A Review of Data Confidentiality in Removable Media and Symmetric Key Encryption Algorithms

Neel N. Shah
M.E. Scholar
Computer Engineering (IT Systems and Network Security)
Gujarat Technological University,
Ahmedabad

Jigar A. Raval
In - Charge
Computer Center,
Physical Research Laboratory,
Ahmedabad

Gardas Naresh Kumar
Co-ordinator
Centre for Development of Advanced Computing,
Pune

***Abstract*— Data confidentiality in any research organization refers to protecting privacy of an individual's data actively involved in their respective research domains. Data breach can therefore be a big threat to any research organization. Hence it is very important to maintain the confidentiality of data. USB devices provides faster speed for data transmission and are easy to carry at any place, but many research organization denies the usage of removable devices because of security concerns. It is also observed that most of the data theft happens through USB or removable devices as they are more convenient and easy to obtain. In this paper we will analyze different data encryption techniques for removable devices using web based authentication, security tools available for data encryption, performance analysis of various implementations of encryption algorithms, and an efficient data encryption system to compare the execution time and data security with other existing encryption algorithms.**

Keywords — Information security, Cryptography, Encryption, Data confidentiality, Removable media, Ciphertext, Authentication, USB security

I. INTRODUCTION

Cryptography is one of the main pillar of information security. In the present scenario, the data is being used extensively and the amount of data transmission over the network is increasing exponentially. Due to this, safeguarding confidential data is becoming a challenging task as data transmitted over the network is in bulk which leads to possibilities of data theft and data leakage.

Data confidentiality is defined as protecting the privacy of data so that it does not get disclosed to an unauthorized entity. Data created from research institutes are valuable resources for scientific and educational purposes in future [5]. Disclosure of data from any research organization may well cause harm to national security. Hence, researchers of almost all the domains are expected to remain proactive during their research, to ensure that the privacy of an individual research subjects are protected and the information about their respective domains remains confidential [5].

Removable or portable USB devices gives user a convenient access to official data. Hence as their use increases so is the security risks associated with them. Portable removable devices increases the risks of data loss, data exposure to and from any system it is connected with and also are prone to theft. It is also found that most of the malwares today are spreading through removable USB devices.

Many research organizations denies the usage of removable media within their respective working premises due to the security risks associated with them. Many open source encryption tools and techniques are available to safeguard data confidentiality. But to do that it is necessary for an individual to have an awareness and good knowledge about computer and security aspects related with it. Sometimes it may also be possible that removable device can be misplaced or lost, hence the data or information within the device is accessible to an unauthorized entity at that time.

In order to mitigate the risks associated with removable media devices this paper summarizes the data encryption techniques for removable media (USB) to help us understand the two methods namely mutual authentication and key match for security of confidential information, Further this paper also discusses about some of the open source cross platform supported tools available for data encryption, performance analysis of various encryption algorithms for the purpose of maintaining the data confidentiality and lastly we will conclude with the understanding of data encryption system that compares the execution time and data security with other existing encryption algorithms.

II. EXISTING DATA ENCRYPTION TECHNIQUES FOR REMOVABLE DEVICES

Removable devices are easy to use as they are usually plugged in to and from a system without any installation required. Today they are the best options for carrying data to and from the system. Data within a research organization is very much sensitive and hence disclosure of such data can cause the security threats to a nation. So most of the research institutes does not allow the usage of removable devices within the campus premises. The two techniques that can be efficient to protect the sensitive data within a removable device are discussed below:

A. Mutual Encryption

In this technique data is stored in the USB device only after it is encrypted. The advantage of doing it is that whenever the removable device (USB) is lost or misplaced, the sensitive data within it is protected from any unauthorized access. Hence the data remains confidential due to encryption [1].

B. Key Match

In this technique RSA algorithm is used in which a public key is generated by an authentication server (AS) and transferred to client. Keys (RSA public key and randomly generated key from AS when user inputs his/her credentials) are used for the encryption and decryption of messages. Keys exchanged between the client and AS might well be vulnerable to hack as receiving entity do not know that the message received is from the legitimate sender or not. Hence password for confirming the identity is required for protecting the sensitive data. Schnorr's digital signature method was used for the overall security of using this technique [1].

C. Control Protocol for USB devices

The main objective of this protocol is to provide security as well as speed to removable devices. This protocol provides user authentication and key exchange mechanism. Initial registration to access the USB device is required once user is authenticated by the AS. A key is send to the user from AS for encrypting the files which is generated every time during the verification process [1].

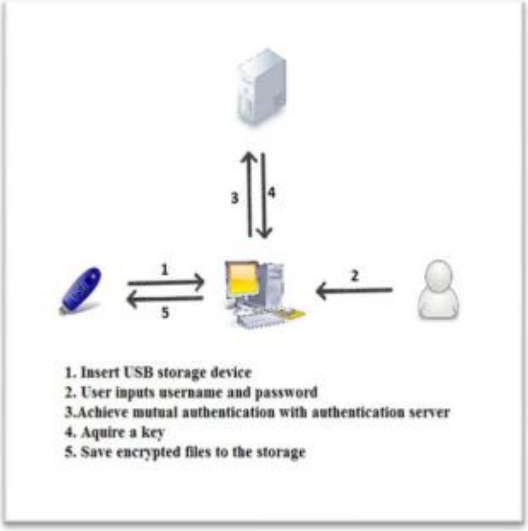
3
4
1
5
2
1. Insert USB storage device
2. User inputs username and password
3.Achieve mutual authentication with authentication server
4. Aquire a key
5. Save encrypted files to the storage

Fig. 1. System Overview [1]

Working of control protocol is described below [1]:

- When a removable device is inserted, the system asks the user to enter the credentials namely username and password for authentication.
- Once the username and password is provided. The credentials are matched with the system and verified. If they are matched then the system treats the user as a valid user, then a session key is generated by the AS and is sent to the user. If the credentials do not match then the system treats the user as an invalid user and restricts the user from accessing the removable device.
- The session key is used to encrypt the files stored in removable USB devices. For decrypting the files the user has to again go through with the same verification process. Every time the user inputs his credentials, a unique session key is generated every time the user logs in.

The above mentioned techniques provide a more efficient and secure transmission of confidential data in and out of a removable media device. The control protocol discussed involves a remote authentication server to verify the authentication of the user and uses RSA algorithm for key match to protect the privacy of the data. It can also provide security against some general attacks that an adversary can implement.

III. SECURITY TOOLS AVAILABLE FOR USB (REMOVABLE MEDIA DEVICES) ENCRYPTION

USB removable devices are easy to carry. But maintaining confidentiality of sensitive data within a removable device is always a risk. If you are a part of any research organization then losing sensitive data can have severe consequences on nation's security and hence it is necessary to maintain the sensitive data confidential. For maintaining confidentiality of sensitive data, following tools or software that are commercially and freely available are discussed below:

A. USB Safeguard

It is a software tool that uses AES-256 encryption for protecting the files within a removable device using a password. It provides the protection of private files as it creates a virtual drive inside the removable device that is password protected. It requires no administrator privileges. No specific installation is required and supports Windows, Mac and Linux systems. It supports file size of atmost 2 GB [5].

B. AESCrypt

AESCrypt is a software tool used for encrypting files. It uses an AES-256 algorithm for file encryption. It is an open source tool available which works on almost all the operating systems namely windows Mac and Linux. In Windows, the user need to right-click on a file, select the desired options available like AES Encrypt or AES Decrypt, then on after entering a password this tool does the desired operation as selected by the user. Similarly if users are using Mac, all they need is to drag a file in an AES program and give the desired password for the respective file. And in in the command line

TABLE I. COMPARISON OF USB ENCRYPTION TOOLS

Tools	Comparison of tools			
	Algorithm Used	***Costing***	***Platforms***	***Limitations***
USB Safeguard	AES	Free up to 2 GB file size	Mac, Linux and Windows'	Max file size of 2GB
AESCrypt	AES	Free	Mac, Linux and Windows'	Max file size 2 GB
TrueCrypt	AES, Serpent or Twofish	Free	Windows	Project already closed and hence not secure
SecurStick	AES	Free	Windows	47 MB

utility, "aescrypt" command is to be executed with filename and password for encrypt and decrypt [9].

C. *TrueCrypt*

TrueCrypt is an open source tool which works on all platforms namely windows, mac and linux. In order to use this tool, the user need to simply plug in removable device and then run the TrueCrypt executable. After that the user need to create a volume for the removable device and after following the instructions properly this tool asks for password to protect user's respective data. TrueCrypt provides AES, Serpent or Twofish encryption algorithm for protecting the data. Data can be recovered by entering the password provided earlier [5].

D. *SecurStick*

SecurStick is another portable removable media device encryption tool that uses AES-256 for securing the sensitive data stored in USB drives and removable media. SecurStick does not require an administrator privilege to use it. It also works in Windows, Linux, and Mac operating systems.

IV. PERFORMANCE ANALYSIS OF ENCRYPTION ALGORITHMS

In this section the performance of various symmetric key algorithms like AES, DES, Blowfish etc. are discussed. The performance of all these algorithms were evaluated on the basis of Architecture, Scalability, Security, Flexibility and limitations. Architecture includes structure, key size, block size and number of processing rounds.

A. *Comparison based on scalability*

Scalability of encryption algorithm is based on the basis of memory usage and performance of encryption (Processing time).

Less Memory usage = More Efficiency,

Hence, less memory usage results in higher efficiency.

Less Processing time = More Encryption rate,

Hence, less processing time results in higher encryption rate. Encryption rate is defined as the processing time required by an encryption algorithm to encrypt data.

Fig. 3. below Shows the comparison of various encyption algorithms based on memory usage and performance of encryptio

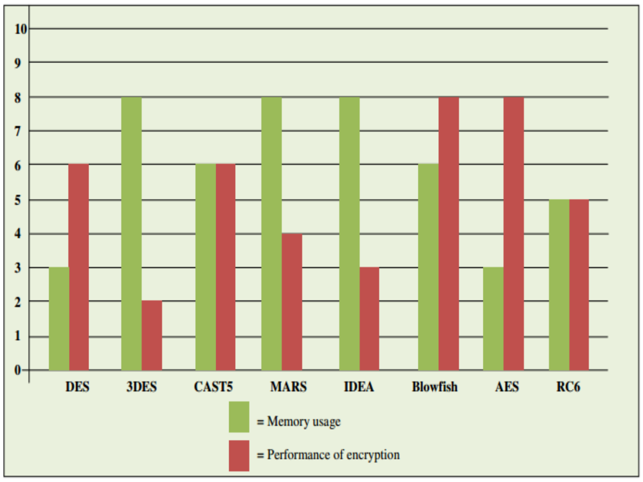

Fig. 2. Comparison of Algorithm based on scalability [4]
(Memory usage and Performance)

It is clear from the above figure that the performance of encryption is nearly equal in both AES and Blowfish algorithms but the memory usage of Blowfish is higher than that of AES, Hence AES is preferreable over Blowfish.

We cannot say that AES is the best among all the algorithms based on the basis of above figure. All the algorithms apart from memory usage and performance must be effective from security point of view as well.

B. *Comparison based on security*

An encryption algorithm or any security system is considered better only if they cope well with the security aspects related with it. Security related features of symmetric encryption algorithms are discussed below.

1) *DES:* The key length of 56 bit is used with DES for the encryption. But today, it can be cracked by brute force attack using 256 combinations. Hence DES is not used for encryption because it is not fully secured [4].
2) *BLOWFISH:* Blowfish uses variable length key of 32-448 bits. In this algorithm each bit of the master key involves multiple round keys and hence independent of another. So we can say that blowfish is a secure algorithm for encryption [4].
3) *3DES*: In 3DES, DES process is performed three times using three different keys for improving the level of security. It uses key size three times larger than DES or simple DES. Hence it is preferred more than DES for encryption [4].
4) *AES:* AES uses variable length keys with key size of 128, 192 or 256 bits. Hence AES provides higher level of security for encryption. Different types of attacks like key attack, differential attack, square attacks were tried to crack this algorithm. But all of these attacks was not able to crack AES. Hence AES is classified as a best and more secured encryption technique [4].

TABLE II. COMPARISON OF SYMMETRIC ENCRYPTION ALGORITHMS

Algorithms	Comparison		
	Key Length (in bits)	**Number of rounds**	**Limitations**
DES	56	16	linear cryptanalysis attacks
Blowfish	32-448	16	Weak keys attacks
3DES	128	16	linear and differential attacks.
AES	128,256,512	10,12,14	Haven't cracked and hence secure.
IDEA	128	8	Brute force attacks
RC6	128,192,256	20	linear cryptanalysis attack
CAST 128	40-128	12 or 16	differential related-key cryptanalysis

5) *IDEA*: IDEA uses 128 bit key size for encryption. It is found that the vulnerabilities in IDEA are less inaccordance with linear and differential attacks. It performs maximum operations for enhancing its security level and hence it provides a strong security against differential attacks [4].

6) *RC6:* RC6 provides security against differential attacks in a better way, RC6 possess the parameter of random series output that provides better protection against the attacks it can experience. In a Linear attack an adversary can apply for 16 rounds of RC6 but it would be impossible to succeed as an adversary has to atleast perform 2^{119} combinations of plaintext [4].

7) *CAST 128:* It uses variable key length operations to increase the level of security. This algorithm provides better security against linear and differential attacks [4].

V. SECURED SYMMETRIC KEY TEXT DATA ENCRYPTION ALGORITHM

Charru, Paramjeet Singh, Shaveta Rani, proposed this algorithm which uses key of variable length that depends on the size of input data that is to be encrypted. The proposed algorithm generates the variable length key from the data which is to be encrypted with the help of some random number. Now if the key size comes out to be four bits then the algorithm takes two characters from the random technique and the remaining two characters from the data that has been provided as an input so that the security of the key is ensured. To further increase the security level exclusive-OR operation is performed on data that is to be encrypted. To decrypt the encrypted data one should be exactly aware about the key characters [2].

A. *Encryption steps*

1) Input the plain text.
2) Calculate the ASCII values for each characters of the plain text.
3) From the input plain text find the minimum ASCII value.
4) Apply the modulus operation on each ASCII value with the minimum value in step 3, and store the results in a content array.
5) Depending on length of input generate a key of variable length.
6) Find ASCII values of the key generated in step 5.
7) Get the minimum ASCII value from step 6.
8) Apply modulus operation on generated key's ASCII values with minimum ASCII value in step 7.
9) Right shift the key one time.
10) Add minimum ASCII value from step 3 to mod key values for obtaining the final key.
11) Add each mod content of data to the final key obtained in step 10.
12) Now from the ASCII values obtained from step 11, Generate encrypted text.
13) Perform XORing on encrypted text to obtain the final cipher text [2].

B. *Decryption steps*

1) Input cipher text.
2) Perform XORing on cipher text.
3) From each character of cipher text, Find minimum ASCII value and store it in minicipher.
4) Get the ASCII values and minimum value of the final key.
5) Calculate the difference of ASCII values of final key and ASCII values of cipher text.
6) To each of the values of difference add minicipher to generate ASCII values of the plain text.
7) Get the plaintext [2].

Below table shows the execution time of encryption for the specified input size and compares the existing algorithms and proposed secure symmetric test data encryption algorithm. **All the simulation has been conducted using visual C# [2].**

TABLE III. EXECUTION TIME (IN MICROSECONDS) COMPARISON [2]

Input Size (in bits)	Encryption with Key Length equal to Input length (Microseconds)	Encryption system with self-generated key of fixed length (Microseconds)	Encryption with variable length key based on input text (Microseconds)
2	3220	849	683
4	3679	926	702
6	3861	1278	1154
8	4748	1349	1240

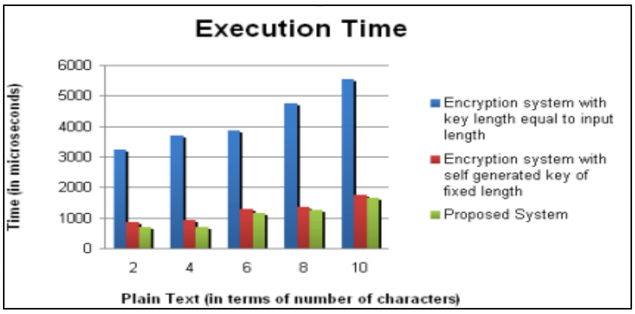

Fig. 3. Graph showing execution time varies with no. of characters [2]

Fig.3. above describes the graph of execution time for encryption of existing algorithms with proposed symmetric text data encryption algorithm.

VI. CONCLUSION AND FUTURE WORK

In a nutshell, a review of different data encryption techniques including a control protocol for data confidentiality of removable device through authentication server is understood, USB device encryption tools (both commercial and open source) for maintaining confidentiality of data and their respective features are analyzed. The limitations associated with the available USB encryption software are also listed in tabular format. Performance of various symmetric encryption algorithms based on the security, scalability etc. are analyzed which helped us in understanding of security features and the respective symmetric encryption algorithms advantages and disadvantages they have against one over other.

On the basis of this analysis it can be concluded that, today AES is more secure for data encryption compared to other symmetric key encryption algorithms available, as it consumes less memory and performance of encryption is fast compared to other symmetric encryption algorithms in existence. We also understood the secured symmetric key encryption algorithm which was used to encrypt the plaintext using the keys generated based on the input.

These algorithms were simulated using visual c# and it was found that the execution time for encryption using variable key length based on the input text was lower in comparison with encrypting the text using fixed key length and with key length same as input text.

Hence based on this review we can say that future work should include a better USB data encryption tool or a software which should be optimal in terms of performance, memory usage and security. Secure text data encryption gives optimal performance for small size text data as discussed. Today, data is growing in huge amount and hence there is a necessity of securing a large size data. Future work should also include a technique that is effective in encrypting large size data like audio, video and image files that should be simulated in all the available platforms like windows, mac and linux with better performance of encryption, strong security level and optimal execution time.

REFERENCES

[1] Shivanku Mahna, Sravan CH, "Data Encryption Techniques for USB," International Journal of Computer Applications, IJCA (0975 – 8887) Volume 104 – No.7, October 2014.

[2] Charru, Paramjeet Singh, Shaveta Rani, "Efficient Text Data Encryption System to Optimize Execution Time and Data Security" International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Volume 4, Issue 7, July 2014.

[3] A.Ramesh, Dr.A.Suruliandi, "Performance Analysis of Encryption Algorithms for Information Security", IEEE Interntional Conference on Circuits, Power and Computing Technologies [ICCPCT-2013].

[4] Md Asif Mushtaque, "Comparative Analysis on Different parameters of Encryption Algorithms for Information Security" International Journal of Computer Sciences and Engineering (ISSN: 2347-2693) Volume-2, Issue-4, April 2014.

[5] 5-best free usb encryption software. [Online] http://www.ilovefreesoftware.com/12/featured/5-best-free-usb-encryption-software.html

[6] Privacy and Confidentiality – current issues in research ethics. [Online] http://ccnmtl.columbia.edu/projects/cire/pac/foundation/

[7] Confidentiality. Wikipedia. [Online] http://en.wikipedia.org/wiki/Confidentiality

[8] USB flash drive security. Wikipedia. [Online] http://en.wikipedia.org/wiki/USB_flash_drive_security

[9] AES Crypt. [Online] https://www.aescrypt.com/

[9] Strength Assessment Of Encryption Algorithms – Whitepaper. [online] http://discretix.com/wpcontent/uploads/2013/02/Strength_Assessment_of_Encryption_Algorithms.pdf