CREDI-crypt: An Improvised Anti-Counterfeiting Technique for Credit Card Transaction System

Santosh Hariharan[*], V Naveen Kumar[*] (Student Member IEEE), Mrigank Rochan[**]

[*] Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Bangalore-560035, Karnataka, India.
[**] Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Bangalore-560035, Karnataka, India.
harisan9@gmail.com, vnaveenkumar@ieee.org, mrigankrochan@gmail.com

Abstract— **Data encryption is vital for e-commerce. Here we introduce a technique for hiding sensitive credit card information in an arbitrary background image without the sensitive data getting exposed in public domains. The paper also makes an attempt to combine Cryptographic and Steganographic technique to provide a reliable security solution for credit card transactions. The proposed technique utilizes a cryptographic code and Arithmetic Coding technique to hide data inside a cover image and then transmit it to the destination over the web network. The paper also provides an additional layer for enhanced security via the use of Hamming Code. A detailed study is made to compare our technique with other prominent techniques like DCT and DWT.**

Keywords— **Cryptography, Arithmetic Coding, Bit Plane Substitution, Hamming Code, DWT, Digital Image Watermarking.**

I. Introduction

Embodying a part of Cryptography, the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity is defined as Steganography. In digital Steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocols. Media files are ideal for steganographic transmission because of their large size. One of the digital Steganographic techniques includes concealing messages within lowest bits of images or media files. The robustness of the encryption depends upon the technique used to encode the stego-data. The main objective of making Steganographic encoding difficult to detect is to ensure that the changes to the cover image by the injection of the embedded image is negligible.

There exist many water marking techniques for information hiding using DCT, DFT, Wavelet transforms [1] [2], but our innovation uses lossless code [3] for hiding a data (credit card information) as a watermark. We use arithmetic coding along with cryptography to encrypt the entire credit card data and then embed it onto the image.

Arithmetic coding unlike variable-length codes generates non-block codes. In arithmetic coding [3], a one to one correspondence between source symbols and code words does not exist. An entire sequence of source symbols (or message) is assigned a single arithmetic code word. The code itself defines the interval of real numbers between 0 and 1. As the number of symbols in the message increases, the interval used to represent it becomes smaller and the number of information units (say, bits) required to represent the interval becomes large, thus making it more secure to transmit data. Each symbol of the message reduces the size of the interval in accordance with its probability of occurrence.

In general, each step of the encoding process, except for the very last, is the same. The encoder has basically just three pieces of data to consider: the next symbol that needs to be encoded, the current interval (at the very start of the encoding process, the interval is set to [0, 1]), but that will change) and the probabilities the model assigns to each of the various symbols.

The encoder divides the current interval into sub-intervals, each representing a fraction of the current interval proportional to the probability of that symbol in the current context. Whichever interval corresponds to the actual symbol that is next to be encoded becomes the interval used in the next step. The final subinterval gives the arithmetic code for the input data sequence.

While modern steganography is growing increasingly diverse, steganography is fundamentally based on problems that are difficult to solve. A problem may be difficult because its solution requires some secret knowledge, such as decrypting an encrypted message or signing some digital document. The problem may also be hard because it is intrinsically difficult to complete, such as finding a message that produces a given hash value. The field of cryptography to which steganography is encompassed, embraces other uses as well. With just a few basic cryptographic tools, it is possible to build elaborate schemes and protocols that allow us to pay using electronic money [4]. Here we have used one cryptographic tool to build a powerful and reliable protocol to hide the credit card information from the intruder.

Hamming codes [5] are error detection-correction codes, detecting to two simultaneous bit errors, and correcting single-bit errors.Mathematically, they are linear block codes. Although primarily used in communication, Hamming codes also serve the purpose of introducing Data Redundancy, extending their use into Staganography.

A bit plane of an image is a set of bits having the same position in the respective binary number. For example, for 16-bit data representation there are 16 bit-planes: the first bit-plane contains the set of the most significant bit and the 16th contains the least significant bit [6] [7]. The most significant bit-plane gives the roughest but the most critical approximation of values of a medium, and the lesser the number of the bit plane, the less is its contribution to the final stage. In bit plane substitution we substitute least significant bits with our desired data (now in binary), since these LSB's are considered to be visually redundant, it does not degrade the image.

The procedure for the encoding, decoding and bit plane substitution is described in section below.

II. METHODOLOGY

The experiment is executed in MATLAB 7.9 running on Windows platform.

A. At encoding site (Customer Name)

We have assumed that, in general, the length of a name on any credit card is limited to a maximum of 20 alphabets (including spaces). The user is prompted for the entering the name on the card (refer Figure 1).The encryption technique incorporated here is the conversion of credit card information (precisely the customer name on the card) to binary form. The algorithm employed is based on the alphabetical position of the alphabets in the string entered as customer name. A to Z has been numbered from 1 to 26 .So an alphabet in the customer name will correspond to its alphabetical position, in the code. The obtained number or position is converted to the 6 digit binary form, so as to keep whole stream of bits always an even number which will create more ambiguity for intruder. The process is iterative until whole string is scanned. This will form our cipher text. The invisibility is further enhanced by introducing redundancy to the cipher text in the form of Hamming code [8]. A hamming code word was generated for every character in the 'Customer Name' string. As mentioned previously, each character was encoded as a 6 bit binary word. Using the formula,

$$2^r \geq m + r + 1 \qquad (1)$$

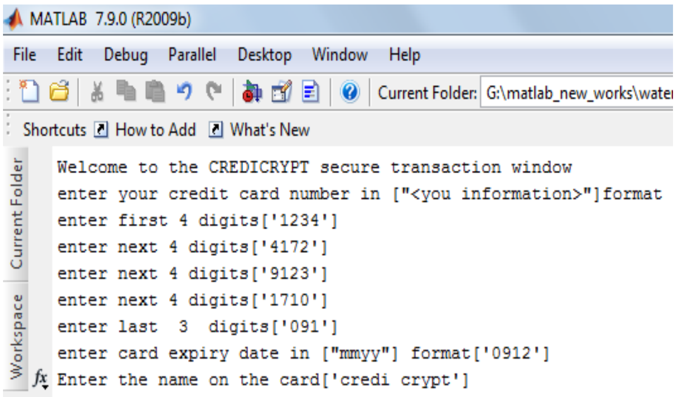
Where r is the number of redundant bits, m is the length of data, the number of redundant bits for the 6 bit cipher text is found out to be 10 bits. Hence using a (10, 4) Hamming generator matrix, hamming codes for each of the character were generated. Hence now each character is a 10 bit data word.

Therefore, the total cipher text length was now 10* (length of name), making it always variable, further complicating the detect ability of the hidden data. The length of the string was also converted into a (11, 4) hamming code before embedding it into the cover image. Hence now the data to be embedded onto the cover image is the 'Customer name' string with each character now a 10 bit data word and the length of the string which was an 11 bit data word.

B. At encoding site (For Customer Card Number)

For encryption of the credit card number we have assumed a credit card number has a 16 digit unique number in addition to the significant 3 digit CVV number which is necessary for all online transactions. The algorithm also encrypts the expiry date in (mmyy format- 4 digits), which is vital for internet transactions.

We take these 19+4 digits separately as 4+4+4+4+3+4 blocks of data and encode them separately [9]. The user inputs the numbers in a sequential manner as prompted by the program in the format as shown in the Figure 1.

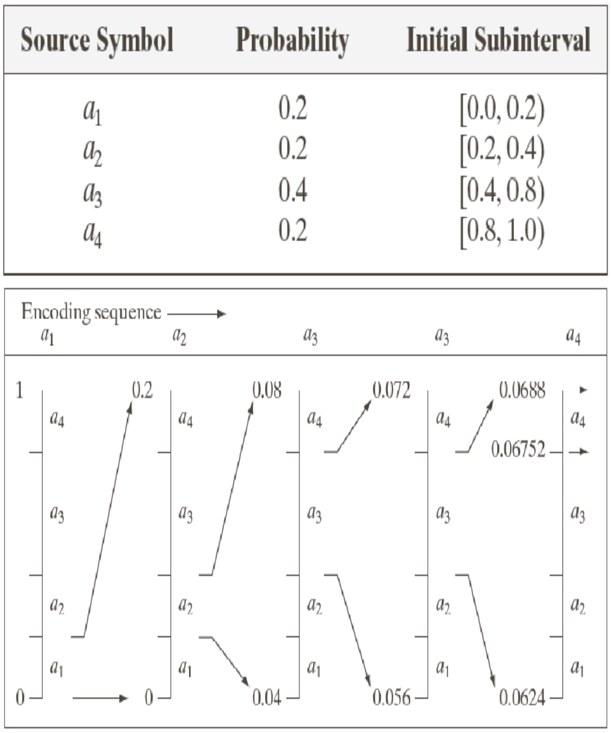

MATLAB 7.9.0 (R2009b)
File Edit Debug Parallel Desktop Window Help
Current Folder: G:\matlab_new_works\wate
Shortcuts
How to Add
What's New
Current Folder
Workspace
fx

```
Welcome to the CREDICRYPT secure transaction window
enter your credit card number in ["<you information>"]format
enter first 4 digits['1234']
enter next 4 digits['4172']
enter next 4 digits['9123']
enter next 4 digits['1710']
enter last  3  digits['091']
enter card expiry date in ["mmyy"] format['0912']
Enter the name on the card['credi crypt']
```

Figure 1. The input data user interface.

The symbol set for this experiment is (0-9) and the corresponding symbol probabilities are as found from large data sample space. The arithmetic encoder takes these probabilities and the symbol set as well as the input sequence and outputs the lower range of the final subinterval. An example of arithmetic coding is shown below in Figure 2. After the arithmetic coding process similar to the above shown example we obtain six double precision values corresponding to the six input sequences. We round these to six significant digits and then convert it to binary using our encoding algorithm. Table 1 illustrates the results of the encoded procedure for a sample input. This code word, along with the cipher text obtained in (A) can now be embedded into the

'LENA' image (refer Figure.3) and the image is compressed and then sent to the receiver.



Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

Figure 2. An example of arithmetic coding [18].

TABLE 1. STEPS TO ENCODE A CREDIT CARD NUMBER

Input data sequence	1234
Corresponding code word	0.161389592813131
Code word in six significant digit	161390
Equivalent 20-bit binary	00100111011001101110

Therefore the entire credit card information has been embedded in the pixels of the image (refer Figure 4). Various level of security level has been integrated and applied on the customer information in order to maintain security services like data confidentiality and data integrity. Thus a three level security scheme is provided.

Figure 3. An original LENA image used for embedding data (1024*1024).

Figure 4. The Embedded Image (1024*1024).

C. At decoding site (Customer Name)

After receiving the embedded image, the receiver will carry out the decryption process using decoding software which employs the exact reverse process. Firstly, the length of the 'Customer name' string is recovered. On the basis of that, remaining data bits are recovered from the image following the reverse algorithm.

Next, the redundant bits are removed with the help of the same hamming code matrix. The extracted code word will be delivered to the decoder. The decoder will go on extracting 6 contiguous bits from the received steam of bits (which will be multiple of 6 always) and convert it to decimal form whose range will be 1 to 26. Then decimal data will be mapped to the position of alphabet and corresponding alphabet will be included in the output string. This recursive process will fetch the original customer information (the printed name) to the receiver. The decoder now proceeds to decode the card number.

D. At the decoding side/receiver side (For Customer Card Number)

The receiver must have our decoding algorithm. In addition to these the symbol probability, symbol set and the length of the sequence must be known. If a password was set at the encoding end then the user must enter the same here. The bits are obtained using our reverse algorithm.

TABLE 2. STEPS TO DECODE A CREDIT CARD NUMBER

Binary data from image after concatenation	00100111011001101110
Corresponding decimal number	161390
Decimal number after division by 10^6	0.161390
Obtained credit card number	1234

Table 2 illustrates the decoding procedure for the input data in

Table 1. The final decoded credit card information is shown in Figure 5.

III. PERFORMANCE EVALUATION

We evaluated the performance of our algorithm using 1024 * 1024 'LENA' image as a cover image. The performance of the algorithm was measured using the two metrics given below.

A. Imperceptibility [10]

Imperceptibility is the measure of the perceived quality of the image after the addition of the watermark. For this purpose the statistics of Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) are used. The PSNR is given in equation 2.

$$PSNR_{dB} = 10log_{10}\frac{MAX_I^2}{MSE} \qquad (2)$$

The PSNR for our embedded image was calculated using equation 2.

B. Robustness [10]

Robustness is the measure of the immunity of watermark against an attempt to remove or degrade it using different digital signal processing attacks. One of the most common is JPEG compression, which is a watermark removal attack [12]. The similarity between the attacked image and the original image can be found using the correlation factor given in equation 3.

$$\rho = \frac{\sum_{i=1}^{n} w(i)*w'(i)}{\sqrt{\sum_{i=1}^{n} w(i)^2 * \sqrt{\sum_{i=1}^{n} w\prime(i)^2}}} \qquad (3)$$

where n is the number of pixels in the image, w is the original image and w' is the attacked image.

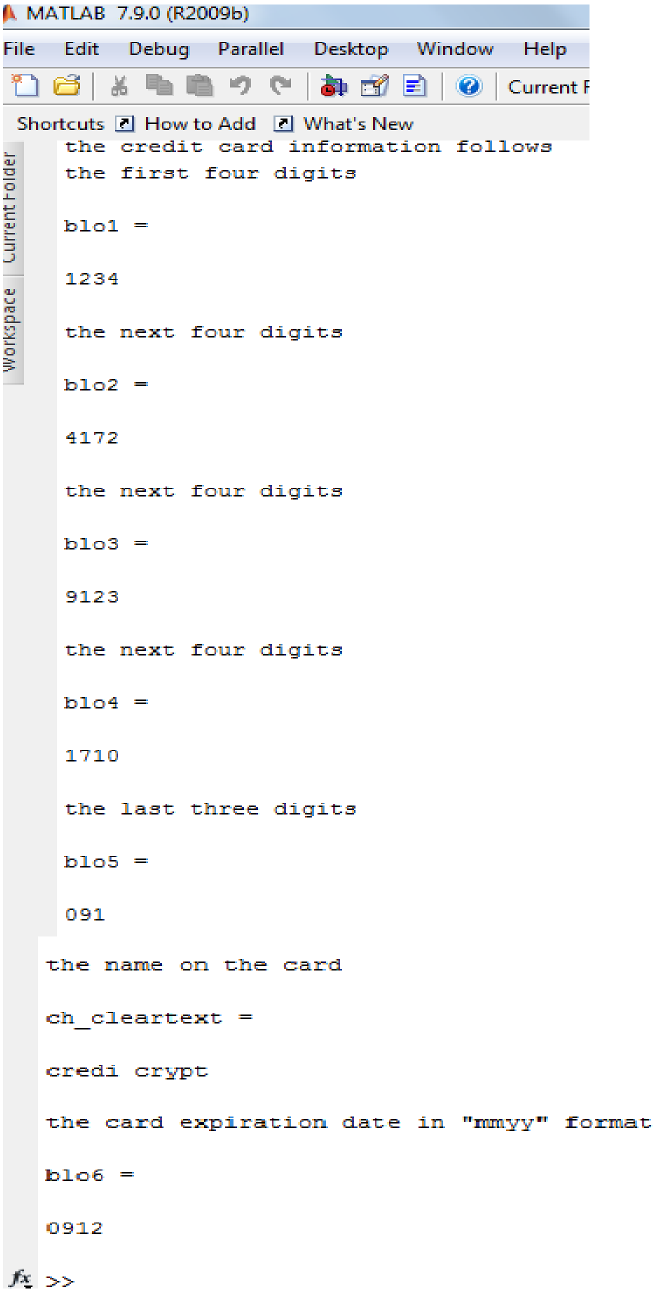
The results of the performance evaluation are given in Table 3 and Table 4 respectively. The results indicate a significant improvement in the evaluation parameters (particularly PSNR) which translate into improved security for the encrypted data.

TABLE 3. MSE AND PSNR VALUES FOR VARIOUS WATERMARKING TECHNIQUES

	Credi crypt	DCT	DWT(LL2)
Mean Square Error (MSE)	$5.655*10^{-4}$	$3.4*10^{-3}$	$1.39*10^{-4}$
PSNR	80.640	72.8500	76.4459

TABLE 4. CORRELATION VALUES FOR JPEG COMPRESSION ATTACKS (QUALITY = 80).

	Credi crypt	DCT	DWT(LL)
Correlation Factor	0.9987	0.98143	0.96279


MATLAB 7.9.0 (R2009b)
File Edit Debug Parallel Desktop Window Help
Current F
Shortcuts How to Add What's New
Current Folder
Workspace

```
the credit card information follows
the first four digits

blo1 =

1234

the next four digits

blo2 =

4172

the next four digits

blo3 =

9123

the next four digits

blo4 =

1710

the last three digits

blo5 =

091

the name on the card

ch_cleartext =

credi crypt

the card expiration date in "mmyy" format

blo6 =

0912

>>
```

Figure 5. Decoded Credit card information.

IV. Conclusions

We have basically conceptualized and implemented a reliable system which will revolutionalize the way sensitive data are being transferred via internet. We have been able to achieve all our objectives which were set for this system and also overcome challenges posed in the way successfully. The concept which we have produced is tested using Matlab 7.9. We have compared our technique with other reported methods [1] [2] and the results show a significant improvement in imperceptibility and robustness over other techniques.

The technique is inexpensive and can be efficiently used within a group of systems. Here we have also tried to compare our lossless techinque with the lossy watermarking techniques like DWT, DCT to show that this techinque can be effectively used in future applications of credit card information security. Also looking at a bigger picture, it can be used as a technique which can be employed to counter unauthorized access of sensitive information over the internet. Thus we believe that this paper in its own way will make a significant contribution in the field of e-commerce.

The application of this system is enormous and not limited to e-commerce. However, we have started its use using credit card information. While exploring other applications many challenges may arise but, taking a leaf out of the famous quote

> "Technology always calls for invention. Man always calls for innovation. When both these factors try to fulfill each other the result is always ingenuity."

References

[1] Ali Al-Haj "Combined DWT-DCT Digital Image Watermarking". Journal of Computer Science 3(9): 740- 746, 2007.

[2] Shikha Tripati, R.C. Jain *et al*, "Novel DCT and DWT based Watermarking Techniques for Digital Images". 18th International Conference on Pattern Recognition.

[3] Ian H Witten, Radford M Neal, john G Clemens "Arithmetic Coding for Data Compression".

[4] Hegde, C.; Manu, S.; Shenoy, P.D.; Venugopal, K.R.; Patnaik, L.M. "Secure Authentication using Image Processing and Visual Cryptography for Banking Applications", Advanced Computing and Communications, 2008. ADCOM 2008. 16th International Conference on; Publication Year: 2008 , Page(s): 65 – 72

[5] Chin-Chen Chang; The Duc Kieu; Yung-Chen Chou; "A High Payload Steganographic Scheme Based on (7, 4) Hamming Code for Digital Images", Electronic Commerce and Security, 2008 International Symposium on., Publication Year: 2008 , Page(s): 16 – 21

[6] Hang Min – Sun, King Hang Wang, Chih- Chang Liang, Yih Sein Kao, "A LSB Compatible Steganography".

[7] Chan, C. and L. Cheng, 2004. Hiding Data in Images by simple LSB substitution, Pattern Recognition, 37(3):469 – 474.

[8] Zhao-Xia Yin; Chin-Chen Chang; Yan-Ping Zhang; "A High Embedding Efficiency Steganography Scheme for Wet Paper Codes" Information Assurance and Security, 2009. IAS '09. Fifth International Conference on Volume: 2 Publication Year: 2009 , Page(s): 611 - 614

[9] Ravishankar, S; Hariharan, Santosh and Kumar, Naveen V.; 'A Reliable Anti- counterfeiting technique using Lossless Code',Proc. Of The 2010 International conference on Image Processing ,Computer Vision and Pattern Recognition (IPCV '10).

[10] Ejima M. and A. Myzaki, 2001. "On the evaluation of performance of digital watermarking in the frequency domain," in Proc. Of the IEEE Int. Conf. on Image Processing, 2: 546-549.

[11] Rao, K. and P. Yip. Discrete Cosine Transform: Algorithms, advantages, applications. Academic Press, USA, 1990.

[12] Voloshynovskiy, S., S.Pereira and T. Pun, 2001. "Attacks On digital watermarks: Classification, Estimation-based Attacks, and benchmarks," Comm. Magazine, 39(8): 118-126.

[13] Hyungjin Kim, Jiangtao Wen, Villasenor, J.D.; "Secure Arithmetic Coding", Signal Processing, IEEE Transactions on ,Publication Year: 2007 , Page(s): 2263 – 2272.

[14] Yu-Chee Tseng; Yu-Yuan Chen; Hsiang-Kuang Pan, "A secure data hiding scheme for binary images"; Communications, IEEE Transactions on ,Publication year: 2002

[15] Chin-Chen Chang; Yen-Chang Chen; Chia-Chen Lin; ,"A Resistant Secret Sharing Scheme" , Information Assurance and Security, 2009. IAS '09. Fifth International Conference on, Publication Year: 2009 , Page(s): 61 – 64

[16] Boris Shimanovsky, Jessica Feng and Miodrag Potkonjak, "Hiding Data in DNA".

[17] Mohanty, S.P.; Sheth, R.; Pinto, A.; Chandy, M.; "CryptMark: A Novel Secure Invisible Watermarking Technique for Color Images", Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on, Publication Year: 2007.

[18] R.C.Gonzalez, R.E .Woods. "Digital Image Processing", 2nd edition, Prentice Hall, Upper Saddle River, NewJersey: McGraw-Hill, 1964, pp. 409–510.