Security Design in Human Computation Games

Su-Yang Yu and Jeff Yan

School of Computing Science
Newcastle University
United Kingdom, NE1 7RU
{s.y.yu, jeff.yan}@ncl.ac.uk

Abstract. We consider a binary labelling problem: for some machine learning applications, two types of distinct objects are required to be labeled respectively, before a classifier can be trained. We show that the famous ESP game and variants would not work well on this binary labelling problem. We discuss how to design a new human computation game to solve this problem. It turns out that interesting but subtle security issues emerge in the new game. We introduce novel gaming mechanisms, such as 'guess disagreement', which improve the game's security, usability and productivity simultaneously.

Keywords: Binary labeling, collusion, cheating, CAPTCHA, 'Cat or Dog' game.

1 Introduction

The emerging research area of human computation studies how to solicit voluntary human efforts to solve difficult problems that known computer algorithms cannot yet tackle efficiently. Typical human computation systems include computer games that people play just for fun, but their game play contributes to collectively solving large-scale computational problems.

It has been known for long that an essential security issue in online game design is tackling cheats [7]. Human computation games are typically web based, and they are not an exception.

In this paper, we first consider a binary labelling problem that the famous ESP game [4] would not work well: for some applications, two types of objects are required to be labeled respectively; each type will have a distinct label, but the total number of possible labels is two. Our aim is to design a new human computation game to address this problem. It turns out that that some interesting but subtle security issues have to be addressed in the new game.

We first give some details of the binary labeling problem, and discuss why the state of the art cannot offer a satisfactory solution. Then, we discuss the design of our new game, including main game mechanisms, design rationale behind, and some key implementation details – security, usability, and game productivity are key issues in our design. Next, we report a pilot study that was designed to evaluate the fun level and the utility of the game.

2 The Binary Labeling Problem

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) [5] is now almost a standard security technology, and has been widely deployed by commercial websites as a defence against undesirable or malicious Internet bots. It is widely accepted that a robustness evaluation is necessary before the deployment of any CAPTCHA, since if a deployed CAPTCHA is not strong enough, bots can easily bypass its defence, rendering it useless.

Asirra [1] is an image-recognition based CAPTCHA proposed by Microsoft. Instead of asking a user to recognize a distorted text, this CAPTCHA requires people to tell whether an image is a cat or dog. In 2008, Golle [2] proposed a machine learning method for evaluating the strength of Asirra. In his work, a large set of sample images was manually labeled as cat or dog according to the content of the images. Then image features such as color, texture and shape were extracted from each sample image to build and train a binary classifier, which was then used to recognize whether a new image is cat or dog. This process turns out to be an effective method for evaluating how robust an image based CAPTCHA such as Asirra is in resistance to adversarial attacks. Labeling a large sample set (13,000 images in the case of [2]) is a key step in this method. However, it cannot be automated since the state of the art of computer vision does not work well in recognizing images of cats and dogs – otherwise the CAPTCHA would have no reason to exist at all. It is tedious and expensive to label these images by hand.

On the other hand, a key component of Asirra is a huge database of labeled images of cats and dogs. Asirra relies on this database to generate CAPTCHA challenges, and determine whether an answer given by a user is correct or not. For security reasons, this database is kept confidential. That is why Golle [2] had to manually label a large number of images for his independent security evaluation.

The Asirra database has over three millions of labeled images, and it grows by nearly 10,000 images everyday [1]. All the images are provided by Petfinder.com, the largest web site in the world devoted to finding homes for homeless animals, and their volunteer workers have manually created the image labels. Given the amount of images involved, this labeling process is even more tedious and more expensive than what is required for evaluating image CAPTCHA's robustness. That is, to keep Asirra running, a large amount of human labor has been, and will be, involved with maintaining its database.

The specific problem that we consider is therefore to label images of cats and dogs both for enabling the Asirra service and for carrying out its robustness evaluation. To generalize it, the problem we want to solve is binary labeling, where two types of objects are required to be labeled respectively; each type will have a distinct label, but the total number of possible labels is two. Other practical scenarios also require binary labeling. For example, in order to evaluate their recent quantum computer, Google researchers had to manually label car images and those that do not contain any cars [9].

3 Why Would the State of the Art Not Work?

The famous ESP game [4] was designed to create labels for images on the Internet and thus improving the quality of image search. It is a two-player collaborative game. An image randomly picked from the Internet is displayed to both players who cannot communicate with each other. Each is asked to guess what the partner is typing. When both players type the same word, it turns out that that word can be an accurate label for describing the image. This simple game has achieved a huge success – it not only has collected a large number of useful labels for images on the Internet (Google licensed this game to create their own image labeler), but also kicked off the new research field of human computation.

However, the ESP game would not work well to address the binary labelling problem that we discuss in the current paper, for the following reasons.

First, it would be easy to cheat in the game. To agree on an image in the ESP game, two players are not required to type the same word at the same time, but each must type the same word at some point while the image is displayed on the screen [4]. That is, if we denote by A a set of guesses that player a has entered, and by B a set of guesses that her partner has entered, once $A \cap B \neq \emptyset$, an agreement is reached by two players, and the intersection of A and B is accepted as a valid label for the image. Therefore, cheaters can exploit this agreement algorithm to cheat with binary labeling as follows: a player types in both possible labels for an image, and the other types in either. In the worst case, the cheaters can easily win the game without producing a single valid label – for example, the second player always gives a tag different from the content of an image. This cheat would not work with content-rich images, since possible labels for such images are typically neither fixed nor predictable.

Second, it would be boring to play the ESP game if there are only two possible tags for describing each image. A major fun element of the ESP game stems from that you have to guess how other people think. Typically, agreeing on an appropriate name for an image in the game creates an enjoyable feeling of 'extra-sensory perception' about each other within a partnership: 'hey, I read your mind!' This fun element is highly dependent on the choice of images used. If an image implies only a small number of possible labels, it will be trivial for two players to reach an agreement. Thus, the enjoyment from the ESP effect of 'reading each other's mind' will quickly diminish. To make a game fun to play, it is essential to make sure that its difficulty level is right. However, playing with images with only two possible labels would simply make the ESP game too easy, and thus not much fun to play. A plausible solution is to mix these images of cats or dogs with content-rich images in the game so that people do not have to do trivial labelling too often. However, this will slow down productivity for binary labeling. Moreover, the cheating issue discussed above is still applicable.

Third, some of the mechanisms in the ESP game would not work for binary labelling. For example, the ESP game proposed to detect cheaters involving with a global agreement strategy (e.g. both players type 'a' for every image) by monitoring the average time two players agree on images: a sharp drop in this average time indicates the existence of a cheating agreement [4]. For binary

labeling, this method would fail to differentiate between cheaters using a global strategy and those honest but really good players who can both respond and type fast – the latter do not have to ponder for long, since there are only two fixed options for each image.

Taboo words – another key feature in ESP – cannot be applied in a binary labelling setting, either. In ESP, taboo words serves two purposes: i) to harvest additional labels from the images by blacklisting common labels from the images' past incarnations thus forcing players to enter new labels, and ii) to prevent the use of a global agreement strategy by making a label agreed by a pair of players a taboo word across an entire session of their game play – the idea is that two players cannot agree on different images with the same word in the same session. Unfortunately, in a binary labelling setting, the introduction of taboo words would make the game unplayable. Due to the first purpose that taboo words serve, since there are only two possible labels, the most common input will be the correct label, and thus making it the first taboo word. This will consequently force the players to provide only the wrong label, until the wrong label is also tabooed, by which all the valid inputs are effectively blacklisted. On the other hand, due to the second purpose of taboo words, each valid label will be allowed only once in a single game session. That is, only two images – for one image of cat and the other of dog – will be properly labeled.

Magic Bullet [8] is a game that we designed to label images that are not content-rich, such as those containing only a single character or a short text string. In a typical setting, MB is a four-player game in which two teams compete against each other, with two players in each team. All of the players will share the view of the same gaming area, a screen with two targets – one is for their team and the other for their opponents' team. During each round, a randomly chosen image is shown to all four players. The team who first agrees on the image wins the current round of the game, and the image will turn into a bullet and shoot to their target. The movement of the bullet can drastically change as if by magic. For example, the bullet starts to move towards the target of the team that hit a key first. But if the other team reaches an agreement first, the bullet will change its direction to hit the winning team's target. As such, the game was named 'Magic Bullet'. A lab study suggested that this game maintains a high level of fun for players, and that people's game play creates accurate labels for images that contain only individual characters [8]. However, this game is not suitable for binary labeling because its matching algorithm is the same as the one used in the ESP game and thus vulnerable to the simple cheating discussed earlier.

4 A 'Cat Or Dog' Game

'Cat or Dog' is an online game that turns a task of labelling images of cats and dogs into a fun experience. People play the game just for fun, but their game play labels each image used in the game with a very high accuracy. This game also provides a generic solution to the binary labeling problem.

4.1 General Description of the Game

The theme of our 'Cat or Dog' game is on a treasure island, where pet animals walk past an area carrying gold coins, and players are teamed up to compete against others in order to attain the gold by winning the animal's affections.

In a typical setting, four players are randomly selected and allocated into two teams, with two people in each team. There are two treasure chests on the screen one for each team, and are located on the left/right side of the game area (See Figure 1). In each game session, the players must try to outscore their opponents by getting golden coins from pet animals within the allocated time limit of 2 minutes.

Fig. 1. The 'Cat or Dog' game. Players try to obtain golden coins by winning the pet animals' affections through agreeing on their identifications. The dog is running back to her previous route after dropping some coins in the winning team's treasure chest.

Each game session consists of an arbitrary number of rounds. At the beginning of a round, a pet – a cat or a dog – will run between the treasure chests

of two teams. The pet will appear on either the top or bottom end of the game screen and move towards the opposite end.

When the pet appears the players must identify the kind of animal she most looks like, and try to get her attention by pressing the corresponding key ('c' for 'cat' and 'd' for 'dog'). Each player will only get a single chance to call the pet during each round. That is, once a player has entered a key, their keyboard will be 'locked' until the next round. The pet will drop golden coins to the team that first agrees on their identification of her. However, the pet will become annoyed with the team who first disagrees on her identification, and will give the coins to their opponents instead. The pet will move towards the winning team, drop some golden coins in their treasure chest, and then run back to its previous route. When she is out of the game screen, it indicates the end of a round.

A round is over either when the allocated time for that round has past, or when a team has reached an agreement or a disagreement before their opponents do so.

The time of how long each round will be, i.e. the speed at which a pet moves, differs and is decided by the game server. It is significantly longer towards the beginning of a game session, and gradually speeds up towards the end of the session. The faster a pet moves, the higher number of points the winning team will get.

The round time is chosen to be between 2 and 5 seconds for the following reasons. First, we wanted the round to be fast enough since one major element of fun we expect is the speed of competition. Second, the animation must also be slow enough so that the images are still easily recognisable. This is due to the concerns for players getting eye strain from trying to recognise fast moving images for a relatively long period of time.

There are also two reasons for making the rounds gradually speed up in a session. First, this allows players time to slowly ease into the game. Second, this encourages the loosing team not to give up, as a higher speed also issues higher rewards so there is plenty of opportunity for them to catch up.

4.2 Main Design Rationale

The 'Cat or Dog' game is a variant of our own Magic Bullet game [8], with some innovative extensions that will be discussed below.

An initial design of the 'Cat or Dog' game we conceived was a direct variant of the MB game as follows. The character image is replaced by an image of a cat or a dog. Players are expected to type 'c' for cat and 'd' for dog, and the team that first reaches an agreement wins. But a new matching mechanism, which we call *turn-based matching*, will be used to determine whether two players have agreed on an image, and it works as follows. After a player has given a guess on the image, they will no longer be able to send out any additional guesses until their partner has also sent out something for that turn. Then instead of comparing all the keys both players have sent, only the keys during the same turn are compared against each other. If it is a match then they score the points;

if it is not a match then their keyboard is unlocked again to mark the start of the next turn.

However, this initial design would introduce some subtle usability and security issues. For example, in this design, an indicator such as a red/green light will be needed so that players know whether their keyboards are currently locked or unlocked respectively. For example, the light starts off as green at the beginning of a round. As soon as the player has entered a guess it turns red. The light will be reset to green again either when their partner has also given a guess or at the beginning of the next round. Without such an indicator, there would be a major usability issue: since there could be multiple turns in a round, a player would easily become confused about when to enter their guess, or whether their guess has been taken into count by the game.

However, cheaters could exploit this initial design of the game. Specifically, this design would allow the cheaters to collude through an in-game covert channel, which uses entities not normally viewed as a communication channel to transfer information. Some more apparent cheating threats in this game, together with their defence, will be discussed later in this paper. Some older cheating cases in online games involving covert channels were discussed in [7].

Typically, cheaters can be divided into two categories each with a unique motive as follows.

- Type 1: those who just want to pollute our data.
- Type 2: those who just want to score higher points than others.

Both types of cheaters will want to establish a protocol with their partner in the game to achieve their desired aims. Type 1 cheater will want to provide the label that is opposite to that of the image, and Type 2 cheaters will want to agree with their partner on a specific key so that they could react faster than their opponents.

In the course of game play, the indicator could leak critical information to the cheaters, enabling them to form a collusion protocol that would not have been probable in Magic Bullet. Since there are only two options ('c' or 'd') in 'Cat or Dog' as opposed to 36 (a-z & 0-9) in MB, after a player has given a guess, she can easily deduce what her partner has entered by looking at the indicator and the outcome of that turn.

A player can deduce what their partner has entered in a turn, because in this scenario there can only be 1 of 3 possible outcomes: i) their team has won that turn, ii) they have lost that turn, or iii) the turn is still ongoing and the indicator resets back to green. If they have lost that turn, then the player cannot deduce useful information, as her partner might have entered the same guess as she did, a guess different from the one she did, or simply have not entered anything yet. Otherwise, the key that the partner has entered can be derived as shown in Table 1.

The critical observation is that a cheater can easily convey via the indicator their intention of either always creating a wrong label (Type 1 cheating) or only agreeing on one kind of animal but not the other (Type 2 cheating), and such intention can be equally easily detected by a willing collaborator to establish a

Player Guess	**Partner Guess**	
	Turn won	Indicator resets
C	C	D
D	D	D

Table 1. How cheaters can determine their partner's input using the indicators

colluding protocol in the game. They do not have to know each other, and do not have to have agreed upon a cheating protocol before the game. Once the in-game protocol is established, the cheaters no longer need to recognise the content of any image. Therefore they can react faster than their honest opponents, and thus achieve their desired aims.

The current design of the 'Cat or Dog' game, as implemented (and described in the previous section), addresses the above issue by making the following three design choices:

i) The number of turns allowed for each round is limited to only one;
ii) The indicator is removed;
iii) 'Guess disagreement' is introduced as a game mechanism. That is, the team that first reaches a disagreement loses the current round.

By limiting the number of turns per round to one, we not only encourage players to be accurate, but also remove the necessity of an indicator. Limiting the number of turns to one per round and removing the indicator together not only makes it harder for a cheater to convey or deduce the critical information about the intentions, but also make both the game and its interface simpler and easier to follow – it is our goal to make the game and its interface as simple as possible, which we believe can be critical to make a human computation game successful. The turn limitation also slows down the pace at which two cheaters could establish a collusion protocol.

Unlike in ESP and MB, where only guess agreement is used as a game mechanism, 'guess disagreement' is introduced in our new game to serve the following purposes. First, this makes it even harder for a cheater to deduce what is entered by her partner, since there are now more possible ways than before for a team to win a round. Second, the penalty introduced by the 'first disagreement' rule also encourages players to be accurate. Common game mechanisms utilized in human computation games are summarized in [6], but no prior art has utilized 'guess disagreement'.

Overall, these three design choices will not only lower the chances of data pollution by Type 1 cheaters because it becomes much harder for them to establish a protocol with their partner, but also put off Type 2 cheaters, since they are more likely to be penalised more than they would gain whilst trying to establish a protocol. In other words, the current design of the game limits in-game (covert) communication to mitigate cheating, and encourages players to be accurate to improve the quality of labels produced by the game.

4.3 Implementation and Other Details

'Cat or Dog' is designed as a web-based game. It follows the client-server architecture, and is implemented with the Google Web Toolkit (GWT). GWT allows applications to be written in Java, and it will compile the code into optimized JavaScript for deployment. Since all major browsers support JavaScript by default, end-users do not have to install anything on their computers before they can play the game. Our game can run with all the major browsers on any operating system.

4.4 Further Cheating Mitigation

Our 'Cat or Dog' game also supports some further cheating mitigation methods, including the following.

- Player queuing and random pairing: players who log on at the same or similar time will not necessarily be paired together.
- IP address checks: to make sure that players are not paired with themselves or with people who have a similar address.
- Trap. Some images are manually labeled and will be used as trap at a random interval. If players keep getting the trap images wrong then they will get flagged as cheaters and blacklisted, all label data from them will be discarded.

Trap could be the ultimate cheating prevention method, but decreases the throughput of data produced by the game. So it makes sense to make a cheater's life harder by other means, such as those discussed earlier. In addition, it is likely that not all images used in a game session will be labelled. The unlabelled images can also include traps. This problem can be addressed by inserting more traps; however doing so could drastically lower the throughput rate of the game. Instead, it is better if any unlabelled images themselves were to be reused after a random number of rounds.

4.5 Bots

When there are not enough human players, our system will automatically enable bots to play with waiting people. Like in the MB game, we support two types of bots. One type acts as a single player by simply replaying data from old games, and the other (which we call a Tailored Response Bot or TRB for short) plays as a single team by performing actions at response times tailored to an opponent team's performance.

A TRB never has to decide which key to press. Whether or not the team represented by a TRB reaches an agreement on an image is entirely the result of flipping a coin. Typically, a TRB monitors response times of its opponent team in previous rounds of the current game session, and generates some response times for its own team around those of the opponent team (or uses predefined values at the beginning of a session). At these intervals, TRB will flip a biased

coin, with the result being either an agreement being reached or not. The bias of the coin depends on the scores of the current game session; it will be more in favor of the TRB if the opponent team is winning and less otherwise. The purpose is to keep the TRB's score around that of its opponent team, this way the stronger player(s) in the opponent team can be pushed to test their abilities and the weaker player(s) would not feel too overwhelmed.

With the support of these bots, even a single human player can play our game. That is, the human player partners with a replay bot, competing against a TRB bot. This game type can be used to either verify the correctness of the labels, or detect cheaters. Further discussions of both types of bots are in [8].

5 Evaluations

Our evaluation includes two parts. First, we show that the game is indeed enjoyable. Second, we estimate the accuracy and throughput of the data produced by the game.

Since our game has not been formally released to the public, we carried out an evaluation with a pilot study. We have advertised for volunteers to people (including visiting students) in our computer science department and a local software company. All the feedbacks were collected in an anonymous way, but an informed consent was obtained from the participants.

The images used in our game were collected from the Asirra website [3]. We wrote a Java program to automatically request images used in this CAPTCHA, and to download them into a designated folder. Traps were not used in the pilot study.

5.1 The Level of Fun

We used a questionnaire to survey the level of fun that the players experienced when playing the game. Table 2 shows the average rating (on a five point scale) to questions related to the enjoyability of the game.

A limitation of this survey is that it had a limited scale. We will be soon making our game available online, and it is our future work to gauge the fun level of our game using a much larger number of participants. We are particularly interested to see how many people will come back to play the game again, and how often they will play.

5.2 Data Quality and Throughput

Label accuracy. In our study, 68 game sessions (including incomplete ones) were played, in which in total of 1906 sample images were used, with 1613 labeled. A manual inspection shows that 1585 images were correctly labeled, giving an accuracy rate of 98.26%.

Throughput. On average, a single game session produced 13 (std dev=1.77) correct labels per minute, giving 780 labels per human hour. The average number

Question	Rating		
	mean	**std dev**	**% at 4 or above**
Did you find the game fun to play? a	3.74	0.75	68.19
Did you like playing with your partner? b∗	3.83	0.94	59.10
Are you likely to play this game again? c	3.22	1.17∗∗	36.36

Table 2. How enjoyable is 'Cat or Dog'? Average rating on the scale of 1 to 5, provided by 23 players who filled in the survey after playing the game. Higher scores are better.
a 1=No fun at all, 2=not much fun, 3=average, 4=some good fun, and 5=extremely fun
b 1=strongly disagree, 2= somewhat disagree, 3= neither agree or disagree, 4= somewhat agree, and 5= strongly agree
c 1= highly unlikely, 2= unlikely, 3=maybe, 4= very likely, and 5=definitely
∗ Bots were disabled in order to measure this fun element.
∗∗ This relatively large value was largely caused by two participants who ticked Option 1 as their answers.

of labels collected per minute in the ESP game by two players was 3.89, giving 233.4 labels per hour. It is reasonable that it takes more time on average to harvest a label in the ESP game than in the Cat or Dog game.

It is worthwhile to note that when a TRB bot is enabled in our game, the same labeling rate of 780 labels per hour will be achieved while only two human players are required. That is, two humans play in the same team, competing against a TRB. This effectively doubles the throughput per player.

On the other hand, the game supports a large number (denoted by n) of parallel sessions. The throughput of the game can be quickly scaled by a factor of n, which is constrained only by the network bandwidth and the game server's CPU and memory.

6 Conclusion

We have showed that our 'Cat or Dog' game is not only fun to play, but also produces highly accurate labels for images of cats and dogs. By labeling such images, the game can serve two security purposes. One is to enable the service of Microsoft's Asirra CAPTCHA, and the other to streamline the evaluation of Assira's robustness. In the mean while, the charity organization Petfinder.com can also deploy our game to make it both easy and fun for volunteers to create their pet catalogue.

Our game can be extended to tackle any other binary labelling problems, but a new story line might be needed for the extended game – what a story line is most suitable depends on the types of objects to be classified. To the best of our knowledge, this work is the first to discuss the weakness of the famous ESP game in terms of binary labeling, and offers the first effective alternative solution. As a whole, our game is a novel interactive system, with multiple innovations. In particular, we introduce 'guess disagreement', a novel mechanism for human

computation games. This mechanism not only makes it harder for cheaters to collude in our game, but also improves the accuracy of labels produced by the game. It is interesting future work to explore this mechanism's applicability to the design of future human computation games.

In designing our game, we have also learned an interesting lesson: By making an appropriate design choice (i.e. allowing only one turn per round in the game), we not only simplified the game, its interface and the way a player interacts with the game, but also mitigated a serious security threat. This illustrates that simplifying the design of a system can improve its usability and security simultaneously.

7 Acknowledgements

We thank all participants of our study, and thank Brian Randell, Anirban Bhattacharyya, Ahmad El Ahmad and Haryani Zakari for their help and support.

References

1. Jeremy Elson, John R. Douceur, Jon Howell, Jared Saul. 'Asirra: a CAPTCHA that exploits interest-aligned manual image categorization'. ACM CCS 2007, pp 366-374.
2. Philippe Golle. 'Machine learning attacks against the Asirra CAPTCHA', CCS 2008, ACM Press. pp 535-542.
3. MSR Asirra: A Human Interactive Proof. Available at http://research.microsoft.com/asirra/. As accessed on Apr 18, 2009.
4. Luis von Ahn and Lora Dabbish, 'Labeling Images with a Computer Game', CHI 2004, ACM Press. pp319-326.
5. L von Ahn, M Blum and J Langford. 'Telling Humans and Computer Apart Automatically', CACM, V47, No2, 2004.
6. Luis von Ahn and Lora Dabbish. 'Designing games with a purpose', CACM vol51, no8, 2008. pp58-67.
7. Jeff Yan. 'Security Design in Online Games'. In Proc. of the 19th Annual Computer Security Applications Conference (ACSAC), IEEE Computer Society, 2003. pp. 286-295.
8. Jeff Yan, Su-Yang Yu. 'Streamlining Attacks on CAPTCHAs with a Computer Game'. Proc. of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09), Pasadena, California, USA, July 11-17, 2009. pp. 2095-2100.
9. Hartmut Neven, Vasil S. Denchev, Marshall Drew-Brook, Jiayong Zhang, William G. Macready, Geordie Rose. 'Binary Classification using Hardware Implementation of Quantum Annealing', Neural Information Processing Systems conference (NIPS 2009).