A Novel Approach Using Advanced Encryption Standard to Implement Hard Disk Security

Minal Moharir[1] and Dr A V Suresh

[1]Department of Information Science & Engineering, R V College of Engineering, Bangalore, India
moharirminal@gmail.com
[2] R V College of Engineering,Bangalore, India
sureshav@rvce.edu.in

ABSTRACT

The objective of the paper is to develop an proficient and economical method for Hard Disk Drive(HDD) Security. The task is implemented using Full Disk Encryption (FDE) with Advanced Encryption Standards(AES) for data security of Personal Computers(PCS) and Laptops . The focus of this work is to authenticate and protect the content of HDD from illegal use. The paper proposes a novel approach for protecting a HDD based on Partial Disk Encryption(PDE) which one of the flavour of FDE. The proposed method is labelled as DiskTrust. FDE encrypts entire content or a single volume on your disk. Symmetric key uses same key for encryption as well for decryption. DiskTrust uses these two technology to build cost effective solution for small scale applications. Finally, the applicability of these methodologies for HDD security will be evaluated on a set of data files with different key sizes.

KEYWORDS

INFORMATION SECURITY, INTEGRITY, CONFIDENTIALITY, AUTHENTICATION, ENCRYPTION.

1. INTRODUCTION

As of January 2011 the internet connected an estimated 941.7 million computers in more than 450 countries on every continent, even Antarctica (Source: Internet Software Consortium's Internet Domain Survey; www.isc.org/index.pl). The internet is a collection of loosely connected networks that can be accessible by individual computer hosts, in a variety of ways, to anyone with a computer and a network connection[1]. Thus, the individuals and organizations can reach anywhere, any time with the help of internet.

There is a risk while accessing information on Internet. These risks tincludes valuable information , that can be lost, stolen, changed, or misused. The electronically recorded information is more vulnerable than if the same information is printed on paper and locked in a file cabinet. Intruders do not have to enter an office or home; they may not even be in the same country. They can steal or tamper the information without touching a piece of paper or a photocopier. In this way security of stored information is an important issue. The proposed paper consider the security of Hard Disk Drive which is a fundamental element in computing chain.

The paper organized as follows. Related work, gap & problem is described in Section 2. A view of simulation and experimental design is given in section 3. Simulation results are shown in section 4. Finally the conclusions are drawn section 5.

2. RELATED WORK

The related survey is divided into two parts. The first part is survey about full disk encryption. The second part is survey about advanced encryption standards.

Information security is the process of safe-garding information. It protects its availability, privacy and integrity[2]. Most of the companies store business and personal information on computer. Much of the information stored is highly confidential and not accessible to public. Without this information, business cannot run effectively[3]. A robust Information security systems need to be implemented to protect this information. There are various ways to implement Information security systems. One of the popular technique is full disk encryption. Full Disk Encryption (FDE) is the safest way to protect digital assets[4], the hard drive is a critical element in the computing chain because it is where sensitive data is stored. Full disk encryption increases the security of information stored on a laptop significantly[5]. It helps to keep business critical data secure. Moreover, full disk encryption helps to meet several congressional requirements.

Min Liang and Chao wen Chang (2010 IEEE) described a full disk encryption scheme based on XEN virtual machine which is stored in a security flash disk. XEN is used to encrypt (decrypt) all the data in hard disk and manage the whole system.

Li Jun & Yu Huiping (2010 IEEE) introduced the data encryption technologies of encrypting file system (EFS) and traditional full-disk encryption (FDE), and points out the problems of data encryption of EFS and FDE. Combined with the features of trusted platform module (TPM)[6], this paper constructed a trusted full-disk encryption (TFDE)[7] based on TPM.

The second part of survey covers implementation of Encryption Algorithms. There are many encryption algorithms widely available and used in information security. They can be classified as Symmetric (private) and Asymmetric (public) keys encryption. In Symmetric keys encryption, also called as secret key encryption, only one key is used to encrypt and decrypt data. The encryption/decryption key should be distributed before transmission between entities. Key is an important entity. If flaccid key is used in algorithm then every one may decrypt the data. The Symmetric key encryption strength depends on the size of key used. The algorithm using longer key is harder to break than algorithm using smaller key[8]. The paper uses Symmetric key cryptography to implement disk security. The related work with respect to performance of various encryption algorithm is as follows

Jyothi Yenuguvanilanka Omar Elkeelany (2008 IEEE), the paper analyzed the performance of Rijndael AES Encryption algorithm of key length 128 bits. The hardware models based on HDL and IP core are used to analyze the performance of the algorithm. The encryption time and also the performance metrics such as size, speed and memory utilization are evaluated, using these models.

Dazhong Wang & Xiaoni Li (2009 IEEE) presented the design, implementation and performance of a FIPS – approved cryptographic algorithm – Advanced Encryption Standard (AES), which can be used to protect electronic data.

El-Sayed Abdoul-Moaty ElBadawy & all (ICES 2010), This paper proposed a new chaos AES algorithm for data security. The algorithm is based on substituting the Rijndael affine transformation S-box by another one based on chaos theory. The new S-box has a low correlation and exhibits a significant performance improvement with an acceptable complexity addition.

S.Anandi Reddy & M.Arul Kumar M.Tech.,(2011 IEEE) The paper, used concurrent structure independent fault detection schemes for designing high performance and reliable architecture of the AES. For high performance applications, instead of using look-up tables

alone for the implementation of S-box and inverse S-box and their parity predictions, logic gate implementations based on composite fields are also used.

2.1 Research Gap

- The FDE technology discussed in above survey are encrypting the entire contents of Hard disk Drive. However encryption of the entire HDD is expensive in terms of time and cost. The large scale industries needs this much of tough security, as well they can accommodate big cost. For small industries or institution or personal users the data security is needed for partial data, so need some cost effective security scheme.
- The Symmetric Key Cryptography(SKC) is best for the security of personal devices as no need to share the key.
- Rijndael is most robust & better performance algorithm in available SKC

3. PROBLEM STATEMENT

To develop HDD security technique labeled as DiskTrust. DiskTrust technology uses PDE, creates authorized invisible volume on HD & implements SKC with Rijndael to secure the data stored on secured volume .

The technical objectives of the thesis are:
1. Create Hidden partition
2. Check authentication
3. Store/access data from the hidden volume
4. Execute encryption/decryption algorithm while reading /writing data on Hard Disk Drive.

4. AES IMPLEMENTATION

The standard AES algorithm is as follows:
KeyExpansion: The keys are derived from the cipher key using Rijndael's key schedule

4.1 Initial Round

- AddRoundKey: each byte of the state is combined with the round key using bitwise xor

4.2 Rounds

- SubBytes—It is a non-linear substitution step where each byte is replaced with another according to a lookup table.
- ShiftRows—It is a transposition step where each row of the state is shifted cyclically a certain number of steps.
- MixColumns—Its is a mixing operation which operates on the columns of the state, combining the four bytes in each column
- AddRoundKey

4.3 Final Round (no MixColumns)

- SubBytes
- ShiftRows
- AddRoundKey

4.4 Modifications

To improve the performance following modifications are done:
In order to enhance the security and reliability of AES, we bring in three changes.

In the standard Rijndael implementation the SubBytes oeration is modified as follows:

A non-linear substitution step where each byte is replaced with another according to a Calculation.

Original Operation: The **SubBytes** operation is a non-linear byte substitution. It operates on each byte of the state independently. The **substitution table (S-Box)** is invertible. It is constructed using the composition of two transformations.

Each byte of the state is then substituted by the value in the S-Box whose index corresponds to the value in the state:

$$a(i,j) = SBox[a(i,j)]$$

Modifications: Accessing data form the table, saving recent data on stack takes few memory cycles, to avoid this the following modifications are done:

$$a(i,j) = SubByteFun[a(i,j)]$$
$$SubByteFun = a(i, j) \text{ EX-OR } IV(i, j)$$

IV(i,j) is 4×4 vector.

In each iterative round, apart from the usual four above mentioned operations, we also include two new operations: The Arithmetic Operator and The Route Cipher.

We also modify the key schedule so as to increase the number of the AES encryption rounds. For example, for 16 byte key, we generate 336 bit key instead of the usual 176 bit key. By this process, we are able to successfully process 20+1 rounds instead of the previous 10+1 rounds for the 16 byte key. Lets have a look at the modifications and there implications.

Arithmetic Operation

In this operation, each element of the state is arithmetically added by a number depending on their row number.

The 1st row is added to 1.
The 2nd row is added to 2.
The 3rd row is added to 3.
The 4th row is added to 4.

To retain the symmetric nature of AES, during decryption we have inversed the process by subtracting the corresponding same numbers.

The 1st row is added to 1.
The 2nd row is added to 2.
The 3rd row is added to 3.
The 4th row is added to 4.

Route cipher

In a route cipher, the plaintext was first written out in a grid of given dimensions, and then read off in a pattern given in the key. For example, using the same plaintext:
W R I O R F E O E
E E S V E L A N J

The DiskTrust technique implements AES & improved AES with different key size.

5. SIMULATION & DESIGN

This section describes design and GUI implementation, some of the important results that were found as part of the implementation.

5.1 Implementation of Hidden Volume

DiskTrust Security user interfaces are shown below in the screenshots. The user interface is basically a frame work application where user can use the application

5.1.1 Main Application Window
The Main application window holds multiple options such as CreateVolume, Mount and Dismount All.

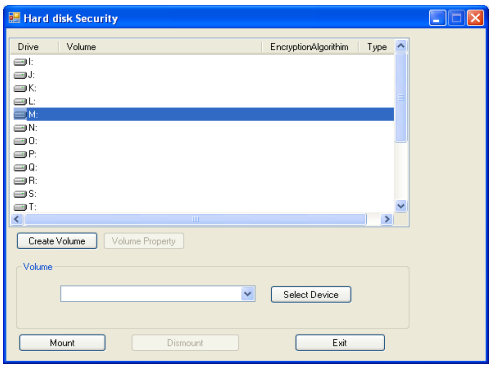

Fig.1 Screenshot of Main Application Window

This is the first step from methodology which create a hidden volume on the HDD.

5.1.2 Volume Location
Volume Location Window allows the user to select location where the user wants to create the volume.

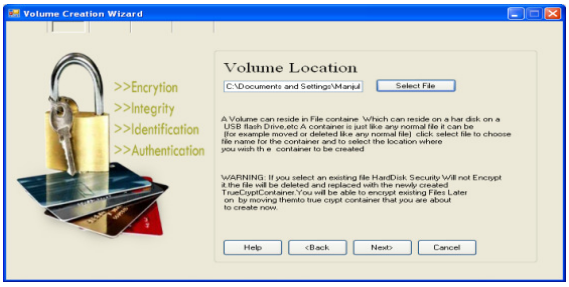

Fig.2 Screenshot of Volume Location Window

5.2 Volume Password

Volume Password window, where user can enter login and password. Here user authentication is checked.

Fig.3 Screenshot of Volume Password Window

5.3Encrypt or decrypt data using AES and imprives AES with different key size while retrieving from hidden volume:

For our experiment, we use a laptop PentiumV 2.4 GHz CPU, in which performance data is collected. In the experiments, the laptop encrypts a different file size ranges from 321K byte to 7.139Mega Byte. Several performance metrics are collected:
1- encryption time
2- CPU process time
3- CPU clock cycles and battery power.
The encryption time is the time that an encryption algorithm needs to produce a cipher text from a plaintext. Encryption time is used to evaluate the encryption scheme. It indicates the speed of encryption. The CPU process time is the time that a CPU is took only to the particular process of calculations. It shows the load of the CPU. The CPU time is directly proportional to the load of the CPU. The CPU clock cycles are used as a metric, reflecting the power consumption of the CPU while operating on encryption operations. Each CPU cycle will consume a small amount of energy.

6 SIMULATION RESULTS

The effect of changing key size of AES on power consumption. The performance comparison point is the changing different key sizes for AES algorithm. In case of AES, We consider the three different key sizes possible. In case of AES it can be seen that higher key size leads to clear change in the battery and time consumption. It can be seen that going from 128 bits key to 192 bits causes increase in power and time consumption about 8% and to 256 bit key causes an increase of 16% . The simulation results with different key sizes are as shown in Table1.

Table1. Time for Different Key Size

AES Key Size	AES 128	AES 192	AES 256
Time in Milliseconds	287	310	330

The graphical representation of the given data is as follows,

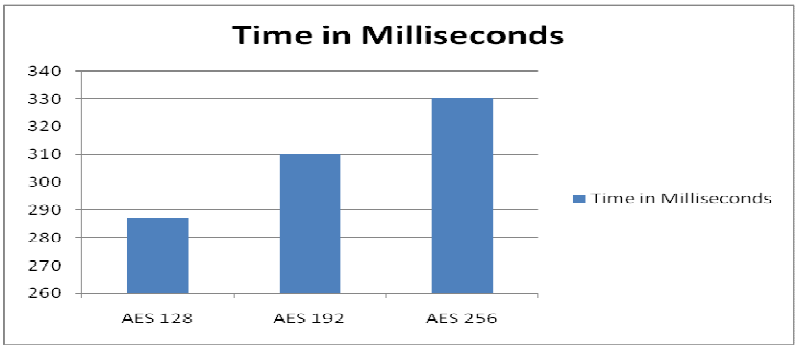

Fig.4 Time with different key size

The analysis with improved AES is shown in Table3.

Table2. Time for Different Key Size

Improved AES Key Size	I-AES 128	I-AES 192	I-AES 256
Time in Milliseconds	250	300	330

The graphical representation of the given data is as follows,

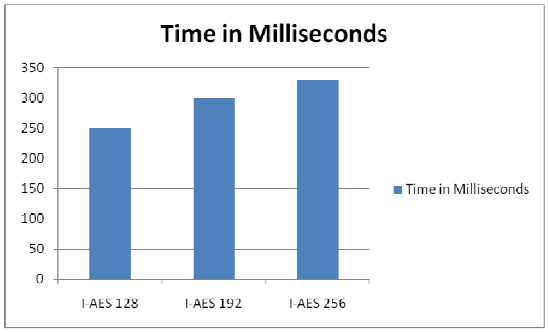

Fig.4 Time with different key size

Some more simulations results are as follows, we shows performance of above two algorithms on different file size.

Table3. Time for Different File size

Algorithm	**500MB File**			
	Rijndael (AES-128)	Proposed	Rijndael (AES-128)	Proposed
Time in ms	285	212	401	358

The graphical representation of the given data is as follows,

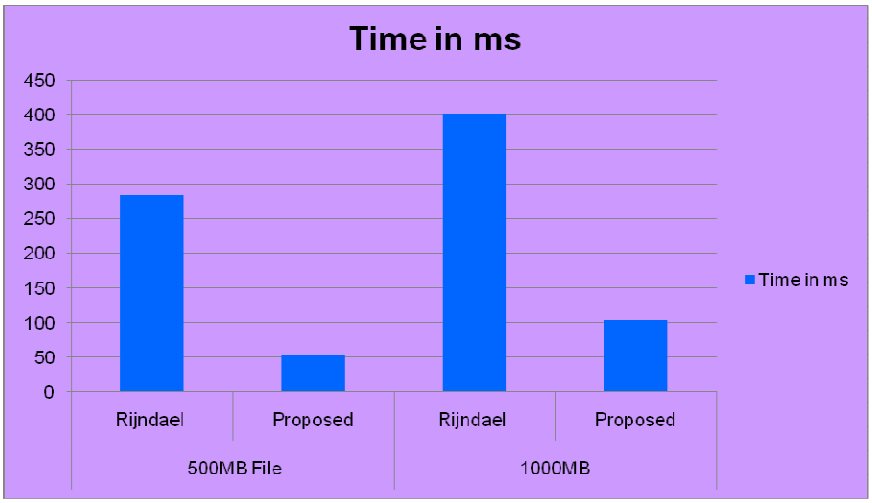

Fig.5 Time with different File size

7. CONCLUSION

Proposed DiskTrust model is better suited for disk security of personal devices such PCs/Laptops over existing techniques. Disktrust model is cost effective & userfriendly.

- DiskTrust stores data on hidden volume so the user's information is in accessible or invisible to the unauthorized user.
- DiskTrust provides user's authentication.
- DiskTrust provides confidentiality by encrypting the data stored in invisible & authenticated disk volume.

With Modified Rijndael algorithm, for 128-bits block & 128-bits key encryption time required is 250ms. This gives 023% performance over standard Rijndeal.

8. REFERENCES

1. J. Blomer, V. Krummel, "Fault Based Collision Attacks on AES", FDTC 2006, pp. 106-120.

2. J. Blomer and J-P Seifert, "Fault Based Cryptanalysis of the Advanced Encryption Standard (AES)", CHESS 2003, pp. 162-181.

3. C.-N. Chen, S.-M. Yen, "Differential Fault Analysis on AES Key Schedule and Some Countermeasures", Australasian Conference on Information Security and Privacy 2003, LNCS 2727, Springer-Verlag, pp. 118–129.

4. C. Giraud, "DFA on AES", 4th International Conference on AES, Springer publisher, pp. 27-41.

5. A. Moradi et al. "A Generalized Method of Differential Fault Attack Against AES Cryptosystem ", Cryptographic Hardware and Embedded Systems - CHES 2006, pp 91-100.

6. FIPS-197 , "Advanced Encryption Standard (AES)", Federal Information Processing Standards Publication 197, http://csrc.nist.gov/publications/, November 26, 2001

7. Karri, R.; Wu, K.; Mishra, P. & Kim, Y., "Concurrent Error Detection Schemes for Fault-Based Side-Channel Cryptanalysis of Symmetric Block Ciphers", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, N. 12, December 2002, pp. 1509-1517.

8. Maistri, P.; Vanhauwaert, P. & Leveugle, R., "A Novel Double-Data- Rate AES Architecture Resistant against Fault Injection", Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC.2007.8, pp. 54- 61.

9. Monnet, Y.; Renaudin, M. & Leveugle, R. "Designing Resistant Circuits against Malicious Faults Injection Using Asynchronous Logic", IEEE Trans. on Computers, Vol. 55, N. 9, September 2006, pp. 1104-1115.

10. Wu, K; Karri, R.; Kuznetsov, G. & Goessel, M., "Low Cost Concurrent Error Detection for the Advanced Encryption Standard", International Test Conference, 2004. pp 1242- 1248.

11. Yen, C.H. & Wu, B.F. "Simple Error Detection Methods for Hardware Implementation of Advanced Encryption Standard", IEEE Trans. On Computers, June 2006, Vol. 55, No.6, pp 720-731.

12. Mozaffari Kermani M., Reyhani-Masoleh A., "Parity-Based Fault Detection Architecture of S-box for Advanced Encryption Standard," pp.572-580, 21st IEEE Int. Symp. on Defect and Fault-Tolerance in VLSI Systems (DFT'06), 2006.

13. Dusart, P.; Letourneux, G. & Vivolo, O. "Differential Fault Analysis on A.E.S.", Applied Cryptography and Network Security, Springer Ed., Vol. 2846/2003, pp 293-306.

14. A. Bosio, G. Di Natale, "LIFTING: a Flexible Open-Source Fault Simulator", Proc. Of the IEEE Asian Test Symposium, 2008, pp. 35-40

ACKNOWLEDGEMENTS

The authors would like to thank all who helped in implementing this work.

Author1: Minal Moharir

Minal Moharir is working as **Lecturer** in R V College of Engineering, Bangalore, India. She has done her master, **M.Tech in Computer Network & Engineering** from VTU, Belgaum, India. She is **pursuing** her **Ph.D** from Avinashi Lingam University, Coimbatore, India. She has **8-years teaching experience**. She has presented many papers in various national & International conferences and reputed Journals.

Author2: Dr. A.V.Suresh

Dr A V Suresh is working as a **Head of the Department** in **R V College of Engineering, Bangalore, India.** He is **Ph.D from Mysore University in 2003**. He has **21-years teaching experience** and 8-years research experience. His area of interest is **Machine Tool Engineering , Quality assurance, Operation Research**. He has presented many papers in various national & International conferences and reputed Journals.