Data Encryption Techniques for USB

Shivanku Mahna
Amity School of Engineering and Technology
Amity University
Sec 125, Noida, UP

Sravan CH
Amity School of Engineering and Technology,
Amity University
Sec 125, Noida, UP

ABSTRACT
Universal Serial Bus (USB) external devices are high speed flash/external drives which can boast data transfer speeds, about as large as 5Gbps. Host controlled USB is the fastest way of transferring material to an external device available today. It is easy, convenient and fast to access these USB devices as compared to other external devices such as CD, DVD, Floppy drive etc. USB is easy to use as they're plug and play devices and do not require any additional installation. The other big advantage of USB is that the data transmission is done at the same speeds, irrespective of the size of the data being copied. Thus USB and USB devices have become most popular interface standard for hardware connection. USB devices however lacks in security as any user can access the USB device and the data associated with it, leaving the data at huge potential risk. Thus to make the system more reliable and fast, we propose methods namely, Mutual Encryption and Key Match in order to provide security to the sensitive information on the device as well as to make the transactions or communication done using USB more encrypted.

General Terms
Authentication, USB, Cryptography

1. INTRODUCTION
USB devices are well known for the ease with which they connect to a system and the speed they provide for fast and smooth access of data. These high speed devices are more efficient than any other external devices in the world. But what it lacks big time is security. There is no authentication process provided by either any pre programmed algorithm in USB or by any pre installed software in USB devices that vouches for a secure transfer of data, thus making the data on these devices extremely unsafe and easy to exploit by a hacker or attacker.

This poses a serious threat to the security of important data stored in the computers. This becomes an even larger area of concern at places like banks where leaking of sensitive information or even one unauthorized transaction can lead to disastrous results.

Hence the need of the hour is to develop a way out so as to restrict unauthorized users or so called hackers from accessing files that contain sensitive data. We have tried to make USB device data safe by using two techniques namely Mutual Encryption and Key Match.

In Mutual Encryption, the data, before being directly stored in the USB is first encrypted and then stored in USB devices. What it does is that it protects our personal and precious data from being accessed by an unknown person in case the device is lost or stolen. The data is thus secure due to encryption.

In Key Match, we have used RSA algorithm. The objective of using this algorithm is that when two sides are trying to communicate, a key is generated by the Authentication server (AS) and transferred to client safely. Subsequently, the keys generated by RSA algorithm can also be used for encrypting the message meant for transmission. However, when the keys are being transferred, an attacker might pose as the sending end or a receiving end and easily hack the sensitive information. Since the receiving end cannot ensure that this message is being sent from the sending end and vice versa, therefore, to make the transaction a safer one, a password for identity confirmation on both receiving and sending end is a must. And for the same, a password verification system is made by us that combines Schnorr's digital signature scheme and the RSA algorithm to improve the overall security of the data while using this method. Below in the paper, information about all these security methods are then followed by the current trends and practices which are being used for providing security to the data. The section following it provides a just about the various cryptographic algorithms. Then the fourth section explains the parameters being used along with the System design and the last section explains Security analysis which tells how secure a system is to general attacks that happen while using a USB device.

2. OVERVIEW OF SYSTEM
Here a control protocol is designed in such way that it provides security as well as speed to USB devices. This protocol implements user authentication along with key exchange agreement.

User authentication is done during the registration phase of the USB device by providing a username and password to the user. First time registration is required by every user before accessing the USB devices. The user must remember the username and password to access the USB device.

When a USB device is connected to the system, the system checks whether the username and password entered match to the system or not. If the username and password does match to the system then he/she is allowed access to the USB device. The USB device encrypts each and every file using a key which is generated every time in verification process. To access the files on the USB device, the user has to acquire the same session key to read/write these files. A unique session key is generated for each file based on username, password and the private key of AS.

This system is very secure as only the users with valid authentication and key can access the USB devices. The user does not have to worry even if the USB device is lost as all the files stored in it are encrypted using a key and thus they cannot be decrypted without the valid key. If the user wants to distribute files to other people malevolently, the file is still secure as the user cannot receive the agreement key to decrypt the files.

The functioning of the system is described as follows

1. When a user inserts a USB device, the system forces the user to enter username and password for authentication to access the USB device
2. After the username and password is entered. The details are matched with that of the system and verified. If the details match then the system treats the user as valid user and a session key is provided by the authentication server. If the details do not match with that of the system then the system treats the user as invalid user and restricts the user any permission to access the USB device.
3. The session key provided by the authentication server is user to encrypt the files stored in USB devices. If the user wishes to decrypt these files the user has to pass through the above mentioned verification process again.

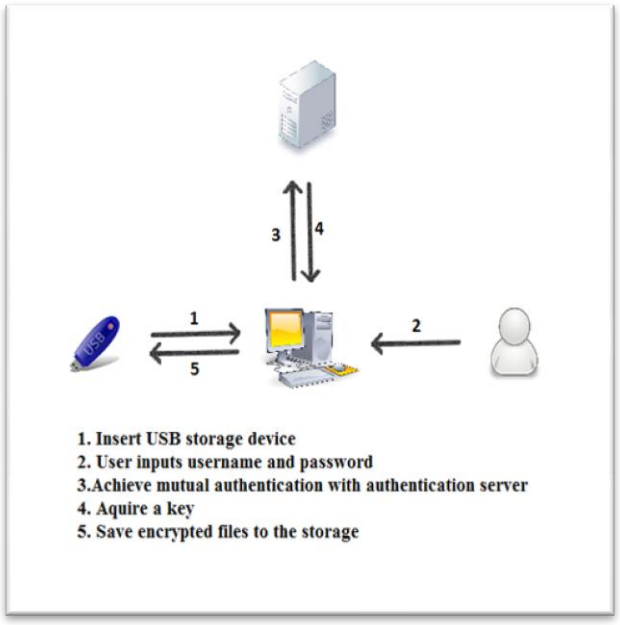

Figure 1:- System Overview

3. CRYPTOGRAPHIC ALGORITHMS

A USB device uses Encryption algorithm to encrypt/decrypt the files before storing them on the device. Doing this helps in securing the sensitive and private data in case the device is stolen or lost. There are two types of encryption algorithms used in cryptography. They are symmetric and asymmetric algorithms.

AES, DES, IDEA, Triple-DES etc are some of the examples of Symmetric encryption algorithms whereas Asymmetric encryption algorithms have Diffie- Hellman, RSA, DSA etc as its examples. But both these types do have some short coming or the other in them .Hence the algorithms formed by the combination of Symmetric and Asymmetric algorithms in order to remove the short comings of either of the algorithms is known as Hybrid algorithms. [3]

To improve data security in USB, RSA algorithm(a type of asymmetric algorithm) will be used as it will help in providing a secure key exchange in mutual encrypton process recommended in the paper and fo the key match process , we will be using Schnorr's Digital Signature scheme. The reason behind choosing RSA algorithm and the difference between each of the asymmetric algorithms is given below in a tabular form [1] :-

- *RSA Algorithm :-*

1. Take two different prime numbers a and b.

2. Calculate $x = ab$.

(x is used as modulus here.)

3. Calculate $\varphi(x) = (a - 1)(b - 1)$.

(φ is Euler's totient function)

4. Choose an integer w such that $1 < w < \varphi(x)$ and w and $\varphi(x)$ are co-prime.

(w is released as the public key exponent)

5. Given (d*w)mod $\varphi(x) = 1$, calculate $d = w^{-1} \bmod \varphi(x)$.

(d is kept as the private key exponent)

6. The public key consists of the modulus x and the encryption exponent w whereas the private key consists of the modulus x and the decryption exponent d which must be kept secret.

- *For Encryption :-*
  1. Let message P is to be transmitted.
  2. Convert P into an integer p, such that $0 < p < x$ by padding scheme.
  3. Calculate the cipher text c by using formula

     $c = m^w \pmod{x}$.
  4. Then it transmits the message or text c.

- *For Decryption :-*

1. Recover p from c by using private key exponent d by calculating

$$p = c^d \pmod{x}.$$

2. Now since we have the value of p, we can recover the original message P by reversing the padding scheme.[2]

CHARACTERISTIC	DIFFIE-HELLMAN	RSA	DSA
Proposed/Given By	Whitfield Diffie and Martin Hellman	Rivest, Shamir and Adleman	NIST
Key generation speed and verification speed	Fast in Key Generation and slow in verification	Slow in Key Generation and fast in verification	Fast in Key Generation and very slow in verification
Primarily used for	Key Generation and Encryption/ Decryption	Key Generation and Encryption/ Decryption	Key Generation

B. Digital Signature Scheme :-

Schnorr's signature scheme is used to limit the number of signatures. The scheme employs a subgroup of order b in Z*a, where a is a prime number. It also requires a hash function H such that :-

H: {0, 1}* −> Zb.

- *Key Generation Algorithm*

1. Select prime numbers a and b such that b divides (a − 1).

2. Select a random integer r such that :- 1 ≤ r ≤ b − 1.

3. Compute y = gr mod a.

4. A's public key is (a, b, α, y), and A's secret key is r.

- *Signature Algorithm*

1. Select a random secret integer L such that, 1 ≤ L ≤ b – 1

2. Compute r = gL mod p, x = H(p‖r), and s = x · w + k mod b

3. A's signature for p is the pair (s, w).

- *Signature Verification*

1. Compute v = gs · y − w mod a, and w' =H(a||v)

2. Accept the signature if and only if w = w'. [4]

4. SYSTEM DESIGN

Registration is required to access the USB device. After the user has registered, to encrypt or to decrypt any file the user has to go through verification and data encryption phase through which a session key is generated

A. Parameters and Symbols

1. a, *b* : Two large primes a and b, where b | a-1.

2. *g*, *G* : *g* is an element chosen from Z*a and having an order of b; *G* is the cyclic group generated by *g*.

3. *uid*, *pwd* : User account (user name) and password.

4. *x*, *Y*: Server's private key and public key; *Y* = *gx* mod *a.*

5. *h*(.), *H*(.) : One way collision-resistant hash functions; *h*(.) maps arbitrarily long strings to strings of fixed length, and *H*(.) maps to elements of the cyclic group *G*.

6. ||: Concatenate operate.

7. *Fn* : Filename for encryption.

8. *File* : File for encryption.

9. *EK*[.] : Symmetric encryption function with respect to a key *K*.

10. *DK*[.] : Symmetric decryption function with respect to a key *K*.

B. Registration Phase

Registration is required to access a USB device. During the registration phase, when the user inserts the USB device, one set of username (uid) and password are chosen by the user. To calculate hpwd = H(pwd), pwd should be substituted into a hash function. Then uid and hpwd is sent to the authentication server. When the authentication server receives this message, a random number k is chosen by the server by which *r = hpwd k* mod a and *r*1 = *g k* mod *b*. Then the value of *e* = *h*(u*id* || *r* || *r*1) is computed, the private key x is used to calculate *s* = (*k* - *e* . *x*) mod b, and save (*e*, *s*) to the user's storage device. When the user receives the triplet (*e*, *r*, *s*), the value of e is checked by the user if e is equal to *h*(u*id* || *r* || *g s y e* mod b). If it is valid then registration is complete. The data transmission during this phase is done through a secure channel. To prevent Brute force attack by guessing of the password, the user is forced to select a password with high complexity.

After the registration phase is completed, whenever the user wishes to access the USB device, the user needs to enter valid username and password to achieve mutual authentication with the authentication server, the authentication server then generates an encryption key. The detailed procedure is described as below

Step 1:

The user inserts the USB storage device. The user is then requested to enter valid username and password. The pwd is substituted into a hash function to calculate hpwd. Then the user is allowed to choose large random prime integers ak and bk, the modulus n=ak.bk. Then Euler's totient function Φ(n)=(ak-1)(bk-1) is also calculated. Choose a Public key exponent ek that is co-prime with Φ(n) such that 1<ek<Φ(n). Calculate u = hpwd . yn mod a. Then the messages of {uid, Fn, ek, n, u, e, s} are sent to the authentication server by the user.

Step 2:

After authentication server receives the messages {uid, Fn, ek, n, u, e, s}, it then uses its long term private key x and calculates hpwd = u/(y n mod a) and k = s + e . x mod b. The authentication server then validates to verifiy whether e = h(uid || hpwdk mod a || g s y e mod a). If they are equal then the user is termed legal user. If not, the session is terminated and he is not allowed any access rights. After which the authentication server uses the filename Fn as well as long term private key x to calculate m = h(x || Fn). Here m is encrypted to generate cipher text c= mek mod n. Finally, message authentication code (MAC) is calculated by the authentication sever

MAC = h(uid || hpwd || m || ek) and thus the user receives the generated message {c, MAC}.

Step 3:

When the user receives the generated message {c, MAC}, public key exponent ek and Φ(n) are used to calculate and generate private key exponent dk. C is decrypted using d to retrieve m, which is used to generate a session key sk, Then MAC = h(uid || hpwd || m || ek) is verified. If they are equal, then the user is valid and a mutual authentication is achieved between the user and the authentication server, and the user will calculate a = h(uid || hpwd || m) to generate an encryption key sk using the equation sk = (Y) c = g x . c mod a.

Step 4:

After achieving mutual authentication, the session key sk is calculated by sk = gx . c mod a. Whenever a user wishes to access the USB storage device this encryption key is to be used to encrypt the file, i.e., as E sk [File], this will encrypt the file and does not allow any access to the file without proper authentication. To decrypt the file, the user should undergo the same steps to obtain sk (D sk [E sk [File]]).

5. SECURITY ANALYSIS

System Analysis means determining whether the project is economically, socially, technologically and organizationally feasible.

- *Correctness*

The protocol suggested in this paper will prevent loss or theft of any private, confidential or sensitive data from the USB device. The design suggested makes file transfer via USB interface a safer experience by blocking the file transfer till the user does not pass through the suggested authentication procedure. If the device is being used by a valid user, then the required files are transferred to peripheral device in encrypted format. The key used for encryption is formed by using Username, Password and filename. If the user wants to read that file, first he has to decrypt it and only then the user can access the data. For decryption, user has to go through the same authentication procedure and have to obtain same keys which were used during the encryption process.

- *Offline password guessing*

If the USB is lost or stolen, the confidential data stored in it is secured as to access the data, the user is required to decrypt the data by entering the username and password. Even if the user tries to guess the password, it will be an extremely difficult task as the id and password is based on solving Discrete Logarithmic Problems [5].

- *Discrete Logarithmic problem*

1. During the data encryption and verification phase, even to guess the value of parameters, the hacker has to pass through the Discrete Logarithmic Problems which makes even guessing the parameters value a very hard task.

2. Discrete Logarithmic problem are the ones where the variables have n number of solutions.

3. For Ex: X%2=1; to get answer as 1 ,X can have any value ranging from X=3,5,7,9,11 ...etc

A Session Key is generated for each of the verification messages in the suggested protocol. Without knowing the values of the prime numbers selected- pk and qk and the value of the private key x, hacker cannot decrypt the file. So the protocol resists offline password attacks pretty successfully as well. [6].

- *Replay attack and Stolen verifier attack*

If hacker tries to use a phishing method and traps the user by showing a false login message and tries to get some information about the parameters being used.But the hacker doesnt know the values of the two prime numbers selected, i.e, pk and qk. Therefore the hacker doesnt know how p is used to calculate session key sk.So, even if he finds the session key, the password is still required, therefore making the protocol a success against the stolen verifier attacks as well [7].

6. CONCLUSIONS

The Proposed methods and ways provide a more secure and efficient transfer of personal data in and out of a USB device. The protocol employs a remote authentication server to verify the authentication of the user and uses asymmetric cryptographic RSA algorithm to implement key matching to protect the privacy of the data being transmitted into a storage device. The above mentioned security system can also resist some general attacks that a hacker might do, thus making it a really reliable way of data encryption.

7. ACKNOWLEDGMENT

We would like to express our gratitude towards a number of people whose unconditional support and considerations have been an invaluable asset during the course of this research work.

7. REFERENCES

[1] Gustavus J. Simmons,1979. “Symmetric and Asymmetric Encryption” in Computing Surveys, Vol. 11, No. 4, December 1979 edition.

[2] E. V. Vetvitskii, A. V. Plotnikov, D. A. Prilutskii, S. V. Selishchev, “Use of the USB Universal Serial Bus in computer systems”, ,Springer online journal, July–August edition, 2000, Volume 34, Issue 4, pp 167-172

[3] W. Diffie and M. Hellman,1976. “New directions in cryptography”, IEEE Transactions on Information Theory, Vol. 22, No. 6, pp. 644-654,1976.

[4] Chul-Joon Choi, Zeen Kim and Kwangjo Kim “Schnorr Signature Scheme with Restricted Signing Capability and Its Application”.

[5] Mrs. C. Shoba Bindu, Dr P. Chandra Sekhar Reddy and Dr B.Satyanarayana, “Improved Remote User Authentication Scheme Preserving User Anonymity”, International Journal of Computer Science and Network Security, VOL.8 No.3, March 2008.

[6] T. A. El Gamal, “A public-key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Transactions on Information Theory, Vol. 31, No. 4, pp. 469-472, 1985.

[7] Lu Zhu, Sheng Yu and Xing Zhang, 2008. “Improvement Upon Mutual Password Authentication Scheme”, 978-0-7695-3560-9/08 IEEE DOI 10.1109/ISBIM, 2008.