacks. On implementing this project, the users are registered in a secured network. Data remanence is the outstanding depiction of the digital data that vestiges even after attempts have been made to remove or erase data. It makes involuntary disclosure of sensitive files to be disclosed in an uncontrolled environment. Thus the enhanced AES is used efficiently to solve the problem of side channel attack during encryption process with higher level of security.

IV. PROPOSED ALGORITHM

Improved Advanced Techniques

A. Substitue Bytes Method

In the SubBytes step, each byte in the *state* matrix is altered with a SubByte using an 8-bit substitution box, the Rijndael S-box. This procedure results in on condition that the non-linearity in the cipher. The S-box used is subsequent from the multiplicative inverse over, known to have good non-linearity property. To avoid attacks based on simple algebraic properties, the S-box is constructed by integrating the inverse function with an invertible affine renovation. The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points. While performing the decryption, Inverse SubBytes step is used, which requires first taking the affine transformation and then finding the multiplicative inverse (just reversing the steps used in SubBytes step).

(Is-box[s-box(a)]= a) though it should not be its self inverse

i.e. s-box(a) 6= Is-box(a)

B. Shift Row Transformation Method

The ShiftRows step operates on the rows of the state; it regularly shifts the bytes in each row by a firm balance. In AES, the first row is left unaffected. Each byte of the second row is replaced one to the left. Similarly, the third and fourth rows are shifted by balancing the two and three respectively. For 128 bits and 192 bits, the changing pattern is the same. Row n is shifted left spherical by n-1 bytes. In this way, each column of the output state of the ShiftRows step is hovering of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). For a 256-bit block, the first row is unbothered and the changing for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively this change only be valid for the Rijndael cipher when worn with a 256-bit block, as AES do not use 256-bit blocks. The importance of this step is to steer clear of the columns being linearly autonomous, in which case, AES degenerates into four independent block ciphers.The fourth row of the byte is shifted 3 bytes to the left in a circular style.

C. Mix Column Transformation

This stage (known as MixColumn) is primarily a substitution but it sorts the use of arithmetic of GF(28). Each column is activated on separately. Every byte of a column is plotted into a new value that is a function of all four bytes in the column. The revolution can be determined by the consequent matrix multiplication on state.

$$\begin{bmatrix} k0 \\ k1 \\ k2 \\ k3 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 & 1 \\ 2 & 3 & 3 & 1 \\ 4 & 2 & 1 & 1 \\ 1 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} m0 \\ m1 \\ m2 \\ m3 \end{bmatrix}$$

Each element of the product matrix is the addition of products of elements of one row and one column. In this case the separate additions and multiplications are done in GF(28). The MixColumns transformation of a single column is thus expressed as j ($0 \leq j \leq 3$) of state can be expressed as:

k0=5m0+2m1+1m2+1m3
k1=2m0+3m1+3m2+1m3
k2=4m0+2m1+1m2+1m3
k3=1m0+1m1+2m2+3m3

The InvMixColumns is defined by the following matrix multiplication:

$$\begin{bmatrix} l0 \\ l1 \\ l2 \\ l3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} n0 \\ n1 \\ n2 \\ n3 \end{bmatrix}$$

This can be expressed as follows.,

l0=14n0+11n1+13n2+9n3
l1=9n0+14n1+11n2+13n3
l3=13n0+9n1+14n2+11n3
l4=11n0+13n1+9n2+14n3

If the label of these A and A−1 individually and the label state the mix columns operation as S and consequently as S’,
This is defined as:

AS = S’

V. SYSTEM DESIGN

System Design involves identification of classes their relationship as well as their collaboration.

In Figure [1] the user first login using own username and passwords and one has to register by entering all details and then login. Once login is completed uploading and downloading can be done using AES encryption and decryption process. The key is generated which encrypts along with the file and transfers it to the user. The admin accepts and sends a mail that is used to decrypt the file. This hence makes AES encryption more secure from the side channel attack.

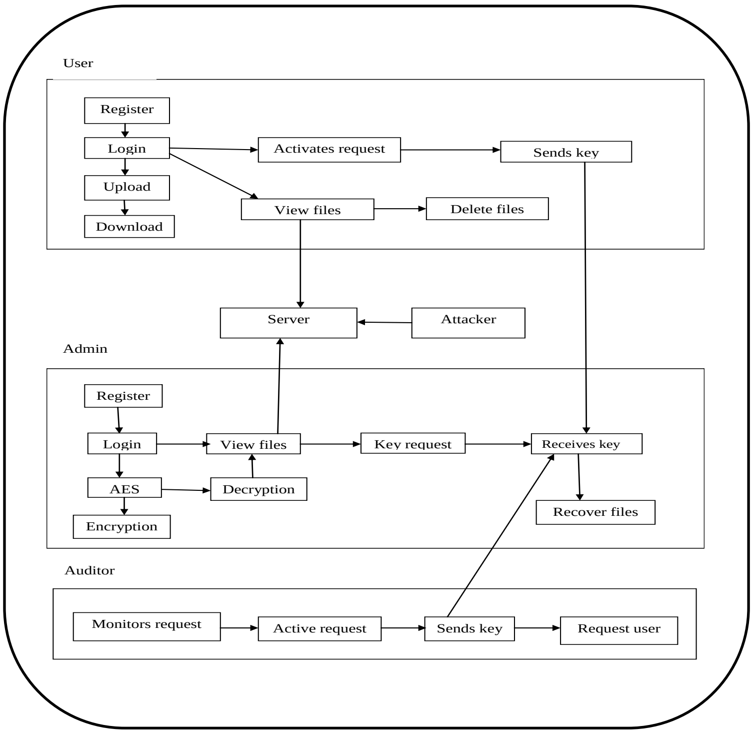


Fig. 1. System Architecture.

A. *Description of AES*

The Advanced Encryption Standard (AES) is a measurement for the encryption of electronic data time-honored by the U.S. National Institute of Standards and Technology (NIST) in 2001. Rijndael is a family of ciphers with different key and block sizes. For AES, NIST preferred three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256 bits.

Rijndael was designed to have the following characteristics:
• Confrontation against all known attacks.
• Speed and code neatness on a wide range of platforms.
• Design cleanness.

B. Implementation of AES

KeyExpansions—round keys are derived from the cipher key using Rijndael's key program. AES requires a separate 128-bit round key block for each round plus one more.

InitialRound

AddRoundKey— each byte of the state is united with a block of the round key using bitwise xor.

SubBytes— a non-linear substitution step where each byte is replaced with another according to a lookup table

ShiftRows—The ShiftRows step functions on the rows of the state; it at regular intervals shifts that required bytes in every row by a definite offset value. In AES, the first row ruins unaltered. Every byte of the second row is shifted one to the left.

MixColumns—In the MixColumns step, In tis process the four bytes of every column are mutually an invertible linear conversion. The MixColumns function comprises four bytes as input and outputs four bytes, where each and every input byte affects all four output bytes. Both techniques of ShiftRows, MixColumns provides diffusion in the cipher.

- AddRoundKey
- Final Round (no MixColumns)
- SubBytes
- ShiftRows
- AddRoundKey.

C. Divide the States and Leaks in Each AES Round

State group mapping. The calculation procedure fretful in state group mapping comprises of one state group in βi from the state group in αi as the state group mapping, which can be uttered as

$$i \rightarrow yi$$

Leak group. According to the software understanding of AES, there are 21 leaks analyze in one state group mapping, which can form a leak group and can be denoted as

$$Nxi \rightarrow yi$$

D.Conquer the States and Leaks in Each AES Round

Let E(x) denote the entropy of the state P. Let x denote a state byte, D(x) denote the deduction set on the value of L(x), x denote one possible candidate of L(x), and I s(x, D(x)) denote the function of D(x).

If the informations produced are not leaked, then the attack complexity of the founded enumerations and improved method of enumerations becomes identical. But in general, there always exist some amount of leakage on the intermediate states and attack complexity. And as a result, the improved numerations are much lower.

E.Search for the Master Key

In order to the AES key schedule, recovering any round key is equal to the recovery of the master key. The candidates of the master key by the technique of brute-force search of Ri for each round and then the method of intersection is used among the candidates in multiple rounds to compute the final search of the master key. The complication while computing the intersection varies with the candidate size in different rounds, which is inexpensive with small size and concentrated with large.

VI. RESULTS

A.Timing attack

It is based on measuring the time it takes to perform operations. This information is about the secret keys. If a Unit is vulnerable, the attack is computational simple and often requires only cipher text. Reasons includes unnecessary operations, branching, conditional statements etc. timing measurement are fed into arithmetic model that can provide the guessed key bit with some degree of certainty. The equalization can be caused by always performing both operations regardless of the operation that is required at any given time. At any stage where one of the operations that is required to run, both should be executed and consequences of the avoidable operation is to be silently ignored

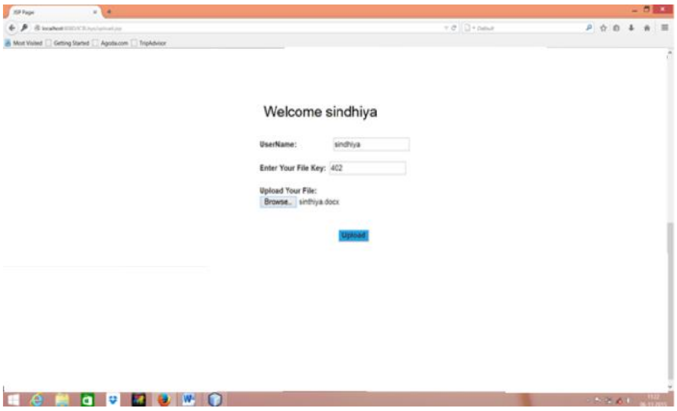
B. Power Monitoring Attack

Power analysis is a form of side channel attack in which the attacker studies the power consumption of a cryptographic hardware device. The attack can non-invasively extort cryptographic keys and other secret information from the devices. SPA involves the interpreting power traces or graphs over time. DPA is more complex form of power analysis which can allow an attacker to compute the intermediate values from multiple cryptographic operations. Dummy registers and gates are added on which useless operations are made to balance power consumption into a constant value. Whenever an operation is performed in hardware, a harmonizing operation should be performed on a dummy element to assure that the total power consumption of the unit remains balanced according to some higher value. Such techniques, by which the power consumption is constant and independent on input and key bits, prevents all sorts of power consumption attacks such as SPA and DPA.

C. Data Remanence Attack

Data remanence is the outstanding depiction of the digital data that remains even after attempts made to remove it. This residue may result from data being left intact by nominal file deletion by reformatting storage media that does not remove data previously written to the media or through the physical properties of the storage media that allow previously written data to be recovered. It may take inadvertent disclosure of sensitive information possible should the storage media be released into an uncontrolled environment. Various techniques have been developed to countermeasure data remanence. These are classified into clearing, purging and destruction.

These specific methods include overwriting, media destruction. Effective application of countermeasures can be complicated by several factors including media that are in accessible media that cannot effectively by erased, advanced storage systems that maintain histories of data throughout the data’s lifecycle and persistence of data in memory that is typically volatile.



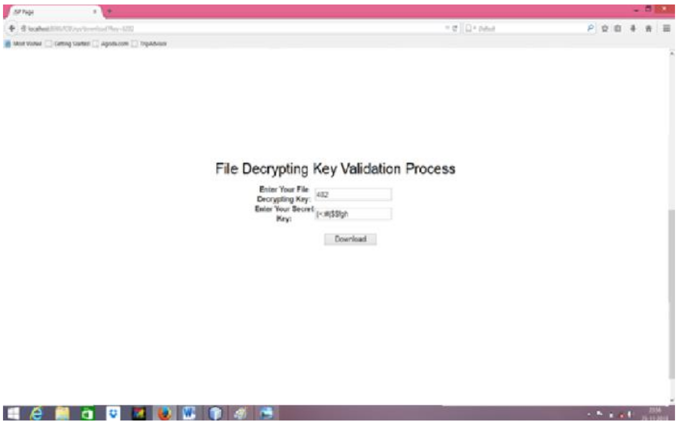
Welcome sindhiya

UserName: sindhiya

Enter Your File Key: 402

Upload Your File: sinthiya.docx

Fig 8. Encryption

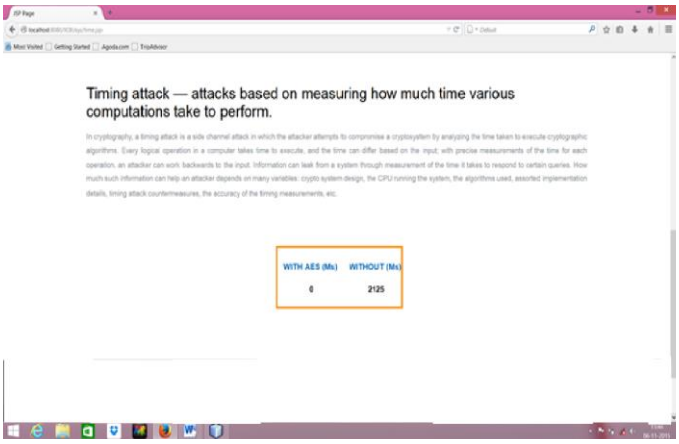


File Decrypting Key Validation Process

Enter Your File Decrypting Key: 402

Enter Your Secret Key: |<.#@$$fgh

Fig 9. Decryption

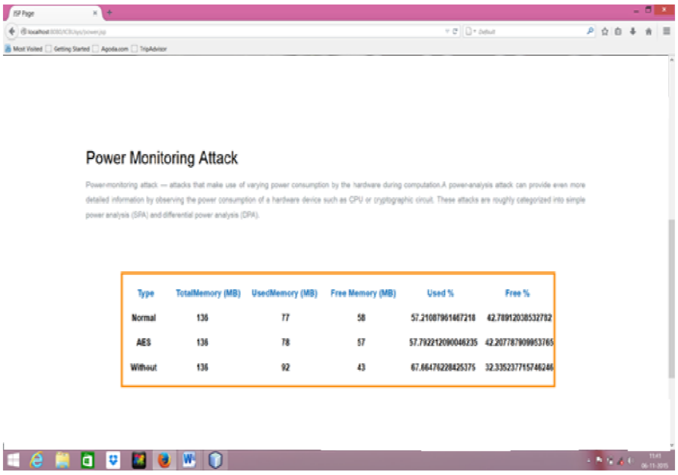


Timing attack — attacks based on measuring how much time various computations take to perform.

In cryptography, a timing attack is a side channel attack in which the attacker attempts to compromise a cryptosystem by analyzing the time taken to execute cryptographic algorithms. Every logical operation in a computer takes time to execute, and the time can differ based on the input; with precise measurements of the time for each operation, an attacker can work backwards to the input. Information can leak from a system through measurement of the time it takes to respond to certain queries. How much such information can help an attacker depends on many variables: crypto system design, the CPU running the system, the algorithms used, assorted implementation details, timing attack countermeasures, the accuracy of the timing measurements, etc.

WITH AES (Ms)	WITHOUT (Ms)
0	2125

Fig 10. Timing Attack



Power Monitoring Attack

Power-monitoring attack — attacks that make use of varying power consumption by the hardware during computation.A power-analysis attack can provide even more detailed information by observing the power consumption of a hardware device such as CPU or cryptographic circuit. These attacks are roughly categorized into simple power analysis (SPA) and differential power analysis (DPA).

Type	TotalMemory (MB)	UsedMemory (MB)	Free Memory (MB)	Used %	Free %
Normal	136	77	58	57.21087961467218	42.78912038532782
AES	136	78	57	57.792212090046235	42.207787909953765
Without	136	92	43	67.66476228425375	32.335237715746246

Fig 11. Power Monitoring Attack

VII. Conclusion

The proposed method constructs the success of the algorithm and the prevention methods are also applied to defend the side channel attacks which depends user data and the network channel. Through of various prevention techniques against the side channel attack, the data is transferred securely and the attackers are unable to retrieve the data. This increases the level of security in AES by preventing it from side channel attacks and uses AES efficiently and provides the secured system of data flow of users data.

References

[1] Pei Luo and Adam ding," Towards Secure cryptographic software implementation against side channel power analysis attack" Toronto, ON, Canada July 27, 2015 to July 29, 2015pp: 144-148.

[2] Nicolas Veyrat-Charvillon, Beno t Gerard, Francois-Xavier Standaert, "Security Evaluations Beyond Computing PowerHow to Analyze Side-Channel Attacks you Cannot Mount?", UCL Crypto Group, Universit_e catholique de Louvain.Place du Levant 3, B-1348, Louvain-la-Neuve, Belgium.

[3] Xinjie Zhao, Fan Zhang, Shize Guo,Tao Wang, Zhijie Shi, Huiying Liu1, and Keke Ji1,"MDASCA: An Enhanced Algebraic Side-Channel Attack for Error Tolerance and New Leakage Model Exploitation", Ordnance Engineering College, Shijiazhuang, Hebei, University of Connecticut, Storrs, Connecticut, USA The Institute of North Electronic Equipment, Beijing, China.

[4] Charles Bouillaguet, Patrick Derbez, and Pierre-Alain Fouque," Automatic Search of Attacks on round-reduced AES and Applications",ENS, CNRS, INRIA, rue d'Ulm, 75005 Paris, France.

[5] Itai Dinur and Adi Shamir,"Side Channel Cube Attacks on Block Ciphers",Computer Science department The Weizmann Institute Rehobot 76100, Israel.

[6] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi ."Template Attacks",IBM Watson Research Center,P.O. Box 704 Yorktown Heights, NY 10598.

[7] A. G. Bayrak, F. Regazzoni, P. Brisk, F. X. Standaert, and P. Ienne, "A first step towards automatic application of power analysis countermeasures," in Design Automation Conf., 2011, pp. 230–235.

[8] G. Agosta, A. Barenghi, and G. Pelosi, "A code morphing methodology to automate power analysis countermeasures," in Design Automation Conf., 2012, pp. 77–82.

[9] A. Moss, E. Oswald, D. Page, and M. Tunstall, "Compiler assisted masking," in Cryptographic Hardware and Embedded Systems, 2012, pp. 58–75.

[10] A. Bayrak, F. Regazzoni, D. Novo, and P. Ienne, "Sleuth: Automated verification of software power analysis countermeasures," in Cryptographic Hardware and Embedded Systems, 2013, pp. 293–310.