ADVANCED ENCRYPTION STANDARD MODES OF OPERATION USING VITAL INFROMATION FOR DRIP RECOVERY WITH CLAIM

Mr. S.P.SANTHOSHKUMAR[#1],Mr. M. PRAVEENKUMAR[#2],Mr. T. GOWDHAMAN[#3],

#1Assistant Professor, Department of Computer Science and Engineering,
Rathinam Technical Campus, Coimbatore, India,
#1Assistant Professor, Department of Information Technology,
Rathinam Technical Campus, Coimbatore, India,
#1Assistant Professor, Department of Computer Science and Engineering,
Rathinam Technical Campus, Coimbatore, India,
[1]santhosh.cse@rathinam.in
[2]praveen.it@rathinam.in
[3]gowdhaman.cse@rathinam.in

***Abstract* - Side Channel Attacks (SCA) are attacks that are based on Side Channel Information. Side channel information is information that can be retrieved from the encryption device that is neither the plaintext to be encrypted nor the cipher text resulting from the encryption process. The real threat of SCA lies in the ability to mount attacks over small parts of the key and to collective information over different encryptions. The threat of SCA can be dissatisfied by changing the secret key at every run. Indeed, many contributions in the domain of leakage resilient cryptography tried to achieve this goal. In this paper, we propose a generic framework of lightweight key updating that can protect the current cryptographic standards and evaluate the minimum requirements for heuristic SCA-security and timing attack.Then, we propose a complete solution to protect the implementation of any standard mode of Advanced Encryption Standard (AES) using vital information for drip recovery with claim.**

***Keywords* - Advanced Encryption Standard, Side Channel Attacks, Hardware security (side channels), Timing Attackand CryptographicScheme.**

I. INTRODUCTION

Side-channel cryptanalysis is a branch of cryptography in which sensitive information isgained from the physical implementation of a target cryptosystem. This is in contrast withother forms of cryptanalysis where the algorithms and their underlying computational problems are attacked. All electronic devices leak information in a multitude of ways.

Prominentexamples of this are temperature, power consumption, time taken for computations, acoustics and electromagnetic emanations. In general, these types of information leakages may betied in some way to the types of operations that the cryptographic algorithm is performing [5].

Each center point fills in as a host and in addition a switch in the networks [1]. While getting data from the peers, the peer requires joint effort with each other to forward the data bundles, and this is known as Wireless Local Area Network [3].

This trademark gives a major issue from the parts of security. In fact, an application affects some stringent role on the security of the framework topology, routing and information activity [2]. For instance, the region and composed exertion of malignant peers in the framework may cut down the routing process that collapsesthe framework operations.

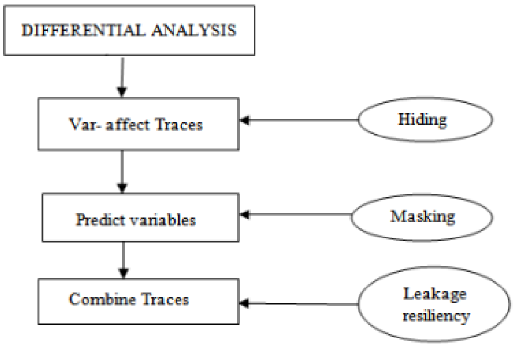

Fig. 1: Supports of SCA attacks

The three support of SCA attacks were listed as follows:

- The leakage traces were affected by the complex variables.
- The hypothetical complex variables were valued by Eve.
- The information can be combined from changed traces.

Hiding techniques depends on breaking the association between the intermediate variables and the recognizable spillage by minimizing the trace utilizing Signal to Noise proportion. This can be experts using balanced circuits or confusion generators. Disastrously, a cryptographic module utilizing hiding plans expends a part of territory [5].

However, the proposed solutions were computationally intensive and were not designed to solve the problem of the current cryptographic schemes. In this paper, we propose a generic framework of lightweight key updating that can protect the current cryptographic standards and evaluate the minimum requirements for heuristic SCA-security.

Then, we propose a complete solution to protect the implementation of any standard mode of Advanced Encryption Standard. Our solution maintains the same level of SCA-security (and sometimes better) as the state of the art, at a negligible area overhead while doubling the throughput of the best previous work.

II. LITERATURE SURVEY

The threat considered in this paper is that Eve recovers the secret key of a hardware implementation of AES. Classical cryptography assumes that Eve can choose the input plaintext and the output cipher text [6]. SCA further assumes that Eve knows the underlying implementation and can capture the instantaneous power consumption. In the domain of leakage resiliency, it is also assumed that Eve can run any polynomial-time function (called leakage function) on the power consumption to recover some bits of the secret key.

The two categories of key-updating are stateless and stateful [7]. One mechanism or the other is sufficient for a limited set of applications. However, the two mechanisms are both required for a complete and generic solution.

Stateless key-updating assumes that the two communicating parties share only the secret key and a public variable (nonce) i.e. there is no shared secret state between them[8]. This updating mechanism is required whenever there is no synchronization between the two communicating parties e.g. during initialization of a secret channel. Stateless key-updating provides a complete solution for applications with single cryptographic execution e.g. challenge response protocols.

Stateful key-updating assumes that the two communicating parties share a common secret state (other than the key). They both can update the secret key into a new key without requiring any external variables. This scheme can provide a complete solution for synchronized applications e.g. key-fobs.

In this paper, the following are monitored for consider to do the further process

- There is no provably secure construction that supports stateless key-updating.
- Intuitively speaking, the secret key cannot be updated to a new key unless a public variable is used (assuming no synchronization). Once a public variable interacts with a secret key, SCA will be possible. Some contributions tried to secure the stateless key-updating mechanism through hiding and masking.
- Although this approach limits the implementation overhead exclusively to the key-updating mechanism, allowing the use of unprotected cryptographic cores, the overall overhead is still significant

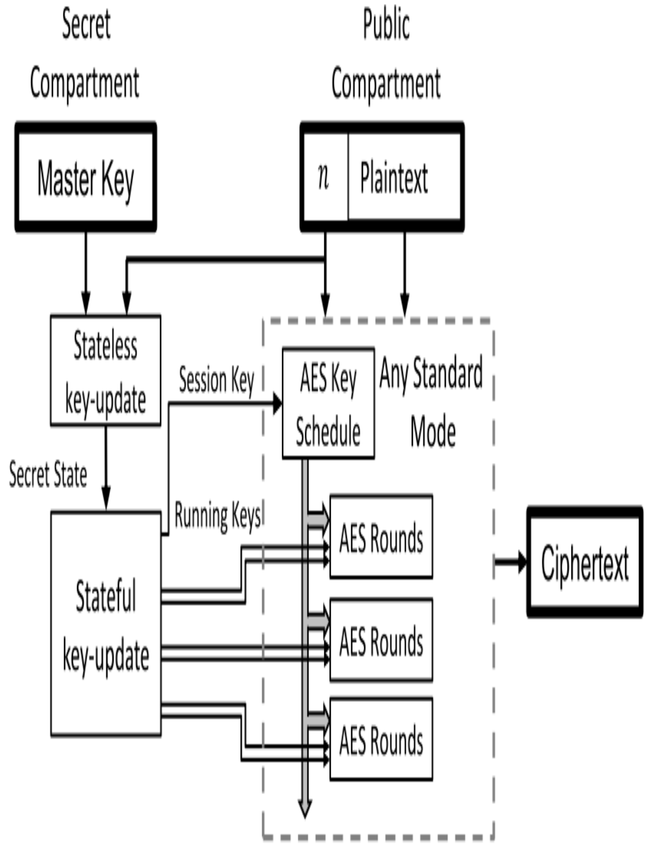

Fig. 2: Block Diagram of AES

III. STATEFUL AND STATELESS KEY UPDATION

The proposed solution at the system level works as follows. We assume that an application on Device A needs to send secure data to an application on Device B. Both devices share a secret key, which we name master key. They can initiate the channel by exchanging a public nonce, and send the secure data using any cryptographic primitive (AES) running in a mode of operation. Although the black-box security of these modes is guaranteed by the cryptographic primitive, security is not guaranteed if Eve can monitor Device A[6].

Here, we target protecting the master key against any SCA attack. Device A starts with a stateless key-updating mechanism to compute a pseudorandom secret state out of the master key and the nonce. Then, the stateful key-updating is executed, to compute running keys.

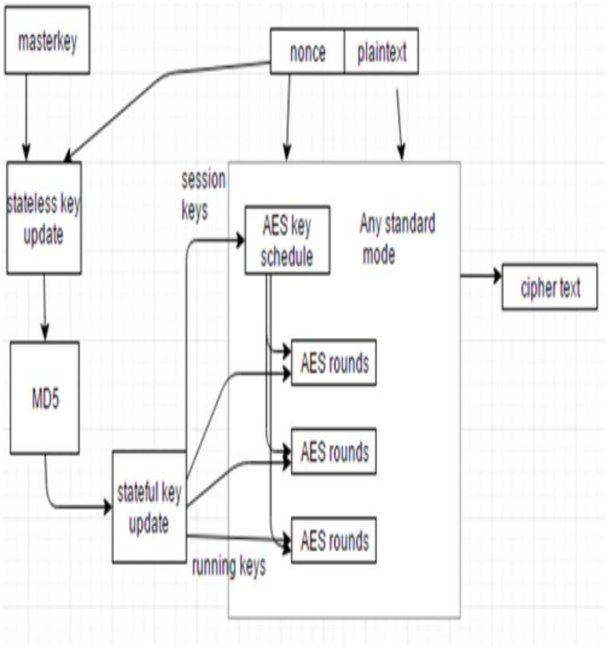

Fig.3: MD5 algorithm with stateful and stateless key updation

In this the input from the AES rounds will be a plaintext. The plain text and randomly generated key be combined by aprocess called encryption. The output from the first round is sent as the input for the second round. The sameprocedure should be repeated for the subsequent rounds. The randomkeys shouldbe used exactly once to improve more security to the system. The first key should be the session key and rest all are the running keys. The session key is represented as the ki and running keys are represented by rki. The output from this known as the cipher text. This stateless and stateful key updation is used to provide security to the key generation algorithm[7]. In order to enhance the security this paper implements the MD5 algorithm in between stateless and stateful key updation.

Many researchers have been focused in this area. In paper [11] they proposed new DFA has been proposed against AES algorithm for different length of the bit key. This has the capability of retrieving the fault and correctly encrypted data. With three pair they could find the key of AES.

In paper [12] the implementation of AES cryptography which is suitable for 128 bit and performs fault injection attack against unsecured AES and fault detection which is exposed is discussed. New approach in paper [13] that would reduce the time complexity of Algebraic side channel attack is introduced. This paper proposes the complete diffusion feature in one AES rounds is exploited. In paper [14]the increase in number of rounds provides more security for the cryptographic process of AES is discussed. This mechanism provides the less transfer of data with providing high speed.

Finally, the actual cryptographic mode is called using the input data and the same previously used nonce.

Our solution honors the tree structure for the stateless key-updating. Each step of the tree involves processing a single bit of the nonce through a lightweight whitening function (Wt: whitening in the tree).

The tree starts from the master key, and ends with a pseudorandom secret state. For the stateful key-updating, we use a chain of whitening functions (Wc: whitening in the chain). Every execution of the whitening function generates a new running key.

We focus on achieving a sound security at the smallest implementation cost (area and performance). To achieve this goal, we propose a generic framework for lightweight key-updating and evaluate the minimum requirements for SCA-security. Then, we propose a solution that maintains the same level of SCA-security (and sometimes better) as the state of the art, at a negligible area overhead while doubling the throughput of the best previous work.

IV. EXPERIMENTAL RESULT & ANALYSIS

The program can be retested by using the different types of output. The input can be alphabets, symbols, numbers, etc.

A. Time

Time is the important criteria for finding the efficiency of an algorithm. The comparison table will be as follows.

TABLE I

AES	AES with stateless and stateful key update	AES with stateless, stateful key update and MD5 algorithm
1.9874s	1.983s	7.8742s

B. Layers

Layer defines the security of the system. More the number of layers more will be the security.

	AES	AES with stateful, stateless key update	AES with stateful, stateless key update. and MD5 hash algorithm
Number of rounds	1	2	3
Encryption	1	10	10

V. CONCLUSION

Key administration assumes an essential part in cryptography as the premise for securing cryptographic systems that provides privacy, entity validation, information origin validation, information integrity, and computerized signatures. The objective of a decent cryptographic outline is to decrease more mind boggling issues to the correct administration and safe-guarding of a smaller number of cryptographic keys, at last secured through trust in equipment or programming by physical detachment or procedural controls. Dependence on physical and procedural security (e.g., secured rooms with detached hardware), tamper-resistant equipment, and trust in countless people is minimized by concentrating trust in smaller number of effortlessly checked, controlled, and dependable components anda generic framework for lightweight key-updating and evaluate the minimum requirements for SCA-security. During these key updation there is a chance of side-channel attack by the outsiders. In order to provide more security to the side-channel attack this paper implemented the stateless and stateful key updationalong with MD5. It is a hashing algorithm that is implemented to provide more security to the key generation than the stateless and stateful key updation. The efficiency of our proposed system lies in the running time where generation of hash for stateless key using MD5 hash algorithm does not take more time. The security has been enhanced with the addition of layer MD5 in our proposed system.

ACKNOWLEDGEMENT

	Mr. S. P. Santhoshkumar is currently working as an Assistant Professor in Computer Science and Engineering at Rathinam Technical Campus, Tamilnadu, India. He received a Master of Engineering from Anna University of Technology, Coimbatore, India.
	Mr. M. Praveenkumar is currently working as an Assistant Professor in Information Technology at Rathinam Technical Campus, Tamilnadu, India. He received a Master of Engineering from Anna University of Technology, Coimbatore, India.
	Mr. T. Gowdhaman is currently working as an Assistant Professor in Computer Science and Engineering at Rathinam Technical Campus, Tamilnadu, India. He received a Master of Engineering from Anna University, Chennai, India.

REFERENCES

[1] MostafaTaha, Member, IEEE, and Patrick Schaumont, Senior Member, IEEE, “Key Updating for Leakage Resiliency With Application to AES Modes of Operation”, IEEE Transactions On Information Forensics And Security, Vol. 10, No. 3, March , 2015

[2] P.-C. Tsou, J.-M.Chang, H.-C.Chao, and J.-L. Chen, “CBDS: A cooperative bait detection scheme to prevent malicious node for MANET based on hybrid defense architecture,” in Proc. 2nd Intl. Conf. Wireless Commun., VITAE, Chennai, India, Feb. 28–Mar., 03, 2011, pp. 1–5.

[3] S. Corson and J. Macker, RFC 2501, Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, Jan. 1999. (Last retrieved March 18, 2013).

[4] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In M.Wiener, editor, Advances in Cryptology - CRYPTO’99 , volume 1666 of Lecture Notes in Computer Science, pages 388-397, Berlin, Hiedelberg, New York, 1999. Springer-Verlag.

[5] C. Chang, Y.Wang, and H. Chao, “An efficientMesh-based core multicast routing protocol on MANETs,” J. Internet Technol., vol. 8, no. 2, pp. 229– 239, Apr. 2007.

[6] S. Das, J.K.M.S. Uz Zaman, R. Ghosh, 2013, Generation of AES S-Boxes with various modulus and additive constant polynomials and testing their randomization” International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA).

[7] Junfeng Chu, Mohammed Benaissa, 2011, A Novel Architecture of Implementing Error Detecting AES using PRNS”, Euromicro Conference on Digital System Design.

[8] V.Gopi, Dr.E.Logashanmugam, 2013, Design And Analysis Of Nonlinear AES S-Box And Mix-Column Transformation With The Pipelined Architecture“, International Conference on Current Trends in Engineering and Technology, ICCTET’13.

[9] Raphael C.-W. Phan, 2014, Impossible differential cryptanalysis of round Advanced Encryption Standard”, Information Processing Letters 91, 33-38.

[10] Orr Dunkelman, Nathan Keller, 2010, The effects of the omission of last round’s MixColumns on AES”, Information Processing Letters 110, 304–308.

[11] Christophe Clavier, Antoine Wurcker, 2013, Reverse Engineering of a Secret AES-like Cipher by Ineffective Fault Analysis, Workshop on Fault Diagnosis and Tolerance in Cryptography.

[12] Sourabh Chandra, Bidisha Mandal, S ksafikulAlam, Siddhartha Bhattacharyya, 2015, Content based double encryption algorithm using symmetric key Cryptography”, International Conference on Recent Trends in Computing (ICRTC 2015).

[13] Takeshi Kumaki, Masaya Yoshikawa, Takeshi Fujino, 2012, Cipher-destroying and secret-key-emitting hardware Trojan against AES core, IEEE Transactions on Information Forensics and Security.

[14] Chong Hee Kim,2012, Improved Differential Fault Analysis on AES Key Schedule, IEEE Transactions on Information Forensics And Security, Vol. 7, No. 1.