A Symmetric Key Generation for File Encryption and Protection using/by USB Storage Device

Kajal K. Isamaliya[1] Mitesh R. Patel[2] Ankur P. Desai[3] Pankaj Singh Parihar[4]
[1, 2, 4] M.Tech. [3]Assistant Professor
[1, 2, 4] Computer Science Eng. Department, ITM College, Bhailwara, Rajasthan,India.
[3]Electrical Eng. Department, Shri Sitrambhai Naranji Patel Institute of Technology

Abstract-- Most of the file protection softwares provides either one level protection of the simple encryption method or two level protection which is password protection and encryption using simple key generation algorithm which key is generated by the softwares by using characters of password given by the users. In such a softwares 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 protection for such a problem which generate Symmetry key using USB storage device to encrypt file. This research work will carry out not only protection two level protection but also provide extra third level protection to protect file using USB storage device. 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/Storage. You should provide facility to send password on sms and even email so that authorised 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. This provides you security for your data and private information when your data is in transit or in the email as attachment. Any person getting hold of your files will not be able to use them unless he/she has the USB Storage from the same set. We are planning to use an USB Storage to store the encryption key and also uses 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 used in military communication. Advantage of USB Storage is that its easily available at reasonable rates. SO no special hardware is required for this. A storage or memory USB Storage, also called a memory stick, provides a convenient means to pass files between computers or devices. The memory stick contains a rewritable solid-state memory chip that does not require power to retain its contents. As capacities have grown and price has dropped, these portable, plug-and-play storage drives have replaced floppy disks and even writable discs for exchanging files and archiving data.

I. INTRODUCTION

A. *What is Cryptography?*

Cryptography is one of the most complex aspects used by a software developer. Using cryptographic algorithm it requires a high level of mathematical knowledge. Fortunately, with Microsoft .NET, newly created classes wrap up these sophisticated algorithms into fairly easy-to-use properties and methods this paper gives you an overview of the cryptography support that is provided by the .NET Framework.

However Following jargons give you knowledge about Cryptography:

1) Data that can be read and understood with its original form is called 'plaintext' or 'cleartext'.
2) The method of disguising plaintext in such a way as to hide its meaning is called 'Encryption'.
3) Encrypting plaintext results is now in unreadable form of data called 'Ciphertext'. You use encryption to make the information secure and hidden from anyone for whom it is not intended, even those who can see the encrypted data.
4) The process of reversing Ciphertext to its original plaintext is called 'Decryption'.
5) And finally 'key' is a string of bits used for encrypt and decrypt the information to be transmitted. It is a randomly generated set of numbers/ characters that is used to encrypt/decrypt information.

B. *Types of Cryptography*

1) *Private Key Encryption*

Private Key encryption, also referred to as conventional or symmetric or single-key encryption was the only available option prior to the advent of Public Key encryption in 1976. This form of encryption was used by emperors like Julius Caesar and other military organizations to convey secret messages. This key requires all communicating parties, to share a common key. With private-key encryption, you encrypt a secret message using a key that only you know. To decrypt the message, you need to use the same key. Private-key cryptography is effective only if the key can be kept secret. Despite the potential weakness of private-key encryption, it is very easy to implement and computationally doesn't consume excessive resources.

2) *Public-key encryption*

Public key encryption algorithms are based on the premise that each sender and recipient has a private key, known only to him/her and a public key, which can be known by anyone. Each encryption/decryption process requires at least one public key and one private key. Each is related to the other mathematically, such that messages encrypted with the public key can only be decrypted with the corresponding private key.

3) *.NET and Cryptography*

.NET provides a set of cryptographic objects, supporting well-known algorithms and common uses including hashing, encryption, and generating digital signatures. These objects are designed in a manner that facilitates the incorporation of these basic capabilities into more complex operations, such as signing and encrypting a document. Cryptographic objects are used by .NET to support internal services, but are also available to developers who need cryptographic support. The .NET Framework provides implementations of many such standard cryptographic algorithms and objects. Similar to the ready availability of simple authentication features within the .NET Framework, cryptographic primitives are also easily accessible to developers via stream-based managed code libraries for encryption, digital signatures, hashing, and random number generation.

The System, Security. Cryptography namespace in the .NET Framework provides these cryptographic services.

C. The Algorithm support includes:

1) RSA and DSA public key (asymmetric) encryption

Asymmetric algorithms operate on fixed buffers. They use a public-key algorithm for encryption/decryption. An example for asymmetric algorithms is the RSA algorithm which is so named after its three inventors Rivest, Shamir, and Adleman. It is a popular public-key algorithm - the de facto standard - for digital signatures and can be used for encryption as well. The DSA_CSP is an implementation of the digital signature algorithm (DSA). This is a public-key algorithm. It can be used to create and verify a digital signature.

DES, TripleDES, and RC2 private key (symmetric) encryption-symmetric algorithms are used to modify variable length buffers and perform one operation for periodical data input. They use a single secret key to encrypt and decrypt data. The Data Encryption Standard (DES) is a world-wide standard for data encryption. It is the most popular encryption algorithm. It is implemented by the DES_CSP class. This class represents a stream where you pour in data that is encrypted/decrypted using a single key. The Triple DES encryption algorithm operates on a block of data three times using one key. RC2 stands for Rivest Cipher.

2) MD5 and SHA1 hashing

MD5 - Message Digest 5-is a one-way hash algorithm. Given variable length data as input it always produces a 128-bit hash value. The Secure Hash Algorithm (SHA) also is a one-way hash algorithm that produces a 160-bit hash value, which is longer than the MD5 produced hash value.

In this paper we using RSA Algorithm for encryption and decryption.

II. BASIC THEORY: HOW THIS WORKS?

1) Generates a Unique protection-key for each USB stick / Storage
2) Encrypt Original file for first level protection , Encryption is based on customized algorithm so that its notpossible to decrypt without knowing the keys used in Encryption process.
3) Generate digital signatures (Master key and Subkey) randomly or as per Users selection and store digital signatures in encrypted form.
4) Denies access to unauthorized users or invalid Storage
5) Preserves Encryption status even after reboot
6) Facility to upload file on FTP server using FTP protocol
7) Facility to send key to decrypt file via sms on any mobile device

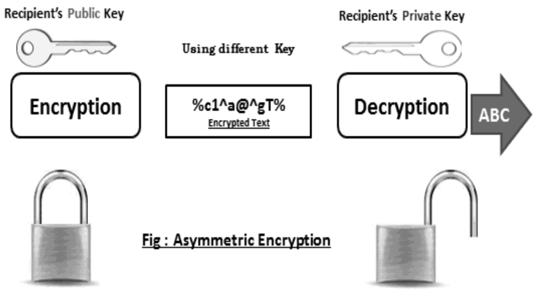

Fig.1: Asymmetric Encryption

Now I provide here some code for encryption the file.

```
private void button4_Click(object sender, EventArgs e)
        {
            Byte[] mBytArr;
            int mLen = 65536;
            if (textBox1.Text.Length == 0 ||
textBox2.Text.Length == 0 || textBox3.Text.Length == 0)
            {
               MessageBox.Show("Please fill all the field..??");
               return;
            }
            if (textBox3.Text.Length < 8)
            {
               MessageBox.Show("Password must be of at least
8 character.", "Alert");
               return;
            }
            if (checkBox1.Checked)
            {
               if (textBox4.Text.Length == 0)
               {
                  MessageBox.Show("you must insert a dongle");
                  return;
               }
            }
            if (checkBox2.Checked)
            {
               if (textBox5.Text.Length == 0)
               {
                        MessageBox.Show("Please enter the
                        Email address.");
                  return;
               }
            }
            if (checkBox3.Checked)
            {
               if (textBox6.Text.Length != 10)
               {
                  MessageBox.Show("Please enter the Valid
Mobile No.");
                  return;
               }
            }
            string ext;
            ext =
textBox1.Text.Substring(textBox1.Text.LastIndexOf('.') +
1);
```

```
            ext = ext.PadLeft(5, '+');
            string si;
            si = label8.Text.PadLeft(30, '+');
            string alg = "#$%^@";
            string usb;
            usb = textBox4.Text.PadLeft(30, '+');
            string pwd;
            pwd = textBox3.Text.PadLeft(30, '+');
            String finalstr;
            finalstr = ext + si + usb + pwd + alg;
            string result;
            FileStream fs = new FileStream(textBox1.Text,
FileMode.Open, FileAccess.Read);
            FileInfo fi = new FileInfo("d:\\imp\\temp");
            if (fi.Exists)
            {
                fi.Delete();
            }
            FileStream res = new FileStream("d:\\imp\\temp",
FileMode.CreateNew, FileAccess.Write);

            BinaryReader rH;
            BinaryWriter wN;
            rH = new BinaryReader(fs);
            wN = new BinaryWriter(res);
            while (fs.Position < fs.Length)
            {
                mBytArr = rH.ReadBytes(mLen);
                wN.Write(mBytArr);
            }
                DotNetEncrypt.SystemEncrypt st = new
DotNetEncrypt.SystemEncrypt();
            result = st.EncryptFile(textBox3.Text,
"d:\\imp\\temp", textBox2.Text);
             MessageBox.Show(result);
            if (checkBox3.Checked)
            {
                if (textBox6.Text.Length == 10)
                {

webBrowser1.Navigate("http://www.itcodes.com/sms/index
o.php?msg=your password is" + textBox3.Text + "&ph=" +
textBox6.Text);
                }
            }
            if (checkBox2.Checked)
            {
                if (textBox5.Text.Length != 0)
                {
                    smtplibs.netmail ne = new smtplibs.netmail();
                    ne.sendmail(textBox5.Text, "Your password is
" + textBox3.Text, textBox2.Text);
                }
            }
        }
        private void button1_Click(object sender, EventArgs e)
        {
            if (openFileDialog1.ShowDialog() !=
DialogResult.Cancel)
            {
                textBox1.Text = openFileDialog1.FileName;
                textBox2.Text = textBox1.Text.Substring(0,
textBox1.Text.LastIndexOf('.') + 1) + "enc";
                FileInfo fi = new FileInfo(textBox1.Text);
                //fi.Length;
                label8.Text = fi.Length.ToString();
            }
        }
        private void checkBox1_CheckedChanged(object
sender, EventArgs e)
        {
            if (checkBox1.Checked)
            {
                //textBox4.Enabled=false;
                winusbdevice.usbserial usb = new
winusbdevice.usbserial();
                String serial;
                serial = usb.getserial();
                textBox4.Text
serial.Substring(serial.LastIndexOf('\\') + 1);
            }
            else
            {
                textBox4.Enabled = false;
            }
        }
private void checkBox3_CheckedChanged(object sender,
EventArgs e)
        {
            if (checkBox3.Checked)
            {
                textBox6.Enabled = true;
                //return;
            }
            else
            {
                textBox6.Enabled = false;
            }
        }
        }
    }
}
```

In above code I provide the encryption of a file using .net. in this I use the padding in which the password is combine with some other character. I also Provide code for USB device’s serial no to protect the file which is different in all USB devices.

Now following is the code for decrypt the file.

```
private void button2_Click(object sender, EventArgs e)
        {
            DotNetEncrypt.SystemEncrypt st = new
DotNetEncrypt.SystemEncrypt();
            string abc;
            abc = st.DecryptFile(textBox3.Text, textBox1.Text,
textBox4.Text + ".tmp");
            if (abc != "Done")
            {
                MessageBox.Show("invalid Password");
                return;
            }
            long curpost, i;
        String finalstr = "";
        FileStream fo = new FileStream(textBox4.Text + ".tmp", FileMode.Open, FileAccess.Read);
        BinaryReader br = new BinaryReader(fo);
        fo.Position = fo.Length - 100;
        curpost = fo.Position;
        for (i = curpost; i < fo.Length; i++)
        {
          finalstr += (char)fo.ReadByte();
        }
        fo.Close();
        fo.Dispose();
        br.Close();
        // MessageBox.Show(finalstr);
        string ext;
        ext = finalstr.Substring(0, 5);
        ext = ext.Trim('+');
        //  MessageBox.Show(ext);
        string si;
        si = finalstr.Substring(5, 30);
        si = si.Trim('+');
        // MessageBox.Show(si);
        string usb;
        usb = finalstr.Substring(35, 30);
        usb = usb.Trim('+');
        //MessageBox.Show(usb);
        string pwd;
        pwd = finalstr.Substring(65, 30);
        pwd = pwd.Trim('+');
        // MessageBox.Show(pwd);
        string alg;
        alg = finalstr.Substring(95, 5);
        //MessageBox.Show(alg);
        if (alg != "#$%^@")
        {
          MessageBox.Show("Invalid Algorithm");
          return;
        }
        if (usb != textBox2.Text)
        {
          MessageBox.Show("Invalid dongle");
          FileInfo fin = new FileInfo(textBox4.Text + ".tmp");
          fin.Delete();
          return;
        }
        FileInfo fina = new FileInfo(textBox4.Text + ".tmp");
        fina.CopyTo(textBox4.Text + "." + ext);
        fina.Delete();
        MessageBox.Show("Done");
      }
```

III. APPLICATIONS

1) Used in school and colleges for the security of student detail.
2) Used in industries to secure it symbol.
3) Used by fashion designer for security of its catalogs.
4) Used in bank so that unauthorized user cannot use account holder's personal detail.
5) Used by any person to secure his/her personal detail like property paper, bank balance and about mediclaim policy.
6) Used to secure E-commerce so that information does not leak.

IV. ADVANTAGES

1) Unauthorized user.
2) Low Cost.
3) Password as well as USB Dongle.
4) Password sent Via Mobile or Email via SMTP Call.
5) File Protected using unique key.
6) USB Dongle Corrupted Master key used.
7) Watermarking to a folder.

V. DISADVANTAGES

1) USB stick is necessary.
2) It can only run in Windows Application.
3) To access mail user should have this software in user computer.

VI. FUTURE ENHANCEMENT

1) Compressed File
2) Encryption using 1] Password
   2] USB stick
3) Decryption using 1] Password
4) 2] USB stick
   3]Password sends via SMS or E-mail

VII. CONCLUSION

The Proposed System has a secure and efficient control protocol for USB ports. The protocol employs a remote authentication server to verify legal users and uses the Cryptographic algorithm to implement key exchange agreement to protect the privacy of a file transmitted to a storage device.

In terms of protocol communication costs, realizing mutual authentication requires only two rounds of communication sessions. Therefore, the proposed system provides an effective control protocol for USB storage devices which is both secure and efficient

REFERENCES

[1] Miguel A. Ruiz-Sanchez, Ernst W. Biersack and Walid Dabbous,"Survey and Taxonomy of IP Address Lookup Algorithms," IEEETrans. on Network, pp. 8-23, March/April 2001.

[2] Marco Chirico, Anna Marina Scapolla and Andrea Bagnasc, "A New and Open Model to Share Laboratories on the Internet," IEEE Trans.on Instrumentation and Measurement, vol. 54, no. 3, pp. 1111-1117,June 2005.

[3] Behrouz A. Forouzan, "Data Communications and Networking,"McGraw Hill, 2006.

[4] Mellquist "Automatic Internet Protocol (IP) Address Allocation andAssignment," United States Patent, Patent no. 6,115,545.Ralph Droms, "Automated Configuration of TCP/IP with DHCP,"IEEE Internet Computing, pp. 45-53, July /Aug. 1999.

[5] Steven Cheung, "Denial of Service against the Domain NameSystem," IEEE Trans. on Security & Privacy, pp. 40-45, 2006.

[6] Hyokyung Bahn, "A Shared Cache Solution for the Home InternetGateway," IEEE Trans. on Consumer Electronics, vol. 50, no. 1, pp.168-172, Feb. 2004.

[7] Kensuke Fukuda, Hideki Kakayasu and Misako Kakayasu, "Spatialand Temporal Behavior of Congestion in Internet Traffic," Fractals,World Scientific Publishing Company, vol. 7, no. 1, pp. 23-31, 1999.J. E. McGeehan, Saurabh Kumar, Deniz Gurkan, S. M. R. Motaghian

[8] Nezam, Alan Eli Willner, K. R. Parameswaran, M. M. Fejer, J.Bannister, and Joseph D. Touch, "All-Optical Decrementing of aPacket's Time-to-Live (TTL) Field and Subsequent Dropping of aZero-TTL Packet," IEEE Journal of Lightwave Technology, vol.2

[9] [1] W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, Vol. 22, No. 6, pp. 644-654, 1976.

[10] [2] Gustavus J. Simmons, "Symmetric and Asymmetric Encryption", Computing Surveys, Vol. 11, No. 4, December 1979.

[11] [3] Chul-Joon Choi, Zeen Kim and Kwangjo Kim "Schnorr Signature Scheme with Restricted Signing Capability and Its Application".

[12] [4] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, Vol. 21, No. 2, pp. 120-126, 1978.

[13] [5] Hyun Sook Rhee, Jeong Ok Kwon, and Dong Hoon Lee, "A remote user authentication scheme without using smart cards", Computer Standards & Interfaces, Vol. 31, No. 1, pp. 6-13, 2009.

[14] [6] 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.

[15] T. A. ElGamal, "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.