Sensor Captchas
On the Usability of Instrumenting Hardware Sensors to Prove Liveliness

Thomas Hupperich[1], Katharina Krombholz[2], and Thorsten Holz[1]

[1] Horst Görtz Institute for IT-Security (HGI), Ruhr-Universität Bochum, Germany
[2] SBA Research, Vienna, Austria

Abstract. A CAPTCHA is a challenge-response test often used on the Web to determine whether a Web site's visitor is a human or an automated program (so called *bot*). Existing and widely used CAPTCHA schemes are based on visual puzzles that are hard to solve on mobile devices with a limited screen. We propose to leverage movement data from hardware sensors to build a CAPTCHA scheme suitable for mobile devices. Our approach is based on human motion information and the scheme requires users to perform gestures from everyday life (e. g., *hammering* where the smartphone should be imagined as a hammer and the user has to hit a nail five times). We implemented a prototype of the proposed method and report findings from a comparative usability study with 50 participants. The results suggest that our scheme outperforms other competing schemes on usability metrics such as solving time, accuracy, and error rate. Furthermore, the results of the user study indicate that gestures are a suitable input method to solve CAPTCHAs on (mobile) devices with smaller screens and hardware sensors.

Keywords: CAPTCHAs, Motion-based Liveliness Test, Device Sensors

1 Introduction

CAPTCHAs[3] (*Completely Automated Public Turing tests to tell Computers and Humans Apart*) are challenge-response tests used to distinguish human users from automated programs masquerading as humans. Due to the increasing abuse of resources on the Web (e.g., automated creation of web site accounts that are then used to perform nefarious actions), captchas have become an essential part of online forms and the Internet ecosystem. They typically consist of visual puzzles intended to be easy to solve for humans, yet difficult to solve for computers [17]. The same idea can also be applied to audio puzzles such that visually impaired persons can also prove that they are humans and not computer programs. In reality, however, these puzzles are often time-consuming and sometimes hard to solve for human users [6]. Furthermore, visual pattern recognition algorithms gradually improved in the last years and this makes automated captcha solving feasible. For example, Burzstein et al. [3, 4] highlighted

[3] For better readability, we write the acronym in lowercase in the following.

that due to the arms race between captcha designers and OCR algorithms, we must reconsider the design of (reverse) Turing tests from ground up. As such, there is a continous arms race to design captcha schemes that are secure against automated attacks, but still useable for humans.

In the last few years, mobile devices have become a primary medium for accessing online resources. While most web content has already been adjusted to smaller screens and touchscreen interactions, most captcha schemes still suffer from these usability constraints and are perceived as error-prone and time-consuming by their users: several studies demonstrated that captcha usability in the mobile ecosystem is still an unsolved challenge [3–6, 15, 20], According to Reynaga et al. [14], captchas are primarily evaluated on their security and limited usability work has been carried out to evaluate captcha schemes for mobile device usage. With the emerging proliferation of wearable devices such as smartwatches, it becomes inevitable to re-think user interactions with captchas in order to successfully tell humans and computers apart, without placing the burden on users that struggle with hard-to-solve visual or audio puzzles.

In this paper, we present *Sensor Captchas*, a captcha scheme designed for mobile devices. Based on previously published findings, we collected a set of design recommendations to tie our design decisions to. We propose motion features from hardware sensors as a novel input paradigm for mobile captchas. A user is expected to perform gestures from everyday actions which might either be know or imagined easily, such as for example *hammering* where the smartphone should be imagined as a hammer and the user has to hit a nail five times, or *drinking*, where a user is asked to drink from the smartphone, imagining it is a glass of water. Our approach is solely based on state-of-the-art sensors available in most smartphones and wearables such as gyroscope and accelerometer, and obviates the need for users to solve complex graphical puzzles on small screens.

We implemented a prototype of the proposed scheme and present a repeated measures user study to compare our approach to state-of-the-art visual captcha schemes (namely reCAPTCHA and noCAPTCHA[4]) as well as an innovative mechanism called *Emerging Image captcha* [18].

Our findings show that sensor data is a suitable input for captcha challenges with high success rate and low solving time when leveraging gestures. While some gestures are easier to solve than other movements, the overall rate of solving successes shows the feasibility of our approach. Users rated our captcha mechanism comparable to established captcha schemes and we are able to show a learning effect within the first 15 challenges.

In summary, we make the following contributions:

- We design an extensible captcha scheme using accelerometer and gyroscope data as user input and machine learning classification for challenge validation.
- Based on a prototype implementation of the proposed scheme, we conduct a thorough user study with 50 participants to evaluate the usability of our approach, including a survey for direct user feedback.

[4] noCAPTCHA is also referred to as *new reCAPTCHA* [9]

- We compare our approach to well-known, established captcha methods (reCAPTCHA and noCAPTCHA) as well as another innovative scheme (Emerging Images) regarding success rates, solving times, and user experience.

2 Related Work

Captchas are a controversial topic discussed amongst researchers and practitioners. The main reason for this observation is the fact that captchas put a lot of burden on a user, while they are often not reliable when it comes to distinguishing human users from automated programs. Many approaches have been presented in scientific literature and by companies such as Google, but most of these schemes are still susceptible to different types of attacks.

Bursztein et al. identified major shortcomings of text captchas and proposed design principles for creating secure captchas [4]. They focus on interaction with desktop computers and do not consider usability shortcomings of captcha interactions on mobile devices. Fidas et al. validated visual captchas regarding their solvability based on empirical evidence from an online survey [6]. They found that background patterns are a major obstacle to correctly identify characters, but provide little to no additional security. Reynaga et al. presented a comparative study of different captcha systems and their performance when accessed via a smartphone [14]. They argue that visual captchas are hard to solve on mobile devices and that usability could be increased by limiting the number of tasks and by presenting simpler and shorter challenges with little or no obfuscation. Furthermore, distraction from the main task should be minimized by presenting unobtrusive captchas that are isolated from the rest of the web form. These factors highlight the need to develop novel captcha schemes that overcome the limitations of visual captchas. Reynaga et al. also conducted a comparative user study of nine captcha schemes and provided a set of ten specific design recommendations based on their findings [15]. Bursztein et al. reported findings from designing two new captcha schemes at Google and presented findings from a consumer survey [5]. Xu et al. [19] explored the robustness and usability of moving-image video captchas (*emerging captchas*) to defeat the shortcomings of simple image captchas and discussed potential attacks. Jiang et al. proposed gesture-based captchas that obviates the need to type letters by using swipe gestures and other touch-screen interactions additionally [12]. However, such complex methods may state a high burden to users. Gao et al. proposed a captcha scheme utilizing emerging images as a game [7]. Such game-based captchas are solved and validated client-side making them vulnerable to replay attacks.

reCAPTCHA and noCAPTCHA by Google Inc. are field-tested, established mechanisms [8]. However, both methods disclose unapparent downsides: reCAPTCHA is used to digitalize street view addresses as well as books and magazines. noCAPTCHA implements behavioral analysis and browser fingerprinting. Information that is used for fingerprinting includes but is not limited to: installed browser plugins, user agent, screen resolution, execution time, timezone, and number of user actions – including clicks, keystrokes and touches – in the captcha frame. It also tests the behavior of many browser-specific functions as well as CSS rules and checks the rendering of canvas elements [1]. While these information are used for liveliness detection and therefore fit the aim of captchas, it can also be used for thorough user tracking, raising privacy concerns [11].

3 Sensor Captchas

Modern mobile devices contain a multitude of hardware sensors, including accelerometers and gyroscopes which are accessible via Web techniques like JavaScript and HTML5. These sensors are so accurate that it is possible to detect steps of a walking person [16] and to distinguish between certain user actions [10]. As a main difference to existing captcha schemes, we utilize these hardware sensors as input channel for solving a challenge. The benefit of this input channel is that a user does not need to type text on a small softkeyboard on a smartphone, but he can use a simple movement to prove liveliness.

In practice, a Web site provider aims to distinguish a human user from an automated bot and therefore utilizes a captcha challenge. In our approach, this challenge is represented by a gesture a user has to perform. We explored possible gestures for such challenges as they need to satisfy several requirements:

- **Understandable:** Users need to be able to understand the challenges and what they are supposed to do immediately.
- **Accurate:** The challenge needs to enable a precise differentiation between human users and automated bots.
- **Deterministic:** The choice whether a human or a bot is currently visiting a Web site needs to be deterministic.
- **Solvable:** It must be possible to solve the challenge within a reasonable amount of time.

3.1 Gesture Design

In an early stage of our research, we chose very simple gestures like moving a device in a circle clockwise. While these movements were easy to understand by a user, it was hardly possible to precisely distinguish between gestures due to too much variance: we did not include any precise statements about size and speed of the movement, so users were not able to solve these challenges accurately. Learning from these findings, we chose five gestures for our user study which are derived from everyday actions a user might either know or imagine easily:

- **Hammering:** The smartphone should be imagined as hammer and a user has to hit a nail five times.
- **Bodyturn:** A user is challenged to turn all around counter-clockwise.
- **Fishing:** The smartphone should be imagined as fishing rod which is to cast.
- **Drinking:** A user is asked to drink from the smartphone, imagining it is a glass of water.
- **Keyhole:** The smartphone is an imaginary key which is to be put in a door lock and rotated left and right like unlocking a door.

Note that these gestures can be easily extended, e. g., by randomly choosing the number of times the “hammer” has to hit the imaginary nail or by taking a clockwise bodyturn into account. With such variations, more gestures are possible so that in a practical use not only five movements are available, but a great variety of different challenges can be designed. The gestures can be presented to users in different ways. For our prototype and user study, we described all gestures to perform in short texts. Pictures showing a drawing of a human performing the asked movement or even an animated image or a short video clip can alternatively present the challenge.

When a user performs a gesture, accelerometer and gyroscope readings are recorded and afterwards transferred to the Web server. On the server side, we use a machine learning classifier to determine whether the sensor data matches the challenged gesture. If the data can be classified as the demanded gesture, the captcha has been solved successfully. If it is rejected by the classifier or matches a wrong gesture, the captcha has failed. Using machine learning technology in our captcha scheme is based on the following observation: If a captcha is based on text input, the challenge text is generated first and held by the server. When the user enters the text, this input can be compared to the generated text immediately. In our scenario, there is no challenge data generated first which the user input can be compared to. It is not usable to generate three-dimensional acceleration data and challenge a user to perform exactly this movement with a smartphone. Hence, we need a decider which is capable of distinguishing characteristics of one movement from another and ultimately determine whether given data matches a specific gesture. A machine learning classifier is an appropriate mechanism for this task as it describes a classification problem.

3.2 Satisfaction of Requirements

We ground our captcha scheme in design principles suggested in existing scientific work on captcha usability, such as Reynaga et al. [14], Fidas et al. [6], and Bursztein et al. [3]. In the following, we present a collection of design principles and recommendations from these publications and argue how our design addresses these features.

Challenges

– *Deploy one task only.* Optional features hinder usability on small screens where captcha solving is already more time-consuming than on desktop computers. Challenges should be designed with a one-task only focus.
– *Leverage complexity.* Visual puzzles suffer from an arms race between captcha providers and pattern recognition algorithms that sometimes even perform better than human beings. Although finding a more difficult problem in computer vision will increase the cognitive load on the user side, captchas need to be challenging and of a complex domain.
– *Using cognitive behavior.* Everyday life movements such as the one used for our challenges are capable of shifting captcha interactions to a domain beyond visual puzzles and touchscreen interaction. As the gestures are found in everyday life, we believe it is an easy task for humans to perform them, yet hard to fake for automated programs.

- *Strive for a minimalistic interface.* An interface should focus on the essential and be minimalistic. Our captcha challenges can be displayed and solved even from wearables such as smartwatches.

Environment of Use

- *Expect common conditions.* Features which may fail in commonly expected environmental conditions should be avoided. Our design fulfils this recommendation although the performance of gestures may be conspicuous.
- *Minimize load.* For our approach, bandwidth usage is minimized as challenge descriptions are provided verbatim. Also, the data transmitted to the server consists of raw sensor data, as the decision whether the captcha was solved directly is performed on the server side to prevent attacks on the client.
- *Rely on default software.* For correct operation, a scheme should not rely on technologies that cannot be assumed obligatory. Our implementation is based on JavaScript which is supported by standard mobile browsers.

Engineering

- *Ensure compatability.* To reach a majority of users, input mechanisms should be cross-platform compatible and not interfere with normal operations. Our approach is solely based on input from motion sensors which are state-of-the-art in smartphones and smartwatches.
- *Aim for high robustness.* Errors must not interfere with normal operations of the browser. Our scheme does not interfere with other operations.
- *Support isolation.* The captcha challenge should be separated from the rest of the Web form. Our captchas may even be shown on another site of a form.
- *Enable consistency.* Orientation and size of the captcha should be kept consistent with the rest of the web form. As our challenge description is text-based or image-based, its presentation can easily be adjusted.

Privacy

- *Maximize user privacy.* Additionally to the design principles listed above, we aim to spotlight user privacy. A user input should not be replaced by user fingerprinting as seen in [1]. Our goal is to propose a scheme that minimizes the impact on user privacy and works without collecting sensitive information on the users and their devices.

4 User Study

We implemented a prototype of the proposed scheme and conducted a comparative evaluation to assess the usability of our new captcha scheme against already existing solutions. In the following, we provide details on both aspects.

4.1 Design and Procedure

Our user study is divided into two phases: first, a preliminary study was carried out to determine a suitable time frame for gesture performance, the best parameters for the machine learning classifier as well as the ground truth for the main user study. Both phases are described in more detail below. Figure 1 illustrates the complete user study setup.

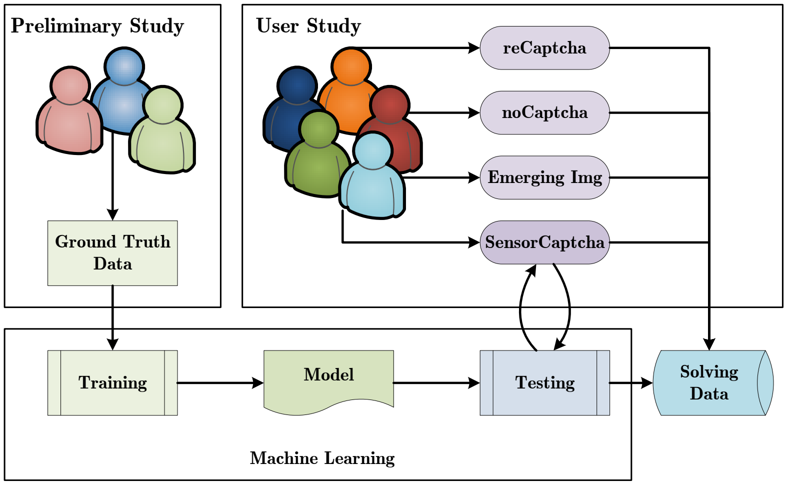


Fig. 1. User Study Setup

Preliminary Study

Sensor Captchas rely on motion input from hardware sensors and machine learning techniques to prove that the user is human. In order to train a model, we conducted a preliminary study. We built a data set of ground truth by instructing 20 participants to perform the movements and gestures described in Section 3. Then, we let them solve the challenges in a controlled environment under the following two conditions:

1. The challenges were not chosen randomly but assigned to the participants. Every user had to perform the same number of challenges. More precisely, every user performed every gesture three times.
2. We observed the users solving the challenges and instructed them if they made mistakes to ensure the correct performance.

The sensor data obtained in this preliminary study is used as ground truth for further processing. As the data collection was performed in a controlled environment and under the supervision of two experimenters, we assume that the gestures have been performed correctly.

To find the best-performing classifier, we conducted cross validations and classification experiments with algorithms from different families, including support vector machines, k-Nearest Neighbor, and different ensemble methods. Our results suggest that a *Random Forest* classifier performs best on our ground truth and thus we used this algorithm to generate a model that was then used in the actual user study.

Main User Study

We include three other captcha mechanisms besides Sensor Captchas in our main study: two schemes are well-known and commonly used in practice, while the other one is an experimental approach from a recent research paper:

1. *reCAPTCHA* is a well-proven text-based input mechanism. A user is asked to type words or numbers shown in an often distorted or blurred image.
2. *noCAPTCHA* is the field-tested successor of reCAPTCHA and challenges the user to select from nine images all these showing specific items, e. g., trees. It also instruments behavioral analysis.
3. *Emerging Images* relies on moving image recognition. A user has to type letters which are shown in an animated image series. This method has been proposed by Xu et al. [18].

While reCAPTCHA and noCAPTCHA are established mechanisms already used by Internet users and website providers every day, Emerging Images and Sensor Captcha represent scientific approaches and have not yet been deployed in a real-world environment.

We chose a repeated measures design for our lab study, i.e., every participant had to solve puzzles from every captcha scheme in a controlled environment at our university campus. It was important to us to observe sources of errors in order to improve our design. Each participant was asked to solve a minimum of 15 challenges per scheme. We designed our study to present the challenges in randomized order to reduce any bias or fatigue effects. As all participants were asked to solve captchas of all four types, we were able to gather comprehensive solving data, including the number of correctly solved captchas and failures as well as the amount of time needed to solve each captcha. As our implementation was written in JavaScript, the participants were encouraged to use their own devices to avoid bias and distractions from the study tasks due to unfamiliarity with the device. Even though we had two backup devices with us, all participants used their own devices.

After completing the captcha challenges, the participants filled out a short questionnaire (see Section 5.3 for a complete listing of these questions). In addition, one experimenter took notes in order to collect qualitative in-situ reactions and comments. This information was collected to understand particular difficulties and misunderstandings about the presented puzzles and the way of solving them. We believe these explorative findings are valuable to improve the usability of our captcha scheme.

4.2 Implementation

reCAPTCHA as well as noCAPTCHA are operated by Google Inc. and provide an API which we used to include these methods in our study. The Emerging

Images technique has been provided and hosted by Gerardo Reynaga, School of Computer Science at Carleton University, Ottawa Canada for the duration of our test. We implemented our Sensor Captchas and a survey site from which the participants accessed the different captcha challenges and the questionnaire. The web site was implemented in JavaScript and contained a general information page and a separated page for every captcha method. Each of these pages contained a short description on how to solve this captcha and a start button. After tapping the start button, a form containing the captcha challenge and a submit button were displayed. For every captcha, we measured the solving time as duration between tapping the start button and tapping the form submit button. Hence we only measured the time it took the user to mentally process the captcha challenge and to input a correct solution. This way, we managed to measure the solving time irrespective of network delays, implementation issues and other technical factors. After a captcha challenge was completed, we stored the following information: A **name** every user could choose freely, the current **date**, the **captcha result** which is either success or failure, the **duration** a user needed for the solving attempt, and a **unique user key** which was generated automatically and stored in the browser's local storage as anonymous identifier.

reCAPTCHA and noCAPTCHA provide an API, so this information could be obtained and stored automatically except for one limitation: noCAPTCHA does not provide a way to check the result of a single challenge. If a challenge has not been solved correctly, the next challenge is displayed to the user automatically without triggering a Javascript event. Hence, it is not possible to record noCAPTCHA failures without interfering with Google's API and violating the way it works which may have voided results and measurements. As there is no API available for Emerging Images Captcha, we manually kept track of successes, failures, and solving durations and entered this data by hand.

Regarding Sensor Captchas, we additionally stored the following information: The **sensor data**, including accelerometer and gyroscope readings as arrays (of the dimensions x, y, and z as well as α, β, and γ), the **original challenge** which was displayed to the user, and the **classification result** which leads to a captcha success only if it matches the original challenge.

After tapping the submit button on the Sensor Captcha page, sensor events were measured for five seconds which we set as time frame to perform the gesture. We designed the gesture movements in such way that they are practical to perform within this time and tested every gesture beforehand. Our preliminary study showed that five seconds are a reasonable amount of time to make all required movements. Though, this parameter can be analyzed and adjusted in future studies. After this time, all data was submitted automatically, so that users did not have to tap another submit button in addition. The sensor data was sent to a socket parsing the data to our machine learning classifier, retrieving the classification result and finally storing all these information in the database. This functionalities were programmed in Python, implementing a Random Forest classifier from scikit-learn [2].

4.3 Recruitment and Participants

We recruited 50 participants between December 2015 and February 2016 at the university campus and a major computer security conference. Most participants were students at our university from different branches of study, including information technology, medicine, arts and science. While the youngest participant was 18 years old and the oldest 55, the majority was aged between 20 and 35 years; male and female in approximately equal shares. All participants were familiar with the purpose of captchas on websites and reported to have used established methods before. To comply with ethical guidelines from our university, we did not collect any personal identifiable information. We only collected information on age, gender and whether the participants had a background in information technology. Every session lasted about 20 minutes per participant and they were compensated for their time with a voucher of a major online shop.

5 Evaluation

In the following, we compare the different captcha schemes regarding successfull solving of challenges and amount of time needed to solve challenges. Concerning Sensor Captchas, we analyze the suitability of gestures as well as the survey included in our user study. Finally, we investigate whether a habituation effect can be asserted and shed light on the precision of our machine learning classifier.

5.1 Comparison of Mechanisms

To compare the solvability among all considered captcha mechanism, we measured the successes and failures. A success represents the correct solution of a captcha, while a failure represents a wrong input.

In our study, about 85 % of all reCAPTCHA challenges were successfully solved by the participants. As discussed in Section 4.2, it is not possible to catch success and failure cases of noCAPTCHA without interfering. Emerging Images seem to constitute a rather hard challenge, as only about 44 % of all challenges could be solved correctly. In contrast, Sensor Captchas achieve a high success rate: Of all provided gesture challenges, the participants were able to correctly solve about 92 % , making this mechanism to be reckoned with. These preliminary results of our study suggest that users were able to solve more Sensor Captchas correctly than challenges of any other type. Note that for Sensor Captchas, a failure may not only redound upon a wrong user input – namely not performing the challenge gesture – but also upon a misclassification by our machine learning algorithm. This factor will be discussed below.

As described in Section 4.2, we measured the time users needed to solve every single challenge. Hence, we can analyze how much time is needed on average to succeed at every mechanism. Table 2 shows the average amount of time per mechanism and captcha result.

We observe that in general failures take more time for reCAPTCHA as well as Emerging Images. The reason for this lies probably in the way of user input:

Table 1. Success rates (SR)

Mechanism	SR	Mean	SD
reCAPTCHA	0.8463	0.8698	0.3356
Emerging Images	0.4396	0.4491	0.4976
Sensor Captcha	0.9160	0.4813	0.4997

Table 2. Average solving times

Mechanism	S	F	Total	Mean	SD
reCAPTCHA	12.22	26.36	14.39	12.4260	18.5934
noCAPTCHA	-	-	26.99	24.1814	17.8862
Emerging Images	21.91	24.29	23.24	26.1504	29.4114
Sensor Captchas	12.35	8.85	12.05	12.2519	7.10444

SR = success rate, S = successes, F = failures, SD = standard deviation

Users have to read and decipher letters or numbers first. Depending on the specific challenge, this may be difficult so that hard challenges are more likely to fail but also take more time. We observed these cases to annoy many users as they first needed to invest high effort to recognize the challenge's letters or numbers and then fail anyway. For Sensor Captchas, we can see a lower solving time for failures than for successes, indicating that users may have failed to solve the challenge because they did not read the description text carefully enough.

We found noCAPTCHA to take generally more time than reCAPTCHA, which may be explained by the fact that reCAPTCHA applies browser fingerprinting first and then displays the challenge. Comparing the total time users were taken to solve captchas, reCAPTCHA is the fastest mechanism – probably because it is a practical method many users are already familiar with. Nevertheless, reCAPTCHA is directly followed by Sensor Captchas, suggesting that this approach is practicable and showing that users are able to perform the challenge's gestures in a reasonable amount of time. Please note that Sensor Captchas' solving time can be influenced by adjusting the time window for performing a gesture. We based an interval of five seconds upon our preliminary study but increasing this time would result in higher solving durations while decreasing could make it impossible to perform a gesture thoroughly.

Our study has a repeated-measures design, so every participant was exposed to every condition. Therefore, we analyzed our data with repeated measures analyses of variance (ANOVAs). Table 1 shows not only the success rates of the captcha mechanisms but also their mean and standard deviation of successes, represented by 1 for success and 0 for failure. We see that the mean of Sensor Captchas resides within the standard deviation of reCAPTCHA and vice versa. Hence, differences between these two schemes are statistically not significant and may represent random errors. In contrast, the correct solving rate of Sensor Captchas is significantly higher as of the Emerging Images mechanism, meaning that even if the random error is considered, the succcess rate of Sensor Captchas is superior. Similar trends can be observed regarding the solving times of each mechanism in Table 2: There is no statistically significant difference between Sensor Captchas and reCAPTCHA regarding the time a user takes to solve a captcha. Though, the mean solving times of these two mechanisms are significantly lower compared to noCAPTCHA and Emerging Images. We can conclude that Sensor Captchas and reCAPTCHA can be solved faster than noCAPTCHA and Emerging Images, even if the random error is taken into account.

5.2 Gesture Analysis

After comparing Sensor Captchas to other captcha mechanisms regarding success rates and solving times, we aim to analyze the gestures in detail. We conducted experiments to ascertain which gestures are accurate to perform and which movements happen to be related to other gestures. Table 3 shows the solving rates and error rates per gesture. We see that `bodyturn` and `keyhole` challenges were in general solved correctly, meaning that the sensor events measured during a user's gesture performance could be matched to the challenged gesture. `Bodyturn` and `keyhole` were correctly solved by about 97 % and 96 % in total. For both, the highest mismatching was to the `hammering` gesture, meaning if a user input could not be related to the challenge, it was classified as `hammering`. For the `drinking` movement, still about 92 % of the challenges were solved correctly. The gestures `fishing` and `hammering` seem to be prone for errors: Of all `hammering` challenges, about 85 % could be solved correctly and in case of the `fishing` gesture only about 79 % . We also see that `fishing` and `hammering` are the least precise gestures as about 14 % of all `fishing` challenges were classified as `hammering` and about 5 % of all `hammering` challenges were mistakenly related to the `fishing` gesture. This confusion can be explained by the movement itself: For `hammering`, users had to move their devices in one axis up and down, so this gesture is not very complex. For `fishing` applies the same as this movement also involves only one axis and although there are differences like the number of accelerations (`hammering` requires several acceleration moves in order to hit the imaginary nail five times while the fishing rod is casted only once), this low complexity leads to confusion about these two gestures. For the same reason, the `fishing` gesture was sometimes classified as `drinking`, although this happened only in about 4 % of all fishing challenges. In about 8 % of all `hammering` challenges, the sensor data was related to the `keyhole` gesture. The reason for this might be that users may have slightly turned their phones while `hammering` their devices on an imaginary nail. This resulted in movements in the z dimension which is an essential part of the `keyhole` gesture. The gestures `drinking`, `keyhole`, and `bodyturn` show only negligible errors and mistaken relations to other gestures. In general, only the `hammering` gesture yields potential for errors and should be excluded or enhanced in further studies. If this is fixed, the `fishing` gesture may presumably perform better as well because there will no confusion with the `hammering` movement any more.

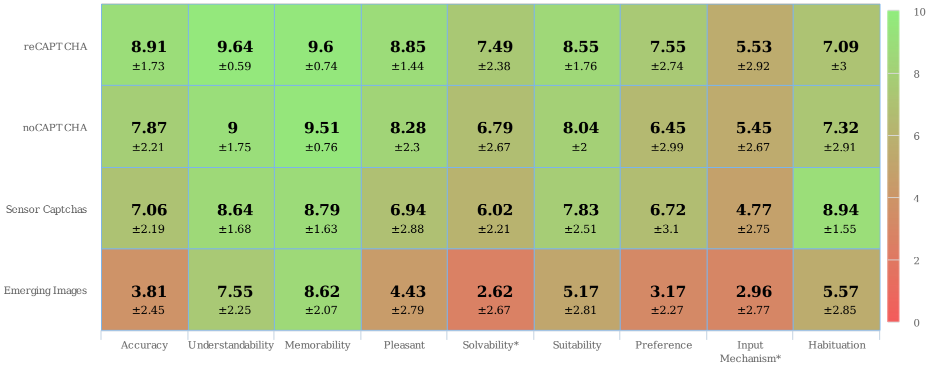
Table 3. Solving rates and error rates per gesture

	Categorized as				
Gesture	`bodyturn`	`drinking`	`keyhole`	`fishing`	`hammering`
bodyturn	0.9720	0.0	0.0	0.0	0.0279
drinking	0.0	0.9174	0.0642	0.0091	0.0091
keyhole	0.0065	0.0130	0.9608	0.0	0.0196
fishing	0.0222	0.0444	0.0	0.7889	0.1444
hammering	0.0162	0.0	0.0813	0.0487	0.8537

5.3 Survey Results

As a part of our study, users had to participate in a survey, rating all four captcha mechanisms regarding nine aspects. We leveraged a ten-levelled Likert scale for every item, adopted and extended statements from previous research by Reynaga et al. [15] to allow a direct comparison to this work. In detail, we let the users rate the following statements (* represents inverted items):

- **Accuracy**: It was easy to solve the challenges accurately.
- **Understandability**: The challenges were easy to understand.
- **Memorability**: If I did not use this method for several weeks, I would still be able to remember how to solve challenges.
- **Pleasant**: The captcha method was pleasant to use.
- **Solvability***: It was hard to solve captcha challenges.
- **Suitability**: This method is well suitable for smartphones.
- **Preference**: On a mobile, I would prefer this captcha method to others.
- **Input Mechanism***: This method is more prone to input mistakes.
- **Habituation**: With frequent use, it get easier to solve the challenges.



	Accuracy	Understandability	Memorability	Pleasant	Solvability*	Suitability	Preference	Input Mechanism*	Habituation
reCAPTCHA	**8.91** ±1.73	**9.64** ±0.59	**9.6** ±0.74	**8.85** ±1.44	**7.49** ±2.38	**8.55** ±1.76	**7.55** ±2.74	**5.53** ±2.92	**7.09** ±3
noCAPTCHA	**7.87** ±2.21	**9** ±1.75	**9.51** ±0.76	**8.28** ±2.3	**6.79** ±2.67	**8.04** ±2	**6.45** ±2.99	**5.45** ±2.67	**7.32** ±2.91
Sensor Captchas	**7.06** ±2.19	**8.64** ±1.68	**8.79** ±1.63	**6.94** ±2.88	**6.02** ±2.21	**7.83** ±2.51	**6.72** ±3.1	**4.77** ±2.75	**8.94** ±1.55
Emerging Images	**3.81** ±2.45	**7.55** ±2.25	**8.62** ±2.07	**4.43** ±2.79	**2.62** ±2.67	**5.17** ±2.81	**3.17** ±2.27	**2.96** ±2.77	**5.57** ±2.85

Fig. 2. Mean Likert-scores and standard deviations from survey

Figure 2 reports the mean Likert scale responses from *strongly disagree* = 1 to *strongly agree* = 10. Also, the colors in the figure represent the scale, from red representing *strongly disagree* to green as *strongly agree*.

The established captcha mechanisms in our study – namely noCAPTCHA and reCAPTCHA – were in general rated high regarding accuracy, understandability, memorability, pleasant use, and suitability for mobile devices. Many users stated that they were familiar with these methods and therefore could easily solve the given challenges as the task was immediately clear. For understandability and memorability, we observe a low standard deviation among the ratings. In contrast, high standard deviation among participant ratings can be seen regarding the preferred captcha mechanism. This item holds a deviation of 2.99 for noCAPTCHA and 2.74 for reCAPTCHA, showing that users are at odds if they preferred these established methods which is substantiated by high standard deviation regarding input mistakes (“input mechanism”) showing 2.67 for noCAPTCHA and 2.92 for reCAPTCHA. For some users, these captchas seem to work well and are easy to use. Anyway, other users are not comfortable with them and would not prefer these methods on mobile devices.

Although Sensor Captcha holds the highest solving rate, users are not accustomed to this mechanism which results in a generally lower rating compared to the established captcha methods reCAPTCHA and noCAPTCHA. Sensor Captchas keeps up with established mechanisms regarding accuracy, understandability, memorability, suitability, preference and input mechanism – differences of these ratings are smaller than one. Significant differences can be seen regarding the ratings “pleasant” which may be rooted in the fact that the participants were not used to Sensor Captcha and the gestures require movement of body(parts) which users may be uncomfortable with in public environments and “solvability”. This is contradictory to the high solving rates and shows that users find it hard to solve Sensor Captchas although they were able to do so in most cases. The high rating of habituation shows that participants adjudge a high learnability to Sensor Captchas, hence long term studies may improve the perception of solvability as well. We also shed light on habituation aspects in the next section. The items of our questionnaire which were rated with a low value also show a high deviations: While “pleasant”, “preference”, and “input mechanism” show the lowest user ratings, the standard deviations are rather high with 2.88, 3.1, and 2.75. This shows that there is a wide array of user opinions and while some participants found Sensor Captcha not pleasant and would not prefer this mechanism, other users indeed stated the opposite and would prefer our mechanism to established captcha methods. Furthermore, the lowest standard deviation of 1.55 holds “habituation” which states that the majority of users think that continuous use would increase the solvability and easy-of-use of Sensor Captcha.

Emerging Images as another innovative captcha mechanism was rated well regarding understandability and memorability showing that users are familiar with text inputs and understand the task of typing letters from a sequence of images easily. Anyway, participants found it hard to solve these challenges, given a low rating of accuracy, solvability, and pleasant-of-use. This might be the reason why most users would not prefer this method and stated that it is prone to errors (“input mechanism”). In contrast to Sensor Captcha, users are not optimistic whether a continuous use of Emerging Images may improve the solvability and handling, though, “habituation” holds the highest standard deviation of 2.85 for Emerging Images which shows that some users may get familiar with it.

Informal Participant Statements

Participants were free to leave comments, so we could get a more detailed feedback on our study and scheme. Many users demanded animations for the description of gestures. As this may probably improve the understandability, accuracy, and solvability of Sensor Captchas, we will implement this feature in the future.

A few users stated that the chosen gestures were not suitable for everyday use. Indeed, for Sensor Captchas to evolve into an established captcha method, the available gestures need to be reassessed. We abstracted gestures from everyday actions because simple movements were prone to errors and misunderstandings (see Section 3). Still, casting an imaginary fishing rod may be imaginable but not an action users want to perform in public environments.

Some users stated that it is hard to solve text-based and image-based captchas – reCAPTCHA and noCAPTCHA – on a smartphones screen because it may be too small to comfortably display all images or the softkeyboard additionally to the challenge. This supports our original motivation for Sensor Captcha.

5.4 Habituation

According to the survey results, many users think that solving Sensor Captchas will get more and more comfortable and easy by using the scheme. Although the long term habituation to Sensor Captcha is left for future work, we investigate if users were able to improve their success rates during our user study. Like described in Section 4.1, every user tried to solve at least 15 Sensor Captchas. While only about 49 % of all participants were able to solve the very first Sensor Captcha correctly, we notice a clear trend that more gestures could be performed successfully the more captchas have been tried to solve. The average success rate among all users for the 15th Sensor Captcha is about 84 % which supports the assumption that users may probably habituate to this captcha mechanism fast. To test a possible correlation between the number of solving attempts and the number of successes, we calculate the Pearson correlation coefficient ρ. Taking all user data into account, $\rho = 0.7238$, which proves a strong positive linear relationship statistically and verifies that with increasing number of challenges the number of successes also increases in our user study.

5.5 Classification

There exist two possible factors for captcha failure: Not only humans may fail at solving a captcha challenge, but the machine learning classifier may fail at matching correct gesture data. To shed light on possible false classifications, we calculated precision and recall for different machine learning algorithms. In our scenario, a *false positive* is represented by the case that sensor data not belonging to a specific gesture will be accepted as correct solution for this gesture; in extreme case random sensor data is wrongly classified as correct solution to a challenge.

Consequently, if a correct sensor data input is mistakenly rejected by the classifier, this case states a *false negative*. Note that in context of captchas, false positives are worse compared to false negatives: If users were sporadically not recognized as human, they would have to solve a second captcha at worst. But if a bot was mistakenly accepted as human, it could circumvent the captcha protection. Correct classification of sensor data to the right gesture is a *true positive*, while a correct rejection of non-matching data constitutes a *true negative*. On this basis, we are able to calculate precision and recall of all data obtained in the user study. Figure 3 illustrates precision recall graphs of different classifiers which were to be considered.

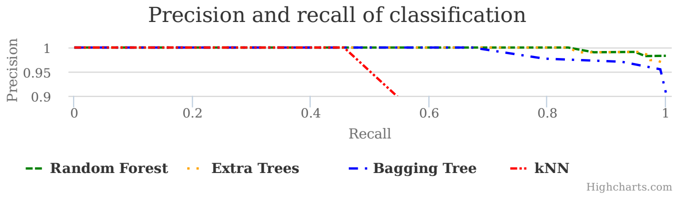


Fig. 3. Classification precision and recall

Given the data set of our user study, including accelerometer and gyroscope data of all performed gestures, the classifiers *Random Forest*, *Extra Trees*, and *Bagging Tree* yield a very high precision in distinguishing the gestures. Only the *kNearestNeighbor* algorithm (testing $k = 1$, $k = 5$, $k = 10$) was not capable of precisely classifying the gestures. While this one achieves an AUC of only 0.7899, Bagging Tree achieved an AUC of 0.9917, Extra Trees of 0.9972 and finally Random Forest of 0.9989. This confirms our choice to implement a Random Forest classifier in our user study back end. As shown in Figure 3, the classifier is capable of determining whether given sensor data satisfies a specific gesture at high precision. Hence, misclassifications are negligible in our study and we are able to ascribe most captcha failures to user input errors.

6 Discussion

The proposed mechanism meets the common requirements to captcha schemes: the main goal of telling computers and human apart by a challenge as simple as possible is achieved. We also satisfy common design principles for captcha methods as discussed in Section 3.2. In this section, we discuss security as well was potential limitations of our approach, and ideas for future work. Although our survey results indicate that users feel Sensor Captchas to be less accurate and solvable than established methods, our approach achieved the highest success rate and took users the least time to solve challenges. It thus might break the arms race in computer vision powered by more and more captcha mechanisms based on visual puzzles. The fact that the decision about success or failure is made server-side raises the bandwidth use in contrast to captcha schemes which work client-side only. However, the size of transferred sensor data is reasonable and deciding about a challenge's solution server-side is more secure. On average, the sensor readings of accelerometer and gyroscope take 5 KB in total.

6.1 Security Considerations

Basing liveliness determination on hardware sensor data enables new attack vectors aiming at data manipulating. An attacker may record sensor data and provide it as solution to a given challenge. As our captcha scheme currently supports five gestures only, a replay attack succeeds with a theoretic probability of 0.2 which needs to be reduced by more varieties of gesture challenges. Thus, even with such extensions, the entropy of our approach will not exceed the entropy of text-based captchas. A bot could solve Sensor Captcha challenges if correct input data is available for every single gesture and if the automated solver furthermore is able to recognize the challenge presented. As this applies to all common captcha schemes, it also applies to our approach. While an attacker may perform all gestures once and record the corresponding sensor data, the hardness of challenge recognition is essential for most captcha schemes. The security of text-based captchas especially relies on the assumption that the challenge is hard to identify. To harden a scheme against this attack vector, the way of presenting challenges could be randomly chosen to complicate automated detection.

Alternatively, an attacker could try to exploit the machine learning classification by replaying data of a different challenge than the presented. To test this behavior, we conducted a replay attack experiment choosing sensor measurements including accelerometer data and gyroscope data from the user study and attempt to solve a given challenge. We repeat this procedure 500 times to simulate such replay attacks under the same conditions like in our user study. Note that we do not use random sensor data but real-world sensor readings we obtained in our user study before. Leveraging completely random data may also be a possible scenario, but a less sophisticated attack. As a result, in two cases a sensor data replay of an original `fishing` challenge was misclassified as `hammering` leading to a false positive. One replay of a `hammering` gesture was accepted as solution to the `keyhole` challenge. As we already know, `hammering` tends to be misclassified (see Section 5), so diversifying this gestures may harden our system against this type of attack. All the other attacks – making a share of 99.4 % – were correctly rejected by our machine learning algorithm.

If a user's mobile is treated as untrusted or maliciously infected device it may falsify sensor data. This would enable to tamper user input used for solving the presented challenge. However, if malware is able to change the input – e. g., by manipulating system drivers requiring root access or by tampering the browser environment –, no captcha scheme can guarantee a correctly transferred input.

We designed our system in a way that the decision whether or not a captcha is solved successfully is made server-side. If it was made client-side like in game-based captchas [7], replay attacks might be more feasible as the attacker would only have to replay the decision instead of determining the challenge and provide previously recorded data for solving.

Finally, we focussed our studies on the general feasibility of sensor-based motion captchas and especially on usability aspects.

6.2 Limitations

Our work builds on existing captcha designs and lessons learned from previous studies. As we focussed on usability aspects of captchas, we assume that the implementations of our captcha schemes are secure and best-case implementations. A limitation of our prototype implementation is that it is a proof-of-concept and was first tested on users in the course of this study. Also the set of challenges our system provides is not sufficient to be resilient to replay attacks in practice.

For our comparative user study, we recruited participants around the university campus, hence our sample is biased towards this particular user group. Also, the participants solved the captcha puzzles in a controlled environment while an experimenter was present. We did not deploy our captcha scheme in the wild and therefore do not have data on the captcha performance in a real-world setting where users have to deal with environmental constraints. Also, we did not collect any evidence on whether the our scheme is applicable in all real-world situations, such as when a user performs a task on the phone while in a meeting. Due to the fact that sensor captchas require the user to move their device, they are potentially not applicable in some situations where a less obtrusive approach would be preferred by most users. We still believe that our results provide valuable insights to how users interact with the different types of captchas. We found that metrics like solving time, memorability, and error rate do not necessarily correspond to the perceived usefulness and user satisfaction.

7 Conclusion

In this work, we demonstrated that motion information from hardware sensors available in mobile devices can be used to tell computers and humans apart. Due to several limitations such as smaller screens and keyboards, traditional captcha schemes designed for desktop computers are often difficult to solve on smartphones, smartwatches, and other kinds of mobile devices. In order to tackle the challenges implied by these constraints, we designed a novel captcha scheme and evaluated it against already existing approaches found in the Internet ecosystem and in scientific literature. Our results indicate that sensor-based captchas are a suitable alternative when deployed on mobile devices as they perform well on usability metrics such as user satisfaction, accuracy, error rate, and solving time.

As our scheme requires users to perform gestures with a device in their hand, we plan to conduct a longitudinal field study to collect evidence on the feasibility of motion input in the wild (i. e., in situations where users are constrained by environmental conditions and unobtrusive interactions with their device) as well as involving wearables as input devices. For future work, we aim to iteratively improve the design and number of challenges. Although most gestures of user study were suitable, their movements need to be revised for everyday use and the entropy need to be increased by new gestures. Additionally, users would benefit from images or animations showing the challenge. Participants of our study agreed with Kluever et al. [13] that images and animations presenting a challenge are more enjoyable. Finally, conducting a long term study with participants using our mechanism regularly may confirm our findings on habituation effects.

References

1. Inside ReCaptcha. `https://github.com/neuroradiology/InsideReCaptcha`, accessed: 2016-03-01
2. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J.,

Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software. In: ECML PKDD Workshop. pp. 108–122 (2013)

3. Bursztein, E., Aigrain, J., Moscicki, A., Mitchell, J.C.: The end is nigh: Generic solving of text-based captchas. In: 8th USENIX Workshop on Offensive Technologies (WOOT 14) (2014)
4. Bursztein, E., Martin, M., Mitchell, J.: Text-based captcha strengths and weaknesses. In: Proceedings of the 18th ACM conference on Computer and communications security. pp. 125–138. ACM (2011)
5. Bursztein, E., Moscicki, A., Fabry, C., Bethard, S., Mitchell, J.C., Jurafsky, D.: Easy does it: More usable captchas. In: Proceedings of the 32nd annual ACM conference on Human factors in computing systems. pp. 2637–2646. ACM (2014)
6. Fidas, C.A., Voyiatzis, A.G., Avouris, N.M.: On the necessity of user-friendly captcha. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2623–2626. ACM (2011)
7. Gao, S., Mohamed, M., Saxena, N., Zhang, C.: Emerging Image Game CAPTCHAs for Resisting Automated and Human-Solver Relay Attacks. In: 31st Annual Computer Security Applications Conference. ACSAC, ACM (2015)
8. Google Inc.: Introducing noCAPTCHA. `http://goo.gl/x7N7qt`, 2016-03-01
9. Google Inc.: reCAPTCHA – Easy on Humans Hard on Bots. `https://www.google.com/recaptcha/intro/index.html`, accessed: 2016-03-01
10. He, H.: HAR on Smartphones Using Various Classifiers (2013)
11. Hupperich, T., Maiorca, D., Kührer, M., Holz, T., Giacinto, G.: On the Robustness of Mobile Device Fingerprinting. In: Proceedings of the 31st Annual Computer Security Applications Conference. ACSAC, ACM (2015)
12. Jiang, N., Dogan, H.: A gesture-based captcha design supporting mobile devices. In: Proceedings of the 2015 British HCI Conference. pp. 202–207. ACM (2015)
13. Kluever, K.A., Zanibbi, R.: Balancing usability and security in a video captcha. In: 5th Symposium on Usable Privacy and Security. SOUPS, ACM (2009)
14. Reynaga, G., Chiasson, S.: The usability of captchas on smartphones. In: Security and Cryptography (SECRYPT) 2013 (2013)
15. Reynaga, G., Chiasson, S., van Oorschot, P.C.: Exploring the usability of captchas on smartphones: Comparisons and recommendations. In: NDSS Workshop on Usable Security USEC 2015. NDSS (2015)
16. Sinofsky, S.: Supporting sensors in Windows 8, `http://blogs.msdn.com/b/b8/archive/2012/01/24/supporting-sensors-in-windows-8.aspx,2016-04-24`,
17. Von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: Captcha: Using hard ai problems for security. In: Advances in Cryptology – EUROCRYPT 2003. Springer (2003)
18. Xu, Y., Reynaga, G., Chiasson, S., Frahm, J.M., Monrose, F., van Oorschot, P.: Security Analysis and Related Usability of Motion-Based CAPTCHAs: Decoding Codewords in Motion. IEEE TDSC 11(5) (Sept 2014)
19. Xu, Y., Reynaga, G., Chiasson, S., Frahm, J.M., Monrose, F., Van Oorschot, P.: Security and usability challenges of moving-object captchas: decoding codewords in motion. In: 21st USENIX Security Symposium. pp. 49–64 (2012)
20. Yan, Jeff and El Ahmad, Ahmad Salah: Usability of CAPTCHAs or usability issues in CAPTCHA design. In: Proceedings of the 4th symposium on Usable privacy and security. pp. 44–52. ACM (2008)