International Journal of Technology and Engineering System (IJTES)
©gopalax-Publisher of International Journals
ISSN: 0976-1345 pp. 34-37
available at : www.ijcns.com
Vol 3.No.1 Jan- March 2012

UR6: A Fast And Flexible Encryption Algorithm To Implement HardDisk Security

G. RAMESH[1] Dr. R. UMARANI[2]
1Research Scholar, Research and Development Centre,Bharathiyar University, Coimbatore,INIDA
[2]Associate Professor in Computer Science, Sri Sarada college for women, Salem -16, INDIA
[1]mgrameshmca@yahoo.com, [2]umainweb@gmail.com

Abstract: **The main objective of the paper is to study and develop an efficient method for Hard Disk Drive(HDD)and removable storage media's data Security using Full Disk Encryption (FDE) with our UR6 encryption algorithm for data security specifically for Personal Computers(PCS) and Laptops. Encryption of sensitive data and personal identifiable information is no longer optional. Whether it is data breach legislation, privacy regulation or good old fashioned intellectual property protection, organizations must protect their sensitive data. The focus of this work is to authenticate and protect the content of HDD and removable storage media from illegal use. The paper proposes adaptive methods for protecting a HDD and removable storage media based on Full Disk Encryption. The UR6 method is labeled as Disk Trust. The Full Disk Encryption encrypts entire content or a single volume on your disk. Disk Trust implements Symmetric key cryptography with our UR6 algorithm. This UR6 algorithm is designed and developed by G. Ramesh etal., in the year 2011. Finally, the applicability of these methodologies for Hard Disk Drive security will be evaluated on a set of data files with different key sizes.**

Keywords**:** Information Security,Integrity UMARAM, confidentiality**,** Authentication, Encryption ,Full Disk Encryption

1. INTRODUCTION

This disk encryption program creates multiple encrypted disks for storage of confidential information. Encrypted disks behave like regular disks, your programs can use them in a usual way, and there is no need to reconfigure them. Automatic data encryption is transparent files are encrypted on the fly when they are written to the encrypted disk, and decrypted when read from it. Access to the encrypted disk is monitored by *Disk Firewall*, a unique data protection mechanism that guards your data from Trojans, viruses or other types of malware. Disk Firewall controls which applications are allowed to access the encrypted disk. If a specific application is not found in the white-list, it will be unable to read or change the confidential information stored on the encrypted disk.

- Disk encryption uses disk encryption software to encrypt every bit of data that goes on a Disk or disk volume.
- Disk encryption prevents unauthorized access to data storage.
- The term "full disk encryption"(or whole disk encryption) is often used to signify that everything on a disk is encrypted.

The related survey is divided into two parts. The first part is survey about full disk encryption. The second part is survey about our UR6 algorithm standards.

A. Full Disk Encryption

Information security is the process of protecting information. It protects its availability, privacy and integrity. More companies store business and individual information on computer than ever before. Much of the information stored is highly confidential and not for public viewing. Without this information, it would often be very hard for a business to operate. 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, 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. It helps to keep business critical data absolutely confidential. Moreover, full disk encryption helps to meet several legislative requirements.

Various techniques to implement FDE are discussed as follows: Tabel1:

Table 1.1 Various Techniques to implement FDE

Name of The Technique	*Developed By*	*Released year*	*Licenced By*	*Operating System*
TrueCrypt	*TrueCrypt Foundation*	*2009*	*Freeware*	*Linux and Windows*
Discryptor	*Cosect*	*2008*	*Commercial, Closed Source*	*Windows and Vista*
DriveSntry	*DriveSentry*	*2008*	*Freeware, Closed Source*	*Windows and Vista*
R-Crypto	*R-Tools Technology Inc*	*2008*	*Free, Closed Source*	*Windows XP, Vista*

Brief definitions of the most common encryption techniques are given as follows:

DES: (Data Encryption Standard), was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology).DES is (64 bits key size with 64 bits block size) .Since that time, many attacks and methods recorded the weaknesses of DES, which made it an insecure block cipher [3],[4].

3DES: 3DES is an enhancement of DES; it is 64 bit block size with 192 bits key size. In this standard the encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time. It is a known fact that 3DES is slower than other block cipher methods [3].

RC2: RC2 is a block cipher with a 64-bits block cipher with a variable key size that range from 8 to128 bits. RC2 is vulnerable to a related-key attack using 234 chosen plaintexts [3].

RC4: RC4 was developed by Ron Rivest in 1987. It is a variable-key-size stream cipher. It is a cipher with a key size of up to 2048 bits (256 bytes). The algorithm is very fast. Its security is unknown, but breaking it does not seem trivial either. Because of its speed, it may have uses in certain applications. It accepts keys of arbitrary length. RC4 is essentially a pseudo random number generator, and the output of the generator is exclusive-ored with the data stream. For this reason, it is very important that the same RC4 key never be used to encrypt two different data streams.

SEED: SEED is a block cipher developed by the Korea Information Security Agency since 1998. Both the block and key size of SEED are 128 bits and it has a Feistel Network structure which is iterated 16 times. It has been designed to resist differential and linear cryptanalysis as well as related key attacks. SEED uses two 8x8 S-boxes and mixes the XOR operation with modular addition. SEED has been adopted as an ISO/IEC standard (ISO/IEC 18033-3), an IETF RFC, RFC 4269 as well as an industrial association standard of Korea (TTAS.KO-12.0004/0025).

SERPENT: Serpent is a very fast and reasonably secure block cipher developed by Ross Anderson, Eli Biham and Lars Knudsen. Serpent can work with different combinations of key lengths. Serpent was also selected among other five finalists to become the new federal Advanced Encryption Standard (AES).

TEA: TEA is a Tiny Encryption Algorithm is a very fast and moderately secure cipher produced by David Wheeler and Roger Needham of Cambridge Computer Laboratory. There is a known weakness in the key schedule, so it is not recommended if utmost security is required. TEA is provided in 16 and 32 round versions. The more rounds (iterations), the more secure, but slower.

BLOWFISH*:* Blowfish is block cipher 64-bit block - can be used as a replacement for the DES algorithm. It takes a variable length key, ranging from 32 bits to 448 bits; default 128 bits. Blowfish is unpatented, license-free, and is available free for all uses. Blowfish has variants of 14 rounds or less. Blowfish is successor to Twofish [5].

TWOFISH: Twofish is a symmetric block cipher. Twofish has a block size of 128 bits and accepts keys of any length up to 256 bits.Twofish has key dependent S-boxes like Blowfish. Twofish encryption algorithm was designed by Bruce Schneier, John Kelsey, Chris Hall, Niels Ferguson, David Wagner and Doug Whiting. The National Institute of Standards and Technology (NIST) investigated Twofish as one of the candidates for the replacement of the DES encryption algorithm.

AES*:* AES is a block cipher .It has variable keylength of 128, 192, or 256 bits; default 256. it encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size. AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices[6]. Also, AES has been carefully tested for many security applications [3], [7].

RC6*:* RC6 is block cipher derived from RC5. It was designed to meet the requirements of the Advanced Encryption Standard competition. RC6 proper has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits. Some references consider RC6 as Advanced Encryption Standard [8].

IDEA: IDEA is the second version of a block cipher designed and presented by Lai and Massey. It is a 64-bit iterative block cipher with a 128-bit key and eight rounds. While the cipher is not Feistel, decryption is carried out in the same manner as encryption once the decryption subkeys have been calculated from the encryption subkeys. The cipher structure was designed to be easily implemented in both software and hardware, and the security of IDEA relies on the use of three incompatible types of arithmetic operations on 16-bit words. The speed of IDEA in software is similar to that of DES.

CAST: CAST stands for Carlisle Adams and Stafford Tavares, the inventors of CAST. CAST is a popular 64-bit block cipher which belongs to the class of encryption algorithms known as Feistel ciphers. CAST-128 is a DES-like Substitution-Permutation Network (SPN) cryptosystem. It has the Feistel structure and utilizes eight fixed S-boxes. CAST-128 supports variable key length between 40 and 128 bits.

CAST-128: CAST is resistant to both linear and differential cryptanalysis. Currently, there is no known way of breaking CAST short of brute force. CAST is now the default cipher in PGP.

UMARAM: The UMARAM is a Symmetrical encryption algorithm. The key generation generates 16-keys during 16-rounds.One key of them is used in one round of the encryption or decryption process. The new algorithm uses a key size of 512-bits to encrypt a plaintext of 512-bits during the 16-rounds. In this Algorithm, a series of transformations have been used depending on S-BOX, different shift processes, XOR-Gate, and AND-Gate. The S-Box is used to map the input code to another code at the output. It is a matrix of16×16×16 .The S-Box consists of 16-slides, and each slide having 2-D of16×16 . The numbers from 0 to 255 are arranged in random positions in each slide.

UR5: The UR5 Algorithm is a new symmetrical encryption algorithm was designed by G.Ramesh etal. in the year 2010. [10].A block encryption algorithm is UR5 in this approach. In this Algorithm, a series of transformations have been used depending on S-BOX, XOR Gate, and AND Gate. The UR5 algorithm encrypts a plaintext of size 64-bits by a key size of 64-bits. It uses eight rounds for encryption or decryption process. It overcomes some drawbacks of the other algorithms.

This paper is organized as follows. Section 2 gives UR6 Symmetrical Algorithm, Section 3 gives experimental design for metric of new algorithm. Section 4 gives experimental results of the UR6 algorithm. Conclusions are presented in section 5.

2. BACKGROUND AND SCOPE

The Full Disk encryption 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. DiskTrust, the technology UR6 proposed here, creates a hidden volume on HDD, which is not visible, accessible to the unauthorized user. The data store in this hidden volume is encrypted using our UR6 encryption algorithm. In this way DiskTrust technology follows CIA properties of secure information along with hidden partition.

A. Problem Definition

DiskTrust technology implements security on the hard drive itself, to provide a foundation for trusted computing.

The technical objectives of the paper are:

1. Create Hidden partition
2. Execute issuance protocol to check authentication.
3. Execute encryption/decryption algorithm while reading /writing data on Hard DiskDrive

3. SIMULATION AND DESIGN

This section describes some of the important results that were found as part of the implementation.

A. 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

B. Main Application Window

The Main application window holds multiple options such as Device Selection, Create Volume, Mount and Dismount All.

Figure. 1 Screenshot of Main Application Window

C. Volume Location

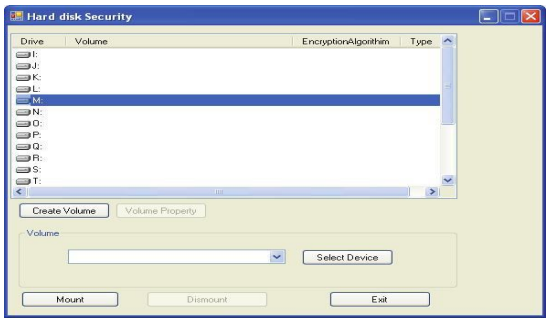
Volume Location Window allows the user to select the file for which user want to create volume.

D. Volume Password

Volume Password window will allow the user to enter the password and confirm Password. Password implements user authentication.

Figure.2 Screenshot of Volume Location Window

Figure 3. Screenshot of Volume Password Window



E. Encrypt or decrypt data 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 10.139Mega Byte. Several performance metrics are collected: They are 1.Encryption Time 2. CPU Process Time3. CPU Clock Cycles and battery Power.

*1- **Encryption time***

The encryption time is considered the time that an encryption algorithm takes to produce a cipher text from a plaintext. Encryption time is used to calculate the of an encryption scheme. It indicates the speed of encryption.

*2- **CPU process time***

The CPU process time is the time that a CPU is committed only to the particular process of calculations. It reflects the load of the CPU. The more CPU time is used in the encryption process, the higher is the load of the CPU.

*3- **CPU clock cycles and battery power.***

The CPU clock cycles are a metric, reflecting the energy consumption of the CPU while operating on encryption operations. Each cycle of CPU will consume a small amount of energy.

Table 2: Siumlation results with different keysizes

UR6 Algorithm Keysize	UR6 128	UR6 192	UR6 256
Time in Milliseconds	**310**	**366**	**410**

4. SIMULATION RESULTS

The effect of changing key size of UR6 on power consumption. The performance comparison point is the changing different key sizes for UR6 algorithm. In case of UR6, We consider the three different key sizes possible. In case of UR5 it can be seen that higher key size leads to clear change in the battery and time consumption. The simulation results with different keysizes are as shown in Table2.

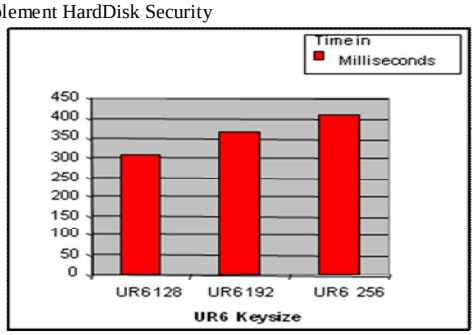


Figure. 4 Time with different Key Sizes

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% [9].

5. CONCLUSION

Full disk encryption (FDE) appears to offer an ideal solution to the losses of data on laptops, CDs and thumb drives. The Disk Trust technique proposed in the paper instead encrypting the entire contents of disk, it encrypts a data stored on single volume which less expensive in terms of time & cost. Disk trust provides authentication, integrity and confidentiality for stored data. Disk Trust implements Symmetric key cryptography with UR6. The UR6 is more promising according to results.

REFERENCE

[1] William A. Arbaugh, Angelos D. Keromytis, David J.Farber, and Jonathan M. Smith, Automated Recovery in a Secure Bootstrap Process, Network and Distributed System Security Symposium, Internet Society, March 1998

[2] Siani Pearson Trusted Computing Platforms: TCPATechnology in Context, Prentice Hall PTR, 2002.

[3] G. Piret, J.J. Quisquater. "A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad" Workshop on Cryptographic Hardware and Embedded Systems, CHES 2003. LNCS, vol. 2779, Springer- Verlag, pp. 77-88, 2003.

[4] Hiroshi Maruyama and others, Linux with TCPA Integrity Measurement, IBM Research, Tokyo Research Laboratory, January 28, 2003.

[5] Hiroshi Maruyama and others, Trusted Platform on demand (TPod), IBM, and February 1, 2004.

[6]. Michael Austin Halcrow, eCryptfs: An Enterprise-class Cryptographic Filesystem for Linux, International Business Machines Inc., 2005.

[7] Daniela A. S. de Oliveira, Jedidiah R. Crandall, and others, Exec Recorder: VM-based full-system replay for attack analysis and system recovery, Proceedings of the 1st workshop on Architectural and system support for improving software dependability, ACM, 2006.

[8] Peng Shaunghe, Han Zhen, Enhancing PC Security with a U-Key, IEEE Security and Privacy, Volume 4, Issue 5, September 2006.

[9] López-Ongil et al. "Autonomous Fault Emulation: A New FPGA-based Acceleration System for Hardness Evaluation" IEEE T. on Nuclear Science, Vol. 54, Issue1, Part 2, pp. 252- 261, Feb. 2007.

[10] W. Diffie and S. Landau, Privacy on the Line: The Politics of Wiretapping and Encryption, updated and expanded edition, MIT Press, 2007, pp. 280–285.
[11] H. Rezaei Ghaleh, PC Secure Bootstrapping, M.Sc. Thesis, Islamic Azad University of Qazvin, March 2008.
[12] BitLocker Drive Encryption Technical Overview, Microsoft TechNet, 2008.
[13] Alexei Czeskis David J. St. Hilaire Karl Koscher Steven D.Gribble, 2008, Defeating Encrypted and Deniable File Systems:TrueCrypt v5.1a and the Case of the Tattling OS and Applications
[14] David Challener, Kent Yoder, Ryan Catherman, DavidSafford, Leendert Van Doorn, A Practical Guide to TrustedComputing, IBM Press, 1 edition, January 6, 2008.
[15] Minal Moharir, Dr.A.V.Suresh " An adaptive technique using Advanced Encryption standard to implement Harddisk Security" International journal of innovative Technology & Creative engineering, Vol.1 No.4 April 2011.
[16] G. Ramesh, Dr. R. Umarani "A Novel Symmetrical Encryption Algorithm with High Security Based on Key Updating" gopalax Journals , International Journal of Computer Network and Security (IJCNS) Vol. 3 No. 1 pp 57-69, http://www.ijcns.com/pdf/207.pdf
[17]G. Ramesh and R. Umarani , Data Security in Local Area Network Based on Fast Encryption Algorithm, Communications in Computer and Information Science, 2010, Volume 89, Part 1, 11-26
http://www.springerlink.com/content/m3301508219h7g66/
[18]Ramesh, G. Umarani, R. ,UMARAM: A novel fast encryption algorithm for data security in local area network http://ieeexplore.ieee.org /xpl/ freeabs_all.jsp? arnumber=5670740