RESEARCH PAPER OPEN ACCESS

Artificial Intelligence: Concept of Security

Saroj Kumar [1], Santosh Kumar [2]
Maharishi University of Information Technology [1], *Lucknow*, IIM Road, Sitapur By-pass Lucknow
Maharishi University of Information Technology [2], *Lucknow*, IIM Road, Sitapur By-pass Lucknow

ABSTRACT

Artificial intelligence (AI) is the ability of a computer program or a machine to think and learn. It is also a field of study which tries to make computers "smart". In general use, the term "artificial intelligence" means a machine which mimics human cognition .At least some of the things we associate with other minds, such as learning and problem solving can be done by computers, though not in the same way as we do. Consciousness is only marginally relevant to artificial intelligence. AI is successful in finding computational solutions of difficult problems such as vision, language, locomotion and Security. Our main concern is to solve the problem of captcha security, an automated test that humans can pass, but current computer programs can't pass: any program that has high success over a captcha can be used to solve an unsolved Artificial Intelligence (AI) problem. We provide several novel constructions of captchas. Since captchas have many applications in practical security, our approach introduces a new class of hard problems that can be exploited for security purposes. Much like research in cryptography has a positive impact on algorithms for factoring and discrete log, we hope that the use of hard AI problems for security purposes allows us to advance the field of Artificial Intelligence.

Keyword:- Artificial intelligence – AI, Captcha

I. INTRODUCTION

A captcha is a cryptographic protocol whose underlying hardness assumption is based on an AI problem. A CAPTCHA (acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart") is a type of challenge-response test used in computing to determine whether or not the user is human. The term was coined in 2003 by Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. This form of CAPTCHA requires that the user type the letters of a distorted image, sometimes with the addition of an obscured sequence of letters or digits that appears on the screen. Because the test is administered by a computer, in contrast to the standard Turing test that is administered by a human, a CAPTCHA is sometimes described as a reverse Turing test. This user identification procedure has received many criticisms, especially from disabled people, but also from other people who feel that their everyday work is slowed down by distorted words that are difficult to read. It takes the average person approximately 10 seconds to solve a typical CAPTCHA.

A captcha is a program that can generate and grade tests that: (A) most humans can pass, but (B) current computer programs can't pass. Such a program can be used to differentiate humans from computers and has many applications for practical security.

i. **Online Polls**. In November 1999, slashdot.com released an online poll asking which was the best graduate school in computer science (a dangerous question to ask over the web!). As is the case with most online polls, IP addresses of voters were recorded in order to prevent single users from voting more than once. However, students at Carnegie Mellon found a way to stuff the ballots by using programs that voted for CMU thousands of times. CMU's score started growing rapidly. The next day, students at MIT wrote their own voting program and the poll became a contest between voting ”bots". MIT finished with 21,156 votes, Carnegie Mellon with 21,032 and every other school with less than 1,000. Can the result of any online poll be trusted? Not unless the poll requires that only humans can vote.

ii. **Free Email Services**. Several companies (Yahoo!, Microsoft, etc.) offer free email services, most of which suffer from a specific type of attack: \bots" that sign up for thousands of email accounts every minute. This situation can be improved by requiring users to prove they are human before they can get a free email account. Yahoo!, for instance, uses a captcha of our design to prevent bots from registering for accounts. Their captcha asks users to read a distorted word such as the one shown below (current computer programs are not as good as humans at reading distorted text).

Fig: Example of CAPTCHA[1]

iii. **Search Engine Bots**. Some web sites don't want to be indexed by search engines. There is an html tag to prevent search engine bots from reading web pages, but the tag doesn't guarantee that bots won't read the pages; it only serves to say \no bots, please". Search engine bots, since they usually belong to large companies, respect web pages that don't want to allow them in. However, in order to truly guarantee that bots won't enter a web site, captchas are needed.

iv. **Worms and Spam**. Captchas also offer a plausible solution against email worms and spam: only accept an email if you know there is a human behind the other computer. A few companies, such as www.spamarrest.com are already marketing this idea.

v. **Preventing Dictionary Attacks**. Pinkas and Sander [11] have suggested using captchas to prevent dictionary attacks in password systems. The idea is simple: prevent a computer from being able to iterate through the entire space of passwords by requiring a human to type the passwords.

The goals of this paper are to lay a solid theoretical foundation for captchas,to introduce the concept to the cryptography community, and to present several novel constructions.

Process of recognition of each alphabets or any characters in CAPTCHA

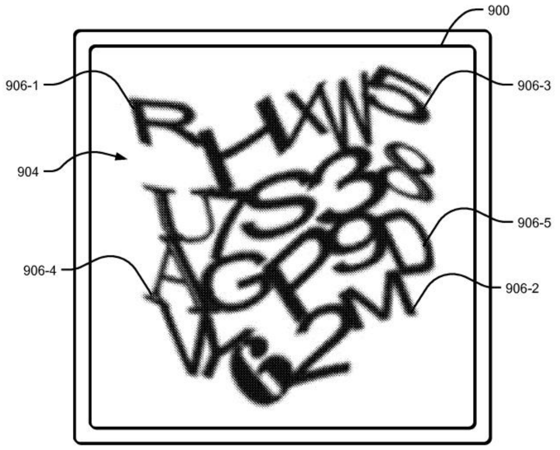

FIG: (2) Mapping of characters using coordinates (x,y)

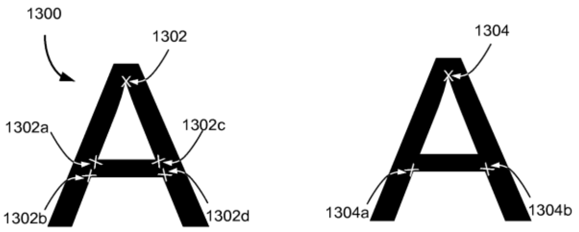

FIG: 3(a)

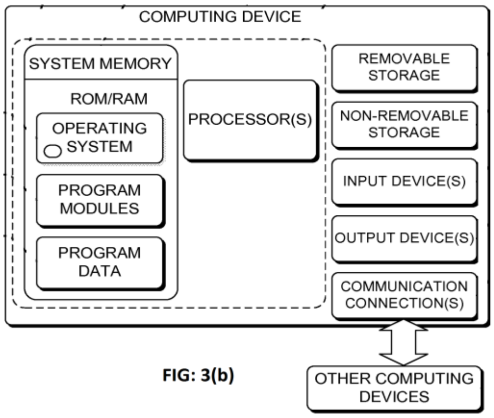

FIG: 3(b)

FIG: 3(a) Dimensions of a particular alphabet (A) with their coordinates and alignment axis
FIG: 3(b) Block Diagram showing the process of recognizing each captcha characters by the computer

II. RELATION TO AI

While used mostly for security reasons, CAPTCHAs also serve as a benchmark task for artificial intelligence technologies. According to an article by Ahn, Blum and Langford, “Any program that passes the tests generated by a CAPTCHA can be used to solve a hard unsolved AI problem.”

They argue that the advantages of using hard AI problems as a means for security are twofold. Either the problem goes unsolved and there remains a reliable method for distinguishing humans from computers, or the problem is solved and a difficult AI problem is resolved along with it. In the case of image and text based CAPTCHAs, if an AI were capable of accurately completing the task without exploiting flaws in a particular CAPTCHA design, then it would have solved the problem of developing an AI that is capable of complex object recognition in scenes.

i. Circumvention

There are a few approaches to defeating CAPTCHAs: using cheap human labor to recognize them, exploiting bugs in the implementation that allow the attacker to completely bypass the CAPTCHA, and finally using machine learning to build an automated solver. According to former Google click fraud czar Shuman Ghosemajumder, there are numerous criminal services which solve CAPTCHAs automatically.

ii. Machine learning-based attacks

In its earliest iterations there was not a systematic methodology for designing or evaluating CAPTCHAs. As a result, there were many instances in which CAPTCHAs were of a fixed length and therefore automated tasks could be constructed to successfully make educated guesses about where segmentation should take place. Other early CAPTCHAs contained limited sets of words, which made the test much easier to game. Still others made the mistake of relying too heavily on background confusion in the image. In each case, algorithms were created that were successfully able to complete the task by exploiting these design flaws. These methods proved brittle however, and slight changes to the CAPTCHA were easily able to thwart them. Modern CAPTCHAs like reCAPTCHA no longer rely just on fixed patterns but instead present variations of characters that are often collapsed together, making segmentation almost impossible. These newest iterations have been much more successful at warding off automated tasks.

In October 2013, artificial intelligence company Vicarious claimed that it had developed a generic CAPTCHA-solving algorithm that was able to solve modern CAPTCHAs with character recognition rates of up to 90%. However, Luis von Ahn, a pioneer of early CAPTCHA and founder of reCAPTCHA, expressed skepticism, stating: "It's hard for me to be impressed since I see these every few months." He pointed out that 50 similar claims to that of Vicarious had been made since 2003.

In August 2014 at Usenix WoOT conference Bursztein et al. presented the first generic CAPTCHA-solving algorithm based on reinforcement learning and demonstrated its efficiency against many popular CAPTCHA schemas. They concluded that text distortion based CAPTCHAs schemes should be considered insecure moving forward.

iii. Cheap or unwitting human labor

It may be possible to subvert CAPTCHAs by relaying them to a sweatshop of human operators who are employed to decode CAPTCHAs. A 2005 paper from a W3C working group stated that such an operator "could easily verify hundreds of them each hour".[7] In 2010 the University of UCSD conducted a large scale study of those CAPTCHA's farms and found out that the retail price for solving one million CAPTCHAs is as low as $1,000.

Another technique used consists of using a script to re-post the target site's CAPTCHA as a CAPTCHA to a site owned by the attacker, which unsuspecting humans visit and correctly solve within a short while for the script to use. However, there is controversy around the economic viability of such attack.

iv. Insecure implementation

Howard Yeend has identified two implementation issues with poorly designed CAPTCHA systems:

a) Some CAPTCHA protection systems can be bypassed without using OCR simply by reusing the session ID of a known CAPTCHA image

b) CAPTCHAs residing on shared servers also present a problem; a security issue on another virtual host may leave the CAPTCHA issuer's site vulnerable

Sometimes, if part of the software generating the CAPTCHA is client-side (the validation is done on a server but the text that the user is required to identify is rendered on the client side), then users can modify the client to display the un-rendered text. Some CAPTCHA systems use MD5 hashes stored client-side, which may leave the CAPTCHA vulnerable to a brute-force attack.

III. FLOW DIAGRAM OF CAPTCHA

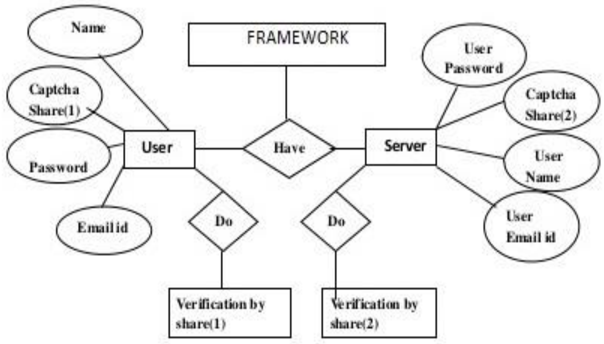

FIG:(4) ER – Diagram showing the working of CAPTCHA over the server

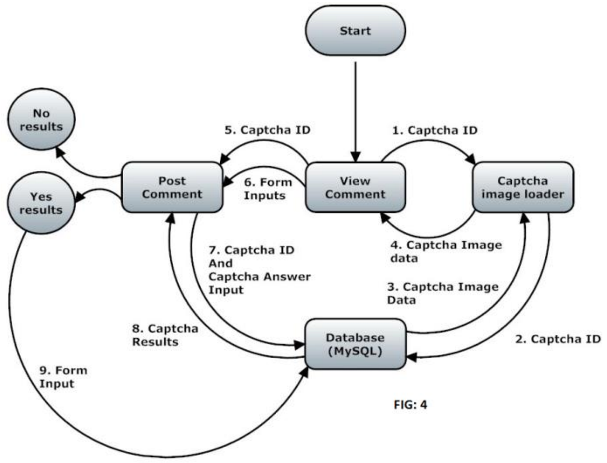

FIG: (5) Flow diagram of captcha processing .processing incudes checking of authorization and providing authentication

3.1 AUTHENTICATION PROCEDURE

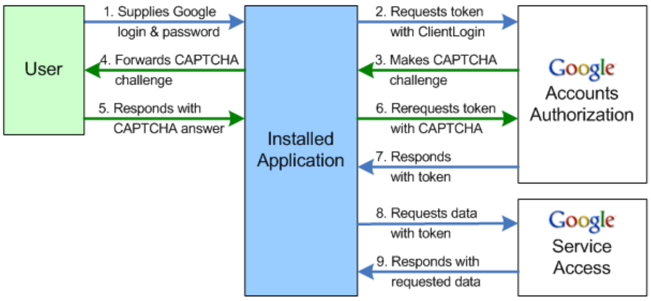
User
1. Supplies Google login & password
4. Forwards CAPTCHA challenge
5. Responds with CAPTCHA answer
Installed Application
2. Requests token with ClientLogin
3. Makes CAPTCHA challenge
6. Rerequests token with CAPTCHA
7. Responds with token
8. Requests data with token
9. Responds with requested data
Google Accounts Authorization
Google Service Access

FIG: (6) Authentication process performed by the Google, while verifying the client using reCAPTCHA

3.2 Concept of reCAPTCHA

reCAPTCHA is a CAPTCHA-like system designed to establish that a computer user is human (normally in order to protect websites from bots) and, at the same time, assist in the digitization of books. reCAPTCHA was originally developed by Luis von Ahn, Ben Maurer, Colin McMillen, David Abraham and Manuel Blum at Carnegie Mellon University's main Pittsburgh campus. It was acquired by Google in September 2009.reCAPTCHA has completed digitizing the archives of *The New York Times* and books from Google Books, as of 2011. The archive can be searched from the *New York Times* Article Archive, where more than 13 million articles in total have been archived, dating from 1851 to the present day. Through mass collaboration, reCAPTCHA was helping to digitize books that are too illegible to be scanned by computers, as well as translate books to different languages, as of 2015.

The system has been reported as displaying over 100 million CAPTCHAs every day, on sites such as Facebook, TicketMaster, Twitter, 4chan, CNN.com, StumbleUpon, Craigslist , and the U.S. National Telecommunications and Information Administration's digital TV converter box coupon program website (as part of the US DTV transition).

reCAPTCHA's slogan was **"Stop Spam, Read Books."**, until the introduction of a new version of the reCAPTCHA plugin in 2014; the slogan has now disappeared from the website and from the classic version of the reCAPTCHA plugin. A new system featuring image verification was also introduced. In this system, users are asked to just click on a checkbox (the system will verify whether the user is a human or not, for example, with some clues such as already-known cookies or mouse movements within the ReCAPTCHA frame) or, if it fails, select one or more images from a selection of nine images.

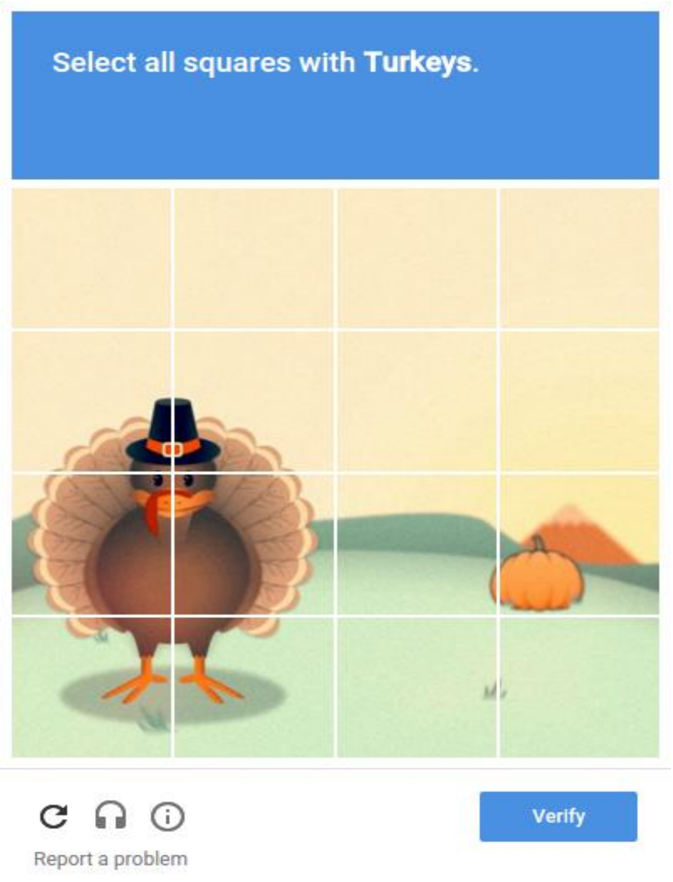
Select all squares with **Turkeys**.

FIG: (7) Example of reCAPTCHA

Solution to a perform a Complex Pattern for the reCAPTCHA ,which is hard to be crawled by the web crawler and will restrict the spam ware and bots to get the authentication without permission .this type of reCAPTCHA code will change their patterns in some interval of time to avoid the identification by the crawler's or limit the use of a particular CAPTCHA over a specific interval of time only (use and throw)

The main problem occurs in the traditional captcha authentication, is that it is easily traceable or identifiable by the spam ware and the bots. Whenever we want to login in the secured website than there is a field which is mandatory for the login process which includes the client details and a special secured field is asked by the client for there authentication called CAPTCHA Code which is type of characters and set of numbers or mathematical equations which have some answer .After filling all details with CAPTCHA by the client, it will submit the request .after submitting the request the client details and CAPTCHA are received by the server over which client want to login . firstly the server checks the captcha code and verify it with their database and if found ok then move to the client details and checks the details of user and if found ok then it give the authorization to their respective client.

But the problem is that the captcha used by the highly secured websites which have the server support used the captcha, and that captcha is repeated over a approximately 10 million verification which is easily crawled by the web crawler and make the website Vulnerable easily and here is the loop hole occurs in the website which is identified by the web crawler .this problem is very serious issues regarding the security of the authorized authentication.

To avoid this problem the researcher invented the concept of CAPTCHA which give the website a extra strength or security to the websites .but the traditional CAPTCHA code are found insufficient to provide the security to the websites .the problem with the traditional captcha is that it is repeatable ,it means it can be same captcha code for two different websites authentication at different time duration and another problem is that the characters used in the captcha code has some coordinates and alignment axis, which have a demerit for the security process which is identified by bots easily.

The solution for this problem is to make more complex and validate a particular CAPTCHA CODE for a specific time only after which it will automatically deleted form the servers database and never come into existence .by which we can make sure that the possibility of crawling of the CAPTCH CODE reduces and which result in securing the captcha from bots and spam ware .

Also we can make sure that the servers of the captcha provider will setup a different special database storage which is inconsistent and do not support replication of the database and must follow the concept of single data storage and will directly connected to the server and provide a single CAPTCHA pattern to a single user at a time .no multiple use of same captcha should me possible in this database which result is that it avoids the database crawling process and make them anti- crawl field.

Flow diagram of the solution–

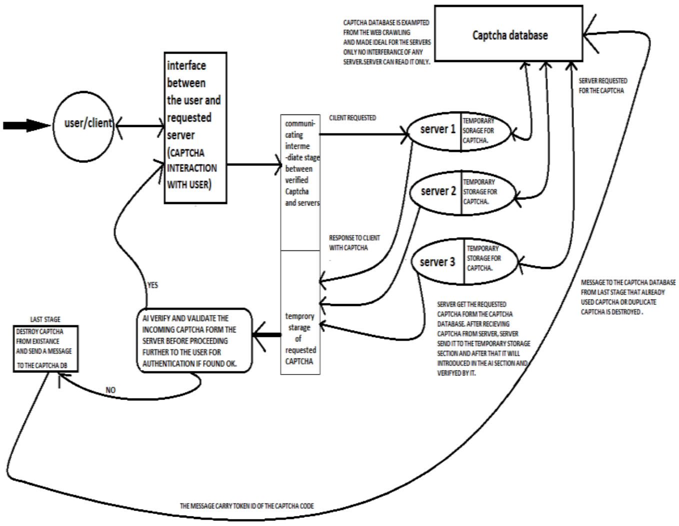
CAPTCHA DATABASE IS EXAMPTED FROM THE WEB CRAWLING AND MADE IDEAL FOR THE SERVERS ONLY NO INTERFERANCE OF ANY SERVER.SERVER CAN READ IT ONLY.
Captcha database
interface between the user and requested server (CAPTCHA INTERACTION WITH USER)
SERVER REQUESTED FOR THE CAPTCHA
user/client
communi-cating interme -diate stage between verified Captcha and servers
CILENT REQUESTED
server 1
TEMPORARY SORAGE FOR CAPTCHA.
server 2
TEMPORARY STORAGE FOR CAPTCHA.
RESPONSE TO CLIENT WITH CAPTCHA
server 3
TEMPORARY STORAGE FOR CAPTCHA.
MESSAGE TO THE CAPTCHA DATABASE FROM LAST STAGE THAT ALREADY USED CAPTCHA OR DUPLICATE CAPTCHA IS DESTROYED .
YES
SERVER GET THE REQUESTED CAPTCHA FORM THE CAPTCHA DATABASE. AFTER RECIEVING CAPTCHA FROM SERVER, SERVER SEND IT TO THE TEMPORARY STORAGE SECTION AND AFTER THAT IT WILL INTRODUCED IN THE AI SECTION AND VERIFYED BY IT.
LAST STAGE
DESTROY CAPTCHA FROM EXISTANCE AND SEND A MESSAGE TO THE CAPTCHA DB
AI VERIFY AND VALIDATE THE INCOMING CAPTCHA FORM THE SERVER BEFORE PROCEEDING FURTHER TO THE USER FOR AUTHENTICATION IF FOUND OK.
temprory starage of requested CAPTCHA
NO
THE MESSAGE CARRY TOKEN ID OF THE CAPTCHA CODE

FIG: (8) PROVISIONAL FLOW DIAGRAM OF SOLUTION FOR THE CAPTCHA PROBLEMS. THIS FLOW DIAGRAM SHOWS ALL THE MAJOR STEPS REGARDING PROCESSING OF CAPTCHA WITHOUT ANY VULNERABILITY. IT ALSO DEFINES THE NEW CHANGES,WHICH CAN RESOLVE THE PREVIOUS PROBLEMS VERY WELL

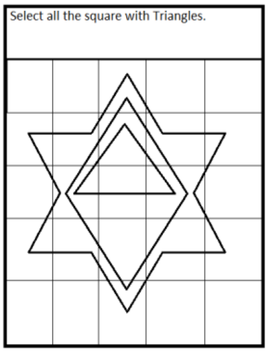

Fig: (9) concept of complex reCAPTCHA. This reCAPTCHA pattern is hard to identified by the web crawler

IV. IMPLEMENTATIONS

The reCAPTCHA tests are displayed from the central site of the reCAPTCHA project, which supplies the words to be deciphered. This is done through a JavaScript API with the server making a callback to reCAPTCHA after the request has been submitted. The reCAPTCHA project provides libraries for various programming languages and applications to make this process easier. reCAPTCHA is a free service (that is, the CAPTCHA images are provided to websites free of charge, in return for assistance with the decipherment), but the reCAPTCHA software itself is not open source.Also, reCAPTCHA offers plugins for several web-application platforms, like ASP.NET, Ruby, or PHP, to ease the implementation of the service.

1) **Security**

The main purpose of a CAPTCHA system is to prevent automated access to a system by computer programs or "bots". On 14 December 2009, Jonathan Wilkins released a paper describing weaknesses in reCAPTCHA that allowed a solve rate of 18%. On 1 August 2010, Chad Houck gave a presentation to the DEF CON 18 Hacking Conference detailing a method to reverse the distortion added to images which allowed a computer program to determine a valid response 10% of the time. The reCAPTCHA system was modified on 21 July 2010, before Houck was to speak on his method. Houck modified his method to what he described as an "easier" CAPTCHA to determine a valid response 31.8% of the time. Houck also mentioned security defenses in the system, including a high security lock out if an invalid response is given 32 times in a row.

On 26 May 2012, Adam, C-P and Jeffball of DC949 gave a presentation at the LayerOne hacker conference detailing how they were able to achieve an automated solution with an accuracy rate of 99.1%. Their tactic was to use techniques from machine learning, a subfield of artificial intelligence, to analyse the audio version of reCAPTCHA which is available for the visually impaired. Google released a new version of reCAPTCHA just hours before their talk, making major changes to both the audio and visual versions of their service. In this release, the audio version was increased in length from 8 seconds to 30 seconds, and is much more difficult to understand, both for humans as well as bots. In response to this update and the following one, the members of DC949 released two more versions of Stiltwalker which beat reCAPTCHA with an accuracy of 60.95% and 59.4% respectively. After each successive break, Google updated reCAPTCHA within a few days. According to DC949, they often reverted to features that had been previously hacked.

On 27 June 2012, Claudia Cruz, Fernando Uceda, and Leobardo Reyes (a group of students from Mexico) published a paper showing a system running on reCAPTCHA images with an accuracy of 82%. The authors have not said if their system can solve recent reCAPTCHA images, although they claim their work to be intelligent OCR and robust to some, if not all changes in the image database.

In an August 2012 presentation given at BsidesLV 2012, DC949 called the latest version "unfathomably impossible for humans" - they were not able to solve them manually either. The web accessibility organization WebAIM reported in May 2012, "Over 90% of respondents [screen reader users] find CAPTCHA to be very or somewhat difficult."

reCAPTCHA frequently modifies its system, requiring spammers to frequently update their methods of decoding, which may frustrate potential abusers.

Only words that both OCR programs failed to recognize are used as control words. Thus, any program that can recognize these words with non-eligible probability would represent an improvement over state of the art OCR programs.

2) **AI Problems as Security Primitives**

Notice that we define hard in terms of the consensus of a community: an AI problem is said to be hard if the people working on it agree that it's hard. This notion should not be surprising to cryptographers: the security of most modern cryptosystems is based on assumptions agreed upon by the community (e.g., we assume that 1024-bit integers can't be factored). The concept of a hard AI problem as a foundational assumption, of course, is more questionable than P 6= NP, since many people in the AI community agree that all hard AI problems are eventually going to be solved. However, hard AI problems may be a more reasonable assumption than the hardness of factoring, given the possibility of constructing a quantum computer. Moreover, even if factoring is shown to be hard in an asymptotic sense, picking a concrete value for the security parameter usually means making an assumption about current factoring algorithms: we only assume that current factoring algorithms that run in current computers can't factor 1024-bit integers. In the same way that AI researchers believe that all AI problems will be solved eventually, we believe that at some point we will have the computational power and algorithmic ability to factor 1024-bit integers. (Shamir and Tromer [13], for instance, have proposed a machine that could factor 1024-bit integers; the machine would cost about ten million dollars in materials.) An important difference between popular cryptographic primitives and AI problems is the notion of a security parameter. If we believe that an adversary can factor 1024-bit integers, we can use 2048-bit integers instead. No such concept exists in hard AI problems. AI problems, as we have defined them, do not deal with asymptotic. However, as long as there is a small gap between human and computer ability with respect to some problem, this problem can potentially be used as a primitive for security: rather than asking the prover to solve the problem once, we can ask it to solve the problem twice. If the prover gets good at solving the problem twice, we can ask it to solve the problem three times, etc. There is an additional factor that simplifies the use of hard AI problems as security primitives. Most applications of captchas require the tests to be answered within a short time after they are presented. If a new program solves the hard AI problems that are currently used, then a different set of problems can be used, and the new program cannot affect the security of applications that were run before it was developed. Compare this to encryption schemes: in many applications the information that is encrypted must remain confidential for years, and therefore the underlying problem must be hard against programs that run for a long time, and against programs that will be developed in the future.1We also note that not all hard AI problems can be used to construct a captcha. In order for an AI problem to be useful for security purposes, there needs to be an automated way to generate problem instances along with their solution. The case is similar for computational problems: not all hard computational problems yield cryptographic primitives.

V. DISCUSSION AND CLOSING REMARKS

a) Interaction with the AI community

A primary goal of the captcha project is to serve as a challenge to the Artificial Intelligence community. We believe that having a well-specified set of goals will contribute greatly to the advancement of the field. A good example of this process is the recent progress in reading distorted text images driven by the captcha inan example. In response to the challenge provided by this test, Malik and Mori[9] have developed a program which can pass the test with probability roughly0:8. Despite the fact that this captcha has no formal proof that a program which can pass it can read under other distributions of image transformations, Malik and Mori claim that their algorithm represents significant progress in the general area of text recognition; it is encouraging to see such progress. For this reason, it is important that even Automated Turing Tests without formal reductions attempt to test ability in general problem domains; and even though these tests may have specific weaknesses it is also important that AI researchers attempting to pass them strive for solutions that generalize.

b) Other AI problem domains

The problems defined in this paper are both of a similar character, and deal with the advantage of humans in sensory processing. It is an open question whether captchas in other areas can be constructed. The construction of a captcha based on a text domain such as text understanding or generation is an important goal for the project (as captchas based on sensory abilities can't be used on sensory-impaired human beings). As mentioned earlier, the main obstacle to designing these tests seems to be the similar levels of program ability in text generation and understanding.

Logic problems have also been suggested as a basis for captchas and these present similar difficulties, as generation seems to be difficult. One possible source of logic problems are those proposed by Bongardin the 70s; indeed presents a test based on this problem set. However, recent progress in AI has also yielded programs which solve these problems with very high success probability, exceeding that of humans.

VI. CONCLUSION

We believe that the fields of cryptography and artificial intelligence have much to contribute to one another. Captchas represent a small example of this possible symbiosis. Reductions, as they are used in cryptography, can be extremely useful for the progress of algorithmic development. We encourage security researchers to create captchas based on different AI problems.

REFERENCES

[1] Hard AI Problems for SecurityLuis von Ahn1, Manuel Blum1, Nicholas J. Hopper1, and John Langford2 Computer Science Dept., Carnegie Mellon University, Pittsburgh PA 15213, USA (2)IBM T.J. Watson Research Center, Yorktown Heights NY 10598, USA *.

[2] Luis von Ahn, Manuel Blum, Nicholas J. Hopper and John Langford. The CAPTCHA Web Page: http://www.captcha.net. 2000.

[3] Luis von Ahn, Manuel Blum and John Langford. Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI. To appear in Communications of the ACM.

[4] Mihir Bellare, Russell Impagliazzo and Moni Naor. Does Parallel Repetition Lower the Error in Computationally Sound Protocols? In 38th IEEE Symposium on Foundations of Computer Science (FOCS' 97), pages 374-383. IEEE Computer Society, 1997.

[5] Mikhail M. Bongard. Pattern Recognition. Spartan Books, Rochelle Park NJ, 1970.

[6] Wikipedia 2013-2015

[7] “WHITE BOOK“ researched by Google in their CAPTCHA Implementation 2015.

[8] Baars, B. J. (1988). A cognitive theory of consciousness. New York: Guilford Press.

[9] Baars, B. J. (1997). In the theater of consciousness: The workspace of the mind. New York: Oxford University Press.

[10] Block, N. (1997). On a confusion about a function of consciousness. In Block et al. (1997), pp. 375–415.

[11] Block, N., Flanagan, O., & G¨uzeldere, G. (eds) (1997). The nature of consciousness: Philosophical debates. Cambridge, Mass.: MIT Press.

[12] Boddy, M. & Dean, T. (1989). Solving time-dependent planning problems. Proc. Ijcai, 11, pp. 979–984.

[13] Campbell, J. (1994). Past, space and self. MIT Press: Cambridge. Chalmers, D. (1996). The conscious mind: In search of a fundamental theory. New York: Oxford University Press.

[14] “Artificial Intelligence: The Next Twenty-Five Years.” Edited by Matthew Stone and Haym Hirsh. AI Magazine, 25th Anniversary Issue. Winter 2005.

[15] Brooks, Rodney. “Artificial Intelligence Laboratory.” Electrons and Bits. Ed. John V. Guttag. Cambridge, MA, Electrical Engineering and Computer Science Department: 2005.

[16] Buchanan, Bruce and McCarthy, John. AAAI 2002. Brief History of Artificial Intelligence. <http://www.aaai.org/AITopics/bbhist.html>.

[17] Buchanan, Bruce G. “A (Very) Brief History of Artificial Intelligence.” AI Magazine, 25th Anniversary Issue. Winter 2005.

[18] Chandler, David. Volkswagen wins robotic race across the desert. NewScientist.com news service. Oct. 10, 2005 <http://www.newscientist.com/article.ns?id=dn8119>.

[19] Cohen, Paul R. “If Not Turing's Test, Then What?” AI Magazine, 25th Anniversary Issue. Winter 2005.

[20] Edwards, Paul N. Closed World: Computers and the Politics of Discourse in Cold World America. Cambridge, MA: The MIT Press, 1996.

[21] Garfinkel, Simon L. LCS: Architects of the Information Society. Ed. Hal Abelson. Thirty-Five Years of the Laboratory for Computer Science at MIT. Cambridge, MA: The MIT Press, 1999.

[22] Greenblatt, Rick. “Podcasts.” Recovering MIT's AI Film History Website. MIT. June 2006. <http://projects.csail.mit.edu/films>.

[23] Güzeldere, Güven, and Stefano Franchi. “Dialogues with Colorful Personalities of early AI.” SEHR: Constructions of the Mind. Vol. 4.2, 24 July 1995. <http://www.stanford.edu/group/SHR/4-2/text/toc.html>.

AUTHORS PROFILE

Mr. Saroj Kumar, PhD Scholar from Maharishi University of Information Technology, Lucknow completed his M.Tech form NIT - Allahabad and more than 10 year in academic in different colleges India.

Dr. Santosh Kumar, Associate Professor and Head in Maharishi University of Information Technology, Lucknow. Serving more than 10 year in Academic Field and Guided more than 10 PhD Scholar in Computer Science Department