2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE
International Conference on Edge Computing and Scalable Cloud (EdgeCom)

# CAPTCHA: Machine or Human Solvers? A Game-theoretical Analysis

Zhen Li
Department of Economics & Management
Albion College, USA
Email: zli@albion.edu

Qi Liao
Department of Computer Science
Central Michigan University, USA
Email: liao1q@cmich.edu

***Abstract*—CAPTCHAs have become an ubiquitous defense used to protect open web resources from being exploited at scale. Traditionally, attackers have developed automatic programs known as CAPTCHA solvers to bypass the mechanism. With the presence of cheap labor in developing countries, hackers now have options to use human solvers. In this research, we develop a game theoretical framework to model the interactions between the defender and the attacker regarding the design and countermeasure of CAPTCHA system. With the result of equilibrium analysis, both parties can determine the optimal allocation of software-based or human-based CAPTCHA solvers. Counterintuitively, instead of the traditional wisdom of making CAPTCHA harder and harder, it may be of best interest of the defender to make CAPTCHA easier. We further suggest a welfare-improving CAPTCHA business model by involving decentralized cryptocurrency computation.**

***Index Terms*—CAPTCHA, Machine Solver, Human Solver, Computer Security, Game Theory, Economics, Cryptocurrency, Blockchain, Bitcoins**

## I. Introduction

Thanks to the botnets, attackers are able to explore the vast amount of network resources in a short amount of time, e.g., registering email accounts for sending out spams. To limit the ability of attackers to scale their activities using automated means, CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) was developed as a reversed Turing test[1] to tell apart a human from a bot. CAPTCHA is and will remain in the foreseeable future as the defense technology to protect server resources from abuse and automated attacks.

To counteract CAPTCHA, the attacker community has developed automated programs known as CAPTCHA solvers. For example, a classic solver will feed an image to optical character recognition (OCR) module to detect the characters in a typical text-based CAPTCHA. The accuracy varies depending on the hardness (e.g., noise) in the images. As a result, the defending community has been designing harder CAPTCHA by using different types of CAPTCHAs [1], [2] and introducing substantial noise. Often it is too difficult even for human. Automated attacks on CAPTCHA systems define a technical arms race between those developing solving algorithms and those developing ever more obfuscated CAPTCHA challenges in response. Arms races are generally considered bad in cybersecurity since the denfender needs to manage thousands (or more) of machines while an attacker only needs to find one vulnerability on a system or one way to bypass the firewall. With CAPTCHA, however, arms race is actually in favor of the defender, i.e., it is cheap for a website to modify an existing CAPTCHA (e.g., changing the angles of characters), but it is expensive for attackers to hire programmers to rewrite a CAPTCHA solver. Automated attacks are also easier to detect [3]. Furthermore, the accuracy rate of automated recognition is usually low, at about 30% at best, further reducing the return on investment (ROI) on what is an expensive and repeated investment in programming.

With the cheap labor in developing countries (e.g., labor force in China, India, Venezuela, etc., is willing to earn only a few cents over solving 100 CAPTCHAs), the attacker communities have recently shifted to human solvers, which are cheaper with much higher accuracies since by definition CAPTCHAs must be solvable by humans. With the practice of botmasters outsorcing the CHATCHA-solving problem to human workers, the defender's dilemma is whether to make CAPTCHAs more complex or less complex. Making them hard pushes the attacker to use human labor, which may not achieve any security gain while sacrificing some user experience by forcing legitimate users to perform more complex tests. Simplyfying them improves user experience at the expense of bringing back automated solvers. As long as the defender provides attackers the option to carry out either bot-based or human-based solving, CAPTCHAs cannot function effectively as a cybersecurity defense measure.

It is imperative to look for alternatives. What measures may defeat human solvers? What if no effective measures against human solvers could be found? In this paper, we seek for possible answers to these questions. We model the interplay between the defender and the attacker as a Stackelberg game, where the defender is the leader, and the attacker is the follower. The defender's choice of CAPTCHA impacts the attacker's choice of test solving methodology and user experience which, in turn, affect the utility of the defender. The interaction between the defender and the attacker in the technological arms race, the defender's dilemma and the attacker's tradeoff between machine and human solving are studied in the game theoretic setting.

Utilizing economic levers, the defender can influence the

[1] Terms CAPTCHAs and tests may be used interchangeably.

978-1-5386-5850-5/18/$31.00 ©2018 IEEE
DOI 10.1109/CSCloud/EdgeCom.2018.00013

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:46:53 UTC from IEEE Xplore. Restrictions apply.

attacker's choice regarding how to solve the tests, with bots or humans, by choosing the complexity of the tests. To effectively prevent both automated and human solving, the design of CAPTCHAs has to be hard on both bots and human solvers, while being friendly to normal users. Based on the game theoretic analysis, we provide some thoughts on this issue. First, rather than making CAPTCHAs harder, multiple easy tests may be better than single hard test by maximizing the time latency cost in human solvers. In particular, the modeling analysis implies that at the presence of human labor, using CAPTCHA as security means to defend against botnet attacks can be effective only in special cases, and we have to think out of the box when it comes to the design and use of CAPTCHA. Second, CPU time-sharing model (e.g., running cyrptocurrency such as bitcoins mining programs) may be incorporated with existing CAPTCHA model to create a win-win situation.

The rest of the paper is organized as follows. In Section II, we discuss related works. In Section III, we define and model the strategic interactions between the defender and the attacker in a Stackelberg security game. We discuss the model implications of the attacker's best response to the defender's strategy of CAPTCHA design and point out the defender's dilemma at the presence of human CAPTCHA-solving services. Based on the modeling analysis, Section IV proposes a methodology that may defeat human solving services. It also discusses an unconventional CAPTCHA business model that can be welfare-enhancing. Section V concludes the paper.

## II. Background and Related Work

CAPTCHAs are motivated by the need to differentiate humans and robots in an online environment. Such mechanisms rely on the gap of intelligence between users and machines to protect web sites with resources shared by users. Widely adopted as defensive scheme against bots, CAPTCHA implementations can be found on more than 3.5 million sites globally, and humans solve such tests more than 300 million times a day [4]. While used mostly for security reasons, CAPTCHA can also be used as a benchmark test for AI (Artificial Intelligence) technologies [5], [6].

Different techniques for designing and generating CAPTCHA have been developed in order to address accessibility and usability while maintaining security [7]. Using distorted text, the color of image, object or the background that is accessible to users but unfriendly to bots, CAPTCHAs can be difficult to solve programmatically [8]. The most common CAPTCHA is text-based due to their easy implementation and usability. It is tractable for humans to recognize and trace a one-stroke character or symbol, but intractable for computer programs [9]. Other types of CAPTCHA include image-based, audio-based, video-based, game-based, and arithmetic-based tests. They are normally used by research networks, but not by many commercial web sites.

Various tests differ in user experience, success rate, and response time [10]. CAPTCHA solving affects user experience, which does not only include usability, but also other cognitive and affective aspects of user experiences in their interaction with networks such as all their emotions, beliefs, preferences, conceptions, psychological and physical reactions, and achievements occurring before, during, and after usage [11]. The success of web sites and networks is to a large extent positively influenced by the extent to which they promote a high-quality experience to their users [12]. Empirical studies on user experiences found most of the users do not like to solve the tests, and they do not feel protected. Although they are mostly familiar with text-based tests, they found it the most frustrating and non-enjoyable. While users found image and game based tests enjoyable, it takes more time for them to respond [10]. None of the existing tests are ideal. Users with learning disabilities have more difficulties in solving the tests [13]. To prevent CAPTCHA from impairing web site usability, the test schemes need to be properly designed to take into consideration user preferences [14].

Ever since its first appearance in early 2000s, CAPTCHAs have been the subject to attacks [15] by automated machine solvers. The most common way to automatically solve text-based CAPTCHAs is to use an OCR software [16]. Recently, the main security mechanism to avoid breaking relies on anti-segmentation techniques [17]. Experiments show that preventing segmentation alone does not provide reliable defense against automated attacks. Most CAPTCHA schemes from popular web sites were found vulnerable to automated attacks with approximately 25% success rates [18], but those breaking systems do not recognize certain versions of CAPTCHA [19].

In addition to automated CAPTCHA solvers using programs, the community has recently explored other mechanisms involving real human in the loop. The initial purpose of the tests is largely defeated when the attacker has access to cheap human labor. Study on the human solvers found that the retail price of human workers employed to decode CAPTCHA is low ($0.5/1000) [3]. Since professional human solvers are employed, the accuracy may actually be higher than an ordinary user. Even interactive CAPTCHA is prone to stream relay attack: a bot can relay the data stream from the server over to a human solver, and then relay back the input to the server [20]. Compared to the vast literature on CAPTCHA development and counterattack measure, there has been limited research on outsourcing CAPTCHA solutions to cheap human labor [3], CAPTCHA farms [21], and the use of Mechanical Turk [22], etc.

## III. The CAPTCHA Game and the Defender's Dilemma

In this section, we capture the interplay between the defender of CAPTCHA-protected network and the attacker (or botmaster) in a Stackelberg game setting. Through exploration of the strategic spaces for both players, we study how the defender's decision making on CAPTCHAs affects the solving methods chosen by the attacker, i.e., whether using software or human.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:46:53 UTC from IEEE Xplore. Restrictions apply.

### A. The Model

We model CAPTCHA-solving defense problem as a Stackelberg security game, in which there are two interest parties (players), i.e., the defender, who manages a server; and the attacker, who owns or rents a botnet and attempts to break CAPTCHA on the server. The defender and normal users may be treated as the same interest party because the goal of the system and network administrator is to ensure smooth user experience and defend against machine manipulation of the network.

In such a game, the defender commits to a randomized CAPTCHA complexity to balance security and user experience. The attacker then optimizes its attack action with respect to the distribution of the defender's actual choice of CAPTCHA. In this context, the defender is the leader of the game that aims at selecting the most effective complexity of CAPTCHA to impede the attack with minimal possible sacrifice in user experience.

*1) Defender's Decision Making:* The payoff of the defender is two dimensional: network security ($S$) and user experience ($E$), i.e., the utility function of the defender is written as $U = S+E$. The defender adopts CAPTCHA to protect the network from unfair usage by bots. The strategy space for the defender is a range of feasible tests with varying complexity. Finding an optimal defense test that balances security and user experience is formulated as follows:

$$\max_{D} U(D) = S(D) + E(D) \qquad s.t. D \in [0, \overline{D}] \tag{1}$$

where $U(D)$ is the defender's utility function for choosing a CAPTCHA test with a particular type and complexity level of $D$. The decision variable and constraint $D \in [0, \overline{D}]$ is the set of tests the defender can choose from: the lower bound $D = 0$ represents for no CAPTCHA protection; the upper bound $D = \overline{D}$ is the most challenging test possible for humans to solve.

The well-being of the defender increases when the network is better CAPTCHA-protected and more user friendly. In practice, the defender often faces a tradeoff when designing CAPTCHA. As the CAPTCHA methods are striving to be difficult for bots, they are gradually becoming difficult and annoying for human users as well [8]. Game theory can choose the security measure which makes the best tradeoff between the incurred cost and level of security achieved. Solving (1), the utility-maximizing test corresponds to a complexity level of $D^o \in [0, \overline{D}]$ that satisfies $S'(D^o) = E'(D^o)$, taking into account the tradeoff.

*2) Attacker's Decision Making:* The decision variable $b$ for the attacker is twofold: whether to use automated software or human to solve CAPTCHAs. The payoff depends on the benefits $B$ to receive upon successful breaking and the cost of breaking. Choosing the optimal breaking method is formulated as follows:

$$\max_{b} \pi(b) = B - C(b) \tag{2}$$

where $\pi(b)$ is the net payoff the attacker is expected to receive from successful CAPTCHA breaking. Since payoff remains the same, maximizing net payoff is equivalent to minimizing solving cost. The determining factor of which solving method to use is the cost effectiveness.

The cost structures of automated solving and human solving are drastically different. Generally, there are two components of a cost function: fixed cost (or lump-sum cost) and variable cost, depending whether a cost changes with the level of output or not.

The fixed cost for the attacker involves hiring elite programmer to develop programs for automated solving and is the major cost. Once the solver is ready, it can be used repeatedly on a large scale. However, a CAPTCHA solver is only effective in solving a particular type of CAPTCHA it is designed for. The attacker may have to rewrite the code facing new tests or even if just a minor modification (e.g., characters rendered at different angles).

The variable cost for machine solving is the cost to rent and maintain botnets, which is normally trivial at the margin. In contrast, when human labor is hired to solve CAPTCHAs, all the costs are variable that is increasing in the unit of labor hired, and there is no lump-sum cost of breaching.

We use the following cost functions to differentiate the accuracy-adjusted costs ($C$) of automated solving and human solving of a predetermined quantity of tests ($N$).

$$C(b) = \begin{cases} F + m_b \times \frac{N}{p_b}, & \text{machine} \\ m_l \times \frac{N}{p_l}, & \text{human} \end{cases}$$

where $F$ is the lump-sum cost of CAPTCHA solver development to target a certain type of test, $m_b$ is the average cost to rent and maintain a bot, and $m_l$ is the average cost of human labor. Let $p_b$ be the accuracy rate of machine (bot) solving and $p_l$ be the accuracy rate of human solving. Thus $\frac{N}{p_b}$ is the number of bots required to successfully solve $N$ tests at the given accuracy rate of automated solving, and $\frac{N}{p_l}$ is the unit of labor necessary to solve $N$ tests at the given accuracy rate of human solving.

Bots and human labor are different in the likelihood to successfully pass a test. Apparently $p_b \ll p_l$. In principle, all CAPTCHA tests should be solvable by human labor because the design of the tests must be feasible for users to solve, i.e., $p_l \to 1$. The accuracy rate of automated solving is much lower.

On per-test basis, the cost functions of automated solving and human solving are

$$\frac{C(b)}{N} = \begin{cases} \frac{F}{N} + \frac{m_b}{p_b}, & \text{machine} \\ \frac{m_l}{p_l}, & \text{human} \end{cases}$$

The best response of the attacker regarding what type of solving method to use depends on the comparison of the costs.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:46:53 UTC from IEEE Xplore. Restrictions apply.

A rational attacker will always choose the option that has the lower cost. In particular, machine solving is the best strategy in case of $\frac{F}{N} + \frac{m_b}{p_b} < \frac{m_l}{p_l}$, and vice verse.

### B. Model Implications

***Proposition 1.*** Events and forces that lead to a less accuracy rate of machine solving ($p_b$), an increased software development cost ($F$), an increased cost of bot maintenance ($m_b$), and a decreased network scale ($N$) will increase the cost of automated solving.

Since $m_b$ is usually small and $N$ is usually large, the conventional practice for the defender is to prevent machine-based solving by making tests hard on bots, thus increasing the CAPTCHA solver development cost and decrease the accuracy rate of automated solving.

***Proposition 2.*** There exists a complexity level of CAPTCHA that increases the cost of machine solving to the break-even point.

Suppose the per-solving benefit for the attacker is $W$. Machine solving breaks even when $\frac{F}{N} + \frac{m_b}{p_b} = W$. The break-even complexity $D^* \in [0, \overline{D}]$ corresponds to the accuracy rate of $p_b^* = \frac{m_b}{W - \frac{F}{N}}$ at which $\frac{F}{N} + \frac{m_b}{p_b^*} = W$.

The break-even $p_b^*$ is the lowest possible accuracy rate machine solvers must achieve to make the effort worthwhile. It is increasing in the solver development cost and decreasing in the scale of the networks adopting the same type of CAPTCHA. Therefore, updating, upgrading and diversifying CAPTCHA can reduce the likelihood of using machine solvers.

If machine solving is the only option for the attacker, increasing the complexity level of tests to $D^*$ shall be financially sufficient to defeat software-based solving. However, at the presence of human solvers, using tests as challenging as $D^*$ may fail to be effective.

***Proposition 3.*** Machine solver is more cost-effective than human solver iif economies of scale can be realized with existing CAPTCHA solver.

The attacker's best strategy is to choose machine solver when $\frac{F}{N} + \frac{m_b}{p_b} < \frac{m_l}{p_l}$. The accuracy rate of human solver is close to 1 ($p_l \rightarrow 1$). The per-bot maintenance cost is trivial ($m_b \rightarrow 0$). Approximately, machine solver is more cost-effective than human solver if and only if $\frac{F}{N} < m_l$. In other words, machine solver could be the attacker's best response if and only if the economies of scale were achieved (e.g., $N \rightarrow \infty$).

The market price of human solving has been at low level. For instance, the CAPTCHA-solving service charged by `anti-captcha.com` starts from $0.50 per 1000 tests. Even for more challenging tests such as `reCAPTCHA` and `FunCaptcha`, the costs start from only $1.80 per 1000 tests. The range of CAPTCHA solver development cost is tight to make machine solving financially advantageous to human labor. If we add the rent and maintenance cost of bots to the formula, the financially feasible range of CAPTCHA solver cost further narrows.

In theory, the defender may defeat machine solver with the use of renovated tests so that the economies of scale can never be achieved with machine solver and the accuracy rate of machine solver remains at a low level. Nevertheless, such practice is not effective to defeat human solver. It works to induce the attacker to shift from machine solver to human solver.

***Proposition 4.*** In theory, there exists a break-even accuracy rate of human solver (which is nearly impossible to realize under the practice of making CAPTCHA hard on bots but easy on humans).

Human solving breaks even when $W = \frac{m_l}{p_l}$ so that the break-even accuracy rate for human solving is $p_l^* = \frac{m_l}{W}$. Human solver is profitable $\forall\, p_l > p_l^*$, which is a wide range. The labor cost of human-solving has been low and will remain low in the foreseeable future. The payoff of successful breaking for the attacker can be significant. Normally, $m_l \ll W$ and $\frac{m_l}{W} \rightarrow 0$ so that the theoretical constraint is extremely difficult if ever possible to be binding under the current practice of making CAPTCHA hard on bots but easy on humans.

## IV. Prevention of Human Solvers

Making CAPTCHA hard on bots and easy on humans is a self-defeating strategy for the defender at the presence of human solvers. It is impossible to defeat human solver by merely changing the type of tests used because such tests must always be solvable by legitimate users as well. In this section, we discuss a plausible way to separate legitimate users from human solvers that leads to a test design methodology to defeat human labor. Furthermore, welfare implications of the methodology is analyzed that leads to a new business model of CAPTCHA (cryptocurrency mining) that may be a win-win situation for the defender and normal users with ambiguous welfare effect on the attacker.

### A. Human CAPTCHA-Solving Market as a Two-Sided Market

A two-sided market is a market where two or more groups of agents interact via intermediaries or platforms. The human CAPTCHA-solving market is a two-sided market. The two sides are attackers and human labor. The intermediaries are test-solving services such as `de-captcher.com` and `anti-captcha.com`. Test-solving transactions between attackers and human labor take place via test-solving services that aggregate the demand for services via a public web site and open API and aggregate the supply of services via online advertisements on work-for-hire sites.

Attackers who seek test-solving services are the customer side. Human labor who solve tests are the seller side. While it appears human labor are workers hired by test-solving services, they are the real source of test-solving capabilities supplied to customers. Thus, the success of CAPTCHA-solving services depends on their ability to bring in members from both sides: attackers who are willing to pay for test-solving services and human labor who are willing to interactively solve tests for pay.

The role of CAPTCHA-solving services can be divided in two parts: connecting attackers and human labor, and providing each side of the trading relation with information

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:46:53 UTC from IEEE Xplore. Restrictions apply.

about the other side. Attackers and human labor benefit from the breadth of offering that is proposed by the services. The higher the number of human labor a service has at present, the higher is the probability that attackers get test solved with a shorter response time; the higher the number of attackers coming to a service, the higher is the probability the human labor gets paid highly. Network effects thus arise naturally from the function of the test-solving service as an intermediary.

### *B. Legitimate Users vs. Human Solvers*

It is hard to differentiate legitimate users from human solvers since they both are human only intent is different. We suggest one possible way to differentiate these two types of users is the time latency. A legitimate user sees a CAPTCHA and is able to solve it instantaneously. For human CAPTCHA-solving service, the attacker first uploads tests to the provider's server, which routes the request to foreign countries and puts the request into a queue waiting for an employee to solve. The human solvers then type the solutions back to the platform to be sent back to the attacker. It is reasonable to assume such extra time lag of indirection and detour. While the number of seconds may depend on the network link quality as well as the request queue length, in theory, the lag must exist regardless how developed the service infrastructure is. The defender may take advantage of the lag to hinder human CAPTCHA solver.

### *C. A Human Solver Discouragement Model*

The defender needs to think out of the box. Rather than making CAPTCHAs more complex and confusing, we suggest making it simple and easy for users to solve. Instead of using one hard CAPTCHA that takes a long time, it may be better to use multiple easy CAPTCHAs, each of which with more aggressive time constraint by refreshing tests quickly enough to disenable human solving services. If the defender simplifies tests to reduce the time it takes legitimate users to solve, the overhead of indirection and routing for human solvers will be maximized.

We formulate the proposed solution, beginning with two assumptions. First, human CAPTCHA-solving services use certain relay mechanism and have noticeable queueing delay in the service infrastructure. Second, work efficiency is uniformly distributed among human solvers and ordinary users.

Formally, let $t \in [0, T]$ be the time for a human to solve a test instantaneously. In absence of CAPTCHA ($D = 0$), $t = 0$; for the most sophisticated test possible ($D = \overline{D}$), $t = T$. The user experience utility function can be written in terms of the time the user spends to solve a test, $E(D) = E(t(D))$. Different from normal users, the attacker needs to upload the test to the server of a CAPTCHA-solving service provider who then sends the answer back to the attacker after human labor types the answer. Suppose the average time lag is $L$ that is constant regardless of the test complexity. $t(D) + L$ is the average response time of CAPTCHA-solving services. The time lag $L$ differentiates normal users from human workers. There exists a test complexity level $D^l$ at which $t(D^l) = L$. For any $D^f \in [0, D^l]$, the defender can set a CAPTCHA refreshing frequency in the range of $[t(D^f), L]$ to block off human CAPTCHA-solving.

By decreasing the success rate of human CAPTCHA-solving, the proposed methodology may force the attacker to shift back to machine solvers. In an extreme case, in which the attacker has ready in hand all machine solvers for CAPTCHAs with low complexity, the additional software development cost would drop to zero, thus machine solver is always the best response. In case a new program is required, machine solver is the best strategy when $\frac{F}{N} + \frac{m_b}{p_b} < \frac{m_l}{p_l}$. The proposed approach of simplifying tests with time latency constraints works to reduce the success rate of human solving services. As $p_l$ decreases, the likelihood for the attacker to choose machine solver over human solver increases.

From Proposition 3, the attacker will always choose machine solver when an existing program can be widely or repeatedly used to dilute fixed cost of software development. Simplified tests that minimize human solving time are not necessarily easy on bots. Ideally, the defender shall choose tests that are easy on humans while staying challenging for bots. If not possible, simplifying tests will induce the attacker back to machine solver, which may benefit the defender if the attacker's botnet resources can somehow be harvested by the defender. Along this line of thinking, we further propose a model as discussed in the next section.

### *D. A Welfare-Improving CAPTCHA Business Model*

Since the major function of CAPTCHA is to distinguish human from machines, the effectiveness of CAPTCHA against botnets is largely invalidated by the existence of human solvers. The above proposed method of preventing human solvers can be welfare-improving to keep the attacker stay with machine solver.

Considering the utility maximization problem of the defender as in (1), the common wisdom of making CAPTCHA difficult on bots but easy on humans is not optimal because of the existence of human solvers. It achieves little (if any) function of CAPTCHA at the cost of deteriorating user experience. Minimizing human test-solving time improves user experience. The plausible loss (if any) is insignificant no matter whether the attacker stays with human solver or shifts to machine solver. The defender may actually benefit from the attacker's shifting to machine solving practice.

To facilitate the proposed human-solver-defeating strategy, the tests must be straightforward enough for users to solve in a short time. If CAPTCHA-solving by attacker cannot be prevented anyway, the defender may want to get payoff from the attacker to balance off. With the rapid rise of cryptocurrency (e.g., the BitCoin price rose from \$900 to \$17,000 in the year of 2017), the reward for blockchain mining becomes substantial. One possibility of new CAPTCHA model is to utilize the botnet resources, as follows.

First, the website owner embeds cryptocurrency's (e.g., Moneor, Bitcoin, etc.) blockchain miner in website. Second, the owner simplifies CAPTCHA and refreshes it frequently enough so that the botmaster always chooses machine solver.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:46:53 UTC from IEEE Xplore. Restrictions apply.

Third, while solving the CAPTCHA, a script runs on the bot harvesting the machine's CPU power to compute hashes for transaction blocks in the blockchain. Lastly, the responses for both CAPTCHA and miner are returned to the web server for verification and the script stops.

With the rewards from cryptocurrency mining, the defender may, for example, offer ad-free services to users. The mining-linked CAPTCHA benefits the defender and majority of legitimate users. The welfare effect on the attacker is however ambiguous. It depends on the relative cost of machine solver and human solver. The attacker is worse off for sure if human solving is more cost effective otherwise.

## V. Conclusion

CAPTCHAs are widely used and will continue to exist in foreseeable future for differentiating users and bots. While CAPTCHA has shown effectiveness in this function, the emergence of human CAPTCHA solvers makes researchers rethink the problem. In this paper, we formally modeled the interdependence of the decision-making by the defender and the attacker in a Stackelberg game theoretic framework. Through best response and strategy analysis, the break even points of whether adopting machine solver or human solver can be determined. In contrary to traditional wisdom to make CAPTCHA harder, we proposed two models that feature easy CAPTCHA with time latency constraints as well as incorporation of cryptocurrency mining into existing CAPTCHA mechanism. The results discourage attackers from using human solvers and generate a welfare-enhancing CAPTCHA business model.

## References

[1] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Pérez-Cabo, "No bot expects the DeepCAPTCHA! introducing immutable adversarial examples, with applications to CAPTCHA generation," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2640–2653, 2017.

[2] B. M. Powell, E. Kalsy, G. Goswami, M. Vatsa, R. Singh, and A. Noore, "Attack-resistant aiCAPTCHA using a negative selection artificial immune system," in *Proceedings of IEEE Security and Privacy Workshops*, San Jose, CA, May 25 2017, pp. 41–46.

[3] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage, "Re: CAPTCHAS – understanding CAPTCHA-solving services in an economic context," in *USENIX Security Symposium*, Washington, D.C., August 11-13 2010.

[4] A. R. Angre, M. D. Kapadia, and M. Ugale, "PiCAPTion: Picture CAPTCHAs for Internet authentication," *International Journal of Computer Applications*, vol. 114, no. 10, pp. 6–9, 2015.

[5] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "Captcha: Using hard AI problems for security," in *Proceedings of the 22nd Internatiaonal Conference on the Theory and Applications of Cryptographic Techniques*, Warsaw, Poland, May 04-08 2003, pp. 294–311.

[6] B. B. Zhu, J. Yan, G. Bao, M. Yang, and N. Xu, "Captcha as graphical passwords  a new security primitive based on hard AI problems," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 891–904, 2014.

[7] M. Moradi and M. Keyvanpour, "CAPTCHA and its alternatives: A review," *Security and Communication Networks*, vol. 8, no. 12, pp. 2135–2156, August 2015.

[8] S. Kulkarni and H. S. Fadewar, "CAPTCHA based web security: An overview," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 11, pp. 154–158, 2013.

[9] M. Takaya, H. Kato, T. Komatsubara, Y. Watanabe, and A. Yamamura, "Recognition of one-stroke symbols by humans and computers," *Procedia-Social and Behavioral Sciences*, vol. 97, pp. 666–674, November 2013.

[10] R. Gafni and I. Nagar, "CAPTCHA-security affecting user experience," *Issues in Informing Science and Information Technology*, vol. 13, pp. 63–77, 2016.

[11] N. Bevan, "What is the difference between the purpose of usability and user experience evaluation methods," in *Proceedings of User Experience Evaluation Methods in Product Development (UXEM09) Workshop*, Uppsala, Sweden, August 24-28 2009.

[12] E. L.-C. Law and P. van Schaik, "Modelling user experience - an agenda for research and practice," *Interacting with Computers*, vol. 22, no. 5, pp. 313–322, 2010.

[13] R. Gafni and I. Nagar, "CAPTCHA: Impact on user experience of users with learning disabilities," *Interdisciplinary Journal of e-Skills and Life Long Learning*, vol. 12, pp. 207–223, 2016.

[14] E. Bursztein, A. Moscicki, C. Fabry, S. Bethard, J. C. Mitchell, and D. Jurafasky, "Easy does it: More usable captchas," in *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Toronto, Canada, April 26 - May 01 2014, pp. 2637–2646.

[15] M. Serrao, S. Salunke, and A. Mathur, "Cracking captches for cash: A review of captcha crackers," *International Journal of Engineering Research & Technology*, vol. 2, no. 1, pp. 1–5, 2013.

[16] S. Azad and K. Jain, "CAPTCHA: Attacks and weaknesses against OCR technology," *Global Journal of Computer Science and Technology*, vol. 13, no. 3, pp. 14–17, 2013.

[17] B. Madar, G. K. Kumar, and C. Ramakrishna, "Captcha breaking using segmentation and morphological operations," *International Journal of Computer Applications*, vol. 166, no. 4, pp. 34–38, 2017.

[18] E. Bursztein, M. Martin, and J. C. Mitchell, "Text-based CAPTCHA strengths and weaknesses," in *Proceedings of the 18th ACM conference on Computer and communications security*, Chicago, IL, October 17-21 2011, pp. 125–138.

[19] C. Cruz-Perez, O. Starostenko, F. Uceda-Ponga, V. Alarcon-Aquino, and L. Reyes-Cabrera, "Breaking reCAPTCHAs with unpredictable collapse: Heuristic character segmentation and recognition," in *Proceedings of Mexican Conference on Pattern Recognition*, Huatulco, Mexico, Juen 27-30 2012, pp. 155–165.

[20] M. Mohamed, N. Sachdeva, M. Georgescu, S. Gao, N. Saxena, C. Zhang, P. Kumaraguru, P. C. van Oorschot, and W.-B. Chen, "A three-way investigation of a game-CAPTCHA: automated attacks, relay attacks and usability," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, Kyoto, Japan, June 4-6 2014, pp. 195–206.

[21] M. Egele, L. Bilge, E. Kirda, and C. Kruegel, "CAPTCHA smuggling: Hijacking web browsing sessions to create CAPTCHA farms," in *Proceedings of the ACM Symposium on Applied Computing*, Sierre, Switzerland, March 22-26 2010, p. 18651870.

[22] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky, "How good are humans at solving CAPTCHAs? a large scale evaluation," in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 16-19 2010, pp. 399–413.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:46:53 UTC from IEEE Xplore. Restrictions apply.