International Journal on Recent Researches In Science, Engineering & Technology
(Division of Computer Science and Engineering)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy.
It is an absolutely free (No processing charges, No publishing charges etc) Journal Indexed in JIR, DIIF and SJIF.
Research Paper
Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105

Volume 4, Issue 9, September 2016.

JIR IF : 2.54
DIIF IF : 1.46
SJIF IF : 1.329

A STUDY ON IMPROVING THE SECURITY AND PERFORMANCE OF AES ALGORITHM FROM SIDE CHANNEL ATTACKS USING CLOUD

M. NAVANEETHA KRISHNAN [1] , DR.R.RAVI [2]
[1]Research Scholar, Department of Computer Science and Engg, Manonmaniam Sundarnar University, Thirunelveli, India
mnksjce@gmail.com

[2]Professor & HOD, Department of Information Technology, Francis Xavier Engg College, Thirunelveli, India
directorresearch@francisxavier.ac.in

Abstract — Cloud services which are fast developing has its own vulnerabilities in the recent past. This paper attempts to set up a secret cloud atmosphere, making obvious a cache based side channel attack and travel around solutions to counter offense the same. A Cloud Computing location to host the attack and preventing the same is set up using an release source software called OpenStack. Based on side channel in sequence obtained from the encryption device in addition to the attack by any brute force is termed as side channel attack. The various types of side channel attacks are, acoustic cryptanalysis attack, timing attack, differential fault analysis, power monitoring attack, data remanence attack and row humor attack. Timing attack make use of timing in revolve and extra input that are made available throughout the usual performance in encryption device. Power monitoring attack measures the control utilization of the cryptographic machine and check the association between instant power spending and secret key information. Repossession of erased data that are available even after more than a few effort are made to take away those is called Data remanence. It creates instinctive revealtion of response in order unhampered into media. A new AES algorithm proposed to prevent side channel attack aligned with attacker unit is Rijndael algorithm. In AES, the text which has been encrypted will be subjected to few rounds of encryption with the key and the same key approves the decryption of folder. An advanced higher performance method has been proposed to overcome the side channel attack.

Keywords — Side channel attack, AES algorithm, Cloud Computing, Rijndael algorithm, openstack

1. Introduction

The Security of cryptographic algorithms can be viewed in two aspects. From the point of arithmetic security, differential, linear, algebraic cryptanalysis and their variants. And their implementations can be considered in the point of view of physical side channel-analysis techniques where physical leakages from the target devices, such as execution time power consumption and electromagnetic emissions are exploited to break the algorithms. Most vulnerable SCAs are embedded systems as attackers often have direct access to the files. Typical SCA techniques include simple power analysis, differential power analysis, correlation power analysis, mutual information analysis, template analysis, stochastic SCA, side-channel cube analysis, algebraic side-channel collision analysis and algebraic SCA. All these attacks use some type of models to compare physical leakages

with actual measurements. Two important classes of SCAs derived from the assumptions are: profiled and non-profiled.

Adversary provided with a training device to characterize physical leakages constitutes profiling phase (in order to obtain a precise leakage model); and in online exploitation phase the attack is mounted against a similar target device under test to perform a secret key extraction. Non-profiled SCA makes use of only the latter phase and consumes less precise leakage model, typically obtained from Engineering perception. As far as key recovery procedure is concerned, SCAs are divided into two categories: divide – and – conquer SCA (which provide distinguishers for small key chunks that are then combined, e.g., using key enumeration) and analytical SCA (which recover the entire key at once, e.g., by solving systems of equations).

The SCAs can be divided into the four types. Analytical SCA is currently a very active area in the crypto community. Traditional SCAs exploit a divide-and-conquer strategy and recover several pieces of a secret key independently. For analytical SCA, both the cipher and the leakages are represented with algebraic equations and the full secret key is recovered at once by solving these equations with different strategies. Since leakages of more rounds can be utilized, this attack has less measurement complexity than traditional SCAs. To attack the software implementation of AES on 8-bit microcontroller, collision-based SCA is combined with algebraic cryptanalysis. The attack is named algebraic side-channel collision analysis .

In the adversary detects the internal collisions (if the values of two intermediate states are equal) in two AES rounds by comparing the patterns of two sections of the power traces and then converts them into equations which can be solved using F4 Gröbner based algorithm Under known plaintext scenario, only requires five power traces to recover the master key of AES. This attack is independent of the leakage model. In ASCA, template attack is used to deduce the Hamming weight (HW) or the accurate value of intermediate states.

This can be done by detecting the external collisions between the targeted power trace in TADUT with the template power trace in TRDUT. The algebraic technique is used to represent both the cipher and the deductions. ZChaff, a type of SAT solver is used to recover the secret key. Compared to other SCA techniques, ASCA which makes use of the side- channel leakages in all cipher rounds and in a singlr trace can recover the key even when exists unknown plaintexts and ciphertexts. The Hamming weight leakage model (HWLM) goes highly compatible with ASCA. Recently, it has also been successfully applied to the hardware implementation of AES on a 65nm ASIC, under the template leakage model (TLM) with a single power trace.

This work aims to study the impact of representation dependence, leakage dependence and cipher dependence on ASCA. Also the resistance of algebraic immunity to block ciphers against ASCA is studied. The original ASCAs assume that the correct deduction on the Hamming weight (HW), or the accurate value of intermediate states, can be profiled from analyzing the side-channel leakages. In practice, robustness of ASCA is caused by noise. To overcome this effect, it is necessary to obtain multiple deductions from the leakage and need to be utilized in the attack. To get rid of this issue, two types of solutions are provided. One solution is to form sets – a group of deductions, then converting them into algebraic equations. The other solution include the imprecise deductions in the equation sets and to deal with these imprecisions via an optimizer (e.g., the SCIP solver).

This technique is denoted as Tolerant Algebraic Side Channel Attack in CHES 2010 and in Eprint 2012/092 . In CHES 2012, Tolerant Algebraic Side Channel Attack is modified to cope with the different probabilities of multiple deductions named probabilistic Tolerant Algebraic Side Channel Attack. Probabilistic Tolerant Algebraic Side Channel Attack has the ability to regain some information lost in other attacks. In general, existing ASCAs adopt off-the-shelf equation solver (e.g., the F4 Gröbner based algorithms in MAGMA solver , SAT solver, mixed integer programming solver). The unique feature of the general solver approach is its application to different cryptographic algorithms. The main problem is failing to take into account the specialized structures or properties of the specific cryptographic algorithms. The results differ depending on the type of solver used, and the time complexity.

When there exists heavy noise and large deduction sets, too many solutions for the equation system exists. Equation set, the general equation solver gives a satisfied or optimized solution but fails to give the correct one, which decreases its success rate. If both the plaintext and ciphertext are present in the equation set, the output solution should be correct but the solver does not give the exact solution in a considerable amount of time. Considering the error tolerant attack scenario, it is significant to find a new technique to minimize the impact of the solver and reduce the time complexity of existing ASCAs on AES. The main idea is inspired by the simple power attack technique in ICISC 02 and the low data complexity attack technique in CRYPTO 11 . The work is utilized for the incomplete diffusion feature in the AES key expansion to recover the secret key of AES with a single power trace. It is also utilized as customized solver approach instead of the general equation solver to solve the equations of Round-Reduced AES. It is interesting to develop the incomplete diffusion feature in the AES encryption procedure and utilize a specialized approach (construct a customized or specialized solver) instead of the general equation solver to improve ASCA. More leakages in the AES encryption procedure, makes the way for the attack to take place under unknown plain text and cipher text scenario. Specialized attack has higher performance the existing ASCAs when incomplete diffusion feature is considered. We name our technique incomplete diffusion of analytical side-channel analysis .

2. Related Work

Nicolas et al[1] have developed a rank estimation algorithm which is used to recognize the key sizes for standaraized symmetric cryptography. This leads to an uncomfortable situation, where the security of an implementation can be anywhere between enumerable values of the full key size.

Charles et al[2] have demonstrated the strengh of the tools, to automatically recognize the new attacks on round-reduced AES through very low data complexity, and thus to find the enhanced attacks on the AES.

Chari et al[3] has demonstrated that the execution of AES, is not adaptable to methods such as SPA and DPA, and can simply be shattered using template attacks with a single sample. Achieving these attacks in such feasible circumstances is due to the fact of handling the noise inside each sample.

ItaiDinur and Adi Shamir[4] developed the concept of leakage attacks on iterated block ciphers, in which the attacker can discover physical searching, power measurement, or any other type of side channel. The unique cube attack needs particularly clean data, however the information provided by side channel attacks is noisy.

Zhao et al[5] have investigated that the Algebraic side-channel attack is a powerful cryptanalysis technique which is quite different from conventional side-channel attacks. To enhance ASCA, they have proposed a Multiple Deductions-based ASCA to cope up with the multiple deductions produced by erroneous measurements or intrusions.

Shaheb et al[6] have discovered that upon removing the quadratic equations, iterated chosen plaintexts, and cube iteration to progress the SCCA on block ciphers can easily be identified. The Side Channel Cube Attack (SCCA) is a sympathetic of Algebraic Side Channel Attack (ASCA) consisting of theoretical and practical phases.

3. Proposed Algorithm

Here the proposed AES is Rijndael algorithm. First, the encryption features are analysed with the advanced AES algorithm. Typical SCA techniques include simple power analysis (SPA), differential power analysis (DPA), correlation power analysis (CPA) , mutual information analysis (MIA), template analysis (TA), stochastic SCA (SSCA), side-channel cube analysis (SCCA), algebraic side-channel collision analysis and algebraic SCA. All these attacks make use of some models where physical leakages are compared with actual measurements.

Improved Advanced Techniques

A. Substitue Bytes Method

This stage (known as SubBytes) is basically a table lookup using a (16×16) matrix of byte values named as s-box. This matrix consists of potential amalgamations of an 8 bit sequence ($28 = 16 \times 16 = 256$). The s-box is not the only random permutation of these standards, it is a well distinct technique for generating the s-box tables. It is not a matter of concern how the s-boxes are made up, we can

simply take them as table lookups. The matrix that gets operated in the process of encryption is known as state. More apprehension is focused on how this matrix is made workable in each round. In this particular round each byte is mapped to a novel byte in the subsequent way: the left most nibble requires a precise row and the right nibble requires a column of the S box. This is formerly used to update the state matrix. The Inverse is used to substitute byte transformation (known as InvSubBytes) which makes usage of an inverse s-box. The s-box considered to be strong is known for cryptanalytic attacks. Precisely, the Rijndael designers hunted a scheme that has a low connection among input bits and output bits, and the assets that the output cannot be defined as a humble mathematical function of the input. In addition to this, the s-box takes no fixed points (s-box(a) = a) and no conflicting fixed points (s-box(a) =−a) where −a is the bitwise compliment of a. The s-box is essential to be an invertible and if decryption is to be made possible (Is-box[s-box(a)]= a) it should not be its self inverse i.e.
s-box(a) 6= Is-box(a)

B. Shift Row Transformation Method

This stage (known as ShiftRows) is a simple permutation and nothing else. The working hierarchy of this method is:

- The first row of the byte in the state is not altered.
- The second row of the byte is shifted 1 bytes to the left in a circular style.
- The third row of the byte is shifted 2 bytes to the left in a circular style.
- The fourth row of the byte is shifted 3 bytes to the left in a circular style.

The Inverse of the Shift Row transformation (known as InvShiftRows) performs these circular shifts in an opposite manner for the last three rows (the first row which is unaltered to begin with). Thus this process may not seem to do much but if you think nearly how the bytes are well-arranged inside the state then it can be realised to have an impact. The state is preserved as an array of four byte columns, such that the first column actually represents bytes 1, 2, 3 and 4. A one byte shift is hence a linear distance of four bytes. The transformation also ensures that the four bytes of one column are ranged out to four different columns.

C. Mix Column Transformation

This stage (known as MixColumn) is fundamentally a substitution but it sortsthe use of arithmetic of GF(28). Each column is activated on separately. Every byte of a column is plotted to a new value which is a function of all the four bytes in the column. The transformation can be determined by the subsequent matrix multiplication on state.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} t_{0,0} & t_{0,1} & t_{0,2} & t_{0,3} \\ t_{1,0} & t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,0} & t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,0} & t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix} = \begin{bmatrix} t'_{0,0} & t'_{0,1} & t'_{0,2} & t'_{0,3} \\ t'_{1,0} & t'_{1,1} & t'_{1,2} & t'_{1,3} \\ t'_{2,0} & t'_{2,1} & t'_{2,2} & t'_{2,3} \\ t'_{3,0} & t'_{3,1} & t'_{3,2} & t'_{3,3} \end{bmatrix}$$

Each component of the product matrix is the sum of product 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 ($0 \le j \le 3$) of state can be expressed as:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

where • signifies multiplication over the finite field GF(28).The InvMixColumns is defined by the following matrix multiplication

$$\begin{bmatrix} 0C & 0F & 0G & 09 \\ 09 & 0C & 0F & 0G \\ 0G & 09 & 0C & 0F \\ 0F & 0G & 09 & 0C \end{bmatrix} \begin{bmatrix} t_{0,0} & t_{0,1} & t_{0,2} & t_{0,3} \\ t_{1,0} & t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,0} & t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,0} & t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix} = \begin{bmatrix} t'_{0,0} & t'_{0,1} & t'_{0,2} & t'_{0,3} \\ t'_{1,0} & t'_{1,1} & t'_{1,2} & t'_{1,3} \\ t'_{2,0} & t'_{2,1} & t'_{2,2} & t'_{2,3} \\ t'_{3,0} & t'_{3,1} & t'_{3,2} & t'_{3,3} \end{bmatrix}$$

If the label of these A and A−1 individually and the label state themix columns operation as S and subsequently as S',

AS = S'

Therefore A−1S'= A−1AS = S

System Design

Typical SCA techniques include simple power analysis (SPA), differential power analysis (DPA), correlation power analysis (CPA) , mutual informationanalysis (MIA), template analysis (TA), stochastic SCA (SSCA), side-channel cube analysis (SCCA),algebraic side-channel collision analysis (ASCCA) and algebraic SCA (ASCA).All these attacks exploit some model of the physical leakages to be compared with actual measurements.

In Figure [1]The sender sends the file which is prevented from all the side channel attacks. The key is generated by the sender which is encrypted along with the file and transferred to the receiver. The admin thus accepts and sends a mail which is used to decrypt the file. This makes the AES encryption more secure from the side channel attacks.

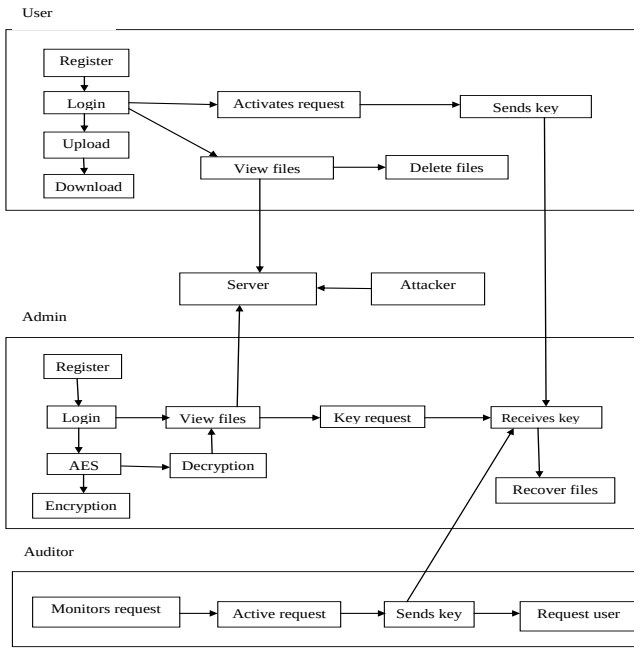

Fig. 1. System Architecture

A.Description of AES

AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. However, AES is quite different from DES in various ways. The Rijndael algorithm allows a variety of block and key sizes and not just 64 bit and 56 bit of DES block and key size. The block and key can infact be chosen independently from 128, 160, 192, 224, 256 bits and need not be the same. However, the AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys - 128, 192, 256 bits. Depending on the version used, the name of the standard is modified to AES-128, AES-192 or AES- 256 respectively. The main difference between AES and DES is such that, DES is not a feistel structure. In a feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. In this case the entire data block is processed in parallel during each round using substitutions and permutations.

Rijndael was designed to have the following characteristics:

• Resistance against all known attacks.

• Speed and code compactness on a wide range of platforms.

• Design Simplicity.

B.Implementation of AES

KeyExpansions—round keys are obtained from the cipher key by means ofRijndael's key schedule. AES needs a distinct 128-bit key for each round plus one more.

InitialRound

AddRoundKey—In the AddRoundKey step, the subkey is joined through the state. On behalf of each round, a subkey is found after the main key byRijndael's key schedule; everysubkey is of the identical size as per the state.

SubBytes—In the SubBytes step, every byte a_{i,j}which is in the state matrix is swappedby a SubByte S(a_{i,j}) by means of an 8-bit substitution box, the Rijndael S-box.Thisprocesshelps in finding the non-linearity in the cipher.

ShiftRows—TheShiftRows step functions on the rows of the state; it cyclically shifts the required bytes in every row by a definite offset value. In AES, the first row remainsunaffected. Every byte of the second row is shifted one place to the left.

MixColumns—In the MixColumns step,four bytes of every column are mutual by an invertible linear transformation. The MixColumns function comprises of four bytes as input and four bytes as output, 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. Performance 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—TheShiftRows 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 this 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.

D.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 ofAES, there are 21 leaks analyze in one state group mapping, which can form a leak group and can be denoted as

$$Nxi \rightarrow yi$$

E.Conquer the States and Leaks in Each AES Round

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

If the information 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.

F.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.

Recovering any round key is equal to the recovery of the master key. The candidates of the master key by the technique of brute-forcesearch 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 complexity while computing the intersection varies with the candidate size in different rounds, which is affordable with small size and intensive with large.

4. Discussions

The main objective of the project is to prevent the file being attacked by the attacker ,by which we prevent the various side channel attacks to the user. Thus by using the AES algorithm we prevent various attacks and share the data from sender to the receiver using proper requirement methods. Thus the AES prove to be the best algorithm to prevent the Side channel attacks from being attacks.

This proposes a new technique named Incomplete Diffusion Analytical Side-channel Analysis (IDASCA). The main analysis of the incomplete diffusion in one AES round is described as the core of IDASCA on AES, which is mainly composed of three steps: states and leaks in each AES round, state from leaks in each AES round and search for the master key of AES.

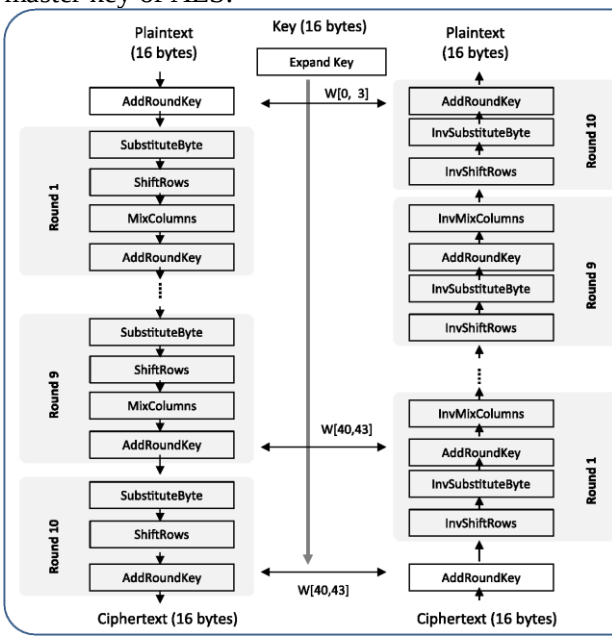

Fig. 2. AES Architecture

5. Results

A.Attack Without Errors

IDASCA can work well in both known plaintext and unknown plaintext/ciphertext scenarios.The output of CR and the round output in the i -th round can be recovered by analyzing the 84 leaks (excluding the 16 leaks in loading the round key Ki) in that round. This is been reduced to a greater extend to analyze the i+1-th round, since the final output of the i -th round, also an input of CR in the i+1-th round, is unknown, it is not possible to extract the round-key from any single round between Round 2 and 9.

B. Attack With Errors of Fixed Deduction Size

In this, the two different values of the deduction size μ are considered, the same as the prob-TASCA in CHES 2012. The first is μ = 2, where they conduct 100 attacks in the first 9 rounds of AES and the average time obtained is 1 second. The results explain that ρd = 80% and ρk = 80% which means that the master key of AES can be recovered by attacking the first round under known plaintext scenario, or unknown plaintext/ciphertext scenario.When $\mu \geq 4$, under unknown plaintext/ciphertext scenario, the effort to handle and reduce E(Ri+1) is unaffordable (296) and IDASCA is prevented using a single power trace of one plaintext. If power traces of multiple plaintext/ciphertext pairs are used, since new unknown 256-bit variables are introduced in each new pair, the complexity of IDASCA would be greater than single pair attack and IDASCA is also prevented.

C. Attack With Errors and Dynamic Deduction Size

In practical, an adaptive technique can apply the dynamic μ approach and select the highest n HW deductions foreach leak, when the sum of these probabilities are over a fixed threshold T .When IDASCA is applied on AES for this scenario, the attack complexity is much lower than the attack with fixed μ.Thus the threshold and and value of μ is not fixed.

D.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

E. 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.

F. 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.

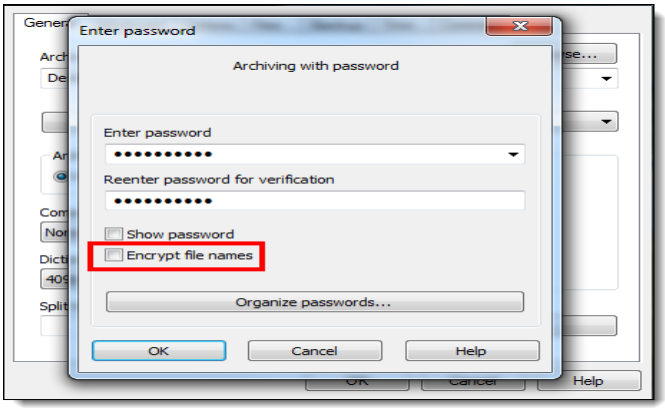

Fig. 3. Encryption

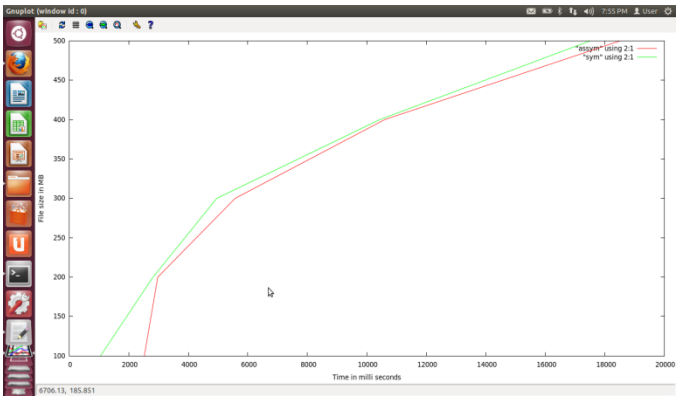

Fig 4. Timing Attack

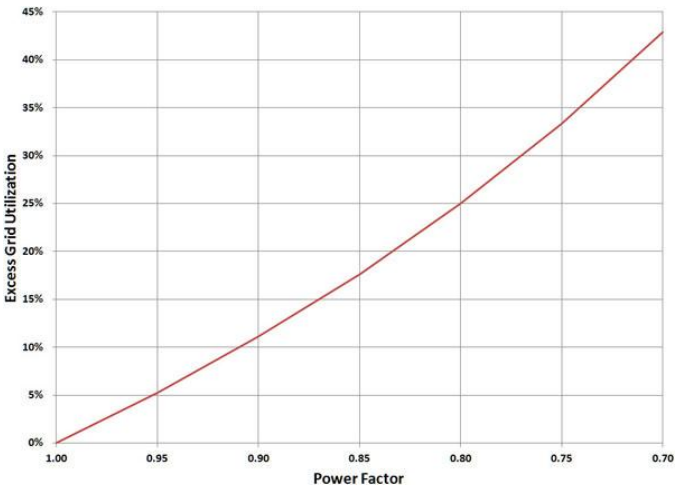

Fig 5. Power Monitoring Attack

6. Conclusion

There are several interesting problems for further research. First is how to utilize the different probabilities of multiple deductions to generate the rank for each state candidate (four bytes). Second is how to apply the key ranking enumeration and estimation algorithms to, and exploit the rank to estimate the remaining key strength in more accurately.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] K. Jung, "Text information extraction in images and video: A survey,"PatternRecognit., vol. 37, no. 5, pp. 977–997, May 2004.

[2] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey," Int. J. Document Anal. Recognit., vol. 7, nos. 2–3,pp. 84–104, 2005.

[3] J. Zhang and R. Kasturi, "Extraction of text objects in video documents: Recentprogress," in Proc. 8th IAPR Int. Workshop Document Anal. Syst., Sep. 2008, pp. 5–17.

[4] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robustreading competitions," in Proc. Int. Conf. Document Anal. Recognit., 2003, pp. 682–687.

[5] S. Lucas, "Icdar 2005 text locating competition results," in Proc. Int.Conf. DocumentAnal. Recognit., 2005, pp. 80–84.

[6] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge2: Reading text in scene images," in Proc. Int. Conf. Document Anal. Recognit., 2011, pp. 1491–1496.

[7] X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in Proc. IEEEComput. Soc. Conf. Comput. Vis. Pattern Recognit., 2004, pp. 366–373.

[8] X. Chen and A. Yuille, "A time-efficient cascade for real-time object detection: Withapplications for the visually impaired," in Proc. IEEE Conf. Comput. Vis. PatternRecognit., Workshops, Jun. 2005, pp. 1–8.

[9] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partitionand grouping," IEEE Trans. Image Process., vol. 20, no. 9, pp. 2594–2605, Sep. 2011.

[10] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations innatural images," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp.1083–1090.