A CONVENTIONAL KEY GENERATION FOR FILE ENCRYPTION METHOD AND PROTECTION USING UNIVERSAL SERIAL BUS (USB) STORAGE DEVICE

Manoj Prajapat[1], Anurag Maloo[2]

[1]M. tech Pursuing, [2]Assistant Professor, Institute of Technology & Management, Bhilwara (India)

ABSTRACT

USB File Lock Using USB Storage Device and Digital Keys (Signature) is a high performance File encryption / protection program, password securing your file against outside unauthorized access by the use of an USB stick/Dongle. You should provide facility to send password on sms and even email so that authorized person can use for decrypt a file. You should provide facility to encrypt all type of files without restrictions. Password encryption process should be there so no one can access password specified directly from the encrypted file.

Most of the file bulwark software's provide either one level bulwark of the simple encryption method or two level protections which is password bulwark and encryption utilizing simple key generation algorithm which key is engendered by the software by utilizing characters of password given by the users.

Keywords: Encryption, Decryption, RSA, Digital Signature

I. INTRODUCTION

Nowadays, the most comfortable removable astronomically immense-capacity data contrivances are connected to the system via the bus Universal Serial Bus (USB). Such contrivances include flash recollection RAM with a capacity of several tens of GB and hard disk drives with a capacity of several TB. The popularity of these contrivances forces the desideratum for mechanisms to ascertain an adequate level of bulwark of data stored on them.

In such a software or applications when once password cracked or reverse engineering is done by simple entering password one can have access the file or decrypt the file. My research work will provide one more level bulwark for such a quandary which engenders Symmetry key utilizing USB storage contrivance to encrypt file. This research work will carry out not only bulwark two level protections but withal provide extra third level bulwark to bulwark file utilizing USB storage contrivance.We are orchestrating to utilize an USB Storage to store the encryption key and additionally use three other verification keys stored inside Storage plus a unique hardware serial number of the Storage. This makes it very safe and can even be utilized in military communication. Advantage of USB Storage is that it’s facilely available at plausible rates. So no special hardware is required for this.

II. SYSTEM REVIEW

- Generates a unique key.
- Encrypt pristine file.
- Generate digital keys desultorily.
- Denies access to unauthorized users or invalid Storage.
- Preserves encryption status even after reboot.
- Facility to upload file.
- Facility to decrypt file via SMS on any mobile contrivance.
- Compress file
- Encryption/decryption

There are two phases first one is Registration phase and second one is verification and key generation phase. In this proposed system we are using RSA algorithm, which was developed by Rivest, Shamir and Adleman in 1977, for key exchange agreement and Digital Signature proposed by schnorr in 1989.

III. CRYPTOGRAPHIC ALGORITHMS

In universal, there are two types of encryption algorithms utilized in cryptography. They are symmetric and asymmetric algorithms.

The distinguishment between Symmetric and Asymmetric algorithms is listed below in Table 1.[2]

Table 1: Distinguishment Between Symmetric and Asymmetric Cryptography

CHARACTERISTIC	SYMMETRIC KEY CRYPTOGRAPHY	ASYMMETRIC KEY CRYPTOGRAPHY
Key Used	Public	Public and Private
Speed	Very Fast	Slow
Size of Resulting Cipher text	Same/Less than Plain text	More than Plain text
Key Agreement	Big Problem	No problem at all

The differences among sundry asymmetric algorithms are given in Table 2. [2]

Table 2: Comparison Among Asymmetric Algorithms

CHARACTERISTIC	DIFFIE-HELLMAN	RSA	DSA
Proposed By	Whitfield Diffie and Martin Hellman	Rivest, Shamir and Adleman	NIST
Speed	Fast in Key production and slow in confirmation	Slow in Key production and fast in confirmation	Fast in Key production and very slow in confirmation
Primarily used for	Key creation and Encryption/ Decryption	Key creation and Encryption/ Decryption	Key creation

3.1 RSA Algorithm

- Key production

1. Select two distinct large massive prime numbers p and q.
2. Compute n = pxq where n is utilized as modulus.
3. Compute $\phi(n) = (p - 1)x(q - 1)$, where ϕ is Euler's totient function.
4. Select an integer e such that $1 < e < \phi(n)$ and e and $\phi(n)$ are co-prime. $Gcd(e, \phi(n))=1$.

- e is relinquished as the public key exponent.

5. Verify $d = e^{-1} \bmod \phi(n)$ i.e. calculate d given $(d*e) \bmod \phi(n) = 1$.

- d is kept back as the private key exponent.

6. The public key consists of the modulus n and the public (or encryption) exponent e. The confidential key consists of the modulus n and the confidential (or decryption) example d which must be kept secret.

- Encryption
  - ➢ Message M is to be transmitted
  - ➢ Change M into an integer m, such that $0 < m < n$ by padding system then compute the cipher text c matching to $c = m^e \pmod n$. Then transmit c.
- Decryption
  - ➢ Recover m from c by utilizing private key exponent d via computing $m = c^d \pmod n$.
  - ➢ Given m, we can instauration the perfect message M by inverting the padding scheme.

3.2 Digital Signature

Schnorr digital signature scheme is utilized to reduce the number of signatures. For common verification, we will be utilize Digital Signature proposed by Schnorr in 1989.

- Key Generation Algorithm
  1. Select two prime numbers p and q.
  2. Select a random integer g such that $g^q = 1 \bmod p$.
  3. (g,p,q) are global parameters to all.
  4. Select a random integer x such that

     $1 \leq x \leq q - 1$
  5. Calculate $y = g^{-x} \bmod q$.
  6. Sender's public key is (p, q, g, y), and Sender's secret key is x.
- Signature Algorithm
  1. select an random secret integer k,

     $1 \leq k \leq q - 1$
  2. Compute $r = g^k \bmod p$, $e = H(m||r)$, and $s = k + x.e \bmod q$
  3. Sender's signature for m is the pair (s, e).
- Signature Verification
  1. Compute $v = g^s \cdot y^e \bmod p$, and $\bar{e} = H(m||v)$
  2. Allow the signature if and only if $e = \bar{e}$ [3][4].

IV. PROPOSED SYSYTEM DESIGN

Utilizer has to first register to the scheme. After registeration phase, when utilizer connects USB set-up and have to go throw confirmation and information encryption stage where session key will be engender which is utilized to encrypt/decrypt file.

4.1 Registration-Phase

- Parameters and Symbols:
  1. Select two prime numbers p and q
  2. Select a random integer g such that $g^q = 1 \bmod p$
  3. (g,p,q) are global parameters to all.
- Signature Algorithm (at AS side)
  1. Select a random integer x such that

     $1 \leq x \leq q - 1$
  2. Calculate $y = g^{-x} \bmod q$.
     a. y is AS's public key and x is its private key.

- ➢ AS generate signature by:-

1. Select an random secret integer k,

   $1 \leq k \leq q - 1$

2. Compute r = hpw^k mod p, r1=g^k mod p,

   e= H(USN||r||r1), and s = k + x.e mod q.

3. Send (e,r,s) to the client.

- Signature Verification (at client side)

1. Compute r1'= $g^s \cdot y^e$ mod p
2. Allow the signature if and only if e =(USN||r||r1') [3][4]

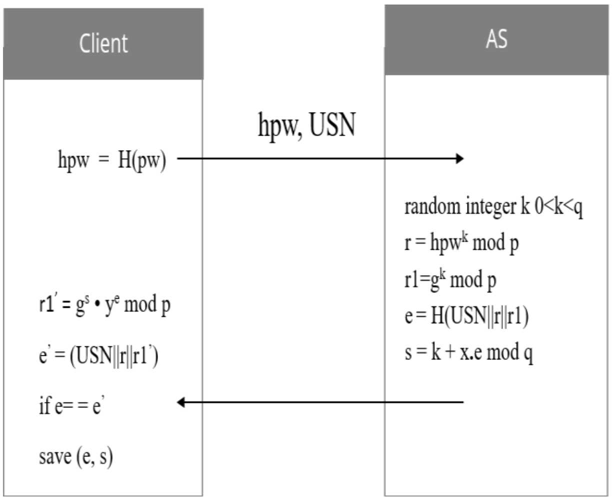

Figure 1: Registration Phase

4.2 Verification and Key Generation Phase

- Parameters and Symbols:

4. p, q : Two hugely massive primes p and q, where q | p-1
5. g : g is an integer such that g^q = 1 mod p
6. USN, pw : USB Serial Number and password
7. x, y: Server's private key and public key; y = g^{-x} mod p
8. h(.), H(.) : One way collision-resistant hash functions; h(.) maps randomly extended strings to strings of fine-tune length, and H(.) maps to elements of the cyclic group G.
9. ||: Concatenate operate
10. Fn : Filename for encryption
11. File : File for encryption
12. E_k[.] : Symmetric encryption purpose with respect to a key K
13. D_k[.] : Symmetric decryption purpose with respect to a key K.

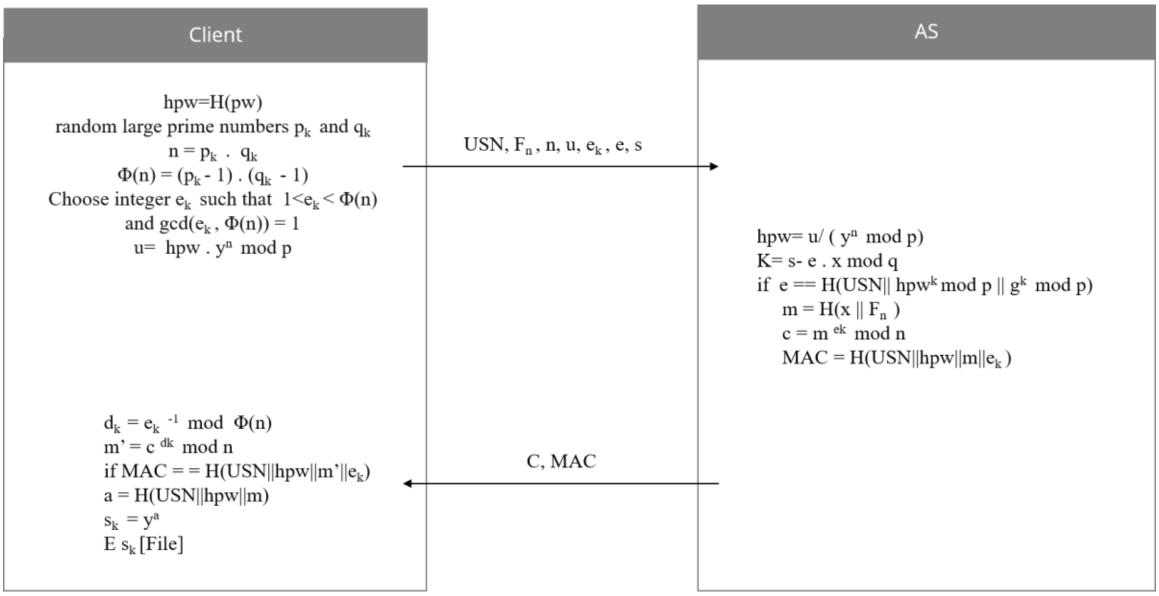
Client
AS
hpw=H(pw)
random large prime numbers p_k and q_k
$n = p_k \cdot q_k$
$\Phi(n) = (p_k - 1) \cdot (q_k - 1)$
Choose integer e_k such that $1<e_k<\Phi(n)$
and $\gcd(e_k, \Phi(n)) = 1$
$u = hpw \cdot y^n \bmod p$
USN, F_n, n, u, e_k, e, s
hpw= u/ (y^n mod p)
K= s- e . x mod q
if e == H(USN|| hpw^k mod p || g^k mod p)
m = H(x || F_n)
c = m^{ek} mod n
MAC = H(USN||hpw||m||e_k)
C, MAC
$d_k = e_k^{-1} \bmod \Phi(n)$
$m' = c^{dk} \bmod n$
if MAC = = H(USN||hpw||m'||e_k)
a = H(USN||hpw||m)
$s_k = y^a$
E s_k [File]

Figure 2: Verification and Key generation phase

After consummate the register stage, and when access the USB storage space set-up, the utilize wants to get shared verification with the verification server utilize the USN and pwd, which engender an encryption key. The announcement process is described in feature below. Figure 2 shows the full procedure of verification and key generation phase.

Step 1

The utilizer affix the USB storage space set-up to a computer from side to side an ordinary process and input the right pw. At this instant, the utilizer (client) will utilize pw to compute hpw utilizing the one way hash function H(.). Then the utilizer culls a random hugely massive prime information p_k and q_k and compute modulus $n=p_k.q_k$. Euler's totient purpose Φ(n)=(pk-1)(qk-1) is withal calculated. Public key exponent ek co-prime with Φ(n) is culled such that 1<ek< Φ(n). Calculate u = hpw . y^n mod p. Determinately, the utilizer will send mail of {USN, Fn, ek, n, u, e, s} to the Authentication Server.

Step 2

After getting {USN, Fn, ek, n, u, e, s}, the AS will use its long word confidential key x to calculate hpw = u / (y^n mod p) and k = s - e .x mod q. Then, the confirmation server will employ parameter it engender to validate whether e = H(USN || hpw^k mod p || g^k mod p). If assenting, then the utilizer in this message is licit. If not, the statement is ended. then, the confirmation server will use the conventional file name Fn and the lasting private key x to calculate m = H(x || Fn), and carry out encryption on m, to engender ciphertext c= $m^e{}_k$ mod n. Determinately, the confirmation server calculate a message confirmation code MAC = h(USN || hpw || m || ek) and sends the engendered communication {c, MAC} to the utilizer.

Step 3

After receiving the message {c, MAC}, the utilizer utilizes the public key exponent e_k and Φ(n) to compute confidential key exponent d_k. m is retrieved by decrypting c utilizing d_k to compute a utilized in production of assembly key s_k, Next the utilizer will verify whether MAC = h(USN || hpw || m || ek). If confirmatory, then common confirmation is achieve among the utilizer and the confirmation server, and the utilizer will compute a = h(USN || hpw || m) and produce an encryption key s_k utilizing the equation $s_k = y^a = g^{-xa}$ mod p.

Step 4

After the utilizer and the confirmation server complete these steps, the assembly key sk can be planned by sk = g^{-xa} mod p. When a utilizer needs to right to use the storage set-up via the USB interface, this encryption key, can be acclimate to encrypt the File, i.e., as E_{sk} [File], to for fend the file and provide private and secure access to the USB set-up. For file decryption, the utilizer wants to endure the similar confirmation steps and get the same key sk to decrypt the file (D_{sk} [E_{sk} [File]]) when accessing it on the USB contrivance.

V. SECURITY ANALYSIS

Organization Analysis denotes determining whether the scheme is inexpensively, extrovertly, technically and directorially feasible.

- Correctness

Our procedure will turn away any secret file loss via USB concept storage space set-up. In our protocol design file move via USB boundary is fruitless till the utilizer does not pass through confirmation process. If the utilizer is applicable then the necessary files are transfer to unimportant set-up (USB) in encrypted arrangement. The key utilize for encryption is compute utilize Username, key and filename. After encryption if utilizer desire to read that folder he has to first decrypt it. For decryption, utilizer has to go from side to side same confirmation process and have to find same key utilize for encryption.

Offline password conjecturing

If the USB is disoriented or purloin, yet USB right of entry is controlled as for decryption, username and password is necessary. Thus obviating secret data stored in USB set-up. If utilizer endeavors to inference code word, it will be hard to him as it include solve separate Logarithmic dilemma [6].

- Discrete Logarithmic quandary

1. In confirmation and information encryption stage if attacker actions to guess the worth of limitation for that he has to bypass from end to end separate Logarithmic predicament.
2. Discrete Logarithmic predicament where changeable have number of solution
3. eg: X%2=1; to get reply as 1 ,X having number of value(X=3,5,7, ...)

Session Key is engender for each validation communication in our procedure. Without kenning pk and qk and confidential key x, assailer cannot decrypt the file. So our procedure resists offline password attack [7].

- Replay attack and Purloined verifier attack

If assailer events to use a capture wiretap validate communication and he obtain some limit but he don't ken pk and qk, denote he don't ken m used to work out session key sk. Albeit he finds sitting key still code word is required, so our procedure can oppose the glommed verifier attack [8].

VI. CONCLUSION

The proposed system provides three level of protection by generating a symmetric key. If unauthorized person get encrypted file then he must know password and USB serial number. However it is difficult to guess the password but its near about impossible to guess the USB serial number, as each USB has a unique hardware serial number. That’s why this system enhancing the security level.

REFERENCES

[1] Z. Zieliński et al., "Secured workstation to process the data of different classificationlevels", J. Telecom. Inform. Technol., no. 3, pp. 5–12,**2012**.

[2] A. Kozakiewicz, A. Felkner, J. Furtak, Z. Zieliński, M. Brudka, and M. Małowidzki, "Secure workstation for special applications", in Secure and Trust Computing, DataManagement, and Applications, C. Lee, J.-M. Seigneur, J. J. Park, and R. R. Wagner,Eds., Com- munications in Computer and Information Science, vol. 187. Berlin: Springer, **2011**, pp. 174–181.

[3] J. Chudzikiewicz, "Zabezpieczenie danych przechowywanych na dyskach zewnętrznych", in Metody wytwarzania i zastosowania sys- temów czasurzeczywistego, L. Trybus and S. Samolej, Eds.Warszawa: Wydawnictwo Komunikacjii Łączności, **2010**, pp. 211–221 (in Polish).

[4] [Imperva 2010] Imperva. Consumer password worst practices. http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf, **2010**

[5] Hyun Sook Rhee, Jeong Ok Kwon, and Dong Hoon Lee, "A remote userauthentication scheme without using smart cards", Computer Standards & Interfaces,Vol. 31, No. 1, pp. 4-13, **2009**.

[6] Technical Documentation"Microsoft Windows Driver Kit (WDK)", Microsoft Corporation, Redmond, **2009**.

[7] [Geambasu 2009] Roxana Geambasu, Tadayoshi Kohno, Amit Levy, and Henry M. Levy. Vanish: Increasing data privacy with self-destructing data. In Proceedings of the 18th USENIX Security Symposium, **2009**.

[8] Technical Documentation "Microsoft Windows Driver Kit (WDK)", Microsoft Corporation, Redmond, **2009**.

[9] EncFS 2008] EncFS. EncFS encrypted file system. http://www. arg0.net/encfs, **2008**.

[10] [Halderman 2008] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: Cold boot attacks on encryption keys. In Proceedings of the 17th USENIX Security Symposium, **2008**.