

DIGITAL DATA PROTECTION USING STEGANOGRAPHY

R. Rejani¹, D. Murugan² and Deepu V. Krishnan³

^{1,2}Department of Computer Science and Engineering, Manonmanium Sundaranar University, India

E-mail: ¹rejani@gmail.com, ²dhanushkodim@yahoo.com

³Infosys Limited, Technopark Campus, Trivandrum, India

E-mail: deepu_krishnan@infosys.com

Abstract

In today's digital world applications from a computer or a mobile device consistently used to get every kind of work done for professional as well as entertainment purpose. However, one of the major issue that a software publisher will face is the issue of piracy. Throughout the last couple of decades, almost all-major or minor software has been pirated and freely circulated across the internet. The impact of the rampant software piracy has been huge and runs into billions of dollars every year. For an independent developer or a programmer, the impact of piracy will be huge. Huge companies that make specialized software often employ complex hardware methods such as usage of dongles to avoid software piracy. However, this is not possible to do for a normal independent programmer of a small company. As part of the research, a new method of software protection that does not need proprietary hardware and other complex methods are proposed in this paper. This method uses a combination of inbuilt hardware features as well as steganography and encryption to protect the software against piracy. The properties or methods used include uniqueness of hardware, steganography, strong encryption like AES and geographic location. To avoid hacking the proposed framework also makes use of self-checks in a random manner. The process is quite simple to implement for any developer and is usable on both traditional PCs as well as mobile environments.

Keywords:

Data protection, Steganography, Stego Image, Cover Image, Software Protection, Encipher, AES, SteganoDB, LSB

1. INTRODUCTION

Information that is exchanged via computer networks is generally specified in the form of data. Data can be stored in various formats. It can be text, image, sound, video, graphics etc. These data are stored in different devices like computers, mobiles, movie players etc. For example, digital camera stores the picture information as image bits. For storing and retrieving information, there is a need for software and devices. Digital data or software which a computer uses can be quite easily copied which is one of its main strength as well as its weakness.

Most of the data transmitted are meant for mere communication purpose but some of these might be meant for confidential purpose like say banking data or ecommerce purpose. Hence, for protecting data methods like encryption are normally used.

Steganography is the science or method of hiding information within an image. The protection offered by steganography can be further enhanced and more robust by using encryption techniques before hiding the text inside an image. According to BSA (Business Software Alliance) study every year the software piracy rate is increasing which has been resulting in huge revenue loss to every software companies. To

overcome this, different methods are used like Licensing Acts, Patents, cryptographic methods, dongles etc. However, these methods have not been 100% effective and using special hardware is expensive and is not a good solution for small developers or companies. This cannot be used in mobiles as well. Hence, there is a need for an alternate software protection framework that is cost effective and at the same time provides excellent protection against hackers.

This paper proposes a new software protection framework, which uses cryptography, steganography as well as a new process to protect both traditional desktop applications as well as the newer mobile apps as well.

2. RELATED WORK

Rejani R et al. [2] presents a new/alternate secure Database system based on steganography for data hiding. The system provides integrity more confidentiality and authentication during access or editing of confidential data. The proposed DB system uses steganography technique to store a database of records. The system allows a user to create basic tables and records, which are hidden from others inside an image. In this paper, we propose an architecture, which can be used by application developers to retrieve data from the created database in an easy manner. The proposed method is highly useful for use as an embedded DB also in mobile computing as it can store small amount of data easily.

Bertrand Anckaert et al. [3] identifies the fundamental weaknesses of existing approaches, resulting from the static nature of defense and the impossibility to prevent the duplication of digital data. A new scheme is presented that enables a more dynamic nature of defense and makes it harder to create an additional, equally useful copy. Furthermore, it enables a fine-grained control over the distributed software. Its strength is based on diversity: each installed copy is unique and updates are tailored to work for one installed copy only.

In [4] data is hidden using Three layers in Audio. Goal of this work is to increase level of security so that data can be guarded. Phad Vitthal et al. [5] proposes a protection scheme using cryptography and PVD & LSB combination of steganography. Shamim Ahmed Laskar and Kattamanchi Hemachandra proposed [6] a high performance JPEG steganography along with a substitution encryption methodology. The approach uses the discrete cosine transform (DCT) technique used in the frequency domain for hiding encrypted data within image. Minati Mishra et al. [7] discusses some different types of data hiding techniques.

3. EXISTING METHODS FOR PROTECTION OF DATA AND SOFTWARE

3.1 H/W MEANS OF PROTECTION

In this method of s/w protection, the publishers generally provide a special h/w that needs to be connected when the s/w is being used.

3.1.1 USB/Serial port based Dongles:

The h/w protection method that most commonly available as dongles. The dongles can be either serial port based or it can be USB based dongles. The dongles are to be connected to the computer at all time for the s/w to work. When the s/w is executed, it first checks for the presence of the dongle and checks whether the encrypted key is present in the dongle's in built memory. If it matches with the registration info then the software is executed, else the software will not run and will throw an error. This kind of s/w protection is often costly as it involves additional h/w and hence used for specialized software, which by itself cost millions to develop and deploy. This method also involves need of installation of additional drivers to make the hardware dongle work. This will often turn out to be cumbersome and difficult to manage in an enterprise environment if there is a need to install the software on hundreds of computers.

3.1.2 CAD/CAM Software:

CAD/CAM software usually used in specialized industries like oil, mining etc. costs millions of dollars to develop and enhance and generally use dongle based protection schemes for protection against possible piracy.

3.1.3 Animation/3D software:

Expensive 3D/Animation software usually cost thousands of dollars and usually use hardware based protection modules.

3.1.4 Steinberg's Key:

A popular example is the dongle called Steinberg Key, used to protect popular Steinberg's products for audio protection and editing solutions. In the case of this particular key, it can be bought separately of the product itself. This dongle apart from the protection part of the software can also be programmed to exactly in such a way, say which features of the software are bought by the user also needs to be enabled. For example when a user buys only say 2 modules of a particular software say which manages the ERP functionality of an Enterprise, the hardware dongle can be programmed in such a way that when it is connected to the computer along with the software only the intended functionality would be working while the dongle is present.

Some of the initial versions of dongle-based protection was based on detection, which means that if the software detects the presence of the dongle then it will work. However, overtime the dongles started getting small amount of data storage capabilities as well, which was enough to store information regarding registration and other vital data, which will assist with software protection.

3.1.5 Smart Cards:

The recent development in h/w based protection is to use smart card based protection where a card reader is attached to the

system and the inserted card is read and provided access to the software.

3.2 S/W MEANS OF PROTECTION WITH THE SUPPORT OF HARDWARE

Various s/w publishers employ different means for protection using software means.

3.2.1 Serial Numbers:

Usually the simplest method that's employed is to use Serial numbers which is generated based on an algorithm to tie a user to a particular software. Later on there were some additional methods used by game developers and the most common method used is usually to detect the presence of the original game cd in the cd drive when the game is being played or started. Another method that is commonly used these days is to include special digital signatures into discs produced at the time of manufacture. Either game(s) or software, when they run, will check the cd and the existence of the copy protection digital signature on the disc. However, they do require special hardware to manufacture these discs.

3.2.2 CD Based Protection:

One of the most common methods to copy music from Audio CDs and DVDs is to use ripping software to convert the tracks to MP3 music. At later stages, to circumvent this, several recording studios introduced monitoring software, which will be installed when the dvd/cd is first inserted into the computer. The software will in turn monitor if any ripping software is being run to copy the music and if it finds any, will disable the access to the drive or stop the software from doing this.

3.2.3 DVD Region Coding:

DVD region coding is another method that is used to region lock software to a particular region so that DVDs that are supposed to be sold in that region will be able to play in that particular DVD Player. However, afterwards several DVD players came that circumvented this and hence this method was rendered useless.

3.2.4 DRM Services:

DRM services make use of authentication servers in the internet when a music or media is played. However, this caused bad user experience when the server went down, as the users were not able to playback the media. This method is limited to media consumption and cannot be used for protection of software.

3.3 S/W MEANS OF PROTECTION

From ancient times, several methods are used for protecting data/information.

3.3.1 Cryptography:

Cryptography is the science using for secure communication. It converts the plaintext to cipher text for preventing unauthorized person to access the message/information. Several algorithms are developed in the intention of protecting data. In those, some famous algorithms are DES, TDES, AES, Blowfish, RSA etc. AES is the commonly used standard by the government of USA.

3.3.2 Steganography:

It is a step forward from Cryptography. It protects the information inside an image. Since nobody suspects about the

concealing of information. Many steganography techniques are established to hide the data effectively. Some famous steganography algorithms are LSB, RGB, PVD, etc.

Apart from these techniques Water marking, Visual cryptography etc are other techniques used for protection of data.

As already discussed, most of the existing methods have many disadvantageous even though they provide a basic level of protection.

Most of the hardware based protection methods are very difficult to implement, as it requires additional hardware like a dongle.

H/W based methods are not cost effective as the additional hardware usually implements proprietary algorithms and hardware

Easy for crackers to identify the existence of additional hardware and hence they would be able to write additional backdoor programs which can clone the functionality of the hardware and there by bypassing the protection method also.

CD based protection of games can be easily manipulated using cloning software

Serial number based protection method can be replicated to other systems and software can be pirated.

Secure id based protection using RSA is cumbersome to maintain and expensive.

4. PROPOSED TECHNIQUE

Considering disadvantages and advantages of the existing techniques a new protection framework proposed. It combines the advantages of cryptography, steganography and hardware features. The new method is devised in such a manner that the implementation is easy and less cumbersome and difficult to hack or crack by others.

This protection scheme can be applicable to both desktop applications as well as mobile applications and any type of applications.

4.1 SOFTWARE PROTECTION FRAMEWORK

The algorithm has two parts one is the Authentication algorithm and second is the protection algorithm.

Registration process using authentication algorithm is the first step. Next step is the stego token creation and then attach this with the software to protect. The final step is the software execution phase where the validation algorithms to verify the authenticity.

4.2 AUTHENTICATION ALGORITHM

Any software when it is initially installed will need to be registered. From this step will start the process for protecting application. To achieve this, there is a need to identify some unique property of the computer, which can be used for registration purpose. The best option here is to make use of the Unique hard disk id, mac address, motherboard serial etc. to authenticate the computer on which the software is installed. The reason why these are chosen is because these are unique to a computer and cannot be changed easily. In fact, to increase the protection levels a combination of these can also be used.

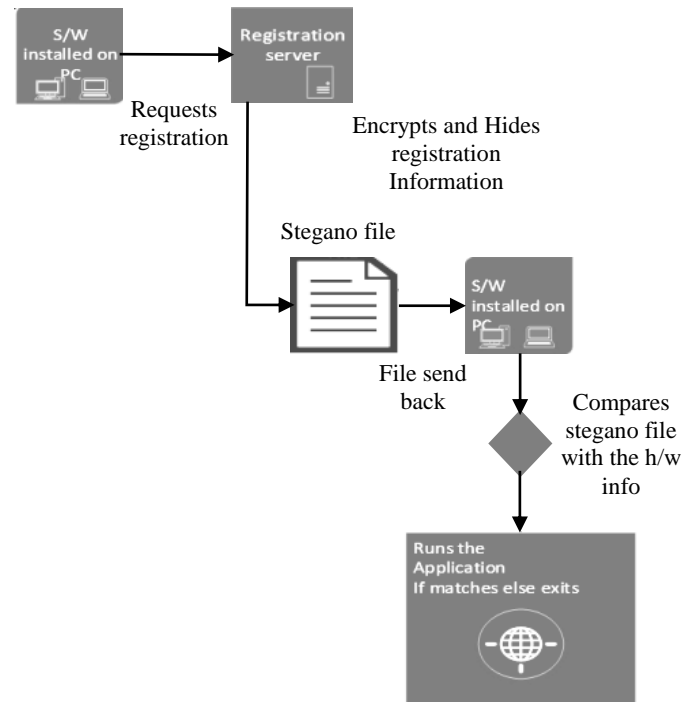


Fig.1. Architecture of software protection

Step 1: Fetch the HDD serial number or MAC address and user information.

In order to make a unique identifier that can be used to authenticate a registered user, the unique properties of the computer need to be found.

Algorithm for getting HDD Serial

Step 1: invoke the Win32_DiskDrive API

Step 2: select the Serial Number

Windows provides the Win32_DiskDrive API function to retrieve the serial number of a hard disk.

Sample code for getting HDD serial number is as below:

```

ManagementObjectSearcher searcher =
newManagementObjectSearcher("root\\CIMV2",
"SELECT * FROM Win32_DiskDrive");
for each (ManagementObjectqueryObj in
searcher.Get())
{
Console.WriteLine("SerialNumber: {0}",
queryObj["SerialNumber"]);
Console.WriteLine("Signature: {0}",
queryObj["Signature"]);
}
}
  
```

Algorithm for getting MAC address

Step 1: invoke the GetAdaptersInfo API

Step 2: select the pAdapterInfo->Address

Sample code for getting MAC address is as below:

```

static void GetMACAddress(void)
{
  
```

```

IP_ADAPTER_INFO AdapterInfo[16];
DWORD dwBufLen = sizeof(AdapterInfo);
DWORD dwStatus = GetAdaptersInfo(AdapterInfo,
&dwBufLen);
PIP_ADAPTER_INFO pAdapterInfo = AdapterInfo;
do {
PrintMACAddress(pAdapterInfo->Address);
pAdapterInfo = pAdapterInfo->Next;
} while(pAdapterInfo);
}

```

Step 2: Encrypt it using AES encryption standard and write it in a file.

```

static byte[] EncryptStringToBytes_Aes(string
plainText, byte[] Key, byte[] IV)
{
if (plainText == null || plainText.Length<= 0)
throw new ArgumentNullException("plainText");
if (Key == null || Key.Length<= 0)
throw new ArgumentNullException("Key");
if (IV == null || IV.Length<= 0)
throw new ArgumentNullException("Key");
byte[] encrypted;
using (AesAlg = Aes.Create())
{
aesAlg.Key = Key;
aesAlg.IV = IV;
ICryptoTransform encryptor =
aesAlg.CreateEncryptor(aesAlg.Key, aesAlg.IV);
using (MemoryStream msEncrypt = new
MemoryStream())
{
using (CryptoStream csEncrypt = new
CryptoStream(msEncrypt, encryptor,
CryptoStreamMode.Write))
{
using (StreamWriter swEncrypt = new
StreamWriter(csEncrypt))
{
swEncrypt.Write(plainText);
}
}
encrypted = msEncrypt.ToArray();
}
}
return encrypted;
}

```

Step 3: Send the file to the License Server

The License server will be located at the software provider end. The file generated at the step2 will be send to the license server. The server will provide HTTP APIs

for receiving the file. Alternatively, it can also use secure file transfer protocol like SFTP to transfer the file.

Step 4: Fetch the user mail id along with the information collected by step 3.

Step 5: Locate the user using GPS (Optional step)

Optionally it can also find out the location of the user based on GPS data and can use that for further protection.

Algorithm for getting GPS location

invoke the Windows.Devices.Geolocation API

select the pAdapterInfo->Address

Step 6: User authentication information is saved in License server.

4.3 PROTECTION ALGORITHM (AUTHENTICATION KEY CREATION)

Step 1: Creation of user authentication information (using the Authentication algorithm).

Step 2: Fetch the Identity information from the stored file (HDD Serial number, MAC Address) place, name, email id etc.

Step 3: Apply the steganoDB package [2].

Create steganoDB structure,

```

{
"First Name": "Arun",
"Last Name": "Kumar",
"Email": "arun.kumar@outlook.com",
"Address Line1": "124#Lane 15",
"Address Line2": "T-Nagar",
"Address Line3": "Chennai-30",
"Phone": "999"
"Key": "U2FsdGVkX19L7/iYCBYbur74I5oNTBL/nBaMPfgg+s="
}

```

SteganoDB is a unique DB structure used to embed data in images that can also be read easily. As part of the research, the properties of JSON structure combined with steganography to create a unique steganoDB. The encrypted value of the hardware properties or the registered user's other information like name, email address all can be easily stored in this steganoDB and when needed can be easily retrieved using any modern programming language. All the modern programming languages like Java or .net support JSON structure for storing of data.

Step 4: Embed the encrypted data inside the given image using Pixel Pattern based Steganography algorithm [8].

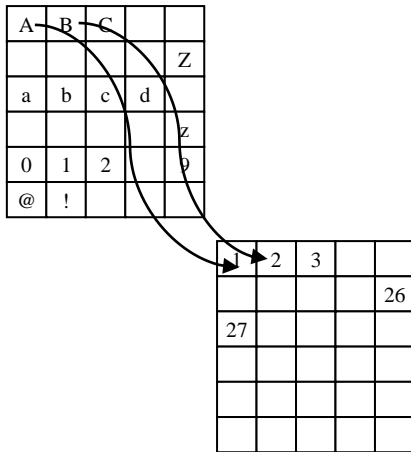


Fig.2. Substitution values

The advantage of this proposed technique is that it will not degrade the image quality as it depends on the pixel values. Hence, the covering image and the steganography image will not have any visual difference and will be prone from any sort of attacks.

There are two techniques used to protect the information within the key file. First is the encryption algorithm and the second will be steganography.




Step 5: Save the encoded image file at the receiver’s end.

Once the stegano key is generated, it is send back to the software, which is running on the desktop. This can be automated via existing protocols like http or ftp. To make it more secure http (https) or ftp (sftp) can be used

Step 6: For each execution of the protected software checks for this encoded key.

Different execution conditions of the software protection algorithm given below:

Table.1. Various Checking conditions

Checking Conditions		Result
 Existing	HDD Serial no. matches with the extracted key and authentication details	Executes the s/w
 Existing	HDD Serial no. does not match with the extracted key	s/w asks for registration
 Not Existing	Not applicable	s/w asks for registration

Step 7: Extract the key from the stego image.

After receiving the file back from the Registration server it will be stored in a predefined area. When the s/w runs, it will first read the steganography file using the reverse process as below:

Get the key from the server,

For all characters in the image metadata

```
{
  For all pixels in the image
  {
    Find the modbits from the pixels positions specified
    Decrypt the modbits
    Display the character continue with next character
  }
}
```

After reading the steg data, first, the serial number will be extracted, and then the reverse process of AES will decrypt it.

Step 8: Decrypting the encrypted text using AES.

Function for decrypting a text which has been encrypted using AES is as below:

```
static string DecryptStringFromBytes_Aes(byte[] cipherText, byte[] Key, byte[] IV)
{
  if (cipherText == null || cipherText.Length <= 0)
    throw new ArgumentNullException("cipherText");
  if (Key == null || Key.Length <= 0)
    throw new ArgumentNullException("Key");
  if (IV == null || IV.Length <= 0)
    throw new ArgumentNullException("Key");
  string plaintext = null;
  using (AesAlg = Aes.Create())
  {
    aesAlg.Key = Key;
    aesAlg.IV = IV;
    ICryptoTransform decryptor =
    aesAlg.CreateDecryptor(aesAlg.Key, aesAlg.IV);
    using (MemoryStream msDecrypt = new
    MemoryStream(cipherText))
    {
      using (CryptoStream csDecrypt = new
      CryptoStream(msDecrypt, decryptor,
      CryptoStreamMode.Read))
      {
        using (StreamReader srDecrypt = new
        StreamReader(csDecrypt))
        {
          plaintext = srDecrypt.ReadToEnd();
        }
      }
    }
  }
}
```

```

}
}
return plaintext;
}
}}

```

Once the decryption process is completed, the decrypted data will be compared again with the captured hard-disk serial number again and if there is a match, the software will be executed.

Step 9: If the key does not match/What happens when a person tries to pirate the software?

In case someone tries to pirate the software by installing it on another PC or tries to just copy, the folder where the software is installed along with the key file, then when the comparison step of the steganography file with the hardware properties of the PC is done it will fail. Moreover, either the software will then do a gracious exit or it can be also programmed to route to a buy page on internet where the particular person will be asked to buy the software by using an online payment.

Step 10: Regular Self-checks to make sure that the software is not tampered by anyone.

the system or sending files from one computer to another computer using the proposed steganography algorithm and steganoDB. The algorithm for this is given below:

Step 1: Select the image file on which the text will be embedded.

Step 2: Request/Choose the password for protecting the database/file.

Step 3: From the sending side the software will request the one-time key from the receiver computer side. From the receiver computer side, the software will detect the hardware feature like hdd serial number or mac address and transmit it back to the sending side. This will be used as the key file for recognizing the receiver. This is a one-time process and will not be needed for subsequent file transfers. For the second time this process not needed until any change needs.

Create steganoDB structure,

```

{
  "First Name": "Arun",
  "Last Name": "Kumar",
  "Email": "arun.kumar@outlook.com",
  "Address Line1": "124#Lane 15",
  "Address Line2": "T-Nagar",
  "Address Line3": "Chennai-30",
  "Phone": "999"
  "Key": "U2FsdGVkX19L7/liYCBYbur74I5oNTBL/nBaMPfgg+s="
}

```

Step 4: Convert the text to be protected into JSON structure.

Step 5: Apply [2] SteganoDB algorithm with the password key used as received hdd or mac address of the receiver.

Step 6: Apply steganography data embedding algorithm [8] and enter the file/database to protect.

Windows.Media.Imaging Namespace provides classes to encode/decode or manipulate images. The features of this Namespace to implement steganography. This function will accept the image to store text to hide. It will store the text into bytes and will return the image.

Step 7: Store/Send the stegano image file.

The method of sending can be either via sftp or https but either way at the receiving side the software will open ports to receive the file and save them locally.

Step 8: The receiver's side receives the file and saves it.

Step 9: Checks the hdd number/ mac address and uses it as the password to decode the image.

Windows provides the Win32_DiskDrive API function to retrieve the serial number of a hard disk.

Sample code for getting HDD serial number is as below:

```

ManagementObjectSearcher searcher =
new ManagementObjectSearcher("root\\CIMV2",
"SELECT * FROM Win32_DiskDrive");
foreach (ManagementObject queryObj in searcher.Get())
{

```

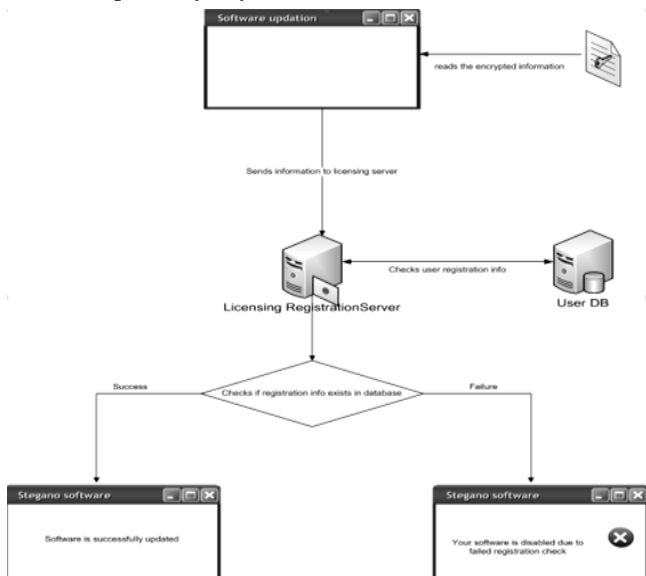


Fig.3. Software regular updating through internet

The final step once a software is installed, is to make sure that the software is not tampered by anyone to use it illegally. To make sure that it does not happen the key properties of the executable file like .crc can be stored to a stegano image with encryption. When the software is executed, randomly it will check the properties and compare with the stegano file and if it fails it will stop the execution of the software.

4.4 FILE/DATABASE PROTECTION ALGORITHM

In this section an additional usage of new steganography method is proposed which can be used for secure transfer of file/database from a laptop/desktop to another computer or it can be securely stored inside the desktop/laptop. This process can be used to protect database or other files in a secure manner inside

```

Console.WriteLine("SerialNumber: {0}",
queryObj["SerialNumber"]);
Console.WriteLine("Signature: {0}",
queryObj["Signature"]);
}
}

```

Step 10: Using Steganography data retrieving algorithm [8], it decodes the image and separates the text stored in the image. Next, the encrypted text will be decoded.

Step 11: Read the text in JSON format and store it in a text file.

Using this method text from .pdf files, database and text files can be protected and shared.

5. EXPERIMENTAL RESULTS

5.1 DESKTOP/LAPTOP APPLICATION

To show the experimental results of the algorithm a prototype created. It works in four steps.

User side:

At the user end there will be a registration wizard as well as the main software. The registration software will get the HDD serial number, encrypt the HDD serial number and send it via email to the authentication server on the internet.

Along with the main software there will be a module present that will check the presence of the key file, then it will decode the key file and will check whether the decoded value is matching with the actual HDD key. If both of them matches, then only the software will execute.

There will an update module that will regularly check the license internet server for any software updates by passing along the registration info as well. If the registration info is not valid, it will exit and will make the software unusable.

Server side:

At the server side there will be a separate software module that will receive the registration request along with the encrypted HDD serial number. It will hide the HDD serial number within the image and will send the image back to the user to complete the registration.


The protection module will use very little system resources and hence the user will not experience any degradation in performance.

The AES algorithm for decoding text from key image will be at the user side however to the user its working will be transparent. At the server side there will be encoding of the text into the key image.

5.1.1 Authentication Algorithm Result:

The registration wizard is the first step while the protected software is executed. This will accept the buyers address and other user information.

5.1.2 Output:



```

{
  "First Name": "Arun",
  "Last Name": "Kumar",
  "Email": "arun.kumar@outlook.com",
  "Address Line1": "124#Lane 15",
  "Address Line2": "T-Nagar",
  "Address Line3": "Chennai-30",
  "Phone": "9895312289",
  "Key": "U2FsdGVkX19L7/IiYCBYbur74I5oNTBL/nBaMPfgg+s="
}

```

Fig.4. Authentication information text

5.1.3 Key File Creation:

This algorithm is executed from the server side on receiving the registration key from the user. The information got from the user via text file is embed in the cover image using SteganoDB algorithm and Pixel Pattern based Steganography algorithm used in this module will then hide the key into an image and this resultant image file is send to the user.

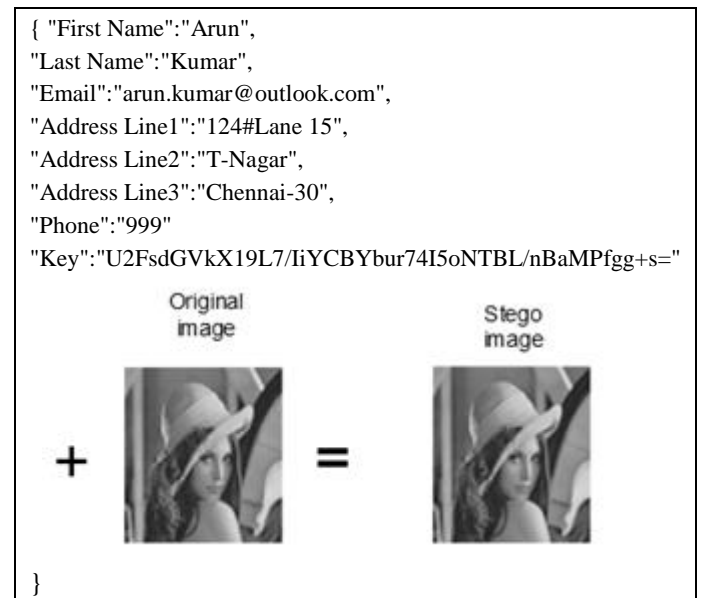


Fig.5. Stego image creation

5.1.4 Software Execution Output:



Fig.6. Screen shots of successful execution

If it does not match, then the software will not execute. It will give the message of registering the software properly.

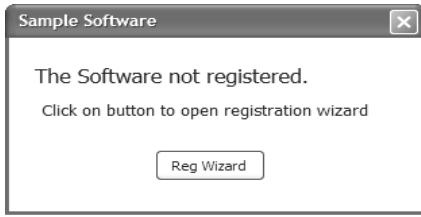


Fig.7. Screen shots of failure execution

5.2 FILE/DATABASE PROTECTION

If there is a need to get data from text within a word document, then the api's provided by Microsoft can be used. Once the input file is provided to the API, it will read the document and will extract the text from the document. The Authentication information becomes one of the input of file protection. It is the key to deciphering the message.

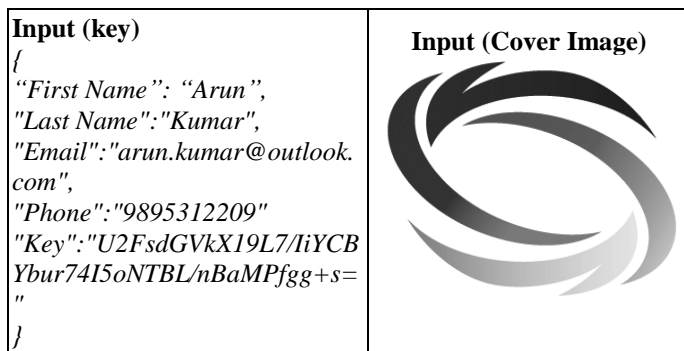


Fig.8. Cover Image

The contents of the word file get using API functions and embed it inside the cover image using Pixel Pattern based Steganography algorithm. In addition, the authentication information also is embed using the Steganodb structure kept inside the cover image. Moreover, gives the output as picture.

5.2.1 Input (Word document):



Fig.9. Word file

5.2.2 Output (Stego Image):



Fig.10. Stego image

While extracting the contents of the file at the receiver side the software will check the authentication information, if the authentication is right the contents of the file is extracted and gives the output as text format.

6. PERFORMANCE ANALYSIS

The registration information stored is as below which will be encrypted value of HDD serial number or MAC address. For the test purpose 4 test cases with test data was used. Two for HDD and two for MAC address taken.

Input data's used for test:

HDD 1:

```
{3WXpf6UXRV5nxAleQml9BRohlKw0Whs/XKGww4OdPb0yJ8qNZc0nSeSM5Kxldosp}
```

HDD 2:

```
{hAJN64oQ1j0=}
```

MAC Address 1:

```
{WZrk6Sbl6d0IQITxOC0FE0pZEI52OzX7}
```

MAC Address 2:

```
{4rTvk7zBIC31+FM/ONmUGIVa4xu0oYTT}
```

```
{
```

```
"First Name": "Arun",
```

```
"Last Name": "Kumar",
```

```
"Email": "arun.kumar@outlook.com",
```

```
"Address Line1": "124#Lane 15",
```

```
"Address Line2": "T-Nagar",
```

```
"Address Line3": "Chennai-30",
```

```
"Phone": "9956128119"
```

```
"Key": "3WXpf6UXRV5nxAleQml9BRohlKw0Whs/XKGww4OdPb0yJ8qNZc0nSeSM5Kxldosp"
```

```
}
```

Criteria 1: Comparison of image quality by embedding the given inputs

The image quality is measured by calculating the value of PSNR, MSE and SSIM values. Different possible inputs are embedded into the given cover image and calculated the corresponding stego image's PSNR, SSIM and MSE values.

Table.2. Image quality of stego image used for software protection

Size of Image	Size of Data	PSNR (db)	SSIM	MSE
250k	218 bytes (HDD2)	99.7989 dB	.99999	.00047
250k	238 bytes (MAC Address 1)	98.8298 dB	.99999	.00059
250k	238 bytes (MAC Address 2)	98.8298 dB	.99999	.00059
250k	270 bytes (HDD1)	94.4382 dB	.99999	.00021

From the Table.2, it is understood that the image quality of the stego image is good. Hence, the noise is less. If adding more data as checking constraint also it will maintains the image quality. This is an added advantage that if enhance the protection by giving other information this algorithm gives good image quality and good support for the protection.

Criteria 2: Comparison of Quality analysis of Data for the above mentioned inputs

The comparison of data quality with other techniques analyzed in [2]. The data maintenance, insertion and processing is good compared to any other data insertion technique. In this case, the data insertion and extraction time from the stego image for various data size is calculated. This will help to calculate whether the data extraction is in time and whether it is correct and reliable. Tests were done using the given inputs and results were obtained clearly and completely in time. It is comparatively fast retrieval than any other technique. The results are summarized as below:

Table.3. Data quality analysis of software protection

Size of Image	Size of Data	Data insertion time	Data extraction time	Accuracy
250k	270 bytes (HDD1)	1.5 sec	800 millisecs	100%
250k	218 bytes (HDD2)	1 sec	750 milli sec	100%
250k	238 bytes (MAC Address 1)	1 sec	750 millisecs	100%
250k	238 bytes (MAC Address 2)	1 sec	750 millisecs	100%

These values show that the data retrieval is fast when compared to any other algorithm. Since the data retrieval is fast the other process are fast getting good response time as total. Due to this, the totality of the algorithm gets good performance and good response time. One more advantage of this is the whole process is not transparent to the user as it runs in background and hence the chance of suspicion avoided. This makes the protection better.

Criteria 3: Comparison of Image quality analysis of Data for file protection

The same method can be used for storing of data from files like text, pdf into image and securely transfer it to the receiver. The data inserted in the case of file protection algorithm is comparatively more than the data inserted in the software protection process.

Table.4. Image quality analysis after inserting file

	Size of Image	Size of Data	PSNR	SSIM	MSE
Cover image1	250k	1000 bytes	94.4382 dB	.99999	.00021
Cover image1	250k	1500 bytes	91.55028 dB	.99999	.00041
Cover image1	250k	2000 bytes	88.9292 dB	.99999	.00075

From the Table.4, it is clear that if more data embedded into the image then also it will change the pixels in the image only minimally. Since similar pixel values position is kept as meta data and also those pixels which are changed also will have only very minor change in the pixel values and hence it gives least changes in the resultant image. This will maintain the image quality and gives better protection for data. Due to this reason, even 2000 bytes of data embedded in the image got good PSNR ratio and SSIM values. Noise is less as compared to any other technique.

Criteria 4: Comparison with Other Protection Techniques

Normally for protection purpose, methods include cryptography, steganography, dongles, serial numbers and other protection techniques. Its usability, convenience, implementation easiness, maintenance of algorithm, extensibility, cost and its security capability is considered. The result of the analysis given in precise below.

Table.5. Comparison with other protection techniques

Technique	Security level	Cost	Protection against hacking & piracy
Serial numbers	Low	Low	Very low. Easily replicable
Hardware dongles	Medium	Very High	High protection
Cryptography	High	Low	Medium protection
Steganography	High	Low	Medium protection
Steganography + hardware features + encryption	Very high	Very Low	Extremely high protection

The protection technique is used as a combination of cryptography, steganography and identification features, which gives more protection than any one of the protection methods using alone. Cryptography have limitations to protecting data. Not all types of data can be protected using cryptography alone. There is chance of getting data reverse engineered using brute force attacks. Steganography also sometimes is prone to steganalysis methods to unhide the data. Hardware dongles mostly give strong protection but it is costly and also it can't be used for all types of systems.

7. MOBILE DATA PROTECTION

The proposed algorithm can be extended to the protection of mobile apps. The basic framework will remain the same except for few changes in the algorithm. Instead of HDD serial number for mobile app protection, for mobiles the IMEI number can be used. Similarly, if an app needs to be limited to a particular geography that can also be done with the help of GPS information. This information can be also hidden using steganography and protected.

8. CONCLUSION

A new software protection framework has been proposed, which uses a new pixel pattern based steganography algorithm, combining it with encryption as well as unique hardware properties. This proposed framework can be used to protect normal computer applications as well as mobile applications against piracy. The existing methods often provide a single layer of protection and hence is quite easy to hack/crack in most cases. However, the proposed method uses different layers of protection. The new steganography algorithm is not easy to detect as it does not change the pixels of an image. And on top it uses strong encryption algorithm like AES to provides extreme protection. It also can use Geolocation based checks and CRC based self-checks to make sure that the software is completely protected against any hacking method. When compared against the existing methods of software protection where this new method's advantages are:

1. The new proposed framework uses the uniqueness of computer or the mobile in which it is installed. Hence, it will not work if the software is pirated from one system to another.
2. The registration process as part of the proposed protection framework uses strong AES and the newly developed Steganography algorithm to protect the key used and hence hackers will not be able to hack this method.
3. The framework does not need any additional or custom software or hardware dongles to work.
4. The registration process proposed uses regular internet and hence does not have any special networking requirement.
5. Since the framework uses triple protection ie h/w properties, strong encryption and custom steganography the protection will be extremely difficult to hack or break.

Prototype applications developed for testing the framework proposed and it was run both in protected as well as unprotected mode. The tests proved the effectiveness of the framework as a data protection method.

REFERENCES

- [1] R. Rejani, D. Murugan and Deepu V. Krishnan, "Novel Software Protection Framework Using Steganography, Cryptography, Uniqueness of Hardware and Self-Checks", *International Journal of Advanced Information Science and Technology*, Vol. 25, No. 25, pp. 32-40, 2014.
- [2] R. Rejani, D. Murugan and Deepu V. Krishnan, "STEGANODB-A Secure Database using Steganography", *ICTACT Journal on Communication Technology*, Vol. 4, No. 3, pp. 785-789, 2013.
- [3] Bertrand Anckaert, Bjorn De Sutter and Koen De Bosschere, "Software Piracy Prevention through Diversity", *Proceedings of the 4th ACM workshop on Digital rights management*, pp. 63-71, 2004.
- [4] Parul and Vikas Kamra, "Three Layer Protection for Secure Data Transmission using Digital Audio as Carrier", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, No. 3, pp. 523-526, 2015.
- [5] S. Phad Vitthal, S. Bhosale Rajkumar and R. Panhalkar Archana, "A Novel Security Scheme for Secret Data using Cryptography and Steganography", *International Journal Computer Network and Information Security*, Vol. 4, No. 2, pp. 36-42, 2012.
- [6] Shamim Ahmed Laskar and Kattamanchi Hemachandra, "Secure Data Transmission Using Steganography and Encryption Technique", *International Journal on Cryptography and Information Security*, Vol. 2, No. 3, pp. 161-172, 2012.
- [7] Minati Mishra, Priyadarsini Mishra and M.C. Adhikary, "Digital Image Data Hiding Techniques: A Comparative Study", *ANSVESA*, Vol. 7, No. 2, pp. 105-115, 2012.
- [8] R. Rejani, D. Murugan and Deepu V. Krishnan, "Pixel Pattern Based Steganography on Images", *ICTACT Journal on Image and Video Processing*, Vol. 5, No. 3, pp. 991-997, 2015.
- [9] Luhn mod N algorithm, Available at: http://en.wikipedia.org/wiki/Luhn_mod_N_algorithm.
- [10] Joan Daemen and Vincent Rijmen, "*The Design of Rijndael: AES-The Advanced Encryption Standard*", Springer, 2002.
- [11] R. Rejani, D. Murugan and Deepu V. Krishnan, "Comparative Study of Spatial Domain Image Steganography Techniques", *International Journal Advanced Networking and Applications*, Vol. 7, No. 2, pp. 2650-2657, 2015.
- [12] Mohammad Tanvir Parvez and Adnan Abdul Aziz Gutub, "RGB Intensity Based Variable-Bits Image Steganography", *Proceedings of IEEE Asia-Pacific Services Computing Conference*, pp. 1322-1327, 2008
- [13] Richa Raja Gautam and Rakesh Kumar Khare, "Real Time Image Security for Mobile Communication Using Image Steganography", *International Journal of Engineering Research & Technology*, Vol. 1, No. 8, pp. 1-5, 2012.
- [14] Sharmishta Desai, Sanaa Amreliwala and Vineet Kumar, "Enhancing Security in Mobile Communication using a Unique Approach in Steganography", *International Journal of Computer Science and Mobile Computing*, Vol. 3, No. 4, pp. 433-439, 2014.
- [15] A.Z. Aos, A.W. Naji, Shihab A. Hameed, Fazida Othman and B.B. Zaidan, "Approved Undetectable-Antivirus Steganography for Multimedia Information in PE-File", *Proceedings of International Association of Computer Science and Information Technology*, pp. 437-441, 2009.
- [16] Huayong Ge, Mingsheng Huang and Qian Wang, "Steganography and Steganalysis Based on Digital Image", *Proceedings of 4th International Congress on Image and Signal Processing*, Vol. 1, pp. 252-255, 2011
- [17] Brijesh Rajput, "A Survey of Contemporary Protection Mechanism for Preventing Piracy Of Digital Discs", *International Journal of Scientific and Technology Research*, Vol. 2, No. 4, pp. 207-211, 2013.