ISSN:0975-9646
BODDAPU SUNITHA et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (5) , 2015, 4575-4581

A Novel Secured Data Exchange Mechanism by using Captcha images

BODDAPU SUNITHA
II nd year M.tech,
, Department of CSE,VIET

SHALINI BHARIDE MTECH
Assistant Professor,
Department of CSE,VIET

Abstract: Many security primitives are based on hard mathematical problems. Using hard AI problems for security is emerging as an exciting new paradigm, but has been under-explored. In this paper, we present a new security primitive based on hard AI problems, namely, a novel family of graphical password systems built on top of Captcha technology, which we call Captcha as graphical passwords (CaRP). CaRP is both a Captcha and a graphical password scheme. CaRP addresses a number of security problems altogether, such as online guessing attacks, relay attacks, and, if combined with dual-view technologies, shoulder-surfing attacks. Notably, a CaRP password can be found only probabilistically by automatic online guessing attacks even if the password is in the search set. CaRP also offers a novel approach to address the well-known image hotspot problem in popular graphical password systems, such as PassPoints that often leads to weak password choices. CaRP is not a panacea, but it offers reasonable security and usability and appears to fit well with some practical applications for improving online security. We are embedding the text into the image orthogonally after that we encode the image and encrypt the image, then we send that captcha to the receiver.

Keyword: captcha,encryption, decryption, encoding, Decoding, embedding.

1. INTRODUCTION

Many security primitives are based on hard mathematical problems. Using hard AI problems for security is emerging as an exciting new paradigm, but has been under- explored. In this paper, we present a new security primitive based on hard AI problems, namely, a novel family of graphical password systems built on top of Captcha technology, which we call Captcha as graphical passwords (CaRP). CaRP is both a Captcha and a graphical password scheme. CaRP addresses a number of security problems altogether, such as online guessing attacks, relay attacks, and, if combined with dual-view technologies, shoulder-surfing attacks. Notably, a CaRP password can be found only probabilistically by automatic online guessing attacks even if the password is in the search set. CaRP also offers a novel approach to address the well-known image hotspot problem in popular graphical password systems, such as PassPoints that often leads to weak password choices. CaRP is not a panacea, but it offers reasonable security and usability and appears to fit well with some practical applications for improving online security.

We consider the problem of embedding one signal (e.g., a digital watermark), within another "host" signal to form a third, "composite" signal. The embedding is designed to achieve efficient tradeoffs among the three conflicting goals of maximizing information-embedding rate, minimizing distortion between the host signal and composite signal, and maximizing the robustness of the embedding. We introduce new classes of embedding methods, termed quantization index modulation (QIM) and distortion-compensated QIM (DC-QIM), and develop convenient realizations in the form of what we refer to as dither modulation. Using deterministic models to evaluate digital watermarking methods, we show that QIM is "provably good" against arbitrary bounded and fully informed attacks, which arise in several copyright applications, and in particular, it achieves provably better rate distortion–robustness tradeoffs than currently popular spread-spectrum and low-bit(s) modulation methods. Furthermore, we show that for some important classes of probabilistic models, DC-QIM is optimal (capacity-achieving) and regular QIM is near-optimal. These include both additive white Gaussian noise (AWGN) channels, which may be good models for hybrid transmission applications such as digital audio broadcasting, and mean-square-error-constrained attack channels that model private-key watermarking applications.

Number of applications have emerged recently that require the design of systems for embedding one signal, sometimes called an "embedded signal" or "watermark," within another signal, called a "host signal." The embedding must be done such that the embedded signal is "hidden," i.e., causes no serious degradation to its host. At the same time, the embedding must be robust to common degradations of the watermarked signal—the watermark must survive whenever the host signal does. In some applications these degradations are the result of benign processing and transmission; in other cases they result from deliberate attacks.

Several of these applications relate to copyright notification and enforcement for audio, video, and images that are distributed in digital formats. In these cases, the embedded signal either notifies a recipient of any copyright or licensing restrictions or inhibits or deters unauthorized copying. For example, this embedded signal could be a digital "fingerprint" that uniquely identifies the original purchaser of the copyrighted work. If illicit copies of the work were made, all copies would carry this fingerprint, thus identifying the owner of the copy from which all illicit copies were made. In another example, the embedded signal could either enable or disable copying by some duplication device that checks the embedded signal before proceeding with duplication. Such a system has been proposed for allowing a copy-once feature in digital video disc recorders .Alternatively, a standards-compliant player could check the watermark before deciding whether or not to play the disc .

Other applications include automated monitoring of airplay of advertisements on commercial radio broadcasts. Advertisers can embed a digital watermark within their ads and count the number of times the watermark occurs during a given broadcast period, thus ensuring that their ads are played as often as promised. In other applications, the embedded signal may be used for authentication of or detection of tampering with the host signal. For example, a digital signature could be embedded in a military map. A number of other national security applications are described and include covert communication, sometimes called "steganography" or low probability of detection communication, and so-called traitor tracing, a version of the digital fingerprinting application described above used for tracing the source of leaked information.

One final application for which the digital watermarking methods developed in this paper are well-suited is the backward-compatible upgrading of an existing communication system, an example of which is the so-called hybrid in-band on-channel digital audio broadcasting .In this application, one would like to simultaneously transmit a digital signal with existing analog (AM and/or FM) commercial broadcast radio without interfering with conventional analog reception. Thus, the analog signal is the host signal and the digital signal is the watermark. Since the embedding does not degrade the host signal too much, conventional analog receivers can demodulate the analog host signal. next-generation digital receivers can decode the digital signal embedded within the analog signal, which may be all or part of a digital audio signal, an enhancement signal used to refine the analog signal, or simply supplemental information such as station identification or traffic information. More generally, the host signal in these hybrid transmission systems could be some other type of analog signal such as video or even a digital waveform—for example, a digital pager signal could be embedded within a digital cellular telephone signal.

In general, designers of information embedding systems for these kinds of applications seek to achieve high embedding rates with high levels of robustness and low levels of embedding-induced distortion. However, in general, these three goals are conflicting. Thus, in this paper we characterize methods in terms of the efficiency with which they trade off rate, distortion, and robustness. For instance, for any minimum embedding rate requirement and maximum acceptable level of embedding distortion, the more efficient an embedding method is, the higher the robustness that can be achieved.

Data Hiding in Binary Image:

We proposes a new method to embed data in binary images, including scanned text, figures, and signatures. The method manipulates "flippable" pixels to enforce specific block based relationship in order to embed a significant amount of data without causing noticeable artifacts. Shuffling is applied before embedding to equalize the uneven embedding capacity from region to region. The hidden data can be extracted without using the original image, and can also be accurately extracted after high quality printing and scanning with the help of a few registration marks. The proposed data embedding method can be used to detect unauthorized use of a digitized signature, and annotate or authenticate binary documents. The paper also presents analysis and discussions on robustness and security issues.

Watermarking and data hiding techniques have been proposed for a variety of digital media applications, including ownership protection, copy control, annotation, and authentication. Most prior works on image data hiding are for color and grayscale images in which the pixels take on a wide range of values .For most pixels, changing the pixel values by a small amount may not be noticeable under normal viewing conditions. This property of human visual system plays a key role in watermarking of perceptual data .For images in which the pixels take value from only a few possibilities, hiding data without causing visible artifacts becomes more difficult. In particular, flipping white or black pixels that are not on the boundary is likely to introduce visible artifacts in binary images.

Hiding data in binary image, though difficult, is getting higher demands from our everyday life. An increasingly large number of digital binary images have been used in business. Handwritten signatures captured by electronic signing pads are digitally stored and used as records for credit card payment by many department stores and for parcel delivery by major courier services such as the United Parcel Service (UPS). Word processing software like Microsoft Word allows a user to store his/her signature in a binary image file for inclusion at specified locations of an electronic document. The documents signed in such a way can be sent directly to a fax machine or be distributed across a network. The unauthorized use of a signature, such as copying it onto an unauthorized payment, is becoming a serious concern. In addition, a variety of important documents, such as social security records, insurance information, and financial documents, have also been digitized and stored. Because of the ease to copy and edit digital images, annotation and authentication of binary images as well as detection of tampering are very important.

Data hiding techniques for these authentication and annotation purposes, possibly as an alternative to or in conjunction with the cryptographic authentication approach. Such targeted applications calls for fragile or semi fragile embedding of many bits. It should be stressed that while it is desirable for the embedded data to have some robustness against minor distortion and preferably to withstand printing and scanning, the robustness of embedded data against intentional removal or other obliteration is not a primary concern. This is because an adversary in authentication applications would have much more incentive to counterfeit valid embedded data than to remove them, and there is no obvious threat of removing embedded data in many annotation applications.

Several methods for hiding data in specific types of binary images have been proposed in literature. Embedded information in dithered images by manipulating the dithering patterns and in fax images by manipulating the run-lengths. Changed line spacing and character spacing to embed information in textual images for bulk electronic publications. These approaches cannot be easily extended

to other binary images and the amount of data that can be hidden is limited.

Data hiding algorithm which enforces the ratio of black versus white pixels in a block to be larger or smaller than 1. Although the algorithm aims at robustly hiding information in binary image, it is vulnerable to many distortions/attacks, and it is not secure enough to be directly applied for authentication or other fragile use. The number of bits that can be embedded is limited because the particular enforcing approach has difficulty in dealing with blocks that have low or high percentage of black pixels. However, the idea of enforcing properties of a group of pixels via the local manipulation of a small number of pixels can be extended as a general framework of data embedding. Another approach of marking a binary document by treating a binary image as a grayscale one and manipulating the luminance of dark pixels slightly so that the change is imperceptible to human eyes yet detectable by scanners. This approach, targeted at intelligent copier systems, is not applicable to bi-level images hence is beyond the scope of this paper. The bi-level constraint also limits the extension of many approaches proposed for grayscale or color images to binary images. For example, applying the spread spectrum embedding.

In a transform-domain additive approach to binary image would not only cause annoying noise on the black–white boundaries, but also have reduced robustness hence limited embedding capacity due to the post-embedding that ensures the marked image is still a bi-level one .For these additive embeddings, hiding a large amount of data and detecting without the original binary image is particularly difficult. In summary, these previously proposed approaches either cannot be easily extended to other binary images, or can only embed a small amount of data.

We propose a new approach that can hide a moderate amount of data in general binary images, including scanned text, figures, and signatures. The hidden data can be extracted without using the original unmarked image, and can also be extracted after high quality printing and scanning with the help of a few registration marks. The approach can be used to verify whether a binary document has been tampered with or not, and to hide annotation labels or other side information.

The paper is organized as follows. The proposed approach is presented in Section II and illustrated using three applications. Further discussions on robustness and security, including such issues as recovering hidden data from high quality printing-and-scanning. we suggests a few directions of future work.

2. METHODOLOGY

DES Algorithm:

High-level description of the algorithm

- Key Expansion—round keys are derived from the cipher key using Rijndael's key schedule
- Initial Round

1. AddRoundKey—each byte of the state is combined with the round key using bitwise xor

- Rounds

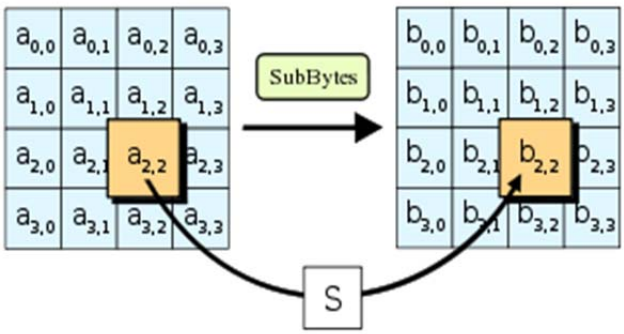
1. SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
2. ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
3. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column
4. AddRoundKey

Final Round (no MixColumns)

1. SubBytes
2. ShiftRows
3. AddRoundKey

The Sub Bytes step

In the SubBytes step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, S; $b_{ij} = S(a_{ij})$.In the SubBytes step, each byte in the array is updated using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over GF(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points.



The Shift Rows step

In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For the block of size 128 bits and 192 bits the shifting pattern is the same. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). In the case of the 256-bit block, the first row is unchanged and the shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively - this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks.

In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte

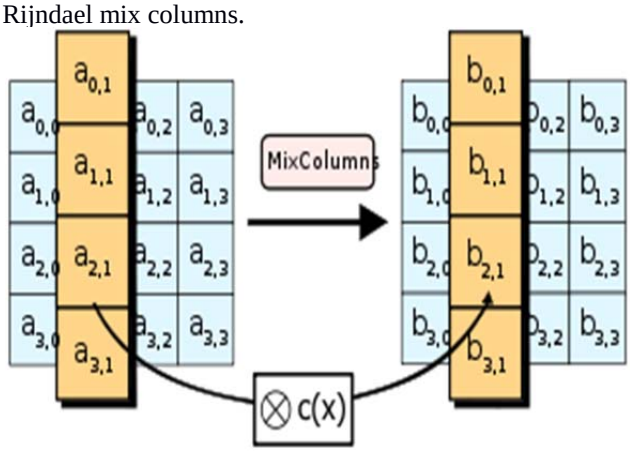
affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher.

During this operation, each column is multiplied by the known matrix that for the 128 bit key is

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}.$$

The multiplication operation is defined as: multiplication by 1 means leaving unchanged, multiplication by 2 means shifting byte to the left and multiplication by 3 means shifting to the left and then performing xor with the initial unshifted value.

In more general sense, each column is treated as a polynomial over GF(2^8) and is then multiplied modulo x^4+1 with a fixed polynomial $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from GF(2)[x]. The MixColumns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix columns.



The AddRoundKey step

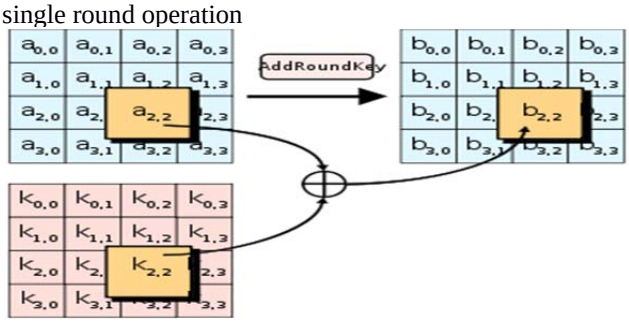
In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

Optimization of the cipher

On systems with 32-bit or larger words, it is possible to speed up execution of this cipher by combining SubBytes and ShiftRows with MixColumns, and transforming them into a sequence of table lookups. This requires four 256-entry 32-bit tables, which utilizes a total of four kilobytes (4096 bytes) of memory—one kilobyte for each table. A round can now be done with 16 table lookups and 12 32-bit exclusive-or operations, followed by four 32-bit exclusive-or operations in the AddRoundKey step

If the resulting four kilobyte table size is too large for a given target platform, the table lookup operation can be performed with a single 256-entry 32-bit (i.e. 1 kilobyte) table by the use of circular rotates.

Using a byte-oriented approach, it is possible to combine the Sub Bytes, Shift Rows, and Mix Columns steps into a single round operation

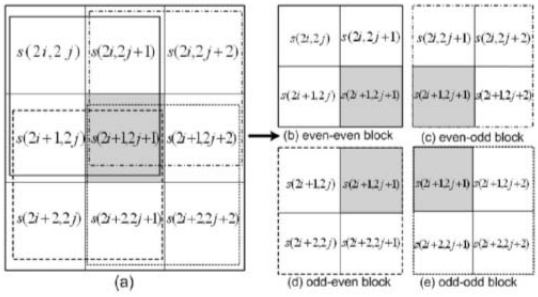


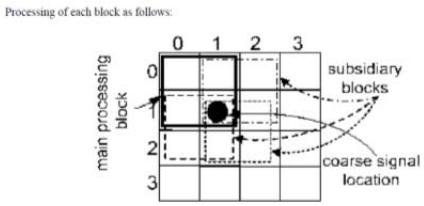
EMBEDDING ALOG0RITHM FOR CAPTCHA:

In this image divided into 3*3 blocks. After we divide the image into 2*2 blocks as shown in below:

These blocks are given below:

1. even-even block
2. even-odd block
3. odd-even block
4. odd-odd block





Sequence of operations to be performed for captcha image security & secured data with captcha image transmission

1) Image as input
2) Watermark embedding in captcha image
3) Authenticator Watermark
4) Swap Embedding in captcha image
5) Watermarked captcha Image

1) Image as input:

We give image as input, process an image in 2x2 pixel blocks. This allows flexibility in tracking the edges and also achieves high computational complexity. The two processing cases that flipping the candidates of one does not affect the *flippability* conditions of another are employed for *orthogonal embedding.*

2) Watermark embedding in captcha image:

Watermarking is a technology for embedding various types of information in digital content. In general, information for protecting copyrights and proving the validity of data is embedded as a watermark. Watermarked content can prove its origin, thereby protecting the data.

3) AuthenticatorWatermark:

In this module we encrypt the data embedded image. The purpose of authenticator watermark of a block is invariant in the watermark embedding process; hence the watermark can be extracted without referring to the original image .The encryption and decryption techniques used in this module.

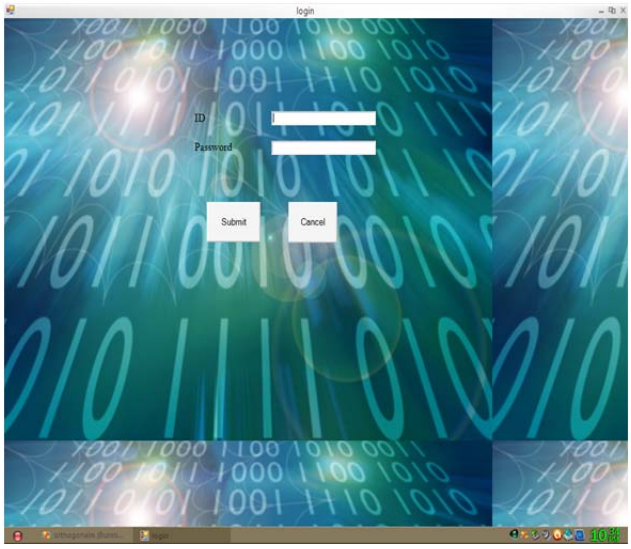
4) Swap Embedding in captcha image:

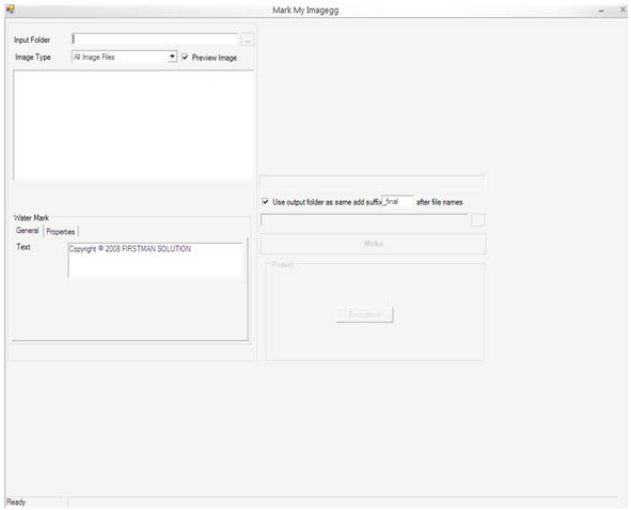
We flip an edge pixel in binary images is equivalent to shifting the edge location horizontally one pixel and vertically one pixel. A horizontal edge exists if there is a transition between two neighboring pixels vertically and a vertical edge exists if there is a transition between two neighboring pixels horizontally. We swap an morphological images.

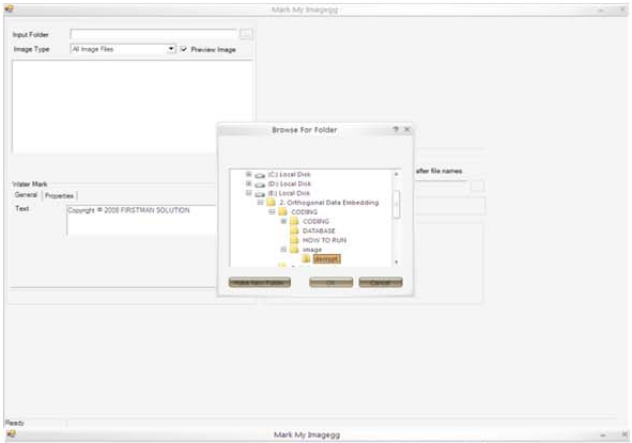
5) Watermarked captcha Image:

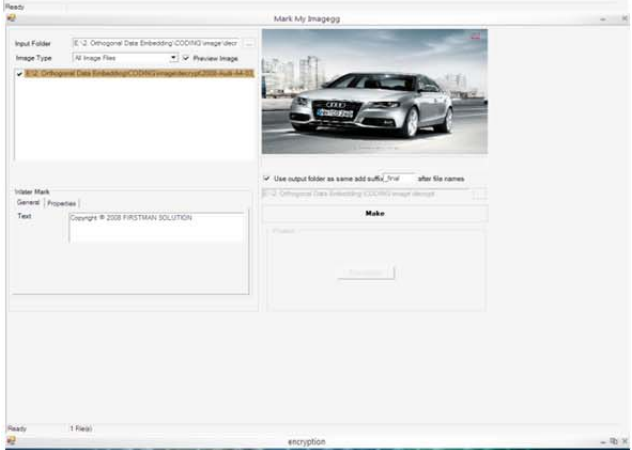
The watermarked image is obtained by computing the inverse for the main processing block to reconstruct its candidate pixels.use this module we going to see the original watermarked image.

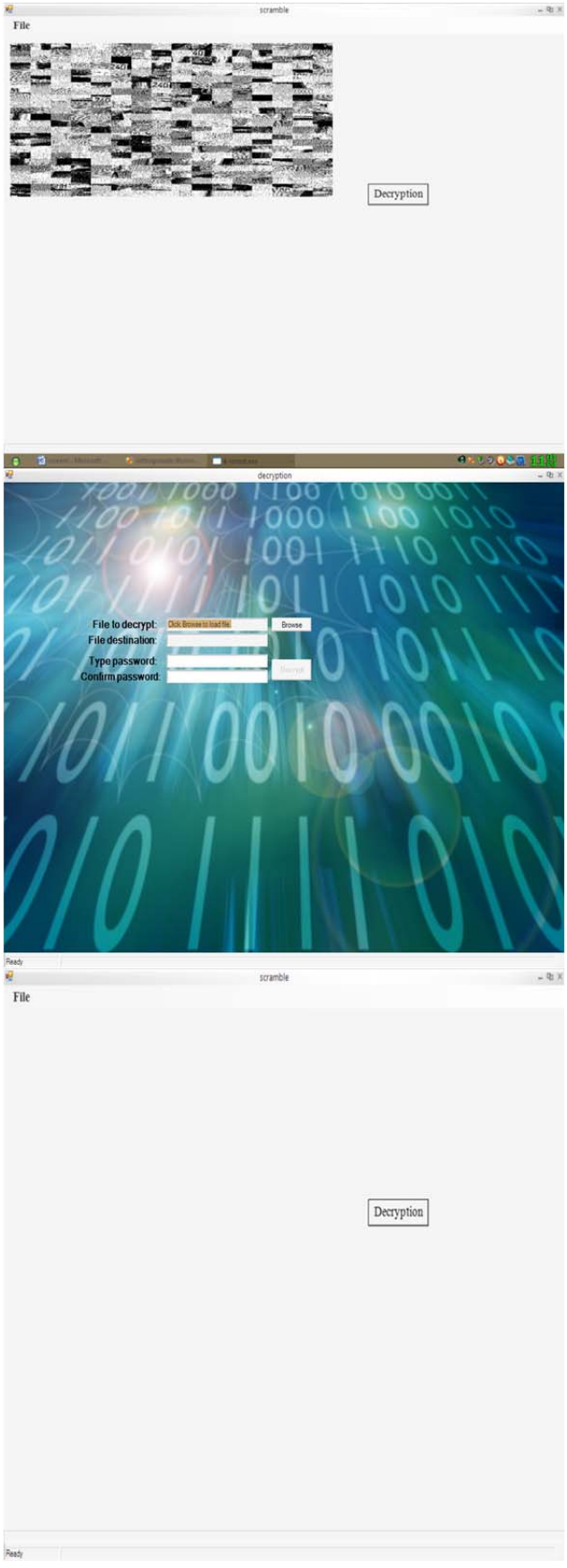
4. DATA ANALYSIS

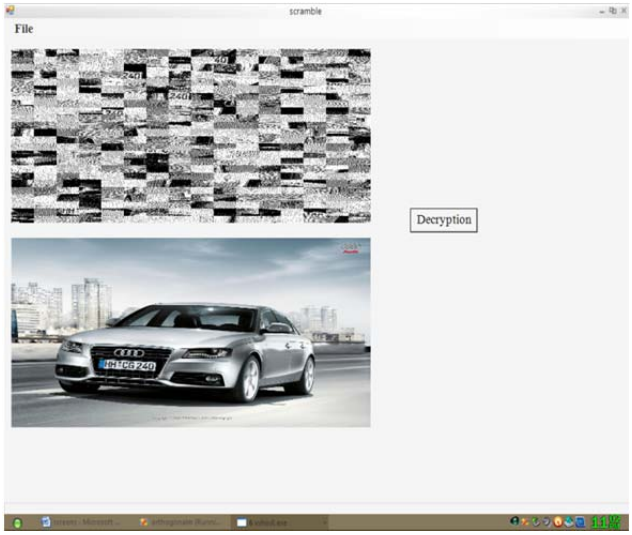












5. CONCLUSION

CaRP is both a Captcha and a graphical password scheme. The notion of CaRP introduces a new family of graphical passwords, which adopts a new approach to counter online guessing attacks: a new CaRP image, which is also a Captcha challenge, is used for every login attempt to make trials of an online guessing attack computationally independent of each other. A password of CaRP can be found only probabilistically by automatic online guessing attacks including brute-force attacks, a desired security property that other graphical password schemes lack. Hotspots in CaRP images can no longer be exploited to mount automatic online guessing attacks, an inherent vulnerability in many graphical password systems. CaRP forces adversaries to resort to significantly less efficient and much more costly human-based attacks. In addition to offering protection from online guessing attacks, CaRP is also resistant to Captcha relay attacks, and, if combined with dual-view technologies, shoulder-surfing attacks. CaRP can also help reduce spam emails sent from a Web email service.

Our usability study of two CaRP schemes we have implemented is encouraging. For example, more participants considered AnimalGrid and ClickText easier to use than PassPoints and a combination of text password and Captcha. Both AnimalGrid and ClickText had better password memorability than the conventional text passwords. On the other hand, the usability of CaRP can be further improved by using images of different levels of difficulty based on the login history of the user and the machine used to log in. The optimal tradeoff between security and usability remains an open question for CaRP, and further studies are needed to refine CaRP for actual deployments.

Like Captcha, CaRP utilizes unsolved AI problems. However, a password is much more valuable to attackers than a free email account that Captcha is typically used to protect. Therefore there are more incentives for attackers to hack CaRP than Captcha. That is, more efforts will be attracted to the following win-win game by CaRP than ordinary Captcha: If attackers succeed, they contribute to improving AI by providing solutions to open problems such

as segmenting 2D texts. Otherwise, our system stays secure, contributing to practical security. As a framework, CaRP does not rely on any specific Captcha scheme. When one Captcha scheme is
broken, a new and more secure one may appear and be converted to a CaRP scheme.Overall, our work is one step forward in the paradigm of using hard AI problems for security. Of reasonable security and usability and practical applications, CaRP has good potential for refinements, which call for useful future work. More importantly, we expect CaRP to inspire new inventions of such AI based security primitives.

REFERENCES

[1] Debra A. Lelewer, Daniel S. Hirschberg "Data Compression", ACM Computing Surveys (CSUR), vol 19, Issue 3, pp. 261 - 296, Sep. 1987

[2] William C. Barker, "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher", National Institute of Standards and Technology, NIST Special Publication 800-67, 2008.

[3] "Announcing the ADVANCED ENCRYPTION STANDARD (AES)", Federal Information Processing Standards Publication 197, Nov. 2001

[4] R.L. Rivest, "The RC5 encryption algorithm", Proceedings of the 1994 Leuven Workshop on Fast Software Encryption, pp. 86-96, Springer- Verlag, 1995.

[5] Ron Rivest, Adi Shamir and Len Adleman, "A method for obtaining Digital Signatures and Public Key Cryptosystems", Communications of the ACM, pp 120-126, Feb. 1978.

[6] Taher ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, v. IT-31, n. 4, 1985, pp.469472 or CRYPTO 84, pp.1018, Springer-Verlag.

[7] Elliptic Curve Cryptography, Certicom Research, 2000

[8] Huffman's original article: D.A. Huffman,"A Method for the Construction of Minimum-Redundancy Codes", Proceedings of the I.R.E., Sep.1952, pp.10981102

[9] Amit Jain, Ravindra Patel, "An Efficient Compression Algorithm (ECA) for Text Data", icsps, pp.762-765, 2009 International Conference on Signal Processing Systems, 2009

[10] Farina, A.; Navarro, G.; Parama, J.R., "Word-Based Statistical Compressors as Natural Language Compression Boosters", Data Compression Conference 2008, pp. 162 - 171, Mar. 2008 332

[11] Bin B. Zhu, Jeff Yan, GuanboBao, Maowei Yang and NingXu, Member, IEEE, ACM "Captcha as Graphical Passwords—A New Security Primitive Based on Hard AI Problems" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 6, JUNE 2011

[12] R. Biddle, S. Chiasson, and P. C. van Oorschot, "Graphical passwords: Learning from the first twelve years," ACM Comput. Surveys, vol. 44, no. 4, 2012. [2] (2012, Feb.).

[13] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, "The design and analysis of graphical passwords," in Proc. 8thUSENIX Security Symp., 1999, pp. 1–15.

[14] H. Tao and C. Adams, "Pass-Go: A proposal to improve the usability of graphical passwords," Int. J. Netw. Security, vol. 7, no. 2, pp. 273–292, 2008.

[15] S. Wiedenbeck, J. Waters, J. C. Birget, A. Brodskiy, and N. Memon, "PassPoints: Design and longitudinal evaluation of a graphical password system," Int. J. HCI, vol. 63, pp. 102–127, Jul. 2005.

[16] A Novel Dual Phase Mechanism for Data Transmission to Provide Compression and Security ,International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 12, December 2013 ISSN: 2277 128X

[17] A Novel Methodology for Secure Communications and Prevention of Forgery Attacks (0975 - 8887) ,International Journal of Computer Applications Volume 96 - Number 22 Year of Publication: 2014

[18] Communication within Sensor Networks by Using Key Distributor , International Journal of Computer Science and Information Technologies, Vol.5(4),2014,4906-4910